Plugin Directory

source: ppv-live-webcams / trunk / inc / shortcodes.php

Last change on this file was 3487812, checked in by videowhisper, 8 days ago

Webcam App integration (Simple Broadcast mode): add webcamapp.php trait, React app assets in static/webcamapp/, routing fixes for interface=webcamapp group mode, readme updated with Webcam App links and demo; remove legacy Flash SWF files and video-js

File size: 499.0 KB

```
Line
1  <?php
2  namespace VideoWhisper\LiveWebcams;
3
4  if ( ! defined( 'ABSPATH' ) ) {
5          exit; // Exit if accessed directly
6  }
7
8  trait Shortcodes {
9
10         // ! Shortcode Implementation
11
12
13         static function enqueueUI() {
14                 wp_enqueue_script( 'jquery' );
15
16                 wp_enqueue_style( 'fomantic-
ui', dirname( plugin_dir_url( __FILE__ ) ) . '/scripts/semantic/semantic.min.css' );
17                 wp_enqueue_script( 'fomantic-
ui', dirname( plugin_dir_url( __FILE__ ) ) . '/scripts/semantic/semantic.min.js', array( 'jquery' ) );
18         }
19
20
21         static function videowhisper_match_form($atts)
22         {
23
24         $options = self::getOptions();
25
26         // shortocode attributes
27         $atts = shortcode_atts(
28                 array(
29                         'postid' => 0,
30                         'registration' => 0,
31                 ),
32                 $atts,
33                 'videowhisper_match_form'
34         );
35         $atts = array_map('esc_attr', $atts);
36
37
38         if (!is_user_logged_in() && !$atts['registration']) return 'Login required to edit and save match
criteria!';
39
40         $current_user = wp_get_current_user();
41
42
43         $userID =  $current_user->ID ;
```

```
Line
44
45          $postID = 0;
46          if ($atts['postid']) $postID = intval($atts['postid']);
47
48          self::enqueueUI();
49
50          $contentCode = '';
51
52          if ( !$atts['registration'] ) $contentCode .= '<form class="ui
' . esc_attr($options['interfaceClass']) . ' form inline segment" method="post" enctype="multipart/form-
data" action="' . esc_url(self::getCurrentURL()) . '" id="addForm"
name="addForm">' . wp_nonce_field('videowhisper_match_form', 'match_form_nonce', true, false);
53
54          $sideCode = [];
55
56                          foreach ( ['me', 'you'] as $side)
57                          {
58                          $sideCode[$side] = '';
59
60                          $fieldex = '';
61                          if ($side == 'you') $fieldex = 'm_'; //match
62
63
if ($options['profileFields'] ?? false) if ( is_array($options['profileFields']) ||
 $options['profileFields'] instanceof Countable ) if ( count($options['profileFields']) ) foreach ( $opti
ons['profileFields'] as $field => $parameters ) if ( isset( $parameters['match'] ) ) if ( $parameters['ma
tch'] != 'mirror' || $side == 'me' )
64                          {
65
66                                          $fieldName = $fieldex . sanitize_title( trim( $field ) );
67
68                                          if ( isset($parameters['instructions']) ) {
69                                                  $htmlInstructions = ' data-
tooltip="' . htmlspecialchars( stripslashes( $parameters['instructions'] ) ) . '"';
70                                          } else {
71                                                  $htmlInstructions = '';
72                                          }
73
74                                          $sideCode[$side] .= '<div
class="field"' . $htmlInstructions . '>';
75
76                                          $iconCode = '';
77
$icon = isset($parameters['icon']) ? stripslashes( $parameters['icon'] ) : ( $options['profileFieldIcon']
 ?? false );
78                                          if ($icon) $iconCode .= '<i class="' . $icon . ' icon"></i> ';
79
80                                          if (!isset($parameters['hideLabel'])) $sideCode[$side] .= '<label
for="' . $fieldName . '">' . $iconCode . $field . '</label>';
81
82                                          //default
83                                          $fieldValue = '';
84
if ( isset($parameters['default']) ) $fieldValue = sanitize_text_field( $parameters['default']);
85
86                                          // save data
87                                          if ( !$atts['registration'] )
88                                          {
89                                                  if ( isset( $_POST[ $fieldName ] ) ) {
90                                                          if ( $_POST['matchFor'] ?? false ) {
91
if ( in_array( $parameters['type'], array('checkboxes', 'multiselect', 'language') ) )
92                                                                                  {
93
$tags = (array) $_POST[ $fieldName ];
```

```
94                                                                  if ( is_array( $tags ) ) {
95  foreach ( $tags as &$tag ) {
96  $tag = sanitize_text_field( $tag );
97                                                                          }
98                                                                          unset( $tag );
99                                                                  } else {
100 $tags = sanitize_text_field( $tags );
101                                                                 }
102                                                                 // Verify nonce before updating
    data
103 if (!isset($_POST['match_form_nonce']) ||
     !wp_verify_nonce($_POST['match_form_nonce'], 'videowhisper_match_form')) {
104                                                                         // Nonce verification
    failed
105                                                                         continue;
106                                                                 }
107
108 if ($postID) update_post_meta( $postID, 'vwf_' . $fieldName, $tags );
109 else update_user_meta( $userID, 'vwf_'  . $fieldName, $tags );
110
111                                                                 } else {
112                                                                 if ($postID)
    update_post_meta( $postID, 'vwf_' . $fieldName, sanitize_text_field( $_POST[ $fieldName ] ) );
113 else update_user_meta( $userID, 'vwf_' . $fieldName, sanitize_text_field( $_POST[ $fieldName ] )  );
114                                                                 }
115                                                         }
116                                                 }
117
118                                                         // get data
119 if ( in_array( $parameters['type'], array('checkboxes', 'multiselect', 'language') ) )
120                                                 {
121 if ($postID) $fieldValue = get_post_meta( $postID, 'vwf_' . $fieldName, true );
122 else $fieldValue = get_user_meta( $userID, 'vwf_' . $fieldName, true );
123
124                                                         if ( ! $fieldValue ) {
125                                                                 $fieldValue = array();
126                                                         }
127                                                         if ( ! is_array( $fieldValue ) ) {
128                                                                 $fieldValue = array( $fieldValue );
129                                                         }
130                                                 } else {
131 if ($postID) $fieldValue = get_post_meta( $postID, 'vwf_' . $fieldName, true );
132 else $fieldValue = get_user_meta( $userID, 'vwf_' . $fieldName, true );
133                                                 }
134
135                                         }
136
137                                         $fieldOptions = [];
138                                         if ( isset($parameters['options']) )
139                                         {
140 $parameters['options'] = stripslashes( $parameters['options'] ?? '' );
141                                         //if $parameters['options'] contains | get      $fieldOptions by
    split by | otherwise by /
```

```
Line
142 $fieldOptions = strpos($parameters['options'], '|') ? explode('|', $parameters['options']) : explode('/', $parameters['options']);
143                                         }
144
145
146                                         // form
147                                         switch ( $parameters['type'] ) {
148
149                                                 case 'select';
150                                                         $sideCode[$side] .= '<SELECT class="ui dropdown search clearable v-select" name="' . $fieldName . '" id="' . $fieldName . '">';
151                                                         $sideCode[$side] .= '<OPTION value="" ' . ( ! $fieldValue ? 'selected' : '' ) . '> ' . ($side == 'me' ? '-' : 'Any') . ' </OPTION>';
152
153                                                         foreach ( $fieldOptions as $fieldOption ) {
154                                                                 $sideCode[$side] .= '<OPTION value="' . htmlspecialchars( stripslashes( $fieldOption ) ) . '" ' . ( $fieldOption == $fieldValue ? 'selected' : '' ) . '>' . htmlspecialchars( $fieldOption ) . '</OPTION>';
155                                                         }
156
157                                                         $sideCode[$side] .= '</SELECT>';
158                                                 break;
159
160                                                 case 'multiselect':
161                                                         $sideCode[$side] .= '<SELECT MULTI class="ui dropdown search clearable v-select" name="' . $fieldName . '[]" id="' . $fieldName . '" multiple>';
162
163                                                         if (is_array($fieldOptions))
164                                                         foreach ( $fieldOptions as $fieldOption ) {
165                                                                 $sideCode[$side] .= '<OPTION value="' . htmlspecialchars( stripslashes( $fieldOption ) ) . '" ' . ( is_array($fieldValue) && in_array( $fieldOption, $fieldValue ) ? 'selected' : '' ) . '>' . htmlspecialchars( $fieldOption ) . '</OPTION>';
166                                                         };
167                                                         $sideCode[$side] .= '</SELECT>';
168
169                                                 case 'checkboxes';
170 if (is_array($fieldOptions)) foreach ( $fieldOptions as $fieldOption ) {
171                                                                 $sideCode[$side] .= '<div class="field"><div class="ui toggle checkbox">
172   <input type="checkbox" name="' . $fieldName . '[]" value="' . htmlspecialchars( stripslashes( $fieldOption ) ) . '" ' . ( ( is_array($fieldValue) && in_array( $fieldOption, $fieldValue ) ) ? 'checked' : '' ) . '>
173   <label>' . htmlspecialchars( $fieldOption ) . '</label></div></div>';
174                                                         } else $sideCode[$side] .= '<div>Empty or incorrect field options: ' . html_esc($parameters['options']). '</div>';
175
176                                                 break;
177
178                                         }
179                                         $sideCode[$side]  .= '</div>'; //field
180
181
182                                         // if ($parameters['instructions']) $contentCode .= '<BR>' . htmlspecialchars($parameters['instructions']);
183                                 }
184                         }
185
186                         $contentCode .= '<h4 class="ui dividing header">I Am</h4>' . $sideCode['me'] ;
187
188                         if ($sideCode['you']) $contentCode .= '<h4 class="ui dividing header">Looking For</h4>' . $sideCode['you'] ;
```

```
189
190                                 $contentCode .='<input type="hidden" name="matchFor" id="matchFor"
value="' . ( $postID ? $postID : ( $userID ? $userID : -1 ) ) . '">';
191
192                                 if ( !$atts['registration'] ) $contentCode .= '<br><input class="ui
button primary" type="submit" name="save" id="save" value="' . __( 'Save', 'ppv-live-webcams' ) . '" />
        </form>';
193
194
195         $contentCode .='<script>jQuery(document).ready(function(){
jQuery(".ui.dropdown.clearable:not(.multi,.fpsDropdown)").dropdown({"clearable": true}); });</script>';
196                 return $contentCode;
197         }
198
199         static function videowhisper_sms_number($atts)
200         {
201         //fill number for SMS notifications
202
203         $options = self::getOptions();
204
205         if (!is_user_logged_in()) return 'Login required to fill SMS number!';
206
207         $current_user = wp_get_current_user();
208
209         // shortocode attributes
210         $atts = shortcode_atts(
211                 array(
212                 ),
213                 $atts,
214                 'videowhisper_sms_number'
215         );
216         $atts = array_map('esc_attr', $atts);
217
218         $htmlCode = '';
219
220         if ($_POST['sms_update'] ?? false)
221         {
222                 $nonce = $_POST['_wpnonce'] ?? '';
223                 if ( ! wp_verify_nonce( $nonce, 'sms_number_update' ) )
224                 {
225                         return '<div class="ui error message">Invalid security token!</div>';
226                 }
227
228                 if ( isset($_POST['sms_number']) ) update_user_meta( $current_user-
>ID, 'sms_number', sanitize_text_field( $_POST['sms_number'] ) );
229         }
230
231                 $current_url = self::getCurrentURL();
232
233                 $htmlCode .= '<form class="ui ' . esc_attr($options['interfaceClass']) . ' form inline
segment" method="post" enctype="multipart/form-data" action="" id="addForm"
name="addForm">' . wp_nonce_field('sms_number_update', '_wpnonce', true, false) . '<div class="two
fields">';
234
235                 $htmlCode .= '<div class="field">
236                   <label>' . esc_html__('SMS Mobile Number', 'ppv-live-webcams') . '</label>
237                   <input type="text" name="sms_number" id="sms_number"
value="' . esc_attr( get_user_meta( $current_user->ID, 'sms_number', true ) ) . '" />
238                   <input type="hidden" name="sms_update" id="sms_update" value="1" />
239                   </div>';
240                 $htmlCode .= '<div class="field"><label>' . esc_html__('Update', 'ppv-live-
webcams') . '</label><input type="submit" id="submitButton" name="submitButton" class="ui submit button"
value="' . esc_attr__('Save', 'ppv-live-webcams') . '" ></div>';
241                 $htmlCode .=  '</div><div
class="field">' . esc_html( $options['sms_instructions'] ) . '</div>';
```

```
Line
242                 $htmlCode .= '</form>';
243
244                 return $htmlCode;
245         }
246
247         static function videowhisper_room_rate($atts)
248         {
249                 //display room ratio & bonus status if enabled
250
251                 // shortocode attributes
252                 $atts = shortcode_atts(
253                         array(
254                                 'roomid' => '',
255                         ),
256                         $atts,
257                         'videowhisper_room_rate'
258                 );
259                 $atts = array_map('esc_attr', $atts);
260
261                 $postID = intval( $atts['roomid'] );
262                 if (!$postID) return 'videowhisper_room_ratio: missing roomid';
263
264
265                 $options = self::getOptions();
266
267                 self::roomStatsUpdate($postID);
268                 $ratio = self::performerRatio( '', $options, $postID );
269
270                 self::enqueueUI();
271
272                 $htmlCode = '<div class="ui ' . esc_attr($options['interfaceClass']) . ' segment">';
273
274                 $htmlCode .= __('Current Earning Rate', 'ppv-live-webcams') . ': ' . esc_html($ratio);
275
276                 if ($options['statsBonusRate'] && $options['statsBonusOnline'])
277                 {
278                         $timeOnline = intval(get_post_meta( $postID, 'statsTimeOnline', true ));
279
280                         $htmlCode .= '<div class="ui ' . esc_attr($options['interfaceClass']) . '
indicating progress" data-value="' . esc_attr($timeOnline) . '" data-
total="' . intval($options['statsBonusOnline']) . '" id="bonusRatio">
281   <div class="bar">
282     <div class="progress"></div>
283   </div>
284   <div class="label">' . esc_html__('Bonus Earning Rate', 'ppv-live-webcams') . ': <i class="gift icon">
</i>' . esc_html($options['statsBonusRate']) . ' <i class="user clock icon">
</i>' . esc_html(self::humanDuration(intval($options['statsBonusOnline']))) . '</div>
285 </div>';
286
287                 $htmlCode .='<pre style="display:none">
288 <script>
289 jQuery(document).ready(function(){
290 jQuery(".ui.progress").progress();
291 })
292 </script>
293 </pre>
294                 ';
295
296                 }
297
298                 $htmlCode .= '</div>';
299
```

```
300                 return $htmlCode;
301         }
302
303
304         static function roomStatsUpdate($postID)
305         {
306
307         if (!$postID) return;
308
309         $options = self::getOptions();
310         if ( ! self::timeTo( $postID . '-roomStats', 60, $options ) ) return;
311
312         $duration = 86400 * $options['statsDuration'];
313         if (!$duration) $duration = 2592000; //30 days default
314         $lastTime = time() - $duration;
315
316         global $wpdb;
317         $table_sessions = $wpdb->prefix . 'vw_vmls_sessions';
318         $table_calls = $wpdb->prefix . 'vw_vmls_private';
319
320         $cnd = '';
321         if ($postID) $cnd = "rid='$postID' AND";
322
323
324                 $timeOnline = 0;
325
326                 //group chats
327                 $sql      = $wpdb->prepare("SELECT * FROM $table_sessions WHERE $cnd broadcaster=%d AND
sdate > %d", 1, $lastTime); //assign by session start time
328                 $sessions = $wpdb->get_results( $sql );
329
330                 if ( $wpdb->num_rows > 0 ) foreach ( $sessions as $session )
331                 {
332
333                                 $timeOnline += $session->edate > 0 ? intval( $session->edate - $session-
>sdate ) : 0 ;
334
335                 }
336
337                 //calls
338                 $sql      = $wpdb->prepare("SELECT * FROM $table_calls WHERE $cnd csdate >
%d", $lastTime);
339                 $sessionsC = $wpdb->get_results( $sql );
340
341                 //sessions
342                 if ( $wpdb->num_rows > 0 ) foreach ( $sessionsC as $session )
343                 {
344                         $timeOnline +=  $session->psdate > 0 ? intval( $session->pedate - $session-
>psdate ) : 0 ;
345                 }
346
347                 update_post_meta( $postID, 'statsTimeOnline', $timeOnline );
348
349                 if ($options['statsBonusOnline'] && $options['statsBonusRate'])
350                 if ( $timeOnline > $options['statsBonusOnline'] )
351                         update_post_meta( $postID, 'vw_bonusRatio', $options['statsBonusRate'] );
352                 else delete_post_meta( $postID, 'vw_bonusRatio' ); //remove bonus ratio
353
354         }
355
356
357         static function videowhisper_reports($atts)
```

```
358         {
359                 // Check user permissions first
360                 if (!is_user_logged_in()) {
361                         return '<div class="ui error message">You must be logged in to view reports.</div>';
362                 }
363
364                 $current_user = wp_get_current_user();
365                 $options = self::getOptions();
366
367                 // shortocode attributes
368                 $atts = shortcode_atts(
369                         array(
370                                 'roomid' => '',
371                         ),
372                         $atts,
373                         'videowhisper_reports'
374                 );
375                 $atts = array_map('esc_attr', $atts);
376
377                 $postID = intval( $atts['roomid'] );
378
379                 // Security check: verify user has access to this room's reports
380                 if ($postID) {
381                         // Only allow room owner, moderators or administrators to view reports
382                         if (!self::isPerformer($current_user->ID, $postID) &&
383                                 !self::isAuthor($postID) &&
384                                 !self::isModerator($current_user->ID, $options) &&
385                                 !current_user_can('manage_options')) {
386                                 return '<div class="ui error message">You do not have permission to view these reports.</div>';
387                         }
388                 }
389
390                 self::enqueueUI();
391                 $htmlCode = '';
392
393                 wp_enqueue_script( 'chart-js', dirname( plugin_dir_url( __FILE__ ) ) . '/scripts/chartjs/chart.min.js', ['jquery'] );
394
395                 // Sanitize and validate date inputs
396                 $startDate = isset($_POST['startDate']) ? sanitize_text_field($_POST['startDate']) : '';
397                 $endDate = isset($_POST['endDate']) ? sanitize_text_field($_POST['endDate']) : '';
398
399                 // Convert to timestamps and validate
400                 $tS = strtotime($startDate);
401                 $tE = strtotime($endDate);
402
403                 // Validate timestamps
404                 if (!$tS || $tS === false) {
405                         $startDate = date("M d, Y", strtotime("-12 month"));
406                         $tS = strtotime($startDate);
407                 }
408
409                 if (!$tE || $tE === false) {
410                         $endDate = date("M d, Y", time());
411                         $tE = strtotime($endDate);
412                 }
413
414                 // Validate deltaDate input (group by parameter)
415                 $validDeltas = array('month', 'week', 'day');
```

```
$deltaDate = isset($_POST['deltaDate']) ? sanitize_text_field($_POST['deltaDate']) : 'month';
                if (!in_array($deltaDate, $validDeltas)) {
                        $deltaDate = 'month';
                }

                // Validate report type
                $validTypes = array('time', 'value');
                $type = isset($_POST['type']) ? sanitize_text_field($_POST['type']) : 'time';
                if (!in_array($type, $validTypes)) {
                        $type = 'time';
                }

                $htmlCode .= '
                <form class="ui ' . esc_attr($options['interfaceClass']) . ' form inline" method="post" enctype="multipart/form-data" action="' . esc_url(add_query_arg( array(), $wp->request ?? '' )) . '" id="addForm" name="addForm">
  <div class="five fields">

    <div class="field">
      <label>' . __('Start Date', 'ppv-live-webcams') . '</label>
      <div class="ui calendar" id="rangestart">
        <div class="ui input left icon">
          <i class="calendar icon"></i>
          <input type="text" name="startDate" placeholder="Start" value="'. $startDate .'">
        </div>
      </div>
    </div>

        <div class="field">
          <label>' . __('End Date', 'ppv-live-webcams') . '</label>
          <div class="ui calendar" id="rangeend">
            <div class="ui input left icon">
              <i class="calendar icon"></i>
              <input type="text" name="endDate" placeholder="End" value="'. $endDate .'">
            </div>
          </div>
        </div>

 <div class="field">
  <label>' . __('Group', 'ppv-live-webcams') . '</label>
  <select name="deltaDate" id="deltaDate" class="ui fluid dropdown v-select">
  <OPTION value="month" ' . ( $deltaDate == 'month' ? 'selected' : '' ) . '>' . __('Monthly', 'ppv-live-webcams') . '</OPTION>
  <OPTION value="week" ' . ( $deltaDate == 'week' ? 'selected' : '' ) . '>' . __('Weekly', 'ppv-live-webcams') . '</OPTION>
  <OPTION value="day" ' . ( $deltaDate == 'day' ? 'selected' : '' ) . '>' . __('Daily', 'ppv-live-webcams') . '</OPTION>
  </select>
 </div>

  <div class="field">
  <label>' . __('Type', 'ppv-live-webcams') . '</label>
  <select name="type" id="type" class="ui fluid dropdown v-select">
  <OPTION value="time" ' . ( $type == 'time' ? 'selected' : '' ) . '>' . __('Time', 'ppv-live-webcams') . '</OPTION>
  <OPTION value="value" ' . ( $type == 'value' ? 'selected' : '' ) . '>' . __('Value', 'ppv-live-webcams') . '</OPTION>
  </select>
 </div>

 <div class="field">
  <label>' . __('Update', 'ppv-live-webcams') . '</label>
```

```
Line
471   <input type="submit" id="submitButton" name="submitButton" class="ui submit button" value="' . __('Update', 'ppv-live-webcams') . '" >
472  </div>
473
474 </div>
475 </form>';
476
477
478                 $htmlCode .= '<div>
479   <canvas id="videowhisperChart"></canvas>
480   </div>';
481
482         global $wpdb;
483         $table_sessions = $wpdb->prefix . 'vw_vmls_sessions';
484         $table_calls = $wpdb->prefix . 'vw_vmls_private';
485
486
487 // Set up time range for reports with secure validation
488 $t1 = $tS; // Already validated
489 $t2 = strtotime($startDate . ' +1 ' . $deltaDate);
490 $endTime = $tE; // Already validated
491
492 // Validate t2 is sane value
493 if (!$t2 || $t2 < $t1 || $t2 > time() + 86400) {
494     $t2 = strtotime($startDate . ' +1 month');
495 }
496
497 // Maximum time range - for security and performance
498 $maxTimeRange = 86400 * 365 * 2; // 2 years
499 if ($endTime - $t1 > $maxTimeRange) {
500     $endTime = $t1 + $maxTimeRange;
501     $endDate = date("M d, Y", $endTime);
502 }
503
504 $tableCode = '<TABLE class="ui table small celled striped">';
505
506 $data = [];
507
508 if ($type == 'value')
509 {
510         $dataSets = [
511         'clientCost' => [ 'label' => 'Client Cost', 'color' => 'rgba(250, 100, 150, 0.8)', 'data' => '' ],
512         'performerShare' => [ 'label' => 'Performer Earnings', 'color' => 'rgba(120, 150, 120, 0.8)', 'data' => '' ],
513         'siteShare' => [ 'label' => 'Site Earnings', 'color' => 'rgba(160, 255, 120, 0.8)', 'data' => '' ],
514
515         'ClientCallCost' => [ 'label' => 'Client Call Costs', 'color' => 'rgba(250, 150, 150, 0.8)', 'data' => '' ],
516         'PerformerCallEarning' => [ 'label' => 'Performer Call Earnings', 'color' => 'rgba(120, 200, 120, 0.8)', 'data' => '' ],
517         'SiteCallShare' => [ 'label' => 'Site Call Earnings', 'color' => 'rgba(160, 255, 170, 0.8)', 'data' => '' ],
518
519         ];
520
521 }
522 else
523 {
524         $dataSets = [
525         'PerformerTime' => [ 'label' => 'Performer Time', 'color' => 'rgba(100, 99, 132, 1)', 'data' => '' ],
```

```
Line
526         'ClientTimeFree' => [ 'label' => 'Client Time Free', 'color' => 'rgba(54, 100, 235, 1)', 'data' => '' ],
527         'ClientTimePaid' => [ 'label' => 'Client Time Paid', 'color' => 'rgba(54, 200, 235, 1)', 'data' => '' ],
528         'PerformerCall' => [ 'label' => 'Performer Call Time', 'color' => 'rgba(200, 100, 150, 1)', 'data' => '' ],
529         'ClientCall' => [ 'label' => 'Client Call Time', 'color' => 'rgba(150, 100, 150, 1)', 'data' => '' ],
530
531         'PerformerAll' => [ 'label' => 'Performer All Time', 'color' => 'rgba(150, 100, 100, 1)', 'data' => '' ],
532         ];
533 }
534
535 $labels = '';
536 $cnd = '';
537 $title = '';
538
539 // Securely build SQL condition for room ID
540 if ($postID) {
541     $cnd = $wpdb->prepare("rid=%d AND", $postID);
542     $post = get_post($postID);
543
544     if ($post) $title = esc_html($post->post_title) . ' : ';
545 }
546
547 while ($t1 < $endTime && $t2 <= $endTime)
548 {
549                 //data segment
550                 $dS = [];
551                 foreach ($dataSets as $key => $dataSet) $dS[$key] = 0;
552
553                 // Use wpdb->prepare to safely construct and execute SQL queries
554                 $sql = $wpdb->prepare(
555                         "SELECT * FROM $table_sessions WHERE $cnd sdate > %d AND sdate < %d",
556                         $t1,
557                         $t2
558                 );
559
560                 $sessions = $wpdb->get_results($sql);
561
562                 //sessions
563                 if ($wpdb->num_rows > 0) foreach ($sessions as $session)
564                 {
565                         if ($type == 'value')
566                         {
567                                 if (!$session->broadcaster)
568                                 {
569                                         $meta = maybe_unserialize($session->meta);
570                                         if (!is_array($meta)) $meta = [];
571
572 if (array_key_exists('cost', $meta)) $dS['clientCost'] += floatval($meta['cost']);
573
574 if (array_key_exists('b_performerEarn', $meta)) $dS['performerShare'] += floatval($meta['b_performerEarn']);
575 else if (array_key_exists('b_performerShare', $meta) && array_key_exists('b_performerRatio', $meta))
576 $dS['performerShare'] += floatval($meta['b_performerShare']) * floatval($meta['b_performerRatio']);
577
578                                         $dS['siteShare'] = $dS['clientCost'] - $dS['performerShare'];
579                                 }
```

```
580                         }
581                         else //time
582                         {
583                                 if ($session->broadcaster)
584                                 {
585                                         $dS['PerformerTime'] += ($session->edate > 0 ? intval($session->edate - $session->sdate) : 0);
586                                         $dS['PerformerAll'] += ($session->edate > 0 ? intval($session->edate - $session->sdate) : 0);
587                                 }
588                                 elseif ($session->rmode == 0) $dS['ClientTimeFree'] += ($session->rsdate ? intval($session->redate - $session->rsdate) : 0);
589                                 else $dS['ClientTimePaid'] += intval($session->redate - $session->rsdate);
590                         }
591                 }
592
593                 //calls - use prepared statement
594                 $sql = $wpdb->prepare(
595                         "SELECT * FROM $table_calls WHERE $cnd csdate > %d AND csdate < %d",
596                         $t1,
597                         $t2
598                 );
599
600                 $sessionsC = $wpdb->get_results($sql);
601
602                 //sessions
603                 if ( $wpdb->num_rows > 0 ) foreach ( $sessionsC as $session )
604                 {
605                         if ($type == 'time')
606                         {
607                                 $dS['PerformerCall'] += ( $session->psdate > 0 ? intval( $session->pedate - $session->psdate ) : 0 );
608                                 $dS['PerformerAll'] += ( $session->psdate > 0 ? intval( $session->pedate - $session->psdate ) : 0 );
609
610                                 $dS['ClientCall'] += ( $session->csdate > 0 ? intval( $session->cedate - $session->csdate ) : 0 );
611                         }
612
613                         if ($type == 'value')
614                         {
615                                 $meta = unserialize( $session->meta, array( 'allowed_classes' => false ) );
616                                 if (!is_array($meta)) $meta = [];
617
618
if (array_key_exists('b_clientCost', $meta)) $dS['ClientCallCost'] += $meta['b_clientCost'];
619
if (array_key_exists('b_performerEarning', $meta)) $dS['PerformerCallEarning'] += $meta['b_performerEarning'];
620
621                                 $dS['SiteCallShare'] = $dS['ClientCallCost'] - $dS['PerformerCallEarning'];
622                         }
623                 }
624
625
626         //data y
627         foreach ($dS as $key => $value)
628         {
629                 $dataSets[$key]['data'] .= ( $dataSets[$key]['data'] !='' ? ', ' : '' ) . $value;
630         }
631
632         //label x
```

```
633         $labels .= ($labels ? ', ' : '') .  '\'' . date( "M d, Y", $t1) .  '\'';
634 
635         //list
636         $tableCode .= '<tr><td>' . date("M d, Y", $t1) . ' - ' . date("M d, Y", $t2) . '</td><td>';
637 
638         foreach ($dataSets as $key => $dataSet) $tableCode .= $dataSet['label'] . ': ' .
    ( $type == 'time' ? self::humanDuration($dS[$key]) : $dS[$key] . ' ' . $options['currency'])  . '<br>';
639 
640         $tableCode .= '</td></tr>';
641 
642         //next
643         $t1 = strtotime( date( "Y-m-d", $t1) . ' +1 ' . $deltaDate);
644         $t2 = min( strtotime( date( "Y-m-d", $t2) . ' +1 ' . $deltaDate), $endTime);
645 }
646 
647 $tableCode .= '</TABLE>';
648 
649 //data
650 
651 $dataCode = 'const data = {
652   labels: [' . $labels . '],
653   datasets: [
654 ';
655         foreach ($dataSets as $key => $dataSet)
656         $dataCode .= '{
657       label: \'' . $dataSet['label'] .  '\',
658       data: [' . $dataSet['data'] .  '],
659       backgroundColor: \'' . $dataSet['color'] . '\',
660     },';
661 
662 $dataCode .= ']
663 };';
664 
665         $htmlCode .= '<pre style="display:none">
666         <script>
667 jQuery(document).ready(function(){
668 
669 jQuery(\'#rangestart\').calendar({
670   type: \'date\',
671   endCalendar: jQuery(\'#rangeend\')
672 });
673 
674 jQuery(\'#rangeend\').calendar({
675   type: \'date\',
676   startCalendar: jQuery(\'#rangestart\')
677 });
678 
679 ' . $dataCode . '
680 
681 const config = {
682   type: \'bar\',
683   data: data,
684   options: {
685     plugins: {
686       title: {
687         display: true,
688         text: \'' . $title . ($type == 'time' ?  __('Time', 'ppv-live-webcams')  :  __('Value', 'ppv-
    live-webcams') ) . '\'
689       },
690     },
691     responsive: true,
692     scales: {
```

```
      x: {
        stacked: true,
      },
      y: {
            beginAtZero: true,
        stacked: true
      }
    }
  }
};

 const vwChart = new Chart( document.getElementById(\'videowhisperChart\').getContext(\'2d\'), config );


 //

})



</script>
</pre>
        ';

                $htmlCode .= $tableCode;

                return $htmlCode;

        }



// tools

        static function to62( $num, $b = 62 ) {
                $base = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
                $r    = $num % $b;
                $res  = $base[ $r ];
                $q    = floor( $num / $b );
                while ( $q ) {
                        $r   = $q % $b;
                        $q   = floor( $q / $b );
                        $res = $base[ $r ] . $res;
                }
                return $res;
        }


        static function to10( $num, $b = 62 ) {

                $base  = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';

                $limit = strlen( $num );
                if (!$limit) return '';

                $res   = strpos( $base, $num[0] );
                for ( $i = 1;$i < $limit;$i++ ) {
                        $res = $b * $res + strpos( $base, $num[ $i ] );
                }

                return $res;
```

```
754         }
755
756
757 static function sectionsLayout( $header, $content, $active = '', $layout = 'tabs', $intro = '' ) {
758                 // renders section based on arrays of $header (titles) and $content with matching keys
759
760                 if ( ! $layout ) {
761                         $layout = 'auto';
762                 }
763                 $headerCode  = '';
764                 $contentCode = '';
765
766                 $isMobile = (bool) preg_match( '#\b(ip(hone|od|ad)|android|opera m(ob|in)i|windows
(phone|ce)|blackberry|tablet|s(ymbian|eries60|amsung)|p(laybook|alm|rofile/midp|laystation
portable)|nokia|fennec|htc[\-_]|mobile|up\.browser|[1-4][0-9]{2}x[1-4][0-9]
{2})\b#i', $_SERVER['HTTP_USER_AGENT'] );
767
768                 $options = get_option( 'VWliveWebcamsOptions' );
769
770                 if ( $layout == 'auto' ) {
771                         if ( $isMobile ) {
772                                 $layout = 'accordion';
773                         } else {
774                                 $layout = 'tabs';
775                         }
776                 }
777
778                         // start
779                 switch ( $layout ) {
780                         case 'tabs':
781                                 if ( $intro ) {
782                                         $headerCode .= '<div class="ui ' . $options['interfaceClass'] . '
horizontal fluid label">' . $intro . '</div>';
783                                 }
784                                 $headerCode .= '<div class="ui ' . $options['interfaceClass'] . '
pointing menu wrappedtabs">';
785
786                                 $contentCode = '';
787                                 break;
788
789                         case 'accordion':
790                                 $headerCode  = '<div class="ui
' . esc_attr($options['interfaceClass']) . ' segment">';
791                                 $contentCode = '<div class="ui
' . esc_attr($options['interfaceClass']) . ' fluid accordion vwSections">';
792                                 if ( $intro ) {
793                                         $contentCode .= '<div class="ui
' . esc_attr($options['interfaceClass']) . ' fluid label">' . esc_html($intro) . '</div>';
794                                 }
795
796                                 break;
797
798                         case 'chapters':
799                                 $headerCode = '<div class="ui
' . esc_attr($options['interfaceClass']) . ' menu vertical fluid">';
800                                 if ( $intro ) {
801                                         $headerCode .= '<div class="ui
' . esc_attr($options['interfaceClass']) . ' fluid label">' . esc_html($intro) . '</div>';
802                                 }
803
804                                 $contentCode = '';
805                                 break;
806                 }
```

```
                // items
                foreach ( $header as $section => $title ) {
                        switch ( $layout ) {
                                case 'tabs':
                                        $headerCode .= '<a class="item
' . ( $active == $section ? 'active' : '' ) . '" data-tab="' . $section . '">' . $title . '</a>';

                                        $contentCode .= '<div class="ui
' . esc_attr($options['interfaceClass']) . ' tab segment ' . ( $active == $section ? 'active' : '' ) . '"
data-tab="' . esc_attr($section) . '">';
                                        $contentCode .= $content[ $section ];
                                        $contentCode .= '<br style="clear:both"></div>';

                                        break;

                                case 'accordion':
                                        $contentCode .= '<div class="ui title
' . ( $active == $section ? 'active' : '' ) . '">
      <i class="dropdown icon"></i>
      ' . $title . '
    </div>
    <div class="content ' . ( $active == $section ? 'active' : '' ) . '">
      ' . $content[ $section ] . '
    </div>';

                                        break;

                                case 'chapters':
                                        $headerCode .= '<a class="ui item
' . ( $active == $section ? 'active' : '' ) . '" href="#' . $section . '">' . $title . '</a>';

                                        $contentCode .= '<a name="' . $section . '"></a>';
                                        $contentCode .= '<div class="ui header">' . $title . '</a>
</div>';
                                        $contentCode .= '<div class="ui
' . $options['interfaceClass'] . ' segment ' . ( $active == $section ? 'active' : '' ) . '">';
                                        $contentCode .= $content[ $section ];
                                        $contentCode .= '<br style="clear:both">';
                                        $contentCode .= '</div>';

                                        break;
                        }
                }

                // end
                switch ( $layout ) {
                        case 'tabs':
                                $headerCode .= '</div>';

                                $contentCode .= '
                        <style>
.wrappedtabs{
  display: flex;
  flex-direction: row;
  flex-wrap: wrap;
}
                        </style>
        <script>
jQuery(document).ready(function(){

        var vwTabs = jQuery(".wrappedtabs.menu .item");
        try{ vwTabs.tab()} catch(error)
```

```
        {
        console.log("Interface error Tabs", error, vwTabs);
        }
});
</script>';

                                break;

                        case 'accordion':
                                $contentCode .= '</div></div>
                        <script>
jQuery(document).ready(function(){

        jQuery(".ui.accordion.vwSections").accordion({animateChildren: false});

});
</script>
';
                                break;

                        case 'chapters':
                                $headerCode .= '</div>';
                                break;
                }

                $contentCode .= '
<style>
.ui.title, .ui.label, .ui.compact.button{
  height: auto !important;
}
</style>';

                return $headerCode . $contentCode;
        }


        static function videowhisper_callnow( $atts ) {
                $options = get_option( 'VWliveWebcamsOptions' );

                // shortocode attributes
                $atts = shortcode_atts(
                        array(
                                'roomid' => '',
                        ),
                        $atts,
                        'videowhisper_callnow'
                );
                $atts = array_map('esc_attr', $atts);

                $postID = intval( $atts['roomid'] );

                if ( ! $postID ) {
                        return 'videowhisper_callnow: missing roomid';
                }

                $post = get_post( $postID );
                if ( ! $post ) {
                        return 'videowhisper_callnow: invalid roomid';
                }

                if ( ! is_user_logged_in() ) {
```

```
924                         return '<a class="ui ' . esc_attr($options['interfaceClass']) . ' button" href="' . esc_url(get_permalink( $options['p_videowhisper_login'] )) . '"> <i class="sign in icon"></i> ' . esc_html__( 'Login to Call', 'ppv-live-webcams' ) . '</a>';
925                 }
926
927                 $name = sanitize_file_name( $post->post_title );
928                 $url  = self::roomURL( $name );
929
930                 self::enqueueUI();
931                 $enterCode = "<!--VideoWhisper.com/PaidVideochat.com/videowhisper_callnow/#$postID-->";
932
933                 $urlCall = add_query_arg( 'vwsm', 'private', $url );
934
935                 $requests_disable = self::is_true( get_post_meta( $post->ID, 'requests_disable', true ) );
936                 $clientCPM        = self::clientCPM( $name, $options, $postID );
937
938                 if ( ! $requests_disable ) {
939                         $enterCode .= '<a class="ui ' . esc_attr($options['interfaceClass']) . ' button" href="' . esc_url($urlCall) . '">' . esc_html__( 'Call Now', 'ppv-live-webcams' ) . ' ' . ( $clientCPM ? esc_html($clientCPM . $options['currencypm']) : '' ) . '</a>';
940                 } else {
941                         $enterCode .= '<a class="ui ' . $options['interfaceClass'] . ' button" href="' . $url . '">' . __( 'Join', 'ppv-live-webcams' ) . '</a>';
942                 }
943
944                 return $enterCode;
945         }
946
947
948         static function videowhisper_streams( $atts ) {
949
950                 $options = self::getOptions();
951
952                 // shortocode attributes
953                 $atts = shortcode_atts(
954                         array(
955                                 'roomid' => '',
956                         ),
957                         $atts,
958                         'videowhisper_streams'
959                 );
960                 $atts = array_map('esc_attr', $atts);
961
962                 $postID = intval( $atts['roomid'] );
963
964                 if ( ! $postID ) {
965                         return 'videowhisper_streams: missing roomid';
966                 }
967
968                 $post = get_post( $postID );
969                 if ( ! $post ) {
970                         return 'videowhisper_streams: invalid roomid';
971                 }
972
973                 $current_user = wp_get_current_user();
974                 $username     = self::performerName( $current_user, $options );
975
976                 self::enqueueUI();
977                 $this_page = self::getCurrentURL();
978                 $htmlCode  = '';
979                 $status = '';
980
```

```
Line
981
982                 //if current user is not room performer or moderator or admin return error
983                 if ( ! self::isPerformer( $current_user->ID, $postID ) && !self::isAuthor( $postID ) && !self::isModerator( $current_user->ID, $options ) && ! current_user_can( 'manage_options' ) ) {
984                         return 'videowhisper_streams: not allowed';
985                 }
986
987
988
989                 $htmlCode .= '<div class="ui accordion fluid vwStreams ' . $options['interfaceClass'] . '">';
990
991
992                         if ( $options['reStreams'] )
993                         {
994
995
996                                 $active = '';
997                                 if ( isset( $_POST['reStream'] ) || isset( $_GET['delete_stream'] ) || isset( $_GET['set_performer'] ) || isset( $_GET['update_streams'] ) || isset( $_GET['streams'] ) ) $active = 'active';
998
999                                 $htmlCode .= '<div class="title ' . $active . '"> <i class="camera video icon"></i> ' . __( 'ReStream RTSP/RTMP', 'ppv-live-webcams' ) . '</div>';
1000
1001                                $htmlCode .= '<div class="content ' . $active . '">';
1002                                $htmlCode .= __( 'Add external streams to room (re-stream), from RTSP IP cameras or similar sources.', 'ppv-live-webcams' );
1003
1004                                //retrieve
1005                                $reStreams = get_post_meta( $postID, 'reStreams', true );
1006                                $currentStream = get_post_meta( $postID, 'stream-name', true );
1007
1008                                if ( ! $reStreams || ! is_array( $reStreams ) ) {
1009                                        $reStreams = array();
1010                                }
1011
1012                                //add
1013                                if ( isset( $_GET['streams'] ) && isset( $_POST['reStream'] ) ) {
1014                                        $reStream = sanitize_text_field( $_POST['reStream'] );
1015                                $reStreamName = sanitize_file_name( sanitize_text_field( $_POST['reStreamName'] ) );
1016
1017
1018                                if ( !strstr( $reStreamName, '.stream' ) ) $reStreamName .= '.stream';
1019
1020                                        if (! in_array( $reStream, $reStreams ) )
1021                                        {
1022                                                $reStreams[ $reStreamName ] = $reStream;
1023                                                update_post_meta( $postID, 'reStreams', $reStreams );
1024                                                $reStreams = get_post_meta( $postID, 'reStreams', true );
1025
1026                                        }
1027                                        else $htmlCode .= '<br>Source already added: ' . esc_html( $reStream ) ;
1028
1029                                        $htmlCode .= self::restreamUpdate($postID, $options, $post);
1030                                }
1031
1032                                //performer
1033                                if ( isset( $_GET['set_performer'] ) ) {
1034                                        $set_performer = sanitize_text_field( $_GET['set_performer'] );
```

```
1035
1036                                    $streamHLS =
($options['httprestreamer'] ? $options['httprestreamer']
: $options['httpstreamer']) . $set_performer . '/playlist.m3u8';
1037
1038                                    //playback as HLS with Wowza SE
1039                                    update_post_meta( $postID, 'stream-name', $set_performer );
1040                                    update_post_meta( $postID, 'stream-hls', $streamHLS );
1041                                    update_post_meta( $postID, 'stream-type', 'restream' );
1042                                    update_post_meta( $postID, 'stream-mode', 'stream' );
1043                                    update_post_meta( $postID, 'stream-updated', time() );
1044
1045                                    $htmlCode .= '<br>Set Performing Stream:
' . esc_html( $set_performer ) ;
1046                            }
1047
1048                            $roomStream   = get_post_meta( $postID, 'stream-name', true );
1049
1050                            //remove
1051                            if ( isset( $_GET['delete_stream'] ) ) {
1052
1053                                    $reStreamName = sanitize_text_field( $_GET['delete_stream'] );
1054
1055
if ( isset( $reStreams[ $reStreamName ] ) ) unset( $reStreams[ $reStreamName ] );
1056                                    update_post_meta( $postID, 'reStreams', $reStreams );
1057
1058                                    //delete file
1059                                    $streamPath = $options['streamsPath'] . '/' . $reStreamName;
1060                                    if ( file_exists( $streamPath ) ) {
1061                                            unlink( $streamPath );
1062                                    }
1063
1064                                    if ( $roomStream == $reStreamName )
1065                                    {
1066                                            //remove
1067                                            update_post_meta( $postID, 'stream-name', '' ); //remove
if set as performer
1068                                            update_post_meta( $postID, 'stream-address', '' );
1069                                            update_post_meta( $postID, 'stream-protocol', 'rtsp' );
1070                                            update_post_meta( $postID, 'stream-type', 'webrtc' );
1071                                            update_post_meta( $postID, 'stream-mode', 'direct' );
1072                                            update_post_meta( $postID, 'stream-updated', time() );
1073                                    }
1074                                    $htmlCode .= '<br>Removed stream: ' . esc_html( $reStreamName ) ;
1075
1076                                    $reStreams = get_post_meta( $postID, 'reStreams', true );
1077                                    $htmlCode .= self::restreamUpdate($postID, $options, $post);
1078                            }
1079
1080
1081                            //update
1082                            if ( isset( $_GET['update_streams'] ) )
1083                            {
1084                                    $htmlCode .= '<br>ReStreams update.';
1085                                    $htmlCode .= self::restreamUpdate($postID, $options, $post);
1086                            }
1087
1088                            if ( ! $reStreams ) {
1089                            $reStreams = array();
1090                            }
1091
1092                            if ( ! is_array( $reStreams ) ) {
```

```
1093                                    $reStreams = array( $reStreams );
1094                            }
1095
1096
1097                            foreach ( $reStreams as $streamName => $source ) {
1098
1099                            $streamPath = $options['streamsPath'] . '/' . $streamName;
1100                            $streamLabel = str_replace('.stream', '', $streamName);
1101
1102
1103                            $htmlCode .= '<br><div class="ui label
' . ( $streamName != $currentStream ? 'basic' : '' ) . '"> <i
class="' . ( file_exists($streamPath) ? 'play' : 'pause' ) . ' icon"></i> ' . $streamLabel . '</div> <div
class="ui label">' . esc_html( $source ) . '</div>';
1104
1105                            if ( $streamName != $performer ) $htmlCode .= '<a class="ui tiny icon
button basic" data-content="Set as Performing Stream" href="' . esc_url( add_query_arg(
1106                                    array(
1107                                            'streams'           => 1,
1108                                            'set_performer' => $streamName,
1109                                    ),
1110                                    $this_page
1111                            ) ) . '"><i class="play circle icon"></i></a>';
1112
1113                            $htmlCode .= '<a class="ui tiny icon button basic" data-content="Remove"
href="' . esc_url( add_query_arg(
1114                                    array(
1115                                            'streams'           => 1,
1116                                            'delete_stream' => $streamName,
1117                                    ),
1118                                    $this_page
1119                            ) ) . '"><i class="x icon"></i></a>';
1120
1121                    } //foreach
1122
1123                    $htmlCode .= '<br><a class="ui small icon button"
href="' . esc_url( add_query_arg(
1124                                    array(
1125                                            'streams'           => 1,
1126                                            'update_streams' => 1,
1127                                    ),
1128                                    $this_page
1129                            ) ) . '"><i class="save icon"></i> Force Update</a>';
1130
1131
1132                    $htmlCode .= '<form class="ui form ' . $status . '" method="post"
enctype="multipart/form-data" action="' . esc_url( add_query_arg( 'streams', 1, $this_page ) ) . '"
id="addForm" name="addForm">';
1133                    $htmlCode .= '<br/><h5 class="ui dividing header"> <i class="plus icon"></i>
' . __( 'Add Source', 'ppv-live-webcams' ) . '</h5>';
1134
1135                    $htmlCode .= '<div class="field">
1136    <label>' . __( 'Stream Name', 'ppv-live-webcams' ) . '</label>
1137    <input type="text" id="reStreamName" name="reStreamName" value="' .  $post-
>post_title . '_' . ( count($reStreams) + 1 )  . '"/>
1138  </div>';
1139
1140                    $htmlCode .= '<div class="field">
1141    <label>' . __( 'Source Address', 'ppv-live-webcams' ) . '</label>
1142    <input type="text" id="reStream" name="reStream" value="" placeholder="' . __( 'rtsp://{server}
[:port]/{path}', 'ppv-live-webcams' ) . '"/>
1143  </div>
1144 ';
1145
```

```
Line
1146                        $htmlCode .= '
1147  <div class="field">
1148
1149  <button type="submit" id="submitButton" name="submitButton" class="ui submit button">
1150  <i class="plus icon"></i>
1151  ' . __( 'Add', 'ppv-live-webcams' ) . '
1152  </button>
1153
1154   </div>
1155  ';
1156                        $htmlCode .= '</form></div>';
1157
1158                        }
1159
1160                if ( $options['pushStreams'] ) {
1161
1162                        $active = '';
1163                        if ( isset($_POST['address']) ||
 isset($_GET['delete_destination']) ) $active = 'active';
1164
1165
1166                        $htmlCode .= '<div class="title ' . $active . '"> <i class="rss icon"></i>
' . __( 'Push RTMP Stream', 'ppv-live-webcams' ) . ': ' . $post->post_title . '</div>';
1167
1168
1169                        $htmlCode .= '<div class="content ' . $active . '">';
1170                        $htmlCode .= __( 'Push performer stream to external RTMP server or
platform.', 'ppv-live-webcams' );
1171
1172                        $pushDestinations = get_post_meta( $postID, 'pushStreams', true );
1173                        if ( ! $pushDestinations ) {
1174                                $pushDestinations = array();
1175                        }
1176                        if ( ! is_array( $pushDestinations ) ) {
1177                                $pushDestinations = array();
1178                        }
1179
1180                        if ( isset($_GET['streams']) && isset($_POST['address']) ) {
1181
$pushDestinations[] = array( 'address' => sanitize_text_field( $_POST['address'] ) );
1182                                update_post_meta( $postID, 'pushStreams', $pushDestinations );
1183                        }
1184
1185                        foreach ( $pushDestinations as $key => $destination ) {
1186                                $address    = $destination['address'];
1187                                $htmlCode .= '<br> #' . $key . ' <div class="ui
label">' . $address . '</div> <a class="ui tiny icon button" href="' . add_query_arg(
1188                                        array(
1189                                                'streams'            => 1,
1190                                                'delete_destination' => $key,
1191                                        ),
1192                                        $this_page
1193                                ) . '"><i class="x icon"></i></a>';
1194
1195                                if ( $_GET['delete_destination'] != '' ) {
1196                                        if ( intval( $_GET['delete_destination'] ) == $key ) {
1197                                                unset( $pushDestinations[ $key ] );
1198
update_post_meta( $postID, 'pushStreams', $pushDestinations );
1199                                                $htmlCode .= ' - removed';
1200                                        }
1201                                }
1202                        }
```

```
1203
1204                        $htmlCode .= '<form class="ui form ' . $status . '" method="post" enctype="multipart/form-data" action="' . add_query_arg( 'streams', 1, $this_page ) . '" id="addForm" name="addForm">';
1205                        $htmlCode .= '<br/><h5 class="ui dividing header"> <i class="plus icon"></i> ' . __( 'Add Destination', 'ppv-live-webcams' ) . '</h5>';
1206                        $htmlCode .= '<div class="field">
1207                          <label>' . __( 'Destination Address', 'ppv-live-webcams' ) . '</label>
1208                          <input type="text" id="address" name="address" value="" placeholder="' . __( 'rtmp[s]://{server}[:port]/{app}/{stream_key}', 'ppv-live-webcams' ) . '">
1209                        </div>
1210                       ';
1211
1212                        $htmlCode .= '
1213                        <div class="field">
1214
1215                        <button type="submit" id="submitButton" name="submitButton" class="ui submit button">
1216                        <i class="plus icon"></i>
1217                        ' . __( 'Add', 'ppv-live-webcams' ) . '
1218                        </button>
1219
1220                         </div>
1221                        ';
1222                        $htmlCode .= '</form></div>';
1223                }
1224
1225 //external encolder
1226
1227         $external_rtmp = self::is_true( get_post_meta( $post->ID, 'external_rtmp', true ) );
1228
1229                if ( $external_rtmp ) {
1230                        $htmlCode .= '<div class="title"> <i class="window maximize icon"></i> ' . __( 'External Broadcast', 'ppv-live-webcams' ) . ': ' . esc_html($post->post_title) . ' (OBS, Larix) </div>';
1231
1232                        $htmlCode .= '<div class="content"><form class="ui form">';
1233
1234                        $htmlCode .= '<div class="ui field">' . '<a href="https://obsproject.com/download" class="ui button"> <i class="ui cloud upload icon"> </I>' . __( 'Download OBS', 'ppv-live-webcams' ) . '</a>' . __( 'OBS Studio is free desktop application for live streaming from Linux, Mac and Windows. Includes advanced composition features, screen sharing, scenes, transitions, filters, media input options.', 'ppv-live-webcams' ) . '</div>';
1235
1236
1237                        if ($options['rtmpServer'] == 'videowhisper') {
1238                                $rtmpAddress =  trim( $options['videowhisperRTMP'] );
1239                                $stream = trim($options['vwsAccount']) . '/' . trim($username) . '?pin=' . self::getPin($current_user->ID, 'broadcast', $options);
1240                                $rtmpURL = $rtmpAddress . '//' . $stream;
1241                        }
1242                        else {
1243                                $rtmpAddress = self::rtmp_address( $current_user->ID, $post->ID, true, $username, $post->post_title, $options );
1244                        //wowza
1245
1246                        $application = substr( strrchr( $rtmpAddress, '/' ), 1 );
1247                        $stream      = $username;
1248
1249                        $adrp1 = explode( '://', $rtmpAddress );
1250                        $adrp2 = explode( '/', $adrp1[1] );
1251                        $adrp3 = explode( ':', $adrp2[0] );
1252
1253                        $application = $adrp2[1];
1254
```

```
1255                        $server = $adrp3[0];
1256                        $port   = $adrp3[1] ?? '';
1257                        if ( ! $port ) {
1258                                $port = 1935;
1259                        }
1260
1261                        $rtmpURL = $rtmpAddress . '/' . $stream;
1262                        }
1263
1264                        $htmlCode .= '<div class="ui field"><label>' . __( 'Server', 'ppv-live-webcams' ) . '</label> <INPUT class="ui fluid input" readonly value="' . esc_attr($rtmpAddress) . '" /><br>' . __( 'RTMP Address / OBS Stream URL: full streaming address. Contains: server, port if different than default 1935, application and control parameters, key. For OBS Settings: Stream > Server.', 'ppv-live-webcams' ) . '</div>';
1265
1266                        $htmlCode .= '<div class="ui field"><label>' . __( 'Stream Key', 'ppv-live-webcams' ) . '</label> <INPUT class="ui fluid input" readonly value="' . esc_attr($stream) . '" /><br>' . __( 'Stream Name / OBS Stream Key: name of stream. For OBS Settings: Stream > Stream Key. For GoCoder top right W Settings: Wowza Streaming Engine > Application > Stream Name', 'ppv-live-webcams' ) . '</div>';
1267
1268                        // $videoBitrate
1269                        $sessionsVars = self::varLoad( $options['uploadsPath'] . '/sessionsApp' );
1270                        if ( is_array( $sessionsVars ) ) {
1271                                if ( array_key_exists( 'limitClientRateIn', $sessionsVars ) ) {
1272
$limitClientRateIn = intval( $sessionsVars['limitClientRateIn'] ) * 8 / 1000;
1273
1274                                        if ( $limitClientRateIn ) {
1275                                                $videoBitrate = $limitClientRateIn - 100;
1276
1277                                                $htmlCode .= '<div class="ui field"><label>' . __( 'Maximum Video Bitrate', 'ppv-live-webcams' ) . '</label> <INPUT class="ui fluid input" readonly value="' . $videoBitrate . '" /><br>' . __( 'Use this value or lower for video bitrate, depending on resolution and audio bitrate. A static background and less motion requires less bitrate than movies, sports, games. For OBS Settings: Output > Streaming > Video Bitrate. For GoCoder bottom left bitrate. Warning: Trying to broadcast higher bitrate than allowed by streaming server will result in disconnects/failures.', 'ppv-live-webcams' ) . '</div>';
1278
1279                                                $htmlCode .= '<div class="ui field"><label>' . __( 'Maximum Audio Bitrate', 'ppv-live-webcams' ) . '</label> <INPUT class="ui fluid input" readonly value="96" /><br>' . __( 'Use this value or lower for audio bitrate. If you want to use higher Audio Bitrate, lower Video Bitrate to compensate for higher audio. For OBS Settings: Output > Streaming > Audio Bitrate. Warning: Trying to broadcast higher bitrate than allowed by streaming server will result in disconnects/failures.', 'ppv-live-webcams' ) . '</div>';
1280
1281                                        }
1282                                }
1283                        }
1284
1285                        $htmlCode .= '<div class="ui field">' . '<a href="https://apps.apple.com/app/larix-broadcaster/id1042474385" class="ui button"> <i class="ui cloud upload icon"></I>' . __( 'Larix Broadcaster for iOS', 'ppv-live-webcams' ) . '</a>' . __( 'Larix Broadcaster free iOS app uses full power of mobile devices cameras to stream live content.', 'ppv-live-webcams' ) . '</div>';
1286
1287                        $htmlCode .= '<div class="ui field">' . '<a href="https://play.google.com/store/apps/details?id=com.wmspanel.larix_broadcaster" class="ui button"> <i class="ui cloud upload icon"></I>' . __('Larix Broadcaster for Android', 'ppv-live-webcams' ) . '</a>' . __( 'Larix Broadcaster free Android app uses full power of mobile devices cameras to stream live content.', 'ppv-live-webcams' ) . '</div>';
1288
1289                        $htmlCode .= '<div class="ui field"><label>' . __( 'URL', 'ppv-live-webcams' ) . '</label> <INPUT class="ui fluid input" readonly value="' . $rtmpURL . '" /><br>' . __( 'RTMP URL / Larix URL: full streaming address with stream name. Contains: server, port if different than default 1935, application and control parameters, key, stream name. For Larix Connections > URL.', 'ppv-live-webcams' ) . '</div>';
1290
```

```
1291                        /*
1292                        $htmlCode .= '<div class="ui field">' . '<a
href="https://apps.apple.com/us/app/wowza-gocoder/id640338185" class="ui button"> <i class="ui cloud
upload icon"></I>' . __( 'Download GoCoder for iOS', 'ppv-live-webcams' ) . '</a>' . __( 'The Wowza
GoCoder™ app from Wowza Media Systems™ is a live audio and video capture and encoding application for iOS
8 and newer. Use the Wowza GoCoder app to broadcast HD-quality live events on the go from any location to
any screen using H.264 adaptive bitrate streaming.', 'ppv-live-webcams' ) . '</div>';
1293
1294                        $htmlCode .= '<div class="ui field">' . '<a
href="https://play.google.com/store/apps/details?id=com.wowza.gocoder&hl=en" class="ui button"> <i
class="ui cloud upload icon"></I>' . __( 'Download GoCoder for Android', 'ppv-live-webcams' ) . '</a>' .
__( 'The Wowza GoCoder™ app from Wowza Media Systems™ is a live audio and video capture and encoding
application for Android 4.4 and later. Use the Wowza GoCoder app to broadcast HD-quality live events on
the go from any location to any screen using H.264 adaptive bitrate streaming.', 'ppv-live-webcams' ) .
'</div>';
1295
1296
1297                        $htmlCode .= '<div class="ui field"><label>' . __( 'Server', 'ppv-live-webcams' )
. '</label> <INPUT class="ui fluid input" readonly value="' . $server . '" /><br>' . __( 'Server IP or
domain. For GoCoder top right W Settings: Wowza Streaming Engine > Host > Server.', 'ppv-live-webcams' )
. '</div>';
1298
1299                        $htmlCode .= '<div class="ui field"><label>' . __( 'Port', 'ppv-live-webcams' ) .
'</label> <INPUT class="ui fluid input" readonly value="' . $port . '" /><br>' . __( 'Server port. For
GoCoder top right W Settings: Wowza Streaming Engine > Host > Port.', 'ppv-live-webcams' ) . '</div>';
1300
1301                        $htmlCode .= '<div class="ui field"><label>' . __( 'Application', 'ppv-live-
webcams' ) . '</label> <INPUT class="ui fluid input" readonly value="' . $adrp2[1] . '" /><br>' . __(
'Application with parameters. For GoCoder top right W Settings: Wowza Streaming Engine > Host >
Application.', 'ppv-live-webcams' ) . '</div>';
1302                        */
1303
1304                        $htmlCode .= '</form> </div>';
1305                }
1306
1307        $htmlCode .= '</div>'; //accordion end
1308
1309        $htmlCode .= '<script>
1310jQuery(document).ready(function(){
1311
1312        jQuery(".ui.accordion.vwStreams").accordion({animateChildren: false, exclusive: false});
1313
1314});
1315</script>
1316';
1317
1318
1319                return $htmlCode;
1320        }
1321
1322
1323        static function redirectURL( $user ) {
1324
1325                $redirect_to = get_site_url();
1326
1327                if ( isset( $user->roles ) && is_array( $user->roles ) ) {
1328
1329                        // check for admins
1330                        if ( in_array( 'administrator', $user->roles ) ) {
1331                                return get_admin_url();
1332
1333                        } else {
1334
1335                                $options = get_option( 'VWliveWebcamsOptions' );
1336
```

```
Line
1337                                    // performer to dashboard
1338                                    if ( in_array( $options['rolePerformer'], $user->roles ) ) {
1339                                            $pid = $options['p_videowhisper_webcams_performer'];
1340                                            if ( $pid ) {
1341                                                    return get_permalink( $pid );
1342                                            } else {
1343                                                    return $redirect_to;
1344                                            }
1345                                    }
1346
1347                                    // studio to studio dashboard
1348                                    if ( in_array( $options['roleStudio'], $user->roles ) ) {
1349                                            $pid = $options['p_videowhisper_webcams_studio'];
1350                                            if ( $pid ) {
1351                                                    return get_permalink( $pid );
1352                                            } else {
1353                                                    return $redirect_to;
1354                                            }
1355                                    }
1356
1357                                    // client to client dashboard
1358                                    if ( in_array( $options['roleClient'], $user->roles ) ) {
1359                                            $pid = $options['p_videowhisper_webcams_client'];
1360                                            if ( $pid ) {
1361                                                    return get_permalink( $pid );
1362                                            } else {
1363                                                    return $redirect_to;
1364                                            }
1365                                    }
1366                            }
1367                    }
1368                    return $redirect_to;
1369
1370            }
1371
1372
1373            // login/logout before output
1374            static function after_setup_theme() {
1375
1376                    if ( isset($_SERVER['REQUEST_METHOD']) && $_SERVER['REQUEST_METHOD'] == 'GET' )
1377                    if ( isset($_GET['logout']) && isset($_GET['verify']) && !isset($_GET['error']) )
1378                    {
1379                                    $error  = '';
1380                                    $status = '';
1381                                    $url = '';
1382
1383                            $options = self::getOptions();
1384
if ($options['p_videowhisper_login']) $url = get_permalink( $options['p_videowhisper_login'] );
1385
1386                            if ( wp_verify_nonce( $_GET['verify'], 'logout' . $_GET['logout'] ) )
1387                            {
1388                            wp_logout();
1389                            }
1390                            else
1391                            {
1392                                    $error .=  '<li>' . __( 'Nonce incorrect for logout. Could be cache
related: Try again!', 'ppv-live-webcams' ) . '</li>';
1393
1394                            }
1395
1396                                    if ( $error ) {
```

```
Line
1397                                    $status = 'warning';
1398
1399                                    if (!$url) $url = wp_login_url();
1400                                    wp_redirect( add_query_arg( 'login_error', urlencode( $error ), $url ) );
1401                            }
1402                            else
1403                            {
1404                                    if (!$url) $url = get_site_url();
1405                                    wp_redirect( $url );
1406                            }
1407
1408                    exit();
1409            }
1410
1411            if ( isset($_SERVER['REQUEST_METHOD']) && $_SERVER['REQUEST_METHOD'] == 'POST' ) {
1412                    if ( $_POST['videowhisper_login'] ?? false) {
1413                            $error  = '';
1414                            $status = '';
1415
1416                            $options = self::getOptions();
1417
1418                            // Check for login rate limiting
1419                            $ip_address = self::get_ip_address();
1420                            $ip_hash = md5($ip_address); // Hash the IP for security
1421                            $transient_name = 'vw_failed_login_' . $ip_hash;
1422                            $failed_attempts = get_transient($transient_name);
1423
1424                            // Set max attempts - default to 5 if not specified in options
1425                            $max_attempts = isset($options['maxLoginAttempts']) ? intval($options['maxLoginAttempts']) : 5;
1426                            $lockout_time = isset($options['loginLockoutTime']) ? intval($options['loginLockoutTime']) : 3600; // Default 1 hour
1427
1428                            // Check if user has too many failed attempts
1429                            if ($failed_attempts && $failed_attempts >= $max_attempts) {
1430                                    $error .= '<li>' . sprintf(__('Too many failed login attempts. Please try again after %d minute(s).', 'ppv-live-webcams'), ceil($lockout_time/60)) . '</li>';
1431                                    wp_redirect(add_query_arg('login_error', urlencode($error), get_permalink($options['p_videowhisper_login'])));
1432                                    exit();
1433                            }
1434
1435                            $login    = sanitize_text_field( $_POST['login'] );
1436                            $password = sanitize_text_field( $_POST['password'] );
1437
1438                            if ( ! $password ) {
1439                                    $error .= '<li>' . __( 'A password is required.', 'ppv-live-webcams' ) . '</li>';
1440                            }
1441                            if ( ! $login ) {
1442                                    $error .= '<li>' . __( 'A login (username) is required.', 'ppv-live-webcams' ) . '</li>';
1443                            }
1444
1445                            if ( ! wp_verify_nonce( $_GET['videowhisper'], 'vw_login' ) ) {
1446                                    $error .=  '<li>' . __( 'Nonce incorrect for login. Could be cache related: Try again!' , 'ppv-live-webcams' ) . '</li>';
1447                            }
1448
1449                            if ( $options['recaptchaSite'] ) {
1450
```

```
1451                                    if ( isset( $_POST['recaptcha_response'] ) ) {
1452                                            if ( $_POST['recaptcha_response'] ) {
1453                                                    // Build POST request:
1454                                                    $recaptcha_response = sanitize_text_field( $_POST['recaptcha_response'] );
1455
1456                                                    // Make and decode POST request:
1457                                                    // $recaptcha = file_get_contents('https://www.google.com/recaptcha/api/siteverify' . '?secret=' . $options['recaptchaSecret'] . '&response=' . $recaptcha_response);
1458                                                    // $recaptchaD = json_decode($recaptcha);
1459                                                    $response = wp_remote_get( 'https://www.google.com/recaptcha/api/siteverify' . '?secret=' . $options['recaptchaSecret'] . '&response=' . $recaptcha_response );
1460                                                    $body = wp_remote_retrieve_body( $response );
1461                                                    $recaptchaD = json_decode( $body );
1462
1463                                                    // Take action based on the score returned:
1464                                                    if ( $recaptchaD->score >= 0.3 ) {
1465                                                            // Verified
1466                                                                    //      $htmlCode = '<!-- VideoWhisper reCAPTCHA v3 score: ' . $recaptchaD->score . '-->';
1467
1468                                                    } else {
1469                                                            // Not verified - show form error
1470                                                            $error .= '<li>' . 'Google reCAPTCHA v3 Failed. Score: ' . $recaptchaD->score . ' . Privacy features may prevent Google from detecting user activity. For Brave browser turn off Brave Shields. Try again with privacy options disabled for this site or use a different browser!' . '</li>';
1471                                                    }
1472                                            } else {
1473                                                    $error .= '<li>' . 'Google reCAPTCHA v3 empty. Make sure you have JavaScript enabled or try a different browser!' . '</li>';
1474                                            }
1475                                    } else {
1476                                            $error .= '<li>' . 'Google reCAPTCHA v3 missing. Make sure you have JavaScript enabled or try a different browser!' . '</li>';
1477                                    }
1478                            }
1479
1480                            if ( $error ) {
1481                                    $status = 'warning';
1482                                    wp_redirect( add_query_arg( 'login_error', urlencode( $error ), get_permalink( $options['p_videowhisper_login'] ) ) );
1483                                    exit();
1484                            } else {
1485
1486                                    $login_user = wp_signon(
1487                                            array(
1488                                                    'user_login'    => $login,
1489                                                    'user_password' => $password,
1490                                                    'remember'      => true,
1491                                            ),
1492                                            is_ssl()
1493                                    );
1494
1495                                    if ( $login_user && ! is_wp_error( $login_user ) ) {
1496
1497                                            update_user_meta( $login_user->ID, 'videowhisper_ip_login', self::get_ip_address() ); // save ip
1498
1499                                            // redirect
1500                                            wp_redirect( self::redirectURL( $login_user ) );
```

```
1501                                            exit(); // you need to exit to get wp_redirect to work
1502                                    }
1503
1504                                    if ( is_wp_error( $login_user ) ) {
1505                                            $status = 'warning';
1506                                            $error .= $login_user->get_error_message() . "\n";
1507
1508                                            // Track failed login attempt
1509                                            $ip_address = self::get_ip_address();
1510                                            $ip_hash = md5($ip_address);
1511                                            $transient_name = 'vw_failed_login_' . $ip_hash;
1512                                            $failed_attempts = get_transient($transient_name);
1513
1514                                            // Increment failed attempts counter
1515                                            if ($failed_attempts) {
1516                                                    $failed_attempts++;
1517                                            } else {
1518                                                    $failed_attempts = 1;
1519                                            }
1520
1521                                            // Store in transient with expiration time (default 1 hour)
1522                                            $lockout_time = isset($options['loginLockoutTime']) ? intval($options['loginLockoutTime']) : 3600;
1523                                            set_transient($transient_name, $failed_attempts, $lockout_time);
1524
1525                                            // Add information about remaining attempts
1526                                            $max_attempts = isset($options['maxLoginAttempts']) ? intval($options['maxLoginAttempts']) : 5;
1527                                            $remaining = $max_attempts - $failed_attempts;
1528                                            if ($remaining > 0) {
1529                                                    $error .= '<li>' . sprintf(__('You have %d login attempts remaining before your IP is temporarily blocked.', 'ppv-live-webcams'), $remaining) . '</li>';
1530                                            }
1531
1532                                            wp_redirect( add_query_arg( 'login_error', urlencode( $error ), get_permalink( $options['p_videowhisper_login'] ) ) );
1533                                            exit();
1534                                    }
1535                            }
1536
1537                            // end process login
1538                    }
1539            }
1540    }
1541
1542
1543    static function sendNotification($userID, $messageSubject, $messageText, $url, $options = null)
1544    {
1545            //notify user $userID by email & SMS
1546            if (!$options) $options = self::getOptions();
1547
1548            $user = get_userdata($userID);
1549            $email = $user->user_email;
1550
1551            //email notification
1552            $email_last =  get_user_meta( $userID, 'email_last', true );
1553            if (!$email_last || $email_last < ( time() - $options['email_cooldown']) )
1554            {
1555            wp_mail( $email, $messageSubject, $messageText . "\r\n" . $url );
1556            update_user_meta( $userID, 'email_last', time() );
```

```
Line
1557                }
1558
1559                //sms notification
1560                if (!$options['sms_number']) return;
1561
1562                $sms_message = $messageSubject . ' ' . $url;
1563
1564                        $sms_number = get_user_meta( $userID, 'sms_number', true );
1565                        if ($sms_number)
1566                        {
1567                                //cooldown
1568                                $sms_last =  get_user_meta( $userID, 'sms_last', true );
1569                                if (!$sms_last || $sms_last < ( time() - $options['sms_cooldown']) )
1570                                {
1571
1572 if ($options['wp_sms']) if ( function_exists('wp_sms_send') ) wp_sms_send($sms_number,  $sms_message);
1573
1574                                if ($options['wp_sms_twilio']) if ( function_exists('twl_send_sms') )
1575                                {
1576 $response = twl_send_sms( array( 'number_to' => $sms_number, 'message' => $sms_message ) );
1577                                        if( is_wp_error( $response ) ) return '<div class="ui
 message">SMS Error: ' . esc_html( $response->get_error_message() ) . '</div>';
1578                                }
1579
1580                                update_user_meta( $userID, 'sms_last', time() );
1581                                }
1582                        }
1583
1584                return;
1585        }
1586
1587        static function videowhisper_password_form( $atts ) {
1588
1589                $options = get_option( 'VWliveWebcamsOptions' );
1590
1591                $atts = shortcode_atts(
1592                        array(
1593                                'role'     => '',
1594                                'register' => 1,
1595                                'terms'    => get_permalink( $options['p_videowhisper_terms'] ),
1596                                'privacy'  => get_permalink( $options['p_videowhisper_privacy'] ),
1597                        ),
1598                        $atts,
1599                        'videowhisper_password_form'
1600                );
1601                $atts = array_map('esc_attr', $atts);
1602                $atts['terms'] = esc_url($atts['terms']);
1603                $atts['privacy'] = esc_url($atts['privacy']);
1604
1605                self::enqueueUI();
1606                $this_page = self::getCurrentURL();
1607                $htmlCode  = '';
1608                $error     = '';
1609
1610                if ( ! $options['loginFrontend'] ) {
1611                        $htmlCode .= '<div class="ui ' . $options['interfaceClass'] . ' segment">';
1612                        $htmlCode .= __( 'Frontend login is disabled from settings.', 'ppv-live-
webcams' );
1613                        $htmlCode .= '<br /> <a class="ui button qbutton"
 href="' . wp_login_url() . '">' . __( 'Login', 'ppv-live-webcams' ) . '</a>';
1614                        $htmlCode .= '</div>';
```

```
1615                        return $htmlCode;
1616                }
1617
1618                // email link
1619                if ( isset($_GET['key']) && isset($_GET['user']) ) {
1620
1621                        $key = filter_input( INPUT_GET, 'key' );
1622                        if ( ! $key ) {
1623                                return '<div class="ui ' . $options['interfaceClass'] . ' segment">The key parameter is invalid.</div>';
1624                        }
1625                        $user_id = filter_input( INPUT_GET, 'user', FILTER_VALIDATE_INT, array( 'options' => array( 'min_range' => 1 ) ) );
1626                        if ( ! $user_id ) {
1627                                return '<div class="ui ' . $options['interfaceClass'] . ' segment">The user_id parameter is invalid.</div>';
1628                        }
1629
1630                        // password form
1631                        $user = get_userdata( $user_id );
1632                        if ( ! $user ) {
1633                                return '<div class="ui ' . $options['interfaceClass'] . ' segment">User does not exist.</div>';
1634                        }
1635
1636                        $code = get_user_meta( $user_id, 'videowhisper_newpassword', true );
1637                        if ( ! $code ) {
1638                                $error .= '<li>' . __( 'Password change request code missing. Already changed?', 'ppv-live-webcams' ) . '</li>';
1639                        }
1640
1641                        // Check token expiration
1642                        $expiry = get_user_meta( $user_id, 'videowhisper_newpassword_expiry', true );
1643                        if ($expiry && time() > intval($expiry)) {
1644                                $error .= '<li>' . __( 'Password reset link has expired. Please request a new one.', 'ppv-live-webcams' ) . '</li>';
1645                                delete_user_meta( $user_id, 'videowhisper_newpassword' );
1646                                delete_user_meta( $user_id, 'videowhisper_newpassword_expiry' );
1647                        }
1648
1649                        // Verify CSRF nonce
1650                        if (!isset($_GET['nonce']) || !wp_verify_nonce($_GET['nonce'], 'reset-password-' . $user_id)) {
1651                                $error .= '<li>' . __( 'Invalid security token. Please request a new password reset link.', 'ppv-live-webcams' ) . '</li>';
1652                        }
1653
1654                        if ( ! $error ) {
1655                                if ( $code != $key ) {
1656                                        $error .= '<li>' . __( 'Code does not match. Make sure you use link from last email if you tried multiple times.', 'ppv-live-webcams' ) . '</li>';
1657
1658                                        // Track failed verification attempts
1659                                        $verify_attempts = get_user_meta( $user_id, 'password_reset_verify_attempts', true );
1660                                        $verify_attempts = $verify_attempts ? $verify_attempts + 1 : 1;
1661                                        update_user_meta( $user_id, 'password_reset_verify_attempts', $verify_attempts );
1662
1663                                        // If too many failed attempts, invalidate the token
1664                                        if ($verify_attempts > 5) {
1665                                                delete_user_meta( $user_id, 'videowhisper_newpassword' );
1666                                                delete_user_meta( $user_id, 'videowhisper_newpassword_expiry' );
```

```
1667                                            $error .= '<li>' . __( 'Too many failed attempts. Please request a new password reset.', 'ppv-live-webcams' ) . '</li>';
1668                                    }
1669                            }
1670                    }
1671
1672                    if ( $error ) {
1673                            $htmlCode .= '<div class="ui warning message">
1674    <div class="header">' . __( 'Error', 'ppv-live-webcams' ) . ':</div>
1675    <ul class="list">
1676      ' . $error . '
1677    </ul>
1678  </div>';
1679                    }
1680
1681                            // show form to provide new password
1682                    if ( ! $error ) {
1683                            $status = '';
1684                            $htmlCode .= '<form class="ui ' . $options['interfaceClass'] . ' form ' . $status . '" method="post" enctype="multipart/form-data" action="' . wp_nonce_url( $this_page, 'vw_password', 'videowhisper' ) . '" id="passwordForm" name="passwordForm">';
1685                            $htmlCode .= '<h4 class="ui dividing header"> <i class="lock icon"></i> ' . __( 'Change Password', 'ppv-live-webcams' ) . ': ' . $user->user_login . '</h4>';
1686
1687                            if ( $options['recaptchaSite'] ) {
1688                                    wp_enqueue_script( 'google-recaptcha-v3', 'https://www.google.com/recaptcha/api.js?render=' . $options['recaptchaSite'], array() );
1689
1690                                    $htmlCode .= '<input type="hidden" name="recaptcha_response" id="recaptchaResponseLogin">';
1691
1692                                    $htmlCode .= '<script>
1693 function onFormSubmitClick() {
1694         grecaptcha.ready(function() {
1695           grecaptcha.execute("' . $options['recaptchaSite'] . '", {action: "password"}).then(function(token) {
1696                   var recaptchaResponse = document.getElementById("recaptchaResponseLogin");
1697                   recaptchaResponse.value = token;
1698                   console.log("VideoWhisper Login: Google reCaptcha v3 updated", token, recaptchaResponse);
1699                   var ticketForm = document.getElementById("passwordForm");
1700                   ticketForm.submit();
1701           });
1702         });
1703       }
1704 </script>
1705 <noscript>JavaScript is required to use <a href="https://paidvideochat.com/">PaidVideochat turnkey live cam site solution</a>. Contact <a href="https://videowhisper.com/">VideoWhisper Online Platforms</a> for clarifications.</noscript>';
1706                            } else {
1707                                    // recaptcha disabled
1708                                    $htmlCode .= '<script>
1709 function onFormSubmitClick(e) {
1710           console.log("VideoWhisper Login: Google reCaptcha v3 disabled");
1711                   var ticketForm = document.getElementById("passwordForm");
1712                   ticketForm.submit();
1713       }
1714 </script>
1715 <noscript>JavaScript is required to use <a href="https://paidvideochat.com/">PaidVideochat turnkey live cam site solution</a>. Contact <a href="https://videowhisper.com/">VideoWhisper Online Platforms</a> for clarifications.</noscript>
1716 ';
1717                            }
1718
```

```
1719                                $htmlCode .= '
1720                        <div class="field">
1721                    <label>' . __( 'Password', 'ppv-live-webcams' ) . '</label>
1722                    <input type="password" name="password" value="' . '" placeholder="' . __( 'Password', 'ppv-live-webcams' ) . '">
1723                  </div>
1724
1725                  <div class="field">
1726                    <label>' . __( 'Password Verification', 'ppv-live-webcams' ) . '</label>
1727                    <input type="password" name="password2" value="' . '" placeholder="' . __( 'Password Verification', 'ppv-live-webcams' ) . '">
1728                  </div>
1729                  ';
1730
1731                                $htmlCode .= '
1732                  <div class="field">
1733                  <input type="hidden" name="videowhisper_password" value="password">
1734                  <input type="hidden" name="user_id" value="' . $user_id . '">
1735                  <input type="hidden" name="key" value="' . $key . '">
1736
1737                  <button type="button" id="submitButton" name="submitButton" class="ui submit button" onclick="onFormSubmitClick()">
1738                  <i class="lock icon"></i>
1739                  ' . __( 'Update', 'ppv-live-webcams' ) . '
1740                </button>
1741                   </div>
1742                  ';
1743
1744                                $htmlCode .= '
1745                                <div class="field">';
1746                                $htmlCode .= ' <a class="ui tiny button basic" href="' . $atts['terms'] . '"> <i class="clipboard list icon"></i> ' . __( 'Terms of Use', 'ppv-live-webcams' ) . '</a> ';
1747                                $htmlCode .= ' <a class="ui tiny button basic" href="' . $atts['privacy'] . '"> <i class="clipboard list icon"></i> ' . __( 'Privacy Policy', 'ppv-live-webcams' ) . '</a> ';
1748                                $htmlCode .= '<br/></div>';
1749
1750                                $htmlCode .= '<br/></form>';
1751
1752                        }
1753                }
1754
1755 if ( isset($_POST['videowhisper_password']) && $_POST['videowhisper_password'] == 'email' && isset($_POST['email']) && $_POST['email'] ) {
1756
1757                        // bot checks
1758                        if ( ! wp_verify_nonce( $_GET['videowhisper'], 'vw_password' ) ) {
1759                                $error .= '<li>' . __( 'Nonce incorrect for password reset.', 'ppv-live-webcams' ) . '</li>';
1760                        }
1761
1762                        if ( $options['recaptchaSite'] ) {
1763
1764                                if ( isset( $_POST['recaptcha_response'] ) ) {
1765                                        if ( $_POST['recaptcha_response'] ) {
1766                                                // Build POST request:
1767 $recaptcha_response = sanitize_text_field( $_POST['recaptcha_response'] );
1768
1769                                                // Make and decode POST request:
1770                                                // $recaptcha = file_get_contents('https://www.google.com/recaptcha/api/siteverify' . '?secret=' . $options['recaptchaSecret'] . '&response=' . $recaptcha_response);
```

```
1771                                            // $recaptchaD = json_decode($recaptcha);
1772                                            $response
 = wp_remote_get( 'https://www.google.com/recaptcha/api/siteverify' . '?
secret=' . $options['recaptchaSecret'] . '&response=' . $recaptcha_response );
1773                                            $body       = wp_remote_retrieve_body( $response );
1774                                            $recaptchaD = json_decode( $body );
1775
1776                                            // Take action based on the score returned:
1777                                            if ( !$recaptchaD->score )
1778                                            {
1779                                                    $error .= '<li>Google reCAPTCHA v3 Failed. No
score received. Try again, disable ad blockers or use a different browser!</li>';
1780
1781                                            }
1782                                            elseif ( $recaptchaD->score >= 0.3 ) {
1783                                                    // Verified
1784                                                    $htmlCode .= '<!-- VideoWhisper reCAPTCHA v3
score: ' . $recaptchaD->score . '-->';
1785
1786                                            } else {
1787                                                    // Not verified - show form error
1788                                                    $error .= '<li>Google reCAPTCHA v3 Failed. Score:
' . $recaptchaD->score . ' . Try again, disable ad blockers or use a different browser!</li>';
1789                                            }
1790                                    } else {
1791                                            $error .= '<li>Google reCAPTCHA v3 empty. Make sure you
have JavaScript enabled or try a different browser!</li>';
1792                                    }
1793                            } else {
1794                                    $error .= '<li>Google reCAPTCHA v3 missing. Make sure you have
JavaScript enabled or try a different browser!</li>';
1795                            }
1796                    }
1797
1798                    $email = sanitize_email( $_POST['email'] );
1799                    if ( ! $email ) {
1800                            $error .= '<li>' . __( 'A valid email address is required.', 'ppv-live-
webcams' ) . '</li>';
1801                    }
1802
1803                    if ( ! $error ) {
1804                            $user = get_user_by( 'email', $email );
1805                            if ( ! $user ) {
1806                                    $error .= '<li>' . __( 'No user found for that email. Try a
different email or register a new account!', 'ppv-live-webcams' ) . '</li>';
1807                            } else {
1808                                    $user_id         = $user->ID;
1809                                    // Generate a cryptographically secure random token instead of
predictable token
1810                                    $code            = bin2hex(random_bytes(32)); // Generate a 64-
character random hex string
1811                                    $expiry          = time() + 3600; // Token expires in 1 hour
1812                                    $activation_link = add_query_arg(
1813                                            array(
1814                                                    'key'   => $code,
1815                                                    'user' => $user_id,
1816                                                    'nonce' => wp_create_nonce('reset-password-
' . $user_id)
1817                                            ),
1818                                            get_permalink( $options['p_videowhisper_password_form'] )
1819                                    );
1820
update_user_meta( $user_id, 'videowhisper_newpassword', $code ); // password reset code
1821
update_user_meta( $user_id, 'videowhisper_newpassword_expiry', $expiry ); // set expiration
```

```
1822
1823                                    // Rate limiting for reset requests
1824 $reset_attempts = get_user_meta( $user_id, 'password_reset_attempts', true );
1825 $reset_attempts_time = get_user_meta( $user_id, 'password_reset_attempts_time', true );
1826
1827                                    // Reset counter if it's been more than an hour since last attempt
1828                                    if ($reset_attempts_time && time() - $reset_attempts_time > 3600) {
1829                                            $reset_attempts = 0;
1830                                    }
1831
1832                                    // Increment attempt counter
1833                                    $reset_attempts = $reset_attempts ? $reset_attempts + 1 : 1;
1834 update_user_meta( $user_id, 'password_reset_attempts', $reset_attempts );
1835 update_user_meta( $user_id, 'password_reset_attempts_time', time() );
1836
1837 wp_mail( $email, $options['passwordSubject'], $options['passwordText'] . "\r\n" . $activation_link );
1838
1839                                    //notify by SMS
1840                                    if ($options['sms_number'])
1841                                    {
1842                                            $sms_number = get_user_meta( $user->ID, 'sms_number', true );
1843                                            // Verify that the SMS number is properly formatted before sending
1844                                            $sms_number = preg_replace('/[^0-9+]/', '', $sms_number);
1845
1846                                            if ($sms_number)
1847                                            {
1848                                                    $sms_sent = false;
1849
1850                                                    //cooldown
1851                                                    $sms_last =  get_user_meta( $user->ID, 'sms_last', true );
1852
1853                                                    if (!$sms_last || $sms_last < ( time() - $options['sms_cooldown']) )
1854                                                    {
1855
1856                                                    // Sanitize the SMS message content
1857 $sms_message = sanitize_text_field($options['passwordSubject']) . ' ' . esc_url($activation_link);
1858
1859 if ($sms_number && $options['wp_sms'] && function_exists('wp_sms_send')) {
1860                                                            wp_sms_send($sms_number, $sms_message);
1861                                                            $sms_sent = true;
1862                                                    }
1863
1864
1865 if ($options['wp_sms_twilio']) if ( function_exists('twl_send_sms') )
1866                                                    {
1867 $response = twl_send_sms( array( 'number_to' => $sms_number, 'message' => $sms_message ) );
1868 if( is_wp_error( $response ) ) $htmlCode .= '<div class="ui message">SMS Error: ' . esc_html( $response->get_error_message() ) . '</div>';
1869                                                            else
1870                                                            { //sms sent
```

```
Line
1871                                                                    $sms_sent = true;
1872                                                            }
1873                                                    }
1874
1875                                                    if ($sms_sent)
1876                                                    {
1877                                                            // update sms
1878                                                            update_user_meta( $user->ID, 'sms_last', time() );
1879                                                    }
1880
1881
1882                                                    }
1883                                            }
1884                                    }
1885
1886                                    $htmlCode .= '<h4 class="ui dividing header"> <i class="lock icon"></i> ' . __( 'Change Password', 'ppv-live-webcams' ) . '</h4>';
1887                                    $htmlCode .= '<div class="ui message">' . __( 'Email was sent. Use link from email to change password.', 'ppv-live-webcams' ) . '</div>';
1888
1889                                }
1890                        }
1891
1892                        if ( $error ) {
1893                                $htmlCode .= '<div class="ui warning message">
1894    <div class="header">' . __( 'Error', 'ppv-live-webcams' ) . ':</div>
1895    <ul class="list">
1896      ' . $error . '
1897    </ul>
1898  </div>';
1899                        }
1900                }
1901
1902    if ( isset($_POST['videowhisper_password']) && $_POST['videowhisper_password'] == 'password' && isset($_POST['password']) ) {
1903                        // change password
1904
1905                        $password  = sanitize_text_field( $_POST['password'] );
1906                        $password2 = sanitize_text_field( $_POST['password2'] ?? '' );
1907                        $user_id   = intval( $_POST['user_id'] ?? 0 );
1908                        $key       = sanitize_text_field( $_POST['key'] ?? '' );
1909
1910                        // security errors: exit
1911                        if ( ! $user_id || ! $key ) {
1912                                return '<div class="ui ' . esc_attr( $options['interfaceClass'] ) . ' segment">Key parameters missing.</div>';
1913                        }
1914                        $user = get_userdata( $user_id );
1915                        if ( ! $user ) {
1916                                return '<div class="ui ' . esc_attr( $options['interfaceClass'] ) . ' segment">User does not exist.</div>';
1917                        }
1918                        $code = get_user_meta( $user_id, 'videowhisper_newpassword', true );
1919                        if ( $code != $key ) {
1920                                return __( 'Code does not match. Make sure you use link from last email if you tried multiple times.', 'ppv-live-webcams' );
1921                        }
1922
1923                        // user errors
1924                        if ( ! $password ) {
1925                                $error .= '<li>' . __( 'A password is required.', 'ppv-live-webcams' ) . '</li>';
```

```
1926                        }
1927                        if ( $password != $password2 ) {
1928                                $error .= '<li>' . __( 'Password verification does not match.', 'ppv-live-webcams' ) . '</li>';
1929                        }
1930
1931                        // bot checks
1932                        if ( ! wp_verify_nonce( $_GET['videowhisper'], 'vw_password' ) ) {
1933                                $error .= '<li>' . __( 'Nonce incorrect for password reset.', 'ppv-live-webcams' ) . '</li>';
1934                        }
1935                        if ( $options['recaptchaSite'] ) {
1936
1937                                if ( isset( $_POST['recaptcha_response'] ) ) {
1938                                        if ( $_POST['recaptcha_response'] ) {
1939                                                // Build POST request:
1940                                                $recaptcha_response = sanitize_text_field( $_POST['recaptcha_response'] );
1941
1942                                                // Make and decode POST request:
1943                                                // $recaptcha = file_get_contents('https://www.google.com/recaptcha/api/siteverify' . '?secret=' . $options['recaptchaSecret'] . '&response=' . $recaptcha_response);
1944                                                // $recaptchaD = json_decode($recaptcha);
1945                                                $response = wp_remote_get( 'https://www.google.com/recaptcha/api/siteverify' . '?secret=' . esc_attr( $options['recaptchaSecret'] )  . '&response=' . esc_attr( $recaptcha_response)  );
1946                                                $body       = wp_remote_retrieve_body( $response );
1947                                                $recaptchaD = json_decode( $body );
1948
1949                                                // Take action based on the score returned:
1950                                                if ( $recaptchaD->score >= 0.3 ) {
1951                                                        // Verified
1952                                                        $htmlCode .= '<!-- VideoWhisper reCAPTCHA v3 score: ' . esc_html( $recaptchaD->score ) . '-->';
1953
1954                                                } else {
1955                                                        // Not verified - show form error
1956                                                        $error .= '<li>Google reCAPTCHA v3 Failed. Score: ' . esc_html( $recaptchaD->score ) . ' . Try again or using a different browser!</li>';
1957                                                }
1958                                        } else {
1959                                                $error .= '<li>Google reCAPTCHA v3 empty. Make sure you have JavaScript enabled or try a different browser!</li>';
1960                                        }
1961                                } else {
1962                                        $error .= '<li>Google reCAPTCHA v3 missing. Make sure you have JavaScript enabled or try a different browser!</li>';
1963                                }
1964                        }
1965
1966                        if ( $error ) {
1967                                $htmlCode .= '<div class="ui warning message">
1968    <div class="header">' . __( 'Error', 'ppv-live-webcams' ) . ':</div>
1969    <ul class="list">
1970      ' . $error . '
1971    </ul>
1972  </div>';
1973                        } else {
1974                                wp_set_password( $password, $user_id );
1975                                delete_user_meta( $user_id, 'videowhisper_newpassword' ); // password changed
1976                                update_user_meta( $user_id, 'videowhisper_ip_newpassword', self::get_ip_address() ); // save ip used to change password
```

```
1977
1978                                $htmlCode .= '<h4 class="ui dividing header"> <i class="lock icon"></i>
' . __( 'Change Password', 'ppv-live-webcams' ) . ': ' . esc_html( $user->user_login ) . '</h4>';
1979                                $htmlCode .= '<div class="ui message">' . __( 'Password was
changed.', 'ppv-live-webcams' ) . '</div>';
1980                                if ( ! is_user_logged_in() ) {
1981                                        $htmlCode .= '<div class="ui
' . esc_attr( $options['interfaceClass'] ) . ' header"> <i class="sign in icon"></i>
' . __( 'Login', 'ppv-live-webcams' ) . '</div>' . do_shortcode( '[videowhisper_login_form]' );
1982                                }
1983                        }
1984                }
1985
1986                // email form
1987                if ( !isset($_POST['videowhisper_password']) && !isset($_GET['key']) ) {
1988                        // provide email form
1989                        $status = '';
1990                        $email = '';
1991
1992                        $htmlCode .= '<form class="ui ' . esc_attr( $options['interfaceClass'] ) . ' form
' . $status . '" method="post" enctype="multipart/form-data"
action="' . wp_nonce_url( $this_page, 'vw_password', 'videowhisper' ) . '" id="passwordForm"
name="passwordForm">';
1993
1994                        $htmlCode .= '<h4 class="ui dividing header"> <i class="lock icon"></i>
' . __( 'Change Password', 'ppv-live-webcams' ) . '</h4>';
1995
1996                        if ( $options['recaptchaSite'] ) {
1997                                wp_enqueue_script( 'google-recaptcha-
v3', 'https://www.google.com/recaptcha/api.js?render=' . $options['recaptchaSite'], array() );
1998
1999                                $htmlCode .= '<input type="hidden" name="recaptcha_response"
id="recaptchaResponseLogin">';
2000
2001                                $htmlCode .= '<script>
2002 function onFormSubmitClick() {
2003         grecaptcha.ready(function() {
2004           grecaptcha.execute("' . esc_attr( $options['recaptchaSite'] ). '", {action:
"password"}).then(function(token) {
2005                   var recaptchaResponse = document.getElementById("recaptchaResponseLogin");
2006                   recaptchaResponse.value = token;
2007                   console.log("VideoWhisper Login: Google reCaptcha v3 updated", token,
recaptchaResponse);
2008                   var ticketForm = document.getElementById("passwordForm");
2009                   ticketForm.submit();
2010           });
2011         });
2012       }
2013 </script>
2014 <noscript>JavaScript is required to use <a href="https://paidvideochat.com/">PaidVideochat turnkey live
cam site solution</a>. Contact <a href="https://videowhisper.com/">VideoWhisper Online Platforms</a> for
clarifications.</noscript>';
2015                        } else {
2016                                // recaptcha disabled
2017                                $htmlCode .= '<script>
2018 function onFormSubmitClick(e) {
2019           console.log("VideoWhisper Login: Google reCaptcha v3 disabled");
2020                   var ticketForm = document.getElementById("passwordForm");
2021                   ticketForm.submit();
2022       }
2023 </script>
2024 <noscript>JavaScript is required to use <a href="https://paidvideochat.com/">PaidVideochat turnkey live
cam site solution</a>. Contact <a href="https://videowhisper.com/">VideoWhisper Online Platforms</a> for
clarifications.</noscript>
2025 ';
```

```
2026                        }
2027
2028                        if ( $error ) {
2029                                $htmlCode .= '<div class="ui warning message">
2030    <div class="header">' . __( 'Error', 'ppv-live-webcams' ) . ':</div>
2031    <ul class="list">
2032      ' . $error . '
2033    </ul>
2034  </div>';
2035                        }
2036
2037                        $htmlCode .= '
2038  <div class="field">
2039    <label>' . __( 'Email', 'ppv-live-webcams' ) . '</label>
2040    <input type="text" name="email" value="' . esc_attr($email) . '" placeholder="' . __( 'Email', 'ppv-
live-webcams' ) . '">
2041    ' . __( 'Provide account email to request password change.', 'ppv-live-webcams' ) . '
2042  </div> ';
2043
2044                        $htmlCode .= '
2045  <div class="field">
2046  <input type="hidden" name="videowhisper_password" value="email">
2047  <button type="button" id="submitButton" name="submitButton" class="ui submit button"
onclick="onFormSubmitClick()">
2048  <i class="mail icon"></i>
2049  ' . __( 'Send', 'ppv-live-webcams' ) . '
2050</button>
2051   </div>
2052  ';
2053
2054                        $htmlCode .= '
2055                                <div class="field">';
2056                        $htmlCode .= ' <a class="ui tiny button basic"
href="' . esc_url( $atts['terms'] ) . '"> <i class="clipboard list icon"></i> ' . __( 'Terms of
Use', 'ppv-live-webcams' ) . '</a> ';
2057                        $htmlCode .= ' <a class="ui tiny button basic"
href="' . esc_url( $atts['privacy'] ) . '"> <i class="clipboard list icon"></i> ' . __( 'Privacy
Policy', 'ppv-live-webcams' ) . '</a> ';
2058                        $htmlCode .= '<br/></div>';
2059
2060                        $htmlCode .= '<br/></form>';
2061
2062                        // end email form
2063                }
2064
2065                return $htmlCode;
2066
2067        }
2068
2069
2070        static function videowhisper_login_form( $atts ) {
2071
2072                $options = get_option( 'VWliveWebcamsOptions' );
2073
2074                // shortocode attributes
2075                $atts = shortcode_atts(
2076                        array(
2077                                'role'     => '',
2078                                'register' => 0,
2079                                'terms'    => get_permalink( $options['p_videowhisper_terms'] ),
2080                                'privacy'  => get_permalink( $options['p_videowhisper_privacy'] ),
2081                        ),
2082                        $atts,
```

```
2083                        'videowhisper_login_form'
2084                );
2085                $atts = array_map('esc_attr', $atts);
2086                $atts['terms'] = esc_url($atts['terms']);
2087                $atts['privacy'] = esc_url($atts['privacy']);
2088
2089                self::enqueueUI();
2090                $this_page = self::getCurrentURL();
2091                $htmlCode  = '';
2092
2093                if ( ! $options['loginFrontend'] ) {
2094                        $htmlCode .= '<div class="ui ' . $options['interfaceClass'] . ' segment">';
2095                        $htmlCode .= __( 'Frontend login is disabled from settings.', 'ppv-live-
webcams' );
2096                        $htmlCode .= '<br /> <a class="ui button qbutton"
href="' . wp_login_url() . '">' . __( 'Login', 'ppv-live-webcams' ) . '</a>';
2097                        $htmlCode .= '</div>';
2098                        return $htmlCode;
2099                }
2100
2101                if ( $_GET['logout'] ?? false ) {
2102                        if ( wp_verify_nonce( $_GET['verify'], 'logout' . $_GET['logout'] ) ) {
2103                                $htmlCode .= '<div class="ui ' . $options['interfaceClass'] . '
segment">' . __( 'You logged out.', 'ppv-live-webcams' ) . '</div>';
2104                        } else {
2105                                return 'Logout nonce error.';
2106                        }
2107                }
2108
2109                // already logged in?
2110                $current_user = wp_get_current_user();
2111                if ( $current_user->ID ) {
2112                        $htmlCode .= '<div class="ui ' . $options['interfaceClass'] . ' segment"> <i
class="user icon"></i>';
2113
2114                        $htmlCode .= __( 'You are currently logged in as', 'ppv-live-webcams' ) . ':
' . $current_user->user_login;
2115                        $htmlCode .= '<BR /><BR /> <a class="ui button"
href="' . wp_nonce_url( add_query_arg( 'logout', $current_user-
>ID, $this_page ), 'logout' . $current_user->ID, 'verify' ) . '"> <i class="sign out icon">
</i>' . __( 'Log Out', 'ppv-live-webcams' ) . '</a> ';
2116
2117                        $htmlCode .= '</div>';
2118
2119                        return $htmlCode;
2120                }
2121
2122                // process login form
2123
2124                $error  = '';
2125                $status = '';
2126
2127                if (isset($_GET['login_error'])) $error = wp_kses(
2128                        $_GET['login_error'],
2129                        array(
2130                                'a'      => array(
2131                                        'href'  => array(),
2132                                        'title' => array(),
2133                                ),
2134                                'br'     => array(),
2135                                'em'     => array(),
2136                                'strong' => array(),
2137                                'li'     => array(),
```

```
2138                            )
2139                    );
2140
2141                    if ( $error ) {
2142                            $status = 'warning';
2143                    }
2144
2145                    // show form
2146                    if ( $_SERVER['REQUEST_METHOD'] != 'POST' || $error ) {
2147
2148                            $htmlCode .= '<form defaultbutton="submitButton" class="ui
' . $options['interfaceClass'] . ' form ' . $status . '" method="post" enctype="multipart/form-data"
action="' . wp_nonce_url( $this_page, 'vw_login', 'videowhisper' ) . '" id="loginForm"
name="loginForm">';
2149
2150                            if ( $options['recaptchaSite'] ) {
2151                                    wp_enqueue_script( 'google-recaptcha-
v3', 'https://www.google.com/recaptcha/api.js?render=' . $options['recaptchaSite'], array() );
2152
2153                                    $htmlCode .= '<input type="hidden" name="recaptcha_response"
id="recaptchaResponseLogin">';
2154
2155                                    $htmlCode .= '<script>
2156 function onLoginSubmitClick(e) {
2157                 e.preventDefault();
2158         grecaptcha.ready(function() {
2159           grecaptcha.execute("' . $options['recaptchaSite'] . '", {action: "login"}).then(function(token)
{
2160                   var recaptchaResponse = document.getElementById("recaptchaResponseLogin");
2161                   recaptchaResponse.value = token;
2162                   console.log("VideoWhisper Login: Google reCaptcha v3 updated", token,
recaptchaResponse);
2163                   var ticketForm = document.getElementById("loginForm");
2164                   ticketForm.submit();
2165           });
2166         });
2167       }
2168 </script>
2169 <noscript>JavaScript is required to use <a href="https://paidvideochat.com/">PaidVideochat turnkey live
cam site solution</a>. Contact <a href="https://videowhisper.com/">VideoWhisper Online Platforms</a> for
clarifications.</noscript>';
2170                            } else {
2171                                    // recaptcha disabled
2172                                    $htmlCode .= '<script>
2173 function onLoginSubmitClick(e) {
2174           e.preventDefault();
2175           console.log("VideoWhisper Login: Google reCaptcha v3 disabled");
2176                   var ticketForm = document.getElementById("loginForm");
2177                   ticketForm.submit();
2178       }
2179 </script>
2180 <noscript>JavaScript is required to use <a href="https://paidvideochat.com/">PaidVideochat turnkey site
solution</a>. Contact <a href="https://videowhisper.com/">VideoWhisper</a> for clarifications.</noscript>
';
2181
2182                            }
2183
2184                            if ( $error ) {
2185                                    $htmlCode .= '<div class="ui warning message">
2186     <div class="header">' . __( 'Login Failed', 'ppv-live-webcams' ) . ':</div>
2187     <ul>
2188       ' . $error . '
2189      </ul>
2190   </div>';
2191                            }
```

```
2192
2193  //preserve inputs
2194  $login = sanitize_text_field( $_POST['login'] ?? '' );
2195  $password = sanitize_text_field( $_POST['password'] ?? '' );
2196
2197                          $htmlCode .= '<div class="field">
2198    <div class="field">
2199      <label>' . __( 'Username', 'ppv-live-webcams' ) . '</label>
2200      <input type="text" name="login" value="' . esc_attr($login) . '" placeholder="">
2201    </div>
2202
2203     <div class="field">
2204      <label>' . __( 'Password', 'ppv-live-webcams' ) . '</label>
2205      <input type="password" name="password" value="" placeholder="">
2206    </div>
2207   ';
2208
2209                          $htmlCode .= '
2210    <div class="field">
2211    <input type="hidden" name="videowhisper_login" value="1">
2212    <button id="submitButton" name="submitButton" class="ui submit button"
      onclick="onLoginSubmitClick(event)">
2213    <i class="sign in icon"></i>
2214    ' . __( 'Login', 'ppv-live-webcams' ) . '
      </button>
2215     </div>
2216    ';
2217
2218
2219                          // <input type="button" id="submitButton" name="submitButton"
      onclick="onLoginSubmitClick()" class="ui submit button" value="" />
2220
2221                          if ( $atts['register'] ) {
2222                                  $htmlCode .= '
2223    <div class="field">';
2224
2225                                  if ( $atts['role'] == $options['roleClient'] || ! $atts['role'] ) {
2226                                          $htmlCode .= ' <a class="ui tiny button basic"
      href="' . get_permalink( $options['p_videowhisper_register_client'] ) . '"> <i class="user plus icon">
      </i> ' . __( 'Register', 'ppv-live-webcams' ) . ' ' . ucwords( $options['roleClient'] ) . '</a> ';
2227                                  }
2228
2229                                  if ( $atts['role'] == $options['rolePerformer'] || ! $atts['role'] ) {
2230                                          $htmlCode .= ' <a class="ui tiny button basic"
      href="' . get_permalink( $options['p_videowhisper_register_performer'] ) . '"> <i class="user plus icon">
      </i> ' . __( 'Register', 'ppv-live-webcams' ) . ' ' . ucwords( $options['rolePerformer'] ) . '</a> ';
2231                                  }
2232
2233                                  if ( $options['studios'] ) {
2234                                          if ( $atts['role'] == $options['roleStudio'] ||
       ! $atts['role'] ) {
2235                                                  $htmlCode .= ' <a class="ui tiny button basic"
      href="' . get_permalink( $options['p_videowhisper_register_studio'] ) . '"> <i class="user plus icon">
      </i> ' . __( 'Register', 'ppv-live-webcams' ) . ' ' . ucwords( $options['roleStudio'] ) . '</a> ';
2236                                          }
2237                                  }
2238
2239                                  if ( $options['p_videowhisper_password_form'] ) {
2240                                          $htmlCode .= ' <a class="ui tiny button basic"
      href="' . get_permalink( $options['p_videowhisper_password_form'] ) . '"> <i class="lock icon"></i>
      ' . __( 'Lost Password', 'ppv-live-webcams' ) . '</a> ';
2241                                  }
2242
```

```
2243                                    $htmlCode .= ' <a class="ui tiny button basic" href="' . esc_url( $atts['terms'] ) . '"> <i class="clipboard list icon"></i> ' . __( 'Terms of Use', 'ppv-live-webcams' ) . '</a> ';
2244                                $htmlCode     .= ' <a class="ui tiny button basic" href="' . esc_url( $atts['privacy'] ) . '"> <i class="clipboard list icon"></i> ' . __( 'Privacy Policy', 'ppv-live-webcams' ) . '</a> ';
2245
2246                                $htmlCode .= '<br/></div>';
2247                        }
2248
2249                        $htmlCode .= '<br/></form>';
2250                        // end show login form
2251                }
2252
2253                return $htmlCode;
2254
2255        }
2256
2257
2258        static function videowhisper_register_form( $atts ) {
2259                //registration form
2260                $options = get_option( 'VWliveWebcamsOptions' );
2261
2262                // shortocode attributes
2263                $atts = shortcode_atts(
2264                        array(
2265                                'role'    => $options['roleClient'],
2266                                'terms'   => get_permalink( $options['p_videowhisper_terms'] ),
2267                                'privacy' => get_permalink( $options['p_videowhisper_privacy'] ),
2268                        ),
2269                        $atts,
2270                        'videowhisper_register_form'
2271                );
2272                $atts = array_map('esc_attr', $atts);
2273                $atts['terms'] = esc_url($atts['terms']);
2274                $atts['privacy'] = esc_url($atts['privacy']);
2275
2276                self::enqueueUI();
2277
2278                if ( ! $options['registerFrontend'] ) {
2279                        return '<div class="ui ' . $options['interfaceClass'] . ' segment">' . __( 'Frontend registration is disabled!', 'ppv-live-webcams' ) . '</div>';
2280                }
2281                if ( $atts['role'] == $options['roleStudio'] && ! $options['studios'] ) {
2282                        return '<div class="ui ' . $options['interfaceClass'] . ' segment">' . __( 'Studio registration is disabled!', 'ppv-live-webcams' ) . '</div>';
2283                }
2284
2285                $error    = '';
2286                $status   = '';
2287                $htmlCode = '';
2288
2289                if ( $_SERVER['REQUEST_METHOD'] == 'POST' ) {
2290
2291                        if ( ! wp_verify_nonce( $_GET['videowhisper'], 'vw_register' ) ) {
2292                                $error .= '<li>' . __( 'Nonce incorrect for registration.', 'ppv-live-webcams' ) . '</li>';
2293                        }
2294
2295                        if (is_user_logged_in()) $error .= '<li>' . __( 'You are already have an account and are logged in.', 'ppv-live-webcams' ) . '</li>';
2296
2297                        $login     = sanitize_text_field( $_POST['login'] ?? '');
```

```
Line
2298                        $firstname = sanitize_text_field( $_POST['firstname'] ?? '' );
2299                        $lastname  = sanitize_text_field( $_POST['lastname'] ?? '' );
2300                        $password  = sanitize_text_field( $_POST['password'] ?? '');
2301                        $password2 = sanitize_text_field( $_POST['password2'] ?? '');
2302
2303                        if (!$firstname) $firstname = $login;
2304
2305                        $email = sanitize_email( $_POST['email'] ?? '') ;
2306                        $terms = sanitize_text_field( $_POST['terms'] ?? '');
2307
2308                        if ( ! $login ) {
2309                                $error .= '<li>' . __( 'A login (username) is required.', 'ppv-live-webcams' ) . '</li>';
2310                        }
2311                        if ( ! $terms ) {
2312                                $error .= '<li>' . __( 'Accepting Terms of Use is required.', 'ppv-live-webcams' ) . '</li>';
2313                        }
2314
2315                        if ( ! $password ) {
2316                                $error .= '<li>' . __( 'A password is required.', 'ppv-live-webcams' ) . '</li>';
2317                        }
2318
2319                        if ( $password != $password2 ) {
2320                                $error .= '<li>' . __( 'Password verification does not match.', 'ppv-live-webcams' ) . '</li>';
2321                        }
2322
2323                        if ( ! $email ) {
2324                                $error .= '<li>' . __( 'A valid email address is required.', 'ppv-live-webcams' ) . '</li>';
2325                        }
2326
2327
2328
2329                        if ( $options['registerIPlimit'] ) {
2330                                $users = get_users(
2331                                        array(
2332                                                'meta_key'     => 'videowhisper_ip_register',
2333                                                'meta_value'   => self::get_ip_address(),
2334                                                'meta_compare' => '=',
2335                                        )
2336                                );
2337                                if ( count( $users ) >= intval( $options['registerIPlimit'] ) ) {
2338                                        $error .= '<li>' . __( 'Registration per IP limit reached.', 'ppv-live-webcams' ) . ' #' . count( $users ) . '</li>';
2339                                }
2340                        }
2341
2342                        if ( $options['recaptchaSite'] ) {
2343
2344                                if ( isset( $_POST['recaptcha_response'] ) ) {
2345                                        if ( $_POST['recaptcha_response'] ) {
2346                                                // Build POST request:
2347                                                $recaptcha_response = sanitize_text_field( $_POST['recaptcha_response'] );
2348
2349                                                // Make and decode POST request:
2350                                                // $recaptcha = file_get_contents('https://www.google.com/recaptcha/api/siteverify' . '?secret=' . $options['recaptchaSecret'] . '&response=' . $recaptcha_response);
2351                                                // $recaptchaD = json_decode($recaptcha);
```

```
2352                                                $response = wp_remote_get( 'https://www.google.com/recaptcha/api/siteverify' . '?secret=' . $options['recaptchaSecret'] . '&response=' . $recaptcha_response );
2353                                                $body       = wp_remote_retrieve_body( $response );
2354                                                $recaptchaD = json_decode( $body );
2355
2356                                                // Take action based on the score returned:
2357                                                if ( isset($recaptchaD->score) && $recaptchaD->score >= 0.3 ) {
2358                                                        // Verified
2359                                                        $htmlCode .= '<!-- VideoWhisper reCAPTCHA v3 score: ' . $recaptchaD->score . '-->';
2360
2361                                                } else {
2362                                                        // Not verified - show form error
2363                                                        $error .= '<li>Google reCAPTCHA v3 Failed. Score: ' . ( $recaptchaD->score ?? 'n/a') . ' . Try again or using a different browser!</li>';
2364                                                }
2365                                        } else {
2366                                                $error .= '<li>Google reCAPTCHA v3 empty. Make sure you have JavaScript enabled or try a different browser!</li>';
2367                                        }
2368                                } else {
2369                                        $error .= '<li>Google reCAPTCHA v3 missing. Make sure you have JavaScript enabled or try a different browser!</li>';
2370                                }
2371                        }
2372
2373                        $persons = intval($_POST['persons'] ?? 0);
2374
2375                        //pre-check if required records were filled
2376                        $requiredPresent = true;
2377                        if ( $atts['role'] == $options['rolePerformer'] || $atts['role'] == $options['roleStudio'] )
2378                                if ( is_array( $options['recordFields'] ) )
2379 foreach ( $options['recordFields'] as $field => $parameters ) if ( $parameters['required'] ?? false ) {
2380
2381                                                $fieldName = sanitize_title( trim( $field ) );
2382
2383 if ($options['multiPerson'] && isset($parameters['personal']) && $parameters['personal'] )
2384                                                {
2385                                                        for ($pi=1; $pi <= $persons; $pi++)
2386                                                        {
2387 $fieldName = sanitize_title( trim( $field ) ) . '_' . $pi ;
2388
2389 if ( $parameters['type'] =='file')
2390                                                                if  (!isset($_FILES[ $fieldName ]['tmp_name']) || !$_FILES[ $fieldName ]['tmp_name'] ) {
2391
2392 $error .= '<li>' . $field . ' : ' . __( 'Required file is missing!', 'ppv-live-webcams' ) . ' '. $pi . ' / '. $persons . '</li>';
2393                                                                }
2394                                                                else
2395                                                                {
2396                                                                        //pre-check extension and size
2397                                                                        $ext = strtolower( pathinfo( sanitize_file_name( $_FILES[ $fieldName ]['name'] ), PATHINFO_EXTENSION ) );
2398 $allowed = array( 'jpg', 'jpeg', 'png', 'gif', 'pdf' );
2399
```

```
Line
2400  if ( ! in_array( $ext, $allowed ) ) {
2401  $error .= '<li>' . esc_html( $field ) . ' / ' . esc_html( $_FILES[ $fieldName ]['name'] ) . ': Unsupported file extension! Please use one of these formats: ' . implode( ', ', $allowed ) . ' '. $pi . ' / '. $persons . '</li>';
2402                                                
```

```
2445                                            }
2446
2447                                            }
2448
2449                        if ( $error ) {
2450                                $status = 'warning';
2451                        } else {
2452
2453                                // Get plugin options for role validation
2454                                $options = get_option( 'VWliveWebcamsOptions' );
2455
2456                                // Build whitelist of allowed registration roles
2457                                $allowed_roles = array();
2458
2459                                // Add core roles
2460                                if ( ! empty( $options['roleClient'] ) ) {
2461                                        $allowed_roles[] = trim( $options['roleClient'] );
2462                                }
2463                                if ( ! empty( $options['rolePerformer'] ) ) {
2464                                        $allowed_roles[] = trim( $options['rolePerformer'] );
2465                                }
2466
2467                                // Add studio role if enabled
2468                                if ( ! empty( $options['studios'] ) && ! empty( $options['roleStudio'] ) ) {
2469                                        $allowed_roles[] = trim( $options['roleStudio'] );
2470                                }
2471
2472                                // Add additional performer roles
2473                                if ( ! empty( $options['rolePerformers'] ) ) {
2474                                        $additional_performers = explode( ',', $options['rolePerformers'] );
2475                                        foreach ( $additional_performers as $role ) {
2476                                                $role = trim( $role );
2477                                                if ( ! empty( $role ) ) {
2478                                                        $allowed_roles[] = $role;
2479                                                }
2480                                        }
2481                                }
2482
2483                                // Add moderator roles
2484                                if ( ! empty( $options['roleModerators'] ) ) {
2485                                        $moderator_roles = explode( ',', $options['roleModerators'] );
2486                                        foreach ( $moderator_roles as $role ) {
2487                                                $role = trim( $role );
2488                                                if ( ! empty( $role ) ) {
2489                                                        $allowed_roles[] = $role;
2490                                                }
2491                                        }
2492                                }
2493
2494                                // Remove duplicates and sanitize
2495                                $allowed_roles = array_unique( array_map( 'sanitize_key', $allowed_roles ) );
2496
2497                                // Validate the requested role
2498                                $requested_role = sanitize_key( trim( $atts['role'] ) );
2499
2500                                // Explicitly reject privileged roles
2501                                $prohibited_roles = array( 'administrator', 'super-
2502 admin', 'super_admin' );
```

```
2503
2504                                    if ( in_array( $requested_role, $prohibited_roles, true ) ) {
2505                                            $error .= '<li>' . __( 'Registration with administrator or super-admin roles is not allowed.', 'ppv-live-webcams' ) . '</li>';
2506                                    } elseif ( ! in_array( $requested_role, $allowed_roles, true ) ) {
2507                                            $error .= '<li>' . __( 'The specified role is not allowed for registration.', 'ppv-live-webcams' ) . '</li>';
2508                                    }
2509
2510                                    if ( $error ) {
2511                                            $status = 'warning';
2512                                    } else {
2513                                            // create role if missing (now validated)
2514                                            if ( ! get_role( $requested_role ) ) {
2515                                                    add_role( $requested_role, ucwords( $requested_role ), array( 'read' => true ) );
2516                                            }
2517
2518                                            // register user
2519                                            $newuser = array(
2520                                                    'user_pass'  => $password,
2521                                                    'user_login' => $login,
2522                                                    'user_email' => $email,
2523                                                    'first_name' => $firstname,
2524                                                    'last_name'  => $lastname,
2525                                                    'role'       => $requested_role,
2526                                            );
2527                                    }
2528                            }
2529
2530                            if ( ! $error ) {
2531
2532                                    $user_id = wp_insert_user( $newuser );
2533
2534                                    if ( $user_id && ! is_wp_error( $user_id ) )
2535                                    if ( !$options['registrationNoActivation'] || ( $options['registrationNoActivation'] == 'client' &&  $atts['role'] != $options['roleClient'] ) )
2536                                    {
2537                                            $code = sha1( $user_id . time() );
2538
2539                                            add_user_meta( $user_id, 'videowhisper_activate', $code, true ); // mark activation required
2540                                            update_user_meta( $user_id, 'videowhisper_ip_register', self::get_ip_address() ); // save ip
2541                                            update_user_meta( $user_id, 'created_by', 'ppv-live-webcams/register_form' ); // for troubleshooting registration origin
2542                                            update_user_meta( $user_id, 'videowhisper_role', $atts['role'] ); // videowhisper registration role
2543
2544                                            $activation_link = add_query_arg(
2545                                                    array(
2546                                                            'key'  => $code,
2547                                                            'user' => $user_id,
2548                                                    ),
2549                                                    get_permalink($options['p_videowhisper_register_activate'])
2550                                            );
2551
2552                                            $headers   = array();
2553                                            $headers[] = 'Content-Type: text/plain; charset=utf-8';
2554                                            $headers[] = 'Content-Transfer-Encoding: bit-8';
2555
2556                                            $message = $options['activateText'] . "\r\n" . $activation_link;
2557
```

```
2558                                    wp_mail($email, $options['activateSubject'], $message, $headers);
2559
2560                                    $htmlCode .= '<div class="ui ' . $options['interfaceClass'] . '
segment">' . esc_html($options['activateMessage']) . '</div>';
2561                                }
2562                                else $htmlCode .= '<div class="ui ' . $options['interfaceClass'] . '
segment">' . __( 'Your account was registered.', 'ppv-live-
webcams' ) . '</div>' . do_shortcode( '[videowhisper_login_form register="0"]' );
2563
2564
2565                                if ( is_wp_error( $user_id ) ) {
2566                                        $status = 'warning';
2567                                        $error .= '<li>' . $user_id->get_error_message() . '</li>';
2568                                } else {
2569
2570                                        //user created
2571                                        $user   = get_userdata( $user_id );
2572
2573                                // process administrative records
2574                                        if ( $atts['role'] == $options['rolePerformer'] ||
 $atts['role'] == $options['roleStudio'] ) {
2575                                                if ( is_array( $options['recordFields'] ) ) {
2576
2577                                                        $error2 = '';
2578
2579                                                        $persons = 1;
2580                                                        $personal = false;
2581                                                        if ($options['multiPerson'])
2582                                                        {
2583
$persons = intval($_POST['persons'] ?? 1);
2584                                                                update_user_meta( $user-
>ID, 'persons', $persons );
2585
2586                                                        //set types for persons
2587
for ($pi = 1; $pi <= $persons; $pi++) if (isset($_POST['person_' . $pi])) update_user_meta( $user-
>ID, 'person_' . $pi, sanitize_text_field( $_POST['person_' . $pi] ) );
2588
2589                                                        }
2590
2591                                                        if ( $user ) {
2592
foreach ( $options['recordFields'] as $field => $parameters ) {
2593
2594
$fieldName = sanitize_title( trim( $field ) ); //base fieldname
2595
2596                                                                $maxperson = $persons;
2597
if ( $options['multiPerson'] && isset($parameters['personal']) && $parameters['personal'] ) $personal = t
rue; else $personal = false;
2598
if (!$personal) $maxperson = 1; // only 1 iteration
2599
2600
for ($pi = 1; $pi <= $maxperson; $pi++)
2601                                                                {
2602
2603                                                                //start person
2604
if ($options['multiPerson'] && $personal) $fieldName = sanitize_title( trim( $field ) ) . '_' . $pi; //mu
ltiple person: add person index to field name
2605
2606                                                                switch ( $parameters['type'] ) {
```

```
2607                                                                    case 'file':
2608 $uploadSuccess = false;
2609 if ( $_FILES[ $fieldName ]['tmp_name'] ?? false && $filename =  $_FILES[ $fieldName ]['tmp_name'] ) {
2610 $deny = false;
2611
2612                                                                                                                                    $ext
      = strtolower( pathinfo( $_FILES[ $fieldName ]['name'] , PATHINFO_EXTENSION  ) );
2613 $allowed = array( 'jpg', 'jpeg', 'png', 'gif', 'pdf' );
2614
2615 if ( ! in_array( $ext, $allowed ) ) {
2616 $error2 .= '<br>' . $fieldName . ' - ' . $filename . ': Unsupported file extension! Please use one of
     these formats: ' . implode( ', ', $allowed );
2617 $deny    = true;
2618                                                                                                                                    }
2619
2620 $maxUpload = intval ($options['maxUpload'] ?? 6000) * 1000;
2621 if ( $_FILES[ $fieldName ]['size'] > $maxUpload ) {
2622 $error2 .= '<br>' . esc_html( $fieldName ) . ': File is too big! Please use a file under
     ' . $maxUpload . 'kb.';
2623 $deny      = true;
2624                                                                                                                                    }
2625
2626 if ( ! $deny ) {
2627 $destination = $options['uploadsPath'] . '/user_' .  $user->user_nicename . '_' .  $user->ID  . '/';
2628 if ( ! file_exists( $destination ) ) {
2629       mkdir( $destination );
2630                                                                                                                                
```

```
2647                }
2648        }
2649
2650        update_user_meta( $user->ID, 'vwf_' . $fieldName, $newpath );
2651        $uploadSuccess = true;
2652                                                                                                                }
2653                                                                                                            }
2654                                                                                                    }
2655
2656 if ($parameters['required'] ?? false) if (!$uploadSuccess) $error2 .= '<br>' . esc_html( $fieldName ) . ': ' . 'Required file not uploaded!';
2657                                                                                                        break;
2658
2659                                                                                                    default:
2660 if ( isset( $_POST[ $fieldName ] ) )
2661 if ( is_string($_POST[ $fieldName ]) ) {
2662 update_user_meta( $user->ID, 'vwf_' . $fieldName, sanitize_textarea_field( $_POST[ $fieldName ] ) );
2663 } else $error2 .= '<br>' . esc_html( $fieldName ) . ': ' . ' Invalid value for field.';
2664                                                                                                        }
2665                                                                                                        //end person
2666                                                                                                 }
2667
2668                                                                                                        // end field
2669                                                                                                }
2670                                                                                        }
2671
2672                                                                                        update_user_meta( $user->ID, 'vwUpdated', time() );
2673                                                                                        self::usersPendingApproval();
2674
2675                                                                                        if ( $error2 ) {
2676                                                                                                $htmlCode .= '<div class="ui ' . $options['interfaceClass'] . ' segment">' . __( 'Some error occurred related to uploads. After email confirmation and login you can retry from Account Records section in Dashboard.', 'ppv-live-webcams' ) . $error2 . '</div>';
2677                                                                                        }
2678
2679                                                                                        // end records
2680
2681                                                                                }
2682                                                                        } //end performer administrative records
2683
2684                                                                        $userID = $user_id;
2685
2686                                                                        //create first room for performer
2687                                                                        if ( $atts['role'] == $options['rolePerformer'] )
2688                                                                        {
2689                                                                                $performerName = self::performerName( $user, $options );
2690 $postID = self::webcamPost( $performerName, $performerName, $userID );
2691                                                                        }
2692
2693
2694                                                //save Mathmaking options
2695                                                if ( $options['match'] )
2696                                                {
```

```
2697
2698                                foreach ( ['me', 'you'] as $side)
2699                                        {
2700
2701                                        $fieldex = '';
2702                                        if ($side == 'you') $fieldex = 'm_'; //match
2703
2704 foreach ( $options['profileFields'] as $field => $parameters ) if ( isset( $parameters['match'] ) ) if ( $parameters['match'] != 'mirror' || $side == 'me' )
2705                                                {
2706
2707 $fieldName = $fieldex . sanitize_title( trim( $field ) );
2708
2709 if ( isset( $_POST[ $fieldName ] ) ) {
2710 if ( $_POST['matchFor'] ) {
2711 if ( in_array( $parameters['type'], array('checkboxes', 'multiselect') ) )
2712                                                                                                                                                {
2713         $tags = (array) $_POST[ $fieldName ];
2714         if ( is_array( $tags ) ) {
2715                 foreach ( $tags as &$tag ) {
2716                         $tag = sanitize_text_field( $tag );
2717                 }
2718                 unset( $tag );
2719         } else {
2720                 $tags = sanitize_text_field( $tags );
2721         }
2722         if ($postID ?? false) update_post_meta( $postID, 'vwf_' . $fieldName, $tags );
2723         else update_user_meta( $userID, 'vwf_'  . $fieldName, $tags );
2724
2725 } else {
2726         if ($postID ?? false)
update_post_meta( $postID, 'vwf_' . $fieldName, sanitize_text_field( $_POST[ $fieldName ] ) );
2727         else update_user_meta( $userID, 'vwf_' . $fieldName, sanitize_text_field( $_POST[ $fieldName ] )
);
2728                                                                                                                                                }
2729                                                                                                                                        }
2730                                                                                                                                }
2731
2732                                                        }
2733                                                }
2734
2735                                        }
2736
2737                                }
2738                        }
2739                }
2740
2741                 if ( $_SERVER['REQUEST_METHOD'] != 'POST' || $error != '' ) {
```

```
2742
2743                        $this_page = self::getCurrentURL();
2744
2745                        $htmlCode .= '<form class="ui ' . $options['interfaceClass'] . ' form
' . $status . '" method="post" enctype="multipart/form-data"
action="' . wp_nonce_url( $this_page, 'vw_register', 'videowhisper' ) . '" id="registrationForm"
name="registrationForm">';
2746                        $htmlCode .= '<h4 class="ui dividing header"> <i class="user plus icon"></i>
' . __( 'Registration', 'ppv-live-webcams' ) . ': ' . ucwords( $atts['role'] ) . '</h4>';
2747
2748                        if ( $options['recaptchaSite'] ) {
2749                                wp_enqueue_script( 'google-recaptcha-
v3', 'https://www.google.com/recaptcha/api.js?render=' . $options['recaptchaSite'], array() );
2750
2751                                $htmlCode .= '<input type="hidden" name="recaptcha_response"
id="recaptchaResponse">';
2752
2753                                $htmlCode .= '<script>
2754 function onSubmitClick(e) {
2755         grecaptcha.ready(function() {
2756           grecaptcha.execute("' . $options['recaptchaSite'] . '", {action:
"register"}).then(function(token) {
2757                   var recaptchaResponse = document.getElementById("recaptchaResponse");
2758                   recaptchaResponse.value = token;
2759                   console.log("VideoWhisper Registration: Google reCaptcha v3 updated", token);
2760                   var ticketForm = document.getElementById("registrationForm");
2761                   ticketForm.submit();
2762           });
2763         });
2764       }
2765 </script>
2766 <noscript>JavaScript is required to use <a href="https://paidvideochat.com/">PaidVideochat turnkey cam
site solution</a>. Contact <a href="https://videowhisper.com/">VideoWhisper live streaming
platforms</a> for clarifications.</noscript>
2767 ';
2768                        } else {
2769                                // recaptcha disabled
2770                                $htmlCode .= '<script>
2771 function onSubmitClick(e) {
2772           console.log("VideoWhisper Registration: Google reCaptcha v3 disabled");
2773                   var ticketForm = document.getElementById("registrationForm");
2774                   ticketForm.submit();
2775
2776       }
2777 </script>
2778 <noscript>JavaScript is required to use <a href="https://paidvideochat.com/">PaidVideochat turnkey cam
site solution</a>. Contact <a href="https://videowhisper.com/">VideoWhisper live streaming
platforms</a> for clarifications.</noscript>
2779 ';
2780                        }
2781
2782                        if ( $error ) {
2783                                $htmlCode .= '<div class="ui warning message">
2784    <div class="header">' . __( 'Could not submit registration', 'ppv-live-webcams' ) . ':</div>
2785    <ul class="list">
2786      ' . $error . '
2787    </ul>
2788  </div>';
2789                        }
2790
2791                        if ( !$options['registrationAanonymous'] ||
 ( $options['registrationAanonymous'] == 'client' &&  $atts['role'] != $options['roleClient'] ) )
2792                        $htmlCode .= '<div class="field">
2793    <label>' . __( 'First Name', 'ppv-live-webcams' ) . '</label>
```

```
2794    <input type="text" name="firstname" value="' . ( $firstname ?? '' ) . '" placeholder="' . __( 'First Name', 'ppv-live-webcams' ) . '">
2795  </div>
2796  <div class="field">
2797    <label>' . __( 'Last Name', 'ppv-live-webcams' ) . '</label>
2798    <input type="text" name="lastname" value="' . ( $lastname ?? '' ). '" placeholder="' . __( 'Last Name', 'ppv-live-webcams' ) . '">
2799  </div>';
2800
2801 $htmlCode .='<div class="field">
2802    <label for="login">' . __( 'Username', 'ppv-live-webcams' ) . ' <div class="ui button tiny basic compact icon" data-tooltip="' . __( 'Required', 'ppv-live-webcams' ) . '"><i class="warning circle icon"></i></div>
2803        </label>
2804    <input type="text" name="login" value="' . ( $login ?? '' ). '" placeholder="' . __( 'Username', 'ppv-live-webcams' ) . '">
2805  </div>
2806
2807   <div class="field">
2808    <label for="password">' . __( 'Password', 'ppv-live-webcams' ) . ' <div class="ui button tiny basic compact icon" data-tooltip="' . __( 'Required', 'ppv-live-webcams' ) . '"><i class="warning circle icon"></i></div>
2809        </label>
2810    <input type="password" name="password" value="' . ( $password ?? '' ). '" placeholder="' . __( 'Password', 'ppv-live-webcams' ) . '">
2811  </div>
2812
2813  <div class="field">
2814    <label for="password2">' . __( 'Password Verification', 'ppv-live-webcams' ) . ' <div class="ui button tiny basic compact icon" data-tooltip="' . __( 'Required', 'ppv-live-webcams' ) . '"><i class="warning circle icon"></i></div>
2815        </label>
2816    <input type="password" name="password2" value="' . ( $password2 ?? '' ) . '" placeholder="' . __( 'Password Verification', 'ppv-live-webcams' ) . '">
2817  </div>
2818
2819  <div class="field">
2820    <label for="email">' . __( 'Email', 'ppv-live-webcams' ) . ' <div class="ui button tiny basic compact icon" data-tooltip="' . __( 'Required', 'ppv-live-webcams' ) . '"><i class="warning circle icon"></i></div>
2821        </label>
2822    <input type="text" name="email" value="' . ( $email ?? '' ) . '" placeholder="' . __( 'Email', 'ppv-live-webcams' ) . '">
2823  </div> ';
2824
2825                        if ( $atts['role'] == $options['rolePerformer'] || $atts['role'] == $options['roleStudio'] ) {
2826                                if ( is_array( $options['recordFields'] ) ) {
2827
2828                                        $htmlCode .= '<h4 class="ui dividing header"> <i class="address book icon"></i> ' . __( 'Administrative Records', 'ppv-live-webcams' ) . '</h4>';
2829
2830                                        $htmlCode .= '<div class="field"><label>' . __( 'Administrative records are only accessible to site administrators for approval and administrative use.', 'ppv-live-webcams' ) . ' <a target="terms" href="' . $atts['privacy'] . '"> <i class="clipboard list icon"></i> ' . __( 'Privacy Policy', 'ppv-live-webcams' ) . '</a></label></div>';
2831
2832                                        if (!$options['multiPerson'])
2833                                        foreach ( $options['recordFields'] as $field => $parameters ) {
2834
2835                                                $fieldName = sanitize_title( trim( $field ) );
2836
2837                                                // if ($parameters['instructions']) $htmlInstructions = ' data-tooltip="' . htmlspecialchars(stripslashes($parameters['instructions'])). '"'; else
2838                                                $htmlInstructions = '';
2839
```

```
Line
2840                                        $htmlIcons = '';
2841                                        if ($parameters['required'] ?? false) $htmlIcons .= '
<div class="ui button tiny basic compact icon" data-tooltip="' . __( 'Required', 'ppv-live-
webcams' ) . '"><i class="warning circle icon"></i></div>';
2842
2843                                        $htmlCode .= '<div
class="field"' . $htmlInstructions . '><label
for="' . $fieldName . '">' . $field . $htmlIcons . '</label>' ;
2844
2845
$fieldValue = sanitize_textarea_field( $_POST[ $fieldName ] ?? '');
2846
2847                                        switch ( $parameters['type'] ) {
2848                                                case 'file':
2849                                                        $htmlCode .= '<INPUT class="ui button"
type="file" name="' . $fieldName . '" id="' . $fieldName . '">';
2850                                                        break;
2851
2852                                                case 'text':
2853                                                        $htmlCode .= '<INPUT type="text"
size="72" name="' . $fieldName . '" id="' . $fieldName . '" value="' . $fieldValue . '">';
2854                                                        break;
2855
2856                                                case 'textarea':
2857                                                        $htmlCode .= '<TEXTAREA type="text"
rows="3" cols="70" name="' . $fieldName . '" id="' . $fieldName . '">' . $fieldValue . '</TEXTAREA>';
2858                                                        break;
2859
2860                                                case 'select':
2861                                                        $htmlCode    .= '<SELECT
name="' . $fieldName . '" id="' . $fieldName . '" class="ui dropdown v-select">';
2862
$fieldOptions = explode( '/', stripslashes( $parameters['options'] ) );
2863
2864                                                        $htmlCode .= '<OPTION value=""
' . ( ! $fieldValue ? 'selected' : '' ) . '> - </OPTION>';
2865
2866
foreach ( $fieldOptions as $fieldOption ) {
2867                                                                $htmlCode .= '<OPTION
value="' . htmlspecialchars( $fieldOption ) . '"
' . ( $fieldOption == $fieldValue ? 'selected' : '' ) . '>' . htmlspecialchars( $fieldOption ) . '</OPTIO
N>';
2868                                                        }
2869
2870                                                        $htmlCode .= '</SELECT>';
2871                                                        break;
2872                                        }
2873                                        if ( $parameters['instructions'] ?? false ) {
2874                                                $htmlCode .= '<div class="ui basic small label
pointing">' . htmlspecialchars( stripslashes( $parameters['instructions'] ) ) . '</div>';
2875                                        }
2876
2877                                        $htmlCode .= '</div>';
2878
2879                                }
2880                                else
2881                                {
2882                                        //multi person account records
2883                                        $htmlCode .= '<div id="typesSelect">';
2884
2885                                        $htmlCode .= '<h5 class="ui dividing header"> <i
class="group icon"></i>' . __( 'Account Persons', 'ppv-live-webcams' ) .  '</h4>';
2886
2887                                        $htmlCode .= '<table class="">';
2888
```

```
2889                                            $ajaxurl = '"' . admin_url( 'admin-ajax.php?action=vmls_recordsform&uid=0' ) . '"';
2890                                            $ajaxurl0 = '';
2891
2892                                            $personsCode = '';
2893
2894                                            $personTypes = explode (',', $options['personTypes']);
2895                                            foreach ($personTypes as $personType)
2896                                            {
2897                                                    $type = trim($personType);
2898                                                    $typeVar = sanitize_title_with_dashes($type);
2899                                                    $htmlCode .= '
2900                                                     <tr><th class="center aligned">' . $type . '</th>
2901                                                     <td>
2902                                                     <div class="inline field increment">
2903                                                            <input type="button" class="ui button decrease" value="-">
2904                                                            <input id=' . $typeVar . ' class="ui number compact" type="number" value="0" min="0">
2905                                                            <input type="button" class="ui button increase" value="+">
2906                                                       </div>
2907                                                    </td>
2908                                                    </tr>
2909                                                    ';
2910
2911                                                    $ajaxurl .= '+ "&' . $typeVar . '=" + jQuery("#' . $typeVar . '").val()';
2912                                                    $personsCode .=   ' + Number(jQuery("#' . $typeVar . '").val())';
2913
2914                                            if ($_POST['persons_' . $typeVar] ?? false) $ajaxurl0 .= '&' . $typeVar . '=' . intval($_POST['persons_' . $typeVar]);
2915                                            }
2916
2917                                            $htmlCode .= '</table>';
2918
2919                                            $htmlCode .= '<p>'. __( 'Specify persons that will be using this account and then use button below to load personal forms.', 'ppv-live-webcams' ) . '</p>';
2920
2921                                            $htmlCode .= '<input id="fillButton" type="button" class="ui button" value="' . __( 'Fill Personal Records', 'ppv-live-webcams' ) . '">';
2922
2923                                            $htmlCode .= '</div>';
2924
2925                                            $htmlCode .= '<div id="videowhisperRecordsForm"></div>';
2926
2927
2928                                            $htmlCode .= '<script>
2929                                            jQuery(".increment .decrease").on("click", function(e) {
2930                                                    e.preventDefault();
2931                                                    var num = jQuery(this).siblings(".number");
2932                                                    num.val(function(i, oldval) {
2933                                                      return Math.max(0,--oldval);
2934                                                    });
2935                                              });
2936
2937                                                    jQuery(".increment .increase").on("click", function(e) {
2938                                                      e.preventDefault();
2939                                                    var num = jQuery(this).siblings(".number");
2940                                                    num.val(function(i, oldval) {
2941                                                      return Math.max(0,++oldval);
```

```
Line
2942                                                    });
2943                                            });
2944
2945                                            jQuery("#fillButton").on("click", function(e) {
2946                                                    e.preventDefault();
2947                                                    totalPersons = 0 ' . $personsCode . ';
2948                                                    if (totalPersons<1)
2949                                                    {
2950                                                            alert("Please add at least one person type to the account!");
2951                                                            return;
2952                                                    }
2953                                                    aurl = ' . $ajaxurl . ' + "&persons=" + totalPersons;
2954                                                    loadRecordsForm("Loading Records Form", aurl);
2955                                                    jQuery("#fillButton").hide();
2956                                                    jQuery("#typesSelect").hide();
2957                                            });
2958
2959            var loaderRF;
2960
2961            function loadRecordsForm(message, aurl){
2962            if (message)
2963            if (message.length > 0)
2964            {
2965              jQuery("#videowhisperRecordsForm").html(message);
2966            }
2967
2968                    if (loaderRF) loaderRF.abort();
2969
2970                    loaderRF = jQuery.ajax({
2971                            url: aurl,
2972                            data: "interface=recordsForm",
2973                            success: function(data) {
2974                                    jQuery("#videowhisperRecordsForm").html(data);
2975                            }
2976                    });
2977            }
2978
2979                                            </script>';
2980
2981                                            //does not refill post data (ajax)
2982                                            /*
2983                                            if ($ajaxurl0)
2984                                            {
2985                                                    $ajaxurl0 = admin_url( 'admin-ajax.php?action=vmls_recordsform&uid=0' ) . $ajaxurl0;
2986                                                    $htmlCode .= '<script>
2987                                                    jQuery(document).ready(function(){
2988                                                            loadRecordsForm("Restoring Submitted Records Form", "' . $ajaxurl0 . '");
2989                                                    });
2990                                                    jQuery("#fillButton").hide();
2991                                                    jQuery("#typesSelect").hide();
2992                                                    </script>';
2993                                            }
2994                                    */
2995
2996                    //the common account records part, in multi person registration
2997                    $htmlCode .= '<h5 class="ui dividing header"> <i class="group icon"></i> ' . __( 'Common Account Records', 'ppv-live-webcams' ) . '</h4>';
2998
```

```
2999  foreach ( $options['recordFields'] as $field => $parameters ) if (!$options['multiPerson'] ||
       !isset($parameters['personal']) || !$parameters['personal']) {
3000
3001                          $fieldName = sanitize_title( trim( $field ) );
3002
3003                          // if ($parameters['instructions']) $htmlInstructions = ' data-tooltip="' .
      htmlspecialchars(stripslashes($parameters['instructions'])). '"'; else
3004                          $htmlInstructions = '';
3005
3006                          $htmlIcons = '';
3007                          if ($parameters['required'] ?? false) $htmlIcons .= ' <div class="ui button tiny
      basic compact icon" data-tooltip="' . __( 'Required', 'ppv-live-webcams' ) . '"><i class="warning circle
      icon"></i></div>';
3008
3009                          $htmlCode .= '<div class="field"' . $htmlInstructions . '><label
      for="' . $fieldName . '">' . $field . $htmlIcons . '</label>' ;
3010
3011                          $fieldValue = sanitize_textarea_field( $_POST[ $fieldName ] ?? '');
3012
3013                          switch ( $parameters['type'] ) {
3014                                  case 'file':
3015                                          $htmlCode .= '<INPUT class="ui button" type="file"
      name="' . $fieldName . '" id="' . $fieldName . '">';
3016                                          break;
3017
3018                                  case 'text':
3019                                          $htmlCode .= '<INPUT type="text" size="72"
      name="' . $fieldName . '" id="' . $fieldName . '" value="' . $fieldValue . '">';
3020                                          break;
3021
3022                                  case 'textarea':
3023                                          $htmlCode .= '<TEXTAREA type="text" rows="3" cols="70"
      name="' . $fieldName . '" id="' . $fieldName . '">' . $fieldValue . '</TEXTAREA>';
3024                                          break;
3025
3026                                  case 'select':
3027                                          $htmlCode    .= '<SELECT name="' . $fieldName . '"
      id="' . $fieldName . '" class="ui dropdown v-select">';
3028  $fieldOptions = explode( '/', stripslashes( $parameters['options'] ) );
3029
3030                                          $htmlCode .= '<OPTION value=""
      ' . ( ! $fieldValue ? 'selected' : '' ) . '> - </OPTION>';
3031
3032                                          foreach ( $fieldOptions as $fieldOption ) {
3033                                                  $htmlCode .= '<OPTION
      value="' . htmlspecialchars( $fieldOption ) . '"
      ' . ( $fieldOption == $fieldValue ? 'selected' : '' ) . '>' . htmlspecialchars( $fieldOption ) . '</OPTIO
      N>';
3034                                          }
3035
3036                                          $htmlCode .= '</SELECT>';
3037                                          break;
3038                          }
3039                          if ( $parameters['instructions'] ?? false ) {
3040                                  $htmlCode .= '<div class="ui basic small label
      pointing">' . htmlspecialchars( stripslashes( $parameters['instructions'] ) ) . '</div>';
3041                          }
3042
3043                          $htmlCode .= '</div>';
3044                  }
3045
3046                                                  //end multi person account records
3047                                          }
```

```
3048                                }
3049                        }
3050
3051                        if ( $options['match'] )
3052                        {
3053                                $htmlCode .= do_shortcode( '[videowhisper_match_form registration="1"]' );
3054                        }
3055
3056
3057
3058                        $htmlCode .= '<div class="ui segment">
3059    <div class="field">
3060      <div class="ui toggle checkbox">
3061        <input type="checkbox" name="terms" ' . ( isset( $terms ) && $terms ? 'checked' : '' ) . ' tabindex="0" class="hidden">
3062        <label>' . __( 'I accept the Terms of Use', 'ppv-live-webcams' ) . ' </label>
3063      </div>
3064          <div class="ui button tiny basic compact icon" data-tooltip="' . __( 'Required', 'ppv-live-webcams' ) . '"><i class="warning circle icon"></i></div>
3065      <a class="ui tiny button" target="terms" href="' . $atts['terms'] . '"> <i class="clipboard check icon"></i> ' . __( 'Review', 'ppv-live-webcams' ) . '</a>
3066    </div>
3067  </div>
3068  <div class="field">
3069  <input type="button" id="submitButton" name="submitButton" onclick="onSubmitClick()" class="ui submit button" value="' . __( 'Register', 'ppv-live-webcams' ) . '" />
3070   </div>
3071  ';
3072
3073                if ( !$options['registrationNoActivation'] || ( $options['registrationNoActivation'] == 'client' && $atts['role'] != $options['roleClient'] ) ) $htmlCode .='<div class="field">
3074   <label><i class="mail icon"></i> ' . __( 'After registration, you will receive a confirmation email with activation link.', 'ppv-live-webcams' ) . ' '.  __( 'Look for the email and check spam folder if necessary, because you will not be able to use account without activation. If you do not receive the email at current provider, register with a different email address.', 'ppv-live-webcams' ) . '</label>
3075   </div>';
3076
3077                        $htmlCode .= '</form> <script>
3078                        jQuery(document).ready(function(){
3079jQuery(".ui.checkbox").checkbox();
3080});
3081                        </script>';
3082                }
3083                return $htmlCode;
3084        }
3085
3086        static function vmls_recordsform()
3087{
3088    $options = self::getOptions();
3089
3090    ob_start(); // Start output buffering
3091
3092    ?>
3093    <!-- Hidden Inputs for Person Types -->
3094    <?php
3095    $persons = [];
3096    $personTypes = explode(',', $options['personTypes']);
3097    foreach ($personTypes as $personType) {
3098        $type = trim($personType);
3099        $typeVar = sanitize_title_with_dashes($type);
3100        $persons[$type] = intval($_GET[$typeVar] ?? 0);
3101        ?>
```

```
3102        <input type="hidden" name="persons_<?php echo esc_attr($typeVar); ?>" id="persons_<?
    php echo esc_attr($typeVar); ?>" value="<?php echo esc_attr($persons[$type]); ?>">
3103        <?php
3104    }
3105
3106    $pi = 1;
3107    $useColumns = $options['registrationColumns'] ?? false;
3108    if ($useColumns) {
3109        echo '<div class="ui two column grid">';
3110    }
3111
3112    // Loop through person types
3113    foreach ($persons as $type => $number) {
3114        if ($number) {
3115            for ($pn = 1; $pn <= $number; $pn++) {
3116                if ($useColumns) {
3117                    echo '<div class="column">';
3118                }
3119
3120                ?>
3121                <h5 class="ui dividing header">
3122                    <i class="user icon"></i> <?php echo esc_html($pi . '. ' . $type . ' ' . $pn . ':
    ' . __('Account Records', 'ppv-live-webcams')); ?>
3123                </h5>
3124                <input type="hidden" name="person_<?php echo esc_attr($pi); ?>" id="person_<?
    php echo esc_attr($pi); ?>" value="<?php echo esc_attr($type); ?>">
3125
3126                <?php
3127                foreach ($options['recordFields'] as $field => $parameters) {
3128                    if (!isset($parameters['personal']) || !$parameters['personal']) {
3129                        continue;
3130                    }
3131
3132                    $fieldName = sanitize_title(trim($field)) . '_' . $pi;
3133                    $fieldValue = sanitize_textarea_field($_POST[$fieldName] ?? '');
3134                    ?>
3135
3136                    <div class="field"<?php echo !empty($parameters['instructions']) ? ' data-
    tooltip="' . esc_attr(stripslashes($parameters['instructions'])) . '"' : ''; ?>>
3137                        <label for="<?php echo esc_attr($fieldName); ?>">
3138                            <?php echo esc_html($field); ?> <?
    php echo ($parameters['required'] ?? false) ? '<div class="ui button tiny basic compact icon" data-
    tooltip="' . esc_attr__('Required', 'ppv-live-webcams')) . '"><i class="warning circle icon"></i>
    </div>' : ''; ?>
3139                        </label>
3140
3141                        <?php
3142                        switch ($parameters['type']) {
3143                            case 'file':
3144                                ?>
3145                                <input class="ui button" type="file" name="<?
    php echo esc_attr($fieldName); ?>" id="<?php echo esc_attr($fieldName); ?>">
3146                                <?php
3147                                break;
3148
3149                            case 'text':
3150                                ?>
3151                                <input type="text" size="72" name="<?php echo esc_attr($fieldName); ?>"
    id="<?php echo esc_attr($fieldName); ?>" value="<?php echo esc_attr($fieldValue); ?>">
3152                                <?php
3153                                break;
3154
3155                            case 'textarea':
3156                                ?>
```

```
3157                                <textarea type="text" rows="3" cols="70" name="<?
    php echo esc_attr($fieldName); ?>" id="<?php echo esc_attr($fieldName); ?>"><?
    php echo esc_textarea($fieldValue); ?></textarea>
3158                                <?php
3159                                break;
3160
3161                            case 'select':
3162                                ?>
3163                                <select name="<?php echo esc_attr($fieldName); ?>" id="<?
    php echo esc_attr($fieldName); ?>" class="ui dropdown v-select">
3164                                    <option value="" <?php selected(!$fieldValue); ?>> - </option>
3165                                    <?php
3166                                    $fieldOptions = explode('/', stripslashes($parameters['options']));
3167                                    foreach ($fieldOptions as $fieldOption) {
3168                                        ?>
3169                                        <option value="<?php echo esc_attr($fieldOption); ?>" <?
    php selected($fieldOption, $fieldValue); ?>><?php echo esc_html($fieldOption); ?></option>
3170                                        <?php
3171                                    }
3172                                    ?>
3173                                </select>
3174                                <?php
3175                                break;
3176                        }
3177
3178                        if (!empty($parameters['instructions'])) {
3179                            ?>
3180                            <div class="ui basic small label pointing"><?
    php echo esc_html(stripslashes($parameters['instructions'])); ?></div>
3181                            <?php
3182                        }
3183                        ?>
3184                    </div>
3185                    <?php
3186                }
3187
3188                if ($useColumns) {
3189                    echo '</div>'; // Close column div
3190                }
3191                $pi++;
3192            }
3193        }
3194    }
3195
3196    if ($useColumns) {
3197        echo '</div>'; // Close grid div
3198    }
3199
3200    ?>
3201    <input type="hidden" name="persons" id="persons" value="<?php echo esc_attr($pi - 1); ?>">
3202    <?php
3203
3204    exit();
3205 }
3206
3207
3208        static function videowhisper_register_activate( $atts ) {
3209
3210                $options = get_option( 'VWliveWebcamsOptions' );
3211
3212                // shortocode attributes
3213                $atts = shortcode_atts(
3214                        array(),
```

```
3215                        $atts,
3216                        'videowhisper_register_activate'
3217                );
3218                $atts = array_map('esc_attr', $atts);
3219
3220                self::enqueueUI();
3221
3222                if ( ! filter_input( INPUT_GET, 'key' ) ) {
3223                        return '<div class="ui ' . $options['interfaceClass'] . ' segment">The key parameter is missing. Make sure your activation link contains a part like key=[code].</div>';
3224                }
3225
3226
$user_id = filter_input( INPUT_GET, 'user', FILTER_VALIDATE_INT, array( 'options' => array( 'min_range' => 1 ) ) );
3227
3228                $htmlCode = '';
3229
3230                if ( $user_id ) {
3231
3232                        $user = get_userdata( $user_id );
3233                        if (!$user) return '<div class="ui ' . $options['interfaceClass'] . ' segment">User not found.</div>';
3234
3235                        // get user meta activation hash field
3236                        $code = get_user_meta( $user_id, 'videowhisper_activate', true );
3237                        if ( $code == filter_input( INPUT_GET, 'key' ) ) {
3238                                delete_user_meta( $user_id, 'videowhisper_activate' ); // no activation required
3239                                $htmlCode .= '<div class="ui ' . $options['interfaceClass'] . ' segment">' . __( 'Your account was activated.', 'ppv-live-webcams' ) . '</div>';
3240
3241                                $htmlCode .= do_shortcode( '[videowhisper_login_form register="0"]' );
3242
3243                                //pending approval message
3244                                if ( in_array( $options['rolePerformer'], $user->roles )  || in_array( $options['roleStudio'], $user->roles )  ) {
3245                                        if ($options['pendingMessage'] ?? false) $htmlCode .= '<div class="ui ' . $options['interfaceClass'] . ' segment">' . wp_kses_post( $options['pendingMessage'] ) . '</div>';
3246
3247                                        $email = $user->user_email ?? '';
3248                                        $email = filter_var( $email, FILTER_VALIDATE_EMAIL );
3249
if ($options['pendingSubject'] ?? false) if ($email) wp_mail( $email, $options['pendingSubject'], $options['pendingText'] . "\r\n" . wp_login_url() );
3250                                }
3251
3252                        } elseif ( ! $code ) {
3253                                $htmlCode .= '<div class="ui ' . $options['interfaceClass'] . ' segment">No activation code is required. This account is already activated.</div>';
3254                        } else {
3255                                $htmlCode .= '<div class="ui ' . $options['interfaceClass'] . ' segment">Wrong activation code. Make sure you copy and paste link in full without extra spaces or encoding!</div>';
3256                        }
3257                } else {
3258                        $htmlCode .= '<div class="ui ' . $options['interfaceClass'] . ' segment">User parameter missing. Make sure your activation link contains a part like &user=[number].</div>';
3259                }
3260
3261                return $htmlCode;
3262        }
3263
3264
```

```
3265
3266        static function wp_authenticate_user( $user, $password ) {
3267                $requires_activation = get_user_meta( $user->ID, 'videowhisper_activate' );
3268                if ( $requires_activation ) {
3269                        return new \WP_Error( 'activation_failed', __( '<strong>ERROR</strong>: User is not activated. Use activation link from registration email!', 'ppv-live-webcams' ) );
3270                }
3271                return $user;
3272        }
3273
3274
3275        // DASHBOARD
3276
3277        // ! Account Status and Records
3278        static function videowhisper_account_records( $atts ) {
3279
3280                $options = self::getOptions();
3281
3282                if ( ! is_user_logged_in() ) {
3283                        return '<div class="ui ' . $options['interfaceClass'] . ' header"> <i class="sign in icon"></i> ' . __( 'Login Required', 'ppv-live-webcams' ) . '</div>' . do_shortcode( '[videowhisper_login_form register="1"]' );
3284                }
3285
3286                $current_user = wp_get_current_user();
3287                $this_page    = self::getCurrentURL();
3288
3289                // ! update account records
3290
3291                $updateRecords = sanitize_text_field( $_GET['updateRecords'] ?? '' );
3292
3293                $verified       = get_user_meta( $current_user->ID, 'vwVerified', true );
3294                $adminSuspended = get_user_meta( $current_user->ID, 'vwSuspended', true );
3295
3296                $htmlCode = '';
3297
3298                $htmlCode .= '<div id="performerStatus" class="ui ' . ( $options['interfaceClass'] ?? '' ) . ' segment"><h4 class="ui header"> <i class="user icon"></i> ' . $current_user->user_login . ' (#' . $current_user->ID . ')</h4>';
3299
3300                if ( $options['performerStatus'] ) {
3301                        $htmlCode .= ' <i class="address book icon"></i> ' . __( 'Account Status', 'ppv-live-webcams' ) . ': ' . ( $verified == '1' ? ( '<i class="check icon"></i>' . __( 'Verified', 'ppv-live-webcams' ) ) : ($verified == '2' ? ( '<i class="x icon"></i>' . __( 'Rejected', 'ppv-live-webcams' ) ) : ( '<i class="x icon"></i>' . __( 'Not Verified', 'ppv-live-webcams' ) ) ) ) . ( $adminSuspended ? ' ' . __( 'Suspended by Admin', 'ppv-live-webcams' ) : '' );
3302
3303                        if ($verified == '2') $htmlCode .= '<div class="ui message red">' . wpautop( get_user_meta( $current_user->ID, 'vwVerifiedReason', true ) ) . '</div>';
3304
3305                        if ( $updateRecords != 'form' ) {
3306                                $htmlCode .= ' <small><a class="ui compact tiny button" href="' . esc_url( add_query_arg( array( 'updateRecords' => 'form' ), $this_page ) ) . '"> <i class="edit icon"></i>' . __( 'Update Account Records', 'ppv-live-webcams' ) . '</a></small>';
3307                        }
3308                };
3309
3310                if ( $options['performerWallet'] ) {
3311                        $balance = self::balance( $current_user->ID );
3312                        if ( $balance ) {
3313                                $htmlCode .= '<p> <i class="money bill icon"></i> ' . __( 'Your active balance', 'ppv-live-webcams' ) . ': ' . $balance . self::balances( $current_user->ID ) . '</p>';
3314                        }
3315                }
```

```
3316
3317                if ( $options['loginFrontend'] ) {
3318                        $htmlCode .= '<a class="ui tiny button"
href="' . wp_nonce_url( add_query_arg( 'logout', $current_user-
>ID, get_permalink( $options['p_videowhisper_login'] ) ), 'logout' . $current_user->ID, 'verify' ) . '">
<i class="sign out icon"></i>' . __( 'Log Out', 'ppv-live-webcams' ) . '</a> ';
3319                }
3320
3321                $htmlCode .= '</div>';
3322
3323                $error = '';
3324
3325                switch ( $updateRecords ) {
3326
3327                        case 'form':
3328                                $htmlCode .= '<div class="ui ' . $options['interfaceClass'] . ' form
segment"><form method="post" enctype="multipart/form-data"
action="' . add_query_arg( array( 'updateRecords' => 'update' ), $this_page ) . '" name="accountForm"
class="w-actionbox">';
3329
3330                                $htmlCode .= '<H4 class="ui ' . $options['interfaceClass'] . '
header">' . __( 'Account Administrative Records', 'ppv-live-webcams' ) . ': ' . $current_user-
>user_login . '</H4> ' . __( 'These details are required by administrators for account approval and
administrative operations.', 'ppv-live-webcams' );
3331
3332                                if ( ! is_array( $options['recordFields'] ) ) {
3333                                        $htmlCode .= '<p>' . __( 'No record fields defined by
administrator!', 'ppv-live-webcams' ) . '</p>';
3334                                } else {
3335
3336                                        $persons = 1;
3337                                        if ($options['multiPerson'])
3338                                        {
3339                                                $persons = get_user_meta( $current_user-
>ID, 'persons', $persons );
3340                                                if (!$persons) $persons = 1;
3341
3342                                                //also check wp nonce from url to add new person
3343
if ( isset($_GET['person_add']) && $_GET['person_add']=='1' )
3344
if (wp_verify_nonce( $_GET['verify'] ?? '', 'person' . $current_user->ID ))
3345                                                {
3346                                                        $persons++;
3347                                                        update_user_meta( $current_user-
>ID, 'persons', $persons );
3348                                                }else $htmlCode .= 'Add person nonce error!';
3349
3350
if ( isset($_GET['person_remove']) && $_GET['person_remove']=='last' )
3351
if (wp_verify_nonce( $_GET['verify'] ?? '', 'person' . $current_user->ID ))
3352                                                {
3353                                                        $persons--;
3354                                                        update_user_meta( $current_user-
>ID, 'persons', $persons );
3355                                                }else $htmlCode .= 'Add person nonce error!';
3356
3357                                                //insert link to add/remove person
3358                                                $htmlCode .= '<div class="ui message">';
3359                                                $htmlCode .=  __( 'Persons', 'ppv-live-webcams' ) . ':
<div class="ui label">' . $persons . '</div>';
3360                                                $htmlCode .= '<a class="ui button tiny" href="' .
wp_nonce_url( add_query_arg( array( 'updateRecords' => 'form', 'person_add' => '1' ), $this_page ) , 'per
son' . $current_user->ID, 'verify' ). '"><i class="plus icon"></i>' . __( 'Add New Person', 'ppv-live-
webcams' ) . '</a>';
3361                                                $htmlCode .= '</div>';
```

```
3362
3363                                                    for ($i = 1; $i <= $persons; $i++)
3364                                                    {
3365                                                            $htmlCode .=  '<h4 class="ui dividing
header">' . __( 'Personal Records', 'ppv-live-webcams' ) . ' ' . $i . '</h4>';
3366
3367                                                            $type = get_user_meta( $current_user-
>ID, 'person_' . $i, true );
3368
3369                                                            //display a select to change type
3370                                                            $htmlCode .= '<div class="field">
<label>' . __( 'Type', 'ppv-live-webcams' ) . '</label>';
3371                                                            $htmlCode .= '<SELECT
name="' . 'person_' . $i . '" id="' . 'person_' . $i . '" class="ui dropdown v-select">';
3372
$personTypes = explode( ',', $options['personTypes'] );
3373                                                            foreach ($personTypes as $personType)
3374                                                                    $htmlCode .= '<OPTION
value="' . esc_attr( trim($personType) ) . '" ' . ( $type ==
trim($personType) ? 'selected' : '' ) . '>' . htmlspecialchars( trim($personType) ) . '</OPTION>';
3375                                                            $htmlCode .= '</SELECT></div>';
3376
3377
foreach ( $options['recordFields'] as $field => $parameters )
3378
if ( isset($parameters['personal']) && $parameters['personal'] )
3379                                                            {
3380
3381
$fieldName = sanitize_title( trim( $field ) ) . '_' . $i; //per person
3382
3383                                                            // if ($parameters['instructions'])
$htmlInstructions = ' data-tooltip="' . htmlspecialchars(stripslashes($parameters['instructions'])). '"';
else
3384                                                            $htmlInstructions = '';
3385
3386                                                            $htmlIcons = '';
3387
if ($parameters['required'] ?? false) $htmlIcons .= ' <div class="ui button tiny basic compact icon"
data-tooltip="' . __( 'Required', 'ppv-live-webcams' ) . '"><i class="warning circle icon"></i></div>';
3388
3389                                                            $htmlCode .= '<div
class="field"' . $htmlInstructions . '><label
for="' . $fieldName . '">' . $field . $htmlIcons . '</label>';
3390
3391                                                            if ( isset( $_POST[ $fieldName ] ) ) {
3392                                                                    if ( $current_user-
>ID == $_POST['profileFor'] ) {
3393
$fieldValue = sanitize_text_field( $_POST[ $fieldName ] );
3394
update_user_meta( $current_user->ID, 'vwf_' . $fieldName, $fieldValue );
3395                                                                    }
3396                                                            }
3397
3398
$fieldValue = htmlspecialchars( get_user_meta( $current_user->ID, 'vwf_' . $fieldName, true ) );
3399
if ($parameters['required'] ?? false) if (!$fieldValue) $error .=  esc_html( $field ) . ':
' . __( 'Required field is empty!', 'ppv-live-webcams' ) . ' ' . $i . '/' . $persons . '<br>';
3400
3401                                                            switch ( $parameters['type'] ) {
3402                                                                    case 'file':
3403                                                                            $htmlCode .= '<INPUT
class="ui button" type="file" name="' . $fieldName . '" id="' . $fieldName . '">';
3404                                                                            $filePath
= get_user_meta( $current_user->ID, 'vwf_' . $fieldName, true );
```

```
3405  if ( file_exists( $filePath ) ) {
3406                                                                  $htmlCode .= '<a target="_download" href="' . self::path2url( $filePath ) . '">' . __( 'Available', 'ppv-live-webcams' ) . '</a> ' . self::humanSize( filesize( $filePath ) ) . '
' . pathinfo( $filePath, PATHINFO_EXTENSION ) . '<br>';
3407                                                                  }
3408                                                                  break;
3409
3410                                                          case 'text':
3411                                                                  $htmlCode .= '<INPUT type="text" size="72" name="' . $fieldName . '" id="' . $fieldName . '" value="' . $fieldValue . '">';
3412                                                                  break;
3413
3414                                                          case 'textarea':
3415                                                                  $htmlCode .= '<TEXTAREA type="text" rows="3" cols="70" name="' . $fieldName . '"
id="' . $fieldName . '">' . $fieldValue . '</TEXTAREA>';
3416                                                                  break;
3417
3418                                                          case 'select':
3419                                                                  $htmlCode    .= '<SELECT name="' . $fieldName . '" id="' . $fieldName . '" class="ui dropdown v-select">';
3420  $fieldOptions = explode( '/', stripslashes( $parameters['options'] ) );
3421
3422                                                                  $htmlCode .= '<OPTION value="" ' . ( ! $fieldValue ? 'selected' : '' ) . '> - </OPTION>';
3423
3424  foreach ( $fieldOptions as $fieldOption ) {
3425  $htmlCode .= '<OPTION value="' . htmlspecialchars( $fieldOption ) . '"
' . ( $fieldOption == $fieldValue ? 'selected' : '' ) . '>' . htmlspecialchars( $fieldOption ) . '</OPTION>';
3426                                                                  }
3427
3428                                                                  $htmlCode .= '</SELECT>';
3429                                                                  break;
3430                                                  }
3431  $htmlCode .= '<small>' . htmlspecialchars( stripslashes( $parameters['instructions'] ?? '' ) ) . '</small>';
3432
3433                                                  $htmlCode .= '</div>';
3434
3435                                          }
3436
3437                                  }
3438
3439                                  $htmlCode .= '<br><a class="ui button tiny" href="' . wp_nonce_url( add_query_arg( array( 'updateRecords' => 'form', 'person_remove' => 'last' ), $this_page ), 'person' . $current_user->ID, 'verify' ). '"><i class="minus icon"></i>' . __( 'Remove Last Person', 'ppv-live-webcams' ) . '</a>';
3440                          }
3441
3442                          $htmlCode .=  '<h4 class="ui header dividing">' . __( 'Account Records', 'ppv-live-webcams' ) . '</h4>';
3443
3444                          foreach ( $options['recordFields'] as $field => $parameters )
3445                          if ( !$options['multiPerson'] || !isset($parameters['personal']) || !$parameters['personal'] )
3446                          {
3447
3448                                  $fieldName = sanitize_title( trim( $field ) );
3449
```

```
Line
3450                                                    // if ($parameters['instructions']) $htmlInstructions = '
data-tooltip="' . htmlspecialchars(stripslashes($parameters['instructions'])). '"'; else
3451                                                    $htmlInstructions = '';
3452
3453                                                    $htmlIcons = '';
3454                                                    if ($parameters['required'] ?? false) $htmlIcons .= '
<div class="ui button tiny basic compact icon" data-tooltip="' . __( 'Required', 'ppv-live-
webcams' ) . '"><i class="warning circle icon"></i></div>';
3455
3456                                                    $htmlCode .= '<div
class="field"' . $htmlInstructions . '><label
for="' . $fieldName . '">' . $field . $htmlIcons . '</label>';
3457
3458                                                    if ( isset( $_POST[ $fieldName ] ) ) {
3459                                                            if ( $current_user-
>ID == $_POST['profileFor'] ) {
3460
$fieldValue = sanitize_text_field( $_POST[ $fieldName ] );
3461                                                                    update_user_meta( $current_user-
>ID, 'vwf_' . $fieldName, $fieldValue );
3462                                                            }
3463                                                    }
3464
3465                                                    $fieldMeta = get_user_meta( $current_user-
>ID, 'vwf_' . $fieldName, true );
3466                                                    if (is_array($fieldMeta))
3467                                                    {
3468                                                            $error .= esc_html( $field ) . ': ' . __( 'Array
field is not supported!', 'ppv-live-webcams' ) . '<br>';
3469                                                            $fieldMeta = implode('|', $fieldMeta);
3470                                                    }
3471
3472                                                    $fieldValue = htmlspecialchars( $fieldMeta );
3473
if ($parameters['required'] ?? false) if (!$fieldValue) $error .=  esc_html( $field ) . ':
' . __( 'Required field is empty!', 'ppv-live-webcams' ) . '<br>';
3474
3475                                                    switch ( $parameters['type'] ) {
3476                                                            case 'file':
3477                                                                    $htmlCode .= '<INPUT class="ui button"
type="file" name="' . $fieldName . '" id="' . $fieldName . '">';
3478                                                                    $filePath
= get_user_meta( $current_user->ID, 'vwf_' . $fieldName, true );
3479                                                                    if ( file_exists( $filePath ) ) {
3480                                                                            $htmlCode .= '<a
target="_download" href="' . self::path2url( $filePath ) . '">' . __( 'Available', 'ppv-live-
webcams' ) . '</a> ' . self::humanSize( filesize( $filePath ) ) . '
' . pathinfo( $filePath, PATHINFO_EXTENSION ) . '<br>';
3481                                                                    }
3482                                                                    break;
3483
3484                                                            case 'text':
3485                                                                    $htmlCode .= '<INPUT type="text"
size="72" name="' . $fieldName . '" id="' . $fieldName . '" value="' . $fieldValue . '">';
3486                                                                    break;
3487
3488                                                            case 'textarea':
3489                                                                    $htmlCode .= '<TEXTAREA type="text"
rows="3" cols="70" name="' . $fieldName . '" id="' . $fieldName . '">' . $fieldValue . '</TEXTAREA>';
3490                                                                    break;
3491
3492                                                            case 'select':
3493                                                                    $htmlCode    .= '<SELECT
name="' . $fieldName . '" id="' . $fieldName . '" class="ui dropdown v-select">';
3494
$fieldOptions = explode( '/', stripslashes( $parameters['options'] ) );
3495
```

```
3496                                                    $htmlCode .= '<OPTION value=""
' . ( ! $fieldValue ? 'selected' : '' ) . '> - </OPTION>';
3497
3498
foreach ( $fieldOptions as $fieldOption ) {
3499                                                            $htmlCode .= '<OPTION
value="' . htmlspecialchars( $fieldOption ) . '"
' . ( $fieldOption == $fieldValue ? 'selected' : '' ) . '>' . htmlspecialchars( $fieldOption ) . '</OPTIO
N>';
3500                                                    }
3501
3502                                                    $htmlCode .= '</SELECT>';
3503                                                    break;
3504                                            }
3505
$htmlCode .= '<small>' . htmlspecialchars( stripslashes( $parameters['instructions'] ?? '' ) ) . '</small
>';
3506
3507                                            $htmlCode .= '</div>';
3508
3509                                    }
3510                                }
3511                                $htmlCode .= '<input type="hidden" name="profileFor" id="profileFor"
value="' . $current_user->ID . '">
3512                        <BR><input class="ui primary button" type="submit" name="updateRecords" id="updateRecords"
value="' . __( 'Save', 'ppv-live-webcams' ) . '" />
3513                        </form></div>';
3514
3515                                break;
3516
3517                        case 'update':
3518                                // var_dump($_POST);
3519
3520                                $persons = 1;
3521                                if ($options['multiPerson'])
3522                                {
3523                                        $persons = get_user_meta( $current_user-
>ID, 'persons', $persons );
3524                                        if (!$persons) $persons = 1;
3525
3526                                        //update person type
3527                                        for ($i = 1; $i <= $persons; $i++)
3528                                        {
3529
$type = sanitize_text_field( $_POST[ 'person_' . $i ] ?? '' );
3530                                                update_user_meta( $current_user-
>ID, 'person_' . $i, $type );
3531
3532                                        }
3533
3534                                }
3535
3536                                if ( is_array( $options['recordFields'] ) ) {
3537                                        if ( $current_user->ID == $_POST['profileFor'] ) {
3538                                                $htmlCode .= '<div class="ui
' . $options['interfaceClass'] . ' form segment">';
3539
3540                                                // var_dump($options['recordFields']);
3541                                                // var_dump($_FILES);
3542
3543
foreach ( $options['recordFields'] as $field => $parameters )
3544                                                {
3545
```

```
3546  if ($options['multiPerson']) $piterations = $persons;
3547                                                  else $piterations = 1;
3548
3549                                                  for ($pi = 1; $pi <= $piterations; $pi++)
3550                                                  {
3551
3552  if ($options['multiPerson'] && isset($parameters['personal']) && $parameters['personal']) $fieldName = sa
      nitize_title( trim( $field ) ) . '_' . $pi;
3553  else $fieldName = sanitize_title( trim( $field ) );
3554
3555                                                  switch ( $parameters['type'] ) {
3556
3557                                                          case 'file':
3558                                                                  $uploadComplete = false;
3559                                                                  $uploadExists = false;
3560
3561  if ( $filename = $_FILES[ $fieldName ]['tmp_name'] ) {
3562                                                                                  $deny = false;
3563
3564                                                                                  $ext
       = strtolower( pathinfo( $_FILES[ $fieldName ]['name'], PATHINFO_EXTENSION ) );
3565  $allowed = array( 'jpg', 'jpeg', 'png', 'gif', 'pdf' );
3566
3567  if ( ! in_array( $ext, $allowed ) ) {
3568  $htmlCode .= '<br>' . esc_html( $field ) . ' - ' . esc_html( $filename ) . ': Unsupported file extension!
      Please use one of these formats: ' . implode( ', ', $allowed );
3569                                                                                          $deny
      = true;
3570                                                                                  }
3571
3572  $maxUpload = intval ($options['maxUpload'] ?? 6000) * 1000;
3573  if ( $_FILES[ $fieldName ]['size'] > $maxUpload ) {
3574  $htmlCode .= '<br>' . esc_html( $field ) . ': File is too big! Please use a file under
      ' . $maxUpload . 'kb.';
3575                                                                                          $deny
      = true;
3576                                                                                  }
3577
3578                                                                                  if ( ! $deny ) {
3579  $destination = $options['uploadsPath'] . '/user_' .  $current_user->user_nicename . '_' .  $current_user-
      >ID  . '/';
3580  if ( ! file_exists( $destination ) ) {
3581  mkdir( $destination );
3582                                                                                          }
3583
3584                                                                                          $newfilename =
      $fieldName . '_' . time() . '.' . $ext ;
3585                                                                                          $newpath
       = $destination . $newfilename;
3586
3587  $errorUp = self::handle_upload( $_FILES[ $fieldName ], $newpath ); // handle trough wp_handle_upload()
3588                                                                                          if ( $errorUp ) {
3589  $htmlCode .= '<br>' . 'Error uploading ' . esc_html( $filename . ':' . $errorUp );
```

```
3590                                                                                    }
3591
3592 if ( ! file_exists( $newpath ) ) {
3593 $htmlCode .= '<br>' . esc_html( $field ) . ': Error uploading ' . esc_html( $newpath );
3594                                                                                    } else {
3595 $oldpath = get_user_meta( $current_user->ID, 'vwf_' . $fieldName, true );
3596 if ( $options['recordClean'] ) {
3597 if ( file_exists( $oldpath ) ) {
3598       unlink( $oldpath );
3599                                                                                                  }
3600                                                                                        }
3601
3602 update_user_meta( $current_user->ID, 'vwf_' . $fieldName, $newpath );
3603 $uploadComplete = true;
3604                                                                                    }
3605
3606                                                                                }
3607                                                                            }
3608
3609 if (!$uploadComplete) if (get_user_meta($current_user-
>ID, 'vwf_' . $fieldName, true)) $uploadExists = true;
3610
3611 if ($parameters['required'] ?? false) if (!$uploadComplete && !$uploadExists) $error .= esc_html( $field
) . ': ' . __( 'Required file missing!', 'ppv-live-webcams' ). ' ' . $pi . '/' . $persons . '<br>' ;
3612                                                                                break;
3613
3614                                                                        default:
3615 if ( isset( $_POST[ $fieldName ] ) ) {
3616                                                                                $fieldValue =
sanitize_textarea_field( $_POST[ $fieldName ] );
3617 update_user_meta( $current_user->ID, 'vwf_' . $fieldName, $fieldValue );
3618                                                                                // $htmlCode .= '<br>' .
$field . ': '.  sanitize_text_field($_POST[$fieldName]);
3619
3620 if ($parameters['required'] ?? false) if (!$fieldValue) $error .=  esc_html( $field ) . ':
' . __( 'Required field missing!', 'ppv-live-webcams' ) . ' ' . $pi . '/' . $persons . '<br>' ;
3621                                                                                }
3622                                                                        } //end type
3623
3624                                                                     } //end iterations
3625
3626                                                                    } // end field
3627
3628                                                                    update_user_meta( $current_user-
>ID, 'vwUpdated', time() );
3629                                                                    self::usersPendingApproval();
3630
3631                                                                    $htmlCode .= '<br>' . __( 'Updated administrative account
records.', 'ppv-live-webcams' ) . '</div>';
3632
3633                                                                    $verified       = get_user_meta( $current_user-
>ID, 'vwVerified', true );
3634
```

```
3635  if (!$error) if ($options['pendingMessage'] ?? false) $htmlCode .= '<p>' . wp_kses_post( $options['pendingMessage'] ) . '</p>';
3636
3637                                                  // end records
3638
3639                                          } else {
3640                                                  $htmlCode .= 'Incorrect profileFor.';
3641                                          }
3642                                  }
3643                                  break;
3644                  }
3645
3646                  if ($error)
3647                  {
3648                    $htmlCode .= '<div class="ui warning message">
3649          <div class="header">' . __( 'Administrative Records Incomplete', 'ppv-live-webcams' ) . ':</div>
3650          <ul class="list">
3651            ' . $error . '
3652          </ul><p>' . __( 'Account can not be verified without required records.', 'ppv-live-webcams' ) . '</p></div>';
3653
3654                   //unverify user if required info is missing
3655                   update_user_meta( $current_user->ID, 'vwVerified', 0 );
3656                  }
3657
3658
3659
3660
3661                  return $htmlCode;
3662          }
3663
3664
3665          // start studio dashboard
3666
3667          // ! Studio Dashboard
3668          static function videowhisper_webcams_studio( $atts ) {
3669                  $options = get_option( 'VWliveWebcamsOptions' );
3670
3671                  self::enqueueUI();
3672
3673                  if ( ! is_user_logged_in() ) {
3674                          return '<div class="ui ' . $options['interfaceClass'] . ' header"> <i class="sign in icon"></i> ' . __( 'Login to manage studio!', 'ppv-live-webcams' ) . '</div>' . do_shortcode( '[videowhisper_login_form register="1" role="' . $options['roleStudio'] . '"]' );
3675                  }
3676
3677                  // if (!is_user_logged_in()) return __('Login to manage studio!','ppv-live-webcams') . '<BR><a class="ui button primary qbutton" href="' . wp_login_url() . '">' . __('Login', 'ppv-live-webcams') . '</a>  <a class="ui button secondary qbutton" href="' . ( ($options['registerFrontend'] && $options['p_videowhisper_register_studio']) ? get_permalink($options['p_videowhisper_register_studio']) : wp_registration_url() ) . '">' . __('Register', 'ppv-live-webcams') . '</a>';
3678
3679                  // shortocode attributes
3680                  $atts = shortcode_atts(
3681                          array(
3682                                  'include_css' => '1',
3683                          ),
3684                          $atts,
3685                          'videowhisper_webcams_studio'
3686                  );
3687                  $atts = array_map('esc_attr', $atts);
3688
```

```
3689                $current_user = wp_get_current_user();
3690                $htmlCode = '';
3691                $performersList = 0;
3692
3693                if ( ! $current_user->ID ) {
3694                        return __( 'Login is required to access this section!', 'ppv-live-webcams' );
3695                }
3696
3697                // access keys
3698                $userkeys   = $current_user->roles;
3699                $userkeys[] = $current_user->user_login;
3700                $userkeys[] = $current_user->user_nicename;
3701                $userkeys[] = $current_user->ID;
3702                $userkeys[] = $current_user->user_email;
3703
3704                $roleS = implode( ',', $current_user->roles );
3705                if ( ! self::any_in_array( array( $options['roleStudio'], 'administrator', 'super
admin' ), $current_user->roles ) ) {
3706                        return __( 'User role does not allow managing studio!', 'ppv-live-webcams' ) . '
(' . $roleS . ')';
3707                }
3708
3709                $this_page = self::getCurrentURL();
3710                $tab       = sanitize_text_field( $_GET['tab'] ?? '' );
3711
3712                if ( ! $tab ) {
3713
$htmlCode .= html_entity_decode( stripslashes( $options['dashboardMessageStudio'] ) );
3714                }
3715
3716                $htmlCode .= do_shortcode( '[videowhisper_account_records]' );
3717                if ( ! self::userEnabled( $current_user, $options, 'Studio' ) ) {
3718                        return $htmlCode . __( 'Your account is not currently enabled. Update your
account records and wait for site admins to approve your account.', 'ppv-live-webcams' );
3719                }
3720
3721                // updating records
3722                if ( $_GET['updateRecords'] ?? false ) {
3723                        if ( $_GET['updateRecords'] != 'update' ) {
3724                                return $htmlCode;
3725                        }
3726                }
3727
3728                        // ! Dashboard Tabs
3729                        /*
3730                          $htmlCode .= '<div class="vwtabs">';
3731
3732                        //! performers tab
3733                        $checked = '';
3734                        if ( $tab == 'performers' || ! $tab) $checked = 'checked';
3735                        if ($checked) $checked1 = true;
3736
3737                        $htmlCode .= '<div class="vwtab">
3738                <input type="radio" id="tab-performers" name="tab-group-1" '.$checked.'>
3739                <label class="vwlabel" for="tab-performers">' . __('Performers', 'ppv-live-webcams') .
'</label>
3740
3741                <div class="vwpanel">
3742                <div class="vwcontent">';
3743                        $htmlCode .= '<h3 class="ui header">' . __('Performers', 'ppv-live-webcams') .
'</h3> ' . __('Manage studio performer accounts.', 'ppv-live-webcams') ;
3744                */
3745                        $active = '';
```

```
3746                $header     = array();
3747                $content    = array();
3748
3749                if ( $tab == 'performers' || ! $tab ) {
3750                        $active = 'performers';
3751                }
3752
3753                $header['performers'] = __( 'Performers', 'ppv-live-webcams' );
3754
3755                $contents = '<h3 class="ui header">' . __( 'Performers', 'ppv-live-webcams' ) . '</h3>
' . __( 'Studio manages multiple performer accounts.', 'ppv-live-webcams' );
3756
3757                if ( $tab == 'performers' ) {
3758
3759                        if ( $collectID = intval( $_GET['collect'] ?? 0 ) ) {
3760                                $studioID = get_user_meta( $collectID, 'studioID', true );
3761
3762                                if ( $studioID == $current_user->ID || $collectID == $current_user-
>ID ) {
3763                                        $balance = self::balance( $collectID, false, $options );
3764
3765                                        if ( $balance > 0 ) {
3766
3767                                                $performer = get_userdata( $collectID );
3768                                                $studio    = get_userdata( $studioID );
3769
3770                                                $collectLast
= floatval( get_user_meta( $collectID, 'studioCollectLast', true ) );
3771                                                $collectTotal = floatval( get_user_meta( $collectID, 'studioCollectTotal', true ) );
3772
3773
self::transaction( 'collect_performer', $collectID, -$balance, 'Studio ' . $studio-
>user_login . 'collected balance.', null, null, $options );
3774
self::transaction( 'collect_studio', $studioID, $balance, 'Collected balance from ' . $performer-
>user_login . '.', null, null, $options );
3775
3776
update_user_meta( $collectID, 'studioCollectLast', $balance );
3777
update_user_meta( $collectID, 'studioCollectTotal', $balance + $collectTotal );
3778
3779                                                $contents .= '<div class="ui segment">' . __( 'Collected
user ballance.', 'ppv-live-webcams' ) . '<br>' . $balance . ' : ' . $performer->user_login . ' -
' . $studio->user_login . '<div>';
3780                                        } else {
3781                                                $contents .= '<BR>' . __( 'No balance to collect!', 'ppv-
live-webcams' );
3782                                        }
3783                                } else {
3784                                        $contents .= '<BR>' . __( 'You are not allowed to collect for
this user!', 'ppv-live-webcams' );
3785                                }
3786                        }
3787
3788                        // Transactions Log
3789                        if ( $transactionsID = intval( $_GET['transactions'] ?? 0 ) ) {
3790
3791                                $studioID = get_user_meta( $transactionsID, 'studioID', true );
3792
3793                                if ( $studioID == $current_user->ID || $transactionsID == $current_user-
>ID ) {
3794
3795                                        $user_info = get_userdata( $transactionsID );
```

```
3796
3797                                    if ( $options['wallet'] == 'MyCred' ) {
3798                                            $contents .= '<h4>' . __( 'Performer Transactions', 'ppv-live-webcams' ) . ': ' . $user_info->user_login . '</h4>';
3799                                            if ( shortcode_exists( 'mycred_history' ) ) {
3800                                                    $contents .= do_shortcode( '[mycred_history user_id=' . $transactionsID . ']' );
3801                                            }
3802
3803                                            $contents .= '<style type="text/css">
3804 .pagination, .pagination > LI {
3805 display: inline !important;
3806 padding: 5px;
3807 margin: 2px;
3808 background-color: #EEE;
3809 }
3810 </style>';
3811                                    }
3812
3813                                    if ( $options['wallet'] == 'WooWallet' ) {
3814                                            $contents .= '<h4>' . __( 'Transactions', 'ppv-live-webcams' ) . '</h4>';
3815                                            if ( shortcode_exists( 'woo-wallet' ) ) {
3816                                                    $contents .= do_shortcode( '[woo-wallet user_id=' . $transactionsID . ']' );
3817                                            }
3818                                    }
3819                                } else {
3820                                            $contents .= '<BR>' . __( 'You are not allowed to view transactions for this user!', 'ppv-live-webcams' );
3821                                }
3822                        }
3823
3824                        switch ( $_GET['view'] ) {
3825                                case 'form':
3826                                        $contents .= '<h4>' . __( 'Create Performer Account', 'ppv-live-webcams' ) . '</h4>';
3827
3828                                        $action = add_query_arg(
3829                                                array(
3830                                                        'tab'  => 'performers',
3831                                                        'view' => 'insert',
3832                                                ),
3833                                                $this_page
3834                                        );
3835
3836                                        $clarificationsMsg = __( 'After setting up performer account you will also receive password to communicate to performer for quick access. Account password should be changed by performer. Warning: Make sure you fill details correctly. Only administrator can delete incorrect accounts.', 'ppv-live-webcams' );
3837                                        $saveMsg           = __( 'Save', 'ppv-live-webcams' );
3838
3839                                        $contents .= <<<HTMLCODE
3840 <form class="ui form" method="post" enctype="multipart/form-data" action="$action" name="adminForm" class="w-actionbox">
3841 <table class="ui table striped" width="500px">
3842 <tr><td>Username</td><td><input size="32" type="text" name="performer_username" id="performer_username" value=""></td></tr>
3843 <tr><td>Email</td><td><input size="32" type="text" name="performer_email" id="performer_email" value="">
     </td></tr>
3844 <tr><td></td><td><input class="ui button" type="submit" name="save" id="save" value="$saveMsg" /></td>
     </tr>
3845 </table>
3846 $clarificationsMsg
3847 </form>
```

```
HTMLCODE;
                                        break;

                                case 'insert':
                                        $contents .= '<h4>' . __( 'Adding Performer Account', 'ppv-live-webcams' ) . '</h4>';

                                        $error = '';

$performer_username = sanitize_text_field( $_POST['performer_username'] );
                                        $performer_email = sanitize_text_field( $_POST['performer_email'] );

                                        if ( username_exists( $performer_username ) ) {
                                                $error = __( 'Username already in use!', 'ppv-live-webcams' );
                                        }
                                        if ( email_exists( $performer_email ) ) {
                                                $error = __( 'Email already in use!', 'ppv-live-webcams' );
                                        }

                                        $args            = array(
                                                'blog_id'      => $GLOBALS['blog_id'],
                                                'meta_key'     => 'studioID',
                                                'meta_value'   => $current_user->ID,
                                                'meta_compare' => '=',
                                        );
                                        $performers      = get_users( $args );
                                        $performersCount = count( $performers );

if ( $options['studioPerformers'] && $performersCount >= $options['studioPerformers'] ) {
                                                $error = __( 'Performers limit reached!', 'ppv-live-webcams' );
                                        }

                                        if ( $error ) {
                                                $contents .= __( 'Could not create performer account: ', 'ppv-live-webcams' ) . $error;
                                        } else {

$random_password = wp_generate_password( $length = 12, $include_standard_special_chars = false );
                                                $user_id = wp_create_user( $performer_username, $random_password, $performer_email );
                                                if ( $user_id > 0 ) {
                                                        $contents .= __( 'Performer account was created', 'ppv-live-webcams' ) . ': <BR>Username: ' . $performer_username . '<BR>' . __( 'Password', 'ppv-live-webcams' ) . ': ' . $random_password . '<BR>' . __( 'Email', 'ppv-live-webcams' ) . ': ' . $performer_email;

                                                        // assign to studio

update_user_meta( $user_id, 'studioID', $current_user->ID );

update_user_meta( $user_id, 'studioLogin', $current_user->user_login );

update_user_meta( $user_id, 'studioPassword', $random_password );

update_user_meta( $user_id, 'studioDisabled', 0 );

                                                        // create performer role if missing
                                                        $role = $options['rolePerformer'];
                                                        $oRole = get_role( $role );
                                                        if ( ! $oRole ) {
```

```
3896  add_role( $role, ucwords( $role ), array( 'read' => true ) );
3897                                                                  $oRole = get_role( $role );
3898                                                          }
3899
3900                                                          // set as performer
3901                                                          wp_update_user(
3902                                                                  array(
3903                                                                          'ID'   => $user_id,
3904  'role' => $options['rolePerformer'],
3905                                                                  )
3906                                                          );
3907
3908  update_user_meta( $user_id, 'videowhisper_ip_register', self::get_ip_address() ); // registration ip
3909                                                          update_user_meta( $user_id, 'created_by', 'ppv-
      live-webcams/studio' ); // for troubleshooting registration origin
3910  update_user_meta( $user_id, 'videowhisper_role', $options['rolePerformer'] ); // registration role
3911
3912                                                          // also create a webcam listing
3913                                                          $newPerformer = get_userdata( $user_id );
3914                                                          $name
       = self::performerName( $newPerformer, $options );
3915
3916  $webcamID = self::webcamPost( $name, $name, $user_id, $current_user->ID );
3917  update_user_meta( $user_id, 'currentWebcam', $webcamID );
3918
3919                                                  }
3920                                          }
3921
3922                                          break;
3923
3924                                  default:
3925                                          $performersList = 1;
3926                          }
3927
3928                          //disable/enable
3929                          if ( $user_id = intval( $_GET['performer_disable'] ?? 0 ) ) {
3930                                  $studioID = get_user_meta( $user_id, 'studioID', true );
3931
3932                                  if ( $studioID == $current_user->ID ) {
3933                                          update_user_meta( $user_id, 'studioDisabled', 1 );
3934                                  } else {
3935                                          $contents .= __( 'That performer is not assigned to this
      studio!', 'ppv-live-webcams' );
3936                                  }
3937                          }
3938
3939                                  if ( $user_id = intval( $_GET['performer_enable'] ?? 0 ) ) {
3940                                  $studioID = get_user_meta( $user_id, 'studioID', true );
3941
3942                                  if ( $studioID == $current_user->ID ) {
3943                                          delete_user_meta( $user_id, 'studioDisabled' );
3944                                  } else {
3945                                          $contents .= __( 'That performer is not assigned to this
      studio!', 'ppv-live-webcams' );
3946                                  }
3947                          }
3948
3949
```

```
3950
3951                } else {
3952                        $performersList = 1; // no special action
3953                }
3954
3955                if ( $performersList ?? false ) {
3956
3957                        $args = array(
3958                                'blog_id'      => $GLOBALS['blog_id'],
3959                                'meta_key'     => 'studioID',
3960                                'meta_value'   => $current_user->ID,
3961                                'meta_compare' => '=',
3962                                'orderby'      => 'registered',
3963                                'order'        => 'DESC',
3964                                'fields'       => 'all',
3965                        );
3966
3967                        $performers = get_users( $args );
3968
3969                        $performersCount = count( $performers );
3970
3971                        $contents .= '<h4>' . __( 'Performers List', 'ppv-live-webcams' ) . '
(' . $performersCount . '/' . ( $options['studioPerformers'] ? $options['studioPerformers'] : '&infin;' )
 . ')</h4>';
3972
3973                        if ( $performersCount ) {
3974                                $contents .= '<table class="ui table celled striped sortable"><thead><tr>
<th class="sorted ascending">' . __( 'Login', 'ppv-live-webcams' ) . '</th><th>' . __( 'Email', 'ppv-
live-webcams' ) . '</th><th>' . __( 'Balance', 'ppv-live-webcams' ) . ' (' . $options['currency'] . ')
</th><th>' . __( 'Collected', 'ppv-live-webcams' ) . '<br>' . __( 'Last', 'ppv-live-webcams' ) . ' /
' . __( 'Total', 'ppv-live-webcams' ) . '</th><th>' . __( 'Status', 'ppv-live-webcams' ) . '</th>
<th>Info</th><th>' . __( 'Current Room', 'ppv-live-webcams' ) . '</th></tr></thead>';
3975
3976                                foreach ( $performers as $performer ) {
3977
3978                                $userLabel = '<i class="icon user"></i>' . $performer->user_login;
3979
3980                                if ( function_exists( 'bp_members_get_user_url' ) ) $userLabel = '<a
href="' . bp_members_get_user_url( $performer->id ). '"><i class="icon user"></i>' . $performer-
>user_login . '</a>';
3981
3982                                        $contents .= '<tr><td><b>' . $userLabel . '</b> </td><td>
<small>' . $performer->user_email .'</small>';
3983
3984                                        $contents .= '</td><td>';
3985                                        $contents .= '<a class="ui compact label"
href="' . add_query_arg(
3986                                                array(
3987                                                        'tab'          => 'performers',
3988                                                        'transactions' => $performer->ID,
3989                                                ),
3990                                                $this_page
3991                                        ) . '">' . self::balance( $performer->ID ) . '</a> ';
3992                                        $contents .= '<a class="ui compact tiny button"
href="' . add_query_arg(
3993                                                array(
3994                                                        'tab'     => 'performers',
3995                                                        'collect' => $performer->ID,
3996                                                ),
3997                                                $this_page
3998                                        ) . '">' . __( 'Collect', 'ppv-live-webcams' ) . '</a>';
3999                                        $contents .= '</td><td>';
4000
```

```
Line
4001                                    $collectLast  = floatval( get_user_meta( $performer->ID, 'studioCollectLast', true ) );
4002                                    $collectTotal = floatval( get_user_meta( $performer->ID, 'studioCollectTotal', true ) );
4003
4004                                    $contents .= $collectLast . '/' . $collectTotal;
4005                                    $contents .= '</td><td>';
4006
4007                                    $disabled = get_user_meta( $performer->ID, 'studioDisabled', true );
4008
4009                                    if ( ! $disabled ) {
4010                                            $contents .= '<a class="ui compact tiny button" href="' . add_query_arg(
4011                                                    array(
4012                                                            'tab'      => 'performers',
4013                                                            'performer_disable'  => $performer->ID,
4014                                                    ),
4015                                                    $this_page
4016                                            ) . '">' . __( 'Enabled', 'ppv-live-webcams' ) . '</a>';
4017                                    } else {
4018                                            $contents .= '<a class="ui compact tiny button grey" href="' . add_query_arg(
4019                                                    array(
4020                                                            'tab'      => 'performers',
4021                                                            'performer_enable'  => $performer->ID,
4022                                                    ),
4023                                                    $this_page
4024                                            ) . '">' . __( 'Disabled', 'ppv-live-webcams' ) . '</a>';
4025                                    }
4026
4027                                    $contents .= '</td><td> <div class="ui label mini basic">';
4028
4029                                    $password = get_user_meta( $performer->ID, 'studioPassword', true );
4030                                    if ( $password ) {
4031                                            $contents .= __( 'Original Password', 'ppv-live-webcams' ) . ': ' . $password . ' ';
4032                                    }
4033
4034                                    $contents .= '</div> </td><td>';
4035
4036                                    $selectWebcam = get_user_meta( $performer->ID, 'currentWebcam', true );
4037                                    if ( $selectWebcam ) {
4038                                            $contents .= '<a href="' . get_permalink( $selectWebcam ) . '"><i class="icon users"></i>' . self::post_title( $selectWebcam ) . '</a> ';
4039                                            $contents .= get_post_meta( $selectWebcam, 'groupMode', true );
4040
4041                                    }
4042                                    $contents .= '</td></tr>';
4043                            }
4044
4045                            $contents .= '</table>';
4046
4047                    wp_enqueue_script( 'semantic-tablesort', dirname( plugin_dir_url( __FILE__ ) ) . '/scripts/semantic/tablesort.js', array( 'fomantic-ui' ) );
4048
4049                            $contents .= '<PRE style="display: none"><SCRIPT>
4050        jQuery(document).ready( function(){
4051                jQuery("table.ui").tablesort();
4052    });</SCRIPT></PRE>';
```

```
4053
4054
4055                        } else {
4056                                $contents .= __( 'Studio has no performer accounts, yet.', 'ppv-live-webcams' );
4057                        }
4058
4059                        if ( ! $options['studioPerformers'] || $performersCount < $options['studioPerformers'] ) {
4060                                $contents .= '<p><a href="' . add_query_arg(
4061                                        array(
4062                                                'tab'  => 'performers',
4063                                                'view' => 'form',
4064                                        ),
4065                                        $this_page
4066                                ) . '" class="ui button">' . __( 'Add Performer', 'ppv-live-webcams' ) . '</a> ' . __( 'Create a new performer account.', 'ppv-live-webcams' ) . '</p>';
4067                        }
4068                } else {
4069                        $contents .= '<p><a class="ui button" href="' . esc_url( add_query_arg( array( 'tab' => 'performers' ), $this_page ) ) . '">' . __( 'View Performers List', 'ppv-live-webcams' ) . '</a></p>';
4070                }
4071
4072                $content['performers'] = $contents;
4073
4074                /*
4075                        $htmlCode .= '
4076                        </div>
4077                </div>
4078                </div>';
4079
4080                */
4081                // ! webcams tab
4082
4083                /*
4084                        $checked = '';
4085                        if ( $tab == 'webcams') $checked = 'checked';
4086                        if ($checked) $checked1 = true;
4087
4088                        $htmlCode .= '<div class="vwtab">
4089                <input type="radio" id="tab-webcams" name="tab-group-1" '.$checked.'>
4090                <label class="vwlabel" for="tab-webcams">Webcams</label>
4091
4092                <div class="vwpanel">
4093                <div class="vwcontent">';
4094                        $htmlCode .= '<h3 class="ui header">' . __('Webcams', 'ppv-live-webcams') . '</h3> ' . __('Manage webcam listings for studio performers.', 'ppv-live-webcams')  ;
4095                */
4096
4097                $header['webcams'] = __( 'Rooms', 'ppv-live-webcams' );
4098
4099                $contents = '<h3 class="ui header">' . __( 'Rooms', 'ppv-live-webcams' ) . '</h3> ' . __( 'Manage room listings for studio performers.', 'ppv-live-webcams' );
4100
4101                if ( $tab == 'webcams' ) {
4102
4103                        $active = 'webcams';
4104
4105                        switch ( $_GET['view'] ) {
4106                                case 'add':
4107                                        $contents .= '<h4>' . __( 'Create Webcam Room Listing', 'ppv-live-webcams' ) . '</h4>';
```

```
4108
4109                                    // performers list
4110                                    if ( ! $performers ) {
4111                                            $args = array(
4112                                                    'blog_id'      => $GLOBALS['blog_id'],
4113                                                    'meta_key'     => 'studioID',
4114                                                    'meta_value'   => $current_user->ID,
4115                                                    'meta_compare' => '=',
4116                                                    'orderby'      => 'registered',
4117                                                    'order'        => 'DESC',
4118                                                    'fields'       => 'all',
4119                                            );
4120
4121                                            $performers = get_users( $args );
4122                                    }
4123
4124                                    if ( count( $performers ) ) {
4125                                            foreach ( $performers as $performer ) {
4126                                                    $performersCode .= '<input type="checkbox" name="selectedPerformers[]" value="' . $performer->ID . '">' . $performer->user_login . '<br>';
4127                                            }
4128                                    } else {
4129                                            $performersCode .= 'Studio has no performer accounts to select from.';
4130                                    }
4131
4132                                    $action = add_query_arg(
4133                                            array(
4134                                                    'tab'  => 'webcams',
4135                                                    'view' => 'insert',
4136                                            ),
4137                                            $this_page
4138                                    );
4139
4140                                    $infoMsg = __( 'Warning: Make sure webcam listing name is correct. Only administrator can delete webcams.', 'ppv-live-webcams' );
4141                                    $saveMsg = __( 'Save', 'ppv-live-webcams' );
4142
4143                                    $contents .= <<<HTMLCODE
4144 <form class="ui form" method="post" enctype="multipart/form-data" action="$action" name="adminForm" class="w-actionbox">
4145 <table class="ui table celled">
4146 <tr><td>Webcam Listing Name</td><td><input size="32" type="text" name="webcam_name" id="webcam_name" value=""></td></tr>
4147 <tr><td>Performers</td><td align="left">$performersCode</td></tr>
4148 <tr><td></td><td><input class="ui button" type="submit" name="save" id="save" value="$saveMsg" /></td></tr>
4149 </table>
4150 $infoMsg
4151 </form>
4152 HTMLCODE;
4153                                    break;
4154
4155                            case 'insert':
4156                                    $contents .= '<h4>Adding Webcam Listing</h4>';
4157
4158                                    $error = '';
4159
4160                                    $webcam_name = sanitize_text_field( $_POST['webcam_name'] );
4161 $selectedPerformers = isset( $_POST['selectedPerformers'] ) ? (array) $_POST['selectedPerformers'] : array(); // array elements sanitized on use
4162
```

```
Line
4163                                    global $wpdb;
4164                                    $pid = $wpdb->get_var( $wpdb->prepare( "SELECT ID FROM $wpdb->posts WHERE post_title = %s AND post_type = %s", $webcam_name, sanitize_text_field( $options['custom_post'] ) ) );
4165
4166                                    if ( ! $pid ) {
4167                                            $post = array(
4168                                                    'post_name' => sanitize_title_with_dashes( $webcam_name ),
4169                                                    'post_title'  => $webcam_name,
4170                                                    'post_author' => $current_user->ID,
4171                                                    'post_type' => sanitize_text_field( $options['custom_post'] ),
4172                                                    'post_status' => 'publish',
4173                                            );
4174
4175                                            $pid = wp_insert_post( $post );
4176
4177                                            // assign to studio
4178                                            update_post_meta( $pid, 'studioID', $current_user->ID );
4179
4180                                            // assign to performers
4181                                            if ( count( $selectedPerformers ) ) {
4182                                                    foreach ( $selectedPerformers as $performer ) {
4183                                                            add_post_meta( $pid, 'performerID', intval( $performer ), false );
4184                                                    }
4185                                            }
4186                                    } else {
4187                                            $error = __( 'Room listing with that name already exists.', 'ppv-live-webcams' );
4188                                    }
4189
4190                                    if ( $error ) {
4191                                            $contents .= 'Could not create webcam listing: ' . $error;
4192                                    } else {
4193                                            $contents .= __( 'Room listing was successfully created!', 'ppv-live-webcams' );
4194                                    }
4195
4196                                    break;
4197
4198                            default:
4199                                    $webcamsList = 1;
4200                    }
4201
4202                    // webcam actions in list view
4203
4204            } else {
4205                    $webcamsList = 1; // no special action
4206            }
4207
4208            if ( $webcamsList ) {
4209
4210                    $args = array(
4211                            'post_type'    => sanitize_text_field( $options['custom_post'] ),
4212                            'meta_key'     => 'studioID',
4213                            'meta_value'   => $current_user->ID,
4214                            'meta_compare' => '=',
4215                            'orderby'      => 'date',
4216                            'order'        => 'DESC',
4217                    );
```

```
4218
4219                        $webcams = get_posts( $args );
4220
4221                        $webcamsCount = count( $webcams );
4222
4223                        $contents .= '<h4>Webcams List
(' . $webcamsCount . '/' . ( esc_html( $options['studioWebcams'] ) ? esc_html( $options['studioWebcams']
) : '&infin;' ) . ')</h4>';
4224
4225                        if ( $webcamsCount ) {
4226                                $contents .= '<table class="ui table celled striped"><thead><tr>
<th>Webcam</th><th>Status</th><th>Performer(s)</th></tr></thead>';
4227
4228                                foreach ( $webcams as $webcam ) {
4229
4230                                        $contents .= '<tr><td><b>' . esc_html( $webcam-
>post_title ) . '</b><td>' . esc_html( $webcam->post_status ) . '</td>';
4231
4232                                        $contents    .= ' </td><td>';
4233                                        $performerIDs = get_post_meta( $webcam-
>ID, 'performerID', false );
4234                                        if ( $performerIDs ) {
4235                                                if ( count( $performerIDs ) ) {
4236                                                        foreach ( $performerIDs as $performerID ) {
4237
$performer = get_userdata( intval( $performerID ) );
4238                                                                $contents .= esc_html( $performer-
>user_login ) . ' ';
4239                                                        }
4240                                                }
4241                                        }
4242
4243                                                $contents .= '</td>';
4244                                        $contents     .= '</tr>';
4245                                }
4246
4247                                $contents .= '</table>';
4248                        } else {
4249                                $contents .= 'Studio has no webcam listings, yet.';
4250                        }
4251
4252                        if ( ! $options['studioWebcams'] || $webcamsCount < $options['studioWebcams'] ) {
4253                                $contents .= '<p><a href="' . add_query_arg(
4254                                        array(
4255                                                'tab'  => 'webcams',
4256                                                'view' => 'add',
4257                                        ),
4258                                        $this_page
4259                                ) . '"  class="ui button">Add Room</a> Create a new webcam room listing.
</p>';
4260                        }
4261                } else {
4262                        $contents .= '<p><a
href="' . esc_url( add_query_arg( array( 'tab' => 'webcams' ), $this_page ) ) . '" class="ui button
secondary">View Your Room List</a></p>';
4263                }
4264
4265                /*
4266                        $htmlCode .= '
4267                        </div>
4268                </div>
4269                </div>';
4270
4271                        $htmlCode .= '
```

```
4272                    </div>';
4273                    */
4274                    $content['webcams'] = $contents;
4275
4276                    // wallets in Studio Dashboard
4277
4278
4279
4280                    // MicroPayments Tab
4281                    if ( $options['performerWallet'] ) {
4282                            if ( $options['wallet'] == 'MicroPayments' || $options['walletMulti'] >= 1 ) {
4283                                    if ( shortcode_exists( 'videowhisper_transactions' ) ) {
4284
4285                                            if ( isset($_GET['orderby']) ||
 isset($_GET['paged']) ) $active = 'micropayments';
4286
4287                                            $header['micropayments'] = __( 'MicroPayments Wallet', 'ppv-live-
webcams' );
4288
4289                                            $content['micropayments'] = '<h3 class="ui
header">' . __( 'MicroPayments Wallet', 'ppv-live-
webcams' ) . '</h3>' . do_shortcode( '[videowhisper_transactions user_id="' . $current_user->ID . '"]' );
4290                                    }
4291                            }
4292                    }
4293
4294
4295                    // WooWallet Tab
4296                    if ( $options['performerWallet'] ) {
4297                            if ( $options['wallet'] == 'WooWallet' || $options['walletMulti'] >= 1 ) {
4298                                    if ( shortcode_exists( 'woo-wallet' ) ) {
4299
4300                                            if ( ($_GET['wallet_action'] ?? false ) ||
 get_query_var( 'wallet_action' ) ) {
4301                                                    $active = 'woowallet';
4302                                            }
4303                                            $header['woowallet'] = __( 'WooCommerce Wallet', 'ppv-live-
webcams' );
4304
4305                                            $content['woowallet'] = '<h3 class="ui
header">' . __( 'TeraWallet WooCommerce', 'ppv-live-webcams' ) . '</h3>' . do_shortcode( '[woo-
wallet]' );
4306                                    }
4307                            }
4308                    }
4309
4310
4311                            // MyCred Transactions
4312                    if ( $options['performerWallet'] ) {
4313                            if ( $options['wallet'] == 'MyCred' || $options['walletMulti'] >= 1 ) {
4314                                    if ( shortcode_exists( 'mycred_history' ) ) {
4315
4316                                            if ( isset($_GET['pagenum']) || get_query_var( 'pagenum' ) ) {
4317                                                    $active = 'mycred';
4318                                            }
4319                                            $header['mycred'] = __( 'MyCred', 'ppv-live-webcams' );
4320
4321                                            $content['mycred']  = __( 'MyCred Wallet', 'ppv-live-
webcams' ) . '</h3>' . do_shortcode( '[mycred_history user_id="current"]' );
4322                                            $content['mycred'] .= '<style type="text/css">
4323 .pagination, .pagination > LI {
4324 display: inline !important;
4325 padding: 5px;
4326 margin: 2px;
```

```
4327 background-color: #EEE;
4328 }
4329 </style>';
4330                                 }
4331                         }
4332                 }
4333
4334 $htmlCode .= self::sectionsLayout( $header, $content, $active, $options['performerLayout'], '' );
4335
4336                 if ( $atts['include_css'] ) {
4337 $htmlCode .= '<STYLE>' . html_entity_decode( stripslashes( $options['dashboardCSS'] ) ) . '</STYLE>';
4338                 }
4339
4340                         return $htmlCode;
4341         }
4342
4343
4344 //ajax function to retrieve location options
4345         static function vmls_location()
4346         {
4347
4348                 $for = sanitize_text_field( $_POST['for'] );
4349
4350                 $fieldName = sanitize_text_field( $_POST['field_name'] );
4351
4352                 $continent0 = sanitize_text_field( $_POST['continent0'] );
4353                 $country0 = sanitize_text_field( $_POST['country0'] );
4354                 $region0 = sanitize_text_field( $_POST['region0'] );
4355
4356                 $continent1 = sanitize_text_field( $_POST['continent'] );
4357                 $country1   = sanitize_text_field( $_POST['country'] );
4358                 $region1    = sanitize_text_field( $_POST['region'] );
4359
4360                 if (!$continent1) $continent1 = $continent0;
4361                 if (!$country1) $country1 = $country0;
4362                 if (!$region1) $region1 = $region0;
4363
4364                         // output clean
4365                         if (ob_get_length()) {
4366                                 ob_clean();
4367                         }
4368
4369                         $options = self::getOptions();
4370
4371
4372                 $response = array();
4373
4374                                 //load location info
4375                                 $locationsPath = $options['uploadsPath'] . '/_locations/';
4376                                 $locationsPathDefaut = dirname(plugin_dir_path( __FILE__ )). '/data/';
4377
4378                                 $fileLocations = $locationsPath . 'locations.json';
4379
4380 if ( ! file_exists( $fileLocations ) )$fileLocations = $locationsPathDefaut . 'locations.json';
4381
4382 if ( file_exists( $fileLocations ) ) $locations = json_decode( file_get_contents( $fileLocations ), true
);
4383                                 if ( ! is_array( $locations ) ) $locations = array();
4384
```

```
Line
4385                                $continents = array();
4386                                $countries = array();
4387                                $regions = array();
4388
4389
4390                                foreach ( $locations as $continent_code => $continent )
4391                                {
4392                                        $continents[ $continent_code ] = $continent['continent_name'];
4393
4394    if ($continent1) if ($continent['continent_name'] == $continent1) foreach ( $continent['countries'] as $c
        ountry_code => $country )
4395                                        {
4396                                                $countries[ $country_code ] = $country['country_name'];
4397
4398                                                if ( isset( $country['regions'] ) ) {
4399    if ($country1) if ($country['country_name'] == $country1) foreach ( $country['regions'] as $region_code =
        > $region ) {
4400
        $regions[ $region_code ] = $region['region_name'];
4401                                                                }
4402                                                }
4403                                        }
4404                                }
4405
4406                                $contentCode = '';
4407
4408                                if (count($countries))
4409                                {
4410                                        echo '<div class="ui fluid search selection dropdown v-select
        vwAjaxDropdown">
4411                                        <input type="hidden" name="' . esc_attr($fieldName) . '_country"
        id="' . esc_attr($fieldName) . '_country" value="' . esc_attr($country1). '"
        onchange="updateLocationField(\'' . esc_attr($fieldName) . '\',
        \'continent\',\'' . esc_attr($continent1) . '\',\'' . esc_attr($country1) . '\',\'' . esc_attr($region1)
        . '\' )" >
4412                                        <i class="dropdown icon"></i>
4413                                        <div class="default text">' .__( 'Country', 'ppv-live-
        webcams' ) . '</div>
4414                                        <div class="menu">';
4415
4416                                        foreach ( $countries as $key => $country ) {
4417                                                echo '<div class="item" data-
        value="' . esc_attr( htmlspecialchars( stripslashes( $country ) ) ) . '"><i
        class="' . esc_attr( self::language2flag($key)) . ' flag"></i>
        ' . esc_html( htmlspecialchars( stripslashes( $country ) ) ) . '</div>';
4418                                        }
4419
4420                                        echo '</div></div>';
4421                                }
4422
4423                                if (count($regions))
4424                                {
4425                                        echo '<div class="ui fluid search selection dropdown v-select
        vwAjaxDropdown">
4426                                        <input type="hidden" name="' . esc_attr($fieldName) . '_region"
        id="' . esc_attr($fieldName) . '_region" value="' . esc_attr($region1). '"  >
4427                                        <i class="dropdown icon"></i>
4428                                        <div class="default text">' .__( 'Region', 'ppv-live-
        webcams' ) . '</div>
4429                                        <div class="menu">';
4430
4431                                        foreach ( $regions as $key => $region ) {
4432                                                echo '<div class="item" data-
        value="' . esc_attr( htmlspecialchars( stripslashes( $region ) ) ) . '">' . esc_html( htmlspecialchars( s
```

```
tripslashes( $region ) ) ) . '</div>';
                                        }

                                        echo '</div></div>';
                                }

                                echo '<script>
                                jQuery(document).ready(function(){
                                        jQuery(".vwAjaxDropdown").dropdown();
                                });
                                </script>';

                                exit;
        }

        // end studio dashboard
        // start performer dashboard

        static function videowhisper_webcams_performer( $atts ) {

                $options = get_option( 'VWliveWebcamsOptions' );

                if ( ! is_user_logged_in() ) {
                        return '<div class="ui ' . $options['interfaceClass'] . ' header"> <i class="sign in icon"></i> ' . __( 'Login to access dashboard!', 'ppv-live-webcams' ) . '</div>' . do_shortcode( '[videowhisper_login_form register="1" role="' . $options['rolePerformer'] . '"]' );
                }

                // '<BR><a class="ui button primary qbutton" href="' . wp_login_url() . '">' . __('Login', 'ppv-live-webcams') . '</a>  <a class="ui button secondary qbutton" href="' . ( ($options['registerFrontend'] && $options['p_videowhisper_register_performer']) ? get_permalink($options['p_videowhisper_register_performer']) : wp_registration_url() ) . '">' . __('Register', 'ppv-live-webcams') . '</a>';

                // shortocode attributes
                $atts = shortcode_atts(
                        array(
                                'include_css' => '1',
                        ),
                        $atts,
                        'videowhisper_webcams_performer'
                );
                $atts = array_map('esc_attr', $atts);

                $current_user = wp_get_current_user();

                if ( ! $current_user->ID ) {
                        return __( 'Login is required to access this section!', 'ppv-live-webcams' );
                }

                $uid = $current_user->ID;

                // access keys
                $userkeys   = $current_user->roles;
                $userkeys[] = $current_user->user_login;
                $userkeys[] = $current_user->user_nicename;
                $userkeys[] = $current_user->ID;
                $userkeys[] = $current_user->user_email;

                $roleS = implode( ',', $current_user->roles );
                if ( ! self::any_in_array( self::getRolesPerformer( $options ), $current_user->roles ) ) {
```

```
Line
4487                        return __( 'User role does not allow publishing webcams!', 'ppv-live-webcams' ) . ' (' . $roleS . ')';
4488                }
4489
4490                // process user's sessions to show updated balance
4491                self::billSessions( $current_user->ID );
4492
4493                // semantic ui : performer dashboard
4494                self::enqueueUI();
4495                $htmlCode = '';
4496
4497                if ( $options['dashboardMessage'] ) {
4498                        $htmlCode .= '<div id="performerDashboardMessage" class="ui ' . $options['interfaceClass'] . ' segment">' . html_entity_decode( stripslashes( $options['dashboardMessage'] ) ) . '</div>';
4499                }
4500
4501                $this_page = self::getCurrentURL();
4502
4503                $htmlCode .= do_shortcode( '[videowhisper_account_records]' );
4504
4505                if ( ! self::userEnabled( $current_user, $options, 'Performer' ) ) {
4506                        return $htmlCode . '<div id="performerDashboardMessage" class="ui yellow segment">' . __( 'Your account is not currently enabled. Update your account records and wait for site admins to approve your account.', 'ppv-live-webcams' ) . '</div>';
4507                }
4508
4509                // disabled by studio: no access
4510                $disabled = get_user_meta( $current_user->ID, 'studioDisabled', true );
4511                if ( $disabled ) {
4512                        return $htmlCode . '<div id="performerDashboardMessage" class="ui red segment">' . __( 'Studio disabled your account: dashboard access is forbidden. Contact studio or site administrator!', 'ppv-live-webcams' ) . '</div>';
4513                }
4514
4515                // updating records
4516                if ( $_GET['updateRecords'] ?? false) {
4517                        if ( $_GET['updateRecords'] != 'update' ) {
4518                                return $htmlCode;
4519                        }
4520                }
4521
4522                        // display dashboard info
4523
4524                        // ! webcams manage: shows when managing
4525
4526                        // ! select webcam listing
4527
4528                $postID = 0;
4529
4530                if ( $selectWebcam = intval( $_GET['selectWebcam'] ?? 0 ) ) {
4531
4532                        // verify if valid (owner or assigned)
4533                        $webcamSelected = get_post( $selectWebcam );
4534                        $warnCode       = '';
4535
4536                        if ( ! $webcamSelected ) {
4537                                $selectWebcam = '';
4538                                $warnCode     .= 'This room post does not exist: #' . $selectWebcam;
4539                        } else {
4540                                $performerIDs = get_post_meta( $webcamSelected->ID, 'performerID', false );
4541                                if ( ! $performerIDs ) {
```

```
4542                                    $performerIDs = array();
4543                            }
4544
4545                            if ( $selectWebcam && $webcamSelected->post_author != $current_user->ID && ! in_array( $current_user->ID, $performerIDs ) ) {
4546                                    $selectWebcam = '';
4547                                    $warnCode     .= 'You are not performer in room: #' . $selectWebcam; }
4548
4549                            if ( $selectWebcam && $webcamSelected->post_type != $options['custom_post'] ) {
4550                                    $selectWebcam = '';
4551                                    $warnCode     .= 'This post is not a valid room type: #' . $selectWebcam . ': ' . $webcamSelected->post_type . '/' . $options['custom_post'] ; }
4552                    }
4553
4554                    if ( $selectWebcam ) {
4555                            update_user_meta( $current_user->ID, 'currentWebcam', $selectWebcam );
4556                            $postID = $selectWebcam;
4557                    } else {
4558                            $htmlCode .= '<BR>' . __( 'Selected webcam listing is invalid.', 'ppv-live-webcams' ) . ' ' . $warnCode;
4559                    }
4560            }
4561
4562            $selectWebcamList = 0;
4563            if ( ! $postID ) {
4564                    // get or setup webcam post for this user
4565                    if ( ! $options['performerWebcams'] ) {
4566                            $postID = self::webcamPost(); // default cam
4567                    } else {
4568                            $postID = get_user_meta( $current_user->ID, 'currentWebcam', true );
4569                            if ( ! $postID ) {
4570                                    $selectWebcamList = 1;
4571                            }
4572                    }
4573            }
4574
4575            // ! manage webcams
4576            if ( $selectWebcamList || ( $_GET['changeWebcam'] ?? '' ) ) {
4577                    $htmlCode .= '<div id="performerWebcamsManage" class="ui ' . $options['interfaceClass'] . ' segment">';
4578
4579                    $postID = ''; // select or manage
4580
4581                    switch ( $_GET['view'] ?? '' ) {
4582                            case 'add':
4583                                    $htmlCode .= '<h4>' . __( 'Create Webcam Listing', 'ppv-live-webcams' ) . '</h4>';
4584
4585                                    $action = add_query_arg(
4586                                            array(
4587                                                    'changeWebcam' => 'manage',
4588                                                    'view'         => 'insert',
4589                                            ),
4590                                            $this_page
4591                                    );
4592
4593                                    $performersCode = '<input size="48" type="text" name="performersCSV" id="performersCSV" value=""><br>' . __( 'Partner performers that can also go live in this room: Comma separated list of performer usernames or emails.', 'ppv-live-webcams' );
4594
4595                                    $msg1 = __( 'Room Listing Name', 'ppv-live-webcams' );
4596                                    $msg2 = __( 'Performers', 'ppv-live-webcams' );
```

```
Line
4597                                    $msg3 = __( 'Save', 'ppv-live-webcams' );
4598                                    $msg4 = __( 'Warning: Make sure webcam listing name is correct. Only administrator can delete webcams.', 'ppv-live-webcams' );
4599
4600                                    $htmlCode .= <<<HTMLCODE
4601 <form class="ui form" method="post" enctype="multipart/form-data" action="$action" name="adminForm">
4602 <table class="g-input" width="500px">
4603 <tr><td>$msg1</td><td><input size="32" type="text" name="webcam_name" id="webcam_name" value=""></td></tr>
4604 <tr><td>$msg2</td><td align="left">$performersCode</td></tr>
4605 <tr><td></td><td><input class="ui button videowhisperButton" type="submit" name="save" id="save" value="$msg3" /></td></tr>
4606 </table>
4607 $msg4
4608 </form>
4609 HTMLCODE;
4610                                    break;
4611
4612                            case 'insert':
4613                                    $htmlCode .= '<h4>' . __( 'Adding Webcam Listing', 'ppv-live-webcams' ) . '</h4>';
4614
4615                                    $error = '';
4616
4617                                    $webcam_name = sanitize_text_field( $_POST['webcam_name'] );
4618
4619                                    global $wpdb;
4620                                    $pid = $wpdb->get_var( $wpdb->prepare( "SELECT ID FROM $wpdb->posts WHERE post_title = '%s' AND post_type='" . $options['custom_post'] . "'", $webcam_name ) );
4621
4622                                    if ( ! $pid ) {
4623                                            $post = array(
4624                                                    'post_name' => sanitize_title_with_dashes( $webcam_name ),
4625                                                    'post_title'  => $webcam_name,
4626                                                    'post_author' => $current_user->ID,
4627                                                    'post_type'   => $options['custom_post'],
4628                                                    'post_status' => 'publish',
4629                                            );
4630
4631                                            $pid = wp_insert_post( $post );
4632
4633                                            // assign to studio
4634                                            update_post_meta( $pid, 'studioID', $current_user->ID );
4635
4636                                            // assign to performers
4637 if ( $performersCSV = sanitize_text_field( $_POST['performersCSV'] ) ) {
4638                                                    $selectedPerformers = array();
4639
4640                                                    $performers = explode( ',', $performersCSV );
4641                                                    if ( count( $performers ) ) {
4642                                                            foreach ( $performers as $performerV ) {
4643                                                                    $performerID = '';
4644                                                                    $value = trim( $performerV );
4645                                                                    if ( is_email( $value ) ) {
4646 if ( $performerID = get_user_by( 'email', $value ) ) {
4647 $selectedPerformers[] = $performerID;
4648                                                                    }
4649 } elseif ( $performerID = get_user_by( 'login', $value ) ) {
```

```
4650                                                                    $selectedPerformers[] = $performerID;
4651                                                                }
4652
4653                                                                if ( $value && ! $performerID ) {
4654                                                                    $htmlCode .= '"' . $value . '" ' . __( 'Warning: Performer not found: use a valid username (login) or email!', 'ppv-live-webcams' ) . '<BR>';
4655                                                                }
4656                                                            }
4657                                                        }
4658
4659                                                        if ( count( $selectedPerformers ) ) {
4660                                                            foreach ( $selectedPerformers as $performerID ) {
4661                                                                add_post_meta( $pid, 'performerID', (int) $performerID, false );
4662                                                            }
4663                                                        }
4664                                                    }
4665                                                } else {
4666                                                    $error = __( 'Room listing with that name already exists.', 'ppv-live-webcams' );
4667                                                }
4668
4669                                                if ( $error ) {
4670                                                    $htmlCode .= __( 'Could not create webcam listing:', 'ppv-live-webcams' ) . $error;
4671                                                } else {
4672                                                    $htmlCode .= __( 'Room listing was successfully created!', 'ppv-live-webcams' );
4673                                                }
4674
4675                                                break;
4676
4677                                            default:
4678                                                $webcamsList = 1;
4679                                        }
4680
4681                                        if ( $webcamsList ?? false ) {
4682
4683                                            $args = array(
4684                                                'post_type'      => $options['custom_post'],
4685                                                'author'         => $current_user->ID,
4686                                                'orderby'        => 'date',
4687                                                'order'          => 'DESC',
4688                                                'posts_per_page' => -1,
4689                                            );
4690
4691                                            $webcamsOwned = get_posts( $args );
4692
4693                                            $args = array(
4694                                                'post_type'      => $options['custom_post'],
4695                                                'meta_key'       => 'performerID',
4696                                                'meta_value'     => $current_user->ID,
4697                                                'meta_compare'   => '=',
4698                                                'orderby'        => 'date',
4699                                                'order'          => 'DESC',
4700                                                'posts_per_page' => -1,
4701
4702                                            );
4703
4704                                            $webcamsAssigned = get_posts( $args );
```

```
4705
4706                                    $webcams = array_merge( $webcamsOwned, $webcamsAssigned );
4707                                    $webcams = array_unique( $webcams, SORT_REGULAR );
4708
4709                                    $webcamsCount = count( $webcams );
4710
4711                                    $htmlCode .= '<h4>' . __( 'Select Room Listing', 'ppv-live-webcams' ) . ' (' . $webcamsCount . '/' . ( $options['performerWebcams'] ? $options['performerWebcams'] : '&infin;' ) . ')</h4>';
4712                                    $htmlCode .= __( 'Select a room to enter, manage, operate.', 'ppv-live-webcams' );
4713
4714                                    $webcamCreated = 0;
4715                                    if ( $webcamsCount ) {
4716                                            $htmlCode     .= '<br>' . __( 'Rooms (webcam venue listings) you created or were granted access to broadcast:', 'ppv-live-webcams' );
4717                                            $currentWebcam = get_user_meta( $current_user->ID, 'currentWebcam', true );
4718
4719                                            foreach ( $webcams as $webcam ) {
4720                                                    $htmlCode .= '<br><a class="ui button secondary" href="' . add_query_arg( array( 'selectWebcam' => $webcam->ID ), $this_page ) . '"><i class="ui icon users"></i><b>' . $webcam->post_title . '</b></a> ';
4721
4722                                                    $htmlCode .=  '<label class="ui label basic">' . get_post_meta( $webcam->ID, 'groupMode', true ) . '</label>';
4723
4724                                                    $performerIDs = get_post_meta( $webcam->ID, 'performerID', false );
4725                                                    if ( $performerIDs ) {
4726                                                            if ( count( $performerIDs ) ) {
4727                                                                    foreach ( $performerIDs as $performerID ) {
4728                                                                            $performer = get_userdata( $performerID );
4729                                                                            $htmlCode .= ' <label class="ui label"><i class="ui icon users"></i>' . $performer->user_login . '</label>';
4730                                                                    }
4731                                                            }
4732                                                    }
4733
4734                                                    if ( $currentWebcam == $webcam->ID ) {
4735                                                            echo ' - ' . __( 'Selected', 'ppv-live-webcams' );
4736                                                    }
4737                                            }
4738                                    } else {
4739                                            $htmlCode .= '<br>' . __( 'You do not have any active webcam room: one will be setup for you and selected.', 'ppv-live-webcams' ) . '<br>';
4740                                            $postID    = self::webcamPost();
4741
4742                                            if ( $postID ) {
4743                                                    $post = get_post( $postID );
4744                                            }
4745                                            if ( ! $post ) {
4746                                                    $postID = 0;
4747                                            }
4748
4749                                            if ( ! $postID ) {
4750                                                    $htmlCode .= 'Error: Could not setup a webcam post!';
4751                                            } else {
4752                                                    $htmlCode     .= '<b>' . $post->post_title . '</b> #' . $postID;
4753                                                    $webcamCreated = 1;
4754                                            }
```

```
4755                                }
4756
4757                                if ( ! $options['performerWebcams'] ||
 $webcamsCount < $options['performerWebcams'] ) {
4758                                        if ( ! $webcamCreated ) {
4759                                                $htmlCode .= '<p><a href="' . add_query_arg(
4760                                                        array(
4761                                                                'changeWebcam' => 'manage',
4762                                                                'view'         => 'add',
4763                                                        ),
4764                                                        $this_page
4765                                                ) . '"  class="ui button">' . __( 'Add Room', 'ppv-live-
webcams' ) . '</a> ' . __( 'Create a new webcam room listing.', 'ppv-live-webcams' ) . '</p>';
4766                                        }
4767                                }
4768                        } else {
4769                                $htmlCode .= '<p><a class="ui button"
href="' . add_query_arg( array( 'changeWebcam' => 'select' ), $this_page ) . '">' . __( 'View Your Room
List', 'ppv-live-webcams' ) . '</a></p>';
4770                        }
4771
4772                        $htmlCode .= '</div>';
4773                }
4774                // ! end webcams manage
4775
4776                if ( ! $postID ) {
4777                        if ( $atts['include_css'] ) {
4778
$htmlCode .= '<STYLE>' . html_entity_decode( stripslashes( $options['dashboardCSS'] ) ) . '</STYLE>';
4779                        }
4780                        return $htmlCode;
4781                }
4782
4783                if ( self::roomSuspended($postID, $options) )
4784                {
4785                        $htmlCode .= '<div class="ui red segment"><h4>' . __('Room Suspended', 'ppv-live-
webcams') . '</h4>' . wp_kses_post( $options['suspendMessage'] ) . '</div>';
4786                        return $htmlCode;
4787                }
4788                // !room updates
4789
4790                // FEATURES
4791                // ! edit room
4792                if ( $_POST['setup'] ?? false ) {
4793
4794                        if ( isset($webcamPost) ) {
4795                                $webcamPost = get_post( $postID );
4796                        }
4797
4798                                //set room subscription tier
4799                                if ( isset( $_POST['subscriptionTier'] ) )
4800                                {
4801                                        $subscriptionTier = intval( $_POST['subscriptionTier'] );
4802
update_post_meta( $postID, 'vw_subscription_tier', $subscriptionTier );
4803                                }
4804
4805                        // costPerMinute
4806                        if ( self::inList( $userkeys, $options['costPerMinute'] ) ) {
4807                                $costPerMinute = round( $_POST['costPerMinute'], 2 );
4808
4809                                // check range
4810                                if ( $costPerMinute > 0 ) {
```

```
4811                                    if ( $options['ppvPPMmin'] > $costPerMinute || $options['ppvPPMmax'] < $costPerMinute ) {
4812                                            $costPerMinute = 0;
4813                                            $htmlCode     .= '<div class="warning">' . __( 'Custom CPM out of range: removed.', 'ppv-live-webcams' ) . ' (' . $options['ppvPPMmin'] . '-' . $options['ppvPPMmax'] . ')</div>';
4814                                    }
4815                            }
4816
4817                            if ( $costPerMinute ) {
4818                                    update_post_meta( $postID, 'vw_costPerMinute', $costPerMinute );
4819                            } else {
4820                                    delete_post_meta( $postID, 'vw_costPerMinute' );
4821                            }
4822
4823                            // audio
4824                            $costPerMinuteAudio = round( $_POST['costPerMinuteAudio'], 2 );
4825
4826                            // check range
4827                            if ( $costPerMinuteAudio > 0 ) {
4828                                    if ( $options['ppvPPMmin'] > $costPerMinuteAudio || $options['ppvPPMmax'] < $costPerMinuteAudio ) {
4829                                            $costPerMinuteAudio = 0;
4830                                            $htmlCode          .= '<div class="warning">' . __( 'Custom CPM out of range: removed.', 'ppv-live-webcams' ) . ' (' . $options['ppvPPMmin'] . '- ' . $options['ppvPPMmax'] . ')</div>';
4831                                    }
4832                            }
4833
4834                            if ( $costPerMinuteAudio ) {
4835                                    update_post_meta( $postID, 'vw_costPerMinuteaudio', $costPerMinuteAudio );
4836                            } else {
4837                                    delete_post_meta( $postID, 'vw_costPerMinuteaudio' );
4838                            }
4839
4840                            // text
4841                            $costPerMinuteText = round( $_POST['costPerMinuteText'], 2 );
4842
4843                            // check range
4844                            if ( $costPerMinuteText > 0 ) {
4845                                    if ( $options['ppvPPMmin'] > $costPerMinuteText || $options['ppvPPMmax'] < $costPerMinuteText ) {
4846                                            $costPerMinuteText = 0;
4847                                            $htmlCode         .= '<div class="warning">' . __( 'Custom CPM out of range: removed.', 'ppv-live-webcams' ) . ' (' . $options['ppvPPMmin'] . '- ' . $options['ppvPPMmax'] . ')</div>';
4848                                    }
4849                            }
4850
4851                            if ( $costPerMinuteText ) {
4852                                    update_post_meta( $postID, 'vw_costPerMinutetext', $costPerMinuteText );
4853                            } else {
4854                                    delete_post_meta( $postID, 'vw_costPerMinutetext' );
4855                            }
4856                    }
4857
4858                    // costPerMinute
4859                    if ( self::inList( $userkeys, $options['costPerMinuteGroup'] ) ) {
4860                            $costPerMinute = round( $_POST['costPerMinuteGroup'], 2 );
4861
4862                            // check range
4863                            if ( $costPerMinute > 0 ) {
```

```
Line
4864                                    if ( $options['ppvPPMmin'] > $costPerMinute || $options['ppvPPMmax'] < $costPerMinute ) {
4865                                            $costPerMinute = 0;
4866                                            $htmlCode     .= '<div class="warning">' . __( 'Custom group CPM out of range: removed.', 'ppv-live-webcams' ) . '</div>';
4867                                    }
4868                            }
4869
4870                            if ( $costPerMinute ) {
4871                                    update_post_meta( $postID, 'vw_costPerMinuteGroup', $costPerMinute );
4872                            } else {
4873                                    delete_post_meta( $postID, 'vw_costPerMinuteGroup' );
4874                            }
4875                    }
4876
4877                    // slots 2way
4878                    if ( isset($options['slots2way']) && self::inList( $userkeys, $options['slots2way'] ) ) {
4879                            $slots2way = round( $_POST['slots2way'] );
4880
4881                            if ( $slots2way ) {
4882                                    update_post_meta( $postID, 'vw_slots2way', $slots2way );
4883                            } else {
4884                                    delete_post_meta( $postID, 'vw_slots2way' );
4885                            }
4886                    }
4887
4888                    // $htmlCode .= 'Upload?'. $_FILES['uploadPicture']['tmp_name'];
4889
4890                    // uploadPicture
4891                    if ( self::inList( $userkeys, $options['uploadPicture'] ) ) {
4892                            if ( $filename = $_FILES['uploadPicture1']['tmp_name'] ) {
4893                                    $htmlCode .= 'Processing picture upload... ';
4894
4895                                    $ext     = strtolower( pathinfo( $_FILES['uploadPicture1']['name'], PATHINFO_EXTENSION ) );
4896                                    $allowed = array( 'jpg', 'jpeg', 'png', 'gif' );
4897                                    if ( ! in_array( $ext, $allowed ) ) {
4898                                            return 'Unsupported file extension!';
4899                                    }
4900
4901                                    list($width, $height) = getimagesize( $filename );
4902
4903                                    if ( $width && $height ) {
4904
4905                                            // delete previous image(s)
4906                                            $htmlCode .= self::delete_associated_media( $postID ); // deletes files, call before copy
4907
4908                                    if (!isset($webcamPost)) $webcamPost = get_post( $postID );
4909                                            $room_name = sanitize_text_field( $webcamPost->post_title );
4910
4911
4912                                            $dir = sanitize_text_field( $options['uploadsPath'] );
4913                                            if ( ! file_exists( $dir ) ) {
4914                                                    mkdir( $dir );
4915                                            }
4916                                            $dir .= '/_pictures';
4917                                            if ( ! file_exists( $dir ) ) {
4918                                                    mkdir( $dir );
```

```
4919                                                    }
4920                                                    // save file
4921                                                    $destination = "$dir/$room_name.$ext";
4922                                                    // copy source
4923
4924    $errorUp = self::handle_upload( $_FILES['uploadPicture1'], $destination ); // handle trough
        wp_handle_upload()
4925                                                    if ( $errorUp ) {
4926                                                            $htmlCode .= '<br>' . 'Error uploading
        ' . esc_html( $filename . ':' . $errorUp );
4927                                                    }
4928
4929                                                    if ( file_exists( $destination ) ) {
4930
4931                                                            if ( $postID ) {
4932    update_post_meta( $postID, 'picture', $destination );
4933    update_post_meta( $postID, 'hasPicture', 1 );
4934                                                            }
4935
4936                                                            // update post image
4937
4938    if ( ! function_exists( 'wp_generate_attachment_metadata' ) ) {
4939                                                                    require ABSPATH . 'wp-
        admin/includes/image.php';
4940                                                            }
4941
4942    $wp_filetype = wp_check_filetype( basename( $destination ), null );
4943
4944                                                            $attachment = array(
4945                                                                    'guid'           => $destination,
4946                                                                    'post_mime_type' => $wp_filetype['type'],
4947                                                                    'post_title'     => $room_name,
4948                                                                    'post_content'   => '',
4949                                                                    'post_status'    => 'inherit',
4950                                                            );
4951
4952    $attach_id = wp_insert_attachment( $attachment, $destination, $postID );
4953                                                            set_post_thumbnail( $postID, $attach_id );
4954
4955                                                            // update post imaga data
4956
4957    $attach_data = wp_generate_attachment_metadata( $attach_id, $destination );
4958    wp_update_attachment_metadata( $attach_id, $attach_data );
4959
4960                                                            if ( ! file_exists( $destination ) ) {
                                                                        $htmlCode .= __( 'ERROR: Missing ', 'ppv-
        live-webcams' ) . $destination;
4961                                                            }
4962
4963                                                            // create thumb
4964
4965    $imageData = file_get_contents( $destination ); // local image file
4966
4967                                                            if ( $imageData ) {
4968                                                                    // $htmlCode .= 'Generating thumb... ';
4969                                                                    $thumbWidth  = $options['thumbWidth'];
4970                                                                    $thumbHeight = $options['thumbHeight'];
```

```
4971  $src = imagecreatefromstring( $imageData );
4972
4973                                                          unset( $imageData );// prevent memory
      leaks
4974
4975  $tmp = imagecreatetruecolor( $thumbWidth, $thumbHeight );
4976                                                          if ( $tmp ) {
4977                                                                  $dir = $options['uploadsPath'];
4978                                                                  if ( ! file_exists( $dir ) ) {
4979                                                                          mkdir( $dir );
4980                                                                  }
4981
4982                                                                  $dir .= '/_thumbs';
4983                                                                  if ( ! file_exists( $dir ) ) {
4984                                                                          mkdir( $dir );
4985                                                                  }
4986
4987  $thumbFilename = "$dir/$room_name-picture.jpg";
4988  imagecopyresampled( $tmp, $src, 0, 0, 0, 0, $thumbWidth, $thumbHeight, $width, $height );
4989
4990  imagejpeg( $tmp, $thumbFilename, 95 );
4991
4992                                                                  // prevent memory leaks in case
      of loops (due to filters on other plugins)
4993                                                                  if ( $src ) {
4994                                                                          imagedestroy( $src );
4995                                                                  }
4996                                                                  if ( $tmp ) {
4997                                                                          imagedestroy( $tmp );
4998                                                                  }
4999
5000  if ( file_exists( $thumbFilename ) ) {
5001                                                                          // detect tiny images
      without info
5002                                                                          // if
      (filesize($thumbFilename)>5000) $picType = 1;
5003                                                                          // else $picType = 2;
5004
5005                                                                          // update post meta
5006                                                                          if ( $postID ) {
5007  update_post_meta( $postID, 'thumbPicture', $thumbFilename );
5008  update_post_meta( $postID, 'hasThumb', 1 );
5009                                                                          }
5010
5011                                                                          // $htmlCode .= '
      Updating picture... ' . $thumbFilename;
5012
5013                                                                          $htmlCode .= __( 'Picture
      Updated.', 'ppv-live-webcams' );
5014                                                                  } else {
5015                                                                          $htmlCode .= __( 'ERROR:
      Cound not create JPG thumb file ', 'ppv-live-webcams' ) . $thumbFilename;
5016                                                                  }
5017                                                          } else {
5018                                                                  $htmlCode .= __( 'ERROR: Cound
      not create temporary image!', 'ppv-live-webcams' );
5019                                                          }
5020                                                  } else {
```

```
5021                                                    $htmlCode .= __( 'ERROR: Failed loading image data ', 'ppv-live-webcams' ) . $destination;
5022                                                }
5023                                            } else {
5024                                                    $htmlCode .= __( 'ERROR: Upload copy failed. File does not exist ', 'ppv-live-webcams' ) . $destination;
5025                                            }
5026                                        } else {
5027                                                $htmlCode .= __( 'ERROR: Could not retrieve image size for ', 'ppv-live-webcams' ) . $filename;
5028                                        }
5029                                }
5030
5031                                $showImage = sanitize_text_field( $_POST['showImage'] );
5032                                update_post_meta( $postID, 'showImage', $showImage );
5033
5034                        }
5035
5036                        if ( $options['lovense'] && $options['lovenseToy'] == 'manual' )
5037                        {
5038                                $lovenseToy = intval( $_POST['lovenseToy'] );
5039                                update_post_meta( $postID, 'lovenseToy', $lovenseToy );
5040                        }
5041
5042
5043                        // category
5044                        if ( self::inList( $userkeys, $options['roomCategory'] ) ) {
5045                                $category = (int) $_POST['newcategory'];
5046                                wp_set_post_categories( $postID, array( $category ) );
5047                        }
5048
5049                        // roomDescription
5050                        if ( self::inList( $userkeys, $options['roomDescription'] ) ) {
5051
5052                                $roomDescription = sanitize_text_field( $_POST['roomDescription'] );
5053                                wp_update_post(
5054                                        array(
5055                                                'ID'           => $postID,
5056                                                'post_content' => $roomDescription,
5057                                        )
5058                                );
5059                        }
5060
5061                        // roomLabel
5062                        if ( self::inList( $userkeys, $options['roomLabel'] ) ) {
5063                                $roomLabel = sanitize_text_field( $_POST['roomLabel'] );
5064                                update_post_meta( $postID, 'vw_roomLabel', $roomLabel );
5065                        }
5066
5067                        // roomBrief
5068                        if ( self::inList( $userkeys, $options['roomBrief'] ) ) {
5069                                $roomBrief = sanitize_text_field( $_POST['roomBrief'] );
5070                                update_post_meta( $postID, 'vw_roomBrief', $roomBrief );
5071                        }
5072
5073                        // roomTags
5074                        if ( self::inList( $userkeys, $options['roomTags'] ) ) {
5075                                $roomTags = sanitize_text_field( $_POST['roomTags'] );
5076                                wp_set_post_tags( $postID, $roomTags, false );
5077                        }
5078
5079                        // banCountries
```

```
Line
5080                        if ( self::inList( $userkeys, $options['banCountries'] ) ) {
5081                                $banCountries = sanitize_text_field( $_POST['banCountries'] );
5082
5083                                $banLocations = str_getcsv( $banCountries, ',' );
5084
5085                                delete_post_meta( $postID, 'vw_banCountries' );
5086
5087                                if ( is_array( $banLocations ) ) {
5088                                        if ( ! empty( $banLocations ) ) {
5089                                                foreach ( $banLocations as $value ) {
5090                                                        if ( trim( $value ) ) {
5091                                                                add_post_meta( $postID, 'vw_banCountries', trim( $value ) );
5092                                                        }
5093                                                }
5094                                        }
5095                                }
5096                        }
5097                        /*
5098                        // private2way
5099                        if ( self::inList( $userkeys, $options['private2way'] ) ) {
5100                                $private2way = sanitize_text_field( $_POST['private2way'] );
5101                                update_post_meta( $postID, 'vw_private2way', $private2way );
5102                        }
5103                        */
5104                        // accessList
5105                        if ( self::inList( $userkeys, $options['accessList'] ) ) {
5106                                $accessList = sanitize_text_field( $_POST['accessList'] );
5107                                update_post_meta( $postID, 'vw_accessList', $accessList );
5108                        }
5109
5110                        // accessPrice
5111                        if ( isset($options['accessPrice']) ) if ( self::inList( $userkeys, $options['accessPrice'] ) ) {
5112
5113                                $accessPrice = round( $_POST['accessPrice'] ?? 0, 2 );
5114                                update_post_meta( $postID, 'vw_accessPrice', $accessPrice );
5115
5116                                if ( class_exists( 'VWpaidMembership' ) ) {
5117                                        // var_dump($webcamPost);
5118
5119                                        if (!isset($webcamPost)) $webcamPost = get_post( $postID );
5120                                        \VWpaidMembership::contentEdit( $webcamPost, array( 'price' => $accessPrice ) );
5121
5122                                } else {
5123                                        // mycred
5124                                        $mCa = array(
5125                                                'status'       => 'enabled',
5126                                                'price'        => $accessPrice,
5127                                                'button_label' => 'Buy Access Now', // default button label
5128                                                'expire'       => 0, // default no expire
5129                                        );
5130
5131                                        if ( $accessPrice ) {
5132                                                update_post_meta( $postID, 'myCRED_sell_content', $mCa );
5133                                        } else {
5134                                                delete_post_meta( $postID, 'myCRED_sell_content' );
5135                                        }
5136                                }
5137                        }
```

```
Line
5138
5139                        // accessPassword
5140                        $accessPassword = ''; // remove if not enabled
5141                        if ( self::inList( $userkeys, $options['accessPassword'] ) ) {
5142                                $accessPassword = sanitize_text_field( $_POST['accessPassword'] );
5143                        }
5144
5145                        // if ($accessPassword) $htmlCode .= 'Webcam was password protected.';
5146
5147                        wp_update_post(
5148                                array(
5149                                        'ID'            => $postID,
5150                                        'post_password' => $accessPassword,
5151                                )
5152                        );
5153 /*
5154                        // logoCustom
5155                        if ( self::inList( $userkeys, $options['logoCustom'] ) ) {
5156                                $logoImage = sanitize_text_field( $_POST['logoImage'] );
5157                                update_post_meta( $postID, 'vw_logoImage', $logoImage );
5158
5159                                $logoLink = sanitize_text_field( $_POST['logoLink'] );
5160                                update_post_meta( $postID, 'vw_logoLink', $logoLink );
5161
5162                                update_post_meta( $postID, 'vw_logo', 'custom' );
5163                        }
5164 */
5165                        // presentationMode
5166 if ( isset($options['presentationMode']) && isset($_POST['presentationMode']) ) if ( self::inList( $userkeys, $options['presentationMode'] ) ) {
5167                                $roomLabel = sanitize_text_field( $_POST['presentationMode'] ?? 0 );
5168                                update_post_meta( $postID, 'vw_presentationMode', $roomLabel );
5169                        } else {
5170                                update_post_meta( $postID, 'vw_presentationMode', '' );
5171                        }
5172                }
5173
5174                // feature updates
5175
5176 /*
5177                // transcode
5178                if ( self::inList( $userkeys, $options['multicam'] ) ) {
5179                        update_post_meta( $postID, 'vw_multicam', $options['multicamMax'] );
5180                } else {
5181                        update_post_meta( $postID, 'vw_multicam', '0' );
5182                }
5183
5184                // transcode
5185                if ( self::inList( $userkeys, $options['transcode'] ) ) {
5186                        update_post_meta( $postID, 'vw_transcode', '1' );
5187                } else {
5188                        update_post_meta( $postID, 'vw_transcode', '0' );
5189                }
5190
5191                // logoHide
5192                if ( self::inList( $userkeys, $options['logoHide'] ) ) {
5193                        update_post_meta( $postID, 'vw_logo', 'hide' );
5194                } else {
5195                        update_post_meta( $postID, 'vw_logo', 'global' );
5196                }
5197 */
```

```
Line
5198                if ( $options['videosharevod'] ) {
5199                        if ( self::inList( $userkeys, $options['videos'] ) ) {
5200                                update_post_meta( $postID, 'vw_videos', '1' );
5201                        } else {
5202                                update_post_meta( $postID, 'vw_videos', '0' );
5203                        }
5204                }
5205
5206                if ( $options['picturegallery'] ) {
5207                        if ( self::inList( $userkeys, $options['pictures'] ) ) {
5208                                update_post_meta( $postID, 'vw_pictures', '1' );
5209                        } else {
5210                                update_post_meta( $postID, 'vw_pictures', '0' );
5211                        }
5212                }
5213
5214                                // schedulePlaylists
5215                if ( ! $options['playlists'] ||
 ! self::inList( $userkeys, $options['schedulePlaylists'] ) ) {
5216                        update_post_meta( $postID, 'vw_playlistActive', '' );
5217                }
5218
5219                                // end room update
5220
5221                                // ! Go Live
5222                if ( isset( $_POST['go-live'] ) || isset( $_GET['livemode'] ) ) {
5223                        // name fix
5224                        $webcamPost = get_post( $postID );
5225                        if ( $webcamPost->post_title != sanitize_text_field( $webcamPost-
>post_title ) ) {
5226                                wp_update_post(
5227                                        array(
5228                                                'ID'         => $postID,
5229                                                'post_title' => sanitize_text_field( $webcamPost-
>post_title ),
5230                                        )
5231                                );
5232                        }
5233
5234                        // set current as room performer
5235                        $performerName = self::performerName( $current_user, $options );
5236
5237                        update_post_meta( $postID, 'performer', $performerName );
5238                        update_post_meta( $postID, 'performerUserID', $current_user->ID );
5239
5240                        update_post_meta( $postID, 'sessionStart', time() );
5241
5242                        self::roomStatsUpdate($postID);
5243
5244                        $link = get_permalink( $postID );
5245
5246                        $archive = 0;
5247
5248                        if ($options['debugMode']) $htmlCode .= '<BR>' . __( 'Room Name', 'ppv-live-
webcams' ) . ': ' . $webcamPost->post_title;
5249                        if ($options['debugMode']) $htmlCode .= '<BR>' . __( 'Room ID', 'ppv-live-
webcams' ) . ': ' . $postID;
5250
5251                        if ( $_GET['livemode'] ?? false ) {
5252                                $mode = sanitize_text_field( $_GET['livemode'] );
5253                        } else {
5254                                $mode = sanitize_text_field( $_POST['groupMode'] ?? '' );
5255                        }
```

```
5256
5257                        $modeRoomInterface = '';
5258
5259                        // mode & parameters
5260                        if ( is_array( $options['groupModes'] ) && $mode ) {
5261                                foreach ( $options['groupModes'] as $groupMode => $modeParameters ) {
5262
5263
5264                                        if ( $mode == $groupMode ) {
5265                                                // 0 if not set
5266                                        $CPMg = 0;
5267
5268                                                if ( $modeParameters['cpm'] ) {
5269
$CPMg = get_post_meta( $postID, 'vw_costPerMinuteGroup', true );
5270                                                        if ( $CPMg ) {
5271                                                                $modeParameters['cpm'] = $CPMg;
5272                                                        } else {
5273                                                                $CPMg = $modeParameters['cpm'];
5274                                                        }
5275                                                }
5276
5277                                                // custom 2 way slots if configured
5278
$slots2way = get_post_meta( $postID, 'vw_slots2way', true );
5279                                                if ( $slots2way ) {
5280                                                        $modeParameters['2way'] = $slots2way;
5281                                                }
5282
5283
if ($options['debugMode']) $htmlCode .= '<pre>' . print_r($modeParameters, true) . '</pre>';
5284
5285
if ($options['debugMode']) $htmlCode .= '<BR>' . __( 'Setting webcam group mode to', 'ppv-live-
webcams' ) . ': ' . $groupMode;
5286
update_post_meta( $postID, 'groupCPM', floatval( $CPMg ) );
5287                                                update_post_meta( $postID, 'groupMode', $groupMode );
5288
update_post_meta( $postID, 'groupParameters', $modeParameters );
5289
5290
foreach ( array( 'interface', 'roomInterface' ) as $interfaceKey ) {
5291
if ( empty( $modeParameters[ $interfaceKey ] ) ) {
5292                                                        continue;
5293                                                }
5294
5295
$interfaceCandidate = sanitize_key( $modeParameters[ $interfaceKey ] );
5296
if ( in_array( $interfaceCandidate, array( 'html5app', 'html5', 'webcamapp', 'flash' ), true ) ) {
5297                                                        $modeRoomInterface = $interfaceCandidate;
5298                                                        break;
5299                                                }
5300                                        }
5301
5302                                        //mode parameters on login
5303
foreach ( array( 'room_random', 'requests_disable', 'room_private', 'calls_only', 'group_disabled', 'room
_slots', 'room_conference', 'conference_auto', 'room_audio', 'room_text', 'vw_presentationMode', 'h5v_aud
io', 'party', 'party_reserved', 'stream_record', 'stream_record_all', 'stream_record_private' ) as $meta
) {
5304
if ( array_key_exists( $meta, $modeParameters ) ) {
```

```
Line
5305 update_post_meta( $postID, $meta, $modeParameters[ $meta ] );
5306 if ($options['debugMode']) $htmlCode .= '<BR> - ' . $meta . ': ' . $modeParameters[ $meta ];
5307                                                 }
5308                                         }
5309
5310                                         if ( $modeParameters['archive'] ?? false ) {
5311                                                 $archive = 1;
5312                                         }
5313
5314                                         if ( $modeParameters['archiveImport'] ?? false ) {
5315                                                 $archiveImport = 1;
5316                                         } else {
5317                                                 $archiveImport = 0;
5318                                         }
5319                                 }
5320                         }
5321                 } else {
5322                         update_post_meta( $postID, 'groupCPM', 0 );
5323                         update_post_meta( $postID, 'groupMode', $mode ? $mode : __( 'Free', 'ppv-live-webcams' ) );
5324                 }
5325
5326                 // room interface: room mode preset takes priority when provided.
5327                 $roomInterface = $modeRoomInterface ? $modeRoomInterface : 'html5app';
5328
5329                 if ( ! $modeRoomInterface && $options['webrtc'] ) {
5330                         $agent   = $_SERVER['HTTP_USER_AGENT'];
5331                         $Android = stripos( $agent, 'Android' );
5332                         $iOS     = ( strstr( $agent, 'iPhone' ) || strstr( $agent, 'iPod' ) || strstr( $agent, 'iPad' ) );
5333                         if ( $iOS || $Android ) {
5334                                 $roomInterface = 'html5app'; // on mobiles only html5/html5app available
5335                         }
5336                 }
5337
5338                 // If mode preset does not set interface, allow manual selection.
5339                 if ( ! $modeRoomInterface && isset( $_POST['roomInterface'] ) ) {
5340                         $roomInterface = sanitize_text_field( $_POST['roomInterface'] );
5341                 }
5342
5343                 // set
5344                 update_post_meta( $postID, 'roomInterface', $roomInterface );
5345
5346                 // go-live room option defaults
5347
5348                 if ( is_array( $options['appSetup'] ) ) {
5349                         if ( array_key_exists( 'Room', $options['appSetup'] ) ) {
5350                                 if ( is_array( $options['appSetup']['Room'] ) ) {
5351                                         foreach ( $options['appSetup']['Room'] as $key => $value ) {
5352 $optionCurrent = get_post_meta( $postID, $key, true );
5353
5354                                                 if ( empty( $optionCurrent ) || $options['appOptionsReset'] ) {
5355 update_post_meta( $postID, $key, $value );
5356
5357 if ($options['debugMode']) $htmlCode .= '<BR>' . $key . ': ' . $value;
5358
```

```
5359                                                    }
5360                                            }
5361                                    }
5362                            }
5363                    }
5364
5365                                    // import previous archives to prevent confusion
5366                                    self::importArchives( $postID );
5367
5368                            /*
5369
5370                    // archiving on streaming server discontinued
5371                    if ( $archive ) {
5372                                            update_post_meta( $postID, 'rtmp_server',
$options['rtmp_server_archive'] );
5373
5374                                    if ( $archiveImport ) {
5375                                    // update info about archived sessions for automated import
5376                                    $archivedSessions = get_post_meta( $postID, 'archivedSessions',
true );
5377                                    if ( ! $archivedSessions ) {
5378                                            $archivedSessions = array();
5379                                    }
5380                                    $archivedSession    = array(
5381                                            'performer'    => $performerName,
5382                                            'sessionStart' => time(),
5383                                            'groupMode'    => $mode,
5384                                    );
5385                                    $archivedSessions[] = $archivedSession;
5386                                    update_post_meta( $postID, 'archivedSessions', $archivedSessions
);
5387                                    }
5388                    } else {
5389                            update_post_meta( $postID, 'rtmp_server', $options['rtmp_server'] );
5390                    }*/
5391
5392                    // remove previous archive server
5393
if ( get_post_meta($postID, 'rtmp_server', true) == $options['rtmp_server'] ) delete_post_meta( $postID,
'rtmp_server' );
5394
5395                    // Checkins
5396                    if ( $options['checkins'] ) {
5397
5398
$selectedPerformers = isset( $_POST['selectedPerformers'] ) ? (array) $_POST['selectedPerformers'] : [];
5399                                    if ( count( $selectedPerformers ) ) {
5400                                            foreach ( $selectedPerformers as $key => $performerID ) {
5401
$selectedPerformers[ sanitize_text_field( $key ) ] = intval( $performerID );
5402                                    }
5403                                    } else $selectedPerformers = []; //for current performer
5404
5405                                    if ( $performersCSV = sanitize_text_field( $_POST['performersCSV'] ) ) {
5406
5407                                            $performers = explode( ',', $performersCSV );
5408                                            if ( count( $performers ) ) {
5409                                                    foreach ( $performers as $performerV ) {
5410
$performerID = '';
5411                                                                                                    $value
 = trim( $performerV );
5412
5413                                                                                    if ( is_email( $value ) ) {
```

```
5414  $performerID = get_user_by( 'email', $value );
5415                                                  } else {
5416  $performerID = get_user_by( 'login', $value );
5417                                                  }
5418
5419                                                  if ( $performerID ) {
5420  if ( ! in_array( $performerID, $selectedPerformers ) ) {
5421  $selectedPerformers[] = $performerID;
5422                                                          }
5423                                                  }
5424
5425                                                  if ( $value && ! $performerID ) {
5426                                                          $htmlCode .= '<BR>' . __( 'Warning:
Performer', 'ppv-live-webcams' ) . ' "' . $value . '" ' . __( 'not found: use a valid username (login) or
email!', 'ppv-live-webcams' ) . '<BR>';
5427                                                  }
5428                                              }
5429                                          }
5430                                  }
5431                          } else $selectedPerformers = [];
5432                          // always checkin current performer
5433                          if ( ! in_array( $current_user->ID, $selectedPerformers ) ) {
5434                                  $selectedPerformers[] = $current_user->ID;
5435                          }
5436                          update_post_meta( $postID, 'checkin', $selectedPerformers );
5437
5438                          //room update when performer enters again, for users to reload
5439                          update_post_meta( $postID, 'updated_room', time() );
5440
5441                          if (!isset($data['status'])) $data['status'] = 'off';
5442                          $toyStatus = $data['status'] == 'on' ? 1 : 0;
5443                          update_post_meta( $postID, 'lovenseToy', $toyStatus );
5444
5445
5446                          // if ($roomInterface != 'html5app' )
5447                          if ($options['debugMode']) $htmlCode .= '<BR>' . __( 'Room Interface', 'ppv-live-
webcams' ) . ': ' . $roomInterface;
5448
5449                          if ($options['debugMode'] &&  $options['checkins']) {
5450                                  $htmlCode .= '<BR>' . __( 'Checked in performers', 'ppv-live-
webcams' ) . ': ' . count( $selectedPerformers );
5451                          }
5452
5453  if ($options['debugMode']) $htmlCode .= '<BR>' . __( 'Assigning you', 'ppv-live-webcams' ) . '
(' . $performerName . ') ' . __( 'as room performer (streamer) and redirecting to webcam room page.
Please wait...', 'ppv-live-webcams' ) . '';
5454
5455                                                  // notify performer is live
5456  self::notifyLive( $current_user, $postID, $options );
5457
5458                                                  $htmlCode .= '<PRE style="display: none">
5459                                                  <SCRIPT>window.location=\'' . $link . '\';
</SCRIPT>
5460                                                  </PRE>
5461                                  <p><a class="ui button" href="' . $link . '"><b>' . __( 'Click
here', 'ppv-live-webcams' ) . '</b> ' . __( 'to access your webcam room if you do not get automatically
redirected', 'ppv-live-webcams' ) . '</a></p>';
5462
5463                                                  return $htmlCode;
```

```
5464                }
5465
5466                // ! Current Webcam
5467                $webcamPost   = get_post( $postID );
5468                $hideRoomFeatures = false;
5469
5470                if ( !$webcamPost ) {
5471                        $htmlCode .= '<div class="ui inverted red segment">ERROR: "Current
' . $options['custom_post'] . '" post not found: #' . $postID . '.';
5472
5473                        $args = array(
5474                                'post_type'      => sanitize_text_field( $options['custom_post'] ),
5475                                'post_status'    => 'publish',
5476                                'post_author'    => $current_user->ID,
5477                                'posts_per_page' => 1,
5478                        );
5479                        $rooms = get_posts( $args );
5480                        if ( count( $rooms ) ) {
5481                                $webcamPost = $rooms[0];
5482                                update_user_meta( $current_user->ID, 'currentWebcam', $webcamPost->ID );
5483                                $htmlCode .= '</br>A room was found and set as current: #' . $webcamPost-
>ID;
5484                        }
5485
5486                        if ( $options['performerWebcams'] )
5487                        $htmlCode .= '<br><a class="ui icon right labeled button"
href="' . add_query_arg( array( 'changeWebcam' => 1 ), $this_page ) . '"> <i class="sync icon"></i>
' . __( 'My Rooms', 'ppv-live-webcams' ) . '</a> ';
5488
5489                        $htmlCode .= '</br>If issue persists, contact administrator to troubleshoot!
</div>';
5490
5491                        $hideRoomFeatures = true;
5492                        return $htmlCode;
5493                }
5494
5495                $htmlCode .= '<div class="ui ' . $options['interfaceClass'] . ' segment"><h4 class="ui
' . $options['interfaceClass'] . ' header"> <i class="user circle icon">
</i>' . ucwords($options['custom_post']). ' '. __( 'Room', 'ppv-live-webcams' ) . ': ' . $webcamPost-
>post_title . '</h4>';
5496
5497                if ( $webcamPost->post_status != 'publish' ) {
5498                        $htmlCode .= '<div class="ui inverted red segment">ERROR: Webcam room post is not
accessible. Status is "' . $webcamPost->post_status . '" and should be "publish". Contact administrator
to publish room or select a different room!</div>';
5499                }
5500
5501                if ( $webcamPost->post_type != $options['custom_post'] ) {
5502
5503                        $warnCode = '';
5504                        if ( $options['performerWebcams'] ) {
5505                                $warnCode .= '<br><a class="ui icon right labeled button"
href="' . add_query_arg( array( 'changeWebcam' => 1 ), $this_page ) . '"> <i class="sync icon"></i>
' . __( 'My Rooms', 'ppv-live-webcams' ) . '</a> ';
5506                        }
5507
5508                        $htmlCode .= '<div class="ui inverted red segment">ERROR: Currently selected
webcam room post is not of correct type. Type is "' . $webcamPost->post_type . '" and should be
"' . $options['custom_post'] . '". As administrator changed room types, select a different room!
' . $warnCode . '</div>';
5509
5510                        $hideRoomFeatures = true;
5511                }
5512
5513                // ! form: group mode
```

```
Line
5514                $CPM     = get_post_meta( $postID, 'vw_costPerMinute', true );
5515                $CPMg    = get_post_meta( $postID, 'vw_costPerMinuteGroup', true );
5516                $cpmCode = '';
5517                // if ($CPM) $cpmCode.=  __('Custom cost per minute in private show:', 'ppv-live-webcams') . ' '. $CPM . ' ';
5518                if ( $CPMg ) {
5519                        $cpmCode .= '<br>* ' . __( 'Custom cost per minute in paid group chat:', 'ppv-live-webcams' ) . ' ' . $CPMg . ' ' . htmlspecialchars( $options['currency'] );
5520                }
5521
5522                if ( $options['performerGolive'] == 'buttons' ) {
5523
5524                        if ( is_array( $options['groupModes'] ) ) {
5525                                $groupModeCrt = get_post_meta( $postID, 'groupMode', true );
5526
5527                                $htmlCode .= '<table class="ui very basic table"><tbody>';
5528
5529                                foreach ( $options['groupModes'] as $groupMode => $modeParameters ) {
5530                                        $htmlCode .= '<tr><td>';
5531                                        $htmlCode .= '<a href="' . esc_url( add_query_arg( 'livemode', $groupMode, $this_page ) ) . '" class="ui icon right labeled button fluid" > ' . $groupMode . ' <i class="play icon"></i></a> </td><td> <small>' . stripslashes( $modeParameters['description'] ) . '</small> ';
5532                                        if ( $modeParameters['cpm'] ) {
5533                                                $htmlCode .= '<small> ' . __( 'Paid Mode', 'ppv-live-webcams' ) . ': ' . $modeParameters['cpm'] . htmlspecialchars( $options['currencypm'] ) . '</small>';
5534                                        }
5535                                        $htmlCode .= '</td></tr>';
5536                                }
5537
5538                                $htmlCode .= '</tbody></table>';
5539                        }
5540
5541                        if ( self::is_true( get_post_meta( $postID, 'room_private', true ) ) ) {
5542                                $htmlCode .= '<div>* ' . __( 'This room is currently private, hidden from room list.', 'ppv-live-webcams' ) . '</div>';
5543                        }
5544
5545                        if ( $options['performerWebcams'] ) {
5546                                $htmlCode .= '<hr class="ui divider"/> <a class="ui icon right labeled button" href="' . add_query_arg( array( 'changeWebcam' => 1 ), $this_page ) . '"> <i class="sync icon"> </i> ' . __( 'My Rooms', 'ppv-live-webcams' ) . '</a> ' . __( 'Switch to different room / listing. Your account can setup and manage multiple room profiles.', 'ppv-live-webcams' );
5547                        }
5548                } else // chat modes in advanced form
5549                        {
5550
5551                        $webcamCode = '';
5552                        if ( $options['performerWebcams'] ) {
5553                                $webcamCode .= '<hr class="ui divider"/> <a class="ui icon right labeled button" href="' . add_query_arg( array( 'changeWebcam' => 1 ), $this_page ) . '"> <i class="sync icon"> </i> ' . __( 'My Rooms', 'ppv-live-webcams' ) . '</a> ' . __( 'Switch to different room / listing. Your account can setup and manage multiple room profiles.', 'ppv-live-webcams' );
5554                        }
5555
5556                        // groupModes
5557                        $modeCode = '';
5558                        if ( is_array( $options['groupModes'] ) ) {
5559                                $groupModeCrt = get_post_meta( $postID, 'groupMode', true );
5560
5561                                $modeCode   .= '<div id="groupModeSelect"><SELECT class="ui dropdown v-select" name="groupMode" id="groupMode">';
5562                                $modeDetails = '';
5563
5564                                foreach ( $options['groupModes'] as $groupMode => $modeParameters ) {
```

```
Line
5565                                    /*
5566                                    if ($modeParameters['cpm'])
5567                                            if ($CPMg) $costCode = ' *' . $CPMg .
htmlspecialchars($options['currencypm']) ;
5568                                            else $costCode = ' ' . $modeParameters['cpm'] .
htmlspecialchars($options['currencypm']);
5569                                            else $costCode = '';
5570                                            */
5571                                    $modeCode .= '<OPTION value="' . $groupMode . '"
' . ( $groupModeCrt == $groupMode ? 'selected' : '' ) . '>' . $groupMode . '</OPTION>';
5572
5573                                    if ( $modeParameters['description'] || $modeParameters['cpm'] ) {
5574                                            $modeDetails .= $modeDetails ? '<br>' : '';
5575
5576                                            $modeDetails .= '<b>' . $groupMode . '</b>' . ':
<small>' . stripslashes( $modeParameters['description'] ) . '</small>';
5577                                            if ( $modeParameters['cpm'] ?? false ) {
5578                                                    $modeDetails .= '<small> ' . __( 'Paid
Mode', 'ppv-live-webcams' ) . ':
' . $modeParameters['cpm'] . htmlspecialchars( $options['currencypm'] ) . '</small>';
5579                                            }
5580                                    }
5581                            }
5582                            $modeCode .= '</SELECT> ' . __( 'Room Mode', 'ppv-live-webcams' ) . ':
' . __( 'Select group mode presets for your new session in this room, before going live.', 'ppv-live-
webcams' ) . '</div>';
5583                    } else {
5584                            $htmlCode .= 'No chat modes defined.';
5585                    }
5586
5587
if ( ! $options['presentation'] && self::collaborationMode( $postID, $options ) ) {
5588                            $modeDetails .= '<br>* ' . __( 'This room is currently in
presentation/collaboration mode.', 'ppv-live-webcams' );
5589                    }
5590
5591                    if ( self::is_true( get_post_meta( $postID, 'room_private', true ) ) ) {
5592                            $modeDetails .= '<br>* ' . __( 'This room is currently private, hidden
from room list.', 'ppv-live-webcams' );
5593                    }
5594
5595                    // style
5596                    $modeDetails = $modeDetails ? '<div class="ui small basic
' . $options['interfaceClass'] . ' segment">' . $modeDetails . '</div>' : '';
5597
5598                    // ! interface select
5599                    $interfaceCode = '';
5600                    $interfaceInfo = '';
5601
5602                    $roomInterface = get_post_meta( $postID, 'roomInterface', true );
5603                    if ( $options['webrtc'] ) {
5604                            if ( ! $roomInterface ) {
5605                                    $roomInterface = 'html5app';
5606                            }
5607
5608                            $agent   = $_SERVER['HTTP_USER_AGENT'];
5609                            $Android = stripos( $agent, 'Android' );
5610                            $iOS     = ( strstr( $agent, 'iPhone' ) || strstr( $agent, 'iPod' ) ||
 strstr( $agent, 'iPad' ) );
5611
5612                            $interfaceCode .= '<div id="roomInterfaceSelect"><SELECT class="ui
dropdown v-select" name="roomInterface" id="roomInterface">';
5613
5614                            // html5 videochat app
```

```
5615                                $interfaceCode .= '<OPTION value="html5app"
' . ( $roomInterface == 'html5app' ? 'selected' : '' ) . '>' . __( 'HTML5', 'ppv-live-webcams' ) . ' -
' . __( 'Recommended', 'ppv-live-webcams' ) . '</OPTION>';
5616                                $interfaceInfo .= ' * ' . __( 'HTML5 Videochat works in mobile and PC
browsers, providing streaming with group chat, private 2 way video calls, tips, conferencing,
collaboration, external broadcasts. Recommended.', 'ppv-live-webcams' );
5617
5618                                // html5 live streaming
5619                                /*
5620                                if ($options['webrtc'] < 6)
5621                                {
5622                                $interfaceCode .= '<OPTION value="html5" ' .
($roomInterface=='html5'?'selected':'') . '>' . __('HTML5 Streaming', 'ppv-live-webcams') . '</OPTION>';
5623                                $interfaceInfo .= '<br>HTML5 Streaming interface provides simple 1 way
video streaming with group chat and tips.';
5624                                }
5625                                */
5626                        }
5627
5628
5629                        $interfaceCode .= '</SELECT> Room Interface: ' . __( 'Select room interface
before going live. Also applies to clients.', 'ppv-live-webcams' );
5630                        $interfaceCode .= '<div class="ui small basic ' . $options['interfaceClass'] . '
segment">' . $interfaceInfo . '</div>';
5631                        $interfaceCode .= '</div>';
5632
5633                        // hide if html5app only
5634
5635                        if ( $options['webrtc'] >= 5 ) {
5636                                $interfaceCode = '<input type="hidden" name="roomInterface"
id="roomInterface" value="html5app">';
5637                                $interfaceInfo = '';
5638                        }
5639
5640                        // ! form: checkin
5641                        $checkinCode = '';
5642                        if ( $options['checkins'] ) {
5643                                $checkinCode .= '<A class="ui icon right labeled button" id="hideshow"
>' . __( 'CheckIn Options', 'ppv-live-webcams' ) . ' <i class="down arrow icon"></i> </a> ' . __( 'Tag
other performers present in session.', 'ppv-live-webcams' );
5644
5645                                $checkinCode .= '<div id="checkinPerformers"><div class="ui
' . $options['interfaceClass'] . ' segment">
5646 <H4 class="ui header">' . __( 'Checkin Performers', 'ppv-live-webcams' ) . '</H4>' . __( 'You can select
partner performers or specify other performers that are present in session.', 'ppv-live-webcams' ) . '
5647 <BR>' . __( 'If paid group chat is enabled, earnings will be shared equally between checked in
performers.', 'ppv-live-webcams' );
5648
5649                                // partner performers
5650                                $performerIDs = get_post_meta( $postID, 'performerID', false );
5651                                if ( $performerIDs ) {
5652                                        if ( count( $performerIDs ) ) {
5653                                                $checkinCode .= '<BR> <label>' . __( 'Partner
Performers', 'ppv-live-webcams' ) . '</label>';
5654                                                foreach ( $performerIDs as $performerID ) {
5655                                                        $performer = get_user_by( 'id', $performerID );
5656                                                        if ( $performer ) {
5657                                                                $checkinCode .= '<input type="checkbox"
name="selectedPerformers[]" value="' . $performer->ID . '">' . $performer->user_login . '<br>';
5658                                                        }
5659                                                }
5660                                        }
5661                                }
5662
5663                                $checkinCode .= '<BR> <label>' . __( 'Checkin Performers', 'ppv-live-
webcams' ) . '</label> <input size="48" type="text" name="performersCSV" id="performersCSV" value="">
```

```
Line
    <BR>' . __( 'Comma separated list of performer usernames or emails.', 'ppv-live-webcams' ) . '';
5664
5665                                $checkinCode .= '</div></div>';
5666
5667                                $checkinCode .= "<PRE style='display: none'>
<SCRIPT>jQuery(document).ready(function(){
5668
5669                jQuery('#checkinPerformers').toggle('fast');
5670
5671        jQuery('#hideshow').on('click', function(event) {
5672             jQuery('#checkinPerformers').toggle('slow');
5673                         jQuery(\".ui.dropdown:not(.multi,.fpsDropdown)\").dropdown();
5674
5675        });
5676    });</SCRIPT></PRE>";
5677                        }
5678
5679                        $htmlCode .= '
5680<form method="post" enctype="multipart/form-data" action="' . $this_page . '" name="goLiveForm" class="">
5681<div class="ui form">
5682<div class="ui field"><button class="ui icon right labeled button big green" type="submit" name="go-live"
id="go-live" value="go-live" > ' . __( 'Go Live', 'ppv-live-webcams' ) . ' <i class="play icon"></i>
</button> </div>
5683<div class="ui field">' . $modeCode . $modeDetails . '</div>
5684<div class="ui field">' . $interfaceCode . '</div>
5685<div class="ui field">' . $checkinCode . '</div>
5686<div class="ui field">' . $webcamCode . '</div>
5687</div>
5688</form>
5689<br style="clear:both">
5690';
5691
5692                }
5693
if ( ! get_post_meta( $postID, 'hasSnapshot', true ) && ! get_post_meta( $postID, 'hasPicture', true ) )
{
5694                        $htmlCode .= '<div class="ui segment secondary">' . __( 'This room can not
currently show in room list as it does not have a snapshot or picture. Rooms show in list after first
broadcast only with a listing image. Broadcast camera for about a minute to generate a snapshot or upload
a room picture!', 'ppv-live-webcams' ) . '</div>';
5695                }
5696
5697                $htmlCode .= '</div>
5698                        ';
5699
5700                if ( $hideRoomFeatures ) {
5701                        return $htmlCode;
5702                }
5703
5704                // ! Dashboard Tabs
5705                // $htmlCode .= '<div class="vwtabs">';
5706
5707                // tab header
5708
5709                /*
5710                        $headerCode .= '<div class="ui ' . $options['interfaceClass'] . ' top attached
tabular menu">';
5711
5712
5713                        $htmlCode .= '
5714                <script>
5715                jQuery(document).ready(function(){
5716
5717                var vwTabs = jQuery(".tabular.menu .item");
```

```
Line
5718                try{ vwTabs.tab()} catch(error)
5719                {
5720                console.log("Interface error Tabs", error, vwTabs);
5721                }
5722                });
5723                </script>';
5724
5725                */
5726                $active  = '';
5727                $header  = array();
5728                $content = array();
5729
5730                if ( $options['calls'] ) {
5731                        /*
5732                        //! calls tab
5733                        $checked = '';
5734                        if ($_GET['calls']) $checked = 'active';
5735                        if ($checked) $checked1 = true;
5736
5737                        $headerCode .= '<a class="item ' . $checked .'" data-tab="calls">' . __('Calls', 'ppv-live-webcams') . '</a>';
5738
5739                        $contentCode .= '<div class="ui ' . $options['interfaceClass'] . ' bottom attached tab segment ' . $checked .'" data-tab="calls">';
5740
5741                        $contentCode .=  '<h3 class="ui header">' . __('Locked Private Calls', 'ppv-live-webcams') . '</H3>';
5742
5743                        $contentCode .= do_shortcode('[videowhisper_cam_calls post_id="' . $postID . '"]');
5744
5745                        $contentCode .= '<br style="clear:both"></div>';
5746                        */
5747
5748                        if ( isset( $_GET['calls'] ) || isset( $_GET['deletecall'] ) ) {
5749                                $active = 'calls';
5750                        }
5751
5752                        $header['calls'] = __( 'Calls', 'ppv-live-webcams' );
5753
5754                        $content['calls']  = '<h3 class="ui header ' . $options['interfaceClass'] .'">' . __( 'Locked Private Calls', 'ppv-live-webcams' ) . '</H3>';
5755                        $content['calls'] .= do_shortcode( '[videowhisper_cam_calls post_id="' . $postID . '"]' );
5756                }
5757
5758                // ! messages tab
5759
5760                if ( $options['messages'] ?? false ) {
5761                        $checked = '';
5762                        if ( $_GET['messages'] ?? false ) {
5763                                $checked = 'active';
5764                        }
5765                        if ( $checked ) {
5766                                $checked1 = true;
5767                        }
5768
5769                        global $wpdb;
5770                        $table_messages = $wpdb->prefix . 'vw_vmls_messages';
5771                        $sqlC           = $wpdb->prepare("SELECT COUNT(id) FROM `$table_messages` WHERE webcam_id=%d AND reply_id=%d AND ldate=%d", $postID, 0, 0); // for webcam, only questions (no replies), not replied (ldate)
5772                        $unrepliedCount = $wpdb->get_var($sqlC);
```

```
5773
5774                        /*
5775                                $headerCode .= '<a class="item ' . $checked .'" data-tab="messages">' .
__('Messages', 'ppv-live-webcams') . ($unrepliedCount?'<div class="ui green left pointing
label">'.$unrepliedCount.'</div>':'') .'</a>';
5776
5777                                $contentCode .= '<div class="ui ' . $options['interfaceClass'] . ' bottom
attached tab segment ' . $checked .'" data-tab="messages">';
5778
5779                                $contentCode .= '<h3 class="ui header">' . __('Paid Questions and
Messages', 'ppv-live-webcams') . '</H3>';
5780
5781                                $contentCode .= do_shortcode('[videowhisper_cam_messages_performer
post_id="' . $postID . '"]');
5782
5783                                $contentCode .= '<br style="clear:both"></div>';
5784                                */
5785
5786                        if ( $_GET['messages'] ?? false ) {
5787                                $active = 'messages';
5788                        }
5789
5790                        $header['messages'] = __( 'Messages', 'ppv-live-
webcams' ) . ( $unrepliedCount ? '<div class="ui green left pointing
label">' . $unrepliedCount . '</div>' : '' );
5791
5792                        $content['messages']  = '<h3 class="ui header ' . $options['interfaceClass'] . '
">' . __( 'Paid Questions and Messages', 'ppv-live-webcams' ) . '</H3>';
5793                        $content['messages'] .= do_shortcode( '[videowhisper_cam_messages_performer
post_id="' . $postID . '"]' );
5794
5795                        if ($options['sms_number']) $content['messages']
.= do_shortcode( '[videowhisper_sms_number]' );
5796
5797                        // $content['messages'] .= '<p>' . __('Clients can send paid questions.', 'ppv-
live-webcams') . '</p>';
5798                }
5799
5800                /*
5801                        $htmlCode .= '<div class="vwtab">
5802                <input type="radio" id="tab-2" name="tab-group-1" '.$checked.'>
5803                <label class="vwlabel" for="tab-2">' . __('Setup', 'ppv-live-webcams') . '</label>
5804
5805                <div class="vwpanel">
5806                <div class="vwcontent">';
5807                */
5808
5809                // ! setup tab
5810
5811                /*
5812                        $checked = '';
5813                        if ($_POST['setup']) $checked = 'active';
5814                        if ($checked) $checked1 = true;
5815
5816                        $headerCode .= '<a class="item ' . $checked .'" data-tab="setup">' . __('Setup',
'ppv-live-webcams') . '</a>';
5817
5818                        $contentCode .= '<div class="ui ' . $options['interfaceClass'] . ' bottom
attached tab segment ' . $checked .'" data-tab="setup">';
5819                */
5820
5821                if ( $options['performerSetup'] ?? false ) {
5822
5823                        if ( $_POST['setup'] ?? false ) {
5824                                $active = 'setup';
```

```
Line
5825                        }
5826
5827                        $header['setup'] = __( 'Setup', 'ppv-live-webcams' );
5828
5829                        $featuresCode = '';
5830
5831
5832                                                $featuresCode .= '<tr><th colspan="2">' . __( 'Room
Access & Monetization', 'ppv-live-webcams' ) . '</td></tr>';
5833
5834
5835                        // costPerMinute
5836                        if ( self::inList( $userkeys, $options['costPerMinute'] ) )
5837                        {
5838
5839
if ( $postID ) $value = get_post_meta( $postID, 'vw_costPerMinute', true );
5840
5841                                if ( $value == '' ) {
5842                                        $value = $options['ppvPPM'];
5843                                }
5844
5845                                if ( $value < $options['ppvPPMmin'] ) {
5846                                        $value = $options['ppvPPMmin'];
5847                                }
5848                                if ( $options['ppvPPMmax'] ) {
5849                                        if ( $value > $options['ppvPPMmax'] ) {
5850                                                $value = $options['ppvPPMmax'];
5851                                        }
5852                                }
5853
5854                                        $cpmRange = '' . __( 'Default', 'ppv-live-webcams' ) . ':
' . $options['ppvPPM'] . ' Min: ' . $options['ppvPPMmin'] . ' Max: ' . $options['ppvPPMmax'];
5855
5856                                $featuresCode .= '<tr><td>' . __( 'Cost Per Minute', 'ppv-live-
webcams' ) . '</td><td><div class="ui right labeled input"><input size=5 name="costPerMinute"
id="costPerMinute" value="' . $value . '"><div class="ui  label">' . esc_html( $options['currency'] ) . '
</div></div><BR>' . __( 'Cost per minute for private shows (set 0 to use default).', 'ppv-live-
webcams' ) . ' ' . $cpmRange . '</td></tr>';
5857
5858
5859                                if ( $options['modeAudio'] )
5860                                {
5861
5862
if ( $postID ) $value = get_post_meta( $postID, 'vw_costPerMinuteaudio', true );
5863
5864                                if ( $value == '' ) {
5865                                        $value = $options['ppvPPMaudio'];
5866                                }
5867
5868                                if ( $value < $options['ppvPPMmin'] ) {
5869                                        $value = $options['ppvPPMmin'];
5870                                }
5871                                if ( $options['ppvPPMmax'] ) {
5872                                        if ( $value > $options['ppvPPMmax'] ) {
5873                                                $value = $options['ppvPPMmax'];
5874                                        }
5875                                }
5876
5877                                        $featuresCode .= '<tr><td>' . __( 'Cost Per Minute', 'ppv-live-
webcams' ) . ': ' . __( 'Audio Only', 'ppv-live-webcams' ) . '</td><td><div class="ui right labeled
input"><input size=5 name="costPerMinuteAudio" id="costPerMinuteAudio" value="' . $value . '"><div
```

```
class="ui  label">' . esc_html( $options['currency'] ) . ' </div></div><BR>' . ' ' . $cpmRange . '</td></tr>';
                                }

                                if ( $options['modeText'] )
                                {

if ( $postID ) $value = get_post_meta( $postID, 'vw_costPerMinutetext', true );

                                if ( $value == '' ) {
                                        $value = $options['ppvPPMtext'];
                                }

                                if ( $value < $options['ppvPPMmin'] ) {
                                        $value = $options['ppvPPMmin'];
                                }
                                if ( $options['ppvPPMmax'] ) {
                                        if ( $value > $options['ppvPPMmax'] ) {
                                                $value = $options['ppvPPMmax'];
                                        }
                                }

                                        $featuresCode .= '<tr><td>' . __( 'Cost Per Minute', 'ppv-live-webcams' ) . ': ' . __( 'Text Only', 'ppv-live-webcams' ) . '</td><td><div class="ui right labeled input"><input size=5 name="costPerMinuteText" id="costPerMinuteText" value="' . $value . '"><div class="ui  label">' . esc_html( $options['currency'] ) . ' </div></div><BR>' . ' ' . $cpmRange . '</td></tr>';
                                        }

                        }

                        // costPerMinuteGroup
                        if ( self::inList( $userkeys, $options['costPerMinuteGroup'] ) ) {
                                if ( $postID ) {
                                        $value = get_post_meta( $postID, 'vw_costPerMinuteGroup', true );
                                }
                                if ( $value == '' ) {
                                        $value = $options['ppvPPM'];
                                }

                                if ( $value < $options['ppvPPMmin'] ) {
                                        $value = $options['ppvPPMmin'];
                                }
                                if ( $options['ppvPPMmax'] ) {
                                        if ( $value > $options['ppvPPMmax'] ) {
                                                $value = $options['ppvPPMmax'];
                                        }
                                }

                                        $cpmRange = '' . __( 'Default', 'ppv-live-webcams' ) . ': ' . $options['ppvPPM'] . ' Min: ' . $options['ppvPPMmin'] . ' Max: ' . $options['ppvPPMmax'];

                                $featuresCode .= '<tr><td>' . __( 'Group Show Cost Per Minute', 'ppv-live-webcams' ) . '</td><td><div class="ui right labeled input"><input size=5 name="costPerMinuteGroup" id="costPerMinuteGroup" value="' . $value . '"><div class="ui label">' . esc_html( $options['currency'] ) . ' </div></div><BR>' . __( 'Cost per minute for group shows. Replaces paid group CPM (set 0 to use default).', 'ppv-live-webcams' ) . ' ' . $cpmRange . '</td></tr>';
                        }



                        //MicroPayment Subscriptions
```

```
Line
5930                        if ( class_exists( 'VWpaidMembership' ) )
5931                        {
5932                                $subscriptions = get_user_meta( $current_user->ID, 'vw_provider_subscriptions', true );
5933                                $content_tier = get_post_meta( $postID, 'vw_subscription_tier', true );
5934
5935                            //$client_tier         = get_user_meta( $userID, 'vw_client_subscription_' . $author->ID, true );
5936
5937                                $optionsMP = get_option( 'VWpaidMembershipOptions' );
5938                             $setupLink = ( $optionsMP['p_videowhisper_provider_subscriptions'] ? ' <a href="' . get_permalink( $optionsMP['p_videowhisper_provider_subscriptions'] ) . '"><i class="lock icon"></i>' . __( 'Setup Subscriptions', 'ppv-live-webcams' ) . '</a>' : '' );
5939
5940                                if ($subscriptions && is_array($subscriptions))
5941                                {
5942
5943                                        $featuresCode .= '<tr><td>' . __( 'Subscription Tier', 'ppv-live-webcams' ) . '</td><td><select name="subscriptionTier" id="subscriptionTier" class="ui dropdown v-select">';
5944
5945                                        $featuresCode     .= '<option value="0" ' . ( $content_tier == 0 ? 'selected' : '' ) . '>' . __('None', 'ppv-live-webcams') . '</option>';
5946
5947                                        foreach ($subscriptions as $tier => $subscription)
5948                                        $featuresCode     .= '<option value="' . $tier . '" ' . ( $content_tier == $tier ? 'selected' : '' ) . '>' . $tier . '. ' . $subscription['name'] . '</option>';
5949
5950                                        $featuresCode     .= '</select>' .  '<br>'. __('Users that buy configured subscription tier or higher can participate in paid group shows for free.', 'ppv-live-webcams') . $setupLink . '</td></tr>';
5951
5952                                }else $featuresCode .= '<tr><td>' . __( 'Subscription Tier', 'ppv-live-webcams' ) . '</td><td>'. __( 'You do not have any subscriptions setup, yet.', 'ppv-live-webcams' ) . $setupLink .   '<br>'. __('Users that buy configured subscription tier or higher can participate in group sessions for free.', 'ppv-live-webcams') . '</td></tr>';
5953
5954                        }
5955
5956                        // accessPrice
5957                        if ( self::inList( $userkeys, $options['accessPrice'] ) ) {
5958                                if ( $postID ) {
5959
5960                                //MicroPayments
5961                                if ( class_exists( 'VWpaidMembership' ) ) {
5962
5963                                        $data = \VWpaidMembership::contentData( $postID);
5964                                        $value = floatval( $data['price'] );
5965
5966                                } else {
5967
5968                                $value = get_post_meta( $postID, 'vw_accessPrice', true );
5969                                        }
5970
5971                                } else {
5972                                        $value = '';
5973                                }
5974
5975 if ( class_exists( 'VWpaidMembership' ) )
5976 {
5977                                $optionsMP = get_option( 'VWpaidMembershipOptions' );
```

```
5978                            $setupLink = '<a
href="' . add_query_arg( 'editID', intval( $postID ), get_permalink( $optionsMP['p_videowhisper_content_e
dit'] ) ) . '"><i class="edit icon"></i> ' . __( 'Setup Monetization', 'ppv-live-webcams' ) . '</a>';
5979
5980                            $featuresCode .= '<tr><td' . __('>Access Price', 'ppv-live-webcams') . '</td>
<td>' . esc_html($value) . esc_html($optionsMP['currency']) .' ' . $setupLink . '</td></tr>';
5981
5982 }
5983 else $featuresCode .= '<tr><td' . __('>Access Price', 'ppv-live-webcams') . '</td><td><input size=5
name="accessPrice" id="accessPrice" value="' . $value . '"><BR>' . __( 'Room access price. Leave 0 for
free access.', 'ppv-live-webcams' ) . '</td></tr>';
5984
5985
5986                         }
5987
5988                                 // accessList
5989                         if ( self::inList( $userkeys, $options['accessList'] ) ) {
5990                                 if ( $postID ) {
5991                                         $value = get_post_meta( $postID, 'vw_accessList', true );
5992                                 } else {
5993                                         $value = '';
5994                                 }
5995
5996                                 $featuresCode .= '<tr><td>' . __( 'Access List', 'ppv-live-
webcams' ) . '</td><td><textarea rows=2 cols="80" name="accessList"
id="accessList">' . $value . '</textarea><BR>' . __( 'User roles, logins, emails separated by comma.
Leave empty to allow everybody to access.', 'ppv-live-webcams' ) . '</td></tr>';
5997                         }
5998
5999
6000
6001                         // accessPassword
6002                         if ( self::inList( $userkeys, $options['accessPassword'] ) ) {
6003                                 $value = $webcamPost->post_password;
6004
6005                                 $featuresCode .= '<tr><td>Access Password</td><td><input size=16
name="accessPassword" id="accessPassword" value="' . $value . '"><BR>Password to protect room page
access. Leave blank to not require password.</td></tr>';
6006                         }
6007
6008                                         // banCountries
6009                         if ( self::inList( $userkeys, $options['banCountries'] ) ) {
6010                                 if ( $postID ) {
6011
$banLocations = get_post_meta( $postID, 'vw_banCountries', false );
6012                                 } else {
6013                                         $banLocations = '';
6014                                 }
6015
6016                                 $value = implode( ', ', $banLocations );
6017
6018                                 $info = '';
6019
6020                                 // $banLocations = str_getcsv($value, ',');
6021                                 $info .= ' (' . count( $banLocations ) . ') ';
6022
6023                                 $clientIP = self::get_ip_address();
6024                                 $info    .= '<br>' . __( 'Detected IP', 'ppv-live-webcams' ) . ':
' . $clientIP;
6025
6026                                 $detectedLocation = self::detectLocation( 'all', $clientIP );
6027
6028                                 $info .= '<br>' . __( 'Detected Location', 'ppv-live-webcams' ) . ':
' . ( $detectedLocation ? implode( ", ", $detectedLocation ) : __( 'Not detected.', 'ppv-live-
webcams' ) );
```

```
6029
6030                                if ( self::detectLocation() === false ) {
6031                                        $info .= '<BR>' . __( 'ERROR: GeoIP extension is required on web host for this functionality.', 'ppv-live-webcams' );
6032                                }
6033
6034                                $featuresCode .= '<tr><td>' . __( 'Ban Countries', 'ppv-live-webcams' ) . '</td><td><textarea rows="2" cols="80" name="banCountries" id="banCountries">' . htmlspecialchars( $value ) . '</textarea><BR>' . __( 'You may want to ban your location for privacy reasons. Or restrict countries and regions where your content is not legal or well received. Room will not be listed and room page will not be accessible from banned location. This functionality is not 100% accurate and may be bypassed with VPNs. Specify list of locations (country, region, city or continent code, separated by comma) that can not access webcam room. ', 'ppv-live-webcams' ) . $info . '</td></tr>';
6035
6036                        }
6037
6038
6039                        $featuresCode .= '<tr><th colspan="2">' . __( 'Room Details', 'ppv-live-webcams' ) . '</td></tr>';
6040
6041
6042                        // uploadPicture
6043                        if ( self::inList( $userkeys, $options['uploadPicture'] ) ) {
6044
6045                                $picCode      = '';
6046                                $picture      = get_post_meta( $postID, 'picture', true );
6047                                $thumbPicture = get_post_meta( $postID, 'thumbPicture', true );
6048
6049                                if ( $picture ) {
6050                                        if ( file_exists( $picture ) ) {
6051                                                $picCode .= '<br>Picture: <a href="' . self::path2url( $picture ) . '?nocache=1" target="_picture">' . basename( $picture ) . '</a> ' . self::humanSize( filesize( $picture ) );
6052                                        }
6053                                }
6054                                if ( $thumbPicture ) {
6055                                        if ( file_exists( $thumbPicture ) ) {
6056                                                $picCode .= '<br>Picture Thumb: <a href="' . self::path2url( $thumbPicture ) . '?nocache=1" target="_picture">' . basename( $thumbPicture ) . '</a> ' . self::humanSize( filesize( $thumbPicture ) );
6057                                        }
6058                                }
6059
6060                                        $featuresCode .= '<tr><td>' . __( 'Picture', 'ppv-live-webcams' ) . '</td><td><input type="file" accept="image/*;capture=camera" name="uploadPicture1" id="uploadPicture1" class="ui button">' . '<BR>' . __( 'Update room picture to use instead of live snapshot.', 'ppv-live-webcams' ) . $picCode . '</td></tr>';
6061
6062                                        $value   = get_post_meta( $postID, 'showImage', true );
6063                                        $picCode = '';
6064
6065                                        $snapshot      = get_post_meta( $postID, 'snapshot', true );
6066                                        $thumbSnapshot = get_post_meta( $postID, 'thumbSnapshot', true );
6067
6068                                        $performerID = get_post_meta( $postID, 'performerUserID', true );
6069                                        $performer   = get_post_meta( $postID, 'performer', true );
6070                                        $avatar = get_avatar_url( $performerID, array( 'size' => max( $options['thumbWidth'], $options['thumbHeight'] ) ) );
6071
6072                                if ( $snapshot ) {
6073                                        if ( file_exists( $snapshot ) ) {
6074                                                $picCode .= '<br>Snapshot: <a href="' . self::path2url( $snapshot ) . '?nocache=1" target="_picture">' . basename( $snapshot ) . '</a> ' . self::humanSize( filesize( $snapshot ) );
```

```
6075                                    }
6076                            }
6077                            if ( $thumbSnapshot ) {
6078                                    if ( file_exists( $thumbSnapshot ) ) {
6079                                            $picCode .= '<br>Snapshot Thumb: <a
href="' . self::path2url( $thumbSnapshot ) . '?nocache=1" target="_picture">
' . basename( $thumbSnapshot ) . '</a> ' . self::humanSize( filesize( $thumbSnapshot ) );
6080                                    }
6081                            }
6082
6083                            if ( $avatar ) {
6084                                    $picCode .= '<br>Performer Avatar: <a href="' . $avatar . '"
target="_picture"> ' . $performer . ' #' . $performerID . '</a> ';
6085                            }
6086
6087                                    $featuresCode .= '<tr><td>' . __( 'Show', 'ppv-live-
webcams' ) . '</td><td><select name="showImage" id="showImage" class="ui dropdown v-select">';
6088                                    $featuresCode     .= '<option value="auto"
' . ( $value == 'auto' ? 'selected' : '' ) . '>' . __( 'Auto', 'ppv-live-webcams' ) . '</option>';
6089                                    $featuresCode     .= '<option value="teaser"
' . ( $value == 'teaser' ? 'selected' : '' ) . '>' . __( 'Teaser', 'ppv-live-webcams' ) . '</option>';
6090                                    $featuresCode     .= '<option value="picture"
' . ( $value == 'picture' ? 'selected' : '' ) . '>' . __( 'Picture', 'ppv-live-webcams' ) . '</option>';
6091                                    $featuresCode     .= '<option value="snapshot"
' . ( $value == 'snapshot' ? 'selected' : '' ) . '>' . __( 'Live Snapshot', 'ppv-live-
webcams' ) . '</option>';
6092                                    $featuresCode     .= '<option value="avatar"
' . ( $value == 'avatar' ? 'selected' : '' ) . '>' . __( 'Performer Avatar', 'ppv-live-
webcams' ) . '</option>';
6093                                    $featuresCode     .= '</select>' . '<BR>' . __( 'Select preview
to show in webcam listings and HTML5 Videochat room while not streaming. Auto shows picture or teaser
when offline and snapshot when online.', 'ppv-live-webcams' ) . $picCode . '</td></tr>';
6094
6095                        }
6096
6097                        if ( $options['lovense'] && $options['lovenseToy'] == 'manual' )
6098                        {
6099                                $value   = get_post_meta( $postID, 'lovenseToy', true );
6100
6101                                $featuresCode .= '<tr><td>' . __( 'Lovense Toy', 'ppv-live-
webcams' ) . '</td><td><select name="lovenseToy" id="lovenseToy" class="ui dropdown v-select">';
6102                                $featuresCode     .= '<option value="0"
' . ( !$value ? 'selected' : '' ) . '>' . __( 'Disabled', 'ppv-live-webcams' ) . '</option>';
6103                                $featuresCode     .= '<option value="1"
' . ( $value ? 'selected' : '' ) . '>' . __( 'Enabled', 'ppv-live-webcams' ) . '</option>';
6104                                $featuresCode     .= '</select>' . '<BR>' . __( 'Show in listings that
Lovense toy is active for this room.', 'ppv-live-webcams' ). '</td></tr>';
6105
6106                        }
6107
6108                        // roomLabel
6109                        $vw_roomLabel = '';
6110                        if ( self::inList( $userkeys, $options['roomLabel'] ) ) {
6111                                if ( $postID ) {
6112                                        $vw_roomLabel = get_post_meta( $postID, 'vw_roomLabel', true );
6113                                };
6114
6115                                $featuresCode .= '<tr><td>' . __( 'Label', 'ppv-live-webcams' ) . '</td>
<td><input size=24 name="roomLabel" id="roomLabel" value="' . $vw_roomLabel . '"><BR>' . __( 'Shows in
room listings instead of name.', 'ppv-live-webcams' ) . '</td></tr>';
6116                        }
6117
6118
6119                        // roomCategory
6120                        $newCat = '';
6121                        if ( self::inList( $userkeys, $options['roomCategory'] ) ) {
```

```
Line
6122                                $cats = wp_get_post_categories( $postID );
6123                                if ( count( $cats ) ) {
6124                                        $newCat = array_pop( $cats );
6125                                }
6126
6127                                $featuresCode .= '<tr><td>' . __( 'Category', 'ppv-live-
webcams' ) . '</td>
<td>' . wp_dropdown_categories( 'show_count=0&echo=0&name=newcategory&hide_empty=0&class=ui+dropdown&sele
cted=' . $newCat ) . '<BR>' . __( 'Category.', 'ppv-live-webcams' ) . '</td></tr>';
6128                        }
6129
6130                        // roomDescription
6131                        if ( self::inList( $userkeys, $options['roomDescription'] ) ) {
6132                                $featuresCode .= '<tr><td>' . __( 'Description', 'ppv-live-
webcams' ) . '</td><td><textarea rows="4" cols="80" name="roomDescription"
id="roomDescription">' . htmlspecialchars( $webcamPost->post_content ) . '</textarea><BR>' . __( 'Room
description: profile, schedule. Shows on room page and full row list layout.', 'ppv-live-
webcams' ) . '</td></tr>';
6133                        }
6134
6135                        // roomBrief
6136                        if ( self::inList( $userkeys, $options['roomBrief'] ) ) {
6137                                if ( $postID ) {
6138                                        $value = get_post_meta( $postID, 'vw_roomBrief', true );
6139                                } else {
6140                                        $value = '';
6141                                }
6142
6143                                $featuresCode .= '<tr><td>' . __( 'Brief', 'ppv-live-webcams' ) . '</td>
<td><textarea rows="2" cols="80" name="roomBrief"
id="roomBrief">' . htmlspecialchars( $value ) . '</textarea><BR>' . __( 'Room brief info: profile,
schedule. Shows in room listings.', 'ppv-live-webcams' ) . '</td></tr>';
6144                        }
6145 /*
6146                        // private2way
6147                        if ( self::inList( $userkeys, $options['private2way'] ) ) {
6148                                if ( $postID ) {
6149                                        $value = get_post_meta( $postID, 'vw_private2way', true );
6150                                } else {
6151                                        $value = '';
6152                                }
6153
6154                                $featuresCode .= '<tr><td>' . __( 'Private Videochat Mode', 'ppv-live-
webcams' ) . '</td><td><select name="private2way" id="private2way" class="ui dropdown v-select">';
6155                                $featuresCode .= '<option value="" ' . ( $value == '' ? 'selected' : '' )
. '>' . __( 'Default', 'ppv-live-webcams' ) . '</option>';
6156                                $featuresCode .= '<option value="1way" ' . ( $value == '1way' ?
'selected' : '' ) . '>' . __( '1 Way', 'ppv-live-webcams' ) . '</option>';
6157                                $featuresCode .= '<option value="2way" ' . ( $value == '2way' ?
'selected' : '' ) . '>' . __( '2 Way', 'ppv-live-webcams' ) . '</option>';
6158                                // $featuresCode .= '<option value="both" '.
($value=='both'?'selected':'').'>Both Modes</option>';
6159                                $featuresCode .= '</select><BR>' . __( 'Allow clients to start their cams
for 2 way videochat.', 'ppv-live-webcams' ) . '</td></tr>';
6160                        }
6161 */
6162                        // roomTags
6163                        if ( self::inList( $userkeys, $options['roomTags'] ) ) {
6164                                $tags = wp_get_post_tags( $postID, array( 'fields' => 'names' ) );
6165                                // var_dump($tags);
6166                                $value = '';
6167
6168                                if ( ! empty( $tags ) ) {
6169                                        if ( ! is_wp_error( $tags ) ) {
6170                                                foreach ( $tags as $tag ) {
```

```
6171                                                    $value .= ( $value ? ', ' : '' ) . $tag;
6172                                            }
6173                                    }
6174                            }
6175
6176                                                    $featuresCode .= '<tr><td>' . __( 'Tags', 'ppv-live-webcams' ) . '</td><td><textarea rows=2 cols="80" name="roomTags" id="roomTags">' . $value . '</textarea><BR>' . __( 'Tags separated by comma. Show in room listings.', 'ppv-live-webcams' ) . '</td></tr>';
6177                    }
6178
6179    /*
6180                    // presentationMode
6181                    if ( self::inList( $userkeys, $options['presentationMode'] ) ) {
6182                            if ( $postID ) {
6183                                    $value = get_post_meta( $postID, 'vw_presentationMode', true );
6184                            } else {
6185                                    $value = '';
6186                            }
6187
6188                            $featuresCode .= '<tr><td>' . __( 'Collaboration Mode', 'ppv-live-webcams' ) . '</td><td><select name="presentationMode" id="presentationMode" class="ui dropdown v-select">';
6189                            $featuresCode .= '<option value="1" ' . ( $value == '1' ? 'selected' : '' ) . '>' . __( 'Enabled', 'ppv-live-webcams' ) . '</option>';
6190                            $featuresCode .= '<option value="0" ' . ( $value == '0' ? 'selected' : '' ) . '>' . __( 'Disabled', 'ppv-live-webcams' ) . '</option>';
6191                            // $featuresCode .= '<option value="both" '.($value=='both'?'selected':'').'>Both Modes</option>';
6192                            $featuresCode .= '</select><BR>' . __( 'Enables collaboration mode with file sharing.', 'ppv-live-webcams' ) . '</td></tr>';
6193                    }
6194    */
6195                    // logoCustom
6196                    if ( self::inList( $userkeys, $options['logoCustom'] ) ) {
6197                            $value         = get_post_meta( $editPost, 'vw_logoImage', true );
6198                            $featuresCode .= '<tr><td>' . __( 'Logo Image', 'ppv-live-webcams' ) . '</td><td><input size=64 name="logoImage" id="logoImage" value="' . $value . '"><BR>' . __( 'Floating logo URL (preferably a transparent PNG image). Leave blank to hide.', 'ppv-live-webcams' ) . '</td></tr>';
6199                            $value         = get_post_meta( $editPost, 'vw_logoLink', true );
6200                            $featuresCode .= '<tr><td>' . __( 'Logo Link', 'ppv-live-webcams' ) . '</td><td><input size=64 name="logoLink" id="logoImage" value="' . $value . '"><BR>' . __( 'URL to open on logo click.', 'ppv-live-webcams' ) . '</td></tr>';
6201                    }
6202    */
6203
6204
6205                    $this_page = self::getCurrentURL();
6206
6207                    $msg1 = __( 'Room Setup', 'ppv-live-webcams' );
6208                    $msg2 = __( 'Setup', 'ppv-live-webcams' );
6209
6210                    $interfaceClass = $options['interfaceClass'];
6211
6212                    if ( $featuresCode ) {
6213                            $content['setup'] = <<<HTMLCODE
6214    <form method="post" enctype="multipart/form-data" action="$this_page" name="adminForm" class="ui $interfaceClass form w-actionbox">
6215    <h3 class="ui $interfaceClass header">$msg1</h3>
6216    <table class="ui $interfaceClass selectable striped table form">
6217    $featuresCode
6218    <tr><td></td><td><input class="ui button primary" type="submit" name="setup" id="setup" value="$msg2" /></td></tr>
6219    </table>
6220    </form>
```

```
HTMLCODE;
                        }

                        /*
                        $contentCode .= '
                        <br style="clear:both">
                        </div>';
                        */
                }

                // !matching tab

                if ($options['match'] ?? false )
                {
                                if ( $_POST['matchFor'] ?? false) {
                                        $active = 'match';
                                }

                        $header['match'] = __( 'Match', 'ppv-live-webcams' );
                        $content['match'] = __('Define criteria for random 2 way video call matches.','ppv-live-webcams') . do_shortcode( "[videowhisper_match_form postid=\"$postID\"]" );

                }


                // ! profile tab

                if ( $options['profiles'] ) {
                        if ( is_array( $options['profileFields'] ) ) {

                                /*
                                $checked = '';
                                if ($_POST['save']) $checked = 'active';
                                if ($checked) $checked1 = true;

                                $headerCode .= '<a class="item ' . $checked .'" data-tab="profile">' . __('Profile', 'ppv-live-webcams') . '</a>';

                                $contentCode .= '<div class="ui ' . $options['interfaceClass'] . ' bottom attached tab segment ' . $checked .'" data-tab="profile">';

                                                */
                                if ( $_POST['profileFor'] ?? false ) {
                                        $active = 'profile';
                                }

                                $header['profile'] = __( 'Profile', 'ppv-live-webcams' );

                                $contentCode = '';
                                /*
                                $htmlCode .= '<div class="vwtab">
                                                <input type="radio" id="tab-profile" name="tab-group-1" '.$checked.'>
                                                <label class="vwlabel" for="tab-profile">' . __('Profile', 'ppv-live-webcams') . '</label>

                                                <div class="vwpanel">
                                                <div class="vwcontent">';
                                                */

                                $contentCode .= '<div class="ui ' . $options['interfaceClass'] . ' form"><form method="post" enctype="multipart/form-data" action="' . $this_page . '" name="profileForm"
```

```
class="w-actionbox">';
                                $contentCode .= '<h3 class="ui ' . $options['interfaceClass'] . ' header">' . ucwords($options['custom_post']). ' '.  __( 'Room Listing Profile', 'ppv-live-webcams' ) . '</H3> ' . __( 'These details will show on webcam room page (under videochat interface).', 'ppv-live-webcams' );

                                $contentCode .= '<div class="ui divider" /></div>';

                                // allowed tags
                                $allowedtags = array(
                                        'a'          => array(
                                                'href'  => true,
                                                'title' => true,
                                        ),
                                        'abbr'       => array(
                                                'title' => true,
                                        ),
                                        'acronym'    => array(
                                                'title' => true,
                                        ),
                                        'b'          => array(),
                                        'blockquote' => array(
                                                'cite' => true,
                                        ),
                                        'cite'       => array(),
                                        'code'       => array(),
                                        'del'        => array(
                                                'datetime' => true,
                                        ),
                                        'em'         => array(),
                                        'i'          => array(),
                                        'q'          => array(
                                                'cite' => true,
                                        ),
                                        'strike'     => array(),
                                        'strong'     => array(),

                                        'ul'         => array(),
                                        'ol'         => array(),
                                        'li'         => array(),

                                        'span'       => array(
                                                'style' => array(),
                                        ),

                                        'p'          => array(
                                                'style' => array(),
                                        ),
                                );

                                $tinymce_options = array(
                                        'plugins'          => 'lists,link,textcolor,hr',
                                        'toolbar1'
 => 'cut,copy,paste,|,undo,redo,|,fontsizeselect,forecolor,backcolor,bold,italic,underline,strikethrough',
                                        'toolbar2'
 => 'alignleft,aligncenter,alignright,alignjustify,blockquote,hr,bullist,numlist,link,unlink',
                                        'fontsize_formats' => '8pt 9pt 10pt 11pt 12pt 14pt 16pt 18pt',
                                );

                                //load location info in profile editor
                                $locationsPath = $options['uploadsPath'] . '/_locations/';
```

```
6334                                $locationsPathDefaut = dirname(plugin_dir_path( __FILE__ )) . '/data/';
6335
6336                                $fileLocations = $locationsPath . 'locations.json';
6337                                $fileLanguages = $locationsPath . 'languages.json';
6338
6339
if ( ! file_exists( $fileLocations ) )$fileLocations = $locationsPathDefaut . 'locations.json';
6340
if ( ! file_exists( $fileLanguages ) )$fileLanguages = $locationsPathDefaut . 'languages.json';
6341
6342
if ( file_exists( $fileLocations ) ) $locations = json_decode( file_get_contents( $fileLocations ), true
);
6343                                if ( ! is_array( $locations ) ) $locations = array();
6344
6345
if ( file_exists( $fileLanguages ) ) $languages = json_decode( file_get_contents( $fileLanguages ), true
);
6346                                if ( ! is_array( $languages ) ) $languages = array('en'=> 'English');
6347
6348                                $continents = array();
6349                                $countries = array();
6350                                $regions = array();
6351
6352                                foreach ( $locations as $continent_code => $continent )
6353                                {
6354                                        $continents[ $continent_code ] = $continent['continent_name'];
6355                                        foreach ( $continent['countries'] as $country_code => $country )
6356                                        {
6357                                                $countries[ $country_code ] = $country['country_name'];
6358
6359                                                if ( isset( $country['regions'] ) ) {
6360
foreach ( $country['regions'] as $region_code => $region ) {
6361                                                                        $regions[ $country_code ]
[ $region_code ] = $region['region_name'];
6362                                                        }
6363                                                }
6364                                        }
6365                                }
6366
6367                                $ajaxurlLocation = admin_url( 'admin-ajax.php' ) . '?
action=vmls_location';
6368
6369                                $contentCode .= '<script>
6370                                function updateLocationField(fieldName, level, continent, country,
region) {
6371                                        var data = {
6372                                                for: "select",
6373                                                field_name: fieldName,
6374                                                level: level,
6375                                                continent0: continent,
6376                                                country0: country,
6377                                                region0: region,
6378                                                continent: jQuery("#" + fieldName + "_continent").val(),
6379                                                country: jQuery("#" + fieldName + "_country").val(),
6380                                                region: jQuery("#" + fieldName + "_region").val(),
6381                                        };
6382
6383                                        if (level == "continent" || level == "country")
6384                                        jQuery.post("' . $ajaxurlLocation . '", data, function(response)
{
6385                                                if (response) jQuery("#" + fieldName +
"_container").html(response);
```

```
Line
6386                                            console.log(response, response.countries);
6387                                    });
6388
6389                                    //update hidden field
6390                                    var value = jQuery("#" + fieldName + "_region").val() + ", " +
jQuery("#" + fieldName + "_country").val() + ", " + jQuery("#" + fieldName + "_continent").val();
6391                                    jQuery("#" + fieldName).val(value);
6392                            }
6393                            </script>';
6394
6395
6396                            foreach ( $options['profileFields'] as $field => $parameters ) {
6397
6398                                    $fieldName = sanitize_title( trim( $field ) );
6399
6400                                    if ( $parameters['instructions'] ?? false ) {
6401                                            $htmlInstructions = ' data-
tooltip="' . htmlspecialchars( stripslashes( $parameters['instructions'] ) ) . '"';
6402                                    } else {
6403                                            $htmlInstructions = '';
6404                                    }
6405
6406                                    $iconCode = '';
6407
$icon = isset($parameters['icon']) ? stripslashes( $parameters['icon'] ) : ( $options['profileFieldIcon']
 ?? false );
6408                                    if ($icon) $iconCode .= '<i class="' . $icon . ' icon"></i> ';
6409
6410                                    $contentCode .= '<div class="field"' . $htmlInstructions . '>
<label for="' . $fieldName . '">' . $iconCode .  $field . '</label>';
6411                                    // $contentCode .= '<span>' .
htmlspecialchars(stripslashes($parameters['instructions'])) . '</span>';
6412
6413
6414                                    // save data
6415                                    if ( isset( $_POST[ $fieldName ] ) ) {
6416
if ( $postID == sanitize_text_field( $_POST['profileFor'] ?? 0 ) ) {
6417
if ( in_array( $parameters['type'], array('checkboxes', 'multiselect', 'language') ) )
6418                                                                    {
6419                                                                            $tags = $_POST[ $fieldName ] ?? array();
6420
6421                                                                            if ( !is_array( $tags ) ) {
6422                                                                                    $tags = explode( ',', $tags );
6423                                                                            }
6424
6425                                                                            foreach ( $tags as &$tag ) {
6426
$tag = sanitize_text_field( trim($tag) );
6427                                                                            }
6428                                                                            unset( $tag );
6429
6430                                                                            //$contentCode .= "--Saving
vwf_$fieldName: " . count( $tags ) . ' items = ' . json_encode( $tags );
6431
6432
update_post_meta( $postID, 'vwf_' . $fieldName, $tags );
6433                                                                    } else if ( $parameters['type'] == 'location' )
6434                                                                    {
6435                                                                            //obtain field value from region,
country, continent separated by ,
6436
$fieldValue = sanitize_text_field( $_POST[ $fieldName . '_region' ] ) . ',
```

```
' . sanitize_text_field( $_POST[ $fieldName . '_country' ] ) . ',
' . sanitize_text_field( $_POST[ $fieldName . '_continent' ] );
6437 update_post_meta( $postID, 'vwf_' . $fieldName, $fieldValue );
6438
6439 update_post_meta( $postID, 'vwf_' . $fieldName . '_continent', sanitize_text_field( $_POST[ $fieldName . '_continent' ] ) );
6440 update_post_meta( $postID, 'vwf_' . $fieldName . '_country', sanitize_text_field( $_POST[ $fieldName . '_country' ] ) );
6441 update_post_meta( $postID, 'vwf_' . $fieldName . '_region', sanitize_text_field( $_POST[ $fieldName . '_region' ] ) );
6442                                                 }
6443                                                 else {
6444                                                         if (is_string($_POST[ $fieldName ])) update_post_meta( $postID, 'vwf_' . $fieldName, wp_kses( $_POST[ $fieldName ], $allowedtags ) );
6445                                                         else $contentCode .= '* Warning: Invalid content for field ' . $fieldName . ' of type ' . $parameters['type'];
6446                                                 }
6447                                         }
6448                                 }
6449
6450                                         // get data for field depending on type
6451 if ( in_array( $parameters['type'], array('checkboxes', 'multiselect', 'language') ) )
6452                                          {
6453 $fieldValue = get_post_meta( $postID, 'vwf_' . $fieldName, true );
6454                                                 if ( ! $fieldValue ) {
6455                                                         $fieldValue = array();
6456                                                 }
6457                                                 if ( ! is_array( $fieldValue ) ) {
6458                                                         $fieldValue = explode( ',', $fieldValue );
6459                                                 }
6460                                 } else {
6461 $fieldValue = get_post_meta( $postID, 'vwf_' . $fieldName, true );
6462                                 }
6463                                 // $fieldValue = htmlspecialchars(stripslashes(get_post_meta( $postID, 'vwf_' . $fieldName, true )));
6464
6465                                 $fieldOptions  = array();
6466                                 if (isset($parameters['options'])) {
6467 $parameters['options'] = stripslashes($parameters['options']);
6468                                                 //if contains | explode by | else by /
6469                                                 if (strpos($parameters['options'], '|') !== false) {
6470                                                         $fieldOptions = explode('|', $parameters['options']);
6471                                                 } else {
6472                                                         $fieldOptions = explode('/', $parameters['options']);
6473                                                 }
6474                                 }
6475
6476                                 // form
6477                                 switch ( $parameters['type'] ) {
6478
6479                                         case 'location';
6480 $continentValue = get_post_meta( $postID, 'vwf_' . $fieldName . '_continent', true );
6481                                                 $countryValue  = get_post_meta( $postID, 'vwf_' . $fieldName . '_country', true );
6482                                                 $regionValue = get_post_meta( $postID, 'vwf_' . $fieldName . '_region', true );
```

```
6483
6484                                            $contentCode .= '
6485                                                <input type="hidden"
name="' . $fieldName . '" id="' . $fieldName . '" value="' . $fieldValue . '">
6486                                                <select class="ui dropdown"
name="' . $fieldName . '_continent" id="' . $fieldName . '_continent"
onchange="updateLocationField(\'' . $fieldName . '\',
\'continent\',\'' . $continentValue . '\',\'' . $countryValue . '\',\'' . $regionValue . '\' )">
6487                                                    <option value="">' . __('Select
Continent', 'ppv-live-webcams') . '</option>';
6488
6489                                            foreach ($continents as $key => $continent) {
6490                                                $contentCode .= '<option
value="' . $continent . '"
' . ($continent == $continentValue ? 'selected' : '') . '>' . htmlspecialchars($continent) . '</option>';
6491                                            }
6492
6493                                            $contentCode .= '</select>';
6494
6495                                            $contentCode .= '<div id="' . $fieldName . '_container">
</div>';
6496
6497                                            //if $continent is set, add JS code to load by ajax
countries from that continent updateLocationField
6498
6499                                            if ($continentValue) $contentCode .= '<script
type="text/javascript">
6500                                            jQuery(document).ready(function() {
6501                                                updateLocationField("' . $fieldName . '",
"continent", "' . $continentValue . '", "' . $countryValue . '", "' . $regionValue . '");
6502                                            });
6503                                            </script>';
6504
6505                                            break;
6506
6507                                            case 'continent';
6508                                            $contentCode .= '<div class="ui fluid search selection
dropdown v-select">
6509                                            <input type="hidden" name="' . $fieldName . '"
id="' . $fieldName . '" value="' . esc_attr($fieldValue). '"  >
6510                                            <i class="dropdown icon"></i>
6511                                            <div class="default text">' .__( 'Continent', 'ppv-live-
webcams' ) . '</div>
6512                                            <div class="menu">';
6513                                            foreach ( $continents as $key => $continent ) {
6514                                                $contentCode .= '<div class="item" data-
value="' . htmlspecialchars( stripslashes( $continent ) ) . '">' . htmlspecialchars( stripslashes( $conti
nent ) ) . '</div>';
6515                                            }
6516                                            $contentCode .= '</div></div>';
6517                                            break;
6518
6519                                            case 'country';
6520                                            $contentCode .= '<div class="ui fluid search selection
dropdown v-select">
6521                                            <input type="hidden" name="' . $fieldName . '"
id="' . $fieldName . '" value="' . esc_attr($fieldValue). '"  >
6522                                            <i class="dropdown icon"></i>
6523                                            <div class="default text">' .__( 'Country', 'ppv-live-
webcams' ) . '</div>
6524                                            <div class="menu">';
6525
6526                                            foreach ( $countries as $key => $country ) {
6527                                                $contentCode .= '<div class="item" data-
value="' . htmlspecialchars( stripslashes( $country ) ) . '"><i class="' . self::language2flag($key) . '
flag"></i> ' . htmlspecialchars( stripslashes( $country ) ) . '</div>';
6528                                            }
```

```
6529                                            $contentCode .= '</div></div>';
6530                                            break;
6531
6532                                            case 'region';
6533                                            $contentCode .= '<div class="ui fluid search selection
dropdown v-select">
6534                                            <input type="hidden" name="' . $fieldName . '"
id="' . $fieldName . '" value="' . esc_attr($fieldValue). '"  >
6535                                            <i class="dropdown icon"></i>
6536                                            <div class="default text">' .__( 'Region', 'ppv-live-
webcams' ) . '</div>
6537                                            <div class="menu">';
6538                                            foreach ( $regions as $country_code => $country ) {
6539                                                    foreach ( $country as $region_code => $region ) {
6540                                                            $contentCode .= '<div class="item" data-
value="' . htmlspecialchars( stripslashes( $region ) ) . '">' . htmlspecialchars( stripslashes( $region
. ' (' . $country_code .')' ) ) . '</div>';
6541                                                    }
6542                                            }
6543                                            $contentCode .= '</div></div>';
6544                                            break;
6545
6546                                            case 'language';
6547                                            $contentCode .= '<div class="ui fluid search multiple
selection dropdown v-select">
6548                                              <input type="hidden" name="' . $fieldName . '" id="' . $fieldName . '"
value="' . esc_attr( implode(',', $fieldValue) ). '"  >
6549                                              <i class="dropdown icon"></i>
6550                                              <div class="default text">' .__( 'Language', 'ppv-live-webcams' ) . '</div>
6551                                              <div class="menu">';
6552                                            foreach ( $languages as $key => $language ) {
6553                                                    $contentCode .= '<div class="item" data-
value="' . htmlspecialchars( stripslashes( $language ) ) . '">' . htmlspecialchars( stripslashes( $langua
ge ) ) . '</div>'; //languages are not countries <i class="' . self::language2flag($key) . ' flag"></i>
6554                                            }
6555                                            $contentCode .= '</div></div>';
6556                                            break;
6557
6558                                            case 'text';
6559                                                    $contentCode .= '<INPUT type="text" size="72"
name="' . $fieldName . '" id="' . $fieldName . '" value="' . $fieldValue . '">';
6560                                            break;
6561
6562                                            case 'textarea';
6563                                                    // $contentCode .= '<TEXTAREA type="text"
rows="3" cols="70" name="'. $fieldName . '" id="'. $fieldName . '">' .$fieldValue. '</TEXTAREA>';
6564
6565                                                    ob_start();
6566                                                    wp_editor(
6567                                                            $fieldValue,
6568                                                            $fieldName,
6569                                                            $settings = array(
6570                                                                    'textarea_rows' => 3,
6571                                                                    'media_buttons' => false,
6572                                                                    'teeny'         => true,
6573                                                                    'wpautop'       => false,
6574                                                                    'tinymce'
 => $tinymce_options,
6575                                                            )
6576                                                    );
6577                                                    $contentCode .= ob_get_clean();
6578
6579                                            break;
6580
```

```
6581                                                case 'select';
6582                                                        $contentCode .= '<SELECT class="ui dropdown search clearable v-select" name="' . $fieldName . '" id="' . $fieldName . '">';
6583                                                        $contentCode .= '<OPTION value="" ' . ( ! $fieldValue ? 'selected' : '' ) . '> - </OPTION>';
6584
6585                                                        foreach ( $fieldOptions as $fieldOption ) {
6586                                                                $contentCode .= '<OPTION value="' . htmlspecialchars( stripslashes( $fieldOption ) ) . '" ' . ( $fieldOption == $fieldValue ? 'selected' : '' ) . '>' . htmlspecialchars( $fieldOption ) . '</OPTION>';
6587                                                        }
6588
6589                                                        $contentCode .= '</SELECT>';
6590                                                break;
6591
6592                                                case 'multiselect';
6593                                                        $contentCode .= '<SELECT MULTIPLE class="ui dropdown multiple search v-select" name="' . $fieldName . '[]" id="' . $fieldName . '" >';
6594                                                        foreach ( $fieldOptions as $fieldOption ) {
6595                                                                $contentCode .= '<OPTION value="' . htmlspecialchars( stripslashes( $fieldOption ) ) . '" ' . ( in_array( $fieldOption, $fieldValue ) ? 'selected' : '' ) . '>' . htmlspecialchars( $fieldOption ) . '</OPTION>';
6596                                                        }
6597
6598                                                        $contentCode .= '</SELECT>';
6599                                                        //$contentCode .= implode ( ', ', $fieldValue );
6600                                                break;
6601
6602                                                case 'checkboxes';
6603                                                        foreach ( $fieldOptions as $fieldOption ) {
6604                                                                $contentCode .= '<div class="field"><div class="ui toggle checkbox">
6605                                                          <input type="checkbox" name="' . $fieldName . '[]" value="' . htmlspecialchars( stripslashes( $fieldOption ) ) . '" ' . ( in_array( $fieldOption, $fieldValue ) ? 'checked' : '' ) . '>
6606                                                          <label>' . htmlspecialchars( $fieldOption ) . '</label></div></div>';
6607                                                        }
6608                                                break;
6609
6610                                        }
6611
6612                                        // if ($parameters['instructions']) $contentCode .= '<BR>' . htmlspecialchars($parameters['instructions']);
6613
6614                                        $contentCode .= '<div class="ui divider" /></div> </div>';
6615                                }
6616
6617                                $contentCode .= '<input type="hidden" name="profileFor" id="profileFor" value="' . $postID . '">
6618                                <BR><input class="ui button primary" type="submit" name="save" id="save" value="' . __( 'Save', 'ppv-live-webcams' ) . '" />
6619                                </form></div>';
6620
6621                                $content['profile'] = $contentCode;
6622
6623                                /*
6624                                $contentCode .= '
6625                                <br style="clear:both">
6626                                </div>
6627                                ';*/
6628                        }
6629                }
6630
```

```
6631
6632                if ($options['categoriesContest'] ?? false)
6633                {
6634                        if ( ($_GET['categoryAdd'] ?? false) || ( $_GET['categoryRemove'] ?? false ) ) {
6635                                $active = 'categories';
6636                        }
6637
6638                        $header['categories'] = __( 'Contests', 'ppv-live-webcams' );
6639                        $content['categories'] = wp_kses_post( $options['categoriesMessage'] ) . '<br>';
6640
6641                        $contestIDs = self::categoryIDs($options['categoriesContest']);
6642                        $categories = wp_get_post_categories( $postID );
6643
6644                        if ($_GET['categoryAdd'] ?? false) {
6645                                $contestID = intval($_GET['categoryAdd']);
6646                                if ($contestID && !in_array($contestID, $categories)) {
6647
wp_set_post_categories( $postID, array_merge($categories, array($contestID)) );
6648                                        $categories = wp_get_post_categories( $postID );
6649                                }
6650                        }
6651
6652                        if  ( $_GET['categoryRemove'] ?? false )
6653                        {
6654                                $contestID = intval($_GET['categoryRemove']);
6655                                if ($contestID && in_array($contestID, $categories)) {
6656
wp_set_post_categories( $postID, array_diff($categories, array($contestID)) );
6657                                        $categories = wp_get_post_categories( $postID );
6658                                }
6659                        }
6660
6661                        foreach ($contestIDs as $contestID)
6662                        {
6663                                $contestCategory = get_term($contestID);
6664                                //check if category is assigned to post $postID
6665
6666                                $content['categories'] .= '<br><div class="ui label
large">' . $contestCategory->name . '</div>' . ( in_array($contestID, $categories) ? ' <i class="icon
check"></i>' : '' ) ;
6667                                $currentURL = self::getCurrentURL();
6668
6669                                if (in_array($contestID, $categories))
6670                                {
6671                                        //include wp_nonce_url to remove category
6672                                        $content['categories'] .= '<a class="ui button negative tiny"
href="' . wp_nonce_url( add_query_arg( 'categoryRemove', $contestID, $currentURL), 'category' . $postID,
'verify' ). '">' . __('Remove', 'ppv-live-webcams') . '</a>';
6673
6674                                }
6675                                else
6676                                {
6677                                        //include wp_nonce_url to add category
6678                                        $content['categories'] .= '<a class="ui button positive tiny"
href="' . wp_nonce_url( add_query_arg( 'categoryAdd', $contestID, $currentURL), 'category' . $postID, 've
rify' ). '">' . __('Add', 'ppv-live-webcams') . '</a>';
6679                                }
6680
6681                        }
6682                }
6683
6684                // ! Restreaming tab
6685
```

```
6686                if (!$options['webrtcOnly'])
6687                if ($options['reStreams'] || $options['pushStreams'] ||
 self::is_true( get_post_meta( $postID, 'external_rtmp', true ) ) ) {
6688                        if ( $_GET['streams'] ?? false ) {
6689                                $active = 'streams';
6690                        }
6691
6692                        $header['streams'] = __( 'Streams', 'ppv-live-webcams' );
6693
6694                        $contentCode = '';
6695
6696                        $contentCode .= do_shortcode( "[videowhisper_streams roomid=\"$postID\"]" );
6697
6698                        $content['streams'] = $contentCode;
6699                }
6700
6701                // ! videos tab
6702                if ( $options['videosharevod'] ) {
6703                        if ( shortcode_exists( 'videowhisper_postvideos' ) ) {
6704                                if ( self::inList( $userkeys, $options['videos'] ) ) {
6705
6706                                        // import saved archives
6707                                        self::importArchives( $postID );
6708
6709                                        /*
6710                                        $checked = '';
6711                                        if ($_GET['playlist_upload'] || $_GET['playlist_import'] ||
$_GET['import_preview']) $checked = 'active';
6712                                        if ($checked) $checked1 = true;
6713
6714                                        $headerCode .= '<a class="item ' . $checked .'" data-
tab="videos">' . __('Videos', 'ppv-live-webcams') . '</a>';
6715
6716                                        $contentCode .= '<div class="ui ' . $options['interfaceClass'] .
' bottom attached tab segment ' . $checked .'" data-tab="videos">';
6717                                        */
6718                                        if ( isset( $_GET['playlist_upload'] ) ||
 isset( $_GET['playlist_import'] ) || isset( $_GET['import_preview'] ) ) {
6719                                                $active = 'videos';
6720                                        }
6721
6722                                        $header['videos'] = __( 'Videos', 'ppv-live-webcams' );
6723
6724                                        $contentCode = '';
6725                                        /*
6726                                        $htmlCode .= '<div class="vwtab">
6727                                        <input type="radio" id="tab-vid" name="tab-group-1" '.$checked.'>
6728                                        <label class="vwlabel" for="tab-vid">' . __('Videos', 'ppv-live-
webcams') . '</label>
6729
6730                                        <div class="vwpanel">
6731                                        <div class="vwcontent">';
6732                                        */
6733
6734                                        // recordings path
6735                                        $dir = $options['uploadsPath'];
6736                                        if ( ! file_exists( $dir ) ) {
6737                                                mkdir( $dir );
6738                                        }
6739                                        $dir .= '/' . sanitize_file_name( $webcamPost->post_title );
6740                                        if ( ! file_exists( $dir ) ) {
6741                                                mkdir( $dir );
6742                                        }
```

```
6743                                    $dir .= '/_recordings';
6744                                    if ( ! file_exists( $dir ) ) {
6745                                            mkdir( $dir );
6746                                    }
6747
6748                                    $contentCode .= '<p>' . __( 'Multiple profile videos can show on webcam listing page. Each listing has own videos.', 'ppv-live-webcams' ) . '</p>';
6749
6750                                    $contentCode .= do_shortcode( "[videowhisper_postvideos_process path=\"$dir\" prefix=\"\" post=\"$postID\"]" );
6751
6752                                    $contentCode .= do_shortcode( "[videowhisper_postvideos path=\"$dir\" prefix=\"\" post=\"$postID\"]" );
6753
6754                                    $contentCode .= '<p>' . __( 'Recorded live streams can be published using Import.', 'ppv-live-webcams' ) . '</p>';
6755
6756                                    // record
6757                                    $contentCode .= '<p><b>' . __( 'Record Perfomer', 'ppv-live-webcams' ) . ':
' . ( self::is_true( get_post_meta( $postID, 'stream_record', true ) ) ? __( 'Enabled', 'ppv-live-webcams' ) : __( 'Disabled', 'ppv-live-webcams' ) ) . '</b><br>' . __( 'Record performer stream.', 'ppv-live-webcams' ) . '</p>';
6758                                    $contentCode .= '<p><b>' . __( 'Record Private', 'ppv-live-webcams' ) . ':
' . ( self::is_true( get_post_meta( $postID, 'stream_record_private', true ) ) ? __( 'Enabled', 'ppv-live-webcams' ) : __( 'Disabled', 'ppv-live-webcams' ) ) . '</b><br>' . __( 'Record in private calls.', 'ppv-live-webcams' ) . '</p>';
6759
6760                                    $contentCode .= '<p><b>' . __( 'Record All', 'ppv-live-webcams' ) . ':
' . ( self::is_true( get_post_meta( $postID, 'stream_record_all', true ) ) ? __( 'Enabled', 'ppv-live-webcams' ) : __( 'Disabled', 'ppv-live-webcams' ) ) . '</b><br>' . __( 'Record streams from all users.', 'ppv-live-webcams' ) . '</p>';
6761
6762                                    $content['videos'] = $contentCode;
6763
6764                                    /*
6765                                    $contentCode .= '
6766                                    <br style="clear: both">
6767                                    </div>
6768                                    ';
6769                                    */
6770                                }
6771                        }
6772                }
6773
6774                        // !  teaser video tab
6775                if ( $options['videosharevod'] ) {
6776                        if ( $options['teaserOffline'] ) {
6777                                if ( shortcode_exists( 'videowhisper_postvideo_assign' ) ) {
6778                                        if ( self::inList( $userkeys, $options['videos'] ) ) {
6779                                                /*
6780                                                $checked = '';
6781                                                if ($_GET['assignVideo']) $checked = 'active';
6782                                                if ($checked) $checked1 = true;
6783
6784                                                $headerCode .= '<a class="item ' . $checked .'" data-tab="teaser">' . __('Teaser', 'ppv-live-webcams') . '</a>';
6785                                                $contentCode .= '<div class="ui ' . $options['interfaceClass'] . ' bottom attached tab segment ' . $checked .'" data-tab="teaser">';
6786
6787                                                */
6788                                                if ( $_GET['assignVideo'] ?? false ) {
6789                                                        $active = 'teaser';
6790                                                }
```

```
6791                                            $header['teaser'] = __( 'Teaser', 'ppv-live-webcams' );
6792
6793                                            /*
6794                                            $htmlCode .= '<div class="vwtab">
6795                                            <input type="radio" id="tab-teaser" name="tab-group-1"
'.$checked.'>
6796                                            <label class="vwlabel" for="tab-teaser">' . __('Teaser',
'ppv-live-webcams') . '</label>
6797
6798                                            <div class="vwpanel">
6799                                            <div class="vwcontent">
6800                                            */
6801                                            $contentCode = do_shortcode( "
[videowhisper_postvideo_assign post_id=\"$postID\" meta=\"video_teaser\"]" );
6802
6803                                            $contentCode .= '<p>' . __( 'Teaser video (featured
video) plays shortly as preview in listings, in room when performer is offline in HTML5 Videochat
interface and shows on profile page. Can be selected from all videos of user.', 'ppv-live-
webcams' ) . '</p>';
6804
6805                                            $content['teaser'] = $contentCode;
6806
6807                                            /*
6808                                            $contentCode .= '
6809                                            <br style="clear: both">
6810                                            </div>
6811                                            ';
6812                                            */
6813                                        }
6814                                    }
6815                                }
6816                            }
6817
6818 /*
6819                                        // ! playlists scheduler tab
6820                            if ( $options['playlists'] ) {
6821                                    if ( self::inList( $userkeys, $options['schedulePlaylists'] ) ) {
6822
6823                                        // activity
6824                                        $editPlaylist = $postID;
6825
6826                                        $playlistActive = (int) $_POST['playlistActive'];
6827
6828                                        wp_enqueue_script( 'jquery' );
6829                                        wp_enqueue_script( 'jquery-ui-core' );
6830                                        wp_enqueue_script( 'jquery-ui-widget' );
6831                                        wp_enqueue_script( 'jquery-ui-dialog' );
6832
6833                                        // css
6834                                        wp_enqueue_style( 'jtable-green', dirname( plugin_dir_url( __FILE__ ) ) .
'/scripts/jtable/themes/lightcolor/green/jtable.min.css' );
6835
6836                                        wp_enqueue_style( 'jtable-flick', dirname( plugin_dir_url( __FILE__ ) ) .
'/scripts/jtable/themes/flick/jquery-ui.min.css' );
6837
6838                                        // js
6839                                        wp_enqueue_script( 'jquery-ui-jtable', dirname( plugin_dir_url( __FILE__
) ) . '/scripts/jtable/jquery.jtable.min.js', array( 'jquery-ui-core', 'jquery-ui-widget', 'jquery-ui-
dialog' ) );
6840
6841                                        $ajaxurl = admin_url() . 'admin-ajax.php?action=vmls_playlist&webcam=' .
$postID;
6842
6843
```

```
                                if ( $_POST['$editPlaylist'] || $_POST['playlistActive'] ) {
                                        $active = 'playlist';
                                }
                                        $header['playlist'] = __( 'Playlist', 'ppv-live-webcams' );
                                        $contentCode        = '';
                                        // activity on this tab
                                        
                                        
                                        
                                        // OS timezone : Wowza time
                                        $timezone = 'UTC';
                                if ( is_link( '/etc/localtime' ) ) {
                                        // Mac OS X (and older Linuxes)
                                        // /etc/localtime is a symlink to the
                                        // timezone in /usr/share/zoneinfo.
                                        $filename = readlink( '/etc/localtime' );
                                        if ( strpos( $filename, '/usr/share/zoneinfo/' ) === 0 ) {
                                                $timezone = substr( $filename, 20 );
                                        }
                                } elseif ( file_exists( '/etc/timezone' ) ) {
                                        // Ubuntu / Debian.
                                        $data = file_get_contents( '/etc/timezone' );
                                        if ( $data ) {
                                                $timezone = $data;
                                        }
                                } elseif ( file_exists( '/etc/sysconfig/clock' ) ) {
                                        // RHEL / CentOS
                                        $data = parse_ini_file( '/etc/sysconfig/clock' );
                                        if ( ! empty( $data['ZONE'] ) ) {
                                                $timezone = $data['ZONE'];
                                        }
                                }
                                
                                        $defaultTimezone = date_default_timezone_get();
                                if ( $timezone ) {
                                        date_default_timezone_set( $timezone );
                                }
                                
                                        // $webcamPost;
                                
                                        $stream = sanitize_file_name( $webcamPost->post_title );
                                
                                        // ! quick loop setup - save loop
                                
                                if ( $loop = sanitize_text_field( $_POST['loop'] ) ) {
                                
                                        $playlist = array();
                                
                                        $item              = array();
                                        $item['Id']        = 1;
                                        $item['Video']     = $loop;
                                        $item['Repeat']    = 1;
                                        $item['Scheduled'] = date( 'Y-m-j h:i:s' );
                                        $item['Order']     = 1;
                                
                                        $playlist[1] = $item;
                                
                                        $playlist[1]['Videos'] = array();
                                
                                        $item           = array();
```

```
6905                                    $item['Video']  = $loop;
6906                                    $item['Start']  = 0;
6907                                    $item['Length'] = -1;
6908                                    $item['Order']  = 1;
6909                                    $item['Id']     = 1;
6910
6911                                    $playlist[1]['Videos'][] = $item;
6912
6913                                    // $playlistPath
6914                                    $uploadsPath = $options['uploadsPath'];
6915                                    if ( ! file_exists( $uploadsPath ) ) {
6916                                            mkdir( $uploadsPath );
6917                                    }
6918                                    $upath = $uploadsPath . "/$stream/";
6919                                    if ( ! file_exists( $upath ) ) {
6920                                            mkdir( $upath );
6921                                    }
6922                                    $playlistPath = $upath . 'playlist.txt';
6923
6924                                    self::varSave( $playlistPath, $playlist );
6925
6926                                    // $contentCode .= '<p>' . __('Video Loop', 'ppv-live-webcams') . ': '.$loop.'</p>';
6927                                    update_post_meta( $postID, 'videoLoop', $loop );
6928
6929                            }
6930
6931                            $currentDate = date( 'Y-m-j h:i:s' );
6932
6933                            // activate playslit
6934                            if ( $_POST['updatePlaylist'] ) {
6935                                    update_post_meta( $postID, 'vw_playlistActive', $playlistActive );
6936                                    self::updatePlaylist( $stream, $playlistActive );
6937                                    update_post_meta( $postID, 'vw_playlistUpdated', time() );
6938
6939                                    if ( $playlistActive ) {
6940                                            $contentCode .= '<p>' . __( 'Playlist is enabled and room stream updated to', 'ppv-live-webcams' ) . ' ' . $stream . '.</p>';
6941                                            update_post_meta( $postID, 'performer', $stream );
6942                                            update_post_meta( $postID, 'performerUserID', $current_user->ID );
6943
6944                                            update_post_meta( $postID, 'edate', time() );
6945
6946                                            // stream is from rtmp server
6947                                            update_post_meta( $postID, 'stream-protocol', 'rtmp' );
6948                                            update_post_meta( $postID, 'stream-type', 'playlist' );
6949
6950                                    }
6951                            }
6952
6953                            // retrieve video list
6954                            $optionsVSV        = get_option( 'VWvideoShareOptions' );
6955                            $custom_post_video = $optionsVSV['custom_post'];
6956                            if ( ! $custom_post_video ) {
6957                                    $custom_post_video = 'video';
6958                            }
6959
6960                            // query
6961                            $args = array(
6962                                    'post_type' => $custom_post_video,
```

```
Line
6963                                                    'author'    => $current_user->ID,
6964                                                    'orderby'   => 'post_date',
6965                                                    'order'     => 'DESC',
6966                                            );
6967
6968                                            $postslist = get_posts( $args );
6969                                            if ( count( $postslist ) > 0 ) {
6970                                                    $videoLoop = get_post_meta( $postID, 'videoLoop', true );
// previous video loop
6971
6972                                                    $quickCode .= '<SELECT id="loop" name="loop" class="ui
dropdown v-select">';
6973                                                    foreach ( $postslist as $item ) {
6974                                                            $video_id = $item->ID;
6975
6976                                                            // retrieve video stream
6977                                                            $streamPath = '';
6978                                                            $videoPath  = get_post_meta( $video_id, 'video-
source-file', true );
6979                                                            $ext        = pathinfo( $videoPath,
PATHINFO_EXTENSION );
6980
6981                                                            // use conversion if available
6982                                                            $videoAdaptive = get_post_meta( $video_id,
'video-adaptive', true );
6983                                                            if ( $videoAdaptive ) {
6984                                                                    $videoAlts = $videoAdaptive;
6985                                                            } else {
6986                                                                    $videoAlts = array();
6987                                                            }
6988
6989                                                            foreach ( array( 'high', 'mobile' ) as $frm ) {
6990                                                                    if ( array_key_exists( $frm, $videoAlts )
) {
6991                                                                            if ( $alt = $videoAlts[ $frm ] )
{
6992                                                                                    if ( file_exists(
$alt['file'] ) ) {
6993                                                                                            $ext        =
pathinfo( $alt['file'], PATHINFO_EXTENSION );
6994                                                                                            $streamPath =
self::path2stream( $alt['file'] );
6995                                                                                            break;
6996                                                                                    }
6997                                                                            }
6998                                                                    }
6999                                                            };
7000
7001                                                            // user original
7002                                                            if ( ! $streamPath ) {
7003                                                                    if ( in_array( $ext, array( 'flv', 'mp4',
'm4v' ) ) ) {
7004                                                                            // use source if
compatible
7005                                                                            $streamPath =
self::path2stream( $videoPath );
7006                                                                    }
7007                                                            }
7008
7009
$quickCode .= '<option value="' . htmlspecialchars( $streamPath ) . '" ' . ( $videoLoop == $streamPath ?
'selected' : '' ) . '>' . $video_id . ' ' . $item->post_title . '</option>';
7010
7011                                                    }
7012                                                    $quickCode .= '</SELECT>';
```

```
Line
7013                                    } else {
7014                                            $quickCode = __( 'No videos found! Please add some videos first.', 'ppv-live-webcams' );
7015                                    }
7016
7017                                    $playlistPage = add_query_arg( array( 'editPlaylist' => $editPlaylist ), $this_page );
7018
7019                                    $msg1 = __( 'Quick Loop Setup', 'ppv-live-webcams' );
7020                                    $msg2 = __( 'Video will play in loop while playlist is active. Warning: Going live (with real webcam) will stop video loop (needs to be setup again). Can also be disabled by setting Playlist Status as Inactive.', 'ppv-live-webcams' );
7021                                    $msg3 = __( 'Set Video Loop', 'ppv-live-webcams' );
7022
7023                                    $contentCode .= <<<HTMLCODE
7024                                    <h3 class="ui header">$msg1</H3>
7025 <form method="post" action="$playlistPage" name="adminForm" class="w-actionbox">
7026 <label>Select Video</label> $quickCode <input class="ui button" type="submit" name="button" id="button" value="$msg3" />
7027 <input type="hidden" name="playlistActive" id="playlistActive" value="1" />
7028 <input type="hidden" name="updatePlaylist" id="updatePlaylist" value="$editPlaylist" />
7029 <BR>$msg2
7030 <br style="clear:both">
7031 </form>
7032 HTMLCODE;
7033
7034                                    // playlistActive
7035                                    $value = get_post_meta( $editPlaylist, 'vw_playlistActive', true );
7036
7037                                    $activeCode .= '<select id="playlistActive" name="playlistActive" class="ui dropdown v-select">';
7038                                    $activeCode .= '<option value="0" ' . ( ! $value ? 'selected' : '' ) . '>' . __( 'Inactive', 'ppv-live-webcams' ) . '</option>';
7039                                    $activeCode .= '<option value="1" ' . ( $value ? 'selected' : '' ) . '>' . __( 'Active', 'ppv-live-webcams' ) . '</option>';
7040                                    $activeCode .= '</select>';
7041
7042                                    $value           = get_post_meta( $editPlaylist, 'vw_playlistUpdated', true );
7043                                    $playlistUpdated = date( 'Y-m-j h:i:s', (int) $value );
7044
7045                                    $value          = get_post_meta( $editPlaylist, 'vw_playlistLoaded', true );
7046                                    $playlistLoaded = date( 'Y-m-j h:i:s', (int) $value );
7047
7048                                    $videosImg = dirname( plugin_dir_url( __FILE__ ) ) . 'scripts/jtable/themes/lightcolor/edit.png';
7049
7050                                    $channelURL = get_permalink( $postID );
7051
7052                                    $msg1 = __( 'Advanced Playlist Editor', 'ppv-live-webcams' );
7053                                    $msg2 = __( 'After editing playlist contents below, update it to apply changes. Last Updated:', 'ppv-live-webcams' );
7054                                    $msg3 = __( 'Playlist is loaded with web application (on access) and reloaded if necessary when users access', 'ppv-live-webcams' );
7055                                    $msg4 = __( 'First create a Schedule (Add new record), then Edit Videos (Add new record under Videos):', 'ppv-live-webcams' );
7056                                    $msg5 = __( 'videochat interface', 'ppv-live-webcams' );
7057                                    $msg6 = __( 'last time reloaded', 'ppv-live-webcams' );
7058
7059                                    // ! jTable
7060                                    $contentCode .= <<<HTMLCODE
7061                                    <h3 class="ui header">$msg1</H3>
7062 <form method="post" action="$playlistPage" name="adminForm" class="w-actionbox">
```

```
Line
7063 <label>Playlist Status</label> $activeCode <input class="ui button" type="submit" name="button" id="button" value="Update" />
7064 <input type="hidden" name="updatePlaylist" id="updatePlaylist" value="$editPlaylist" />
7065 <BR>$msg2 $playlistUpdated $timezone
7066 <BR>$msg3 <a href='$channelURL'>$msg5</a> ($msg6:  $playlistLoaded $timezone).
7067 </form>
7068 <BR>
7069 $msg4
7070         <div id="PlaylistTableContainer" style="width: 600px;"></div>
7071         <script type="text/javascript">
7072
7073                 jQuery(document).ready(function () {
7074
7075                     //Prepare jTable
7076                         jQuery('#PlaylistTableContainer').jtable({
7077                                 title: 'Playlist Contents',
7078                                 defaultSorting: 'Order ASC',
7079                                 toolbar: {hoverAnimation: false},
7080                                 actions: {
7081                                         listAction: '$ajaxurl&task=list',
7082                                         createAction: '$ajaxurl&task=create',
7083                                         updateAction: '$ajaxurl&task=update',
7084                                         deleteAction: '$ajaxurl&task=delete'
7085                                 },
7086                                 fields: {
7087                                         Id: {
7088                                                 key: true,
7089                                                 create: false,
7090                                                 edit: false,
7091                                                 list: false,
7092                                         },
7093                                         //CHILD TABLE DEFINITION
7094                                         Videos: {
7095                     title: 'Videos',
7096                     sorting: false,
7097                     edit: false,
7098                     create: false,
7099                     display: function (playlist) {
7100                         //Create an image that will be used to open child table
7101                         var vButton = jQuery('<IMG src="$videosImg" /><I>Edit Videos</I>');
7102                         //Open child table when user clicks the image
7103                         vButton.click(function () {
7104                             jQuery('#PlaylistTableContainer').jtable('openChildTable',
7105                                     vButton.closest('tr'),
7106                                     {
7107                                         title: 'Videos for Schedule ' + playlist.record.Scheduled,
7108                                         actions: {
7109                                             listAction: '$ajaxurl&task=videolist&item=' + playlist.record.Id,
7110                                             deleteAction: '$ajaxurl&task=videoremove&item=' + playlist.record.Id,
7111                                             updateAction: '$ajaxurl&task=videoupdate',
7112                                             createAction: '$ajaxurl&task=videoadd'
7113                                         },
7114                                         fields: {
7115                                             ItemId: {
7116                                                 type: 'hidden',
7117                                                 defaultValue: playlist.record.Id
7118                                             },
7119                                             Id: {
7120                                                 key: true,
7121                                                 create: false,
```

```
                                        edit: false,
                                        list: false
                                    },
                                                                        Video: {
                                                                            title: 'Video',
                                                                            options: '$ajaxurl&task=source',
                                                                            sorting: false
                                                                        },
                                                                        Start: {
                                                                            title: 'Start',
                                                                            defaultValue: '0',
                                                                        },
                                                                        Length: {
                                                                            title: 'Length',
                                                                            defaultValue: '-1',
                                                                        },
                                                                        Order: {
                                                                            title: 'Order',
                                                                            defaultValue: '0',
                                                                        },
                                    }
                                }, function (data) { //opened handler
                                    data.childTable.jtable('load');
                                });
                        });
                        //Return image to show on the person row
                        return vButton;
                    }

                    },
                                    Scheduled: {
                                        title: 'Scheduled',
                                        defaultValue: '$currentDate',
                                        sorting: false
                                    },
                                    Repeat: {
                                        title: 'Repeat',
                                        type: 'checkbox',
                                        defaultValue: '0',
                                        values: { '0' : 'Disabled', '1' : 'Enabled' },
                                        sorting: false
                                    },
                                    Order: {
                                        title: 'Order',
                                        defaultValue: '0',
                                    }
                                }
                        });

                        //Load item list from server
                        jQuery('#PlaylistTableContainer').jtable('load');
                });
        </script>
        <STYLE>
        .ui-widget
```

```
7177        {
7178                z-index: 1000 !important;
7179        }
7180        </STYLE>
7181
7182 HTMLCODE;
7183
7184                                        $contentCode .= '<BR>' . __( 'Schedule playlist items as: Year-Month-Day Hours:Minutes:Seconds in scheduling server timezone. In example, current server OS time', 'ppv-live-webcams' ) . ': ' . date( 'y-m-j h:i:s' ) . ' ' . __( 'in timezone', 'ppv-live-webcams' ) . ' ' . date_default_timezone_get() . ' (' . __( 'default timezone was', 'ppv-live-webcams' ) . ' ' . $defaultTimezone . ').';
7185
7186                                        if ( date_default_timezone_get() ) {
7187                                                $contentCode .= '<BR>' . __( 'If the schedule time is in the past, each video is loaded in order and immediately replaces the previous video for the stream. Repeat will cause that videos to repeat in loop.', 'ppv-live-webcams' );
7188                                        }
7189
7190                                        // restore default timezone
7191                                        if ( $defaultTimezone ) {
7192                                                date_default_timezone_set( $defaultTimezone );
7193                                        }
7194
7195                                        $contentCode .= '<p>' . __( 'Scheduled videos play instead of performer live stream in videochat interface. Can be selected from all videos of user.', 'ppv-live-webcams' ) . '</p>';
7196
7197                                        $content['playlist'] = $contentCode;
7198
7199                        }
7200                }
7201 */
7202                                        // ! content assets tab in dashboard
7203                if ( $options['micropaymentsAssets'] ?? false ) {
7204                        if ( shortcode_exists( 'videowhisper_content_list' ) ) {
7205                                if ( $_GET['assets'] ?? false ) {
7206                                        $active = 'assets';
7207                                }
7208                                                $header['assets'] = __( 'Assets', 'ppv-live-webcams' );
7209
7210                                                $content['assets'] = __( 'Select digital assets to show on room profile page.', 'ppv-live-webcams' );
7211
7212                                                $this_page = self::getCurrentURL();
7213
7214                                if ( $_POST['save'] ?? false ) {
7215                                        if ( is_array( $_POST['selectedAssets'] ?? '' ) ) {
7216
$contentIDs = get_post_meta( $postID, 'contentIDs', true );
7217                                                if ( ! is_array( $contentIDs ) ) {
7218                                                        $contentIDs = array();
7219                                                }
7220
7221                                                foreach ( $_POST['selectedAssets'] as $selected ) {
7222                                                        $selectedInt = intval( $selected );
7223
7224                                                        $sPost = get_post( abs( $selectedInt ) );
7225                                                        if ( ! $sPost ) {
7226                                                                return 'Post not found!';
7227                                                        }
7228                                                        if ( $sPost->post_author != $current_user->ID ) {
7229                                                                return 'Not your post!';
7230                                                        }
```

```
Line
7231
7232                                                    if ( $selectedInt > 0 ) {
7233
if ( ! in_array( $selectedInt, $contentIDs ) ) {
7234                                                                    $contentIDs[] = $selectedInt; //
add selected ID to show
7235                                                            }
7236                                                    }
7237
7238                                                    if ( $selectedInt < 0 ) {
7239
if ( in_array( -$selectedInt, $contentIDs ) ) {
7240
if ( ( $key = array_search( -$selectedInt, $contentIDs ) ) !== false ) {
7241
unset( $contentIDs[ $key ] ); // find and remove ID to hide
7242                                                                    }
7243                                                            }
7244                                                    }
7245                                            }
7246
7247
update_post_meta( $postID, 'contentIDs', $contentIDs );
7248
7249                                                                    $active = 'assets';
7250
7251                                            }
7252                                    }
7253
7254                                            // current
7255
$contentIDs = get_post_meta( $postID, 'contentIDs', true );
7256                                    if ( ! is_array( $contentIDs ) ) {
7257                                            $contentIDs = array();
7258                                    }
7259
7260                                            $asset_page = intval( $_GET['asset_page'] ?? 0 );
7261                                    if ( $asset_page ) {
7262                                            $active = 'assets';
7263                                    }
7264
7265                                    if ( $asset_page <= 0 ) {
7266                                            $asset_page = 1;
7267                                    }
7268
7269                                            $post_type = array();
7270                                            $postTypes = explode( ',', $options['postTypesAssets'] );
7271                                    foreach ( $postTypes as $postType ) {
7272                                            $post_type[] = trim( $postType );
7273                                    }
7274
7275                                            $args = array(
7276                                                    'author'         => $current_user->ID,
7277                                                    'post_type'      => $post_type,
7278                                                    'orderby'        => 'modified',
7279                                                    'order'          => 'DESC',
7280                                                    'posts_per_page' => 8,
7281                                                    'paged'
=> $asset_page ? $asset_page : 1,
7282                                            );
7283
7284                                            $postslist = new \WP_Query( $args ); // WP root namespace
7285
```

```
Line
7286                                        if ( $postslist->have_posts() ) :
7287
7288                                            $content['assets'] .= '<form class="ui form"
method="post" enctype="multipart/form-data" action="' . $this_page . '"  name="adminAssets">';
7289
7290                                            // console.log(\'selectAll\');
7291
7292                                            $content['assets'] .= '<table class="ui celled
striped table ' . $options['interfaceClass'] . '">
7293 <thead>
7294     <tr>
7295     <th> Select </th>
7296     <th>Asset</th>
7297     </tr>
7298 </thead>
7299 <tbody>';
7300
7301                                            while ( $postslist->have_posts() ) :
7302                                                $postslist->the_post();
7303
7304                                                $content['assets'] .= '<tr>';
7305
7306                                                $content['assets'] .= '<td>
7307                                    <select class="ui ' . $options['interfaceClass'] . ' dropdown"
id="selectedAssets" name="selectedAssets[]">
7308   <option value="' . get_the_id() . '"
' . ( in_array( get_the_id(), $contentIDs ) ? 'selected' : '' ) . '>Show</option>
7309   <option value="-' . get_the_id() . '"
' . ( ! in_array( get_the_id(), $contentIDs ) ? 'selected' : '' ) . '>Hide</option>
7310 </select>
7311                 </td>';
7312
7313                                                $content['assets'] .= '<td> <a
href="' . get_permalink() . '"><i class="box icon">
</i>' . get_the_title() . get_the_post_thumbnail( get_the_id(), array( 150, 150 ), array( 'class' => 'ali
gnright' ) ) . '</a> ' . ucwords( get_post_type() ) . ' </td>';
7314
7315                                                $content['assets'] .= '</tr>';
7316                                            endwhile;
7317                                            $content['assets'] .= '</tbody>
7318   <tfoot>
7319   <tr><th colspan="5">';
7320
7321                                            $content['assets'] .= '<div class="field
inline">  <button class="ui ' . $options['interfaceClass'] . ' button" name="save" id="save" value="save"
type="submit"><i class="cart icon"></i>' . __( 'Update Selection', 'ppv-live-webcams' ) . '</button>
</div>';
7322
7323                                            if ( $asset_page > 1 ) {
7324                                                $content['assets'] .= '<a class="ui
' . $options['interfaceClass'] . ' button" href="' . add_query_arg( 'asset_page', $asset_page -
 1, $this_page ) . '"> <i class="arrow left icon"></i>' . __( 'Previous Page', 'ppv-live-
webcams' ) . '</a>';
7325                                            }
7326                                            if ( $asset_page < $postslist->max_num_pages ) {
7327                                                $content['assets'] .= '<a class="ui
' . $options['interfaceClass'] . ' button"
href="' . add_query_arg( 'asset_page', $asset_page + 1, $this_page ) . '">' . __( 'Next Page', 'ppv-live-
webcams' ) . ' <i class="arrow right icon"></i> </a>';
7328                                            }
7329
7330                                            $content['assets'] .= '</th>
7331   </tr></tfoot></table>';
7332
7333                                            $content['assets'] .= '</form>';
```

```
7334
7335                                                    wp_reset_postdata();
7336                                                                        else :
7337 $content['assets'] .= 'You have no digital assets to list for this room.';
7338                                                                            endif;
7339
7340                                                                        // end products list
7341
7342                         }
7343                 }
7344
7345                                         // ! products tab in dashboard
7346
7347                 if ( $options['woocommerce'] ?? false ) {
7348                         if ( shortcode_exists( 'products' ) ) {
7349                                 if ( $_GET['products'] ?? false ) {
7350                                         $active = 'products';
7351                                 }
7352                                                 $header['products'] = __( 'Products', 'ppv-live-
webcams' );
7353
7354                                                 $content['products'] = __( 'Select products to show on
room profile page.', 'ppv-live-webcams' );
7355
7356                                                 $this_page = self::getCurrentURL();
7357
7358                                 if ( $_POST['save'] ?? false ) {
7359                                         if ( is_array( $_POST['selected'] ?? false ) ) {
7360 $productIDs = get_post_meta( $postID, 'productIDs', true );
7361                                                 if ( ! is_array( $productIDs ) ) {
7362                                                         $productIDs = array();
7363                                                 }
7364
7365                                                 foreach ( $_POST['selected'] as $selected ) {
7366                                                         $selectedInt = intval( $selected );
7367
7368                                                         $sPost = get_post( abs( $selectedInt ) );
7369                                                         if ( ! $sPost ) {
7370                                                                 return 'Post not found!';
7371                                                         }
7372                                                         if ( $sPost->post_author != $current_user->ID ) {
7373                                                                 return 'Not your post!';
7374                                                         }
7375
7376                                                         if ( $selectedInt > 0 ) {
7377 if ( ! in_array( $selectedInt, $productIDs ) ) {
7378                                                                         $productIDs[] = $selectedInt; //
add selected ID to show
7379                                                                 }
7380                                                         }
7381
7382                                                         if ( $selectedInt < 0 ) {
7383 if ( in_array( -$selectedInt, $productIDs ) ) {
7384 if ( ( $key = array_search( -$selectedInt, $productIDs ) ) !== false ) {
7385 unset( $productIDs[ $key ] ); // find and remove ID to hide
7386                                                                         }
7387                                                                 }
7388                                                         }
```

```
7389                                            }
7390
7391    update_post_meta( $postID, 'productIDs', $productIDs );
7392
7393                                                            $active = 'products';
7394
7395                                                            // $content['products'] .= '--' .
        serialize($_POST['selected']) . '--';
7396                                            }
7397                                    }
7398
7399                                            // current
7400    $productIDs = get_post_meta( $postID, 'productIDs', true );
7401                                    if ( ! is_array( $productIDs ) ) {
7402                                            $productIDs = array();
7403                                    }
7404                                            // $content['products'] .= '--' . serialize($productIDs)
        . '--';
7405
7406                                            $product_page = intval( $_GET['product_page'] ?? 0 );
7407                                    if ( $product_page ) {
7408                                            $active = 'products';
7409                                    }
7410
7411                                    if ( $product_page <= 0 ) {
7412                                            $product_page = 1;
7413                                    }
7414
7415                                            $args = array(
7416                                                    'author'         => $current_user->ID,
7417                                                    'post_type'      => 'product',
7418                                                    'orderby'        => 'modified',
7419                                                    'order'          => 'DESC',
7420                                                    'posts_per_page' => 8,
7421                                                    'paged'
        => $product_page ? $product_page : 1,
7422                                            );
7423
7424                                            $postslist = new \WP_Query( $args ); // WP root namespace
7425
7426                                            if ( $postslist->have_posts() ) :
7427
7428                                                    $content['products'] .= '<form class="ui form"
        method="post" enctype="multipart/form-data" action="' . $this_page . '"  name="adminForm">';
7429
7430                                                    // console.log(\'selectAll\');
7431
7432                                                    $content['products'] .= '<table class="ui celled
        striped table ' . $options['interfaceClass'] . '">
        <thead>
            <tr>
            <th> Select </th>
            <th>Asset</th>
            </tr>
        </thead>
        <tbody>';
7433
7434
7435
7436
7437
7438
7439
7440
7441                                                    while ( $postslist->have_posts() ) :
7442                                                            $postslist->the_post();
7443
7444                                                            $content['products'] .= '<tr>';
```

```
7445
7446                                                    $content['products'] .= '<td>
7447                                    <select class="ui ' . $options['interfaceClass'] . ' dropdown"
id="selected" name="selected[]">
7448  <option value="' . get_the_id() . '"
' . ( in_array( get_the_id(), $productIDs ) ? 'selected' : '' ) . '>Show</option>
7449  <option value="-' . get_the_id() . '"
' . ( ! in_array( get_the_id(), $productIDs ) ? 'selected' : '' ) . '>Hide</option>
7450 </select>
7451                 </td>';
7452
7453                                                    $content['products'] .= '<td> <a
href="' . get_permalink() . '"><i class="box icon">
</i>' . get_the_title() . get_the_post_thumbnail( get_the_id(), array( 150, 150 ), array( 'class' => 'ali
gnright' ) ) . '</a> </td>';
7454
7455                                                    $content['products'] .= '</tr>';
7456                                                endwhile;
7457                                                $content['products'] .= '</tbody>
7458  <tfoot>
7459  <tr><th colspan="5">';
7460
7461                                                $content['products'] .= '<div class="field
inline">  <button class="ui ' . $options['interfaceClass'] . ' button" name="save" id="save" value="save"
type="submit"><i class="cart icon"></i>' . __( 'Update Selection', 'ppv-live-webcams' ) . '</button>
</div>';
7462
7463                                                if ( $product_page > 1 ) {
7464                                                    $content['products'] .= '<a class="ui
' . $options['interfaceClass'] . ' button" href="' . add_query_arg( 'product_page', $product_page -
 1, $this_page ) . '"> <i class="arrow left icon"></i>' . __( 'Previous Page', 'ppv-live-
webcams' ) . '</a>';
7465                                                }
7466                                                if ( $product_page < $postslist-
>max_num_pages ) {
7467                                                    $content['products'] .= '<a class="ui
' . $options['interfaceClass'] . ' button"
href="' . add_query_arg( 'product_page', $product_page + 1, $this_page ) . '">' . __( 'Next Page', 'ppv-
live-webcams' ) . ' <i class="arrow right icon"></i> </a>';
7468                                                }
7469
7470                                                $content['products'] .= '</th>
7471  </tr></tfoot></table>';
7472
7473                                                $content['products'] .= '</form>';
7474
7475                                                wp_reset_postdata();
7476                                                                    else :
7477
$content['products'] .= 'You have no products to list for this room.';
7478                                                                        endif;
7479
7480                                                                    // end products list
7481
7482                         }
7483                 }
7484
7485                                    // ! pictures tab
7486                 if ( $options['picturegallery'] ) {
7487                         if ( shortcode_exists( 'videowhisper_postpictures' ) ) {
7488                                 if ( self::inList( $userkeys, $options['pictures'] ) ) {
7489
7490                                    /*
7491                                    $checked = '';
7492                                    if ($_GET['gallery_upload'] || $_GET['gallery_import']) $checked
= 'active';
```

```
Line
7493                                    if ($checked) $checked1 = true;
7494
7495                                    $headerCode .= '<a class="item ' . $checked .'" data-
tab="pictures">' . __('Pictures', 'ppv-live-webcams') . '</a>';
7496                                    $contentCode .= '<div class="ui ' . $options['interfaceClass'] .
' bottom attached tab segment ' . $checked .'" data-tab="pictures">';
7497                                    */
7498
7499                                    if ( isset($_GET['gallery_upload']) ||
 isset($_GET['gallery_import']) ) {
7500                                            $active = 'pictures';
7501                                    }
7502                                    $header['pictures'] = __( 'Pictures', 'ppv-live-webcams' );
7503
7504                                    /*
7505            $htmlCode .= '<div class="vwtab">
7506                                    <input type="radio" id="tab-pic" name="tab-group-1" '.$checked.'>
7507                                    <label class="vwlabel" for="tab-pic">Pictures</label>
7508
7509                                    <div class="vwpanel">
7510                                    <div class="vwcontent">';
7511                                    */
7512                                    $content['pictures']  = do_shortcode( "
[videowhisper_postpictures_process post=\"$postID\"]" );
7513                                    $content['pictures'] .= do_shortcode( "[videowhisper_postpictures
perpage=\"8\" post=\"$postID\"]" );
7514
7515                                    $content['pictures'] .= '<p>' . __( 'Multiple profile pictures
can show on webcam listing page. Each listing has own pictures.', 'ppv-live-webcams' ) . '</p>';
7516
7517                                    /*
7518                                    $contentCode .= '
7519                                    <br style="clear:both">
7520                                    </div>
7521                                    ';
7522                                    */
7523                                }
7524                        }
7525                }
7526
7527                                            // move this at bottom??
7528
$htmlCode .= apply_filters( 'vw_plw_dashboard', '', $postID );
7529
7530                                    $htmlCode .= apply_filters( 'vw_plw_dashboard', '', $postID );
7531
7532                if ( $atts['include_css'] ) {
7533
$htmlCode .= '<STYLE>' . html_entity_decode( stripslashes( $options['dashboardCSS'] ) ) . '</STYLE>';
7534                }
7535
7536                                    $htmlCode .= '<style type="text/css">
7537.pagination, .pagination > LI {
7538display: inline !important;
7539padding: 5px;
7540}
7541</style>';
7542
7543//! chat logs tab
7544$header['files'] = __( 'Logs', 'ppv-live-webcams' );
7545
7546if ( $fileRemove = sanitize_textarea_field( $_GET['fileRemove'] ?? '' ) ) {
7547        $active = 'files';
```

```
7548 }
7549
7550 $contentCode    = '';
7551
7552 $dir = $options['uploadsPath'];
7553 if ( ! file_exists( $dir ) ) {
7554         mkdir( $dir );
7555 }
7556
7557 $dir .= '/' . sanitize_file_name( $webcamPost->post_title );
7558 if ( ! file_exists( $dir ) ) {
7559         mkdir( $dir );
7560 }
7561
7562 $nonce = sanitize_text_field( $_GET['nonce'] ?? '' );
7563 if ( $fileRemove &&  wp_verify_nonce( $nonce, 'fileRemove' . $webcamPost->ID ) )
7564 {
7565         $fileName = base64_decode($fileRemove);
7566
7567         //make sure it does not contain path traversal
7568         if (strpos($fileName, '/') === false && strpos($fileName, '\\') === false) {
7569                 $filePath = $dir . '/' . $fileName;
7570                 if (file_exists($filePath)) {
7571                         unlink($filePath);
7572                         $contentCode .= '<p>' . __('Deleted', 'ppv-live-webcams') . '
' . $fileName . '</p>';
7573                 }
7574         }
7575
7576 } else {
7577         if ($fileRemove) $contentCode .= 'Invalid request!';
7578 }
7579
7580 //find chat log files of format  $dir . "/Log$day.html" and list with most recent first
7581 $files = glob( $dir . '/Log*.html' );
7582 if ( $files ) {
7583         rsort( $files );
7584 }
7585
7586 if ($options['chatlogCleanup'] ?? false) {
7587         //delete files older that cleanup period (days)
7588         $cleanup = $options['chatlogCleanup'];
7589         $now = time();
7590         foreach ($files as $file) {
7591                 $age = $now - filemtime($file);
7592                 if ($age > $cleanup * 24 * 60 * 60) {
7593                         unlink($file);
7594                         //remove from list
7595                         $key = array_search($file, $files);
7596                         if ($key !== false) {
7597                                 unset($files[$key]);
7598                         }
7599                 }
7600         }
7601 }
7602
7603 //list files with links to view (in new tab) and delete with wp_nonce
7604 if (count($files)) {
7605         $contentCode .= '<h3 class="ui header">' . __('Room Chat Logs', 'ppv-live-webcams') . '</h3>';
7606         foreach ($files as $file) {
7607                 $filename = basename($file);
```

```
7608                $age = time() - filemtime($file);
7609                $size = filesize($file) / 1000;
7610                //include wp_nonce for delete
7611                $encodedFilename = base64_encode($filename);
7612                $nonce = wp_create_nonce('fileRemove' . $webcamPost->ID);
7613                $contentCode .= '<a href="' . self::path2url($file). '"
target="_blank">' . $filename . '</a> | ' . self::format_age($age) . ' | ' . $size . 'kb | <a class="ui
small button"
href="' . add_query_arg(array('fileRemove' => $encodedFilename , 'nonce' => $nonce), $this_page) . '">del
ete</a><br>';
7614        }
7615} else {
7616        $contentCode .= __('No chat log files found in this room.', 'ppv-live-webcams') ;
7617}
7618
7619if ($options['chatlogCleanup'] ?? false) $contentCode .= '<p><small>*' . __('Older log files are
automatically deleted after a number of days:', 'ppv-live-webcams') . '
' . $options['chatlogCleanup'] . '</small></p>';
7620
7621
7622//also find files other that chat logs $dir . '/Log*.html' and list with most recent first
7623$files = glob( $dir . '/*' );
7624if ( $files ) {
7625        rsort( $files );
7626}
7627
7628//remove log files Log*.html (already listed)
7629foreach ($files as $key => $file) {
7630        if (strpos($file, 'Log') !== false) {
7631                unset($files[$key]);
7632        }
7633        //also remove folders
7634        if (is_dir($file)) {
7635                unset($files[$key]);
7636        }
7637}
7638
7639if (count($files)) {
7640        $contentCode .= '<h3 class="ui header">' . __('Room Uploads', 'ppv-live-webcams') . '</h3>';
7641        foreach ($files as $file) {
7642                $filename = basename($file);
7643                $age = time() - filemtime($file);
7644                $size = filesize($file) / 1000;
7645                //include wp_nonce for delete
7646                $encodedFilename = base64_encode($filename);
7647                $nonce = wp_create_nonce('fileRemove' . $webcamPost->ID);
7648                $contentCode .= '<a href="' . self::path2url($file). '"
target="_blank">' . $filename . '</a> | ' . self::format_age($age) . ' | ' . $size . 'kb | <a
href="' . add_query_arg(array('fileRemove' => $encodedFilename , 'nonce' => $nonce), $this_page) . '">del
ete</a><br>';
7649        }
7650} else {
7651        $contentCode .= __('No uploads found in this room.', 'ppv-live-webcams') ;
7652}
7653
7654//count total file size in $dir including subfolders
7655$size = 0;
7656foreach (new \RecursiveIteratorIterator(new \RecursiveDirectoryIterator($dir)) as $file) {
7657        $size += $file->getSize();
7658}
7659$contentCode .= '<br><p>' . __('Total room folder size:', 'ppv-live-webcams') . '
' . $size / 1000 . 'kb</p>';
7660
7661
```

```
7662 $content['files'] = $contentCode;
7663
7664                                         // ! bans tab
7665
7666                                         /*
7667                                         $checked = '';
7668                                         if ($banRemove = sanitize_textarea_field($_GET['banRemove'])) $checked = 'active';
7669                                         if ($checked) $checked1 = true;
7670
7671                                         $headerCode .= '<a class="item ' . $checked .'" data-tab="bans">' . __('Bans', 'ppv-live-webcams') . '</a>';
7672                                         $contentCode .= '<div class="ui ' . $options['interfaceClass'] . ' bottom attached tab segment ' . $checked .'" data-tab="bans">';
7673                                         */
7674
7675                 if ( $options['bans'] ?? false ) {
7676                         if ( $banRemove = sanitize_textarea_field( $_GET['banRemove'] ?? '' ) ) {
7677                                 $active = 'bans';
7678                         }
7679                         $header['bans'] = __( 'Bans', 'ppv-live-webcams' );
7680                         $contentCode    = '';
7681
7682                         $bans = get_post_meta( $postID, 'bans', true );
7683
7684                         $bansUpdate = false;
7685                         $banInfo = '';
7686
7687                         if ( $banRemove && $bans ) {
7688                                 if ( is_array( $bans ) ) {
7689                                         foreach ( $bans as $key => $ban ) {
7690 if ( $banRemove == base64_encode( $ban['user'] . $ban['ip'] . $ban['expires'] ) ) {
7691                                                 unset( $bans[ $key ] );
7692                                                 $bansUpdate = 1;
7693                                                 $banInfo   .= '<br>' . __( 'Cleared ban for', 'ppv-live-webcams' ) . ': ' . $ban['user'];
7694                                                 }
7695                                         }
7696                                 }
7697
7698                                 if ( ! $bansUpdate ) {
7699                                         $banInfo .= __( 'Not found!', 'ppv-live-webcams' );
7700                                 }
7701                         }
7702
7703                         // clean bans
7704                         if ( $bans ) {
7705                                 foreach ( $bans as $key => $ban ) {
7706                                         if ( $ban['expires'] < time() ) {
7707                                                 unset( $bans[ $key ] );
7708                                                 $bansUpdate = 1;
7709                                         }
7710                                 }
7711                         }
7712                         if ( $bansUpdate ) {
7713                                 update_post_meta( $postID, 'bans', $bans );
7714                         }
7715                                 /*
7716                                         $htmlCode .= '<div class="vwtab">
7717                                         <input type="radio" id="tab-bans" name="tab-group-1" '.$checked.'>
```

```
7718                                            <label class="vwlabel" for="tab-bans">' . __('Bans', 'ppv-live-webcams') . '</label>
7719
7720                                            <div class="vwpanel">
7721                                            <div class="vwcontent">';
7722                                            */
7723                                    $contentCode .= '<h3 class="ui header">' . __( 'Bans', 'ppv-live-webcams' ) . '</h3>' . $banInfo;
7724                                    if ( $bans ) {
7725                                            foreach ( $bans as $ban ) {
7726                                                    $contentCode .= '- ' . $ban['user'] . ' ' . $ban['ip'] . ' Expires: ' . self::format_age( $ban['expires'] - time() ) . ' <a href="' . add_query_arg( array( 'banRemove' => urlencode( base64_encode( $ban['user'] . $ban['ip'] . $ban['expires'] ) ) ), $this_page ) . '">remove</a><br>';
7727                                            }
7728                                    } else {
7729                                            $contentCode .= __( 'There are no bans for this room.', 'ppv-live-webcams' );
7730                                    }
7731
7732                                            $contentCode .= '<p>' . __( 'Bans occur when performer kicks or bans users from videochat interface. Kicks result in short bans (15 minutes) and full bans have longer cooldown (1 week). Bans prevent viewers from accessing or remaining in chat room.', 'ppv-live-webcams' ) . '</p>';
7733
7734                                            $content['bans'] = $contentCode;
7735                                            /*
7736                                            $contentCode .= '
7737                                            <br style="clear:both">
7738                                            </div>';
7739                                            */
7740                            }
7741
7742                                    // ! credits tabs
7743
7744                                    // update clients count
7745                                    $uid = get_current_user_id();
7746                                    self::billCount( $uid, $options );
7747
7748                                    self::billCountRoom( $postID, $options );
7749                                    self::billSessions( 0, $postID );
7750
7751                                    // MicroPayments Tab
7752                            if ( $options['performerWallet'] ?? false ) {
7753                                    if ( $options['wallet'] == 'MicroPayments' || $options['walletMulti'] >= 1 ) {
7754                                            if ( shortcode_exists( 'videowhisper_transactions' ) ) {
7755
7756                            if ( isset( $_GET['orderby'] ) || get_query_var( 'paged' ) ) {
7757                                            $active = 'MicroPayments';
7758                                    }
7759
7760                                    $header['MicroPayments'] = __( 'MicroPayments', 'ppv-live-webcams' );
7761
7762                                    $content['MicroPayments'] = '<h3 class="ui header">' . __( 'MicroPayments Transactions', 'ppv-live-webcams' ) . '</h3>' . do_shortcode( '[videowhisper_transactions user_id="' . $uid . '"]' );
7763
7764                                    $clientsP = get_user_meta( $uid, 'paidSessionsPrivate', true );
7765                                    $clientsG = get_user_meta( $uid, 'paidSessionsGroup', true );
7766                                    $content['MicroPayments'] .= '<br>' . __( 'Logged days', 'ppv-live-webcams' ) . ': ' . round( $options['ppvKeepLogs'] / 3600 / 24, 2 );
```

```
7767                                    $content['MicroPayments'] .= '<br>' . __( 'Logged private sessions billed', 'ppv-live-webcams' ) . ': ' . $clientsP;
7768                                    $content['MicroPayments'] .= '<br>' . __( 'Logged group sessions billed', 'ppv-live-webcams' ) . ': ' . $clientsG;
7769                                    $content['MicroPayments'] .= '<br>* ' . __( 'Session logs are kept for limited time and transaction logs forever.', 'ppv-live-webcams' );
7770
7771                                }
7772                        }
7773                }
7774
7775
7776                // WooWallet Tab
7777                if ( $options['performerWallet'] ) {
7778                        if ( $options['wallet'] == 'WooWallet' || $options['walletMulti'] >= 1 ) {
7779                                if ( shortcode_exists( 'woo-wallet' ) ) {
7780                                        /*
7781                                        $checked = '';
7782                                        if ($_GET['wallet_action'] || get_query_var('wallet_action')) $checked = 'active';
7783                                        if ($checked) $checked1 = true;
7784
7785                                        $headerCode .= '<a class="item ' . $checked .'" data-tab="woowallet">' . __('TeraWallet', 'ppv-live-webcams') . '</a>';
7786                                        $contentCode .= '<div class="ui ' . $options['interfaceClass'] . ' bottom attached tab segment ' . $checked .'" data-tab="woowallet">';
7787
7788                                        $contentCode .= '<h3 class="ui header">' . __('TeraWallet (WooCommerce)', 'ppv-live-webcams') . '</h3>' . do_shortcode('[woo-wallet]') ;
7789                                        */
7790
7791                                        if ( isset( $_GET['wallet_action'] ) || get_query_var( 'wallet_action' ) ) {
7792                                                $active = 'woowallet';
7793                                        }
7794                                        $header['woowallet'] = __( 'TeraWallet', 'ppv-live-webcams' );
7795
7796                                        $content['woowallet'] = '<h3 class="ui header">' . __( 'TeraWallet (WooCommerce)', 'ppv-live-webcams' ) . '</h3>' . do_shortcode( '[woo-wallet]' );
7797
7798                                        $clientsP = get_user_meta( $uid, 'paidSessionsPrivate', true );
7799                                        $clientsG = get_user_meta( $uid, 'paidSessionsGroup', true );
7800                                        $content['woowallet'] .= '<br>' . __( 'Logged days', 'ppv-live-webcams' ) . ': ' . round( $options['ppvKeepLogs'] / 3600 / 24, 2 );
7801                                        $content['woowallet'] .= '<br>' . __( 'Logged private sessions billed', 'ppv-live-webcams' ) . ': ' . $clientsP;
7802                                        $content['woowallet'] .= '<br>' . __( 'Logged group sessions billed', 'ppv-live-webcams' ) . ': ' . $clientsG;
7803                                        $content['woowallet'] .= '<br>* ' . __( 'Session logs are kept for limited time and transaction logs forever.', 'ppv-live-webcams' );
7804
7805                                        /*
7806                                        $contentCode .= '
7807                                        <br style="clear:both">
7808                                        </div>';
7809                                        */
7810                                }
7811                        }
7812                }
7813
7814                                        // MyCred Transactions
7815                if ( $options['performerWallet'] ) {
7816                        if ( $options['wallet'] == 'MyCred' || $options['walletMulti'] >= 1 ) {
```

```
7817                                if ( shortcode_exists( 'mycred_history' ) ) {
7818                                        /*
7819                                        $checked = '';
7820                                        if ($_GET['page'] || get_query_var('page')) $checked = 'active';
7821                                        if ($checked) $checked1 = true;
7822
7823                                        $headerCode .= '<a class="item ' . $checked .'" data-tab="transactions">' . __('My Credits', 'ppv-live-webcams') . '</a>';
7824                                        $contentCode .= '<div class="ui ' . $options['interfaceClass'] . ' bottom attached tab segment ' . $checked .'" data-tab="transactions">';
7825
7826                                        $contentCode .= '<h3 class="ui header">' . __('My Credits', 'ppv-live-webcams') . '</h3>' . do_shortcode('[mycred_history user_id="current"]') ;
7827                                        */
7828                                        if ( isset($_GET['page']) || get_query_var( 'page' ) ) {
7829                                                $active = 'mycred';
7830                                        }
7831                                        $header['mycred'] = __( 'MyCred', 'ppv-live-webcams' );
7832
7833                                        $content['mycred'] = '<h3 class="ui header">' . __( 'My Credits', 'ppv-live-webcams' ) . '</h3>' . do_shortcode( '[mycred_history user_id="current"]' );
7834
7835                                        $clientsP = get_user_meta( $uid, 'paidSessionsPrivate', true );
7836                                        $clientsG = get_user_meta( $uid, 'paidSessionsGroup', true );
7837
7838                                        $content['mycred'] .= '<br>' . __( 'Logged days', 'ppv-live-webcams' ) . ': ' . round( $options['ppvKeepLogs'] / 3600 / 24, 2 );
7839                                        $content['mycred'] .= '<br>' . __( 'Logged private sessions billed', 'ppv-live-webcams' ) . ': ' . $clientsP;
7840                                        $content['mycred'] .= '<br>' . __( 'Logged group sessions billed', 'ppv-live-webcams' ) . ': ' . $clientsG;
7841                                        $content['mycred'] .= '<br>* ' . __( 'Session logs are kept for limited time and transaction logs forever.', 'ppv-live-webcams' );
7842
7843                                        /*
7844                                        $contentCode .= '
7845                                        <br style="clear:both">
7846                                        </div>';
7847                                        */
7848                                }
7849                        }
7850                }
7851
7852
7853
7854                //reports
7855
7856                if ($options['performerReports'] ?? false )
7857                {
7858                if ( isset( $_POST['deltaDate'] ) || get_query_var( 'deltaDate' ) ) $active = 'reports';
7859
7860                if ( ! $active ) $active = 'reports';
7861
7862                $header['reports'] =  __( 'Reports', 'ppv-live-webcams' ) ;
7863                $content['reports'] = '<h3 class="ui header">' . __( 'Reports', 'ppv-live-webcams' ) . '</h3>';
7864
7865
if ($options['statsBonusRate'] && $options['statsBonusOnline']) $content['reports'] .= do_shortcode( '[videowhisper_room_rate roomid="'. $postID . '"]' );
7866
7867                $content['reports'] .= do_shortcode( '[videowhisper_reports roomid="'. $postID . '"]' );
7868
7869
```

```
Line
7870                                    // ! overview tab
7871                                    /*
7872                                    $checked = '';
7873                                    if (!$checked1) $checked = 'active';
7874
7875
7876                                    $headerCode .= '<a class="item ' . $checked .'" data-
tab="overview">' . __('Overview', 'ppv-live-webcams') . '</a>';
7877                                    $contentCode .= '<div class="ui ' . $options['interfaceClass'] .
' bottom attached tab segment ' . $checked .'" data-tab="overview">';
7878
7879                                    */
7880                }
7881
7882                if ( $options['performerOverview'] ) {
7883
7884                        $header['overview'] = __( 'Overview', 'ppv-live-webcams' );
7885                        if ( ! $active ) $active = 'overview';
7886
7887                        $contentCode = '';
7888                        /*
7889                        $htmlCode .= '<div class="vwtab">
7890                        <input type="radio" id="tab-1" name="tab-group-1" '.$checked.'>
7891                        <label class="vwlabel" for="tab-1">' . __('Overview', 'ppv-live-webcams') .
'</label>
7892
7893                        <div class="vwpanel">
7894                        <div class="vwcontent">
7895                        ';
7896                        */
7897                        $thumbFilename = '';
7898
7899                        $attach_id = get_post_thumbnail_id( $postID );
7900                        if ( $attach_id ) {
7901                                $thumbFilename = get_attached_file( $attach_id );
7902                        }
7903                        $url = get_permalink( $postID );
7904
7905                        $noCache = '?' . ( floor( time() / 10 ) % 100 );
7906
7907                        $contentCode .= '<h3 class="ui header">' . ucwords($options['custom_post']) . '
'. __( 'Room Listing Overview', 'ppv-live-webcams' ) . '</h3>';
7908
7909                        // $contentCode .= $webcamPost->post_content;
7910
7911                        // get/set performer stream name
7912                        $stream        = get_post_meta( $postID, 'performer', true );
7913                        $performerName = self::performerName( $current_user, $options );
7914
7915                        if ( ! $stream ) {
7916                                update_post_meta( $postID, 'performer', $performerName );
7917                                update_post_meta( $postID, 'performerUserID', $current_user->ID );
7918
7919                        }
7920
7921                        $imageCode = '';
7922
7923                        if ($thumbFilename) if ( file_exists( $thumbFilename ) ) {
7924                                $imageCode .= '<a href="' . $url . '"><IMG class="ui small rounded
bordered image right floated" src="' . self::path2url( $thumbFilename ) . $noCache . '"></a>';
7925                        } else {
7926                                $imageCode .= '<a href="' . $url . '"><IMG class="ui small rounded
bordered image right floated" SRC="' . dirname( plugin_dir_url( __FILE__ ) ) . '/no-picture.png"></a>';
```

```
7927                        }
7928
7929                        if (!self::isPerformer(get_current_user_id(), $webcamPost->ID)) $contentCode .= 'Warning: You are not detected as performer!';
7930
7931                        $contentCode .= '<table class="ui ' . $options['interfaceClass'] . ' selectable striped table two">';
7932
7933                        $contentCode .= '<tr><td>' . __( 'Listing (Room) Name', 'ppv-live-webcams' ) . ':<b>' . $webcamPost->post_title . '</b>' . $imageCode . '</td></tr>';
7934                        $contentCode .= '<tr><td>' . __( 'Current Room Performer', 'ppv-live-webcams' ) . ': ' . $stream . '</td></tr>';
7935                        $contentCode .= '<tr><td>' . __( 'Current Room Performer ID', 'ppv-live-webcams' ) . ': ' . get_post_meta( $postID, 'performerUserID', true ) . '</td></tr>';
7936                        $contentCode .= '<tr><td>' . __( 'Your Performer Name', 'ppv-live-webcams' ) . ': ' . $performerName . '</td></tr>';
7937                        $contentCode .= '<tr><td>' . __( 'Room Label', 'ppv-live-webcams' ) . ': ' . ($vw_roomLabel ?? '') . '</td></tr>';
7938                        $contentCode .= '<tr><td>' . __( 'Room Status', 'ppv-live-webcams' ) . ': ' . $webcamPost->post_status . '</td></tr>';
7939                        $contentCode .= '<tr><td>' . __( 'Room URL', 'ppv-live-webcams' ) . ': ' . $url . '</td></tr>';
7940                        $contentCode .= '<tr><td>' . __( 'Room Slug', 'ppv-live-webcams' ) . ': ' . $webcamPost->post_name . '</td></tr>';
7941                        $contentCode .= '<tr><td>' . __( 'Room ID', 'ppv-live-webcams' ) . ': ' . $webcamPost->ID . '</td></tr>';
7942                        $contentCode .= '<tr><td>' . __( 'Room Owner ID', 'ppv-live-webcams' ) . ': ' . $webcamPost->post_author . '</td></tr>';
7943
7944                        $featured = get_post_meta( $postID, 'vw_featured', true );
7945                        if ( empty( $featured ) ) {
7946                                update_post_meta( $postID, 'vw_featured', 0 );
7947                        }
7948                        $contentCode .= '<tr><td>' . __( 'Featured', 'ppv-live-webcams' ) . ': ' . ( $featured ? __( 'Yes', 'ppv-live-webcams' ) : __( 'No', 'ppv-live-webcams' ) ) . '</td></tr>';
7949
7950                        if ( $CPM = self::clientCPM( $webcamPost->post_title, $options, $postID ) ) {
7951                                $contentCode .= '<tr><td>' . __( 'Client CPM', 'ppv-live-webcams' ) . ': ' . $CPM . '</td></tr>';
7952                                $contentCode .= '<tr><td>' . __( 'Earning Ratio', 'ppv-live-webcams' ) . ': ' . self::performerRatio( $webcamPost->post_title, $options, $postID ) * 100 . '%' . '</td></tr>';
7953                        }
7954
7955                        if ( $options['videosharevod'] ) {
7956                                if ( shortcode_exists( 'videowhisper_videos' ) ) {
7957                                                $vw_videos = get_post_meta( $postID, 'vw_videos', true );
7958                                                $contentCode .= '<tr><td>' . __( 'Videos', 'ppv-live-webcams' ) . ': ' . ( $vw_videos ? __( 'Yes', 'ppv-live-webcams' ) : __( 'No', 'ppv-live-webcams' ) ) . '</td></tr>';
7959                                }
7960                        }
7961
7962                        if ( $options['picturegallery'] ) {
7963                                if ( shortcode_exists( 'videowhisper_pictures' ) ) {
7964                                                $vw_pictures = get_post_meta( $postID, 'vw_pictures', true );
7965                                                $contentCode .= '<tr><td>' . __( 'Pictures', 'ppv-live-webcams' ) . ': ' . ( $vw_pictures ? __( 'Yes', 'ppv-live-webcams' ) : __( 'No', 'ppv-live-webcams' ) ) . '</td></tr>';
7966                                }
7967                        }
7968                        /*
7969                        if ($options['transcoding'])
7970                        {
7971                        $vw_transcode = get_post_meta($postID, 'vw_transcode',true);
```

```
7972                        $contentCode .= '<tr><td>' . __('HTML5 Transcoding', 'ppv-live-webcams') . ': ' . ($vw_transcode?__('Yes', 'ppv-live-webcams'):__('No', 'ppv-live-webcams')) . '</td></tr>';
7973                        }
7974
7975
7976                        if ($options['multicamMax'])
7977                        {
7978                        $vw_multicam = get_post_meta($postID, 'vw_multicam', true);
7979                        $contentCode .= '<tr><td>' . __('Multiple Cameras', 'ppv-live-webcams') . ': ' . ($vw_multicam?"Yes ($vw_multicam extra)":'No') . '</td></tr>';
7980                        }
7981                        */
7982
7983                        if ( $options['playlists'] ) {
7984                                if ( self::inList( $userkeys, $options['schedulePlaylists'] ) ) {
7985                                $vw_playlistActive = get_post_meta( $postID, 'vw_playlistActive', true );
7986                                $contentCode      .= '<tr><td>' . __( 'Scheduled Playlist', 'ppv-live-webcams' ) . ': ' . ( $vw_playlistActive ? __( 'Yes', 'ppv-live-webcams' ) : __( 'No', 'ppv-live-webcams' ) ) . '</td></tr>';
7987                                }
7988                        }
7989
7990                        $contentCode .= '<tr><td>' . __( 'Description', 'ppv-live-webcams' ) . ': ' . $webcamPost->post_content . '</td></tr>';
7991
7992                        $contentCode .= '</table>';
7993
7994                        $content['overview'] = $contentCode;
7995                }
7996                                        /*
7997                                        $contentCode .= '
7998                                        <br style="clear:both">
7999                                        </div>
8000                                        ';
8001                                        */
8002
8003                                        /*
8004                                        //build tabs
8005                                        $headerCode .= '</div>';
8006                                        $htmlCode .= $headerCode . $contentCode ;
8007                                        */
8008
8009                        if ( ! $active ) if ( $options['performerSetup'] ) $active = 'setup';
8010
8011                $htmlCode .= self::sectionsLayout( $header, $content, $active, $options['performerLayout'], $webcamPost->post_title );
8012
8013                if ( $options['dashboardMessageBottom'] ) {
8014                        $htmlCode .= '<div id="performerDashboardMessageBottom" class="ui ' . $options['interfaceClass'] . ' segment">' . html_entity_decode( stripslashes( $options['dashboardMessageBottom'] ) ) . '</div>';
8015                }
8016
8017                                        $htmlCode .= '<br style="clear:both">';
8018
8019                                        return $htmlCode;
8020        }
8021
8022
8023        // end performer dashboard
8024
8025        static function generateName( $fn ) {
```

```
8026                $ext = strtolower( pathinfo( $fn, PATHINFO_EXTENSION ) );
8027
8028                // unpredictable name
8029                return md5( uniqid( $fn, true ) ) . '.' . $ext;
8030        }
8031
8032
8033        static function videowhisper_cam_message( $atts ) {
8034                // send a (paid) question or message to webcam profile
8035
8036                self::enqueueUI();
8037                $htmlCode = '';
8038
8039                $options = self::getOptions();
8040
8041                if ( ! is_user_logged_in() ) {
8042                        return '<div class="ui ' . $options['interfaceClass'] . ' header"> <i class="sign in icon"></i> ' . __( 'Login to send questions or messages!', 'ppv-live-webcams' ) . '</div>' . do_shortcode( '[videowhisper_login_form register="1" role="' . $options['roleClient'] . '"]' );
8043                }
8044                // return __('Login to send questions or messages!','ppv-live-webcams') . '<BR><a class="ui button primary qbutton" href="' . wp_login_url() . '">' . __('Login', 'ppv-live-webcams') . '</a>  <a class="ui button secondary qbutton" href="' . wp_registration_url() . '">' . __('Register', 'ppv-live-webcams') . '</a>';
8045
8046                $atts = shortcode_atts(
8047                        array(
8048                                'post_id' => 0, // for setup
8049                        ),
8050                        $atts,
8051                        'videowhisper_cam_message'
8052                );
8053                $atts = array_map('esc_attr', $atts);
8054
8055                // room id
8056                $postID = intval( $atts['post_id'] );
8057                if ( ! $postID ) {
8058                        $postID = intval( $_GET['webcam_id'] ); // pass by URL
8059                }
8060
8061                if ( $postID ) {
8062                        $post = get_post( $postID );
8063                }
8064                if ( ! $post ) {
8065                        $postID = 0;
8066                }
8067
8068                if ( ! $postID ) {
8069                        return 'Error: Could not find this webcam room! #' . $postID;
8070                }
8071
8072                $room = sanitize_file_name( $post->post_title );
8073
8074                $sender_id = get_current_user_id();
8075                $balance   = self::balance( $sender_id );
8076
8077                $messagesCost = floatval( get_post_meta( $postID, 'question_price', true ) );
8078                if ( ! $messagesCost ) {
8079                        $messagesCost = floatval( $options['messagesCost'] );
8080                }
8081
8082                $showNew = 1;
```

```
8083
8084                $closed = intval( get_post_meta( $postID, 'question_closed', true ) );
8085
8086                if ( $closed ) {
8087                        return '<div id="warningMessage" class="ui ' . $options['interfaceClass'] . '
message warning">' . __( 'Messages are currently closed.', 'ppv-live-webcams' ) . '</div>';
8088                }
8089
8090                if ( $_POST['newpaidmessage'] ?? false ) {
8091                        $showNew = 0;
8092
8093                        if ( $balance < $messagesCost ) {
8094                                $htmlCode .= '<div id="warnMessages" class="ui
' . $options['interfaceClass'] . ' segment">' . __( 'You need cost of message in balance to send
it.', 'ppv-live-webcams' ) . '<br>' . $messagesCost . '/' . $balance . '<br><a class="ui button primary
qbutton" href="' . get_permalink( $options['balancePage'] ) . '">' . __( 'Wallet', 'ppv-live-
webcams' ) . '</a>' . '</div>';
8095                        } else {
8096                                global $wpdb;
8097                                $table_messages = $wpdb->prefix . 'vw_vmls_messages';
8098
8099                                $ztime       = time();
8100
$paidmessage = wp_encode_emoji( stripslashes( sanitize_textarea_field( $_POST['paidmessage'] ) ) );
8101
8102                                $sdate = intval( $wpdb->get_var( $wpdb->prepare( "SELECT MAX(sdate) FROM
`$table_messages` WHERE sender_id=%d", $sender_id ) ) );
8103                                if ( $sdate > $ztime - 10 ) {
8104                                        return 'Error: You just sent a paid message or shortcode was
already executed. Try again after 10s.'; // 10s
8105                                }
8106
8107                                $meta                = array();
8108                                $meta['attachments'] = array();
8109
8110                                // Attachments
8111                                $countfiles = 0;
8112                                if ( ! empty( $_FILES ) ) {
8113                                        if ( $_FILES['attachments'] ) {
8114                                                $countfiles = count( $_FILES['attachments'] );
8115                                        }
8116                                }
8117
8118                                if ( $options['attachment_extensions'] ) {
8119                                        if ( $countfiles ) {
8120                                                $dir = $options['uploadsPath'];
8121                                                if ( ! file_exists( $dir ) ) {
8122                                                        mkdir( $dir );
8123                                                }
8124
8125                                                $dir .= "/$room";
8126                                                if ( ! file_exists( $dir ) ) {
8127                                                        mkdir( $dir );
8128                                                }
8129
8130                                                $dir .= '/_attachments';
8131                                                if ( ! file_exists( $dir ) ) {
8132                                                        mkdir( $dir );
8133                                                }
8134
8135                                                $dir .= '/';
8136
8137                                                // Looping all files
```

```
8138 for ( $i = 0;$i < $countfiles;$i++ ) if ( isset($_FILES['attachments']['name'][ $i ]) ) {
8139
8140                                                     $filename = $_FILES['attachments']['name'][ $i ];
8141                                                     if ( ! $filename ) {
8142                                                             break;
8143                                                     }
8144
8145 $ext = strtolower( pathinfo( $filename, PATHINFO_EXTENSION ) );
8146
8147 if ( ! in_array( $ext, self::extensions_attachment() ) ) {
8148                                                             return '<div class="ui message
error">' . __( 'This attachment extension is not allowed:', 'ppv-live-webcams' ) . '
' . $ext . '<br>' . __( 'Allowed attachment extensions:', 'ppv-live-webcams' ) . '
' . $options['attachment_extensions'] . '<br>' . __( 'Try again:', 'ppv-live-webcams' ) . '
' . $paidmessage . '</div>';
8149                                                     }
8150
8151                                                     if ( ! file_exists( $_FILES['attachments']
['tmp_name'][ $i ] ) ) {
8152                                                             return 'Attachment upload failed. Report
to webmaster and include link in message, until sorted out.';
8153                                                     }
8154
8155                                                     if ( $options['attachmentSize'] ) {
8156                                                             if ( filesize( $_FILES['attachments']
['tmp_name'][ $i ] ) > $options['attachmentSize'] ) {
8157       return '<div class="ui message error">' . __( 'This attachment is too big:', 'ppv-live-
webcams' ) . ' ' . $filename . '<br>' . __( 'Allowed attachment size:', 'ppv-live-webcams' ) . '
' . $options['attachmentSize'] . '<br>' . __( 'Try again:', 'ppv-live-webcams' ) . '
' . $paidmessage . '</div>';
8158                                                             }
8159                                                     }
8160
8161                                                     // Upload file
8162                                                     $path = $dir . self::generateName( $filename );
8163
8164                                                     $meta['attachments'][ $i ] = array(
8165 'name' => sanitize_text_field( $filename ),
8166                                                             'file' => $path,
8167 'size' => filesize( $_FILES['attachments']['tmp_name'][ $i ] ),
8168                                                     );
8169
8170                                                     //per file
8171                                                     $file = array(
8172                                                         'name' => $_FILES['attachments']
['name'][ $i ],
8173                                                         'type' => $_FILES['attachments']
['type'][ $i ],
8174                                                         'tmp_name' => $_FILES['attachments']
['tmp_name'][ $i ],
8175                                                         'error' => $_FILES['attachments']
['error'][ $i ] ,
8176                                                         'size' => $_FILES['attachments']
['size'][ $i ]
8177                                                     );
8178
8179 $errorUp = self::handle_upload( $file, $path ); // handle trough wp_handle_upload()
8180                                                     if ( $errorUp ) {
8181                                                             $htmlCode .= '<br>' . 'Error uploading
' . esc_html( $filename . ':' . $errorUp );
```

```
8182                                                    }
8183                                                }
8184                                        }
8185                                }
8186
8187                                // message sender
8188                                $sender = wp_get_current_user();
8189
8190                                // recordings
8191                                $meta['recordings'] = array();
8192
8193                                if ( $options['attachmentRecord'] ) {
8194 $recordings = sanitize_text_field( $_POST['recordings'] ?? false );
8195                                        if ( $recordings ) {
8196
8197 $optionsRecorder = get_option( 'VWinlineRecorderOptions' );
8198                                                $recordingFiles  = explode( ',', $recordings );
8199
8200                                                foreach ( $recordingFiles as $recordingFile ) {
8201                                                        if ( $recordingFile ) {
8202 if ( file_exists( $filePath = $optionsRecorder['uploadsPath'] . '/' . $sender-
>user_login . '/' . $recordingFile ) ) {
8203                                                                $meta['recordings'][] = array(
8204 'name' => sanitize_text_field( $recordingFile ),
8205                                                                        'file' => $filePath,
8206 'size' => filesize( $filePath ),
8207                                                                );
8208                                                        } else {
8209                                                                $htmlCode .= ' Recording file not
found: ' . $filePath;
8210                                                        }
8211                                                }
8212                                        }
8213                                }
8214                        }
8215
8216                        $meta['cost'] = $messagesCost;
8217
8218                        $metaS = serialize( $meta );
8219                        $sqlI = $wpdb->prepare(
8220                                "INSERT INTO `$table_messages` ( `sender_id`, `webcam_id`,
`reply_id`, `sdate`, `message`, `meta`) VALUES ( %d, %d, %d, %d, %s, %s)",
8221                                $sender_id, $postID, 0, $ztime, $paidmessage, $metaS
8222                        );
8223
8224                        $wpdb->query( $sqlI );
8225                        $messageID = $wpdb->insert_id;
8226
8227                        if ( ! $messageID ) {
8228                                return 'Error: Sending message failed! ' . $sqlI;
8229                        }
8230
8231                                // notification to room author
8232                                $author = get_userdata( $post->post_author );
8233                                $link
 = get_permalink( $options['p_videowhisper_webcams_perfomer'] ?? 0 );
8234                        if ( $author ) {
8235                                $mailed = wp_mail( $author-
>user_email, $options['messagesSubject'] . ' ' . ( $sender->display_name ? $sender-
```

```
Line
>display_name : $sender->user_login ), $options['messagesText'] . "\r\n" . $link);
8236
8237                                    //SMS notification
8238                                    if ($options['sms_number'])
8239                                    {
8240                                            $sms_number = get_user_meta( $author->ID, 'sms_number', true );
8241                                            if ($sms_number)
8242                                            {
8243                                                    //cooldown
8244                                                    $sms_last =  get_user_meta( $author->ID, 'sms_last', true );
8245                                                    if (!$sms_last || $sms_last < ( time() - $options['sms_cooldown']) )
8246                                                    {
8247
8248                                                    $sms_message = $options['messagesSubject'] . ' ' . ( $sender->display_name ? $sender->display_name : $sender->user_login ) . ' ' . $link ;
8249
8250                                                    if ($options['wp_sms']) if ( function_exists('wp_sms_send') ) wp_sms_send($sms_number,  $sms_message);
8251
8252                                                    if ($options['wp_sms_twilio']) if ( function_exists('twl_send_sms') )
8253                                                    {
8254                                                    $response = twl_send_sms( array( 'number_to' => $sms_number, 'message' => $sms_message ) );
8255                                                    if( is_wp_error( $response ) ) $htmlCode .= '<div class="ui message"> SMS Error: ' . esc_html( $response->get_error_message() ) . '</div>';
8256                                                    }
8257
8258                                                    update_user_meta( $author->ID, 'sms_last', time() );
8259                                                    }
8260                                            }
8261                                    }
8262
8263                                } else {
8264                                        $htmlCode .= 'Error: User not found to notify!';
8265                                }
8266
8267                                if ( ! $mailed ) {
8268                                        $htmlCode .= 'Warning: Notification email was not sent. Check WordPress mailing settings!';
8269                                }
8270
8271                                // pay for message
8272                                if ( $messagesCost ) {
8273                                        self::transaction( $ref = 'paid_message', $sender_id, -$messagesCost, __( 'Paid Message to', 'ppv-live-webcams' ) . ' <a href="' . self::roomURL( $post->post_name ) . '">' . $post->post_title . '</a>', $messageID, $paidmessage, $options );
8274                                }
8275
8276                                $messageURL = add_query_arg(
8277                                        array(
8278                                                'view'    => 'messages',
8279                                                'msg' => self::to62( $messageID ),
8280                                        ),
8281                                        get_permalink( $postID )
8282                                );
8283
8284                                $htmlCode .= '<div id="successMessage" class="ui ' . $options['interfaceClass'] . ' segment">' . __( 'Your message was sent.', 'ppv-live-webcams' ) .
```

```
8285                                    '<br><a class="ui button primary qbutton"
href="' . $messageURL . '">' . __( 'View Message', 'ppv-live-webcams' ) . ' ' . '</a> ' . '</div>';
8286
8287                            }
8288                    }
8289
8290                    if ( $showNew ) {
8291                            $this_page = self::getCurrentURL();
8292                            $action    = add_query_arg(
8293                                    array(
8294                                            'view'     => 'messages',
8295                                            'messages' => 'add',
8296                                    ),
8297                                    $this_page
8298                            );
8299
8300                            $htmlCode .= '<form method="post" enctype="multipart/form-data"
action="' . $action . '" name="messageForm" class="ui form ' . $options['interfaceClass'] . ' segment">
8301            <div class="field"><textarea class="ui field fluid" name="paidmessage" id="paidmessage"
placeholder="' . __( 'Your Question or Message', 'ppv-live-webcams' ) . '"></textarea> </div>';
8302
8303                            $htmlCode .= ( $options['attachment_extensions'] ? '
8304            <div class="field"><label><i class="attach icon"></i> ' . __( 'Attachments (Optional)', 'ppv-
live-webcams' ) . '</label>
8305            <input type="file" name="attachments[]" id="attachments" multiple placeholder="Attachments"><br>
<small>' . __( 'Supports', 'ppv-live-webcams' ) . ' : ' . $options['attachment_extensions'] . '</small>
8306            </div>
8307            ' : '' );
8308
8309                            if ( $options['attachmentRecord'] ) {
8310                                    if ( shortcode_exists( 'videowhisper_recorder_inline' ) ) {
8311                                            if ( isset($_GET['view']) && $_GET['view'] == 'messages' ) {
8312
$htmlCode .= do_shortcode( '[videowhisper_recorder_inline label="Attach Recordings (Optional)"]' );
8313                                            } else {
8314                                                    $htmlCode .= '<div class="field">' . __( 'Open in full
view to add video recordings:', 'ppv-live-webcams' ) . '  <a class="ui button tiny compact"
href="' . add_query_arg( 'view', 'messages', get_permalink( $postID ) ) . '">' . __( 'Messages', 'ppv-
live-webcams' ) . '</a> </div>';
8315                                            }
8316                                    }
8317                            }
8318
8319                            $htmlCode .= '<div class="field"><input class="ui button videowhisperButton"
type="submit" name="newpaidmessage" id="newpaidmessage" value="' . __( 'Send Message', 'ppv-live-
webcams' ) . '" /> </div>

8320
8321            <input name="post_id" id="post_id" type="hidden" value="' . $postID . '">
8322            <div class="ui message small">'
8323
. ( shortcode_exists( 'videowhisper_wallet' ) ? do_shortcode( '[videowhisper_wallet]' ) : __( 'Your
current balance', 'ppv-live-webcams' ) . ': ' . $balance . ' ' . $options['currency'] )
8324                                    . __( 'Cost per question or message', 'ppv-live-webcams' ) . ':
' . $messagesCost . ' ' . $options['currency'] . '
8325            </div>
8326            </form>';
8327                    }
8328
8329                    return $htmlCode;
8330
8331            }
8332
8333
8334
8335            static function messageReplies( $messageCode ) {
```

```
8336
8337                $id = self::to10( $messageCode );
8338
8339                if ( ! $id ) {
8340                        return 'Error: Incorrect message code! #' . $messageCode;
8341                }
8342
8343                global $wpdb;
8344                $table_messages = $wpdb->prefix . 'vw_vmls_messages';
8345
8346                $sqlr     = $wpdb->prepare("SELECT * FROM $table_messages WHERE id=%d ORDER BY sdate ASC,
id DESC LIMIT 0, 100", $id);
8347                $question = $wpdb->get_row( $sqlr );
8348                if ( ! $question ) {
8349                        return 'Error: Message not found! #' . $id;
8350                }
8351
8352                $post = get_post( $question->webcam_id );
8353                if ( ! $question ) {
8354                        return 'Error: Room listing not found! #' . $questio->webcam_id;
8355                }
8356
8357                $options = self::getOptions();
8358
8359                $htmlCode = '<div class="ui segment ' . $options['interfaceClass'] . '">';
8360                $htmlCode .= '<h4>' . ' <i class="users icon big"></i> ' . $post->post_title . ' / ' . '
#' . $messageCode . '</h4>';
8361                // list messages
8362
8363                $sqlRs = $wpdb->prepare("SELECT * FROM $table_messages WHERE id=%d OR reply_id=%d ORDER
BY sdate ASC, id DESC LIMIT 0, 100", $id, $id);
8364
8365                $results = $wpdb->get_results( $sqlRs );
8366                if ( $wpdb->num_rows > 0 ) {
8367                        $this_page = self::getCurrentURL();
8368
8369                        $htmlCode .= '<div class="ui list">';
8370
8371                        foreach ( $results as $message ) {
8372                                $sender = get_userdata( $message->sender_id );
8373
8374                                $htmlCode .= '<div class="item"> <i class="mail icon big"></i>';
8375                                $htmlCode .= '<div class="content">';
8376                                $htmlCode .= '<div class="header">' . ' <i class="user icon">
</i>' . $sender->user_nicename . ' <i class="clock icon"></i>' . date( DATE_RFC2822, $message-
>sdate ) . '</div>';
8377                                $htmlCode .= '<div class="ui segment
' . $options['interfaceClass'] . '">' . stripslashes( $message->message );
8378
8379                                $attachmentsCount = 0;
8380                                $recordingsCount  = 0;
8381                                if ( $message->meta ) {
8382                                        $meta = unserialize( $message-
>meta, array( 'allowed_classes' => false ) );
8383                                        if ( is_array( $meta ) ) {
8384                                                if ( array_key_exists( 'attachments', $meta ) ) {
8385
$attachmentsCount = count( $meta['attachments'] );
8386                                                }
8387                                                if ( array_key_exists( 'recordings', $meta ) ) {
8388                                                        $recordingsCount = count( $meta['recordings'] );
8389                                                }
8390                                        }
```

```
8391                                }
8392
8393                                if ( $recordingsCount ) {
8394                                        $htmlCode .= '<br>';
8395                                        foreach ( $meta['recordings'] as $attachment )
8396                                                if ( file_exists( $attachment['file'] ) )
8397                                                {
8398                                                $htmlCode .= '<br><a class="ui label small"
href="' . self::path2url( $attachment['file'] ) . '"> <i class="file video icon">
</i>' . $attachment['name'] . ' ' . self::humanSize( filesize( $attachment['file'] ) ) . ' </a> ';
8399
8400                                                }
8401                                                else $htmlCode .=  $attachment['name'] . ' missing ';
8402                                }
8403
8404                                if ( $attachmentsCount ) {
8405                                        $htmlCode .= '<br>';
8406                                        foreach ( $meta['attachments'] as $attachment )
8407                                                if ( file_exists( $attachment['file'] ) )
8408                                                {
8409                                                $url = self::path2url( $attachment['file'] );
8410
8411                                                $htmlCode .= '<br><a class="ui label small"
href="' . $url  . '"> <i class="file icon"></i>' . $attachment['name'] . '
' . self::humanSize( filesize( $attachment['file'] ) ) . ' </a> ';
8412
8413
$ext = strtolower( pathinfo( $attachment['file'], PATHINFO_EXTENSION ) );
8414
if ( in_array($ext, [ 'jpg', 'jpeg', 'gif' ]) ) $htmlCode .= '<br><a href="' . $url . '" class="ui medium
image"> <img src="' . $url . '"> </a>';
8415
8416                                                }
8417                                                else $htmlCode .=  '<span class="ui label
small">' . $attachment['name'] . ' missing ' . '</span>';
8418
8419                                }
8420
8421                                $htmlCode .= '</div></div></div>';
8422                        }
8423
8424                        $htmlCode .= '</div>';
8425
8426                } else {
8427                        $htmlCode .= __( 'No Messages', 'ppv-live-webcams' );
8428                }
8429
8430                $htmlCode .= '</div>';
8431
8432                return $htmlCode;
8433        }
8434
8435
8436        static function videowhisper_cam_messages_performer( $atts ) {
8437                // show messages/questions to performer for answering
8438
8439                // if (!is_user_logged_in()) return __('Login to manage messages and replies!','ppv-live-
webcams') . '<BR><a class="ui button primary qbutton" href="' . wp_login_url() . '">' . __('Login', 'ppv-
live-webcams') . '</a>  <a class="ui button secondary qbutton" href="' . wp_registration_url() . '">' .
__('Register', 'ppv-live-webcams') . '</a>';
8440
8441                if ( ! is_user_logged_in() ) {
8442                        return '<div class="ui ' . $options['interfaceClass'] . ' header"> <i class="sign
in icon"></i> ' . __( 'Login to manage messages and replies!', 'ppv-live-
```

```
Line
webcams' ) . '</div>' . do_shortcode( '[videowhisper_login_form register="1" role="' . $options['rolePerformer'] . '"]' );
8443                }
8444
8445                $atts = shortcode_atts(
8446                        array(
8447                                'post_id' => 0, // for setup
8448                        ),
8449                        $atts,
8450                        'videowhisper_cam_messages_performer'
8451                );
8452                $atts = array_map('esc_attr', $atts);
8453
8454                // room id
8455                $postID = intval( $atts['post_id'] );
8456                if ( ! $postID ) {
8457                        $postID = intval( $_GET['webcam_id'] ); // pass by URL
8458                }
8459
8460                if ( $postID ) {
8461                        $post = get_post( $postID );
8462                }
8463                if ( ! $post ) {
8464                        $postID = 0;
8465                }
8466
8467                if ( ! $postID ) {
8468                        return 'Error: Could not find this webcam room! #' . $postID;
8469                }
8470
8471                $room = sanitize_file_name( $post->post_title );
8472
8473                if ( ! self::isAuthor( $postID ) ) {
8474                        return 'Access to messages not permitted (different room owner)! #' . $postID;
8475                        die;
8476                }
8477
8478                $options = self::getOptions();
8479                $htmlCode = '';
8480
8481                $this_page = self::getCurrentURL();
8482
8483
if (isset( $_GET['messages'] ) ) if ( $_GET['messages'] == 'setup' && $_POST['question_setup'] ) {
8484
8485                        $price = floatval( $_POST['question_price'] );
8486                        if ( ! $price ) {
8487                                $price = $options['messagesCost'];
8488                        }
8489                        update_post_meta( $postID, 'question_price', $price );
8490
8491
update_post_meta( $postID, 'question_closed', intval( $_POST['question_closed'] ) );
8492                }
8493
8494                $messageCode = '';
8495
if (isset( $_GET['messages'] ) ) if ( in_array( $_GET['messages'], array( 'view', 'reply' ) ) ) {
8496                        $messageCode = sanitize_text_field( $_GET['msg'] );
8497
8498                        // performer message reply
8499                        if ( $_POST['newmessage'] ) {
```

```
8500
8501                                global $wpdb;
8502                                $table_messages = $wpdb->prefix . 'vw_vmls_messages';
8503
8504                                $ztime = time();
8505
8506                                $sender_id = get_current_user_id();
8507
8508                                $id = self::to10( $messageCode );
8509                                if ( ! $id ) {
8510                                        return 'Error: Incorrect message code! #' . $messageCode;
8511                                }
8512
8513                                $sqlS        = $wpdb->prepare("SELECT * FROM $table_messages WHERE
    id=%d", $id);
8514                                $mainmessage = $wpdb->get_row( $sqlS );
8515                                if ( ! $mainmessage ) {
8516                                        return 'Error: Message not found! #' . $id;
8517                                }
8518
8519                                $ldate = $wpdb->get_var( $wpdb->prepare( "SELECT MAX(ldate) FROM
    `$table_messages` WHERE id=%d", $id ) );if ( $ldate == $ztime ) {
8520                                        return 'Error: You just sent reply or shortcode was already
    executed.';
8521                                }
8522
8523                                // allowed tags
8524                                $allowedtags = array(
8525                                        'a'          => array(
8526                                                'href'  => true,
8527                                                'title' => true,
8528                                        ),
8529                                        'abbr'       => array(
8530                                                'title' => true,
8531                                        ),
8532                                        'acronym'    => array(
8533                                                'title' => true,
8534                                        ),
8535
8536                                        'b'          => array(),
8537                                        'blockquote' => array(
8538                                                'cite' => true,
8539                                        ),
8540                                        'cite'       => array(),
8541                                        'code'       => array(),
8542                                        'del'        => array(
8543                                                'datetime' => true,
8544                                        ),
8545                                        'em'         => array(),
8546                                        'i'          => array(),
8547                                        'q'          => array(
8548                                                'cite' => true,
8549                                        ),
8550                                        'strike'     => array(),
8551                                        'strong'     => array(),
8552
8553                                        'ul'         => array(),
8554                                        'ol'         => array(),
8555                                        'li'         => array(),
8556
8557                                        'span'       => array(
8558                                                'style' => array(),
```

```
8559                                    ),
8560
8561                                    'p'          => array(
8562                                            'style' => array(),
8563                                    ),
8564                            );
8565
8566                            $message = wp_kses( stripslashes($_POST['msgContent']) , $allowedtags );
8567
8568                                                    $meta = array();
8569                            $meta['attachments']  = array();
8570
8571                            // Attachments
8572                            $countfiles = 0;
8573                            if ( ! empty( $_FILES ) ) {
8574                                    if ( $_FILES['attachments'] ) {
8575                                            $countfiles = count( $_FILES['attachments'] );
8576                                    }
8577                            }
8578
8579                            if ( $options['attachment_extensions'] ) {
8580                                    if ( $countfiles ) {
8581
8582                                            $dir = sanitize_text_field( $options['uploadsPath'] );
8583                                            if ( ! file_exists( $dir ) ) {
8584                                                    mkdir( $dir );
8585                                            }
8586
8587                                            $dir .= "/$room";
8588                                            if ( ! file_exists( $dir ) ) {
8589                                                    mkdir( $dir );
8590                                            }
8591
8592                                            $dir .= '/_attachments';
8593                                            if ( ! file_exists( $dir ) ) {
8594                                                    mkdir( $dir );
8595                                            }
8596
8597                                            $dir .= '/';
8598
8599                                            // Looping all files
8600                                            for ( $i = 0;$i < $countfiles;$i++ ) {
8601
8602                                                    $filename = isset($_FILES['attachments']['name'][ $i ]) ? $_FILES['attachments']['name'][ $i ] : '' ;
8603
8604                                                    if ( ! $filename ) {
8605                                                            break;
8606                                                    }
8607
8608                            $ext = strtolower( pathinfo( $filename, PATHINFO_EXTENSION ) );
8609
8610                            if ( ! in_array( $ext, self::extensions_attachment() ) ) {
8611                                                            return '<div class="ui message error">' . $filename . ') ' . __( 'This attachment extension is not allowed:', 'ppv-live-webcams' ) . ' ' . $ext . '<br>' . __( 'Allowed attachment extensions:', 'ppv-live-webcams' ) . ' ' . $options['attachment_extensions'] . '<br>' . __( 'Try again:', 'ppv-live-webcams' ) . ' ' . $message . '</div>';
8612
8613                                                            exit;
8614                                                    }
```

```
8615
8616                                                    if ( ! file_exists( $_FILES['attachments']['tmp_name'][ $i ] ) ) {
8617                                                            return 'Attachment upload failed. Report to webmaster and include link in message, until sorted out.';
8618                                                    }
8619
8620                                                    if ( $options['attachmentSize'] ) {
8621                                                            if ( filesize( $_FILES['attachments']['tmp_name'][ $i ] ) > $options['attachmentSize'] ) {
8622                                                    return '<div class="ui message error">' . __( 'This attachment is too big:', 'ppv-live-webcams' ) . ' ' . $filename . '<br>' . __( 'Allowed attachment size:', 'ppv-live-webcams' ) . '
' . $options['attachmentSize'] . '<br>' . __( 'Try again:', 'ppv-live-webcams' ) . '
' . $message . '</div>';
8623                                                            }
8624                                                    }
8625
8626                                                    // Upload file
8627                                                    $path = $dir . self::generateName( $filename );
8628
8629                                                    $meta['attachments'][ $i ] = array(
8630                                                    'name' => sanitize_text_field( $filename ),
8631                                                            'file' => $path,
8632                                                    'size' => filesize( $_FILES['attachments']['tmp_name'][ $i ] ),
8633                                                    );
8634
8635                                                    //per file
8636                                                    $file = array(
8637                                                                    'name' => $_FILES['attachments']['name'][ $i ],
8638                                                                    'type' => $_FILES['attachments']['type'][ $i ],
8639                                                                    'tmp_name' => $_FILES['attachments']['tmp_name'][ $i ],
8640                                                                    'error' => $_FILES['attachments']['error'][ $i ] ,
8641                                                                    'size' => $_FILES['attachments']['size'][ $i ]
8642                                                            );
8643
8644                                                    // ( $_FILES['attachments']['tmp_name'][$i], $path);
8645                                                    $errorUp = self::handle_upload( $file, $path ); // handle trough wp_handle_upload()
8646
8647                                                    if ( $errorUp ) {
8648                                                            $htmlCode .= '<br>' . 'Error uploading ' . esc_html( $filename . ':' . $errorUp  )  ;
8649                                                    }
8650
8651                                                    if (!file_exists($path)) $htmlCode .= '<br> Upload failed, file missing: ' . $path;
8652
8653                                                            }
8654                                                    }
8655                                            }
8656
8657                                            // message sender
8658                                            $sender = wp_get_current_user();
8659
8660                                            // recordings
8661                                            $meta['recordings'] = array();
8662                                            $recordings         = sanitize_text_field( $_POST['recordings'] );
```

```
8663                                if ( $recordings ) {
8664
8665                                        $optionsRecorder = get_option( 'VWinlineRecorderOptions' );
8666                                        $recordingFiles  = explode( ',', $recordings );
8667
8668                                        foreach ( $recordingFiles as $recordingFile ) {
8669                                                if ( $recordingFile ) {
8670                                                        if ( file_exists( $filePath = $optionsRecorder['uploadsPath'] . '/' . $sender->user_login . '/' . $recordingFile ) ) {
8671                                                                $meta['recordings'][] = array(
8672                                                                        'name' => sanitize_text_field( $recordingFile ),
8673                                                                        'file' => $filePath,
8674                                                                        'size' => filesize( $filePath ),
8675                                                                );
8676                                                        } else {
8677                                                                $htmlCode .= ' Recording file not found: ' . $filePath;
8678                                                        }
8679                                                }
8680                                        }
8681                                }
8682
8683                                $metaS = serialize( $meta );
8684
8685                                $sqlIM = $wpdb->prepare(
8686                                        "INSERT INTO `$table_messages` ( `sender_id`, `webcam_id`, `reply_id`, `sdate`, `message`, `meta`) VALUES ( %d, %d, %d, %d, %s, %s)",
8687                                        $sender_id, $postID, $id, $ztime, $message, $metaS
8688                                );
8689                                $wpdb->query( $sqlIM );
8690                                $messageID = $wpdb->insert_id;
8691                                if ( ! $messageID ) {
8692                                        return 'Error: Sending message failed! ' . $sqlIM;
8693                                }
8694
8695                                        // notification to client
8696                                        $client = get_userdata( $mainmessage->sender_id );
8697                                        $link = get_permalink( $options['p_videowhisper_webcams_client'] );
8698                                if ( $client ) {
8699                                        $mailed = wp_mail( $client->user_email, $options['messagesSubject'] . ' ' . ( $sender->display_name ? $sender->display_name : $sender->user_login ), $options['messagesText'] . "\r\n" . $link );
8700
8701                                        //notify by SMS
8702                                        if ($options['sms_number'])
8703                                        {
8704                                                $sms_number = get_user_meta( $client->ID, 'sms_number', true );
8705                                                if ($sms_number)
8706                                                {
8707                                                        $sms_last =  get_user_meta( $client->ID, 'sms_last', true );
8708                                                        if (!$sms_last || $sms_last < ( time() - $options['sms_cooldown']) )
8709                                                        {
8710
8711                                                                $sms_message = $options['messagesSubject'] . ' ' . ( $sender->display_name ? $sender->display_name : $sender->user_login ) . ' ' . $link;
8712
8713                                                                if ($options['wp_sms']) if ( function_exists('wp_sms_send') ) wp_sms_send($sms_number,  $sms_message);
8714
```

```
8715  if ($options['wp_sms_twilio']) if ( function_exists('twl_send_sms') )
8716                                                  {
8717  $response = twl_send_sms( array( 'number_to' => $sms_number, 'message' => $sms_message ) );
8718  if( is_wp_error( $response ) ) $htmlCode .= '<div class="ui message">SMS Error: ' . esc_html( $response->get_error_message() ) . '</div>';
8719                                                  }
8720
8721                                                  update_user_meta( $author->ID, 'sms_last', time() );
8722                                                  }
8723                                          }
8724
8725                                  }
8726
8727                          } else {
8728                                  $htmlCode .= 'Error: User not found to notify!';
8729                          }
8730
8731                          if ( ! $mailed ) {
8732                                  $htmlCode .= 'Warning: Notification email was not sent by WP.';
8733                          }
8734
8735                          // update last message time
8736                          $sqlU = $wpdb->prepare("UPDATE `$table_messages` SET `ldate` = %d WHERE id = %d", $ztime, $id);
8737                          $wpdb->query( $sqlU );
8738
8739                          // get paid for first reply
8740                          $sqlC = $wpdb->prepare("SELECT COUNT(id) FROM `$table_messages` WHERE reply_id = %d", $id);
8741                          $repliesCount = $wpdb->get_var( $sqlC );
8742                          if ( ! $repliesCount ) {
8743                                  return 'Error: No replies! ' . $sqlC;
8744                          }
8745
8746                          // get paid
8747  $messagesCost = floatval( get_post_meta( $postID, 'question_price', true ) );
8748                          if ( ! $messagesCost ) {
8749                                  $messagesCost = floatval( $options['messagesCost'] );
8750                          }
8751
8752                          $performerRatio = self::performerRatio( $post->post_name, $options, $postID );
8753
8754                          // only get paid for first reply and  if cost enabled
8755                          if ( $repliesCount == 1 && $messagesCost ) {
8756                                  $messageURL = add_query_arg(
8757                                          array(
8758                                                  'messages' => 'view',
8759                                                  'msg'  => $messageCode,
8760                                          ),
8761                                          $this_page
8762                                  );
8763
8764  self::transaction( $ref = 'paid_message_earn', $sender_id, $messagesCost * $performerRatio, __( 'Paid Message Earning from ', 'ppv-live-webcams' ) . ' <a href="' . $messageURL . '">' . $messageCode . '</a>', $messageID, $message, $options );
8765                                  $htmlCode .= '<div id="successMessage" class="ui ' . $options['interfaceClass'] . ' segment">' . __( 'Your reply was sent and you got paid.', 'ppv-live-webcams' ) . '</div>';
```

```
8766
8767                                } else {
8768                                        $htmlCode .= '<div id="successMessage" class="ui ' . $options['interfaceClass'] . ' segment">' . __( 'Your reply was sent.', 'ppv-live-webcams' ) . '</div>';
8769                                }
8770                        }
8771
8772                        // show msg & replies
8773                        $htmlCode .= self::messageReplies( $messageCode );
8774
8775                        // reply form
8776                        $tinymce_options = array(
8777                                'plugins'          => 'lists,link,textcolor,hr',
8778                                'toolbar1' => 'cut,copy,paste,|,undo,redo,|,fontsizeselect,forecolor,backcolor,bold,italic,underline,strikethrough',
8779                                'toolbar2' => 'alignleft,aligncenter,alignright,alignjustify,blockquote,hr,bullist,numlist,link,unlink',
8780                                'fontsize_formats' => '8pt 9pt 10pt 11pt 12pt 14pt 16pt 18pt',
8781                        );
8782
8783                        $action    = add_query_arg(
8784                                array(
8785                                        'messages' => 'reply',
8786                                        'msg'  => $messageCode,
8787                                ),
8788                                $this_page
8789                        );
8790                        $htmlCode .= '<form method="post" enctype="multipart/form-data" action="' . $action . '" name="messageForm" class="ui form segment">';
8791
8792                        $htmlCode .= '<div class="field"><label><i class="newspaper icon"></i> ' . __( 'Send Reply', 'ppv-live-webcams' ) . '</label>';
8793                        ob_start();
8794                        wp_editor(
8795                                '',
8796                                'msgContent',
8797                                $settings = array(
8798                                        'textarea_rows' => 3,
8799                                        'media_buttons' => false,
8800                                        'teeny'         => true,
8801                                        'wpautop'       => false,
8802                                        'tinymce'       => $tinymce_options,
8803                                )
8804                        );
8805                        $htmlCode    .= ob_get_clean();
8806                        $htmlCode    .= '</div>';
8807
8808                        $htmlCode .= ( $options['attachment_extensions'] ? '
8809        <div class="field"><label><i class="attach icon"></i> ' . __( 'Attachments (Optional)', 'ppv-live-webcams' ) . '</label>
8810        <input type="file" name="attachments[]" id="attachments" multiple placeholder="Attachments"><br><small>' . __( 'Supports', 'ppv-live-webcams' ) . ' : ' . $options['attachment_extensions'] . '</small>
8811        </div>
8812        ' : '' );
8813
8814                        if ( shortcode_exists( 'videowhisper_recorder_inline' ) ) {
8815                                $htmlCode .= do_shortcode( '[videowhisper_recorder_inline label="Attach Recordings (Optional)"]' );
8816                        }
8817
8818                        $htmlCode .= '<div class="field"> <input class="ui button videowhisperButton" type="submit" name="newmessage" id="newmessage" value="' . __( 'Send Reply', 'ppv-live-webcams' ) . '" />
```

```
</div>
        <input name="post_id" id="post_id" type="hidden" value="' . $postID . '">
        </form>';

                }

                // list messages

                global $wpdb;
                $table_messages = $wpdb->prefix . 'vw_vmls_messages';

                $htmlCode .= '<div class="ui segment ' . $options['interfaceClass'] . '">';

                $sql = $wpdb->prepare("SELECT * FROM $table_messages WHERE webcam_id = %d AND reply_id = 0 ORDER BY sdate DESC, id DESC LIMIT 0, 100", $postID);
                $results = $wpdb->get_results( $sql );

                if ( $wpdb->num_rows > 0 ) {
                        $htmlCode .= '<div class="ui list celled ' . $options['interfaceClass'] . '">';

                        foreach ( $results as $message ) {
                                $messageCode1 = self::to62( $message->id );

                                $messageURL = add_query_arg(
                                        array(
                                                'messages' => 'view',
                                                'msg'  => $messageCode1,
                                        ),
                                        $this_page
                                );
                                $sender     = get_userdata( $message->sender_id );

                                // replies
                                $sqlC         = $wpdb->prepare("SELECT COUNT(id) FROM `$table_messages` WHERE reply_id=%d", $message->id);
                                $repliesCount = $wpdb->get_var( $sqlC );

                                $attachmentsCount = 0;
                                if ( $message->meta ) {
                                        $meta = unserialize( $message->meta, array( 'allowed_classes' => false ) );
                                        if ( is_array( $meta ) ) {
                                                if ( array_key_exists( 'attachments', $meta ) ) {
                                                        $attachmentsCount = count( $meta['attachments'] );
                                                }
                                        }
                                }

                                $htmlCode .= '<div class="item ' . $options['interfaceClass'] . '" href="' . $messageURL . '"> <i class="' . ( $messageCode == $messageCode1 ? 'envelope outline' : 'mail' ) . ' icon big"></i>';

                                $htmlCode .= '<div class="content">';
                                $htmlCode .= '<div class="header">' . ' <i class="user icon"></i>' . ( $sender ? ($sender->display_name ? $sender->display_name : $sender->user_login) : '#' . $message->sender_id . ' - User Not Found'). ' <i class="clock icon"></i>' . date( DATE_RFC2822, $message->sdate ) . ' <i class="arrow right icon"></i> #' . $messageCode1 . ( $repliesCount ? ' <i class="check icon"></i> ' . __( 'Replied', 'ppv-live-webcams' ) . '' : ' <i class="spinner loading icon"></i> <i class="green bell icon"></i>' ) . '</div>';

                                $htmlCode .= '<div class="description">' . ( $attachmentsCount ? '<i class="ui attach icon"></i>' . $attachmentsCount . ' ' : '' ) . ' <i class="ui newspaper icon"></i> ' . wp_trim_excerpt( $message->message ) . '</div>';
```

```
8869                                $htmlCode .= '<div class="description"><a href="' . $messageURL . '" class="ui button compact"> <i class="zoom-in icon"></i> ' . __( 'View Message', 'ppv-live-webcams' ) . '</a></div>';
8870
8871                                $htmlCode .= '</div></div>';
8872                        }
8873
8874                        $htmlCode .= __( 'You receive payment for paid messages on reply. Clients can access their messages and replies from client dashboard and in messages section from webcam profile.', 'ppv-live-webcams' );
8875
8876                        $htmlCode .= '</div>';
8877
8878                } else {
8879                        $htmlCode .= __( 'No Messages', 'ppv-live-webcams' );
8880                }
8881                $htmlCode .= '</div>';
8882
8883                // messaging settings
8884
8885                $price = floatval( get_post_meta( $postID, 'question_price', true ) );
8886                if ( ! $price ) {
8887                        $price = $options['messagesCost'];
8888                }
8889
8890                $closed = intval( get_post_meta( $postID, 'question_closed', true ) );
8891
8892                $htmlCode .= '<form class="ui form ' . $options['interfaceClass'] . '" method="post" enctype="multipart/form-data" action="' . add_query_arg( 'messages', 'setup', $this_page ) . '" name="questionSetup">';
8893
8894                $htmlCode .= '<div class="two fields">';
8895
8896                $htmlCode .= '<div class="field"> <label>' . __( 'Questions', 'ppv-live-webcams' ) . ' </label> <select class="ui fluid dropdown" name="question_closed" id="question_closed">
8897                <option value="1" ' . ( $closed ? 'selected' : '' ) . '>' . __( 'Closed', 'ppv-live-webcams' ) . '</option>
8898                <option value="0" ' . ( ! $closed ? 'selected' : '' ) . '>' . __( 'Open', 'ppv-live-webcams' ) . '</option>
8899                </select>
8900                </div>';
8901
8902                $htmlCode .= '<div class="field"> <label>' . __( 'Price per Question', 'ppv-live-webcams' ) . ' </label>
8903                <div class="ui right labeled input">
8904                <input size="8" maxlength="12" type="text" name="question_price" id="question_price" value="' . $price . '"/>
8905                <div class="ui basic label">
8906                ' . $options['currency'] . '
8907                </div>
8908                </div>
8909                </div>';
8910
8911                $htmlCode .= '<div class="field"> <label>' . __( 'Update', 'ppv-live-webcams' ) . '</label> <input class="ui button videowhisperButton" type="submit" name="question_setup" id="question_setup" value="' . __( 'Save', 'ppv-live-webcams' ) . '" /></div>';
8912
8913                $htmlCode .= '</div>';
8914
8915                $htmlCode .= '<input name="post_id" id="post_id" type="hidden" value="' . $postID . '">
8916                </form>';
8917
8918                return $htmlCode;
8919        }
8920
```

```
8921
8922        static function videowhisper_webcams_client( $atts ) {
8923
8924                self::enqueueUI();
8925                $htmlCode = '';
8926
8927                $options = get_option( 'VWliveWebcamsOptions' );
8928
8929                if ( ! is_user_logged_in() ) {
8930                        return '<div class="ui ' . $options['interfaceClass'] . ' header"> <i class="sign in icon"></i> ' . __( 'Login to access client features!', 'ppv-live-webcams' ) . '</div>' . do_shortcode( '[videowhisper_login_form register="1" role="' . $options['roleClient'] . '"]' );
8931                }
8932
8933                // if (!is_user_logged_in()) return __('Login to access as client!','ppv-live-webcams') . '<BR><a class="ui button primary qbutton" href="' . wp_login_url() . '">' . __('Login', 'ppv-live-webcams') . '</a>  <a class="ui button secondary qbutton" href="' . ( ($options['registerFrontend'] && $options['p_videowhisper_register_client']) ? get_permalink($options['p_videowhisper_register_client']) : wp_registration_url() ) . '">' . __('Register', 'ppv-live-webcams') . '</a>';
8934
8935                $htmlCode = '';
8936
8937                $user_ID = get_current_user_id();
8938
8939                $current_user = wp_get_current_user();
8940                $this_page    = self::getCurrentURL();
8941
8942                $htmlCode .= '<div class="ui ' . $options['interfaceClass'] . ' header">' . $current_user->user_login . ( $options['loginFrontend'] ? ' <a class="ui tiny button" href="' . wp_nonce_url( add_query_arg( 'logout', $current_user->ID, get_permalink( $options['p_videowhisper_login'] ) ), 'logout' . $current_user->ID, 'verify' ) . '"><i class="sign out icon"></i>' . __( 'Log Out', 'ppv-live-webcams' ) . '</a>' : '' ) . '</div>';
8943
8944                if ( $options['clientWallet'] ?? false) {
8945                        $htmlCode .= '<div class="ui ' . $options['interfaceClass'] . ' green segment form">';
8946                        $htmlCode .= '<h4> <i class="ui icon money"></i>' . __( 'Active balance', 'ppv-live-webcams' ) . ': ' . self::balance( $user_ID ) . '</h4>';
8947                        $htmlCode .= __( 'All Balances', 'ppv-live-webcams' ) . ': ' . self::balances( $user_ID );
8948                        $htmlCode .= '<br><A class="ui button primary" HREF="' . get_permalink( $options['balancePage'] ) . '"><i class="ui icon wallet"></i>' . __( 'Manage Wallet', 'ppv-live-webcams' ) . '</A>';
8949                        $htmlCode .= '</div>';
8950                }
8951
8952                if ($options['match'] ?? false)
8953                if ( ! self::any_in_array( self::getRolesPerformer(), $current_user->roles ) ) //performers need to go live from own dashboard
8954                {
8955                        $htmlCode .= '<div class="ui ' . $options['interfaceClass'] . ' segment">';
8956                        $htmlCode .= '<h4>' . __( 'Random Mathmaking', 'ppv-live-webcams' ) . '</h4>';
8957                        $htmlCode .= '<A class="ui button" href="' . get_permalink( $options['p_videowhisper_match_form'] ) . '">' . __( 'Match Criteria', 'ppv-live-webcams' ) . '</A>';
8958                        $htmlCode .= '<A class="ui button" href="' . get_permalink( $options['p_videowhisper_match'] ) . '">' . __( 'Random Match', 'ppv-live-webcams' ) . '</A>';
8959                        $htmlCode .= '</div>';
8960                }
8961
8962                if ( $options['messages'] ?? false ) {
8963                        $htmlCode .= '<h3 class="ui ' . $options['interfaceClass'] . ' header">' . __( 'My Questions and Messages', 'ppv-live-webcams' ) . '</h3>';
8964                        $htmlCode .= do_shortcode( '[videowhisper_cam_messages]' );
8965                }
```

```
8966
8967                if ( $options['calls'] ?? false ) {
8968                        $htmlCode .= '<h3 class="ui ' . $options['interfaceClass'] . '
header">' . __( 'My Calls', 'ppv-live-webcams' ) . '</h3>';
8969                        $htmlCode .= do_shortcode( '[videowhisper_cam_calls]' );
8970                }
8971
8972                if ( $options['addRoles'] ?? false ) {
8973                        $htmlCode .= '<h3 class="ui ' . $options['interfaceClass'] . '
header">' . __( 'Add Role', 'ppv-live-webcams' ) . '</h3>';
8974                        $htmlCode .= do_shortcode( '[videowhisper_role_change]' );
8975                }
8976
8977                if ( $options['sms_number'] ?? false )
8978                {
8979                        //$htmlCode .= '<h3 class="ui ' . $options['interfaceClass'] . ' header">' . __(
'SMS Mobile Number', 'ppv-live-webcams' ) . '</h3>';
8980
8981                        $htmlCode .= do_shortcode( '[videowhisper_sms_number]' );
8982                }
8983
8984
if ( $options['clientSubscriptions'] ?? false ) if (shortcode_exists( 'videowhisper_client_subscriptions'
 )) {
8985                        $htmlCode .= '<h3 class="ui ' . $options['interfaceClass'] . '
header">' . __( 'My Subscriptions', 'ppv-live-webcams' ) . '</h3>';
8986                        $htmlCode .= do_shortcode( '[videowhisper_client_subscriptions]' );
8987                }
8988
8989
if ( $options['clientContent'] ?? false ) if (shortcode_exists( 'videowhisper_content' )) {
8990                        $htmlCode .= do_shortcode( '[videowhisper_content]' );
8991                }
8992
8993                if ( ! $htmlCode ) {
8994                        return 'No client dashboard features enabled.';
8995                }
8996
8997                return $htmlCode;
8998        }
8999
9000
9001        static function videowhisper_role_change( $atts ) {
9002
9003                self::enqueueUI();
9004                $htmlCode = '';
9005
9006                if ( ! is_user_logged_in() ) {
9007                        return '<div class="ui ' . $options['interfaceClass'] . ' header"> <i class="sign
in icon"></i> ' . __( 'Login to manage role!', 'ppv-live-
webcams' ) . '</div>' . do_shortcode( '[videowhisper_login_form register="1"]' );
9008                }
9009
9010                $options = get_option( 'VWliveWebcamsOptions' );
9011                if ( ! $options['registrationFormRole'] ) {
9012                        return 'Registration roles disabled!';
9013                }
9014
9015                $roles = array( $options['roleClient'], $options['rolePerformer'] );
9016                if ( $options['studios'] ) {
9017                        $roles[] = $options['roleStudio']; // add studio if enabled
9018                }
9019
9020                $changerole = sanitize_text_field( $_POST['changerole'] ?? '' );
```

```
9021
9022                $current_user = wp_get_current_user();
9023
9024                if ( $changerole ) {
9025                        if ( ! in_array( $changerole, $roles ) ) {
9026                                return 'Unavailable role!';
9027                        }
9028                        if ( in_array( 'administrator', $current_user->roles ) ) {
9029                                return 'Administrator can access all features without other roles!';
9030                        }
9031                        $current_user->add_role( $changerole );
9032                }
9033
9034                $htmlCode = '';
9035
9036                $roleCode = '';
9037                foreach ( $current_user->roles as $role ) {
9038                        $roleCode .= ( $roleCode ? ', ' : '' ) . $role;
9039                }
9040
9041                $htmlCode .= __( 'Your role(s)', 'ppv-live-webcams' ) . ' : ' . $roleCode;
9042
9043                $htmlCode .= '<form method="post" enctype="multipart/form-data"
action="' . self::getCurrentURL() . '" name="callForm" class="ui form ' . $options['interfaceClass'] . '
segment">';
9044
9045                $htmlCode .= '<label for="role"> ' . __( 'Role', 'ppv-live-webcams' ) . '</label> <select
id="changerole" name="changerole" class="ui dropdown v-select">';
9046
9047                foreach ( $roles as $role ) {
9048                        // create role if missing
9049                        if ( ! $oRole = get_role( $role ) ) {
9050                                add_role( $role, ucwords( $role ), array( 'read' => true ) );
9051                                $oRole = get_role( $role );
9052                        }
9053
9054                        $htmlCode .= '<option value="' . $role . '" ' . ( in_array( $role, $current_user-
>roles ) ? 'selected' : '' ) . '>' . ucwords( $oRole->name ) . '</option>';
9055
9056                }
9057
9058                $htmlCode .= '</select>';
9059
9060                $htmlCode .= '<input class="ui button" type="submit" name="update" id="newcall"
value="' . __( 'Add Role', 'ppv-live-webcams' ) . '" /></form>';
9061
9062                return $htmlCode;
9063
9064        }
9065
9066
9067        static function videowhisper_cam_messages( $atts ) {
9068
9069                // shows client messages, to check replies
9070
9071                // if (!is_user_logged_in()) return __('Login to manage messages and replies!','ppv-live-
webcams') . '<BR><a class="ui button primary qbutton" href="' . wp_login_url() . '">' . __('Login', 'ppv-
live-webcams') . '</a>  <a class="ui button secondary qbutton" href="' . wp_registration_url() . '">' .
__('Register', 'ppv-live-webcams') . '</a>';
9072
9073                $options = self::getOptions();
9074                if ( ! is_user_logged_in() ) {
9075                        return '<div class="ui ' . $options['interfaceClass'] . ' header"> <i class="sign
in icon"></i> ' . __( 'Login to manage messages and replies!', 'ppv-live-
```

```
webcams' ) . '</div>' . do_shortcode( '[videowhisper_login_form register="1" role="' . $options['roleClient'] . '"]' );
                }

                $atts = shortcode_atts(
                        array(
                                'post_id' => 0, // 0 shows all
                        ),
                        $atts,
                        'videowhisper_cam_messages_performer'
                );
                $atts = array_map('esc_attr', $atts);

                // room id
                $postID = intval( $atts['post_id'] );
                if ( ! $postID ) {
                        $postID = intval( $_GET['webcam_id'] ?? 0 ); // pass by URL
                }

                $post = false;
                if ( $postID ) {
                        $post = get_post( $postID );
                }

                if ( ! $post ) {
                        $postID = 0;
                }

                // if (!$postID) return 'Error: Could not find this webcam room! #' .  $postID;

                $htmlCode = '';

                $this_page = self::getCurrentURL();

                $sender_id = get_current_user_id();

                $messageCode = '';
                if ( isset( $_GET['msg'] ) && $_GET['view'] == 'messages' ) {
                        $messageCode = sanitize_text_field( $_GET['msg'] );

                        // show msg & replies
                        $htmlCode .= self::messageReplies( $messageCode );
                }

                global $wpdb;
                $table_messages = $wpdb->prefix . 'vw_vmls_messages';

                // list messages
                $htmlCode .= '<div class="ui segment ' . $options['interfaceClass'] . '">';

                $cnd = '';
                if ( $postID ) {
                        $cnd = "AND webcam_id='$postID'";
                }
                $sql = $wpdb->prepare("SELECT * FROM $table_messages WHERE sender_id = %d $cnd AND reply_id = 0 ORDER BY sdate DESC, id DESC LIMIT 0, 100", $sender_id);
                $results = $wpdb->get_results($sql);

                if ( $wpdb->num_rows > 0 ) {
                        $htmlCode .= '<div class="ui list celled ' . $options['interfaceClass'] . '">';

                        foreach ( $results as $message ) {
```

```
9135
9136                                    // replies
9137                                    $sqlC         = $wpdb->prepare("SELECT COUNT(id) FROM `$table_messages`
WHERE reply_id=%d", $message->id);
9138                                    $repliesCount = $wpdb->get_var( $sqlC );
9139
9140                                    $messageCode1 = self::to62( $message->id );
9141
9142                                    $messageURL = add_query_arg(
9143                                            array(
9144                                                    'view'    => 'messages',
9145                                                    'msg' => $messageCode1,
9146                                            ),
9147                                            $this_page
9148                                    );
9149                                    $sender     = get_userdata( $message->sender_id );
9150
9151                                    $htmlCode .= '<div class="item ' . $options['interfaceClass'] . '"> <i
class="' . ( $messageCode == $messageCode1 ? 'envelope outline' : 'mail' ) . ' icon big"></i>';
9152                                    $htmlCode .= '<div class="content">';
9153                                    $htmlCode .= '<div class="header">' . ' <i class="user icon">
</i>' . $sender->user_nicename . ' <i class="clock icon"></i>' . date( DATE_RFC2822, $message-
>sdate ) . ( $repliesCount ? ' <i class="check icon"></i> ' . __( 'Replied', 'ppv-live-
webcams' ) . '' : ' <i class="spinner loading icon"></i>' ) . '</div>';
9154
9155                                    // var_dump($message);
9156
9157                                    $attachmentsCount = 0;
9158                                    if ( $message->meta ) {
9159                                            $meta = unserialize( $message-
>meta, array( 'allowed_classes' => false ) );
9160                                            if ( is_array( $meta ) ) {
9161                                                    if ( array_key_exists( 'attachments', $meta ) ) {
9162
$attachmentsCount = count( $meta['attachments'] );
9163                                                    }
9164                                            }
9165                                    }
9166
9167                                    $htmlCode .= '<div class="description">' . ( $attachmentsCount ? '<i
class="ui attach icon"></i>' . $attachmentsCount . ' ' : '' ) . ' <i class="ui newspaper icon"></i>
' . wp_trim_excerpt( $message->message ) . '</div>';
9168
9169                                    $htmlCode .= '<div class="description"> <a href="' . $messageURL . '"
class="ui button compact"> <i class="zoom-in icon"></i> ' . __( 'View Message', 'ppv-live-
webcams' ) . '</a></div>';
9170
9171                                    $htmlCode .= '</div></div>';
9172                            }
9173
9174                            $htmlCode .= '</div>';
9175
9176                    } else {
9177                            $htmlCode .= __( 'No Messages', 'ppv-live-webcams' );
9178                    }
9179                    $htmlCode .= '</div>';
9180
9181                    return $htmlCode;
9182            }
9183
9184
9185
9186            static function videowhisper_cam_calls( $atts ) {
9187                    // setup and list video calls
```

```
9188
9189                if ( ! is_user_logged_in() ) {
9190                        return __( 'Login to manage calls!', 'ppv-live-webcams' ) . '<BR><a class="ui button primary qbutton" href="' . wp_login_url() . '">' . __( 'Login', 'ppv-live-webcams' ) . '</a>  <a class="ui button secondary qbutton" href="' . wp_registration_url() . '">' . __( 'Register', 'ppv-live-webcams' ) . '</a>';
9191                }
9192
9193                $atts = shortcode_atts(
9194                        array(
9195                                'post_id' => 0, // for setup
9196                        ),
9197                        $atts,
9198                        'videowhisper_cam_calls'
9199                );
9200                $atts = array_map('esc_attr', $atts);
9201
9202                // room id
9203                $post = false;
9204                $postID = intval( $atts['post_id'] );
9205                if ( $postID ) {
9206                        $post = get_post( $postID );
9207                }
9208                if ( ! $post ) {
9209                        $postID = 0;
9210                }
9211
9212                $options = get_option( 'VWliveWebcamsOptions' );
9213
9214                $htmlCode = '';
9215
9216                // user info
9217                // performer (role)
9218                $current_user = wp_get_current_user();
9219                // access keys
9220                $userkeys   = $current_user->roles;
9221                $userkeys[] = $current_user->user_login;
9222                $userkeys[] = $current_user->user_nicename;
9223                $userkeys[] = $current_user->ID;
9224                $userkeys[] = $current_user->user_email;
9225
9226                global $wpdb;
9227                $table_private = $wpdb->prefix . 'vw_vmls_private';
9228
9229                $roleS = implode( ',', $current_user->roles );
9230                if ( $postID ) { // only for a room
9231                        if ( self::any_in_array( self::getRolesPerformer( $options ), $current_user->roles ) ) {
9232                                // main webcam room/post
9233
9234                                /*
9235                                //any owned webcam?
9236                                if (!$postID)
9237                                {
9238                                $postID = $wpdb->get_var( $sql = 'SELECT ID FROM ' . $wpdb->posts . ' WHERE post_author = \'' . $current_user->ID . '\' and post_type=\''.$options['custom_post'] . '\' LIMIT 0,1' );
9239                                }
9240                                //create a room
9241                                if (!$postID) $postID = self::webcamPost(); //default cam
9242                                if ($postID && !$post) $post = get_post($postID);
9243                                if (!$post) $postID = 0;
```

```
9244                                    if (!$postID) return 'Error: Could not find or setup a webcam room for this user!';
9245                                    */
9246
9247    $room_audio = self::is_true( get_post_meta( $postID, 'room_audio', true ) );
9248    $room_text = self::is_true( get_post_meta( $postID, 'room_text', true ) );
9249
9250                                    $showNew = 1;
9251
9252                                    if ( isset( $_GET['calls'] ) && $_GET['calls'] == 'new' )
9253                                    {
9254                                            $htmlCode .= '<div class="ui segment">';
9255
9256                                            $client = trim( sanitize_text_field( $_POST['client'] ) );
9257                                            if ( $client ) {
9258                                                    if ( filter_var( $client, FILTER_VALIDATE_EMAIL ) ) {
9259                                                            $user  = get_user_by( 'email', $client );
9260                                                            $login = $user->user_login;
9261                                                            $email = $client;
9262                                                    } else {
9263                                                            $user  = get_user_by( 'login', $client );
9264                                                            $login = $client;
9265                                                            $email = $user->user_email;
9266                                                    }
9267                                            }
9268
9269                                            if ( $user ) {
9270                                                    $callMode = sanitize_text_field( $_POST['callMode'] );
9271
9272                                                    $callType = '';
9273                                                    if ( $callMode == 'paid-audio' || $callMode == 'free-audio' ) {
9274                                                            $callType = 'audio';
9275                                                    }
9276                                                    if ( $callMode == 'paid-text' || $callMode == 'free-text' ) {
9277                                                            $callType = 'text';
9278                                                    }
9279
9280                                                    if ( $callMode == 'free' || $callMode == 'free-audio' || $callMode == 'free-text' ) {
9281                                                            $clientCPM = 0;
9282                                                    } else {
9283                                                            $clientCPM = self::clientCPM( $post->post_title, $options, $postID, $callType );
9284                                                    }
9285
9286                                                    //downgrade call if room in lesser mode
9287    if ($room_audio) if ($calltype != 'text') $calltype = 'audio';
9288                                                    if ($room_text) $calltype = 'text';
9289
9290                                                    if ( ! $clientCPM && ! $callMode ) {
9291                                                            $callMode = 'free';
9292                                                    }
9293
9294    $performer = self::performerName( $current_user, $options );
9295
9296                                                    $meta = array(
9297                                                            'time'      => time(),
9298                                                            'client'    => $client,
```

```
9299                                                    'email'     => $email,
9300                                                    'username'  => $login,
9301                                                    'callMode'  => $callMode,
9302                                                    'callType'  => $callType,
9303                                                    'clientCPM' => $clientCPM,
9304                                                );
9305
9306                                                $metaS     = serialize( $meta );
9307                                                $room_name = $post->post_name;
9308
9309                                                $sql = $wpdb->prepare(
9310                                                    "INSERT INTO `$table_private` ( `call`, `performer`, `pid`, `client`, `cid`, `rid`, `room`, `status`, `meta` ) VALUES ( %d, %s, %d, %s, %d, %d, %s, %d, %s )",
9311                                                    1, $performer, $current_user->ID, $user->user_login, $user->ID, $postID, $room_name, 0, $metaS
9312                                                );
9313                                                $wpdb->query( $sql );
9314                                                $privateID = $wpdb->insert_id;
9315
9316                                                $callURL = add_query_arg( array( 'call' => self::to62( $privateID ) ), get_permalink( $postID ) );
9317                                                $htmlCode .= __( 'New call was setup', 'ppv-live-webcams' ) . ': <br>' . $callURL;
9318
9319                                                $showNew = 0;
9320                                            } else {
9321                                                $htmlCode .= __( 'Error: Client was not found by email or username! User needs to register to be able to login and access the private call.', 'ppv-live-webcams' );
9322                                                $htmlCode .= '<br>"' . $client . '"';
9323                                            }
9324
9325                                            $htmlCode .= '</div>';
9326                                        }
9327
9328                                        $modeCode = '';
9329
9330                                        if ( $showNew && $postID ) {
9331
9332                                            $modeCode .= '<select style="opacity: 1;" name="callMode" id="callMode" class="ui dropdown v-select">';
9333
9334                                            // paid
9335                                            $clientCPM = self::clientCPM( $post->post_title, $options, $postID );
9336                                            if ( $clientCPM > 0 ) {
9337                                                if ( $options['modeVideo'] != '0' ) if (!$room_audio) if (!$room_text) $modeCode .= '<option value="paid" selected>' . __( 'Paid', 'ppv-live-webcams' ) . ': ' . __( 'Videochat', 'ppv-live-webcams' ) . ' ' . $clientCPM . htmlspecialchars( $options['currencypm'] ) . '</option>';
9338                                            }
9339
9340                                            $clientCPM = self::clientCPM( $post->post_title, $options, $postID, 'audio' );
9341                                            if ( $options['modeAudio'] ) if (!$room_text)  {
9342                                                if ( $clientCPM > 0 ) {
9343                                                    $modeCode .= '<option value="paid-audio">' . __( 'Paid', 'ppv-live-webcams' ) . ': ' . __( 'Audio', 'ppv-live-webcams' ) . ' ' . $clientCPM . htmlspecialchars( $options['currencypm'] ) . '</option>';
9344                                                }
9345                                            }
9346
9347                                            $clientCPM = self::clientCPM( $post->post_title, $options, $postID, 'text' );
9348                                            if ( $options['modeText'] ) {
```

```
9349                                                if ( $clientCPM > 0 ) {
9350                                                        $modeCode .= '<option value="paid-text">' . __( 'Paid', 'ppv-live-webcams' ) . ': ' . __( 'Text', 'ppv-live-webcams' ) . ' ' . $clientCPM . htmlspecialchars( $options['currencypm'] ) . '</option>';
9351                                                }
9352                                        }
9353
9354                                        // free
9355
if ( $options['modeVideo'] != '0' ) if (!$room_audio) if (!$room_text) $modeCode .= '<option value="free">' . __( 'Free', 'ppv-live-webcams' ) . ': ' . __( 'Videochat', 'ppv-live-webcams' ) . '</option>';
9356
9357                                        if ( $options['modeAudio'] ) if (!$room_text) {
9358                                                $modeCode .= '<option value="free-audio">' . __( 'Free', 'ppv-live-webcams' ) . ': ' . __( 'Audio', 'ppv-live-webcams' ) . '</option>';
9359                                        }
9360
9361                                        if ( $options['modeText'] ) {
9362                                                $modeCode .= '<option value="free-text">' . __( 'Free', 'ppv-live-webcams' ) . ': ' . __( 'Text', 'ppv-live-webcams' ) . '</option>';
9363                                        }
9364
9365                                        $modeCode .= '</select>';
9366
9367
$this_page = remove_query_arg( 'deletecall', self::getCurrentURL() );
9368                                        $action = add_query_arg( array( 'calls' => 'new' ), $this_page );
9369                                        $htmlCode .= '<form method="post" enctype="multipart/form-data" action="' . $action . '" name="callForm" class="ui form ' . $options['interfaceClass'] . ' segment">
9370                <div class="field"><label>' . __( 'Private Call Chat Mode', 'ppv-live-webcams' ) . '</label>' . $modeCode . '</div>
9371                <div class="field"><label>' . __( 'Client', 'ppv-live-webcams' ) . '</label> <input class="ui field" name="client" id="client" value="" placeholder="' . __( 'Username or Email', 'ppv-live-webcams' ) . '"></div>
9372                <input class="ui button videowhisperButton" type="submit" name="newcall" id="newcall" value="' . __( 'Setup Call', 'ppv-live-webcams' ) . '" />
9373                <input name="post_id" id="post_id" type="hidden" value="' . $postID . '">
9374 ';
9375                                        if ($room_audio) $htmlCode .= '<p> * ' . __( 'Warning: Room in audio only mode restricts call options.', 'ppv-live-webcams' ) . '</p>';
9376                                        if ($room_text) $htmlCode .= '<p> * ' . __( 'Warning: Room in text only mode restricts call options.', 'ppv-live-webcams' ) . '</p>';
9377
9378                                        $htmlCode .= '<p>' . __( 'Setup locked private calls that can only be accessed by client using dedicated call link. Clients can access their persistent calls list in client dashboard.  ', 'ppv-live-webcams' ) . '</p>
9379                        </form>';
9380
9381                                }
9382                        }
9383                }
9384
9385                // list calls
9386
9387                $htmlCode .= '<div class="ui segment">';
9388
9389                $uid = $current_user->ID;
9390
9391                if ( $delete = intval( $_GET['deletecall'] ?? 0 ) ) {
9392                        $sqlD = $wpdb->prepare("DELETE FROM $table_private WHERE `call` > 0 AND id = %d AND pid = %d", $delete, $uid);
9393                        $wpdb->query( $sqlD );
9394
9395                        $htmlCode .= '<div class="ui message">' . __( 'Call deleted.', 'ppv-live-webcams' ) . ' #' . $delete  . '</div>';
```

```
9396                }
9397
9398                if ( $postID ) {
9399                        $sql = $wpdb->prepare("SELECT * FROM $table_private WHERE rid = %d AND `call` > 0
ORDER BY status ASC, id DESC LIMIT 0, 100", $postID);
9400                } else {
9401                        $sql = $wpdb->prepare("SELECT * FROM $table_private WHERE (cid = %d OR pid = %d)
AND `call` > 0 ORDER BY status ASC, id DESC LIMIT 0, 100", $uid, $uid);
9402                }
9403
9404                $results = $wpdb->get_results( $sql );
9405                if ( $wpdb->num_rows > 0 ) {
9406                        $htmlCode .= '<div class="ui list">';
9407
9408                        foreach ( $results as $private ) {
9409                                $callCode = self::to62( $private->id );
9410                                $callURL
= add_query_arg( array( 'call' => $callCode ), get_permalink( $private->rid ) );
9411
9412                                $clientCPM = 0;
9413                                if ( $private->meta ) {
9414                                        $meta = unserialize( $private-
>meta, array( 'allowed_classes' => false ) );
9415                                        if ( array_key_exists( 'email', $meta ) ) {
9416                                                $metaInfo = $meta['email'];
9417                                        }
9418                                        if ( array_key_exists( 'clientCPM', $meta ) ) {
9419                                                $clientCPM = $meta['clientCPM'];
9420                                        }
9421                                }
9422
9423                                $htmlCode .= '<div class="item"> <i class="phone icon big"></i>';
9424                                $htmlCode .= '<div class="content">';
9425                                $htmlCode .= '<div class="header">' . $private->room . '
#' . $callCode . '   <i class="user icon"></i>' . ( $current_user->ID == $private->pid ? $private-
>client : $private->performer ) . '   ' . ( $clientCPM ? '<i class="money bill alternate outline
icon"></i>' . $clientCPM . ' ' . htmlspecialchars( $options['currencypm'] ) : '' ) . '</div>';
9426                                $htmlCode .= '<div class="description">' . $callURL . '</div>';
9427
9428                                $delCode = '<a href="' . add_query_arg( array( 'deletecall' => $private-
>id ), self::getCurrentURL() ) . '" class="ui button compact red"> <i class="close icon"></i>
' . __( 'Delete', 'ppv-live-webcams' ) . '</a>';
9429
9430                                if ( $private->status == 0 ) {
9431                                        $htmlCode .= '<div class="description"><a href="' . $callURL . '"
class="ui button compact green"> <i class="phone volume icon"></i> ' . __( 'Access', 'ppv-live-
webcams' ) . $delCode . '</a></div>';
9432                                } else {
9433                                        $htmlCode .= '<div class="description"> <div class="ui label"> <i
class="close icon"></i> ' . __( 'Closed', 'ppv-live-webcams' ) . '</div>' . $delCode . '</div>';
9434                                }
9435                                $htmlCode .= '</div></div>';
9436                        }
9437
9438                        $htmlCode .= '</div>';
9439
9440                } else {
9441                        $htmlCode .= __( 'No Calls', 'ppv-live-webcams' );
9442                }
9443
9444                $htmlCode .= '</div>';
9445
9446                return $htmlCode;
9447        }
```

```
9448
9449
9450        static function videowhisper_camvideo( $atts ) {
9451
9452                $atts = shortcode_atts(
9453                        array(
9454                                'cam'     => '',
9455                                'width'   => '640px',
9456                                'height'  => '480px',
9457                                'html5'   => 'auto',
9458                                'post_id' => 0,
9459                        ),
9460                        $atts,
9461                        'videowhisper_camvideo'
9462                );
9463                $atts = array_map('esc_attr', $atts);
9464
9465                $options = get_option( 'VWliveWebcamsOptions' );
9466
9467                $webcamName = sanitize_text_field( $atts['cam'] );
9468                if ( ! $webcamName ) {
9469                        $webcamName = sanitize_file_name( $_GET['cam'] );
9470                }
9471
9472                if ( $atts['post_id'] ) {
9473                        $postID = intval( $atts['post_id'] );
9474                }
9475
9476                if ( ! $postID ) {
9477                        global $wpdb;
9478                        $postID = $wpdb->get_var( $wpdb->prepare( 'SELECT ID FROM ' . $wpdb->posts . '
WHERE post_title = %s and post_type=%s LIMIT 0,1', $webcamName, $options['custom_post'] ) );
9479
9480                }
9481
9482                $roomInterface = get_post_meta( $postID, 'roomInterface', true ); // html5/flash
9483                $streamType    = get_post_meta( $postID, 'stream-type', true ); //
webrtc/flash/external/playlist
9484
9485                $stream = self::webcamStreamName( $webcamName, $postID );
9486
9487                $width = $atts['width'];
9488                if ( ! $width ) {
9489                        $width = '480px';
9490                }
9491                $height = $atts['height'];
9492                if ( ! $height ) {
9493                        $height = '360px';
9494                }
9495
9496                if ( ! $stream ) {
9497                        return 'Watch Video Error: Missing webcam stream name!';
9498                }
9499
9500                // HLS if iOS/Android detected
9501                $agent   = $_SERVER['HTTP_USER_AGENT'];
9502                $Android = stripos( $agent, 'Android' );
9503                $iOS     = ( strstr( $agent, 'iPhone' ) || strstr( $agent, 'iPod' ) ||
 strstr( $agent, 'iPad' ) );
9504                $Safari  = ( strstr( $agent, 'Safari' ) && ! strstr( $agent, 'Chrome' ) );
9505
```

```
9506                $htmlCode .= "<!--
CamVideo:$webcamName|$postID|$roomInterface|$stream|$Android|$iOS|$Safari dH:" . $options['detect_hls'] .
 '-->';
9507
9508                $showTeaser = 0;
9509                if ( $options['teaserOffline'] && ! self::webcamOnline( $postID ) ) {
9510                        $showTeaser = 1; // show offline teaser to client
9511                }
9512
9513                // webrtc
9514                if ( $streamType == 'webrtc' && ! $showTeaser ) {
9515                        return $htmlCode . '<!--H5-WebRTC-->' . do_shortcode( "
[videowhisper_cam_webrtc_playback
room=\"$webcamName\" webcam_id=\"$postID\" width=\"$width\" height=\"$height\" webstatus=\"1\"]" );
9516                }
9517
9518                // always
9519                if ( $atts['html5'] == 'always' ) {
9520                        if ( $iOS || $Safari ) {
9521                                return $htmlCode . '<!--H5-HLS-->' . do_shortcode( "[videowhisper_camhls
webcam=\"$webcamName\" width=\"$width\" height=\"$height\" webstatus=\"1\"]" );
9522                        } else {
9523                                return $htmlCode . '<!--H5-MPEG-->' . do_shortcode( "
[videowhisper_cammpeg webcam=\"$webcamName\" width=\"$width\" height=\"$height\" webstatus=\"1\"]" );
9524                        }
9525                }
9526
9527                // detect for video interface
9528                if ( ( $Android && in_array( $options['detect_mpeg'], array( 'android', 'all' ) ) ) ||
 ( ! $iOS && in_array( $options['detect_mpeg'], array( 'all' ) ) ) ||
 ( ! $iOS && ! $Safari && in_array( $options['detect_mpeg'], array( 'nonsafari' ) ) ) ) {
9529                        return $htmlCode . '<!--MPEG-->' . do_shortcode( "[videowhisper_cammpeg
webcam=\"$webcamName\" width=\"$width\" height=\"$height\" webstatus=\"1\"]" );
9530                }
9531
9532                if ( ( ( $Android ||
 $iOS ) && in_array( $options['detect_hls'], array( 'mobile', 'safari', 'all' ) ) ) ||
 ( $iOS && $options['detect_hls'] == 'ios' ) ||
 ( $Safari && in_array( $options['detect_hls'], array( 'safari', 'all' ) ) ) ) {
9533                        return $htmlCode . '<!--HLS-->' . do_shortcode( "[videowhisper_camhls
webcam=\"$webcamName\" width=\"$width\" height=\"$height\" webstatus=\"1\"]" );
9534                }
9535
9536                $afterCode = <<<HTMLCODE
9537 <style type="text/css">
9538 <!--
9539
9540 #videowhisper_container_$stream
9541 {
9542 position: relative;
9543 width: $width;
9544 height: $height;
9545 border: solid 1px #999;
9546 }
9547
9548 -->
9549 </style>
9550 HTMLCODE;
9551
9552                return $htmlCode . self::app_video( $stream, $webcamName, $width, $height ) . $afterCode;
9553
9554        }
9555
9556
```

```
9557        static function app_video( $stream, $room, $width = '100%', $height = '360px' ) {
9558                // Flash player no longer supported. Use HLS/MPEG or WebRTC players instead.
9559                return '<div class="ui segment"><p>' . __( 'Flash player is no longer supported. Please configure the room to use HTML5 streaming.', 'ppv-live-webcams' ) . '</p></div>';
9560        }
9561
9562
9563        static function videowhisper_cammpeg( $atts ) {
9564                // [videowhisper_cammpeg webcam="webcam name" width="480px" height="360px"]
9565
9566                $stream = '';
9567
9568                $options = get_option( 'VWliveWebcamsOptions' );
9569
9570                if ( is_single() ) {
9571                        if ( get_post_type( $postID = get_the_ID() ) == $options['custom_post'] ) {
9572                                $webcamName = self::post_title( get_the_ID() );
9573                        }
9574                }
9575
9576                        $atts = shortcode_atts(
9577                                array(
9578                                        'webcam' => $webcamName,
9579                                        'width'  => '480px',
9580                                        'height' => '360px',
9581                                ),
9582                                $atts,
9583                                'videowhisper_cammpeg'
9584                        );
9585                        $atts = array_map('esc_attr', $atts);
9586
9587                if ( ! $webcamName ) {
9588                        $webcamName = sanitize_text_field( $atts['webcam'] ); // parameter webcam="name"
9589                }
9590                if ( ! $webcamName ) {
9591                        $webcamName = sanitize_text_field( $_GET['n'] );
9592                }
9593
9594                if ( ! $postID ) {
9595                        global $wpdb;
9596                        $postID = $wpdb->get_var( $wpdb->prepare( "SELECT ID FROM $wpdb->posts WHERE post_title = %s and post_type=%s LIMIT 0,1", sanitize_file_name( $webcamName ), $options['custom_post'] ) );
9597                }
9598
9599                $stream = self::webcamStreamName( $webcamName, $postID, $options );
9600
9601                $width = $atts['width'];
9602                if ( ! $width ) {
9603                        $width = '480px';
9604                }
9605                $height = $atts['height'];
9606                if ( ! $height ) {
9607                        $height = '360px';
9608                }
9609
9610                if ( ! $stream ) {
9611                        return 'Watch MPEG Dash Error: Missing channel name!';
9612                }
9613
9614                $webcamOnline = self::webcamOnline( $postID );
9615                if ( $options['teaserOffline'] && ! $webcamOnline ) {
```

```
Line
9616                        $streamName = '';
9617                        $streamPath = '';
9618
9619                        $video_teaser = get_post_meta( $postID, 'video_teaser', true );
9620                        $streamURL    = self::vsvVideoURL( $video_teaser, $options );
9621                }
9622
9623                if ( $options['transcodingAuto'] && $webcamOnline ) {
9624
$streamName = self::transcodeStream( $stream, 1, $webcamName, 2, 1, $options, $postID ); // require
transcoding name
9625                }
9626
9627                if ( $streamName ) {
9628                        $streamURL = $options['httpstreamer'] . $streamName . '/manifest.mpd';
9629                }
9630
9631                if ( $streamURL ) {
9632                        // not used
9633                        // if (strstr($streamURL,'http://')) wp_enqueue_script('dashjs',
'http://cdn.dashjs.org/latest/dash.all.min.js');
9634                        // else wp_enqueue_script('dashjs',
'https://cdn.dashjs.org/latest/dash.all.min.js');
9635
9636                        // poster
9637                        $thumbUrl = self::webcamThumbSrc( $postID, $stream, $options );
9638
9639                        $htmlCode = <<<HTMLCODE
9640<!--MPEG Pi:$postID Wo:$webcamOnline Wn:$webcamName S:$stream Sn:$streamName T:$video_teaser
Su:$streamURL-->
9641<video id="videowhisper_mpeg_$stream" class="videowhisper_htmlvideo" width="$width" height="$height"
data-dashjs-player autobuffer autoplay loop playsinline muted="muted" controls="true" poster="$thumbUrl"
src="$streamURL">
9642    <div class="fallback" style="display:none">
9643    <IMG SRC="$thumbUrl">
9644            <p>HTML5 MPEG Dash capable browser (i.e. Chrome) is required to open this live stream:
$streamURL</p>
9645        </div>
9646</video>
9647<br>MPEG-DASH: Enable sound from browser controls. Transcoding and HTTP based delivery technology may
involve high latency.
9648HTMLCODE;
9649                } else {
9650                        $htmlCode = '<div class="warning">MPEG Dash format is not currently available for
this stream: ' . $stream . '</div>';
9651                }
9652
9653                return $htmlCode;
9654        }
9655
9656
9657        static function videowhisper_camhls( $atts ) {
9658
9659                // [videowhisper_camhls webcam="webcam name" width="480px" height="360px"]
9660
9661                $stream = '';
9662
9663                $options = get_option( 'VWliveWebcamsOptions' );
9664
9665                if ( is_single() ) {
9666                        if ( get_post_type( $postID = get_the_ID() ) == $options['custom_post'] ) {
9667                                $webcamName = self::post_title( get_the_ID() );
9668                        }
9669                }
```

```
Line
9670
9671                    $atts = shortcode_atts(
9672                            array(
9673                                    'webcam' => $webcamName,
9674                                    'width'  => '480px',
9675                                    'height' => '360px',
9676                            ),
9677                            $atts,
9678                            'videowhisper_camhls'
9679                    );
9680                    $atts = array_map('esc_attr', $atts);
9681
9682            if ( ! $webcamName ) {
9683                    $webcamName = sanitize_text_field( $atts['webcam'] ); // parameter webcam="name"
9684            }
9685            if ( ! $webcamName ) {
9686                    $webcamName = sanitize_text_field( $_GET['n'] );
9687            }
9688
9689            if ( ! $postID ) {
9690                    global $wpdb;
9691                    $postID = $wpdb->get_var( $wpdb->prepare( "SELECT ID FROM $wpdb->posts WHERE
post_title = %s and post_type=%s LIMIT
0,1", sanitize_file_name( $webcamName ), $options['custom_post'] ) );
9692            }
9693
9694            $stream = self::webcamStreamName( $webcamName, $postID, $options );
9695
9696            $width = $atts['width'];
9697            if ( ! $width ) {
9698                    $width = '480px';
9699            }
9700            $height = $atts['height'];
9701            if ( ! $height ) {
9702                    $height = '360px';
9703            }
9704
9705            if ( ! $stream ) {
9706                    return 'Watch HLS Error: Missing channel name!';
9707            }
9708
9709            $webcamOnline = self::webcamOnline( $postID );
9710            if ( $options['teaserOffline'] && ! $webcamOnline ) {
9711                    $streamName = '';
9712                    $streamPath = '';
9713
9714                    $video_teaser = get_post_meta( $postID, 'video_teaser', true );
9715                    $streamURL    = self::vsvVideoURL( $video_teaser, $options );
9716            }
9717
9718            // auto transcoding
9719            if ( $options['transcodingAuto'] && $webcamOnline ) {
9720
$streamName = self::transcodeStream( $stream, 1, $webcamName, 2, 1, $options, $postID ); // require
transcoding name
9721            }
9722
9723            if ( $streamName ) {
9724                    $streamURL = $options['httpstreamer'] . $streamName . '/playlist.m3u8';
9725            }
9726
9727            if ( $streamURL ) {
```

```
Line
9728                        // poster
9729                        $thumbUrl = self::webcamThumbSrc( $postID, $stream, $options );
9730
9731                        $htmlCode = <<<HTMLCODE
9732 <!--HLS:$postID:$webcamName:$stream-->
9733 <video id="videowhisper_hls_$stream" class="videowhisper_htmlvideo" width="$width" height="$height"
     autobuffer autoplay loop controls="true" poster="$thumbUrl">
9734  <source src="$streamURL" type='video/mp4'>
9735    <div class="fallback" style="display:none">
9736    <IMG SRC="$thumbUrl">
9737            <p>HTML5 HLS capable browser (i.e. Safari) is required to open this live stream:
     $streamURL</p>
9738        </div>
9739 </video>
9740 <br>HLS: Enable sound from your browser controls. Transcoding and HTTP based delivery technology may
     involve higher latency.
9741
9742 <br>
9743 HTMLCODE;
9744                } else {
9745                        $htmlCode = '<div class="warning">HLS format is not currently available for this
     stream: ' . $stream . '</div>';
9746                }
9747
9748                return $htmlCode;
9749        }
9750
9751
9752        static function videowhisper_campreview( $atts ) {
9753
9754                $options = get_option( 'VWliveWebcamsOptions' );
9755
9756                $atts = shortcode_atts(
9757                        array(
9758                                'status'   => 'online',
9759                                'order_by' => 'rand',
9760                                'category' => '',
9761                                'perpage'  => 1,
9762                                'perrow'   => 2,
9763                                'width'    => '480px',
9764                                'height'   => '360px',
9765                        ),
9766                        $atts,
9767                        'videowhisper_campreview'
9768                );
9769                $atts = array_map('esc_attr', $atts);
9770
9771                $perPage = intval( $atts['perpage'] );
9772                $perRow  = intval( $atts['perrow'] );
9773
9774                $pstatus  = sanitize_text_field( $atts['status'] );
9775                $order_by = sanitize_text_field( $atts['order_by'] );
9776                $category = sanitize_text_field( $atts['category'] );
9777
9778                $width = sanitize_text_field( $atts['width'] );
9779                if ( ! $width ) {
9780                        $width = '480px';
9781                }
9782                $height = sanitize_text_field( $atts['height'] );
9783                if ( ! $height ) {
9784                        $height = '360px';
9785                }
```

```
9786
9787                $args = array(
9788                        'post_type'      => sanitize_text_field( $options['custom_post'] ),
9789                        'post_status'    => 'publish',
9790                        'posts_per_page' => $perPage,
9791                        'offset'         => 0,
9792                        'order'          => 'DESC',
9793                        'meta_query'     => array(
9794                                'relation'    => 'AND',
9795                                'snapshot'    => array(
9796                                        'key'   => 'hasSnapshot',
9797                                        'value' => '1',
9798                                ),
9799                                'vw_featured' => array(
9800                                        'key'     => 'vw_featured',
9801                                        'compare' => 'EXISTS',
9802                                ),
9803                                'edate'       => array(
9804                                        'key'     => 'edate',
9805                                        'compare' => 'EXISTS',
9806                                ),
9807                        ),
9808                );
9809
9810                if ($pstatus) switch ( $pstatus ) {
9811
9812                        case 'free':
9813                                $args['meta_query']['public']   = array(
9814                                        'key'   => 'privateShow',
9815                                        'value' => '0',
9816                                );
9817                                $args['meta_query']['groupCPM'] = array(
9818                                        'key'   => 'groupCPM',
9819                                        'value' => '0',
9820                                );
9821                                break;
9822                        case 'paid':
9823                                $args['meta_query']['public']   = array(
9824                                        'key'   => 'privateShow',
9825                                        'value' => '0',
9826                                );
9827                                $args['meta_query']['groupCPM'] = array(
9828                                        'key'     => 'groupCPM',
9829                                        'value'   => '0',
9830                                        'compare' => '>',
9831                                );
9832                                break;
9833                        case 'available_free':
9834                                $args['meta_query']['public']   = array(
9835                                        'key'   => 'privateShow',
9836                                        'value' => '0',
9837                                );
9838                                $args['meta_query']['online']   = array(
9839                                        'key'     => 'edate',
9840                                        'value'   => time() - 40,
9841                                        'compare' => '>',
9842                                );
9843                                $args['meta_query']['groupCPM'] = array(
9844                                        'key'   => 'groupCPM',
9845                                        'value' => '0',
9846                                );
```

```
9847                                    break;
9848                            case 'available_paid':
9849                                    $args['meta_query']['public']   = array(
9850                                            'key'   => 'privateShow',
9851                                            'value' => '0',
9852                                    );
9853                                    $args['meta_query']['online']   = array(
9854                                            'key'     => 'edate',
9855                                            'value'   => time() - 40,
9856                                            'compare' => '>',
9857                                    );
9858                                    $args['meta_query']['groupCPM'] = array(
9859                                            'key'     => 'groupCPM',
9860                                            'value'   => '0',
9861                                            'compare' => '>',
9862                                    );
9863                                    break;
9864
9865                            case 'private':
9866                                    $args['meta_query']['private'] = array(
9867                                            'key'   => 'privateShow',
9868                                            'value' => '1',
9869                                    );
9870                                    $args['meta_query']['online']  = array(
9871                                            'key'     => 'edate',
9872                                            'value'   => time() - 40,
9873                                            'compare' => '>',
9874                                    );
9875                                    break;
9876
9877                            case 'online':
9878                                    $args['meta_query']['online'] = array(
9879                                            'key'     => 'edate',
9880                                            'value'   => time() - 40,
9881                                            'compare' => '>',
9882                                    );
9883                                    break;
9884
9885                            case 'public':
9886                                    $args['meta_query']['public'] = array(
9887                                            'key'   => 'privateShow',
9888                                            'value' => '0',
9889                                    );
9890                                    $args['meta_query']['online'] = array(
9891                                            'key'     => 'edate',
9892                                            'value'   => time() - 40,
9893                                            'compare' => '>',
9894                                    );
9895                                    break;
9896
9897                            case 'offline':
9898                                    $args['meta_query']['offline'] = array(
9899                                            'key'     => 'edate',
9900                                            'value'   => time() - 40,
9901                                            'compare' => '<',
9902                                    );
9903                                    break;
9904                    }
9905
9906                    switch ( $order_by ) {
9907
```

```
9908                        case 'default':
9909                                $args['orderby'] = array(
9910                                        'vw_featured' => 'DESC',
9911                                        'edate'       => 'DESC',
9912                                );
9913
9914                                break;
9915
9916                        case 'post_date':
9917                                $args['orderby'] = 'post_date';
9918                                break;
9919
9920                        case 'rand':
9921                                $args['orderby'] = 'rand';
9922                                break;
9923
9924                        default:
9925                                $args['orderby']  = 'meta_value_num';
9926                                $args['meta_key'] = $order_by;
9927                                break;
9928                }
9929
9930                if ( $category ) {
9931                        $args['category'] = $category;
9932                }
9933
9934                $postslist = get_posts( $args );
9935
9936                // list cam previews
9937
9938                $code = '';
9939
9940                if ( count( $postslist ) > 0 ) {
9941                        $k = 0;
9942                        foreach ( $postslist as $item ) {
9943                                if ( $perRow ) {
9944                                        if ( $k ) {
9945                                                if ( $k % $perRow == 0 ) {
9946 $code .= '<br>';
9947                                                }
9948                                        }
9949                                }
9950
9951                                                $webcamName = sanitize_file_name( $item->post_title );
9952
9953                                        $code .= do_shortcode( "[videowhisper_camvideo
cam=\"$webcamName\" width=\"$width\" height=\"$height\"]" );
9954                        }
9955                } else {
9956                        $code .= 'No cams currently match preview criteria.';
9957                        // $code .= var_export($args);
9958                }
9959
9960                return $code;
9961        }
9962
9963
9964        static function videowhisper_camprofile( $atts ) {
9965                $atts = shortcode_atts(
9966                        array(
9967                                'cam'     => $stream,
```

```
9968                                    'post_id' => 0,
9969                            ),
9970                            $atts,
9971                            'videowhisper_camprofile'
9972                    );
9973                    $atts = array_map('esc_attr', $atts);
9974
9975                    $stream = sanitize_text_field( $atts['cam'] );
9976                    if ( ! $stream ) {
9977                            $stream = sanitize_text_field( $_GET['cam'] );
9978                    }
9979                    $stream = sanitize_file_name( $stream );
9980
9981                    $options = get_option( 'VWliveWebcamsOptions' );
9982
9983                    $post_id = intval( $atts['post_id'] );
9984
9985                    if ( $post_id ) {
9986                            $postID = $post_id;
9987                    } elseif ( $stream ) {
9988                            global $wpdb;
9989                            $postID = $wpdb->get_var( $wpdb->prepare( 'SELECT ID FROM ' . $wpdb->posts . '
WHERE post_title = %s and post_type=%s LIMIT 0,1', $stream, $options['custom_post'] ) );
9990                    }
9991
9992                    if ( ! $postID ) {
9993                            return "videowhisper_camprofile error: Webcam not found ($stream/$post_id)!";
9994                    }
9995
9996                    return self::webcamProfile( $postID, $options );
9997            }
9998
9999
10000           static function videowhisper_camcontent( $atts ) {
10001                   $atts = shortcode_atts(
10002                           array(
10003                                   'cam'     => $stream,
10004                                   'post_id' => 0,
10005                           ),
10006                           $atts,
10007                           'videowhisper_camcontent'
10008                   );
10009                   $atts = array_map('esc_attr', $atts);
10010
10011                   $stream = sanitize_text_field( $atts['cam'] );
10012                   if ( ! $stream ) {
10013                           $stream = sanitize_text_field( $_GET['cam'] );
10014                   }
10015                   $stream = sanitize_file_name( $stream );
10016
10017                   $options = get_option( 'VWliveWebcamsOptions' );
10018
10019                   $post_id = intval( $atts['post_id'] );
10020
10021                   if ( $post_id ) {
10022                           $postID = $post_id;
10023                   } elseif ( $stream ) {
10024                           global $wpdb;
10025                           $postID = $wpdb->get_var( $wpdb->prepare( 'SELECT ID FROM ' . $wpdb->posts . '
WHERE post_title = %s and post_type=%s LIMIT 0,1', $stream, $options['custom_post'] ) );
10026                   }
10027
```

```
Line
10028                 if ( $postID && ! $stream ) {
10029                         global $wpdb;
10030                         $stream = $wpdb->get_var( $wpdb->prepare( 'SELECT post_title FROM ' . $wpdb->posts . ' WHERE ID = %d and post_type=%s LIMIT 0,1', $postID, $options['custom_post'] ) );
10031                 }
10032
10033                 if ( ! $postID || ! $stream ) {
10034                         return "videowhisper_camcontent error: Webcam not found ($stream/$post_id)!";
10035                 }
10036
10037                 return self::webcamContent( $postID, $stream, $options, '' );
10038         }
10039
10040
10041         static function videowhisper_caminfo( $atts ) {
10042                 $atts = shortcode_atts(
10043                         array(
10044                                 'cam'    => '',
10045                                 'info'   => 'cpm',
10046                                 'format' => 'csv',
10047                         ),
10048                         $atts,
10049                         'videowhisper_caminfo'
10050                 );
10051                 $atts = array_map('esc_attr', $atts);
10052
10053                 $stream = sanitize_text_field( $atts['cam'] );
10054                 if ( ! $stream ) {
10055                         $stream = sanitize_text_field( $_GET['cam'] );
10056                 }
10057                 $stream = sanitize_file_name( $stream );
10058
10059                 if ( ! $stream ) {
10060                         return 'No cam name!';
10061                 }
10062
10063                 $options = get_option( 'VWliveWebcamsOptions' );
10064
10065                 // custom room cost per minute
10066                 global $wpdb;
10067                 $postID = $wpdb->get_var( $wpdb->prepare( 'SELECT ID FROM ' . $wpdb->posts . ' WHERE post_title = %s and post_type=%s LIMIT 0,1', $stream, $options['custom_post'] ) );
10068
10069                 if ( ! $postID ) {
10070                         return "Cam '$stream' not found!";
10071                 }
10072
10073                 switch ( $atts['info'] ) {
10074                         case 'cpm':
10075                                 $result = self::clientCPM( $stream, $options, $postID );
10076                                 break;
10077
10078                         case 'online':
10079                                 $edate  = get_post_meta( $postID, 'edate', true );
10080                                 $result = self::format_age( time() - $edate );
10081                                 break;
10082
10083                         case 'label':
10084                                 $result = get_post_meta( $postID, 'vw_roomLabel', true );
10085                                 break;
10086
10087                         case 'brief':
```

```
10088                                   $result = get_post_meta( $postID, 'vw_roomBrief', true );
10089                                   break;
10090
10091                           case 'tags':
10092                                   $tags     = wp_get_post_tags( $postID, array( 'fields' => 'names' ) );
10093                                   $roomTags = '';
10094                                   if ( ! empty( $tags ) ) {
10095                                           if ( ! is_wp_error( $tags ) ) {
10096                                                   $result = $tags;
10097                                           }
10098                                   }
10099                                   break;
10100
10101                           case 'performers':
10102                                   $checkin = get_post_meta( $postID, 'checkin', true );
10103                                   if ( $checkin ) {
10104                                           foreach ( $checkin as $performerID ) {
10105 $result[] = self::performerLink( $performerID, $options );
10106                                           }
10107                                   }
10108
10109                                   break;
10110
10111                           case 'groupMode':
10112                                   $result = get_post_meta( $postID, 'groupMode', true );
10113                                   break;
10114
10115                           case 'groupCPM':
10116                                   $result = get_post_meta( $postID, 'groupCPM', true );
10117                                   break;
10118
10119                           default:
10120                                   $result = 'Info type incorrect!';
10121                   }
10122
10123                   switch ( $atts['format'] ) {
10124                           case 'csv':
10125                                   if ( is_array( $result ) ) {
10126                                           foreach ( $result as $tag ) {
10127                                                   $roomTags .= ( $roomTags ? ', ' : '' ) . $tag;
10128                                           }
10129                                           return $roomTags;
10130                                   } else {
10131                                           return $result;
10132                                   }
10133
10134                                   break;
10135
10136                           case 'serialize':
10137                                   return serialize( $result );
10138                           break;
10139
10140                           default:
10141                                   return $result;
10142
10143                           break;
10144                   }
10145
10146           }
10147
```

```
10148
10149           static function videowhisper_webcams( $atts ) {
10150                   $options = get_option( 'VWliveWebcamsOptions' );
10151
10152                   // shortocode attributes
10153                   $atts = shortcode_atts(
10154                           array(
10155                                   'layout'
=> sanitize_text_field( $options['layoutDefault'] ?? 'grid' ),
10156                                   'menu'            => sanitize_text_field( $options['listingsMenu']
?? 'auto' ),
10157                                   'perpage'         => intval( $options['perPage'] ?? 12 ),
10158                                   'ban'             => '0',
10159                                   'perrow'          => '',
10160                                   'order_by'        => 'default',
10161                                   'category_id'
 => isset( $_GET['category_id'] ) ? intval( $_GET['category_id'] ) : '',
10162                                   'pstatus'
 => isset( $_GET['pstatus'] ) ? sanitize_text_field( $_GET['pstatus'] ) : '',
10163                                   'lovense'
 => isset( $_GET['lovense'] ) ? sanitize_text_field( $_GET['lovense'] ) : '',
10164                                   'select_category' => '1',
10165                                   'select_filters'  => '1',
10166                                   'select_tags'     => '1',
10167                                   'select_name'     => '1',
10168                                   'select_order'    => '1',
10169                                   'select_status'   => '1',
10170                                   'select_page'     => '1',
10171                                   'select_layout'   => '1',
10172                                   'include_css'     => '1',
10173                   //              'url_vars'        => '1',
10174                   //              'url_vars_fixed'  => '1',
10175                                   'persistent_off' => '0',
10176                                   'studio_id'       => '',
10177                                   'author_id'       => '',
10178                                   'tags'
=> isset( $_GET['tags'] ) ? sanitize_text_field( $_GET['tags'] ) : '',
10179                                   'name'
=> isset( $_GET['name'] ) ? sanitize_text_field( $_GET['name'] ) : '',
10180                                   'id'              => '',
10181                           ),
10182                           $atts,
10183                           'videowhisper_webcams'
10184                   );
10185                   $atts = array_map('esc_attr', $atts);
10186
10187                   $id = sanitize_text_field( $atts['id'] );
10188                   if ( ! $id ) {
10189                           $id = 1; // uniqid();
10190                   }
10191
10192                   // get vars from url
10193                   /*
10194                   if ( $atts['url_vars'] ) {
10195                           $cid = intval( $_GET['cid'] ?? 0 );
10196                           if ( $cid ) {
10197                                   $atts['category_id'] = $cid;
10198                                   if ( $atts['url_vars_fixed'] ) {
10199                                           $atts['select_category'] = '0';
10200                                   }
10201                           }
10202                   }
10203                   */
```

```
10204
10205                 $htmlCode = '';
10206
10207                 // semantic ui : listings
10208                 self::enqueueUI();
10209
10210                 // ajax url
10211                 $ajaxurl = admin_url() . 'admin-ajax.php?
action=vmls_cams&menu=' . $atts['menu'] . '&layout=' . $atts['layout'] . '&&pp=' . $atts['perpage'] . '&p
r=' . $atts['perrow'] . '&ob=' . $atts['order_by'] . '&cat=' . $atts['category_id'] . '&st=' . $atts['pst
atus'] . '&lt=' . $atts['lovense'] . '&sc=' . $atts['select_category'] . '&sf=' . $atts['select_filters']

. '&so=' . $atts['select_order'] . '&ss=' . $atts['select_status'] . '&sp=' . $atts['select_page'] . '&sl
=' . $atts['select_layout'] . '&sn=' . $atts['select_name'] . '&sg=' . $atts['select_tags'] . '&id=' . es
c_attr( $id ) . '&tags=' . urlencode( $atts['tags'] ) . '&name=' . urlencode( $atts['name'] );
10212
10213                 if ( $atts['studio_id'] ) {
10214                         $ajaxurl .= '&studio_id=' . intval( $atts['studio_id'] );
10215                 }
10216
10217                 if ( $atts['author_id'] ) {
10218                         $ajaxurl .= '&author_id=' . intval( $atts['author_id'] );
10219                 }
10220
10221                 if ( $atts['ban'] ) {
10222                         $ajaxurl .= '&ban=' . $atts['ban'];
10223                 }
10224
10225                                 // custom profile filters
10226                                 if (is_array($options['profileFields']))
10227                                 foreach ( $options['profileFields'] as $field => $parameters ) {
10228
if ( isset($parameters['filter']) && isset($parameters['options']) ) {
10229                                                 $fieldName  = sanitize_title( trim( $field ) );
10230                                                 $fieldMeta  = 'vwf_' . $fieldName;
10231
$fieldValue = sanitize_text_field( $_GET[ $fieldMeta ] ?? '' );
10232                                                 if ( $fieldValue ) {
10233
$ajaxurl .= '&' . $fieldMeta . '=' . urlencode( $fieldValue );
10234                                                 }
10235                                         }
10236                                 }
10237
10238                 $loadingMessage = '<div class="ui active inline text large loader">' . __( 'Loading Live
Webcams', 'ppv-live-webcams' ) . '...</div>';
10239
10240                 if ( ($options['filtersSave'] ?? false) && !$atts['persistent_off'])
10241                 if (is_user_logged_in())
10242                 {
10243                         $ajaxURLall = get_user_meta( get_current_user_id(), 'vwf_filters_url', true );
10244                         if ($ajaxURLall )
10245                         {
10246                                 $ajaxurl = $ajaxURLall;
10247                                 $loadingMessage = '<div class="ui active inline text large
loader">' . __( 'Loading your last search', 'ppv-live-webcams' ) . '...</div>';
10248                         }
10249                 }
10250
10251
10252                 $codeReload = '';
10253
10254                 if (!$options['debugMode']) $codeReload = 'setInterval("loadWebcams'. $id . '()",
15000);';
```

```
10255
10256                $htmlCode .= <<<HTMLCODE
10257 <PRE style="display: none">
10258 <SCRIPT>
10259 var aurl$id = '$ajaxurl';
10260 var \$j = jQuery.noConflict();
10261 var loader$id;
10262
10263         function loadWebcams$id(message){
10264
10265         if (message)
10266         if (message.length > 0)
10267         {
10268           \$j("#videowhisperWebcams$id").html(message);
10269         }
10270
10271                 if (loader$id) loader$id.abort();
10272
10273                 loader$id = \$j.ajax({
10274                         url: aurl$id,
10275                         data: "interfaceid=$id",
10276                         success: function(data) {
10277                                 \$j("#videowhisperWebcams$id").html(data);
10278                                 jQuery(".ui.dropdown:not(.multi,.fpsDropdown)").dropdown();
10279                                 jQuery(".ui.rating.readonly").rating("disable");
10280                         }
10281                 });
10282         }
10283
10284         jQuery(document).ready(function(){
10285                 loadWebcams$id();
10286                 $codeReload
10287         });
10288
10289 </SCRIPT>
10290 </PRE>
10291 <div id="videowhisperWebcams$id" class="videowhisperWebcams">
10292     $loadingMessage
10293 </div>
10294 <div style="clear:both"><div/>
10295 HTMLCODE;
10296
10297                 if ( $atts['include_css'] ) {
10298
$htmlCode .= '<STYLE>' . html_entity_decode( stripslashes( $options['customCSS'] ) ) . '</STYLE>';
10299                 }
10300
10301                 return $htmlCode;
10302         }
10303
10304
10305         // ! AJAX Webcams List
10306         static function vmls_cams_callback() {
10307                  // ajax called
10308
10309                 // cam meta:
10310                 // edate s
10311                 // viewers n
10312                 // maxViewers n
10313                 // maxDate s
10314                 // hasSnapshot 1
```

```
10315                 // privateShow 0
10316
10317                 $options = get_option( 'VWliveWebcamsOptions' );
10318
10319                 $isMobile = (bool) preg_match( '#\b(ip(hone|od|ad)|android|opera m(ob|in)i|windows (phone|ce)|blackberry|tablet|s(ymbian|eries60|amsung)|p(laybook|alm|rofile/midp|laystation portable)|nokia|fennec|htc[\-_]|mobile|up\.browser|[1-4][0-9]{2}x[1-4][0-9]{2})\b#i', $_SERVER['HTTP_USER_AGENT'] );
10320
10321                 // output clean (clear 0)
10322                 if (ob_get_length()) {
10323                         ob_clean();
10324                 }
10325
10326                 // cache : do not generate more often than 14s (client refresh each 15s)
10327                 // client still updates but less load on high volume (many users online)
10328
10329                 /*
10330                 $cacheQuery = $_SERVER['QUERY_STRING'] . '&mob='.$isMobile . '&login=' . is_user_logged_in();
10331                 $cacheKey = sha1($cacheQuery);
10332
10333                 //$cachePath
10334                 $cachePath = $options['uploadsPath'];
10335                 if (!file_exists($cachePath)) mkdir($cachePath);
10336                 $cachePath .= "/_cache/";
10337                 if (!file_exists($cachePath)) mkdir($cachePath);
10338                 $cachePath .= $cacheKey;
10339
10340                 if (!self::timeTo('cl_' . $cacheKey, 9, $options))
10341                         if (file_exists($cachePath))
10342                         {
10343                                 echo self::stringLoad($cachePath);
10344                                 exit;
10345                         }
10346                 */
10347
10348                 if ( $options['videosharevod'] ) {
10349                         $optionsVSV = get_option( 'VWvideoShareOptions' );
10350                 } else $optionsVSV = null;
10351
10352                 $debugCode = '';
10353                 // safari requires muted for autoplay
10354
$isSafari = (bool) ( strpos( $_SERVER['HTTP_USER_AGENT'], 'AppleWebKit' ) && strpos( $_SERVER['HTTP_USER_AGENT'], 'Safari' ) );
10355                 if ( $isSafari ) {
10356                         $previewMuted = 'muted';
10357                 } else {
10358                         $previewMuted = '';
10359                 }
10360
10361                 $isLoggedin = is_user_logged_in();
10362                 $isModerator = 0;
10363                 if  ( $isLoggedin ) $isModerator = self::isModerator( get_current_user_id(), $options);
10364
10365
10366                 $debugMode = $options['debugMode'];
10367
10368                 // widget id, to avoid duplicate function names if using multiple on same page
10369                 $id = sanitize_file_name( $_GET['id'] ?? '' );
10370
10371                 // pagination
```

```
Line
10372                   $perPage = intval( $_GET['pp'] ?? 0 );
10373                   if ( ! $perPage ) {
10374                           $perPage = $options['perPage'];
10375                   }
10376
10377                   $page   = intval( $_GET['p'] ?? 0 );
10378                   $offset = $page * $perPage;
10379
10380                   $perRow = intval( $_GET['pr'] ?? 0 );
10381
10382                   $json = boolval($_GET['json'] ?? false);
10383                   $result = [ 'rooms' => [] ];
10384
10385                   // admin side
10386                   $ban = intval( $_GET['ban'] ?? 0 );
10387
10388                   $category = intval( $_GET['cat'] ?? 0 );
10389                   $pstatus          = sanitize_text_field( $_GET['st'] ?? '');
10390                   $ptoy = sanitize_text_field( $_GET['lt'] ?? '');
10391
10392                   $menu             = sanitize_text_field( $_GET['menu'] ?? '');
10393                   if ( $menu == 'auto' && $isMobile )  $menu = false;             //auto disables on mobile
10394
10395                   // order
10396                   $order_by = sanitize_text_field( $_GET['ob'] );
10397                   if ( ! $order_by ) {
10398                           $order_by = 'default';
10399                   }
10400
10401                   // options
10402                   $selectCategory = intval( $_GET['sc'] ?? 0 );
10403                   $selectOrder    = intval( $_GET['so'] ?? 0 );
10404                   $selectPage     = intval( $_GET['sp'] ?? 0 );
10405                   $selectLayout   = intval( $_GET['sl'] ?? 0 );
10406
10407                   $selectStatus = intval( $_GET['ss'] ?? 0 );
10408
10409                   $selectName = intval( $_GET['sn'] ?? 0 );
10410                   $selectTags = intval( $_GET['sg'] ?? 0 );
10411
10412                   $selectFilters = intval( $_GET['sf'] ?? 0 );
10413
10414                   // studio/user
10415                   $studio_id = intval( $_GET['studio_id'] ?? 0 );
10416                   if ($studio_id == -1) $studio_id = get_current_user_id();
10417
10418                   $author_id = intval( $_GET['author_id'] ?? 0 );
10419                   if ($author_id == -1) $author_id = get_current_user_id();
10420
10421                   // tags,name search
10422                   $tags = sanitize_text_field( $_GET['tags'] ?? '' );
10423                   $name = sanitize_text_field( $_GET['name'] ?? '');
10424                   if ( $name == 'undefined' ) {
10425                           $name = '';
10426                   }
10427                   if ( $tags == 'undefined' ) {
10428                           $tags = '';
10429                   }
10430
10431                   // layout: grid/list/horizontal
10432                   $layout = sanitize_text_field( $_GET['layout'] ?? '' );
```

```
10433                 if ( ! $layout ) {
10434                         $layout = $options['layoutDefault'];
10435                 }
10436                 if ( ! $layout ) {
10437                         $layout = 'grid';
10438                 }
10439
10440                 // thumbs dir
10441                 $dir = $options['uploadsPath'] . '/_thumbs';
10442
10443                 $ajaxurl = admin_url() . 'admin-ajax.php?
action=vmls_cams&menu=' . $menu . '&pp=' . $perPage . '&pr=' . $perRow . '&ss=' . $selectStatus . '&sc='
. $selectCategory . '&so=' . $selectOrder . '&sn=' . $selectName . '&sg=' . $selectTags . '&sp=' . $selec
tPage . '&sl=' . $selectLayout . '&id=' . esc_attr( $id ) . '&tags=' . urlencode( $tags ) . '&name=' . ur
lencode( $name ) . '&sf=' . $selectFilters;
10444
10445                 if ( $studio_id ) {
10446                         $ajaxurl .= '&studio_id=' . $studio_id;
10447                 }
10448
10449                 if ( $author_id ) {
10450                         $ajaxurl .= '&author_id=' . $author_id;
10451                 }
10452
10453                 if ( $ban ) {
10454                         $ajaxurl .= '&ban=' . $ban; // admin side
10455                 }
10456
10457
10458 if ($json)
10459 {
10460         //setup extra fields
10461         $json_fields = [];
10462         foreach ( $options['profileFields'] as $field => $parameters )
10463         if ( ( isset($parameters['filter']) && $parameters['filter'] ) ||
 ( isset ( $parameters['json']) && $parameters['json'] ) )
10464         {       $fieldName  = sanitize_title( trim( $field ) );
10465                 $json_fields[] = 'vwf_' . $fieldName ;
10466         }
10467 }
10468                 //$ajaxURLpag   = $ajaxurl . '&p=' . $page . '&layout=' . $layout. '&st=' . $pstatus;
10469                 $ajaxURLord
 = $ajaxurl . '&cat=' . $category . '&layout=' . $layout . '&st=' . $pstatus . '&lt=' . $ptoy; //order by
10470                 $ajaxURLcat
 = $ajaxurl . '&ob=' . $order_by . '&layout=' . $layout . '&st=' . $pstatus. '&lt=' . $ptoy; // category
select
10471
$ajaxURLstat = $ajaxurl . '&cat=' . $category . '&ob=' . $order_by . '&layout=' . $layout. '&lt=' . $ptoy
; // status select
10472
$ajaxURLtoy = $ajaxurl . '&cat=' . $category . '&ob=' . $order_by . '&layout=' . $layout . '&st=' . $psta
tus; // status toy
10473
10474
$ajaxURLlay = $ajaxurl . '&cat=' . $category . '&ob=' . $order_by . '&p=' . $page. '&st=' . $pstatus . '&
lt=' . $ptoy; // layout select
10475
10476
10477
10478
$ajaxURLall = $ajaxurl . '&cat=' . $category . '&ob=' . $order_by . '&layout=' . $layout . '&st=' . $psta
tus. '&lt=' . $ptoy; // all persist: reload, page, search
10479
10480                 // $htmlCode .= '<div class="videowhisperListOptions">';
10481
```

```
// menu start
                if ( $menu && !$json ) {
                        echo '
<style>
        .vwRoomsSidebar {
    grid-area: sidebar;
  }

  .vwRoomsContent {
    grid-area: content;
    position: relative;
    display: grid;
  }

.vwRoomsWrapper {
        position: relative;
    display: grid;
    grid-gap: 4px;
    grid-template-columns: 120px  auto;
    grid-template-areas: "sidebar content";
    color: #444;
  }

  .ui .title { height: auto !important; background-color: inherit !important}
  .ui .content {margin: 0 !important; }
  .vwRoomsSidebar .menu { max-width: 120px !important;}

 </style>
 <div class="vwRoomsWrapper">
 <div class="vwRoomsSidebar">';

                        if ( $selectCategory ) {
                                echo '
<div class="ui ' . esc_attr( $options['interfaceClass'] ) . ' accordion small vwFields">

  <div class="title">
    <i class="list icon"></i>
    ' . __( 'Category', 'ppv-live-webcams' ) . ' ' . ( esc_html( $category ) ? '<i class="check icon small"></i>' : '' ) . '
  </div>
  <div class="' . ( $category ? 'active' : '' ) . ' content">
  <div class="ui ' . esc_attr( $options['interfaceClass'] ) . ' vertical menu small">
  ';
                                echo '  <a class="' . ( $category == 0 ? 'active' : '' ) . ' item" onclick="aurl' . esc_attr( $id ) . '=\'' . esc_url( $ajaxURLcat ) . '&cat=0\'; loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large loader\\\'>' . __( 'Loading category', 'ppv-live-webcams' ) . '...</div>\')">' . __( 'All Categories', 'ppv-live-webcams' ) . '</a> ';

                                $categories = get_categories( array( 'taxonomy' => 'category' ) );
                                foreach ( $categories as $cat ) {
                                        echo '  <a class="' . ( $category == $cat->term_id ? 'active' : '' ) . ' item" onclick="aurl' . esc_attr( $id ) . '=\'' . esc_html( $ajaxURLcat ) . '&cat=' . esc_attr( $cat->term_id ) . '\'; loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large loader\\\'>' . __( 'Loading category', 'ppv-live-webcams' ) . '...</div>\')">' . esc_html( $cat->name ) . '</a> ';
                                }

                                echo '</div>

  </div>
</div>';
```

```
10535
10536                       }
10537
10538                       if ( $selectStatus ) {
10539                               echo '
10540 <div class="ui ' . esc_attr( $options['interfaceClass'] ) . ' accordion small vwFields">
10541
10542   <div class="title">
10543     <i class="power off icon"></i>
10544     ' . __( 'Status', 'ppv-live-webcams' ) . ' ' . ( $pstatus ? '<i class="check icon small">
      </i>' : '' ) . '
10545   </div>
10546
10547   <div class="' . ( $pstatus != '' ? 'active' : '' ) . ' content">
10548   <div class="ui ' . esc_attr( $options['interfaceClass'] ) . ' vertical menu small">
10549
10550   <a class="' . ( $pstatus == '' ? 'active' : '' ) . ' item"
      onclick="aurl' . esc_attr( $id ) . '=\'' . esc_url( $ajaxURLstat) . '&st=\';
      loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large
      loader\\\'>' . __( 'Updating listings, please wait', 'ppv-live-webcams' ) . '...
      </div>\')">' . esc_attr(self::label( 'All', __( 'All', 'ppv-live-webcams' ), $options )) . '</a>
10551
10552   <a class="' . ( $pstatus == 'online' ? 'active' : '' ) . ' item"
      onclick="aurl' . esc_attr( $id ) . '=\'' . esc_url( $ajaxURLstat) . '&st=online\';
      loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large
      loader\\\'>' . __( 'Updating listings, please wait', 'ppv-live-webcams' ) . '...
      </div>\')">' . esc_attr(self::label( 'Online', __( 'Online', 'ppv-live-webcams' ), $options )) . '</a>
10553   <a class="' . ( $pstatus == 'public' ? 'active' : '' ) . ' item"
      onclick="aurl' . esc_attr( $id ) . '=\'' . esc_url( $ajaxURLstat) . '&st=public\';
      loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large
      loader\\\'>' . __( 'Updating listings, please wait', 'ppv-live-webcams' ) . '...
      </div>\')">' . esc_attr(self::label( 'Available', __( 'Online, Available', 'ppv-live-
      webcams' ), $options )) . '</a>
10554
10555   <a class="' . ( $pstatus == 'paid' ? 'active' : '' ) . ' item"
      onclick="aurl' . esc_attr( $id ) . '=\'' . esc_url( $ajaxURLstat) . '&st=paid\';
      loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large
      loader\\\'>' . __( 'Updating listings, please wait', 'ppv-live-webcams' ) . '...
      </div>\')">' . esc_attr(self::label( 'Paid', __( 'All, Paid', 'ppv-live-webcams' ), $options )) . '</a>
10556   <a class="' . ( $pstatus == 'free' ? 'active' : '' ) . ' item"
      onclick="aurl' . esc_attr( $id ) . '=\'' . esc_url( $ajaxURLstat) . '&st=free\';
      loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large
      loader\\\'>' . __( 'Updating listings, please wait', 'ppv-live-webcams' ) . '...
      </div>\')">' . esc_attr(self::label( 'Free', __( 'All, Free', 'ppv-live-webcams' ), $options )) . '</a>
10557
10558   <a class="' . ( $pstatus == 'available_paid' ? 'active' : '' ) . ' item"
      onclick="aurl' . esc_attr( $id ) . '=\'' . esc_url( $ajaxURLstat) . '&st=available_paid\';
      loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large
      loader\\\'>' . __( 'Updating listings, please wait', 'ppv-live-webcams' ) . '...
      </div>\')">' . esc_attr(self::label( 'Available Paid', __( 'Available, Paid', 'ppv-live-
      webcams' ), $options )) . '</a>
10559   <a class="' . ( $pstatus == 'available_free' ? 'active' : '' ) . ' item"
      onclick="aurl' . esc_attr( $id ) . '=\'' . esc_url( $ajaxURLstat) . '&st=available_free\';
      loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large
      loader\\\'>' . __( 'Updating listings, please wait', 'ppv-live-webcams' ) . '...
      </div>\')">' . esc_attr(self::label( 'Available Free', __( 'Available, Free', 'ppv-live-
      webcams' ), $options )) . '</a>
10560
10561   <a class="' . ( $pstatus == 'private' ? 'active' : '' ) . ' item"
      onclick="aurl' . esc_attr( $id ) . '=\'' . esc_url( $ajaxURLstat) . '&st=private\';
      loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large
      loader\\\'>' . __( 'Updating listings, please wait', 'ppv-live-webcams' ) . '...
      </div>\')">' . esc_attr(self::label( 'In Private', __( 'Online, In Private', 'ppv-live-
      webcams' ), $options )) . '</a>
10562   <a class="' . ( $pstatus == 'offline' ? 'active' : '' ) . ' item"
      onclick="aurl' . esc_attr( $id ) . '=\'' . esc_url( $ajaxURLstat) . '&st=offline\';
      loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large
      loader\\\'>' . __( 'Updating listings, please wait', 'ppv-live-webcams' ) . '...
      </div>\')">' . esc_attr(self::label( 'Offline', __( 'Offline', 'ppv-live-webcams' ), $options )) . '</a>
10563
```

```
</div>

  </div>
</div>';
                        }
                        if ( $selectStatus && $options['lovense'] && $options['lovenseToy'] ) {
                                echo '
<div class="ui ' . esc_attr( $options['interfaceClass'] ) . ' accordion small vwFields">

  <div class="title">
    <i class="game icon"></i>
    ' . __( 'Lovense', 'ppv-live-webcams' ) . ' ' . ( $ptoy != '' ? '<i class="check icon small">
</i>' : '' ) . '
  </div>

  <div class="' . ( $ptoy != '' ? 'active' : '' ) . ' content">
  <div class="ui ' . esc_attr( $options['interfaceClass'] ) . ' vertical menu small">

  <a class="' . ( $ptoy == '' ? 'active' : '' ) . ' item"
onclick="aurl' . esc_attr( $id ) . '=\'' . esc_url( $ajaxURLtoy) . '&lt=\';
loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large
loader\\\'>' . __( 'Updating listings, please wait', 'ppv-live-webcams' ) . '...
</div>\')">' . __( 'Any', 'ppv-live-webcams' ) . '</a>

  <a class="' . ( $ptoy == '1' ? 'active' : '' ) . ' item"
onclick="aurl' . esc_attr( $id ) . '=\'' . esc_url( $ajaxURLtoy) . '&lt=1\';
loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large
loader\\\'>' . __( 'Updating listings, please wait', 'ppv-live-webcams' ) . '...
</div>\')">' . __( 'Available', 'ppv-live-webcams' ) . '</a>

  <a class="' . ( $ptoy == '0' ? 'active' : '' ) . ' item"
onclick="aurl' . esc_attr( $id ) . '=\'' . esc_url( $ajaxURLtoy) . '&lt=0\';
loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large
loader\\\'>' . __( 'Updating listings, please wait', 'ppv-live-webcams' ) . '...</div>\')">' . __( 'Not
Available', 'ppv-live-webcams' ) . '</a>
  </div>

  </div>
</div>';
                        }

                        if ( $selectOrder ) {

                                $optionsOrders = array(
                                        'default'    => __( 'Default Order', 'ppv-live-webcams' ),
                                        'viewers'    => __( 'Current Viewers', 'ppv-live-webcams' ),
                                        'edate'      => __( 'Broadcast Recently', 'ppv-live-webcams' ),
                                        'post_date'  => __( 'Created Recently', 'ppv-live-webcams' ),
                                        'maxViewers' => __( 'Maximum Viewers', 'ppv-live-webcams' ),
                                        'rand'       => __( 'Random', 'ppv-live-webcams' ),
                                );

                                if ( $options['rateStarReview'] ) {
                                        $optionsOrders['rateStarReview_voteCount'] = __( 'Votes', 'ppv-
live-webcams' );
                                        $optionsOrders['rateStarReview_rating']
 = __( 'Rating', 'ppv-live-webcams' );
                                        $optionsOrders['rateStarReview_ratingNumber'] = __( 'Ratings
Number', 'ppv-live-webcams' );
                                        $optionsOrders['rateStarReview_ratingPoints'] = __( 'Rate
Popularity', 'ppv-live-webcams' );
                                        if ( $category ) {

$optionsOrders[ 'rateStarReview_rating_category' . $category ]       = __( 'Rating', 'ppv-live-
webcams' ) . ' ' . __( 'in Category', 'ppv-live-webcams' );
```

```
Line
10610 $optionsOrders[ 'rateStarReview_ratingNumber_category' . $category ] = __( 'Ratings Number', 'ppv-live-webcams' ) . ' ' . __( 'in Category', 'ppv-live-webcams' );
10611 $optionsOrders[ 'rateStarReview_ratingPoints_category' . $category ] = __( 'Rate Popularity', 'ppv-live-webcams' ) . ' ' . __( 'in Category', 'ppv-live-webcams' );
10612                                         }
10613                                 }
10614
10615                                 echo '
10616 <div class="ui ' . esc_attr( $options['interfaceClass'] ) . ' accordion small vwFields">
10617
10618   <div class="title">
10619     <i class="sort icon"></i>
10620     ' . __( 'Order By', 'ppv-live-webcams' ) . ' ' . ( $order_by != 'default' ? '<i class="check icon small"></i>' : '' ) . '
10621   </div>
10622   <div class="' . ( $order_by != 'default' ? 'active' : '' ) . ' content">
10623   <div class="ui ' . esc_attr( $options['interfaceClass'] ) . ' vertical menu small">
10624   ';
10625
10626                                 foreach ( $optionsOrders as $key => $value ) {
10627                                         echo '  <a class="' . ( $order_by == $key ? 'active' : '' ) . ' item"
onclick="aurl' . esc_attr( $id ) . '=\'' . esc_url( $ajaxURLord ) . '&ob=' . esc_attr( $key ) . '\';
loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large
loader\\\'>' . __( 'Ordering Rooms', 'ppv-live-webcams' ) . '...</div>\')">' . esc_html( $value ) . '</a>
';
10628                                 }
10629
10630                                 echo '</div>
10631
10632   </div>
10633 </div>';
10634
10635                         }
10636
10637                         $customFilters = array();
10638
10639                         if ($selectFilters)
10640                         {
10641                         if (is_array($options['profileFields']))
10642                         foreach ( $options['profileFields'] as $field => $parameters ) {
10643                                 if ( isset($parameters['filter']) && ( isset($parameters['options']) || in_array($parameters['type'], ['location', 'language']) )  ) {
10644
10645                                         $fieldName  = sanitize_title( trim( $field ) );
10646                                         $fieldMeta  = 'vwf_' . $fieldName;
10647
10648                                         if ($parameters['type'] == 'location')
10649                                         {
10650                                                 //handle location fields
10651 foreach ( array('continent', 'country', 'region') as $locationField )
10652                                                 {
10653 $fieldMetaL = 'vwf_' . $fieldName . '_'. $locationField;
10654 if (isset( $_GET[ $fieldMetaL ])) $fieldValue = $_GET[ $fieldMetaL ] ?? array();
10655                                                         else $fieldValue = '';
10656
10657 if (is_array($fieldValue)) foreach ($fieldValue as $key => $value) $fieldValue[$key] = sanitize_text_field( $value );
10658                                                         else $fieldValue = false;
```

```
10659
10660                                                   if ( $fieldValue ) {
10661 $customFilters[ $fieldMetaL ] = $fieldValue;
10662                                                   }
10663                                           }
10664                                           continue;
10665                                   }
10666
10667                                   if ( !isset($_GET[ $fieldMeta ]) || is_array($_GET[ $fieldMeta ] ?? ''))
10668                                   {
10669                                           $fieldValue = $_GET[ $fieldMeta ] ?? array();
10670 foreach ($fieldValue as $key => $value) $fieldValue[$key] = sanitize_text_field( $value );
10671
10672                                           //remove duplicates
10673                                           $fieldValue = array_unique($fieldValue);
10674 }else $fieldValue = sanitize_text_field( $_GET[ $fieldMeta ] ?? '' );
10675
10676
10677                                   if ( $fieldValue ) {
10678                                           $customFilters[ $fieldMeta ] = $fieldValue;
10679                                   }
10680                           }
10681                   }
10682
10683                   if ( count( $customFilters ) ) {
10684                           $ajaxURLall = add_query_arg( $customFilters, $ajaxURLall );// update url query
10685                   }
10686
10687                   if (is_array($options['profileFields']))
10688                   foreach ( $options['profileFields'] as $field => $parameters ) {
10689                           if ( isset($parameters['filter']) && isset($parameters['options']) ) {
10690
10691                                   $fieldName = sanitize_title( trim( $field ) );
10692                                   $fieldMeta = 'vwf_' . $fieldName;
10693
10694                                   //if (in_array($parameters['type'], array('checkboxes', 'multiselect')))
10695                                   if ( !isset($_GET[ $fieldMeta ]) || is_array($_GET[ $fieldMeta ] ?? '') )
10696                                   {
10697                                           //get values from field passed as array in GET
10698                                           $fieldValue = $_GET[ $fieldMeta ] ?? array();
10699                                           //if (!is_array($fieldValue)) $fieldValue = array($_GET[ $fieldMeta ]);
10700 foreach ($fieldValue as $key => $value) $fieldValue[$key] = sanitize_text_field( $value );
10701                                           $emptyValue = array();
10702                                   }else
10703                                   {
10704 $fieldValue = sanitize_text_field( $_GET[ $fieldMeta ] ?? '' );
10705                                           $emptyValue = '';
10706                                   }
10707
10708 $parameters['options'] = stripslashes( $parameters['options'] ?? '' );
10709
10710                                   //if $parameters['options'] contains | get      $fieldOptions by split by | otherwise by /
```

```
10711 $fieldOptions = strpos($parameters['options'], '|') ? explode('|', $parameters['options']) : explode('/', $parameters['options']);
10712
10713
10714 $icon = isset($parameters['icon']) ? stripslashes( $parameters['icon'] ) : ( $options['profileFieldIcon'] ?? false );
10715
10716                                         //if (in_array($parameters['type'], array('checkboxes', 'multiselect')))
10717                                         if (is_array($fieldValue))
10718                                                 $active = !empty($fieldValue) ? 'active' : '';
10719                                         else $active = ( $fieldValue ? 'active' : '');
10720
10721                                         echo '
10722 <div class="ui ' . esc_attr( $options['interfaceClass'] ) . ' accordion small vwFields">
10723
10724   <div class="title">
10725     <i class="' . ( $icon ? esc_attr( $icon ) : 'dropdown icon' ) .'"></i>
10726     ' . esc_html( $field ) . ' ' . ( $active ? '<i class="check icon tiny"></i>' : '' ) . '
10727   </div>
10728   <div class="' . esc_attr( $active ) . ' content">
10729   <div class="ui ' . esc_attr( $options['interfaceClass'] ) . ' vertical menu small">
10730   ';
10731
10732                                         echo '  <a class="' . ( $active ? '' : 'active') . ' item" onclick="aurl' . esc_attr( $id ) . '=\'' . add_query_arg( $fieldMeta, $emptyValue, $ajaxURLall ) . '\'; loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large loader\\\'>' . __( 'Filtering Rooms', 'ppv-live-webcams' ) . '...</div>\')">' . __( 'Any', 'ppv-live-webcams' ) .' <i class="window close outline icon small"></i></a> ';
10733
10734                                         //echo json_encode($fieldValue);
10735
10736                                         foreach ( $fieldOptions as $fieldOption ) {
10737
10738                                                 $setValue = $fieldValue; //copy
10739
10740                                                 //if (in_array($parameters['type'], array('checkboxes', 'multiselect')))
10741                                                 if ( is_array($fieldValue) )
10742                                                 {
10743                                                         if (in_array($fieldOption, $setValue))
10744                                                         {
10745                                                                 //remove from array
10746 $setValue = array_diff($setValue, array($fieldOption));
10747                                                         }else
10748                                                         {
10749 $setValue[] = sanitize_text_field( $fieldOption );
10750                                                         }
10751                                                         $urlQuery = add_query_arg( $fieldMeta, $setValue, $ajaxURLall );
10752
10753 $active = in_array($fieldOption, $fieldValue) ? 'active' : '';
10754                                                 }else
10755                                                 {
10756                                                         $urlQuery = add_query_arg( $fieldMeta, sanitize_text_field( $fieldOption ), $ajaxURLall );
10757 $active = $fieldValue == $fieldOption ? 'active' : '';
10758                                                 }
10759
```

```
Line
10760                                                   echo '  <a class="' . esc_attr($active) . ' item"
onclick="aurl' . esc_attr( $id ) . '=\'' . esc_url( $urlQuery ) . '\';
loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large
loader\\\'>' . __( 'Filtering Rooms', 'ppv-live-webcams' ) . '...
</div>\')">' . esc_html( $fieldOption ) . ' '. ( $active ? '<i class="check icon tiny">
</i>' : '' ). '</a> ';
10761                                           }
10762
10763                                           echo '</div>
10764
10765   </div>
10766 </div>';
10767
10768                                   }
10769                           }
10770                   //end $selectFilters
10771                   }
10772
10773                           echo '
10774 <PRE style="display: none"><SCRIPT language="JavaScript">
10775 jQuery(document).ready(function()
10776 {
10777 jQuery(".ui.accordion.vwFields").accordion({exclusive:false});
10778 });
10779 </SCRIPT></PRE>
10780 ';
10781                           echo '</div><div class="vwRoomsContent">';
10782                   }
10783
10784                   //end menu
10785
10786                   // ! header option controls
10787 if (!$json)
10788 {
10789                   echo '<div class="ui ' . esc_attr( $options['interfaceClass'] ) . ' tiny equal width
form"><div class="inline fields">';
10790
10791                   if ( $selectCategory && ! $menu ) {
10792                           echo '<div
class="field">' . wp_dropdown_categories( 'echo=0&name=category' . esc_attr( $id ) . '&hide_empty=1&class
=ui+dropdown+fluid+v-select&show_option_all=' . esc_attr(self::label( 'All Categories', __( 'All
Categories', 'ppv-live-webcams' ), $options )) . '&selected=' . $category ) . '</div>';
10793                           echo '<script>var category' . esc_attr( $id ) . ' =
document.getElementById("category' . esc_attr( $id ) . '"); category' . esc_attr( $id ) . '.onchange =
function(){aurl' . esc_attr( $id ) . '=\'' . esc_url( $ajaxURLcat ) . '&cat=\'+ this.value;
loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large loader\\\'>Loading
category...</div>\')}
10794                           </script>';
10795                   }
10796
10797                   if ( $selectStatus && ! $menu ) {
10798                           echo '<div class="field"><select class="ui dropdown v-select fluid"
id="pstatus' . esc_attr( $id ) . '" name="pstatus' . esc_attr( $id ) . '"
onchange="aurl' . esc_attr( $id ) . '=\'' . esc_url( $ajaxURLstat) . '&st=\'+ this.value;
loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large
loader\\\'>' . __('Updating listings, please wait', 'ppv-live-webcams') . '...</div>\')">';
10799
10800                           echo '<option value="all"' . ( $pstatus == '' ? '
selected' : '' ) . '>' . esc_attr(self::label( 'All', __( 'All', 'ppv-live-
webcams' ), $options )) . '</option>';
10801
10802                           echo '<option value="online"' . ( $pstatus == 'online' ? '
selected' : '' ) . '>' . esc_attr(self::label( 'Online', __( 'Online', 'ppv-live-
webcams' ), $options )) . '</option>';
10803
```

```
10804                         echo '<option value="public"' . ( $pstatus == 'public' ? '
selected' : '' ) . '>' . esc_attr(self::label( 'Available', __( 'Online, Available', 'ppv-live-
webcams' ), $options )) . '</option>';
10805
10806                         echo '<option value="paid"' . ( $pstatus == 'paid' ? '
selected' : '' ) . '>' . esc_attr(self::label( 'Paid', __( 'All, Paid', 'ppv-live-
webcams' ), $options )) . '</option>';
10807
10808                         echo '<option value="free"' . ( $pstatus == 'free' ? '
selected' : '' ) . '>' . esc_attr(self::label( 'Free', __( 'All, Free', 'ppv-live-
webcams' ), $options )) . '</option>';
10809
10810                         echo '<option value="available_paid"' . ( $pstatus == 'available_paid' ? '
selected' : '' ) . '>' . esc_attr(self::label( 'Available Paid', __( 'Available, Paid', 'ppv-live-
webcams' ), $options )) . '</option>';
10811
10812                         echo '<option value="available_free"' . ( $pstatus == 'available_free' ? '
selected' : '' ) . '>' . esc_attr(self::label( 'Available Free', __( 'Available, Free', 'ppv-live-
webcams' ), $options )) . '</option>';
10813
10814                         echo '<option value="private"' . ( $pstatus == 'private' ? '
selected' : '' ) . '>' . esc_attr(self::label( 'In Private', __( 'Online, In Private', 'ppv-live-
webcams' ), $options )) . '</option>';
10815
10816                         echo '<option value="offline"' . ( $pstatus == 'offline' ? '
selected' : '' ) . '>' . esc_attr(self::label( 'Offline', __( 'Offline', 'ppv-live-
webcams' ), $options )) . '</option>';
10817
10818                         echo '</select></div>';
10819
10820                 }
10821
10822                 if ( $selectOrder && ! $menu ) {
10823
10824                         echo '<div class="field"><select class="ui dropdown v-select fluid"
id="order_by' . esc_attr( $id ) . '" name="order_by' . esc_attr( $id ) . '"
onchange="aurl' . esc_attr( $id ) . '=\'' . esc_url( $ajaxURLord ) . '&ob=\'+ this.value;
loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large loader\\\'>Ordering
webcams...</div>\')">';
10825                         echo '<option value="">' . __( 'Order By', 'ppv-live-webcams' ) . ':</option>';
10826
10827                         echo '<option value="default"' . ( $order_by == 'default' ? '
selected' : '' ) . '>' . __( 'Default Order', 'ppv-live-webcams' ) . '</option>';
10828
10829                         echo '<option value="viewers"' . ( $order_by == 'viewers' ? '
selected' : '' ) . '>' . __( 'Current Viewers', 'ppv-live-webcams' ) . '</option>';
10830
10831                         echo '<option value="edate"' . ( $order_by == 'edate' ? '
selected' : '' ) . '>' . __( 'Broadcast Recently', 'ppv-live-webcams' ) . '</option>';
10832                         echo '<option value="post_date"' . ( $order_by == 'post_date' ? '
selected' : '' ) . '>' . __( 'Created Recently', 'ppv-live-webcams' ) . '</option>';
10833
10834                         echo '<option value="maxViewers"' . ( $order_by == 'maxViewers' ? '
selected' : '' ) . '>' . __( 'Maximum Viewers', 'ppv-live-webcams' ) . '</option>';
10835
10836                         echo '<option value="rand"' . ( $order_by == 'rand' ? '
selected' : '' ) . '>' . __( 'Random', 'ppv-live-webcams' ) . '</option>';
10837
10838                         if ( $options['rateStarReview'] ) {
10839
10840                                 echo '<option
value="rateStarReview_voteCount"' . ( $order_by == 'rateStarReview_voteCount' ? '
selected' : '' ) . '>' . __( 'Votes', 'ppv-live-webcams' ) . '</option>';
10841
10842                                 echo '<option
value="rateStarReview_rating"' . ( $order_by == 'rateStarReview_rating' ? '
selected' : '' ) . '>' . __( 'Rating', 'ppv-live-webcams' ) . '</option>';
10843                                 echo '<option
value="rateStarReview_ratingNumber"' . ( $order_by == 'rateStarReview_ratingNumber' ? '
```

```
selected' : '' ) . '>' . __( 'Ratings Number', 'ppv-live-webcams' ) . '</option>';
10844                                   echo '<option
value="rateStarReview_ratingPoints"' . ( $order_by == 'rateStarReview_ratingPoints' ? '
selected' : '' ) . '>' . __( 'Rate Popularity', 'ppv-live-webcams' ) . '</option>';
10845
10846                                   if ( $category ) {
10847                                           echo '<option
value="rateStarReview_rating_category' . esc_attr( $category ) . '"' . ( $order_by == 'rateStarReview_rat
ing_category' . $category ? ' selected' : '' ) . '>' . __( 'Rating', 'ppv-live-webcams' ) . ' ' . __( 'in
Category', 'ppv-live-webcams' ) . '</option>';
10848                                           echo '<option
value="rateStarReview_ratingNumber_category' . esc_attr( $category ) . '"' . ( $order_by == 'rateStarRevi
ew_ratingNumber_category' . $category ? ' selected' : '' ) . '>' . __( 'Ratings Number', 'ppv-live-
webcams' ) . ' ' . __( 'in Category', 'ppv-live-webcams' ) . '</option>';
10849                                           echo '<option
value="rateStarReview_ratingPoints_category' . esc_attr( $category ) . '"' . ( $order_by == 'rateStarRevi
ew_ratingPoints_category' . $category ? ' selected' : '' ) . '>' . __( 'Rate Popularity', 'ppv-live-
webcams' ) . ' ' . __( 'in Category', 'ppv-live-webcams' ) . '</option>';
10850                                   }
10851                           }
10852
10853                           echo '</select></div>';
10854
10855                   }
10856
10857
10858                   if ($selectFilters) //special filter types with many options always show as dropdowns
10859                   {
10860
10861                           //todo: optimize by loading only continents
10862
10863                           //load location info
10864                           $locationsPath = $options['uploadsPath'] . '/_locations/';
10865                           $locationsPathDefaut = plugin_dir_path( __FILE__ ) . 'data/';
10866
10867                           $fileLocations = $locationsPath . 'locations.json';
10868                           $fileLanguages = $locationsPath . 'languages.json';
10869
10870
if ( ! file_exists( $fileLocations ) )$fileLocations = $locationsPathDefaut . 'locations.json';
10871
if ( ! file_exists( $fileLanguages ) )$fileLanguages = $locationsPathDefaut . 'languages.json';
10872
10873                           $languages = array('en'=> 'English');
10874                           if ($options['languageFilters'] ?? false)
10875                           {
10876
if ( file_exists( $fileLanguages ) ) $languages = json_decode( file_get_contents( $fileLanguages ), true
);
10877                           if ( ! is_array( $languages ) ) $languages = array('en'=> 'English');
10878                           }
10879
10880                           $locations = array();
10881                           $continents = array();
10882                           $countries = array();
10883                           $regions = array();
10884
10885                           if ($options['locationFilters'] ?? false)
10886                           {
10887
if ( file_exists( $fileLocations ) ) $locations = json_decode( file_get_contents( $fileLocations ), true
);
10888                           if ( ! is_array( $locations ) ) $locations = array();
10889
10890                           foreach ( $locations as $continent_code => $continent )
```

```
10891                         {
10892                                 $continents[ $continent_code ] = $continent['continent_name'];
10893                                 foreach ( $continent['countries'] as $country_code => $country )
10894                                 {
10895                                         $countries[ $country_code ] = $country['country_name'];
10896
10897                                         if ($options['locationFilters'] != 2)
10898                                         if ( isset( $country['regions'] ) ) {
10899 foreach ( $country['regions'] as $region_code => $region ) {
10900                                                 $regions[ $country_code ]
[ $region_code ] = $region['region_name'];
10901                                         }
10902                                         }
10903                                 }
10904                         }
10905
10906                         }
10907
10908                                 if (is_array($options['profileFields']))
10909                                 foreach ( $options['profileFields'] as $field => $parameters ) {
10910 if ( isset($parameters['filter']) && (in_array($parameters['type'], ['location', 'language']) ) )
10911                                         {
10912                                                 $fieldName  = sanitize_title( trim( $field ) );
10913                                                 $fieldMeta  = 'vwf_' . $fieldName;
10914
10915                                                 if ( is_array($_GET[ $fieldMeta ] ?? '' ) )
10916                                                 {
10917                                                         //get values from field passed as array in GET
10918                                                         $fieldValue = $_GET[ $fieldMeta ] ?? array();
10919 foreach ($fieldValue as $key => $value) $fieldValue[$key] = sanitize_text_field( $value );
10920
10921                                                         //remove duplicates
10922                                                         $fieldValue = array_unique($fieldValue);
10923
10924                                                 }else
10925                                                 {
10926 $fieldValue = array( sanitize_text_field( $_GET[ $fieldMeta ] ?? '' ) );
10927                                                 }
10928
10929                                                 if (in_array('-', $fieldValue)) $fieldValue = array('');
10930
10931 $icon = isset($parameters['icon']) ? stripslashes( $parameters['icon'] ) : ( $options['profileFieldIcon']
 ?? false );
10932
10933                                                 switch ($parameters['type'])
10934                                                 {
10935                                                         case 'language':
10936
10937                                                                 if ($options['languageFilters'] ?? false)
10938                                                                 {
10939                                                                 echo '<div class="field"><select multiple
class="ui dropdown multiple search v-select fluid" id="' . esc_attr($fieldMeta) . esc_attr($id) . '"
name="' . esc_attr($fieldMeta) . esc_attr($id) . '[]"
onchange="aurl' . esc_attr($id) . '=\'' . esc_url( remove_query_arg($fieldMeta, $ajaxURLall) ) . '&\' +
jQuery.param({\'' . esc_js($fieldMeta . '[]') . '\':
jQuery(\'#' . esc_js($fieldMeta) . esc_attr($id) . '\').val() }); loadWebcams' . esc_attr($id) . '(\'<div
class=\\\'ui active inline text large loader\\\'>' . __('Updating listings, please wait', 'ppv-live-
webcams') . '...</div>\')">';
10940
```

```
Line
10941                                                                   echo '<option value="">' . esc_html($field) . '</div>';
10942                                                                   echo '<option value="-">' . esc_html(self::label( 'Any', __( 'Any', 'ppv-live-webcams' ), $options )) . '</option>';
10943
10944                                                                   foreach ( $languages as $key => $language ) {
10945                                                                         $selected = in_array($language , $fieldValue) ? ' selected' : '';
10946
10947                                                                          echo '<option value="' . esc_attr(htmlspecialchars( stripslashes( $language ) )) . '" ' . esc_attr($selected) . '>' . esc_html(htmlspecialchars( stripslashes( $language ) )) . '</option>';
10948                                                                   }
10949
10950                                                                   echo '</select></div>';
10951                                                                   } else echo '<!-- ' . esc_attr($fieldMeta) . esc_attr($id) . ':  Language filters disabled -->';
10952
10953                                                                   break;
10954
10955                                                                   case 'location':
10956
10957                                                                   if ($options['locationFilters'] ?? false)
10958                                                                                                                                           {
10959
10960                                                                                                                                           $fieldValue = array();
10961                                                                   foreach ([ 'continent', 'country', 'region' ] as $locationField)
10962                                                                                                                                           {
10963                                                                   $fieldMetaL = 'vwf_' . $fieldName . '_'. $locationField;
10964                                                                   if (isset( $_GET[ $fieldMetaL ])) $fieldValue[$locationField] = $_GET[ $fieldMetaL ] ?? array();
10965                                                                   else $fieldValue[$locationField] = [];
10966                                                                                                                                           }
10967
10968                                                                                                                                                                                                   // Continent Dropdown
10969                                                                                                                                                                                                   echo '<div class="field"><select multiple class="ui dropdown multiple search v-select fluid" id="' . esc_attr($fieldMeta) . esc_attr($id) . '_continent" name="' . esc_attr($fieldMeta) . esc_attr($id) . '_continent[]" onchange="aurl' . esc_attr($id) . '=\'' . esc_url( remove_query_arg($fieldMeta . '_continent', $ajaxURLall) ) . '&\' + jQuery.param({\'' . esc_js($fieldMeta . '_continent[]') . '\': jQuery(\'#' . esc_js($fieldMeta) . esc_attr($id) . '_continent\').val() }); loadWebcams' . esc_attr($id) . '(\'<div class=\\\'ui active inline text large loader\\\'>' . __('Updating listings, please wait', 'ppv-live-webcams') . '...</div>\')">';
10970
10971                                                                                                                                                                                                   echo '<option value="">Continent ' . esc_html($field) . '</option>';
10972                                                                                                                                                                                                   echo '<option value="-">' . esc_html(self::label( 'Any', __( 'Any', 'ppv-live-webcams' ), $options )) . '</option>';
10973
10974                                                                   foreach ( $continents as $key => $continent ) {
10975                                                                   $selected = in_array($continent , $fieldValue['continent']) ? ' selected' : '';
10976                                                                                                                                                                                                
```

```
10982                                                
```

```
11019                                                   }
11020
11021                                           }
11022                                   }
11023
11024                   }
11025
11026
11027                   if ($selectFilters && ! $menu ) //filters with options
11028                   {
11029
11030                           /*
11031                   // custom profile filters
11032                   if (is_array($options['profileFields']))
11033                   foreach ( $options['profileFields'] as $field => $parameters ) {
11034                           if ( isset($parameters['filter']) && isset($parameters['options']) ) {
11035
11036                                   $fieldName  = sanitize_title( trim( $field ) );
11037                                   $fieldMeta  = 'vwf_' . $fieldName;
11038
11039                                   if (is_array($_GET[ $fieldMeta ] ?? ''))
11040                                   {
11041                                           $fieldValue = $_GET[ $fieldMeta ] ?? array();
11042                                           foreach ($fieldValue as $key => $value) $fieldValue[$key] =
sanitize_text_field( $value );
11043                                   }else $fieldValue = sanitize_text_field( $_GET[ $fieldMeta ] ?? '' );
11044
11045                                   if ( $fieldValue ) {
11046                                           $customFilters[ $fieldMeta ] = $fieldValue;
11047                                   }
11048                           }
11049                   }
11050                                   if ( count( $customFilters ) ) {
11051                           $ajaxURLall = add_query_arg( $customFilters, $ajaxURLall );// update url query
11052                   }
11053                   */
11054
11055
11056
11057                   if (is_array($options['profileFields']))
11058                   foreach ( $options['profileFields'] as $field => $parameters ) {
11059                           if ( isset($parameters['filter']) && isset($parameters['options']) ) {
11060
11061                                   $fieldName = sanitize_title( trim( $field ) );
11062                                   $fieldMeta = 'vwf_' . $fieldName;
11063
11064                                   if ( is_array($_GET[ $fieldMeta ] ?? '' ) )
11065                                   {
11066                                           //get values from field passed as array in GET
11067                                           $fieldValue = $_GET[ $fieldMeta ] ?? array();
11068
foreach ($fieldValue as $key => $value) $fieldValue[$key] = sanitize_text_field( $value );
11069
11070                                           //remove duplicates
11071                                           $fieldValue = array_unique($fieldValue);
11072
11073                                   }else
11074                                   {
11075
$fieldValue = array( sanitize_text_field( $_GET[ $fieldMeta ] ?? '' ) );
11076                                   }
11077
```

```
11078                                   if (in_array('-', $fieldValue)) $fieldValue = array('');
11079
11080                                   $parameters['options'] = stripslashes( $parameters['options'] ?? '' );
11081
11082                                   //if $parameters['options'] contains | get      $fieldOptions by split by | otherwise by /
11083   $fieldOptions = strpos($parameters['options'], '|') ? explode('|', $parameters['options']) : explode('/', $parameters['options']);
11084
11085   $icon = isset($parameters['icon']) ? stripslashes( $parameters['icon'] ) : ( $options['profileFieldIcon'] ?? false );
11086
11087                                   //if (in_array($parameters['type'], array('checkboxes', 'multiselect')))
11088                                   if (is_array($fieldValue))
11089                                           $active = !empty($fieldValue) ? 'active' : '';
11090                                   else $active = ( $fieldValue ? 'active' : '');
11091
11092   //echo json_encode($fieldValue);
11093
11094   echo '<div class="field"><select multiple class="ui dropdown multiple search v-select fluid" id="' . esc_attr($fieldMeta) . esc_attr($id) . '" name="' . esc_attr($fieldMeta) . esc_attr($id) . '[]" onchange="aurl' . esc_attr($id) . '=\'' . esc_url( remove_query_arg($fieldMeta, $ajaxURLall) ) . '&\' + jQuery.param({\'' . esc_js($fieldMeta . '[]') . '\': jQuery(\'#' . esc_js($fieldMeta) . esc_attr($id) . '\').val() }); loadWebcams' . esc_attr($id) . '(\'<div class=\\\'ui active inline text large loader\\\'>' . __('Updating listings, please wait', 'ppv-live-webcams') . '...</div>\')">';
11095
11096   echo '<option value="">' . esc_attr($field). '</option>';
11097
11098   echo '<option value="-">' . esc_html(self::label( 'Any', __( 'Any', 'ppv-live-webcams' ), $options )) . '</option>';
11099
11100
11101                                   foreach ( $fieldOptions as $fieldOption ) {
11102
11103                                           $selected = in_array($fieldOption, $fieldValue) ? ' selected' : '';
11104                                           echo '<option value="' . esc_attr($fieldOption) . '"' . esc_attr($selected) . '>' . esc_html( $fieldOption ) . '</option>';
11105                                   }
11106
11107                                   echo '</select></div>';
11108
11109                           }
11110                   }
11111           //end $selectFilters
11112           }
11113
11114           //persistent filters
11115           if ($options['filtersSave'] ?? false)
11116           if (is_user_logged_in())
11117           {
11118                   update_user_meta( get_current_user_id(), 'vwf_filters_url', $ajaxURLall  );
11119           }
11120
11121                   if ( $selectTags || $selectName ) {
11122                           echo '<div class="field"></div>'; // separator
11123
11124                           if ( $selectTags ) {
11125                                   echo '<div class="field" data-tooltip="Tags, Comma Separated"><div class="ui left icon input fluid"><i class="tags icon"></i><INPUT class="videowhisperInput" type="text" size="12" name="tags" id="tags" placeholder="' . __( 'Tags', 'ppv-live-webcams' ) . '" value="' . esc_attr( htmlspecialchars( $tags ) ) . '">
```

```
11126                                   </div></div>';
11127                   }
11128
11129                   if ( $selectName ) {
11130                           echo '<div class="field"><div class="ui left corner labeled input fluid">
<INPUT class="videowhisperInput" type="text" size="12" name="name" id="name"
placeholder="' . __( 'Name', 'ppv-live-webcams' ) . '"
value="' . esc_attr( htmlspecialchars( $name ) ) . '">
11131   <div class="ui left corner label">
11132     <i class="asterisk icon"></i>
11133   </div>
11134                                   </div></div>';
11135                   }
11136
11137                   // search button
11138                   echo '<div class="field"><button class="ui icon button" data-tooltip="Search"
type="button" name="button" id="submit" value="' . __( 'Search', 'ppv-live-webcams' ) . '"
onclick="aurl' . esc_attr( $id ) . '=\'' . esc_url( $ajaxURLstat) . '&tags=\' +
document.getElementById(\'tags\').value +\'&name=\' + document.getElementById(\'name\').value;
loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large loader\\\'>Searching
webcams...</div>\')"><i class="search icon"></i></button>
11139                    <button style="float:right" data-tooltip="Reload" class="ui icon button"
type="button" name="reload" id="reload" value="' . __( 'Reload', 'ppv-live-webcams' ) . '"
onclick="aurl' . esc_attr( $id ) . '=\'' . esc_url( $ajaxURLall ) . '\';
loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large loader\\\'>Reloading
Rooms List...</div>\')"><i class="sync icon"></i></button>
11140                    </div>';
11141               }
11142
11143               // reload button
11144               // if ($selectCategory || $selectOrder || $selectTags || $selectName) echo '<div
class="field" ></div>';
11145
11146               echo '</div></div>';
11147
11148 //!json
11149 }
11150
11151               $year = time() - 31536000;
11152
11153               // ! meta query
11154               $meta_query = array(
11155                       'relation'    => 'AND',
11156                       'vw_featured' => array(
11157                               'key'     => 'vw_featured',
11158                               'compare' => 'EXISTS',
11159                       ),
11160                       'edate'       => array(
11161                               'key'     => 'edate',
11162                               'compare' => 'EXISTS',
11163                       ),
11164               );
11165
11166               // has thumb
11167               if ($options['listingsThumbsOnly'])
11168               $meta_query['hasThumb'] = array(
11169                       'relation' => 'OR',
11170                       array(
11171                               'key'   => 'hasSnapshot',
11172                               'value' => '1',
11173                       ),
11174                       /*
11175                       array(
11176                               'key'   => 'hasPicture',
```

```
11177                                   'value' => '1',
11178                           ),*/
11179                   );
11180
11181                   if ($options['sightengine'] ?? false) $meta_query['suspended'] = array(
11182                           'key'     => 'vwSuspended',
11183                           'compare' => 'NOT EXISTS',
11184                   );
11185
11186                   // hide private rooms
11187                   $meta_query['room_private'] = array(
11188                           'relation' => 'OR',
11189                           array(
11190                                   'key'   => 'room_private',
11191                                   'value' => 'false',
11192                           ),
11193                           array(
11194                                   'key'   => 'room_private',
11195                                   'value' => '',
11196                           ),
11197                           array(
11198                                   'key'     => 'room_private',
11199                                   'compare' => 'NOT EXISTS',
11200                           ),
11201                   );
11202
11203
11204                   // location filter: may impact query performance even timeout on databases with many
metas
11205
11206                   if ( !$options['listingsDisableLocation'] && !$isModerator)
11207                   {
11208                   $clientLocation = self::detectLocation( 'all' ); // array
11209
11210                   if ( $clientLocation ) {
11211                           if ( is_array( $clientLocation ) ) {
11212                                   if ( ! empty( $clientLocation ) ) {
11213                                           $meta_query['location'] = array(
11214                                                   'relation' => 'OR',
11215                                                   array(
11216                                                           'key'     => 'vw_banCountries',
11217                                                           'value'   => $clientLocation,
11218                                                           'compare' => 'NOT IN',
11219                                                   ),
11220                                                   array(
11221                                                           'key'     => 'vw_banCountries',
11222                                                           'compare' => 'NOT EXISTS',
11223                                                   ),
11224                                           );
11225
11226                                   }
11227                           }
11228                   }
11229
11230                   }
11231
11232                           // ! query args (starter pack)
11233                           $args = array(
11234                                   'post_type'      => $options['custom_post'],
11235                                   'post_status'    => 'publish',
11236                                   'posts_per_page' => $perPage,
```

```
11237                   'offset'         => $offset,
11238                   'order'          => 'DESC',
11239                   'meta_query'     => $meta_query,
11240                   'cache_results'  => false, //own implementation with listingsCache
11241           );
11242
11243           if ( ! $pstatus ) {
11244                   $pstatus = '';
11245           }
11246
11247           if ( $studio_id ) {
11248                   $args['meta_query'][] = array(
11249                           'key'   => 'studioID',
11250                           'value' => $studio_id,
11251                   );
11252           }
11253
11254           if ( $author_id ) {
11255                   $args['author'] = $author_id;
11256           }
11257
11258           if ($ptoy != '')        $args['meta_query']['lovenseToy']   = array(
11259                   'key'   => 'lovenseToy',
11260                   'value' => $ptoy,
11261           );
11262
11263           if ($pstatus) switch ( $pstatus ) {
11264                   case 'free':
11265                           $args['meta_query']['public']   = array(
11266                                   'key'   => 'privateShow',
11267                                   'value' => '0',
11268                           );
11269                           $args['meta_query']['groupCPM'] = array(
11270                                   'key'   => 'groupCPM',
11271                                   'value' => '0',
11272                           );
11273                           break;
11274                   case 'paid':
11275                           $args['meta_query']['public']   = array(
11276                                   'key'   => 'privateShow',
11277                                   'value' => '0',
11278                           );
11279                           $args['meta_query']['groupCPM'] = array(
11280                                   'key'     => 'groupCPM',
11281                                   'value'   => '0',
11282                                   'compare' => '>',
11283                           );
11284                           break;
11285                   case 'available_free':
11286                           $args['meta_query']['public']   = array(
11287                                   'key'   => 'privateShow',
11288                                   'value' => '0',
11289                           );
11290                           $args['meta_query']['online']   = array(
11291                                   'key'     => 'edate',
11292                                   'value'   => time() - 40,
11293                                   'compare' => '>',
11294                           );
11295                           $args['meta_query']['groupCPM'] = array(
11296                                   'key'   => 'groupCPM',
11297                                   'value' => '0',
```

```
11298                                   );
11299                                   break;
11300                           case 'available_paid':
11301                                   $args['meta_query']['public']   = array(
11302                                           'key'   => 'privateShow',
11303                                           'value' => '0',
11304                                   );
11305                                   $args['meta_query']['online']   = array(
11306                                           'key'     => 'edate',
11307                                           'value'   => time() - 40,
11308                                           'compare' => '>',
11309                                   );
11310                                   $args['meta_query']['groupCPM'] = array(
11311                                           'key'     => 'groupCPM',
11312                                           'value'   => '0',
11313                                           'compare' => '>',
11314                                   );
11315                                   break;
11316
11317                           case 'private':
11318                                   $args['meta_query']['private'] = array(
11319                                           'key'   => 'privateShow',
11320                                           'value' => '1',
11321                                   );
11322                                   $args['meta_query']['online']  = array(
11323                                           'key'     => 'edate',
11324                                           'value'   => time() - 40,
11325                                           'compare' => '>',
11326                                   );
11327                                   break;
11328
11329                           case 'online':
11330                                   $args['meta_query']['online'] = array(
11331                                           'key'     => 'edate',
11332                                           'value'   => time() - 40,
11333                                           'compare' => '>',
11334                                   );
11335                                   break;
11336
11337                           case 'public':
11338                                   $args['meta_query']['public'] = array(
11339                                           'key'   => 'privateShow',
11340                                           'value' => '0',
11341                                   );
11342                                   $args['meta_query']['online'] = array(
11343                                           'key'     => 'edate',
11344                                           'value'   => time() - 40,
11345                                           'compare' => '>',
11346                                   );
11347                                   break;
11348
11349                           case 'offline':
11350                                   $args['meta_query']['offline'] = array(
11351                                           'key'     => 'edate',
11352                                           'value'   => time() - 40,
11353                                           'compare' => '<',
11354                                   );
11355                                   break;
11356                   }
11357
11358                   if ($order_by) switch ( $order_by ) {
```

```
11359                                   case 'default':
11360                                           $args['orderby'] = array(
11361                                                   'vw_featured' => 'DESC',
11362                                                   'edate'       => 'DESC',
11363                                           );
11364                                           break;
11365
11366                                   case 'post_date':
11367                                           $args['orderby'] = 'post_date';
11368                                           break;
11369
11370                                   case 'rand':
11371                                           $args['orderby'] = 'rand';
11372                                           break;
11373
11374                                   default:
11375                                           $args['orderby']  = 'meta_value_num';
11376                                           $args['meta_key'] = $order_by;
11377                                           break;
11378                           }
11379
11380                           if ( $category ) {
11381                                   $args['category'] = $category;
11382                           }
11383
11384                           if ( $tags ) {
11385                                   $tagList = explode( ',', $tags );
11386                                   if ( is_array($tagList) ) foreach ( $tagList as $key => $value )
11387                                   {
11388                                           $tagList[ $key ] = trim( $tagList[ $key ] );
11389                                   }
11390
11391                                   $args['tax_query'] = array(
11392                                           array(
11393                                                   'taxonomy' => 'post_tag',
11394                                                   'field'    => 'slug',
11395                                                   'operator' => 'AND',
11396                                                   'terms'    => $tagList,
11397                                           ),
11398                                   );
11399                           }
11400
11401                           if ( $name ) {
11402                                   $args['s'] = $name;
11403                           }
11404
11405                           // custom profile filters - get postz
11406                           if (is_array($options['profileFields']))
11407                           foreach ( $options['profileFields'] as $field => $parameters ) {
11408                                   if ( isset($parameters['filter']) ) {
11409
11410                                           $fieldName  = sanitize_title( trim( $field ) );
11411                                           $fieldMeta  = 'vwf_' . $fieldName;
11412
11413                                           //skip if not set or location
11414 if (!isset($_GET[ $fieldMeta ]) && $parameters['type'] != 'location') continue;
11415
11416                                           if ( $parameters['type'] == 'location' )
11417                                           {
11418
```

```
Line
11419                                                   $locations = 0;
11420                                                   $argsL = array();
11421
foreach (['continent', 'country', 'region'] as $location)
11422                                                   {
11423
$fieldMetaL = 'vwf_' . $fieldName . '_' . $location;
11424
if (isset($_GET[ $fieldMetaL ])) $fieldValue = $_GET[ $fieldMetaL ] ?? '';
11425                                                           else $fieldValue = '';
11426
11427                                                           if ($fieldValue)
11428                                                           {
11429                                                                   $argsL[$location] = array(
11430                                                                           'key'   => $fieldMetaL,
11431                                                                           'value' => $fieldValue,
11432                                                                   );
11433                                                           }
11434                                                           $locations++;
11435                                                   }
11436
11437                                                   if ($locations)
11438                                                   {
11439                                                           $args['meta_query'][ $fieldMeta ] = $argsL;
11440                                                           $args['meta_query'][ $fieldMeta ]
[ 'relation' ] = 'OR';
11441                                                   }
11442                                                   continue;
11443                                           }
11444
11445                                           //regular fields
11446                                           if ( is_array( $_GET[ $fieldMeta ] ?? '') )
11447                                           {
11448                                                   $fieldValue = $_GET[ $fieldMeta ] ?? array();
11449
foreach ($fieldValue as $key => $value) $fieldValue[$key] = sanitize_text_field( $value );
11450                                           }else
11451                                           {
11452
$fieldValue = sanitize_text_field( $_GET[ $fieldMeta ] ?? '' );
11453                                           }
11454
11455                                           if ( $fieldValue ) {
11456                                                   if ( is_array($fieldValue) )
11457                                                   {
11458                                                           if (in_array('-
', $fieldValue)) $fieldValue = array();
11459
11460                                                           //add a wp meta query to match any of the values
considering both $fieldValue and meta are arrays
11461                                                           if (!empty($fieldValue))
11462                                                           {
11463                                                                   $args['meta_query']
[ $fieldMeta ] = ['relation' => 'OR' ];
11464                                                                   $fieldValue = array_unique($fieldValue);
11465
11466
if (in_array($parameters['type'], array('checkboxes', 'multiselect', 'language')))
11467
foreach ($fieldValue as $key => $value) $args['meta_query'][ $fieldMeta ][] =
11468                                                                                   [
11469                                                                                           'key'     => $fieldMeta,
11470                                                                                           // field is stored as a
serialized array (e.g., a:4:{i:0;s:7:"English";i:1;s:7:"Spanish";i:2;s:7:"Italian";i:3;s:6:"French";}) in
the WP database
```

```
11471                                                                   'value'   => '"' . $value . '"',
11472                                                                   'compare' => 'LIKE',
11473                                                           ];
11474                                                           else
11475 foreach ($fieldValue as $key => $value) $args['meta_query'][ $fieldMeta ][] =
11476                                                           [
11477                                                                   'key'   => $fieldMeta,
11478                                                                   'value' => $value,
11479                                                           ];
11480
11481                                                   }
11482
11483                                           } else {
11484                                                   $args['meta_query'][ $fieldMeta ] = array(
11485                                                           'key'   => $fieldMeta,
11486                                                           'value' => $fieldValue,
11487                                                   );
11488                                           }
11489                                   }
11490                           }
11491                   }
11492
11493
11494                   //db query with cache
11495                   if ($options['listingsCache'] ?? false)
11496                   {
11497                           //https://developer.wordpress.org/reference/functions/wp_cache_set/
11498                           $listingsCache = intval( $options['listingsCache'] );
11499                           if ($listingsCache < 5) $listingsCache = 5;
11500
11501                           //force check in case cache is not properly purged
11502                           $listingsUpdate = false;
11503 if ( self::timeTo( 'listingsCache', $listingsCache, $options ) ) $listingsUpdate = true;
11504
11505                           $cacheKey = json_encode($args);
11506                           if ($listingsUpdate ||
  !$postslist = wp_cache_get($cacheKey, 'VideoWhisper') ) //generate new
11507                           {
11508                           $postslist = get_posts( $args );
11509
wp_cache_set($cacheKey, $postslist, 'VideoWhisper', $listingsCache); //short to list live rooms
11510                           echo '<!--VideoWhisper Room List: Saved Cache
'. esc_html($listingsCache) .'s -->';
11511                           } else echo '<!--VideoWhisper Room List: Cache Result
'. esc_html($listingsCache) .'s -->';
11512
11513                   } else
11514                   {
11515                           echo '<!--VideoWhisper Room List: Cache Disabled-->';
11516                           $postslist = get_posts( $args );
11517                   }
11518
11519                   if ($options['debugMode']) echo '<!--VideoWhisper Room List Query:
'. json_encode($args) .' -->';
11520
11521                   if (!$json) echo '<!--VideoWhisper Room List Start--><div
id="videowhisperListings'. esc_attr($id) .'" class="videowhisper' . ucwords($layout) . '">';
11522
11523                   // ! list cams
11524                   $listingsCount = 0;
11525                   if (is_array($postslist)) $listingsCount = count( $postslist );
```

```
11526
11527                   if ( $listingsCount > 0 ) {
11528
11529                   if (!$json)
11530                   {
11531                           if ( $layout == 'horizontal' ) {
11532   $listingTemplate = $listingTemplate2 = $listingTemplate1 = html_entity_decode( stripslashes( $options['listingTemplateHorizontal'] ) );
11533                           }elseif ( $layout == 'list' ) {
11534   $listingTemplate = $listingTemplate2 = $listingTemplate1 = html_entity_decode( stripslashes( $options['listingTemplateList'] ) );
11535                           } else {
11536   $listingTemplate1 = html_entity_decode( stripslashes( $options['listingTemplate'] ) );
11537   $listingTemplate2 = html_entity_decode( stripslashes( $options['listingTemplate2'] ) );
11538                                   $listingTemplate  = $listingTemplate2;
11539                           }
11540
11541   $templateSearch = array( '#name#', '#age#', '#clientCPM#', '#roomBrief#', '#roomTags#', '#url#', '#snapshot#', '#thumbWidth#', '#thumbHeight#', '#banLink#', '#groupMode#', '#groupCPM#', '#performers#', '#currency#', '#preview#', '#enter#', '#paidSessionsPrivate#', '#paidSessionsGroup#', '#rating#', '#performerStatus#', '#roomCategory#', '#featuredReview#', '#roomDescription#', '#featured#', '#buttonChat#', '#buttonCall#', '#buttonMessage#', '#vote#', '#icons#' );
11542                           // $templateReplace  = array($name, $age, $clientCPM, $roomBrief, $roomTags, $url, $snapshot, $thumbWidth, $thumbHeight, $banLink);
11543                   }
11544
11545                   $showBig = 0;
11546
11547                           $k = 0;
11548                           foreach ( $postslist as $item ) {
11549                                   //per webcam post item
11550
11551                                   $ritem = [];
11552
11553                                   if (!$json)
11554                                   {
11555                                    if ( $layout == 'grid' ) {
11556                                           if ( $perRow ) {
11557                                                   if ( $k ) {
11558                                                           if ( $k % $perRow == 0 ) {
11559                                                                   echo '<br>';
11560                                                           }
11561                                                   }
11562                                           }
11563                                   }
11564
11565                                                   $k++;
11566
11567                                           $listingTemplate = $listingTemplate1;
11568
11569                                   $showBig = 0;
11570
11571                                   if ( $options['listingBig'] ) {
11572                                           if ( $options['listingBig'] == 1 && $k == 1 ) {
11573                                                   $listingTemplate = $showBig = 1;
11574                                           }
11575                                           if ( $options['listingBig'] > 1 ) {
11576                                                   if ( $k % $options['listingBig'] == 1 ) {
11577                                                           $showBig = 1;
```

```
11578                                                   }
11579                                           }
11580
11581                                           if ( $showBig ) {
11582                                                   $listingTemplate = $listingTemplate2;
11583                                           }
11584                                   }
11585
11586                           }
11587                                   // else $listingTemplate = $listingTemplate1;
11588
11589                                   $k++;
11590
11591                                   $name = sanitize_file_name( $item->post_title );
11592
11593                                   $banLink = '';
11594
11595                                   if ( $ban && !$json) {
11596                                           $banLink = '<a class = "button" href="admin.php?page=live-webcams&ban=' . urlencode( $name ) . '">' . __( 'Ban This Webcam', 'ppv-live-webcams' ) . '</a><br>';
11597                                   }
11598
11599                                   $edate = intval( get_post_meta( $item->ID, 'edate', true ) );
11600                                   $age   = self::format_age( time() - $edate );
11601
11602                                   $privateShow = get_post_meta( $item->ID, 'privateShow', true );
11603
11604                                   // performer status
11605                                   if ( time() - $edate > 40 ) {
11606                                           $performerStatus = 'offline';
11607                                   } elseif ( $privateShow ) {
11608                                           $performerStatus = 'private';
11609                                   } else {
11610                                           $performerStatus = 'public';
11611                                   }
11612
11613
11614
11615                                   $clientCPM = self::clientCPM( $name, $options, $item->ID );
11616
11617                                   // ip camera or playlist : update snapshot when listing
11618                                   if ( get_post_meta( $item->ID, 'vw_ipCamera', true ) || ( get_post_meta( $item->ID, 'vw_playlistActive', true ) && $options['playlists'] ) ) {
11619                                           self::streamSnapshot( $name, true, $item->ID );
11620                                           // echo 'Updating IP Cam Snapshot: ' . $stream;
11621                                   }
11622
11623                                   switch ( $options['webcamLink'] ) {
11624                                           case '':
11625                                           case 'room':
11626                                                   $url = self::roomURL( $name );
11627                                                   break;
11628
11629                                           case 'custom':
11630                                                   $url = get_post_meta( $item->ID, 'customRoomLink', true );
11631                                                   if ( ! $url ) {
11632                                                           $url = $options['webcamLinkDefault'] . $name;
11633                                                           update_post_meta( $item->ID, 'customRoomLink', $url );
11634                                                   }
```

```
11635                                                   break;
11636
11637                                           case 'auto':
11638                                                   $url = get_post_meta( $item->ID, 'customRoomLink', true );
11639                                                   if ( ! $url ) {
11640                                                           $url = self::roomURL( $name );
11641                                                   }
11642                                                   break;
11643
11644                                   }
11645
11646
11647
11648                                   $thumbWidth  = $options['thumbWidth'];
11649                                   $thumbHeight = $options['thumbHeight'];
11650
11651                                   $snapshot = self::webcamThumbSrc( $item->ID, $name, $options, $age, true );
11652
11653                                   $previewCode = self::webcamThumbCode( $item->ID, $name, $options, $snapshot, $optionsVSV, $isMobile, $showBig, $previewMuted, $age );
11654
11655                                   $roomBrief = get_post_meta( $item->ID, 'vw_roomBrief', true );
11656
11657                                   $tags     = wp_get_post_tags( $item->ID, array( 'fields' => 'names' ) );
11658                                   $roomTags = '';
11659                                   if ( ! empty( $tags ) ) {
11660                                           if ( ! is_wp_error( $tags ) ) {
11661                                                   foreach ( $tags as $tag ) {
11662                                                           $roomTags .= ( $roomTags ? ', ' : '' ) . $tag;
11663                                                   }
11664                                           }
11665                                   }
11666
11667                                           $roomLabel = get_post_meta( $item->ID, 'vw_roomLabel', true );
11668                                   if ( ! $roomLabel ) {
11669                                           $roomLabel = $name;
11670                                   }
11671
11672                                   $roomCategory = '';
11673                                   $cats         = wp_get_post_categories( $item->ID, array( 'fields' => 'names' ) );
11674                                   if ( ! empty( $cats ) ) {
11675                                           foreach ( $cats as $category ) {
11676                                                   $roomCategory .= ( $roomCategory ? ', ' : '' ) . $category;
11677                                           }
11678                                   }
11679
11680                                   $roomDescription = $item->post_content;
11681
11682                                   $groupMode = get_post_meta( $item->ID, 'groupMode', true );
11683                                   if ( ! $groupMode ) {
11684                                           $groupMode = 'Free';
11685                                   }
11686
11687                                   $groupCPM = get_post_meta( $item->ID, 'groupCPM', true );
11688                                   if ( ! $groupCPM ) {
11689                                           $groupCPM = 0;
```

```
11690                                   }
11691
11692                                   // extras
11693                                   $groupParameters = get_post_meta( $item->ID, 'groupParameters', true );
11694
11695                                   // sales counters
11696                                   $paidSessionsPrivate = get_post_meta( $item->ID, 'paidSessionsPrivate', true );
11697                                   $paidSessionsGroup   = get_post_meta( $item->ID, 'paidSessionsGroup', true );
11698
11699
11700                                   $vw_featured  = get_post_meta( $item->ID, 'vw_featured', true );
11701
11702                           if ($json)
11703                                   {
11704                                           $ritem['name'] = $name;
11705                                           $ritem['performerStatus'] = $performerStatus;
11706                                           $ritem['lastOnline'] = $age;
11707                                           $ritem['url'] = $url;
11708                                           $ritem['snapshot'] = $snapshot;
11709                                           $ritem['label'] = $roomLabel;
11710                                           $ritem['category'] = $roomCategory;
11711                                           $ritem['tags'] = $roomTags;
11712                                           $ritem['brief'] = $roomBrief;
11713                                           $ritem['description'] = $roomDescription;
11714                                           $ritem['clientCPM'] = $clientCPM;
11715                                           $ritem['groupMode'] = $groupMode;
11716                                           $ritem['groupCPM'] = $groupCPM;
11717                                           $ritem['featured'] = $vw_featured;
11718                                           $ritem['maxViewers'] = intval( get_post_meta( $item->ID, 'maxViewers', true ) );
11719                                           $ritem['viewers'] = intval( get_post_meta( $item->ID, 'viewers', true ) );
11720                                   }
11721
11722                                   //icons: lovense toy
11723                                   $iconsCode = '';
11724                                   if ($options['lovense'] && $options['lovenseToy'])
11725                                   {
11726                                           $lovenseToy = get_post_meta( $item->ID, 'lovenseToy', true );
11727                                           if ($lovenseToy) $iconsCode .= '<i class="ui icon game"></i>';
11728                                   }
11729
11730
11731                                           $ratingCode = '';
11732                                           $voteCode = '';
11733                                   if ( $options['rateStarReview'] ) {
11734
11735                                   //votes
11736                                   if (!$json) $voteCode = do_shortcode( '[videowhisper_vote post_id="' . $item->ID . '"]' );
11737
11738                                   //rating
11739                                           $rating = floatval( get_post_meta( $item->ID, 'rateStarReview_rating', true ) );
11740                                           if ($json) $ritem['rating'] = number_format($rating, 2);
11741                                           $max    = 5;
11742                                           if ( $rating > 0  && !$json) {
11743                                                   $ratingCode = '<div class="ui yellow star rating readonly" data-rating="' . round( $rating * $max ) . '" data-max-rating="' . $max . '"></div>'; // .
```

```
Line
    number_format($rating * $max,1)  . ' / ' . $max
11744                                           }
11745
11746                                           $featuredReview = '<div class="ui stackable
cards">' . do_shortcode( '[videowhisper_review_featured post_id="' . $item->ID . '"]' ) . '</div>';
11747                                           }
11748
11749                                           // featured webcam
11750                                           $featuredCode = '';
11751                                           if ( $vw_featured ) {
11752                                                   $featuredCode = '<div class="videowhisperWrap"><span
class="videowhisperFeatured">' . __( 'Featured', 'ppv-live-webcams' ) . ( $vw_featured > 1 ? '
x' . $vw_featured : '' ) . '</span></div>';
11753                                           }
11754
11755                                           // #performers#
11756                                           $checkin = get_post_meta( $item->ID, 'checkin', true );
11757                                           if ( $checkin ) {
11758                                                   if ( ! is_array( $checkin ) ) {
11759                                                           $checkin = array( $checkin );
11760                                                   }
11761                                           }
11762
11763                                           $performersCode = '';
11764
11765                                           if ( $checkin ) {
11766                                                   foreach ( $checkin as $performerID ) {
11767
$performersCode .= ( $performersCode ? ',' : '' ) . self::performerLink( $performerID, $options );
11768                                                   }
11769                                           }
11770
11771                                           if ($json)
11772                                           {
11773                                                   $ritem['performers'] = strip_tags($performersCode);
11774                                           }
11775
11776
if ( strip_tags($performersCode) == $roomLabel ) $performersCode ='';
11777
11778                                           $urlCall = add_query_arg( 'vwsm', 'private', $url );
11779
11780                                           $requests_disable = self::is_true( get_post_meta( $item-
>ID, 'requests_disable', true ) );
11781
11782                                           $enterLabel = __( 'Enter', 'ppv-live-webcams' );
11783
11784                                           $group_disabled = self::is_true( get_post_meta( $item-
>ID, 'group_disabled', true ) );
11785                                           if ( $group_disabled ) {
11786                                                   $url        = $urlCall;
11787                                                   $enterLabel = __( 'Call', 'ppv-live-webcams' );
11788
11789                                                   if ( ! $isLoggedin ) {
11790                                                           $url
= ( $options['p_videowhisper_login'] > 0 ? get_permalink( $options['p_videowhisper_login'] ) : wp_login_u
rl() );
11791                                                           $enterLabel = __( 'Login', 'ppv-live-webcams' );
11792                                                   }
11793                                           }
11794
11795                                           // #enter#
11796                                           $enterCode = '<div class="videowhisperEnterDropdown"><a
href="' . $url . '"><button class="videowhisperEnterButton">' . $enterLabel . '</button></a><div
```

```
Line
    class="videowhisperEnterDropdown-content">';
11797
11798                                   $buttonChat    = '<a class="videowhisperButtonChat"
href="' . $url . '">' . $enterLabel . '</a>';
11799                                   $buttonMessage = '';
11800                                   $buttonCall    = '';
11801
11802                                   if ($json)
11803                                   {
11804                                           $ritem['enterUrl'] = $url;
11805                                           $ritem['enterLabel'] = $enterLabel;
11806                                   }
11807
11808                                   if (!$json)
11809                                   {
11810                                   //enter menu
11811                                   if ( ! $isLoggedin ) {
11812
11813                                           if ( ! $groupCPM && ! $group_disabled ) {
11814                                                   $enterCode .= '<a
href="' . $url . '">' . $groupMode . '</a>';
11815                                           }
11816                                           $enterCode .= '<a
href="' . ( $options['p_videowhisper_login'] > 0 ? get_permalink( $options['p_videowhisper_login'] ) : wp
_login_url() ) . '">' . __( 'Login for More', 'ppv-live-webcams' ) . ' ...' . '</a>';
11817                                   } else {
11818                                           if ( ! $group_disabled ) {
11819                                                   $enterCode .= '<a
href="' . $url . '">' . $groupMode . ( $groupCPM ? '
' . $groupCPM . $options['currencypm'] : '' ) . '</a>';
11820                                           }
11821
11822                                           // special modes
11823                                           $room_audio = self::is_true( get_post_meta( $item-
>ID, 'room_audio', true ) );
11824                                           $room_text = self::is_true( get_post_meta( $item-
>ID, 'room_text', true ) );
11825
11826                                   if ( ! $requests_disable ) if (!$room_audio) if (!$room_text) if ( $options['modeVideo'] != '0' )
11827                                           {
11828                                                   $enterCode .= '<a
href="' . $urlCall . '">' . __( 'Video Call', 'ppv-live-webcams' ) . '
' . ( $clientCPM ? $clientCPM . $options['currencypm'] : '' ) . '</a>';
11829                                                   $buttonCall = '<a class="videowhisperButtonCall"
href="' . $urlCall . '">' . __( 'Video Call', 'ppv-live-webcams' ) . '
' . ( $clientCPM ? $clientCPM . $options['currencypm'] : '' ) . '</a>';
11830                                           }
11831
11832                                           if ( $options['modeAudio'] ) if (!$room_text) {
11833
$clientCPM = self::clientCPM( $name, $options, $item->ID, 'audio' );
11834                                                   if ( ! $requests_disable ) {
11835                                                           $enterCode .= '<a
href="' . $urlCall . '&calltype=audio">' . __( 'Audio Call', 'ppv-live-webcams' ) . '
' . ( $clientCPM ? $clientCPM . $options['currencypm'] : '' ) . '</a>';
11836
11837                                                           $buttonCall = '<a
class="videowhisperButtonCall" href="' . $urlCall . '">' . __( 'Audio Call', 'ppv-live-webcams' ) . '
' . ( $clientCPM ? $clientCPM . $options['currencypm'] : '' ) . '</a>';
11838
11839                                                   }
11840                                           }
11841
11842                                           if ( $options['modeText'] ) {
```

```
11843                                                   $clientCPM = self::clientCPM( $name, $options, $item->ID, 'text' );
11844                                                   if ( ! $requests_disable ) {
11845                                                           $enterCode .= '<a
href="' . $urlCall . '&calltype=text">' . __( 'Text Chat', 'ppv-live-webcams' ) . '
' . ( $clientCPM ? $clientCPM . $options['currencypm'] : '' ) . '</a>';
11846
11847                                                           $buttonCall = '<a
class="videowhisperButtonCall" href="' . $urlCall . '">' . __( 'Text Chat', 'ppv-live-webcams' ) . '
' . ( $clientCPM ? $clientCPM . $options['currencypm'] : '' ) . '</a>';
11848                                                   }
11849                                           }
11850
11851                                           if ( $options['messages'] ) {
11852
11853                                                   $messagesCost = floatval( get_post_meta( $item-
>ID, 'question_price', true ) );
11854                                                   if ( ! $messagesCost ) {
11855                                                           $messagesCost = floatval( $options['messagesCost'] );
11856                                                   }
11857                                                   $closed = intval( get_post_meta( $item-
>ID, 'question_closed', true ) );
11858
11859                                                   if (!$closed) $enterCode   .= '<a
href="' . add_query_arg( 'view', 'messages', $url ) . '">' . __( 'Question', 'ppv-live-webcams' ) . '
' . ( $messagesCost ? $messagesCost . $options['currency'] : '' ) . '</a>';
11860
11861                                                   $buttonMessage = '<a
class="videowhisperButtonMessage"
href="' . add_query_arg( 'view', 'messages', $url ) . '">' . __( 'Question', 'ppv-live-webcams' ) . '
' . ( $messagesCost ? $messagesCost . $options['currency'] : '' ) . '</a>';
11862
11863                                           }
11864
11865                                           if ( is_array( $groupParameters ) ) {
11866
if ( isset($groupParameters['2way']) && $groupParameters['2way'] > 0 ) {
11867                                                   $c2way = 0;
11868
11869                                                   $mode2way = get_post_meta( $item-
>ID, 'mode2way', true );
11870                                                   if ( is_array( $mode2way ) ) {
11871                                                           $c2way = count( $mode2way );
11872                                                   }
11873
11874
11875                                                   }
11876                                           }
11877
11878                                           if ( $options['voyeurAvailable'] == 'always' ||
 ( $options['voyeurAvailable'] == 'private' && $privateShow ) ||
 ( $options['voyeurAvailable'] == 'public' && ! $privateShow ) ) {
11879                                                   if ( is_array( $groupParameters ) ) {
11880
if ( array_key_exists( 'voyeur', $groupParameters ) ) {
11881
if ( $groupParameters['voyeur'] ) {
11882                                                                                   $enterCode .= '<a
href="' . add_query_arg( 'vwsm', 'voyeur', $url ) . '">' . __( 'Voyeur', 'ppv-live-
webcams' ) . ( $groupParameters['cpmv'] ?
' . $groupParameters['cpmv'] . $options['currencypm'] : '' ) . '</a>';
11883
11884                                                                   }
11885                                                   }
11886                                           }
11887                                   }
```

```
11888                                   }
11889                               //
11890                               }
11891
11892
11893
11894 if ($json) $ritem['profileUrl'] = add_query_arg( 'view', 'content', $url );
11895
11896                                   if (!$json)
11897                                   {
11898                                   $enterCode .= '<a
      href="' . add_query_arg( 'view', 'content', $url ) . '">' . __( 'Profile', 'ppv-live-webcams' ) . '</a>';
11899                                   $enterCode .= '</div></div>';
11900
11901                                   // replace
11902
      $templateReplace = array( $roomLabel, $age, $clientCPM, $roomBrief, $roomTags, $url, $snapshot, $thumbWid
      th, $thumbHeight, $banLink, $groupMode, $groupCPM, $performersCode, $options['currency'], $previewCode, $
      enterCode, $paidSessionsPrivate, $paidSessionsGroup, $ratingCode, $performerStatus, $roomCategory, $featu
      redReview ?? '', $roomDescription, $featuredCode, $buttonChat, $buttonCall, $buttonMessage, $voteCode, $i
      consCode );
11903
11904
      echo str_replace( $templateSearch, $templateReplace, $listingTemplate )  . esc_html( $debugCode );
11905                                   }
11906
11907                                   if ($json)
11908                                   {
11909                                           //special profile fields
11910
      if ( $json_fields && count($json_fields) ) foreach ($json_fields as $field)
11911                                           {
11912                                                   $value = get_post_meta( $item-
      >ID, $field, true );
11913                                                   if ($value) $ritem[$field] = $value;
11914                                           }
11915
11916                                           $ritem['ban'] = get_post_meta( $item-
      >ID, 'vw_banCountries', true );
11917
11918                                           $result['rooms'] []= $ritem;
11919                                   }
11920
11921                                   //per item
11922                               }
11923                               //not empty
11924                       } else {
11925
11926                               if ($json)
11927                               {
11928                                   $result['warning'] = __( 'No rooms match current criteria. A room
      requires a picture or snapshot from live broadcast to get listed.', 'ppv-live-webcams' );
11929                               }
11930                               else
11931                               {
11932                               echo __( 'No rooms match current criteria. A room requires a picture or
      snapshot from live broadcast to get listed.', 'ppv-live-webcams' );
11933                               if ( $debugMode ) {
11934                                   echo 'Debug (no webcams found):<br><textarea readonly cols="120"
      rows="4">';
11935                                   var_dump( $args );
11936                                   echo '</textarea>';
11937                               }
11938
```

```
11939                                   }
11940                           }
11941
11942                           if (!$json) echo '</div><!--VideoWhisper Room List End-->'; // end room list
11943
11944                   //horizontal slider buttons in container
11945                           if ($listingsCount == $perPage || $page) //only if listings
11946                           if (!$json && $layout == 'horizontal')
11947                           {
11948
11949                                   if ($listingsCount == $perPage || $page) echo '<a
class="horizontalPrevious ui icon button tiny compact" href="javascript:void(0);"
onClick="' . ($page > 0  ? 'if
(document.getElementById(\'videowhisperListings' . esc_attr( $id ) . '\').scrollLeft == 0) {
aurl' . esc_attr( $id ) . '=\'' . esc_url( $ajaxURLall) . '&p=' . intval( $page - 1 ) . '\';
loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large loader\\\'>Loading
previous page...</div>\') };' : '' )  . ';
document.getElementById(\'videowhisperListings' . esc_attr( $id ) . '\').scrollLeft -=
' . intval( $options['thumbWidth'] + 4 ) . '"><i class="left arrow icon"></i></a> ';
11950
11951                                   if ($listingsCount == $perPage) echo '<a class="horizontalNext ui icon
button tiny compact" href="javascript:void(0);" onClick="var vwElement =
document.getElementById(\'videowhisperListings' . esc_attr( $id ) . '\');
' . ( count( $postslist ) == $perPage ? 'var maxScrollLeft = vwElement.scrollWidth -
vwElement.clientWidth; if (vwElement.scrollLeft >=  maxScrollLeft) {
aurl' . esc_attr( $id ) . '=\'' . esc_url( $ajaxURLall) . '&p=' . intval( $page + 1 ) . '\';
loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large loader\\\'>Loading next
page...</div>\') }' : '' )  . '; vwElement.scrollLeft += ' . intval( $options['thumbWidth'] + 4 ) . ';">
<i class="right arrow icon"></i></a> ';
11952
11953                           }
11954
11955                           if ($json)
11956                           {
11957                                   echo json_encode($result);
11958                                   die();
11959                           }
11960
11961                           // footer start
11962                           if (!$json)
11963                           {
11964
11965                           echo '<div class="ui clearing divider hidden tiny"></div>';
11966
11967                           echo '<div class="ui' . esc_attr( $options['interfaceClass'] ) . ' equal width
grid">';
11968
11969                           // pagination
11970                           if ( $selectPage ) {
11971                                   //mage sure pagination is displayed under current content
11972                                   echo '<div class="column"><div class="ui form"><div class="inline
fields">';
11973
11974                                   if ( $page > 0 && $layout != 'horizontal' ) {
11975                                           echo ' <a class="ui labeled icon button"
href="javascript:void(0);"
onclick="aurl' . esc_attr( $id ) . '=\'' . esc_url( $ajaxURLall) . '&p=' . intval( $page - 1 ) . '\';
loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large loader\\\'>Loading
previous page...</div>\');"><i class="left arrow icon"></i> ' . __( 'Previous', 'ppv-live-
webcams' ) . '</a> ';
11976                                   }
11977
11978                                   echo '<a class="ui secondary button" href="#"> ' . __( 'Page', 'ppv-live-
webcams' ) . ' ' . intval( $page + 1 ) . ' </a>';
11979
11980
if (is_array( $postslist )) if ( count( $postslist ) == $perPage && $layout != 'horizontal' ) {
```

```
11981                                   echo ' <a class="ui right labeled icon button"
href="javascript:void(0);"
onclick="aurl' . esc_attr( $id ) . '=\'' . esc_url( $ajaxURLall) . '&p=' . intval( $page + 1 ) . '\';
loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large loader\\\'>Loading next
page...</div>\');">' . __( 'Next', 'ppv-live-webcams' ) . ' <i class="right arrow icon"></i></a> ';
11982                                   }
11983
11984                                   echo '</div></div></div>';
11985                           }
11986
11987                           if ( $selectLayout ) {
11988                                   echo '
11989 <div class="right floated column">
11990 <div class="ui icon buttons right floated">
11991   <a class="ui button ' . ( $layout == 'grid' ? 'active' : '' ) . '" data-tooltip="' . __( 'Grid', 'ppv-
live-webcams' ) . '"  href="JavaScript: void()"
onclick="aurl' . esc_attr( $id ) . '=\'' . esc_attr( $ajaxURLlay ) . '&layout=grid\';
loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large loader\\\'>Loading grid
layout...</div>\');">
11992     <i class="th icon"></i>
11993   </a>
11994   <a class="ui button ' . ( $layout == 'list' ? 'active' : '' ) . '" data-tooltip="' . __( 'Full Row
List', 'ppv-live-webcams' ) . '" href="JavaScript: void()"
onclick="aurl' . esc_attr( $id ) . '=\'' . esc_attr( $ajaxURLlay ) . '&layout=list\';
loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large loader\\\'>Loading list
layout...</div>\');">
11995     <i class="th list icon"></i>
11996   </a>
11997  <a class="ui button ' . ( $layout == 'horizontal' ? 'active' : '' ) . '" data-
tooltip="' . __( 'Horizontal Slider', 'ppv-live-webcams' ) . '" href="JavaScript: void()"
onclick="aurl' . esc_attr( $id ) . '=\'' . esc_attr( $ajaxURLlay ) . '&layout=horizontal\';
loadWebcams' . esc_attr( $id ) . '(\'<div class=\\\'ui active inline text large loader\\\'>Loading
horizontal slider layout...</div>\');">
11998     <i class="alternate arrows horizontal icon"></i>
11999   </a>
12000 </div>
12001 </div>';
12002                           }
12003
12004                           echo '</div>';
12005                           }
12006                           // footer end
12007
12008                           if ( ! $isMobile && $options['videosharevod'] && !$json ) {
12009
12010                                   echo '
12011 <PRE style="display: none"><SCRIPT language="JavaScript">
12012 jQuery(document).ready(function()
12013 {
12014
12015 var hHandlers = jQuery(".videowhisperWebcam")
12016 .on("mouseenter", hoverVideoWhisper)
12017 .on("click", hoverVideoWhisper)
12018 .on("mouseleave", outVideoWhisper);
12019
12020 var hHandlers2 = jQuery(".videowhisperWebcam2").hover( hoverVideoWhisper, outVideoWhisper );
12021
12022 function hoverVideoWhisper(e) {
12023         var vid = jQuery("video", this).get(0);
12024         if (vid && vid.paused) {
12025                 vid.play().catch(function(error) {
12026                         if (error.name === "NotAllowedError") {
12027                                 // Handle the NotAllowedError, for example, by showing a message to the
user
12028                         }
```

```
12029                         console.warn("Autoplay video on hover error:", error);
12030                 });
12031         }
12032 }
12033
12034 function outVideoWhisper(e) {
12035     var vid = jQuery(\'video\', this).get(0);
12036     if (vid) vid.pause();
12037 }
12038 });
12039 </SCRIPT></PRE>
12040 ';
12041
12042                         }
12043
12044                 //close layout with menu
12045                 if ( $menu && !$json) echo '</div></div>';
12046
12047
12048                                         if ( $debugMode ) {
12049                                                 global $wpdb;
12050                                                 echo '<!-- Debug: Query:
'. wp_kses_post( $cacheKey ) . '-->';
12051                                         }
12052
12053                         // echo '<!-- Generated '. date(DATE_RFC2822) . ' ' . $cacheQuery . ' ' .
$cacheKey . ' -->';
12054
12055                         // self::stringSave($cachePath, $htmlCode);
12056
12057                         self::scriptThemeMode($options);
12058
12059                         die();
12060         }
12061
12062         // end room list
12063
12064         static function scriptThemeMode($options)
12065 {
12066         $theme_mode = '';
12067
12068         //check if using the FansPaysite theme and apply the dynamic theme mode
12069
if (function_exists('fanspaysite_get_current_theme_mode')) $theme_mode = fanspaysite_get_current_theme_mo
de();
12070
12071         if (!$theme_mode) $theme_mode = $options['themeMode'] ?? '';
12072
12073         if (!$theme_mode) return '<!-- No theme mode -->';
12074
12075         // JavaScript function to apply the theme mode
12076         echo '<script>
12077         if (typeof setConfiguredTheme !== "function")  // Check if the function is already defined
12078         {
12079
12080                 function setConfiguredTheme(theme) {
12081                         if (theme === "auto") {
12082                                 if (window.matchMedia && window.matchMedia("(prefers-color-scheme:
dark)").matches) {
12083                                         document.body.dataset.theme = "dark";
12084                                 } else {
12085                                         document.body.dataset.theme = "";
12086                                 }
```

```
12087                         } else {
12088                                 document.body.dataset.theme = theme;
12089                         }
12090
12091                         if (document.body.dataset.theme == "dark")
12092                         {
12093                         jQuery("body").find(".ui").addClass("inverted");
12094                         jQuery("body").addClass("inverted");
12095                         }else
12096                         {
12097                                 jQuery("body").find(".ui").removeClass("inverted");
12098                                 jQuery("body").removeClass("inverted");
12099                         }
12100
12101                         console.log("PaidVideochat/setConfiguredTheme:", theme);
12102                 }
12103         }
12104
12105         setConfiguredTheme("' . esc_js($theme_mode) . '");
12106
12107         </script>';
12108 }
12109
12110
12111         static function videowhisper_cam_webrtc_playback( $atts ) {
12112                 $stream  = '';
12113                 $postID  = 0;
12114                 $options = get_option( 'VWliveWebcamsOptions' );
12115
12116                 if ( is_single() ) {
12117                         $postID = get_the_ID();
12118                         if ( get_post_type( $postID ) == $options['custom_post'] ) {
12119                                 $room = self::post_title( $postID );
12120                         } else {
12121                                 $postID = 0;
12122                         }
12123                 }
12124
12125                 if ( ! $room ) {
12126                         $room = sanitize_text_field( $_GET['room'] );
12127                 }
12128
12129                 $atts = shortcode_atts(
12130                         array(
12131                                 'room'      => $room,
12132                                 'width'     => '480px',
12133                                 'height'    => '360px',
12134                                 'webcam_id' => $postID,
12135                                 'silent'    => 0,
12136                         ),
12137                         $atts,
12138                         'videowhisper_cam_webrtc_playback'
12139                 );
12140                 $atts = array_map('esc_attr', $atts);
12141
12142                 if ( $atts['room'] ) {
12143                         $room = sanitize_text_field( $atts['room'] ); // parameter channel="name"
12144                 }
12145                 if ( $atts['webcam_id'] ) {
12146                         $postID = sanitize_text_field( $atts['webcam_id'] );
12147                 }
```

```
12148                 $width = sanitize_text_field( $atts['width'] );
12149                 if ( ! $width ) {
12150                         $width = '100%';
12151                 }
12152                 $height = sanitize_text_field( $atts['height'] );
12153                 if ( ! $height ) {
12154                         $height = '360px';
12155                 }
12156
12157                 $room = sanitize_file_name( $room );
12158
12159                 if ( ! $postID && $room ) {
12160                         global $wpdb;
12161                         $postID = $wpdb->get_var( $wpdb->prepare( "SELECT ID FROM $wpdb->posts WHERE post_title = %s and post_type=%s LIMIT 0,1", sanitize_file_name( $room ), $options['custom_post'] ) );
12162                 }
12163
12164                 if ( ! $room ) {
12165                         return 'WebRTC Playback Error: Missing webcam room name!';
12166                 }
12167
12168                 $stream = self::webcamStreamName( $room, $postID );
12169
12170                 $userID = 0;
12171                 if ( is_user_logged_in() ) {
12172                         $userName = $options['userName'];
12173                         if ( ! $userName ) {
12174                                 $userName = 'user_login';
12175                         }
12176                         $current_user = wp_get_current_user();
12177                         if ( $current_user->$userName ) {
12178                                 $username = sanitize_file_name( $current_user->$userName );
12179                         }
12180                         $userID = $current_user->ID;
12181                 }
12182
12183                 // detect browser
12184                 $agent   = $_SERVER['HTTP_USER_AGENT'];
12185                 $Android = stripos( $agent, 'Android' );
12186                 $iOS     = ( strstr( $agent, 'iPhone' ) || strstr( $agent, 'iPod' ) || strstr( $agent, 'iPad' ) );
12187                 $Safari  = ( strstr( $agent, 'Safari' ) && ! strstr( $agent, 'Chrome' ) );
12188                 $Firefox = stripos( $agent, 'Firefox' );
12189
12190                 $codeMuted = '';
12191                 if ( ! $Firefox ) {
12192                         $codeMuted = 'muted';
12193                 }
12194
12195                 // WebRTC playback: detect source type and transcode if necessary
12196                 $streamProtocol = get_post_meta( $postID, 'stream-protocol', true ); // rtmp/webrtc
12197                 $streamMode     = get_post_meta( $postID, 'stream-mode', true ); // direct/safari_pc
12198
12199                 if ( $streamProtocol == 'rtsp' && $streamMode == 'direct' ) {
12200                         $stream_webrtc = $stream;
12201                 } else {
12202                         $stream_webrtc = self::transcodeStreamWebRTC( $stream, $postID, $options );
12203                 }
12204
12205                 if ( ! $stream_webrtc ) {
12206                         $htmlCode .= 'Error: WebRTC stream name not available!';
12207                 }
```

```
12208
12209 $streamQuery = self::webrtcStreamQuery( $userID, $postID, 0, $stream_webrtc, $options, 0, $room );
12210
12211                 self::enqueueUI();
12212
12213                 wp_enqueue_script( 'webrtc-
adapter', dirname( plugin_dir_url( __FILE__ ) ) . '/scripts/adapter.js', array( 'jquery' ) );
12214                 wp_enqueue_script( 'videowhisper-webrtc-
playback', dirname( plugin_dir_url( __FILE__ ) ) . '/scripts/vwrtc-
playback.js', array( 'jquery', 'webrtc-adapter' ) );
12215
12216                 $wsURLWebRTC       = $options['wsURLWebRTC'];
12217                 $applicationWebRTC = $options['applicationWebRTC'];
12218
12219                 $htmlCode .= <<<HTMLCODE
12220                 <div class="videowhisper-webrtc-video">
12221                 <video id="remoteVideo" class="videowhisper_htmlvideo" autoplay playsinline controls
$codeMuted style="width:$width; height:$height"></video>
12222                 <!--$streamProtocol|$stream|$stream_webrtc-->
12223                 </div>
12224
12225                 <div>
12226                         <span id="sdpDataTag"></span>
12227                 </div>
12228
12229                 <script type="text/javascript">
12230
12231                         var videoBitrate = 360;
12232                         var audioBitrate = 64;
12233                         var videoFrameRate = "29.97";
12234                         var videoChoice = "$videoCodec";
12235                         var audioChoice = "$audioCodec";
12236
12237                         var userAgent = navigator.userAgent;
12238                     var wsURL = "$wsURLWebRTC";
12239                         var streamInfo = {applicationName:"$applicationWebRTC",
streamName:"$streamQuery", sessionId:"[empty]"};
12240                         var userData = {param1:"value1","videowhisper":"webrtc-playback"};
12241
12242                 jQuery( document ).ready(function() {
12243                 browserReady();
12244 });
12245
12246                 </script>
12247 HTMLCODE;
12248
12249                 if ( ! $atts['silent'] ) {
12250                         if ( $codeMuted ) {
12251                                 $htmlCode .= 'Playback is muted to allow auto play: enable audio from
player controls.';
12252                         }
12253                 }
12254                 return $htmlCode;
12255         }
12256
12257
12258         static function videowhisper_cam_webrtc_broadcast( $atts ) {
12259                 $stream = '';
12260
12261                 if ( ! is_user_logged_in() ) {
12262                         return "<div class='error'>" . __( 'Broadcasting not allowed: Only logged in
users can broadcast!', 'ppv-live-webcams' ) . '</div>';
```

```
12263                 }
12264
12265                 $options = get_option( 'VWliveWebcamsOptions' );
12266
12267                 // 1. webcam post
12268                 $postID = get_the_ID();
12269                 if ( is_single() ) {
12270                         if ( get_post_type( $postID ) == $options['custom_post'] ) {
12271                                 $room = self::post_title( $postID );
12272                         }
12273                 }
12274
12275                 // 2. shortcode param
12276                 // $stream = get_post_meta($postID, 'performer', true);
12277
12278                 $atts = shortcode_atts(
12279                         array(
12280                                 'room'      => $room,
12281                                 'webcam_id' => $postID,
12282                         ),
12283                         $atts,
12284                         'videowhisper_cam_webrtc_broadcast'
12285                 );
12286                 $atts = array_map('esc_attr', $atts);
12287
12288                 if ( $atts['room'] ) {
12289                         $room = $atts['room'];
12290                 }
12291
12292                 $room = sanitize_file_name( $room );
12293
12294                 if ( ! $room ) {
12295                         return "<div class='error'>Can't load application: Missing room name!</div>";
12296                 }
12297
12298                 $postID = $atts['webcam_id'];
12299
12300                 if ( $room && ! $postID ) {
12301                         global $wpdb;
12302                         $postID = $wpdb->get_var( $wpdb->prepare( "SELECT ID FROM $wpdb->posts WHERE
post_title = %s and post_type=%s LIMIT 0,1", sanitize_file_name( $room ), $options['custom_post'] ) );
12303                 }
12304
12305                 $post = get_post( $postID );
12306                 if ( ! $post ) {
12307                         return "<div class='error'>Webcam post not found!</div>";
12308                 }
12309
12310                 // performer check
12311                 $current_user = wp_get_current_user();
12312
12313                 $loggedin  = 0;
12314                 $msg       = '';
12315                 $performer = 0;
12316                 $balance   = 0;
12317                 $uid       = 0;
12318
12319                 // user info
12320                 if ( $current_user ) {
12321                         if ( $current_user->ID > 0 ) {
12322
```

```
12323                                   // identify if user can be room performer
12324                                   if ( self::isRoomPerformer( $post, $current_user ) ) {
12325                                           $performer = 1;
12326                                   } else {
12327                                           $performer = 0;
12328                                   }
12329                                   if ( ! $performer ) {
12330                                           return "<div class='error'>Only room performers are allowed to broadcast!</div>";
12331                                   }
12332
12333                                   $performerName = self::performerName( $current_user, $options );
12334                                   $stream        = $performerName;
12335
12336                                   // $debug .= "p_a:".$post->post_author .".uID:".$current_user->ID.".pName:$performerName.s:$stream";
12337                                   $uid = $current_user->ID;
12338
12339                                   if ( $uid ) {
12340                                           self::billSessions( $uid );
12341                                           $balance        = self::balance( $uid );
12342                                           $balancePending = $balance;
12343                                   }
12344                           }
12345                   }
12346
12347                   $streamQuery = self::webrtcStreamQuery( $current_user->ID, $postID, 1, $stream, $options, 0, $room );
12348
12349                   // detect browser
12350                   $agent   = $_SERVER['HTTP_USER_AGENT'];
12351                   $Android = stripos( $agent, 'Android' );
12352                   $iOS     = ( strstr( $agent, 'iPhone' ) || strstr( $agent, 'iPod' ) || strstr( $agent, 'iPad' ) );
12353                   $Safari  = ( strstr( $agent, 'Safari' ) && ! strstr( $agent, 'Chrome' ) );
12354                   $Firefox = stripos( $agent, 'Firefox' );
12355
12356                   // publishing as WebRTC - save info for responsive playback
12357                   update_post_meta( $postID, 'performer', $stream );
12358                   update_post_meta( $postID, 'performerUserID', $current_user->ID );
12359
12360                   update_post_meta( $postID, 'stream-protocol', 'rtsp' );
12361                   update_post_meta( $postID, 'stream-type', 'webrtc' );
12362                   update_post_meta( $postID, 'roomInterface', 'html5' );
12363
12364                   if ( ! $iOS && $Safari ) {
12365                           update_post_meta( $postID, 'stream-mode', 'safari_pc' ); // safari on pc encoding profile issues
12366                   } else {
12367                           update_post_meta( $postID, 'stream-mode', 'direct' );
12368                   }
12369
12370                   self::enqueueUI();
12371
12372                   wp_enqueue_script( 'webrtc-adapter', dirname( plugin_dir_url( __FILE__ ) ) . '/scripts/adapter.js', array( 'jquery' ) );
12373                   wp_enqueue_script( 'videowhisper-webrtc-broadcast', dirname( plugin_dir_url( __FILE__ ) ) . '/scripts/vwrtc-publish.js', array( 'jquery', 'webrtc-adapter' ) );
12374
12375                   $wsURLWebRTC       = $options['wsURLWebRTC'];
12376                   $applicationWebRTC = $options['applicationWebRTC'];
12377
```

```
12378                 $videoCodec = $options['webrtcVideoCodec']; // 42e01f
12379                 $audioCodec = $options['webrtcAudioCodec']; // opus
12380
12381                 $videoBitrate = (int) $options['webrtcVideoBitrate'];
12382                 if ( ! $videoBitrate ) {
12383                         $videoBitrate = 400; // 400 max for tcp with Wowza
12384                 }
12385
12386                 $htmlCode .= "<!--
WebRTC_Broadcast|r:$room|p:$postID|$agent|i:$iOS|a:$Android|Sa:$Safari|Ff:$Firefox-->";
12387
12388                 $broadcastCode = <<<HTMLCODE
12389                 <div class="videowhisper-webrtc-camera">
12390                 <video id="localVideo" class="videowhisper_htmlvideo" autoplay playsinline muted
style="width:640px;height:480px;"></video>
12391                 </div>
12392
12393                 <div class="ui segment form">
12394                         <span id="sdpDataTag">Connecting...</span>
12395
12396 <hr class="divider" />
12397     <div class="field inline">
12398         <label for="videoSource">Video Source </label><select class="ui dropdown v-select"
id="videoSource"></select>
12399     </div>
12400
12401     <div class="field inline">
12402         <label for="videoResolution">Video Resolution </label><select class="ui dropdown v-select"
id="videoResolution"></select>
12403     </div>
12404
12405          <div class="field inline">
12406         <label for="audioSource">Audio Source </label><select class="ui dropdown v-select"
id="audioSource"></select>
12407     </div>
12408
12409                 </div>
12410
12411
12412                 <script type="text/javascript">
12413
12414                         var userAgent = navigator.userAgent;
12415                     var wsURL = "$wsURLWebRTC";
12416                         var streamInfo = {applicationName:"$applicationWebRTC",
streamName:"$streamQuery", sessionId:"[empty]"};
12417                         var userData = {param1:"value1","videowhisper":"webrtc-broadcast"};
12418                         var videoBitrate = $videoBitrate;
12419                         var audioBitrate = 64;
12420                         var videoFrameRate = "29.97";
12421                         var videoChoice = "$videoCodec";
12422                         var audioChoice = "$audioCodec";
12423
12424                 jQuery( document ).ready(function() {
12425                 browserReady();
12426                 jQuery(".ui.dropdown:not(.multi,.fpsDropdown)").dropdown();
12427
12428 });
12429                 </script>
12430 HTMLCODE;
12431
12432                 // AJAX Chat for WebRTC broadcasting
12433
12434                 // htmlchat ui
```

```
12435                 // css
12436 wp_enqueue_style( 'jScrollPane', dirname( plugin_dir_url( __FILE__ ) ) . '/htmlchat/js/jScrollPane/jScrollPane.css' );
12437 wp_enqueue_style( 'htmlchat', dirname( plugin_dir_url( __FILE__ ) ) . '/htmlchat/css/chat-broadcast.css' );
12438
12439                 // js
12440                 wp_enqueue_script( 'jquery' );
12441                 wp_enqueue_script( 'jScrollPane-mousewheel', dirname( plugin_dir_url( __FILE__ ) ) . '/htmlchat/js/jScrollPane/jquery.mousewheel.js' );
12442 wp_enqueue_script( 'jScrollPane', dirname( plugin_dir_url( __FILE__ ) ) . '/htmlchat/js/jScrollPane/jScrollPane.min.js' );
12443 wp_enqueue_script( 'htmlchat', dirname( plugin_dir_url( __FILE__ ) ) . '/htmlchat/js/script.js', array( 'jquery', 'jScrollPane' ) );
12444
12445                 $ajaxurl = admin_url() . 'admin-ajax.php?action=vmls_htmlchat&room=' . urlencode( sanitize_file_name( $stream ) );
12446
12447                 $loginCode = '<a href="' . wp_login_url() . '">Login is required to chat!</a>';
12448 $buttonSFx = dirname( plugin_dir_url( __FILE__ ) ) . '/videowhisper/templates/messenger/message.mp3';
12449                 $tipsSFx = dirname( plugin_dir_url( __FILE__ ) ) . '/videowhisper/templates/messenger/tips/';
12450
12451                 if ( $options['tips'] ) {
12452
12453                         // broacaster: only balance
12454
12455                         $tipbuttonCodes = '<p>Viewers can send you tips. Balance will update shortly after receiving a tip.</p>';
12456
12457                         $tipsCode = <<<TIPSCODE
12458 <div id="tips" class="ui segment form">
12459 <div class="inline fields">
12460
12461 <div class="ui label olive large">
12462   <i class="money bill alternate icon large"></i>Balance: <span id="balanceAmount" class="inline">-</span>
12463 </div>
12464
12465 $tipbuttonCodes
12466 </div>
12467 </div>
12468 TIPSCODE;
12469                 }
12470
12471                 $htmlCode .= <<<HTMLCODE
12472 <div id="videochatContainer">
12473 <!--Room:$stream-->
12474 <div id="streamContainer">
12475 $broadcastCode
12476 </div>
12477
12478 <div id="chatContainer">
12479
12480     <div id="chatUsers" class="ui segment"></div>
12481
12482     <div id="chatLineHolder"></div>
12483
12484     <div id="chatBottomBar" class="ui segment">
12485         <div class="tip"></div>
```

```
12486
12487         <form id="loginForm" method="post" action="" class="ui form">
12488 Login is required to chat!
12489                 </form>
12490
12491         <form id="submitForm" method="post" action="" class="ui form">
12492             <input id="chatText" name="chatText" class="rounded" maxlength="255" />
12493             <input type="submit" class="ui button" value="Submit" />
12494         </form>
12495
12496     </div>
12497 </div>
12498 </div>
12499 $tipsCode
12500
12501 <script>
12502 var vwChatAjax= '$ajaxurl';
12503 var vwChatButtonSFx =  '$buttonSFx';
12504 var vwChatTipsSFx =  '$tipsSFx';
12505 </script>
12506
12507
12508 HTMLCODE;
12509
12510                 if ( $options['transcodingWarning'] >= 1 ) {
12511                         $htmlCode .= '<p class="info"><small>Warning: WebRTC will play directly where
                        possible, depending on settings and viewer device. If transcoding is needed for playback, it may take up
                        to a couple of minutes for transcoder to start and WebRTC published stream to become available for RTMP
                        and HLS/MPEG DASH playback.</small></p>';
12512                 }
12513
12514                 return $htmlCode;
12515
12516         }
12517
12518
12519
12520
12521         static function webcamLinks( $postID, $options ) {
12522
12523
                if ( ! $options['profiles'] && ! $options['messages'] && ! $options['videosharevod'] && ! $options['pictu
                regallery'] && ! $options['rateStarReview'] ) {
12524                         return '';
12525                 }
12526
12527                 $addCode = '';
12528
12529                 $addCode .= '<a class="ui button tiny compact"
                href="' . get_permalink( $postID ) . '">' . __( 'Room Page', 'ppv-live-webcams' ) . '</a> ';
12530                 $addCode .= '<a class="ui button tiny compact"
                href="' . add_query_arg( 'view', 'content', get_permalink( $postID ) ) . '">' . __( 'Room Profile
                Content', 'ppv-live-webcams' ) . '</a> ';
12531                 if ( $options['profiles'] ) {
12532                         $addCode .= '<a class="ui button tiny compact"
                href="' . add_query_arg( 'view', 'profile', get_permalink( $postID ) ) . '">' . __( 'Room Profile
                Info', 'ppv-live-webcams' ) . '</a> ';
12533                 }
12534
12535                 if ( $options['messages'] ) {
12536                         $addCode .= '<a class="ui button tiny compact"
                href="' . add_query_arg( 'view', 'messages', get_permalink( $postID ) ) . '">' . __( 'Messages', 'ppv-
                live-webcams' ) . '</a> ';
12537                 }
```

```
12538
12539                   return $addCode;
12540           }
12541
12542
12543           static function webcamContent( $postID, $room, $options, $content = '' ) {
12544
12545                   $header   = array();
12546                   $contents = array();
12547                   $active  = '';
12548
12549                   $addCode = '';
12550
12551                   /*
12552                           $addCode .= '
12553                   <script>
12554                   jQuery(document).ready(function(){
12555
12556                   var vwTabs = jQuery(".tabular.menu .item");
12557                   if (vwTabs) vwTabs.tab();
12558                   else console.log("Interface warning: No tabs found in webcamContent");
12559
12560                   });
12561                   </script>';
12562
12563                           //tab header
12564                           $addCode .= '<a name="profile"></a> <div class="ui ' . $options['interfaceClass'] . ' top attached tabular menu">';
12565
12566                           $addCode .= '<a class="item active" data-tab="profile">' . __('Profile', 'ppv-live-webcams') . '</a>';
12567
12568                           if ($options['videosharevod']) $addCode .= '<a class="item" data-tab="videos">' . __('Videos', 'ppv-live-webcams') . '</a>';
12569                           if ($options['picturegallery']) $addCode .= '<a class="item" data-tab="pictures">' . __('Pictures', 'ppv-live-webcams') . '</a>';
12570                           if ($options['rateStarReview']) $addCode .= '<a class="item" data-tab="reviews">' . __('Reviews', 'ppv-live-webcams') . '</a>';
12571                           if ($options['messages']) $addCode .= '<a class="item" data-tab="messages">' . __('Messages', 'ppv-live-webcams') . '</a>';
12572
12573                           $addCode .= '</div>';
12574                   */
12575                   if ( $options['profiles'] ) {
12576                           $header['profile'] = __( 'Profile', 'ppv-live-webcams' );
12577                   }
12578                   if ( $options['videosharevod'] ) {
12579                           $header['videos'] = __( 'Videos', 'ppv-live-webcams' );
12580                   }
12581                   if ( $options['picturegallery'] ) {
12582                           $header['pictures'] = __( 'Pictures', 'ppv-live-webcams' );
12583                   }
12584                   if ( $options['rateStarReview'] ) {
12585                           $header['reviews'] = __( 'Reviews', 'ppv-live-webcams' );
12586                   }
12587
12588                   // digital content / assets from micropayments
12589                   $contentIDs = get_post_meta( $postID, 'contentIDs', true );
12590                   if ( ! is_array( $contentIDs ) ) {
12591                           $contentIDs = array();
12592                   }
12593                   if ( $options['micropaymentsAssets'] ) {
12594                           if ( shortcode_exists( 'videowhisper_content_list' ) ) {
```

```
12595                                   if ( count( $contentIDs ) ) {
12596                                           $header['content'] = __( 'Content', 'ppv-live-webcams' );
12597                                   }
12598                           }
12599                   }
12600
12601                   // products
12602                   $productIDs = get_post_meta( $postID, 'productIDs', true );
12603                   if ( ! is_array( $productIDs ) ) {
12604                           $productIDs = array();
12605                   }
12606                   if ( $options['woocommerce'] ) {
12607                           if ( shortcode_exists( 'products' ) ) {
12608                                   if ( count( $productIDs ) ) {
12609                                           $header['products'] = __( 'Products', 'ppv-live-webcams' );
12610                                   }
12611                           }
12612                   }
12613
12614                   if ( $options['messages'] ) {
12615                           $header['messages'] = __( 'Messages', 'ppv-live-webcams' );
12616                   }
12617
12618                   if ( ! count( $header ) ) {
12619                           return '';
12620                   }
12621
12622                           // tab : profile
12623                           // $addCode .= '<div class="ui ' . $options['interfaceClass'] . ' bottom attached tab segment active" data-tab="profile">';
12624
12625                           $addCode = ' <div class="ui grid">';
12626
12627                   if ( $options['profiles'] ) {
12628
12629                           $active = 'profile';
12630
12631                           if ( $options['videosharevod'] ) {
12632                                   $video_teaser = get_post_meta( $postID, 'video_teaser', true );
12633
12634                                   if ( $video_teaser ) {
12635                                           $addCode .= '<div class="item"><h3 class="ui ' . $options['interfaceClass'] . ' header">' . __( 'Teaser', 'ppv-live-webcams' ) . '</h3> <div class="ui ' . $options['interfaceClass'] . ' segment" style="min-width:320px">' . do_shortcode( '[videowhisper_player video="' . $video_teaser . '"]' . '</div></div>' );
12636                                   }
12637                           }
12638
12639                           if ( $options['profileCall'] ?? true ) $addCode .= '<div class="item"><h3 class="ui ' . $options['interfaceClass'] . ' header"></h3>' . do_shortcode( '[videowhisper_callnow roomid="' . $postID . '"]' ) . '</div>';
12640
12641                           $addCode .= '<div class="item">';
12642                           // ! show viewers
12643
12644                           if ( $options['viewersCount'] || $options['salesCount'] ) {
12645                                   $addCode .= '<h3 class="ui ' . $options['interfaceClass'] . ' header">' . __( 'Meta', 'ppv-live-webcams' ) . '</h3><div class="ui ' . $options['interfaceClass'] . ' segment">';
12646
12647                                   if ( $options['viewersCount'] ) {
12648
```

```
12649 $maxViewers = intval( get_post_meta( $postID, 'maxViewers', true ) );
12650
12651                                                 if ( $maxViewers > 0 ) {
12652                                                         $maxDate
= intval( get_post_meta( $postID, 'maxDate', true ) );
12653                                                         $addCode .= __( 'Maximum viewers', 'ppv-live-
webcams' ) . ': ' . $maxViewers;
12654                                                         if ( $maxDate ) {
12655                                                                 $addCode .= ' on ' . date( 'F j, Y, g:i
a', $maxDate );
12656                                                         }
12657                                                 }
12658
12659                                 }
12660
12661                                 // ! show clients
12662                                 if ( $options['salesCount'] ) {
12663 $clientsP = get_post_meta( $postID, 'paidSessionsPrivate', true );
12664                                         $clientsG = get_post_meta( $postID, 'paidSessionsGroup', true );
12665
12666                                         $addCode .= '<div id="vwPaidSessions">' . __( 'Paid
sessions', 'ppv-live-webcams' ) . ': ' . $clientsP . ' ' . __( 'private', 'ppv-live-webcams' ) . ',
' . $clientsG . ' ' . __( 'group', 'ppv-live-webcams' ) . '. ' . __( 'Logged days', 'ppv-live-
webcams' ) . ': ' . round( $options['ppvKeepLogs'] / 3600 / 24, 2 ) . ' </div>';
12667                                 }
12668
12669                                 $addCode .= '</div>';
12670                         }
12671
12672                         // ! show profile
12673                         $profileCode = self::webcamProfile( $postID, $options );
12674                         if ( $profileCode ) {
12675                                 $addCode .= '<h3 class="ui ' . $options['interfaceClass'] . '
header">' . __( 'Profile Info', 'ppv-live-webcams' ) . '</h3>' . $profileCode;
12676                         }
12677
12678                         //description
12679                         if ( $content ) {
12680                                 $addCode .= '<h3 class="ui ' . $options['interfaceClass'] . '
header">' . __( 'Description', 'ppv-live-webcams' ) . '</h3><div class="ui
' . $options['interfaceClass'] . ' segment">' . $content . '</div>';
12681                         }
12682
12683                         $addCode .= '</div></div>';
12684
12685                         $contents['profile'] = $addCode;
12686                 }
12687
12688                 // ! show videos
12689                 if ( $options['videosharevod'] ) {
12690
12691                         if ( ! $active ) {
12692                                 $active = 'videos';
12693                         }
12694
12695                         $addCode = '';
12696                         // tab : videos
12697                         // $addCode .= '<div class="ui ' . $options['interfaceClass'] . ' bottom attached
tab segment" data-tab="videos">';
12698
12699                         $vw_videos = get_post_meta( $postID, 'vw_videos', true );
12700
```

```
Line
12701                         if ( $vw_videos ) {
12702                                 if ( shortcode_exists( 'videowhisper_videos' ) ) {
12703                                         $addCode .= '<h3 class="header">' . __( 'Videos', 'ppv-live-webcams' ) . '</h3>' . do_shortcode( '[videowhisper_videos playlist="' . $room . '" include_css="1"]' );
12704                                 } else {
12705                                         $addCode .= 'Warning: shortcodes missing. Plugin <a href="https://videosharevod.com">Video Share VOD</A> should be installed and enabled or feature disabled.';
12706                                 }
12707                         }
12708
12709                                 // $addCode .= '</div>';
12710
12711                                 $contents['videos'] = $addCode;
12712                 }
12713
12714                 // ! show pictures
12715                 if ( $options['picturegallery'] ) {
12716                         $addCode = '';
12717                         // tab : pictures
12718                         // $addCode .= '<div class="ui ' . $options['interfaceClass'] . ' bottom attached tab segment" data-tab="pictures">';
12719
12720                         $vw_pictures = get_post_meta( $postID, 'vw_pictures', true );
12721                         if ( $vw_pictures ) {
12722                                 if ( shortcode_exists( 'videowhisper_pictures' ) ) {
12723                                         $addCode .= '<h3 class="ui header">' . __( 'Pictures', 'ppv-live-webcams' ) . '</h3>' . do_shortcode( '[videowhisper_pictures perpage="12" gallery="' . $room . '" include_css="1"]' );
12724                                 } else {
12725                                         $addCode .= 'Warning: shortcodes missing. Plugin <a href="https://wordpress.org/plugins/picture-gallery/">Picture Gallery</A> should be installed and enabled or feature disabled.';
12726                                 }
12727                         }
12728
12729                                 // $addCode .= '</div>';
12730                                 $contents['pictures'] = $addCode;
12731
12732                 }
12733
12734                 // ! show reviews
12735                 if ( $options['rateStarReview'] ) {
12736                         $addCode = '';
12737
12738                         // tab : reviews
12739                         // $addCode .= '<div class="ui ' . $options['interfaceClass'] . ' bottom attached tab segment" data-tab="reviews">';
12740
12741                         if ( shortcode_exists( 'videowhisper_review' ) ) {
12742                                 $addCode .= '<h3 class="ui header">' . __( 'My Review', 'ppv-live-webcams' ) . '</h3>' . do_shortcode( '[videowhisper_review content_type="webcam" post_id="' . $postID . '" content_id="' . $postID . '"]' );
12743                         } else {
12744                                 $addCode .= 'Warning: shortcodes missing. Plugin <a target="_plugin" href="https://wordpress.org/plugins/rate-star-review/">Rate Star Review</a> should be installed and enabled or feature disabled.';
12745                         }
12746
12747                         if ( shortcode_exists( 'videowhisper_reviews' ) ) {
12748                                 $addCode .= '<h3 class="ui header">' . __( 'Reviews', 'ppv-live-webcams' ) . '</h3>' . do_shortcode( '[videowhisper_reviews post_id="' . $postID . '"]' );
12749                         }
12750
12751                         // $addCode .= '</div>';
```

```
12752                         $contents['reviews'] = $addCode;
12753
12754                 }
12755
12756                 // content
12757
12758                 if ( $options['micropaymentsAssets'] ) {
12759                         if ( shortcode_exists( 'videowhisper_content_list' ) ) {
12760                                 if ( count( $contentIDs ) ) {
12761
12762                                         $addCode = '';
12763
12764                                         $addCode .= '<h3 class="ui header">' . __( 'Room Content', 'ppv-live-webcams' ) . '</h3>' . do_shortcode( '[videowhisper_content_list ids="-1,' . implode( ',', $contentIDs ) . '"]' );
12765
12766                                         // $addCode .= '</div>';
12767                                         $contents['content'] = $addCode;
12768
12769                                 }
12770                         }
12771                 }
12772
12773                 // products
12774
12775                 if ( $options['woocommerce'] ) {
12776                         if ( shortcode_exists( 'products' ) ) {
12777                                 if ( count( $productIDs ) ) {
12778
12779                                         $addCode = '';
12780
12781                                         $addCode .= '<h3 class="ui header">' . __( 'Room Products', 'ppv-live-webcams' ) . '</h3>' . do_shortcode( '[products ids="' . implode( ',', $productIDs ) . '"]' );
12782
12783                                         // $addCode .= '</div>';
12784                                         $contents['products'] = $addCode;
12785                                 }
12786                         }
12787                 }
12788
12789                         // ! show messages
12790
12791                 if ( $options['messages'] ) {
12792
12793                         if ( ! $active ) {
12794                                 $active = 'messages';
12795                         }
12796
12797                         $addCode = '';
12798
12799                         // tab : messages
12800                         // $addCode .= '<div class="ui ' . $options['interfaceClass'] . ' bottom attached tab segment" data-tab="messages">';
12801
12802                         $addCode .= self::webcamMessages( $postID, $options, $room );
12803
12804                         // $addCode .= '</div>';
12805                         $contents['messages'] = $addCode;
12806
12807                 }
12808
12809 return self::sectionsLayout( $header, $contents, $active, $options['profileLayout'], $room );
```

```
12810
12811                   // return $addCode;
12812           }
12813
12814
12815           static function webcamMessages( $postID, $options = null, $name = '' ) {
12816                   if ( ! $options ) {
12817                           $options = get_option( 'VWliveWebcamsOptions' );
12818                   }
12819                   $addCode = '';
12820
12821                   $addCode .= '<h3 class="ui header">' . ( $name ? $name . ': ' : '' ) . __( 'Send Question or Message', 'ppv-live-webcams' ) . '</h3>' . do_shortcode( '[videowhisper_cam_message post_id="' . $postID . '"]' );
12822
12823                   $addCode .= '<h3 class="ui header">' . ( $name ? $name . ': ' : '' ) . __( 'My Previous Questions and Messages', 'ppv-live-webcams' ) . '</h3>' . do_shortcode( '[videowhisper_cam_messages post_id="' . $postID . '"]' );
12824                   return $addCode;
12825           }
12826
12827
12828           static function webcamProfile( $postID, $options ) {
12829                   if ( ! $options ) {
12830                           $options = self::getOptions();
12831                   }
12832
12833                   // allowed tags
12834                   $allowedtags = array(
12835                           'a'          => array(
12836                                   'href'  => true,
12837                                   'title' => true,
12838                           ),
12839                           'abbr'       => array(
12840                                   'title' => true,
12841                           ),
12842                           'acronym'    => array(
12843                                   'title' => true,
12844                           ),
12845                           'b'          => array(),
12846                           'blockquote' => array(
12847                                   'cite' => true,
12848                           ),
12849                           'cite'       => array(),
12850                           'code'       => array(),
12851                           'del'        => array(
12852                                   'datetime' => true,
12853                           ),
12854                           'em'         => array(),
12855                           'i'          => array(),
12856                           'q'          => array(
12857                                   'cite' => true,
12858                           ),
12859                           'strike'     => array(),
12860                           'strong'     => array(),
12861
12862                           'ul'         => array(),
12863                           'ol'         => array(),
12864                           'li'         => array(),
12865
12866                           'span'       => array(
12867                                   'style' => array(),
```

```
12868                         ),
12869
12870                         'p'          => array(
12871                                 'style' => array(),
12872                         ),
12873                 );
12874
12875                 $profileCode = '<div>';
12876
12877                 if ( is_array( $options['profileFields'] ) ) {
12878                         foreach ( $options['profileFields'] as $field => $parameters ) {
12879                                 $fieldName = sanitize_title( trim( $field ) );
12880
12881
12882                                 // retrieve value
12883                                 $fieldValue = get_post_meta( $postID, 'vwf_' . $fieldName, true );
12884
12885                                 //$profileCode .= "* $fieldName=$fieldValue | ";
12886
12887 if ( in_array( $parameters['type'], array('checkboxes', 'multiselect', 'language') ) )
12888                                  {
12889                                         if ( ! $fieldValue ) {
12890                                                 $fieldValue = array();
12891                                         }
12892                                         if ( ! is_array( $fieldValue ) ) {
12893                                                 $fieldValue = array( $fieldValue );
12894                                         }
12895                                 } else {
12896 if (is_string($fieldValue)) $fieldValue = wp_kses( $fieldValue, $allowedtags );
12897                                         else $profileCode .= '* Warning: Invalid field value for
' . $field . '';
12898                                 }
12899
12900                                 if ( $fieldValue ) {
12901                                         if ( ! is_array( $fieldValue ) ) {
12902                                                                                 $profileCode .= '<div class="ui
' . $options['interfaceClass'] . ' segment">
12903                                                 <h4 class="ui ' . $options['interfaceClass'] . '
header">' . $field . '</h4>' . $fieldValue . '</div>';
12904                                         } else {
12905
12906
12907                                                 $fieldValues = '';
12908                                         //      $fieldValues .= 'vwf_' . $fieldName . '=' . json_encode(
$fieldValue );
12909
12910                                                 foreach ( $fieldValue as $fieldItem ) {
12911                                                         $fieldValues .= ( $fieldValues ? ',
' : '' ) . $fieldItem;
12912                                                 }
12913
12914                                                 $profileCode .= '<div class="ui
' . $options['interfaceClass'] . ' segment">
12915                                                 <h4 class="ui ' . $options['interfaceClass'] . '
header">' . $field . '</h4>' . $fieldValues . '</div>';
12916                                         }
12917                                 }
12918                         }
12919                 }
12920                 $profileCode .= '</div>';
12921                 return $profileCode;
12922
```

```
12923         }
12924
12925
12926         static function videowhisper_videochat( $atts ) {
12927                 // Shortcode: shows videochat interface depending on mode, device
12928
12929                 $stream = ''; // room name
12930
12931                 $options = self::getOptions();
12932                 // username used with application
12933                 /*
12934                         $userName =  $options['userName'];
12935                         if (!$userName) $userName='user_nicename';
12936                         global $current_user;
12937                         get_currentuserinfo();
12938                         if ($current_user->$userName) $username=sanitize_file_name($current_user->$userName);
12939                         $postID = 0;
12940                         */
12941
12942                 // 1. webcam post
12943                 $postID = 0;
12944                 $room = '';
12945
12946                 if ( is_single() ) {
12947                         if ( get_post_type( $postID = get_the_ID() ) == $options['custom_post'] ) {
12948                                 $room   = self::post_title( $postID );
12949                                 $stream = $room;
12950                         } else {
12951                                 $postID = 0; // post or page?
12952                         }
12953                 }
12954
12955                 $atts = shortcode_atts(
12956                         array(
12957                                 'room'      => $stream,
12958                                 'html5'     => '0',
12959                                 'webcam_id' => $postID,
12960                         ),
12961                         $atts,
12962                         'videowhisper_videochat'
12963                 );
12964                 $atts = array_map('esc_attr', $atts);
12965
12966                 // 2. shortcode param
12967                 // $stream = get_post_meta($postID, 'performer', true);
12968                 if ( !$room ) {
12969                         $room = $atts['room'];
12970                 }
12971
12972                 $room   = sanitize_file_name( $room );
12973                 if (!$postID) $postID = $atts['webcam_id'];
12974
12975                 global $wpdb;
12976                 if ( ! $room && $postID ) {
12977                         $room = $wpdb->get_var( $wpdb->prepare( "SELECT post_title FROM $wpdb->posts WHERE ID = %d and post_type=%s LIMIT 0,1", $postID, $options['custom_post'] ) );
12978                 }
12979
12980                 if ( ! $room ) {
12981                         return "<div class='ui segment red'>Can't load application: Missing room name!</div>";
```

```
12982                   }
12983
12984                   if ( $room && ! $postID ) {
12985                           $postID = $wpdb->get_var( $wpdb->prepare( "SELECT ID FROM $wpdb->posts WHERE
post_title = %s and post_type=%s LIMIT 0,1", sanitize_file_name( $room ), $options['custom_post'] ) );
12986                   }
12987
12988                   $ztime = time();
12989
12990                   // update access time, if anybody accessed including visitors, 20s precision
12991                   $accessed = intval( get_post_meta( $postID, 'accessed', true ) );
12992                   if ( $ztime - $accessed > 20 )update_post_meta( $postID, 'accessed', $ztime );
12993
12994                   // login
12995                   if ( is_user_logged_in() ) {
12996                           $current_user = wp_get_current_user();
12997                           $uid          = $current_user->ID;
12998                           $balance = self::balance( $current_user->ID );
12999
13000                           $post      = get_post( $postID );
13001                           $performer = 0;
13002                           if ( self::isRoomPerformer( $post, $current_user ) ) {
13003                                   $performer = 1;
13004                           }
13005
13006                           $accessedUser = intval( get_post_meta( $postID, 'accessedUser', true ) );
13007                           if ( $ztime -
 $accessedUser > 10 ) update_post_meta( $postID, 'accessedUser', $ztime );
13008
13009
13010                   } else {
13011                           $uid       = 0;
13012                           $performer = 0;
13013                           $balance = 0;
13014
13015                           // paid room requires login
13016                           $groupCPM = get_post_meta( $postID, 'groupCPM', true );
13017                           if ( $groupCPM ) {
13018                                   return '<div class="ui segment red"><div class="ui header"><i
class="ticket icon"></i>' .$room. '</div>Only registered and logged in users can access rooms in paid
mode. Requires: ' . $groupCPM .' ' . $options['currencypm']. '<BR><a class="ui button primary qbutton"
href="' . wp_login_url() . '">' . __( 'Login', 'ppv-live-webcams' ) . '</a> <a class="ui button
secondary qbutton" href="' . wp_registration_url() . '">' . __( 'Register', 'ppv-live-webcams' ) . '</a>
</div>';
13019                           }
13020                   }
13021
13022                   $attsPrivate = '';
13023                   if ( ! $performer ) {
13024                           // free mode limits: check before rendering videochat room
13025                           global $wpdb;
13026                           $table_sessions = $wpdb->prefix . 'vw_vmls_sessions';
13027
13028                           if ( ! isset( $clientIP ) ) {
13029                                   $clientIP = self::get_ip_address();
13030                           }
13031
13032                           if ( $uid ) {
13033                                   $cnd = "uid='$uid'";
13034                           } else {
13035                                   $cnd = "ip='$clientIP' AND uid='0'";
13036                           }
13037
```

```
13038                         $h24      = time() - 86400;
13039                         $sqlC     = $wpdb->prepare("SELECT SUM(edate-sdate) FROM `$table_sessions` WHERE $cnd AND broadcaster='0' AND sdate > %d AND edate > %d", $h24, $h24);
13040                         $freeTime = $wpdb->get_var( $sqlC );
13041
13042                         if ( $uid ) {
13043 if ( $freeTime > $options['freeTimeLimit'] && $options['freeTimeLimit'] > 0 ) {
13044
13045                                         if ( !$options['freeTimeBalance'] ) $disconnect = __( 'Free chat daily time limit reached: You can only access paid group rooms today.', 'ppv-live-webcams' ) . ' (' . $freeTime . 's > ' . $options['freeTimeLimit'] . 's ' . $cnd . ')';
13046 elseif ( $balance < $options['freeTimeBalance'] ) $disconnect = __( 'Free chat daily time limit reached. You get unlimited free chat time if you add a minimum balance.', 'ppv-live-webcams' ) . ' (' . $options['freeTimeBalance'] . $options['currency'] . ')';
13047
13048                                 }
13049                         }
13050                         if ( ! $uid ) {
13051 if ( $freeTime > $options['freeTimeLimitVisitor'] && $options['freeTimeLimitVisitor'] > 0 ) {
13052                                         $disconnect = __( 'Free chat daily visitor time limit reached: Register and login for more chat time today!', 'ppv-live-webcams' ) . ' (' . $freeTime . 's > ' . $options['freeTimeLimitVisitor'] . 's ' . $cnd . ')';
13053                                 }
13054                         }
13055
13056                         if ( isset( $disconnect ) ) {
13057                                 return "<div class='ui segment red'>$disconnect</div>";
13058                         }
13059
13060                                         // buddypress groups
13061 $buddypressGroup = get_post_meta( $postID, 'buddypressGroup', true );
13062                         if ( $buddypressGroup ) {
13063                                 if ( function_exists( 'groups_is_user_member' ) ) {
13064                                         if ( ! groups_is_user_member( $uid, $buddypressGroup ) ) {
13065                                                 return '<div class="ui segment red">' . __( 'This is a BuddyPress group room and you are not a member!', 'ppv-live-webcams' ) . '</div>';
13066                                         }
13067                                 }
13068                         }
13069                 }
13070
13071                 $htmlCode = '';
13072
13073                 // flash/webrtc room interface
13074                 $roomInterface  = get_post_meta( $postID, 'roomInterface', true ); // flash/html5/html5app
13075                 $playlistActive = get_post_meta( $postID, 'vw_playlistActive', true );
13076
13077                 // special modes (for clients)
13078                 $isVoyeur  = 0;
13079                 $is2way    = 0;
13080                 $isPrivate = 0;
13081
13082                 $specialMode = sanitize_file_name( $_GET['vwsm'] ?? '' );
13083
13084                 if ( $specialMode ) {
13085                         if ( is_user_logged_in() ) {
13086
13087                                 $groupParameters = get_post_meta( $postID, 'groupParameters', true );
13088                                 // var_dump($groupParameters);
13089
```

```
13090                                   // enable mode list for room and capabilities for user:
13091
13092                                   switch ( $specialMode ) {
13093
13094                                           case 'private':
13095                                                   $isPrivate = 1;
13096                                                   $calltype
= sanitize_file_name( $_GET['calltype'] ?? '' );
13097
13098                                                   //downgrade if room in lesser mode
13099                                                   
$room_audio = self::is_true( get_post_meta( $postID, 'room_audio', true ) );
13100                                                   
$room_text = self::is_true( get_post_meta( $postID, 'room_text', true ) );
13101                                                   
if ($room_audio) if ($calltype != 'text') $calltype = 'audio';
13102                                                   if ($room_text) $calltype = 'text';
13103
13104                                                   if ( ! $attsPrivate ) {
13105                                                           $attsPrivate = 'private="1"
type="' . $calltype . '"';
13106                                                   }
13107
13108                                                   //notification to performer
13109                                                   if ($uid && !$performer)
13110                                                   {
13111                                                   
$performerID = intval( get_post_meta( $postID, 'performerUserID', true ) );
13112                                                   
if ($options['privateNotification']) if ($performerID) self::sendNotification($performerID, $options['pri
vateSubject'] . ' ' . $current_user-
>display_name , $options['privateText'], get_permalink( $postID ), $options);
13113                                                   }
13114
13115                                                   break;
13116
13117                                           case '2way':
13118                                                   $mode2way = get_post_meta( $postID, 'mode2way', true );
13119
13120                                                   if ( ! is_array( $mode2way ) ) {
13121                                                           $mode2way = array();
13122                                                   }
13123
13124                                                   // add user if slots available
13125                                                   if ( count( $mode2way ) < $groupParameters['2way'] ) {
13126                                                           $mode2way[ $uid ] = time();
13127                                                   
update_post_meta( $postID, 'mode2way', $mode2way );
13128                                                           $is2way = 1;
13129                                                   }
13130                                                   break;
13131
13132                                           case 'voyeur':
13133                                                   
$modeVoyeur = get_post_meta( $postID, 'modeVoyeur', true );
13134                                                   if ( ! is_array( $modeVoyeur ) ) {
13135                                                           $modeVoyeur = array();
13136                                                   }
13137
13138                                                   if ( $groupParameters['voyeur'] ) {
13139                                                           $modeVoyeur[ $uid ] = time();
13140                                                   
update_post_meta( $postID, 'modeVoyeur', $modeVoyeur );
13141                                                           $isVoyeur = 1;
```

```
13142                                                   }
13143                                                   break;
13144                                           }
13145                                   } else {
13146                                           return '<div class="ui ' . $options['interfaceClass'] . ' header"> <i class="sign in icon"></i> ' . __( 'Login to access special chat mode!', 'ppv-live-webcams' ) . '</div>' . do_shortcode( '[videowhisper_login_form register="1" role="' . $options['roleClient'] . '"]' );
13147                                   }
13148                           }
13149
13150                           $isCall = 0;
13151                           if ( $_GET['call'] ?? false ) {
13152                                   $isCall = 1;
13153                           }
13154
13155                           if ( ! $performer ) {
13156                                   $group_disabled = self::is_true( get_post_meta( $postID, 'group_disabled', true ) );
13157                                   if ( $group_disabled ) {
13158                                           if ( ! $isPrivate && ! $isVoyeur && ! $isCall ) {
13159                                                   return '<div class="ui segment red">' . __( 'Group chat is disabled. This room can only be used for instant or locked calls.', 'ppv-live-webcams' ) . '</div>';
13160                                           }
13161                                   }
13162                           }
13163
13164
13165                           // HLS if iOS/Android detected
13166                           $agent   = $_SERVER['HTTP_USER_AGENT'];
13167                           $Android = strstr( $agent, 'Android' );
13168                           $iOS     = ( strstr( $agent, 'iPhone' ) || strstr( $agent, 'iPod' ) || strstr( $agent, 'iPad' ) );
13169                           $Safari  = ( strstr( $agent, 'Safari' ) && ! strstr( $agent, 'Chrome' ) );
13170
13171                           $htmlCode .= "<!-- VideoChat:$roomInterface room:$room postID:$postID Performer:$performer SpecialMode:$specialMode 2Way:$is2way Voyeur:$isVoyeur Device:$Android|$iOS|$Safari($agent) HTML5:-->";
13172
13173                           // handle paused reStreams
13174                                   $restreamPaused = false;
13175                                   if ( $options['reStreams'] )
13176                                   {
13177                                           $reStreams = get_post_meta( $postID, 'reStreams', true );
13178                                           if ( $reStreams ) {
13179                                                   $restreamPaused = get_post_meta( $postID, 'restreamPaused', true );
13180                                                   $htmlCode .= self::restreamPause( $postID, $stream, $options );
13181                                           }
13182                                   }
13183
13184                           $fallback = 0;
13185
13186                           {
13187                                   if ( $roomInterface == 'webcamapp' ) {
13188                                           if ( ! empty( $options['webcamAppEnabled'] ) ) {
13189                                                   $webcamView = $performer ? 'BroadcastChat' : 'PlayChat';
13190                                                   $htmlCode .= do_shortcode( '[videowhisper_webcam_app room="' . $room . '" view="' . $webcamView . '"]' );
13191                                                   $htmlCode .= apply_filters( 'vw_plw_videochat', '', $postID );
13192                                                   return $htmlCode;
13193                                           }
13194
13195                                           // Preserve compatibility when webcamapp is configured but disabled.
```

```
13196                                   $roomInterface = 'html5app';
13197                           }
13198
13199                           if ( $options['webrtc'] > 2 ) {
13200                                   if ( isset($_GET['html5app']) || $roomInterface == 'html5app' ||
 $roomInterface == 'flash' || ! self::webcamOnline( $postID ) ) {
13201                                           $htmlCode .= do_shortcode( "[videowhisper_cam_app
room=\"$room\" webcam_id=\"$postID\" $attsPrivate]" );
13202                                           $htmlCode .= apply_filters( 'vw_plw_videochat', '', $postID );
13203                                           return $htmlCode;
13204                                   }
13205                           }
13206
13207                           // performer
13208                           if ( $performer ) {
13209                                   if ( $roomInterface == 'html5' || $Android || $iOS ||
 ( $_GET['html5'] && $options['htmlchatTest'] ) || $atts['html5'] || $roomInterface == 'flash' ) {
13210                                           $htmlCode .= do_shortcode( "[videowhisper_cam_webrtc_broadcast
room=\"$room\" webcam_id=\"$postID\"]" );
13211                                           $htmlCode .= apply_filters( 'vw_plw_videochat', '', $postID );
13212                                           return $htmlCode;
13213                                   }
13214                           } else // client
13215                                   {
13216
13217                                   if ( $options['teaserOffline'] && ! self::webcamOnline( $postID ) ) {
13218                                           $showHTML5  = 1;
13219                                           $comment   .= ' Teaser';
13220                                           $fallback   = 1;
13221                                           $showTeaser = 1;
13222                                   }
13223
13224                                   // client in webrtc room
13225                                   if ( $roomInterface == 'html5' && ! $showTeaser ) {
13226                                           if ( ! $isVoyeur ) {
13227                                                   $htmlCode .= do_shortcode( "[videowhisper_htmlchat
room=\"$room\" post_id=\"$postID\" width=\"$width\" height=\"$height\"]" );
13228                                           } else {
13229                                                   $htmlCode .= do_shortcode( "
[videowhisper_cam_webrtc_playback
room=\"$room\" webcam_id=\"$postID\" width=\"$width\" height=\"$height\"]" );
13230                                           }
13231
13232                                           $htmlCode .= apply_filters( 'vw_plw_videochat', '', $postID );
13233                                           return $htmlCode;
13234                                   }
13235
13236                                   if ( $options['transcoding'] >= 3 ) {
13237                                           if ( $playlistActive && ! $performer ) {
13238                                                   $showHTML5 = 1;
13239                                                   $comment  .= ' Playlist';
13240                                                   $fallback  = 1;
13241                                           }
13242                                   }
13243
13244                                   // detection for client videochat interface
13245
if ( ( $Android && in_array( $options['detect_mpeg'], array( 'android', 'all' ) ) ) ||
 ( ! $iOS && in_array( $options['detect_mpeg'], array( 'all' ) ) ) ||
 ( ! $iOS && ! $Safari && in_array( $options['detect_mpeg'], array( 'nonsafari' ) ) ) ) {
13246                                           $showHTML5 = 1;
13247                                           $fallback  = 1;
13248                                           $comment  .= ' detectMPEG';
13249                                   }
```

```
13250
13251                                   /*
13252                                           if ($options['htmlchat'] && !$isVoyeur)
13253                                           {
13254                                                   $htmlCode .= do_shortcode("[videowhisper_htmlchat room=\"$room\" post_id=\"$postID\" width=\"$width\" height=\"$height\"]");
13255                                                   $fallback = 1;
13256                                           }
13257                                   else
13258                                   {
13259                                           $htmlCode .= do_shortcode("[videowhisper_cammpeg webcam=\"$room\" width=\"$width\" height=\"$height\" webstatus=\"1\"]");
13260                                           $fallback = 2;
13261                                   }
13262                                   */
13263
13264                                   if ( ( ( $Android || $iOS ) && in_array( $options['detect_hls'], array( 'mobile', 'safari', 'all' ) ) ) || ( $iOS && $options['detect_hls'] == 'ios' ) || ( $Safari && in_array( $options['detect_hls'], array( 'safari', 'all' ) ) ) ) {
13265                                           $showHTML5 = 1;
13266                                           $fallback  = 1;
13267                                           $comment  .= ' detectHLS';
13268                                   }
13269                                   /*
13270                                           if ($options['htmlchat'] && !$isVoyeur)
13271                                           {
13272                                                   $htmlCode .=  do_shortcode("[videowhisper_htmlchat room=\"$room\" post_id=\"$postID\" width=\"$width\" height=\"$height\" ]");
13273                                                   $fallback = 1;
13274                                           }
13275                                   else
13276                                   {
13277                                           $htmlCode .=  do_shortcode("[videowhisper_camhls webcam=\"$room\" width=\"$width\" height=\"$height\" webstatus=\"1\"]");
13278                                           $fallback = 3;
13279                                   }
13280                                   */
13281
13282                                   if ( $showHTML5 ) {
13283                                           $htmlCode .= '<!--$showHTML5: ' . $comment . ' -->';
13284                                           if ( ! $isVoyeur ) {
13285                                                   $htmlCode .= do_shortcode( "[videowhisper_htmlchat room=\"$room\" post_id=\"$postID\" width=\"$width\" height=\"$height\"]" );
13286                                           } else {
13287                                                   $htmlCode .= '<H4>Voyeur Mode</H4>' . do_shortcode( '[videowhisper_camvideo post_id="' . $postID . '" cam="' . $room . '" html5="always"]' );
13288                                           }
13289                                   }
13290                           }
13291
13292                           // forced testing, even when  should show flash
13293                           if ( $options['htmlchatTest'] && ( $_GET['htmlchat'] || $_GET['html5'] ) ) {
13294                                   $htmlCode .= '<!--Test Chat-->' . do_shortcode( "[videowhisper_htmlchat post_id=\"$postID\" room=\"$room\"]" );
13295                                   $fallback  = 1;
13296                           }
13297
13298                           if ( $options['htmlchatTest'] && $_GET['hls'] ) {
13299                                   $htmlCode .= '<!--Test HLS-->' . do_shortcode( "[videowhisper_camhls webcam=\"$room\" width=\"$width\" height=\"$height\"]" );
13300                                   $fallback  = 3;
13301                           }
```

```
13302
13303                   if ( $options['htmlchatTest'] && $_GET['mpeg'] ) {
13304                           $htmlCode .= '<!--Test MPEG-->' . do_shortcode( "[videowhisper_cammpeg webcam=\"$room\" width=\"$width\" height=\"$height\"]" );
13305                           $fallback  = 2;
13306                   }
13307
13308                   if ( $fallback ) {
13309                           $htmlCode .= '<!--HTML5 Videochat End-->';
13310                   }
13311           }
13312
13313           // No interface matched
13314           if ( ! $fallback ) {
13315                   if ( $isVoyeur ) {
13316                           $htmlCode .= '<H4>Voyeur Mode</H4>' . do_shortcode( '[videowhisper_camvideo cam="' . $room . '"]' );
13317                   } else {
13318                           $htmlCode .= '<div class="ui segment"><p>' . __( 'Please configure this room to use HTML5 interface from room settings.', 'ppv-live-webcams' ) . '</p>';
13319                           if ( $performer && $options['webrtc'] >= 2 ) {
13320                                   $htmlCode .= '<a class="ui button secondary" href="' . add_query_arg( array( 'html5app' => '1' ), get_permalink( $postID ) ) . '">' . __( 'Try HTML5 Videochat', 'ppv-live-webcams' ) . '</a>';
13321                           }
13322                           $htmlCode .= '</p></div>';
13323                   }
13324           }
13325
13326           $htmlCode .= apply_filters( 'vw_plw_videochat', '', $postID );
13327
13328           return $htmlCode;
13329   }
13330
13331
13332   static function videowhisper_htmlchat( $atts ) {
13333           // [videowhisper_htmlchat room="webcam name" width="480px" height="360px"]
13334
13335           $atts = shortcode_atts(
13336                   array(
13337                           'room'        => '',
13338                           'post_id'     => '0',
13339                           'videowidth'  => '480px',
13340                           'videoheight' => '360px',
13341                   ),
13342                   $atts,
13343                   'videowhisper_htmlchat'
13344           );
13345           $atts = array_map('esc_attr', $atts);
13346
13347           $room = sanitize_text_field( $atts['room'] );
13348           if ( ! $room ) {
13349                   $room = sanitize_text_field( $_GET['room'] );
13350           }
13351
13352           $room = sanitize_file_name( $room );
13353
13354           $postID = intval( $atts['post_id'] );
13355
13356           $options = get_option( 'VWliveWebcamsOptions' );
13357
13358           if ( ! $postID ) {
13359                   global $wpdb;
```

```
13360
13361                       $postID = $wpdb->get_var( $wpdb->prepare( 'SELECT ID FROM ' . $wpdb->posts . ' WHERE post_title = %s and post_type=%s LIMIT 0,1', $room, $options['custom_post'] ) );
13362
13363               }
13364
13365               $videowidth = $atts['videowidth'];
13366               if ( ! $videowidth ) {
13367                       $videowidth = '480px';
13368               }
13369               $videoheight = $atts['videoheight'];
13370               if ( ! $videoheight ) {
13371                       $videoheight = '360px';
13372               }
13373
13374               if ( ! $room ) {
13375                       return 'HTML AJAX Chat Error: Missing room name!';
13376               }
13377
13378               $isPerformer = self::isAuthor( $postID ); // is current user performer?
13379
13380               self::enqueueUI();
13381
13382               // htmlchat ui
13383               // css
13384 wp_enqueue_style( 'jScrollPane', dirname( plugin_dir_url( __FILE__ ) ) . '/htmlchat/js/jScrollPane/jScrollPane.css' );
13385 wp_enqueue_style( 'htmlchat', dirname( plugin_dir_url( __FILE__ ) ) . '/htmlchat/css/chat-watch.css' );
13386
13387               // js
13388               wp_enqueue_script( 'jquery' );
13389               wp_enqueue_script( 'jScrollPane-mousewheel', dirname( plugin_dir_url( __FILE__ ) ) . '/htmlchat/js/jScrollPane/jquery.mousewheel.js' );
13390 wp_enqueue_script( 'jScrollPane', dirname( plugin_dir_url( __FILE__ ) ) . '/htmlchat/js/jScrollPane/jScrollPane.min.js' );
13391 wp_enqueue_script( 'htmlchat', dirname( plugin_dir_url( __FILE__ ) ) . '/htmlchat/js/script.js', array( 'jquery', 'jScrollPane' ) );
13392
13393               $ajaxurl = admin_url() . 'admin-ajax.php?action=vmls_htmlchat&room=' . urlencode( $room );
13394
13395               $videoCode = do_shortcode( '[videowhisper_camvideo cam="' . $room . '" post_id="' . $postID . '" html5="always"]' );
13396
13397               $loginCode = '<a class="ui button" href="' . wp_login_url() . '">Login is required to chat!</a>';
13398
13399 $buttonSFx = dirname( plugin_dir_url( __FILE__ ) ) . '/videowhisper/templates/messenger/message.mp3';
13400               $tipsSFx = dirname( plugin_dir_url( __FILE__ ) ) . '/videowhisper/templates/messenger/tips/';
13401
13402               if ( $options['tips'] ) {
13403
13404                       // tip options
13405                       $tipOptions = stripslashes( $options['tipOptions'] );
13406                       if ( $tipOptions ) {
13407                               $p = xml_parser_create();
13408                               xml_parse_into_struct( $p, trim( $tipOptions ), $vals, $index );
13409                               $error = xml_get_error_code( $p );
13410                               xml_parser_free( $p );
```

```
13411
13412                                   if ( is_array( $vals ) ) {
13413                                           foreach ( $vals as $tKey => $tip ) {
13414                                                   if ( $tip['tag'] == 'TIP' ) {
13415                                                           if ( $tip['attributes']['AMOUNT'] != 'custom' ) {
13416                                                                   // var_dump($tip['attributes']);
13417                                                                   $amount = intval( $tip['attributes']
['AMOUNT'] );
13418                                                                   if ( ! $amount ) {
13419                                                                           $amount = 1;
13420                                                                   }
13421                                                                   $label = $tip['attributes']['LABEL'];
13422                                                                   if ( ! $label ) {
13423                                                                           $label = '$1 Tip';
13424                                                                   }
13425                                                                   $note = $tip['attributes']['NOTE'];
13426                                                                   if ( ! $note ) {
13427                                                                           $label = 'Tip';
13428                                                                   }
13429                                                                   $sound = $tip['attributes']['SOUND'];
13430                                                                   if ( ! $sound ) {
13431                                                                           $sound = 'coins1.mp3';
13432                                                                   }
13433                                                                   $image = $tip['attributes']['IMAGE'];
13434                                                                   if ( ! $image ) {
13435                                                                           $image = 'gift1.png';
13436                                                                   }
13437
13438                                                                   $imageURL = $tipsSFx . $image;
13439
13440                                                                   $tipbuttonCodes .= <<<TBCODE
13441         <div class="tipButton ui labeled button small" tabindex="0" amount="$amount" label="$label"
note="$note" sound="$sound" image="$image">
13442   <div class="ui button">
13443     <img class="mini image avatar" src="$imageURL"> $label
13444   </div>
13445   <a class="ui basic label small">
13446     $amount
13447   </a>
13448 </div>
13449 TBCODE;
13450                                                           }
13451                                                   }
13452                                           }
13453                                   }
13454                           }
13455
13456                                   $balanceURL = '#';
13457                           if ( $options['balancePage'] ) {
13458                                   $balanceURL = get_permalink( $options['balancePage'] );
13459                           }
13460
13461                           $tipsCode = <<<TIPSCODE
13462 <div id="tips" class="ui segment form">
13463 <div class="inline fields">
13464
13465 <a href="$balanceURL" target="_balance" class="ui label olive large">
13466   <i class="money bill alternate icon large"></i>Balance: <span id="balanceAmount" class="inline"> -
</span>
13467 </a>
13468
13469 $tipbuttonCodes
```

```
13470 </div>
13471 </div>
13472 TIPSCODE;
13473                 }
13474 
13475                 $htmlCode = <<<HTMLCODE
13476 <div id="videochatContainer">
13477 <!--$room-->
13478 <div id="streamContainer">
13479 $videoCode
13480 </div>
13481 
13482 <div id="chatContainer">
13483     <div id="chatUsers" class="ui segment"></div>
13484 
13485     <div id="chatLineHolder"></div>
13486 
13487     <div id="chatBottomBar" class="ui segment">
13488 
13489         <div class="tip"></div>
13490 
13491         <form id="loginForm" method="post" action="" class="ui form">
13492 $loginCode
13493                 </form>
13494 
13495         <form id="submitForm" method="post" action="" class="ui form">
13496             <input id="chatText" name="chatText" class="rounded" maxlength="255" />
13497             <input id="submit" type="submit" class="ui button" value="Submit" />
13498         </form>
13499 
13500     </div>
13501 
13502 </div>
13503 </div>
13504 $tipsCode
13505 
13506 <script>
13507 var vwChatAjax= '$ajaxurl';
13508 var vwChatButtonSFx =  '$buttonSFx';
13509 var vwChatTipsSFx =  '$tipsSFx';
13510 
13511 var \$jQ = jQuery.noConflict();
13512 \$jQ(document).ready(function(){
13513 \$jQ('.tipButton').popup();
13514 });
13515 </script>
13516 
13517 HTMLCODE;
13518 
13519                 return $htmlCode;
13520 
13521         }
13522 
13523 
13524 
13525 }
```

About
Showcase
Learn
Get Involved
News
Themes
Documentation
Events
Hosting
Plugins
Developers
Donate ↗
Privacy
Patterns
WordPress.tv ↗
Five for the Future

WordPress.com ↗
Matt ↗
bbPress ↗
BuddyPress ↗

CODE IS POETRY
The WordPress® trademark is the intellectual property of the WordPress Foundation.