

```
1  <?php
2  /*****
3   * Author : Philippe Ku
4   * School / Class : CFPT Informatique / I.FA-P3B
5   * Date : 15.06.2016
6   * Programm : Event gestion website
7   * File : mysql.inc.php
8   * Description : This file contains the constans for the database connection
9   * Version : 1.10
10  *****/
11
12  define('DB_NAME', 'tpi');
13  define('DB_HOST', 'localhost');
14  define('DB_PASSWORD', 'TPI2016');
15  define('DB_USER', 'TPI2016');
16
```

```

1  <?php
2
3  /*****
4   * Author : Philippe Ku
5   * School / Class : CFPT Informatique / I.FA-P3B
6   * Date : 15.06.2016
7   * Programm : Event gestion website
8   * File : functions.php
9   * Description : This file is the model and contains all the functions of my website
10  * Version : 1.10
11  * *****/
12
13  //----- DATABASE FUNCTIONS -----\\
14
15  require 'mysql.inc.php';
16
17  /**
18   * Function that creates a database connector or returns it.
19   * @staticvar type $dbc // Static connector to the database.
20   * @return \PDO // The PDO object (PHP data object).
21   */
22  function myDatabase() {
23      static $dbc = null;
24
25      if ($dbc == null) {
26          try {
27              $dbc = new PDO('mysql:host=' . DB_HOST . ';dbname=' . DB_NAME, DB_USER,
28                  DB_PASSWORD, array(PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8",
29                      PDO::ATTR_PERSISTENT => true));
30          } catch (Exception $e) {
31              echo 'Erreur : ' . $e->getMessage() . '<br />';
32              echo 'N° : ' . $e->getCode();
33              die('Could not connect to MySQL');
34          }
35      }
36      return $dbc;
37  }
38
39  /**
40   * Function that inserts new events into the database.
41   * @param type $datas // The informations that we insert into the database.
42   */
43  function insertNewEvent($datas) {
44      $query1 = "INSERT INTO location (street, city, iso_country) VALUES (:street,
45          :city, :iso_country)";
46      $query2 = "SELECT id_location FROM location WHERE street = :street AND city =
47          :city AND iso_country = :iso_country";
48      $query3 = "INSERT INTO event (title, dateStart, dateEnd, description, image,
49          id_location, id_user) VALUES (:title, :dateStart, :dateEnd, :description,
50          :image, :id_location, :id_user)";
51
52      $ps1 = myDatabase()->prepare($query1);
53      $ps2 = myDatabase()->prepare($query2);
54      $ps3 = myDatabase()->prepare($query3);
55
56      $ps1->bindParam(':street', $datas['event_street'], PDO::PARAM_STR);
57      $ps1->bindParam(':city', $datas['event_city'], PDO::PARAM_STR);

```

```
53     $ps1->bindParam(':iso_country', $datas['event_country'], PDO::PARAM_STR);
54
55     $ps2->bindParam(':street', $datas['event_street'], PDO::PARAM_STR);
56     $ps2->bindParam(':city', $datas['event_city'], PDO::PARAM_STR);
57     $ps2->bindParam(':iso_country', $datas['event_country'], PDO::PARAM_STR);
58
59     $ps1->execute();
60     $isok = $ps2->execute();
61     $isok = $ps2->fetchAll(PDO::FETCH_NUM);
62
63     $ps3->bindParam(':title', $datas['event_title'], PDO::PARAM_STR);
64     $ps3->bindParam(':dateStart', $datas['event_datestart'], PDO::PARAM_STR);
65     $ps3->bindParam(':dateEnd', $datas['event_dateend'], PDO::PARAM_STR);
66     $ps3->bindParam(':description', $datas['event_desc'], PDO::PARAM_STR);
67     $ps3->bindParam(':image', $datas['event_image'], PDO::PARAM_STR);
68     $ps3->bindParam(':id_location', $isok[0][0]);
69     $ps3->bindParam(':id_user', $datas['user_id'], PDO::PARAM_STR);
70
71     $ps3->execute();
72 }
73
74 /**
75  * Function that inserts a new user into the database.
76  * @param type $datas // The informations that we insert into the database.
77  */
78 function insertNewUser($datas) {
79     $query = "INSERT INTO user (email, nickname, password) VALUES (:email,
80         :nickname, :password)";
81
82     $ps = myDatabase()->prepare($query);
83
84     $ps->bindParam(':email', $datas['user_email'], PDO::PARAM_STR);
85     $ps->bindParam(':nickname', $datas['user_nickname'], PDO::PARAM_STR);
86     $ps->bindParam(':password', $datas['user_password'], PDO::PARAM_STR);
87
88     $ps->execute();
89 }
90
91 /**
92  * Function that inserts a new comment into the database.
93  * @param type $datas // The informations that we insert into the database.
94  */
95 function insertNewComment($datas) {
96     $query = "INSERT INTO comment (time, text, id_user, id_event) VALUES (:time,
97         :text, :id_user, :id_event)";
98
99     $ps = myDatabase()->prepare($query);
100
101     $ps->bindParam(':time', $datas['comment_time'], PDO::PARAM_STR);
102     $ps->bindParam(':text', $datas['comment_text'], PDO::PARAM_STR);
103     $ps->bindParam(':id_user', $datas['user_id'], PDO::PARAM_STR);
104     $ps->bindParam(':id_event', $datas['event_id'], PDO::PARAM_STR);
105
106     $ps->execute();
107 }
108
109 /**
```

```

108  * Function that checks if the user is logged, with the correct credentials.
109  * @param type $datas          // The informations that we select from the database.
110  * @return boolean            // Returns TRUE if checked or FALSE if not.
111  */
112  function checkUser($datas) {
113      $query = "SELECT * FROM user where email = :email";
114
115      $ps = myDatabase()->prepare($query);
116      try {
117          $ps->bindParam(':email', $datas['user_email'], PDO::PARAM_STR);
118          $isok = $ps->execute();
119          $isok = $ps->fetchAll(PDO::FETCH_NUM);
120          if (count($isok)) {
121              if ($isok[0][1] == $datas['user_email'] && $isok[0][3] == sha1($datas[
122                  'user_password'])) {
123                  // Stores the user id and email in session.
124                  $_SESSION['user_id'] = $isok[0][0];
125                  $_SESSION['user_email'] = $isok[0][1];
126
127                  $isok = true;
128              } else {
129                  $isok = false;
130              }
131          } catch (PDOException $e) {
132              $isok = false;
133          }
134          return $isok;
135      }
136
137  /**
138   * Function that returns the group of the user.
139   * @param type $datas          // The informations that we select from the database.
140   * @return boolean            // Returns the group of the user.
141   */
142  function getUserGroupFromDatabase($datas) {
143      $query = "SELECT privilege FROM user WHERE email = :email";
144
145      $ps = myDatabase()->prepare($query);
146
147      try {
148          $ps->bindParam(':email', $datas['user_email'], PDO::PARAM_STR);
149          $isok = $ps->execute();
150          $isok = $ps->fetchAll(PDO::FETCH_NUM);
151      } catch (PDOException $e) {
152          $isok = false;
153      }
154      return $isok[0][0];
155  }
156
157  /**
158   * Function that gets the different countries (iso and name) from the database and
159   * put them into an option list.
160   */
161  function getCountriesList() {
162      $query = "SELECT iso, name FROM country ORDER BY name ASC";
163      $qr = myDatabase()->query($query);

```

```

163     $result = $qr->fetchAll();
164     foreach ($result as $row) {
165         $country_iso = $row['iso'];
166         $country_name = $row['name'];
167         echo '<option value="' . $country_iso . '">' . $country_name . '</option>';
168     }
169 }
170
171 /**
172  * Function that displays all the events of the current day with a status unban
173  * (status = 1) order by the the event id.
174  */
175 function getCurrentDayEvents() {
176     $date = date("Y-m-d");
177     $query = "SELECT event.id_event, title, dateStart, dateEnd, image, street, city,
178     iso_country FROM event, location WHERE event.id_location = location.id_location
179     AND event.status = 1 AND dateStart = :day ORDER BY event.id_event";
180     $ps = myDatabase()->prepare($query);
181     $ps->bindParam(':day', $date, PDO::PARAM_STR);
182     $ps->execute();
183     $results = $ps->fetchAll(PDO::FETCH_ASSOC);
184     foreach ($results as $row) {
185         $event_id = $row['id_event'];
186         $event_title = $row['title'];
187         $event_datestart = $row['dateStart'];
188         $event_dateend = $row['dateEnd'];
189         $event_image = $row['image'];
190         $event_street = $row['street'];
191         $event_city = $row['city'];
192         $event_country = $row['iso_country'];
193
194         echo '<div class="col-xs-6 col-lg-4">
195             <a href="event_details.php?event_id=' . $event_id . '"
196             class="thumbnail">
197                 
199             </a>
200             <h3>' . $event_title . '</h3>
201             <p>Date de début : ' . $event_datestart . '</p><p>Date de fin : '
202             . $event_dateend . '</p>
203             <p>Lieu : ' . $event_street . ' ' . $event_city . ' ' .
204             $event_country . '</p>
205             <p><a class="btn btn-primary" href="event_details.php?event_id='
206             . $event_id . '" role="button">Détails &raquo;</a></p>
207         </div>';
208     }
209 }
210
211 /**
212  * Function that displays all the events of the week with a status unban (status =
213  * 1) order by the the event id.
214  */
215 function getCurrentWeekEvents() {
216     // Today day of week
217     $n = date("N");
218     // The first day of the week (Monday).
219     $first = abs($n - 1);

```

```

211 // The last day of the week (Sunday).
212 $last = abs(7 - $n);
213
214 // Format the dates.
215 $sFirst = 'P' . $first . 'D';
216 $sLast = 'P' . $last . 'D';
217
218 // The day of today with an interval with the first day of the week.
219 $today = new DateTime();
220 $monday = $today->sub(new DateInterval($sFirst));
221 // The day of today with an interval with the last day of the week.
222 $today = new DateTime();
223 $sunday = $today->add(new DateInterval($sLast));
224
225 $dateMonday = $monday->format('Y-m-d');
226 $dateSunday = $sunday->format('Y-m-d');
227 $query = "SELECT * FROM event, location WHERE dateStart BETWEEN :dateMonday AND
:dateSunday AND status = 1 AND event.id_location = location.id_location ORDER BY
event.id_event ASC";
228 $ps = myDatabase()->prepare($query);
229 $ps->bindParam(':dateMonday', $dateMonday, PDO::PARAM_STR);
230 $ps->bindParam(':dateSunday', $dateSunday, PDO::PARAM_STR);
231 $ps->execute();
232 $results = $ps->fetchAll(PDO::FETCH_ASSOC);
233 foreach ($results as $row) {
234     $event_id = $row['id_event'];
235     $event_title = $row['title'];
236     $event_datestart = $row['dateStart'];
237     $event_dateend = $row['dateEnd'];
238     $event_image = $row['image'];
239     $event_street = $row['street'];
240     $event_city = $row['city'];
241     $event_country = $row['iso_country'];
242
243     echo '<div class="col-xs-6 col-lg-4">
244         <a href="event_details.php?event_id=' . $event_id . '"
245             class="thumbnail">
246             
248             </a>
249             <h3>' . $event_title . '</h3>
250             <p>Date de début : ' . $event_datestart . '</p><p>Date de fin : '
251                 . $event_dateend . '</p>
252             <p>Lieu : ' . $event_street . ' ' . $event_city . ' ' .
253                 $event_country . '</p>
254             <p><a class="btn btn-primary" href="event_details.php?event_id='
255                 . $event_id . '" role="button">Détails</a></p>
256         </div>';
257     }
258 }
259 /**
260  * Function that displays all the events of the month with a status unban (status =
261  * 1) order by the the event id.
262  */
263 function getCurrentMonthEvents() {
264     $date = date("m");

```

```

260     $query = "SELECT event.id_event, title, dateStart, dateEnd, image, street, city,
                iso_country FROM event, location WHERE event.id_location = location.id_location
                AND event.status = 1 AND MONTH(dateStart) = :month ORDER BY event.id_event";
261     $ps = myDatabase()->prepare($query);
262     $ps->bindParam(':month', $date, PDO::PARAM_STR);
263     $ps->execute();
264     $results = $ps->fetchAll(PDO::FETCH_ASSOC);
265     foreach ($results as $row) {
266         $event_id = $row['id_event'];
267         $event_title = $row['title'];
268         $event_datestart = $row['dateStart'];
269         $event_dateend = $row['dateEnd'];
270         $event_image = $row['image'];
271         $event_street = $row['street'];
272         $event_city = $row['city'];
273         $event_country = $row['iso_country'];
274
275         echo '<div class="col-xs-6 col-lg-4">
276             <a href="event_details.php?event_id=' . $event_id . '"
                class="thumbnail">
277                 
278             </a>
279             <h3>' . $event_title . '</h3>
280             <p>Date de début : ' . $event_datestart . '</p><p>Date de fin : '
                . $event_dateend . '</p>
281             <p>Lieu : ' . $event_street . ' ' . $event_city . ' ' .
                $event_country . '</p>
282             <p><a class="btn btn-primary" href="event_details.php?event_id='
                . $event_id . '" role="button">Détails</a></p>
283         </div>';
284     }
285 }
286
287 /**
288  * Function that displays the detail of an event.
289  * @param type $event_id // The id of the event.
290  */
291 function getEventDetail($event_id) {
292     $query = "SELECT event.id_event, title, dateStart, dateEnd, image, description,
                street, city, iso_country FROM event, location WHERE event.id_event = $event_id
                AND event.status = 1 AND event.id_location = location.id_location";
293     $ps = myDatabase()->prepare($query);
294     $ps->execute();
295     $results = $ps->fetchAll(PDO::FETCH_ASSOC);
296     foreach ($results as $row) {
297         $event_id = $row['id_event'];
298         $event_title = $row['title'];
299         $event_datestart = $row['dateStart'];
300         $event_dateend = $row['dateEnd'];
301         $event_desc = $row['description'];
302         $event_image = $row['image'];
303         $event_street = $row['street'];
304         $event_city = $row['city'];
305         $event_country = $row['iso_country'];
306
307         echo '<div class="panel">

```

```

308         <div class="panel-heading">
309             <div class="text-center">
310                 <div class="row">
311                     <div class="col-sm-9">
312                         <h1 class="pull-left">' . $event_title . '</h1>
313                     </div>
314                     <div class="col-sm-3">
315                         <h4 class="pull-right">';
316 // If the user is a member or an admin display the add comment link.
317 if (isUserMember() || isUserAdmin()) {
318     echo '<a href="insert_comment.php?event_id=' . $event_id . '" class="btn
    btn-primary">Ajouter un commentaire</a>';
319 }
320 echo '</h4>
    </div>
    </div>
    </div>
    </div>
    <div class="panel-body">
    <div class="thumbnail">
327         
328     </div>
329     <p>Description : ' . $event_desc . '</p>
330     <p>Date de début : ' . $event_datestart . '</p>
331     <p>Date de début : ' . $event_dateend . '</p>
332     <p>Lieu : ' . $event_street . ' ' . $event_city . ' ' .
        $event_country . '</p>
333     <div class="pull-right"><a class="btn btn-primary"
        href="index.php" role="button">Accueil</a></div>
334 </div>
335 </div>';
336 }
337 }
338
339 /**
340  * Function that returns the events searched in the searchbar.
341  * @param type $search_query // The information to search.
342  */
343 function searchAnEvent($search_query) {
344     $query = "SELECT event.id_event, title, dateStart, dateEnd, image, street, city,
    iso_country FROM event, location WHERE event.id_location = location.id_location
    AND event.status = 1 AND title LIKE '%" . $search_query . "%'";
345     $qsr = myDatabase()->query($query);
346     $results = $qsr->fetchAll(PDO::FETCH_ASSOC);
347     foreach ($results as $row) {
348         $event_id = $row['id_event'];
349         $event_title = $row['title'];
350         $event_datestart = $row['dateStart'];
351         $event_dateend = $row['dateEnd'];
352         $event_image = $row['image'];
353         $event_street = $row['street'];
354         $event_city = $row['city'];
355         $event_country = $row['iso_country'];
356
357     echo '<div class="row col-md-offset-0"><h1>Résultat(s) de la recherche "' .

```



```

        $search_query . '"</h1><br/>
358         <div class="col-xs-6 col-lg-4">
359             <a href="event_details.php?event_id=' . $event_id . '"
                class="thumbnail">
360                 
361             </a>
362             <h3>' . $event_title . '</h3>
363             <p>Date de début : ' . $event_datestart . '</p><p>Date de fin : '
                . $event_dateend . '</p>
364             <p>Lieu : ' . $event_street . ' ' . $event_city . ' ' .
                $event_country . '</p>
365             <p><a class="btn btn-primary" href="event_details.php?event_id='
                . $event_id . '" role="button">Détails</a></p>
366         </div>';
367     }
368 }
369
370 /**
371  * Function that displays all the unban comments.
372  * @param type $event_id          // The id of the event.
373  */
374 function displayCommentUser($event_id) {
375     $query = "SELECT comment.id_comment, time, text, nickname, avatar FROM comment,
        user, event WHERE event.id_event = $event_id AND comment.id_user = user.id_user
        AND comment.id_event = event.id_event AND comment.status = 1";
376     $ps = myDatabase()->prepare($query);
377     $ps->execute();
378     $results = $ps->fetchAll(PDO::FETCH_ASSOC);
379     foreach ($results as $row) {
380         $comment_id = $row['id_comment'];
381         $comment_text = $row['text'];
382         $comment_datetime = $row['time'];
383         $comment_pseudo = $row['nickname'];
384         $comment_avatar = $row['avatar'];
385
386         echo '<div class = "panel-body">
387             <div id="comments" class="col-lg-12">
388                 <ul class="media-list forum">
389                     <li class="media well">
390                         <div class="pull-left user-info col-lg-1" href="#">
391                             
392                             <br/>
393                             <strong>' . $comment_pseudo . '</strong>
394                             <br>
395                         </div>
396                         <div class="media-body">
397                             ' . $comment_text . '
398                         </div>
399                         <div id="postOptions" class="media-right"
                                style="width:100px;">' . $comment_datetime . '
400                             <br/>
401                             <br/>
402                             <form method="POST" action="#">
403                             <input type="hidden" name="hidden_id" value = ' .

```

```

                                $comment_id . '>
405                                </form>
406                            </div>
407                        </li>
408                    </ul>
409                </div>
410            </div>';
411        }
412    }
413
414    /**
415     * Function that displays all the comments ban and unban.
416     * @param type $event_id // The id of the event.
417     */
418    function displayCommentAdmin($event_id) {
419        $query = "SELECT comment.id_comment, time, text, nickname, avatar,
        comment.status FROM comment, user, event WHERE event.id_event = $event_id AND
        comment.id_user = user.id_user AND comment.id_event = event.id_event";
420        $ps = myDatabase()->prepare($query);
421        $ps->execute();
422        $results = $ps->fetchAll(PDO::FETCH_ASSOC);
423        foreach ($results as $row) {
424            $comment_id = $row['id_comment'];
425            $comment_text = $row['text'];
426            $comment_datetime = $row['time'];
427            $comment_pseudo = $row['nickname'];
428            $comment_avatar = $row['avatar'];
429            $comment_status = $row['status'];
430
431            echo '<div class = "panel-body">
432                <div id="comments" class="col-lg-12">
433                    <ul class="media-list forum">
434                        <li class="media well">
435                            <div class="pull-left user-info col-lg-1" href="#">
436                                
437                                <br/>
438                                <strong>' . $comment_pseudo . '</strong>
439                                <br>
440                            </div>
441                            <div class="media-body">
442                                ' . $comment_text . '
443                            </div>
444                            <div id="postOptions" class="media-right"
445                                style="width:100px;">' . $comment_datetime . '
446                                <br/>
447                                <br/>
448                                <form method="POST" action="#">
449                                    <input type="hidden" name="hidden_idcommentadmin"
                                    value = ' . $comment_id . '>';
450                            if ($comment_status == 1) {
451                                echo '<button style="width:100px;" type="submit" name="ban_comment"
                                class="btn btn-danger btn-sm">Bloquer <span class="glyphicon
                                glyphicon-remove"></span></button><br/>';
452                            } else {
453                                echo '<button style="width:100px;" type="submit" name="unban_comment"

```

```

        class="btn btn-success btn-sm">D  bloquer <span class="glyphicon
        glyphicon-ok"></span></button><br/>';
454     }
455     echo '<button onclick="return confirmDeletion()"
        style="width:100px;" type="submit" name="delete_comment" class="btn
        btn-primary btn-sm">Supprimer <span class="glyphicon
        glyphicon-trash"></span></button>
456         </form>
457     </div>
458 </li>
459 </ul>
460 </div>
461 </div>';
462     }
463 }
464
465 /**
466  * Function that modify the user informations from the database.
467  * @param type $datas      // The informations that we insert into the database.
468  */
469 function editUserData($datas) {
470     $query = "UPDATE user SET name = :name, firstname = :firstname, organisation =
        :organisation, adress = :adress, avatar = :avatar, description = :description
        WHERE id_user = :id_user";
471     $ps = myDatabase()->prepare($query);
472
473     $ps->bindParam(':name', $datas['user_name'], PDO::PARAM_STR);
474     $ps->bindParam(':firstname', $datas['user_firstname'], PDO::PARAM_STR);
475     $ps->bindParam(':organisation', $datas['user_organisation'], PDO::PARAM_STR);
476     $ps->bindParam(':adress', $datas['user_adress'], PDO::PARAM_STR);
477     $ps->bindParam(':avatar', $datas['user_avatar'], PDO::PARAM_STR);
478     $ps->bindParam(':description', $datas['user_desc'], PDO::PARAM_STR);
479     $ps->bindParam(':id_user', $datas['user_id'], PDO::PARAM_STR);
480
481     $ps->execute();
482 }
483
484 /**
485  * Function that modify the event informations from the database.
486  * @param type $datas      // The informations that we insert into the database.
487  */
488 function editUserEvent($datas) {
489     $query = "UPDATE event, location SET title = :title, street = :street, city =
        :city, iso_country = :iso_country, dateStart = :dateStart, dateEnd = :dateEnd,
        description = :description, image = :image WHERE id_event = :id_event AND
        event.id_location = location.id_location";
490     $ps = myDatabase()->prepare($query);
491
492     $ps->bindParam(':title', $datas['event_title'], PDO::PARAM_STR);
493     $ps->bindParam(':street', $datas['event_street'], PDO::PARAM_STR);
494     $ps->bindParam(':city', $datas['event_city'], PDO::PARAM_STR);
495     $ps->bindParam(':iso_country', $datas['event_country'], PDO::PARAM_STR);
496     $ps->bindParam(':dateStart', $datas['event_datestart'], PDO::PARAM_STR);
497     $ps->bindParam(':dateEnd', $datas['event_dateend'], PDO::PARAM_STR);
498     $ps->bindParam(':description', $datas['event_desc'], PDO::PARAM_STR);
499     $ps->bindParam(':image', $datas['event_image'], PDO::PARAM_STR);
500     $ps->bindParam(':id_event', $datas['event_id'], PDO::PARAM_STR);

```

```
501
502     $ps->execute();
503 }
504
505 /**
506  * Function that bans a comment.
507  * @param type $datas          // The informations that we update into the database.
508  */
509 function banComment($datas) {
510     $query = "UPDATE comment SET status = 0 WHERE id_comment = :id_comment";
511     $ps = myDatabase()->prepare($query);
512     $ps->bindParam(':id_comment', $datas['comment_id'], PDO::PARAM_STR);
513     $ps->execute();
514 }
515
516 /**
517  * Function that unbans a comment.
518  * @param type $datas          // The informations that we update into the database.
519  */
520 function unbanComment($datas) {
521     $query = "UPDATE comment SET status = 1 WHERE id_comment = :id_comment";
522     $ps = myDatabase()->prepare($query);
523     $ps->bindParam(':id_comment', $datas['comment_id'], PDO::PARAM_STR);
524     $ps->execute();
525 }
526
527 /**
528  * Function that delete from the database the selected event posted by the user.
529  * @param type $datas          // The informations that we delete into the database.
530  */
531 function deleteComment($datas) {
532     $query = "DELETE FROM comment WHERE id_comment = :id_comment";
533     $ps = myDatabase()->prepare($query);
534     $ps->bindParam(':id_comment', $datas['comment_id'], PDO::PARAM_STR);
535     $ps->execute();
536 }
537
538 /**
539  * Function that bans an event.
540  * @param type $datas          // The informations that we update into the database.
541  */
542 function banEvent($datas) {
543     $query = "UPDATE event SET status = 0 WHERE id_event = :id_event ";
544     $ps = myDatabase()->prepare($query);
545     $ps->bindParam(':id_event', $datas['event_id'], PDO::PARAM_STR);
546     $ps->execute();
547 }
548
549 /**
550  * Function that unbans an event.
551  * @param type $datas          // The informations that we update into the database.
552  */
553 function unbanEvent($datas) {
554     $query = "UPDATE event SET status = 1 WHERE id_event = :id_event";
555     $ps = myDatabase()->prepare($query);
556     $ps->bindParam(':id_event', $datas['event_id'], PDO::PARAM_STR);
557     $ps->execute();
```

```
558 }
559
560 /**
561  * Function that deletes an event.
562  * @param type $datas // The informations that we delete into the database.
563  */
564 function deleteEvent($datas) {
565     $query = "DELETE FROM event WHERE id_event = :id_event";
566     $ps = myDatabase()->prepare($query);
567     $ps->bindParam(':id_event', $datas['event_id'], PDO::PARAM_STR);
568     $ps->execute();
569 }
570
571 /**
572  * Function that bans a user.
573  * @param type $datas // The informations that we update into the database.
574  */
575 function banUser($datas) {
576     $query = "UPDATE user SET status = 0 WHERE id_user = :id_user";
577     $ps = myDatabase()->prepare($query);
578     $ps->bindParam(':id_user', $datas['user_id'], PDO::PARAM_STR);
579     $ps->execute();
580 }
581
582 /**
583  * Function that unbans a user.
584  * @param type $datas // The informations that we update into the database.
585  */
586 function unbanUser($datas) {
587     $query = "UPDATE user SET status = 1 WHERE id_user = :id_user";
588     $ps = myDatabase()->prepare($query);
589     $ps->bindParam(':id_user', $datas['user_id'], PDO::PARAM_STR);
590     $ps->execute();
591 }
592
593 /**
594  * Function that deletes an user.
595  * @param type $datas // The informations that we delete into the database.
596  */
597 function deleteUser($datas) {
598     $query = "DELETE FROM user WHERE id_user = :id_user";
599     $ps = myDatabase()->prepare($query);
600     $ps->bindParam(':id_user', $datas['user_id'], PDO::PARAM_STR);
601     $ps->execute();
602 }
603
604 /**
605  * Function that displays all the events posted by a user.
606  * @param type $datas // The informations that we select from the database.
607  */
608 function displayAllUserEvents($datas) {
609     $query = "SELECT event.id_event, title, dateStart, dateEnd, image, description,
        street, city, iso_country FROM event, location WHERE id_user = :id_user AND
        event.id_location = location.id_location";
610     $ps = myDatabase()->prepare($query);
611     $ps->bindParam(':id_user', $datas['user_id'], PDO::PARAM_STR);
612     $ps->execute();
```

```

613     $results = $ps->fetchAll(PDO::FETCH_ASSOC);
614     foreach ($results as $row) {
615         $event_id = $row['id_event'];
616         $event_title = $row['title'];
617         $event_street = $row['street'];
618         $event_city = $row['city'];
619         $event_country = $row['iso_country'];
620         $event_datestart = $row['dateStart'];
621         $event_dateend = $row['dateEnd'];
622         $event_desc = $row['description'];
623         $event_image = $row['image'];
624         echo '<tr>
625             <td>' . $event_title . '</td>
626             <td>' . $event_street . '</td>
627             <td>' . $event_city . '</td>
628             <td>' . $event_country . '</td>
629             <td>' . $event_datestart . '</td>
630             <td>' . $event_dateend . '</td>
631             <td>' . $event_desc . '</td>
632             <td>' . $event_image . '</td>
635             <td>
636                 <form method="POST" action="#">
637                 <input type="hidden" name="hidden_idevent" value = ' . $event_id
638                     . '>
639                 <a href="edit_event.php?event_id=' . $event_id . '" class="btn
640                     btn-primary btn-sm">Modifier <span class="glyphicon
641                     glyphicon-pencil"></span></a>
642                 <button onclick="return confirmDeletion()" style="width:100px;"
643                     type="submit" name="delete_userevent" class="btn btn-danger
644                     btn-sm">Supprimer <span class="glyphicon
645                     glyphicon-trash"></span></button></td>
646             </form>
647             </tr>;'
648     }
649 }
650
651 /**
652  * Function that displays all the events posted by all the users.
653  */
654 function displayAllAdminEvents() {
655     $query = "SELECT * FROM event, location WHERE event.id_location =
656         location.id_location";
657     $ps = myDatabase()->prepare($query);
658     $ps->execute();
659     $results = $ps->fetchAll(PDO::FETCH_ASSOC);
660     foreach ($results as $row) {
661         $event_status = $row['status'];
662         $event_id = $row['id_event'];
663         $event_title = $row['title'];
664         $event_street = $row['street'];
665         $event_city = $row['city'];
666         $event_country = $row['iso_country'];
667         $event_datestart = $row['dateStart'];
668         $event_dateend = $row['dateEnd'];
669         $event_desc = $row['description'];

```

```

661         $sevent_image = $row['image'];
662         echo '<tr>
663             <td>' . $sevent_title . '</td>
664             <td>' . $sevent_street . '</td>
665             <td>' . $sevent_city . '</td>
666             <td>' . $sevent_country . '</td>
667             <td>' . $sevent_datestart . '</td>
668             <td>' . $sevent_dateend . '</td>
669             <td>' . $sevent_desc . '</td>
670             <td>' . $sevent_image . '</td>
671             <td>
672             <form method="POST" action="#">
673             <input type="hidden" name="hidden_ideventadmin" value = ' .
$sevent_id . '>';
674         if ($sevent_status == 1) {
675             echo '<button style="width:100px ;" type="submit" name="ban_adminevent"
class="btn btn-danger btn-sm">Bloquer <span class="glyphicon
glyphicon-remove"></span></button>';
676         } else {
677             echo '<button style="width:100px;" type="submit" name="unban_adminevent"
class="btn btn-success btn-sm">Débloquer <span class="glyphicon
glyphicon-ok"></span></button>';
678         }
679         echo '<button onclick="return confirmDeletion()" style="width:100px;
margin-left:5px;" type="submit" name="delete_adminevent" class="btn
btn-primary btn-sm" > Supprimer <span class="glyphicon
glyphicon-trash"></span></button></td>
680             </form>
681             </tr>';
682     }
683 }
684
685 /**
686  * Function that display all the users of the database.
687  */
688 function displayAllUsers() {
689     $query = "SELECT * FROM user WHERE privilege = 0";
690     $ps = myDatabase()->prepare($query);
691     $ps->execute();
692     $results = $ps->fetchAll(PDO::FETCH_ASSOC);
693     foreach ($results as $row) {
694         $user_id = $row['id_user'];
695         $user_status = $row['status'];
696         $user_email = $row['email'];
697         $user_nickname = $row['nickname'];
698         $user_name = $row['name'];
699         $user_firstname = $row['firstname'];
700         $user_organisation = $row['organisation'];
701         $user_adress = $row['adress'];
702         $user_avatar = $row['avatar'];
703         $user_desc = $row['description'];
704         echo '<tr>
705             <td>' . $user_email . '</td>
706             <td>' . $user_nickname . '</td>
707             <td>' . $user_name . '</td>

```



```

708         <td>' . $user_firstname . '</td>
709         <td>' . $user_organisation . '</td>
710         <td>' . $user_adress . '</td>
711         <td>' . $user_avatar . '</td>
712         <td>' . $user_desc . '</td>
713         <td>
714         <form method="POST" action="#">
715         <input type="hidden" name="hidden_iduseradmin" value = ' .
            $user_id . '>';
716         if ($user_status == 1) {
717             echo '<button style="width:100px ;" type="submit" name="ban_adminuser"
                class="btn btn-danger btn-sm">Bloquer <span class="glyphicon
                glyphicon-remove"></span></button>';
718         } else {
719             echo '<button style="width:100px;" type="submit" name="unban_adminuser"
                class="btn btn-success btn-sm">Débloquer <span class="glyphicon
                glyphicon-ok"></span></button>';
720         }
721         echo '<button onclick="return confirmDeletion()" style="width:100px;
            margin-left:5px;" type="submit" name="delete_adminuser" class="btn
            btn-primary btn-sm">Supprimer <span class="glyphicon
            glyphicon-trash"></span></button></td>
722         </form>
723     </tr>';
724 }
725 }
726
727 /**
728  * Function that displays the profile of a user.
729  * @param type $datas          // The informations that we update into the database.
730  */
731 function displayUserProfile($datas) {
732     $query = "SELECT * FROM user WHERE id_user = :id_user";
733     $ps = myDatabase()->prepare($query);
734     $ps->bindParam(':id_user', $datas['user_id'], PDO::PARAM_STR);
735     $ps->execute();
736     $results = $ps->fetchAll(PDO::FETCH_ASSOC);
737     foreach ($results as $row) {
738         $user_email = $row['email'];
739         $user_nickname = $row['nickname'];
740         $user_name = $row['name'];
741         $user_firstname = $row['firstname'];
742         $user_organisation = $row['organisation'];
743         $user_adress = $row['adress'];
744         $user_avatar = $row['avatar'];
745         $user_desc = $row['description'];
746         echo '<tr>
747             <td>' . $user_email . '</td>
748             <td>' . $user_nickname . '</td>
749             <td>' . $user_name . '</td>
750             <td>' . $user_firstname . '</td>
751             <td>' . $user_organisation . '</td>
752             <td>' . $user_adress . '</td>
753             <td><img class="avatar img-circle img-thumbnail"
                src="images/avatar_images/' . $user_avatar . '"width="64"

```



```

                                alt="Generic placeholder image">' . $user_avatar . '</td>
754                                <td>' . $user_desc . '</td>
755                            </tr>';
756                        }
757                    }
758
759                /**
760                 * Function that select all the informations of the event and put them in an array.
761                 * @param type $datas          // The informations that we select into the database.
762                 * @return type          // The array with the selected info from the database.
763                 */
764                function getEventInfo($datas) {
765                    $query = "SELECT * FROM event, location WHERE id_event = :id_event AND
                        event.id_location = location.id_location";
766                    $ps = myDatabase()->prepare($query);
767                    $ps->bindParam(':id_event', $datas['event_id'], PDO::PARAM_STR);
768                    $ps->execute();
769                    $results = $ps->fetchAll(PDO::FETCH_ASSOC);
770                    return $results;
771                }
772
773                /**
774                 * Function that select all the informations of the user and put them in an array.
775                 * @return type          // The array with the selected info from the database.
776                 */
777                function getUserInfo() {
778                    $user_id = getUserId();
779                    $query = "SELECT * FROM user WHERE id_user = :id_user";
780                    $ps = myDatabase()->prepare($query);
781                    $ps->bindParam(':id_user', $user_id, PDO::PARAM_STR);
782                    $ps->execute();
783                    $results = $ps->fetchAll(PDO::FETCH_ASSOC);
784                    return $results;
785                }
786
787                //----- GENERAL FUNCTIONS -----\\
788
789                // Upload constants.
790                define('INPUT', 'event_image'); // The name of the input file.
791                define('TARGET', './images/event_images/'); // The target directory for the event
                        image.
792                define('TARGET2', './images/avatar_images/'); // The target directory for the
                        avatar image.
793                define('MAX_SIZE', 10000000); // Maximum size in octets for the file.
794                define('WIDTH_MAX', 2000); // Maximum width in pixels for the file.
795                define('HEIGHT_MAX', 2000); // Maximum height in pixels for the file.
796
797                /**
798                 * Fonction that uploads the event image in the correct folder.
799                 * @return type          // The error messages.
800                 */
801                function uploadEventImage() {
802
803                    $stabExt = array('jpg', 'JPG', 'PNG', 'JPEG', 'GIF', 'TIFF', 'BMP', 'gif', 'png',
                        'jpeg', 'bmp', 'tiff'); // Allowed extensions.
804                    $infosImg = array();
805

```

```
806 // Variables.
807 $extension = '';
808 $message = '';
809 $nomImage = '';
810
811 $status = false;
812
813 // Create the repository if it doesn't exists.
814 if (!is_dir(TARGET)) {
815     if (!mkdir(TARGET, 0755)) {
816         exit('Erreur : le répertoire cible ne peut être créé ! Vérifiez que vous
            disposiez des droits suffisants pour le faire ou créez le manuellement !'
            );
817     }
818 }
819 // Check if the field is filled.
820 if (!empty($_FILES[INPUT]['name'])) {
821     // Get the file extension.
822     $extension = pathinfo($_FILES[INPUT]['name'], PATHINFO_EXTENSION);
823     // Check the file extension.
824     if (in_array(strtolower($extension), $tabExt)) {
825         // Get the size of the file.
826         if (!$_FILES[INPUT]['tmp_name'] == "") {
827             $infosImg = getimagesize($_FILES[INPUT]['tmp_name']);
828             // Check the type of file and the header (if it's not an image file).
829             if ($infosImg[2] >= 1 && $infosImg[2] <= 14) {
830                 // Check the size and dimension of the file.
831                 if (($infosImg[0] <= WIDTH_MAX) && ($infosImg[1] <= HEIGHT_MAX)
                    && (filesize($_FILES[INPUT]['tmp_name']) <= MAX_SIZE)) {
832                     // Run through the errors array
833                     if (isset($_FILES[INPUT]['error']) && UPLOAD_ERR_OK ===
                        $_FILES[INPUT]['error']) {
834                         // Rename the file with a uniq id.
835                         $nomImage = uniqid() . '.' . $extension;
836                         // Store the value in session.
837                         $_SESSION['imageevent_name'] = $nomImage;
838
839                         // If it's ok then test the upload.
840                         if (move_uploaded_file($_FILES[INPUT]['tmp_name'],
                            TARGET . $nomImage)) {
841                             $message = 'Upload réussi !';
842                             $status = true;
843                         } else {
844                             // Else, display a system error.
845                             $message = 'Problème lors de l\'upload !';
846                         }
847                     } else {
848                         $message = 'Une erreur interne a empêché l\'upload de
                            l\'image';
849                     }
850                 } else {
851                     // Else, display a size image error.
852                     $message = 'Erreur dans les dimensions de l\'image !';
853                 }
854             }
855         } else {
856             // Else, display a type of image error.
```

```

857         $message = 'Le fichier envoyé n\'est pas une image, ou le fichier
            est trop grand, merci de consulter l\'aide en ligne!';
858     }
859     } else {
860         // Else, display an extension error.
861         $message = 'L\'extension du fichier est incorrecte !';
862     }
863     } else {
864         // Else, display an empty file error.
865         $message = 'Veuillez remplir le formulaire svp !';
866     }
867     return array($message, $status);
868 }
869
870 /**
871  * Fonction that uploads the avatar image in the correct folder.
872  * @return type      // The error messages.
873  */
874 function uploadAvatarImage() {
875
876     $tabExt = array('jpg', 'JPG', 'PNG', 'JPEG', 'GIF', 'TIFF', 'BMP', 'gif', 'png',
877         'jpeg', 'bmp', 'tiff'); // Allowed extensions.
878     $infosImg = array();
879
880     // Variables.
881     $extension = '';
882     $message = '';
883     $nomImage = '';
884
885     $status = false;
886
887     // Create the repository if it doesn't exists.
888     if (!is_dir(TARGET2)) {
889         if (!mkdir(TARGET2, 0755)) {
890             exit('Erreur : le répertoire cible ne peut être créé ! Vérifiez que vous
891                 disposiez des droits suffisants pour le faire ou créez le manuellement !'
892             );
893         }
894     }
895
896     // Check if the field is filled.
897     if (!empty($_FILES[INPUT]['name'])) {
898         // Get the file extension.
899         $extension = pathinfo($_FILES[INPUT]['name'], PATHINFO_EXTENSION);
900         // Check the file extension.
901         if (in_array(strtolower($extension), $tabExt)) {
902             // Get the size of the file.
903             if (!$FILES[INPUT]['tmp_name'] == "") {
904                 $infosImg = getimagesize($FILES[INPUT]['tmp_name']);
905                 // Check the type of file and the header (if it's not an image file).
906                 if ($infosImg[2] >= 1 && $infosImg[2] <= 14) {
907                     // Check the size and dimension of the file.
908                     if (($infosImg[0] <= WIDTH_MAX) && ($infosImg[1] <= HEIGHT_MAX)
909                         && (filesize($FILES[INPUT]['tmp_name']) <= MAX_SIZE)) {
910                         // Run through the errors array
911                         if (isset($_FILES[INPUT]['error']) && UPLOAD_ERR_OK ===
912                             $_FILES[INPUT]['error']) {
913                             // Rename the file with a uniq id.

```

```

908         $nomImage = uniqid() . '.' . $extension;
909         // Store the value in session.
910         $_SESSION['imageavatar_name'] = $nomImage;
911
912         // If it's ok then test the upload.
913         if (move_uploaded_file($_FILES[INPUT]['tmp_name'],
TARGET2 . $nomImage)) {
914             $message = 'Upload réussi !';
915             $status = true;
916         } else {
917             // Else, display a system error.
918             $message = 'Problème lors de l\'upload !';
919         }
920     } else {
921         $message = 'Une erreur interne a empêché l\'upload de
l\'image';
922     }
923 } else {
924     // Else, display a size image error.
925     $message = 'Erreur dans les dimensions de l\'image !';
926 }
927 }
928 } else {
929     // Else, display a type of image error.
930     $message = 'Le fichier envoyé n\'est pas une image, ou le fichier
est trop grand, merci de consulter l\'aide en ligne!';
931 }
932 } else {
933     // Else, display an extension error.
934     $message = 'L\'extension du fichier est incorrecte !';
935 }
936 } else {
937     // Else, display an empty file error.
938     $message = 'Veuillez remplir le formulaire svp !';
939 }
940 return array($message, $status);
941 }
942
943 //----- SESSION FUNCTIONS -----\\
944
945 /**
946  * Function that checks if the user is logged or not.
947  * @return boolean True // If logged return TRUE.
948  */
949 function isLoggedIn() {
950     return (isset($_SESSION['login'])) ? $_SESSION['login'] : false;
951 }
952
953 /**
954  * Function that returns the id of the user from the sessions.
955  * @return type // The id of the user from the sessions.
956  */
957 function getUserId() {
958     return (isset($_SESSION['user_id'])) ? $_SESSION['user_id'] : -1;
959 }
960
961 /**

```

```
962     * Function that define if the user is logged or not.
963     * @param type $value          // TRUE if logged or FALSE if not.
964     */
965     function setLoggedIn($value) {
966         $_SESSION['login'] = $value;
967     }
968
969     /**
970     * Function that sets the group in the session.
971     * @param type $value          // The group you want to put in session.
972     */
973     function setUserGroup($value) {
974         $_SESSION['group'] = $value;
975     }
976
977     /**
978     * Function that returns the group of the user from the session.
979     * @return type          // Return the group of the user.
980     */
981     function getUserGroup() {
982         return (isset($_SESSION['group'])) ? $_SESSION['group'] : -1;
983     }
984
985     /**
986     * Function that disconnect the user and destroy the session.
987     */
988     function logOutUser() {
989         $_SESSION = array();
990         session_destroy();
991     }
992
993     /**
994     * Function that sets the privilege to member.
995     * @return type          // The privilege 0 (member).
996     */
997     function isUserMember() {
998         return (getUserGroup() == 0);
999     }
1000
1001     /**
1002     * Function that sets the privilege to admin.
1003     * @return type          // The privilege 1 (admin).
1004     */
1005     function isUserAdmin() {
1006         return (getUserGroup() == 1);
1007     }
1008
```

```
1  <?php
2
3  /*****
4   * Author : Philippe Ku
5   * School / Class : CFPT Informatique / I.FA-P3B
6   * Date : 15.06.2016
7   * Programm : Event gestion website
8   * File : controller.php
9   * Description : This file is the controller that checks all the datas of my website
10  * Version : 1.10
11  * *****/
12
13  session_start();
14
15  // Sets the errors to null.
16  $signin_error = "";
17  $passwordcheck_error = "";
18  $passwordlength_error = "";
19  $date_error = "";
20  $editdate_error = "";
21  $comment_error = "";
22  $banconnect_error = "";
23
24  // Checks if the insert event button is pressed.
25  if (isset($_POST['insert_event'])) {
26      // Getting the datas with security check from the form field.
27      $datas['event_title'] = filter_var($_POST['event_title'],
28          FILTER_SANITIZE_SPECIAL_CHARS);
29      $datas['event_street'] = filter_var($_POST['event_street'],
30          FILTER_SANITIZE_SPECIAL_CHARS);
31      $datas['event_city'] = filter_var($_POST['event_city'],
32          FILTER_SANITIZE_SPECIAL_CHARS);
33      $datas['event_country'] = filter_var($_POST['event_country'],
34          FILTER_SANITIZE_SPECIAL_CHARS);
35      $datas['event_datestart'] = filter_var($_POST['event_datestart'],
36          FILTER_SANITIZE_SPECIAL_CHARS);
37      $datas['event_dateend'] = filter_var($_POST['event_dateend'],
38          FILTER_SANITIZE_SPECIAL_CHARS);
39      $datas['event_desc'] = $_POST['event_desc']; // The TinyMCE script
40      handles the security check.
41      $datas['user_id'] = getUserId(); // Get the user_id from the sessions.
42
43      if ($_POST['event_datestart'] <= $_POST['event_dateend']) {
44          // Upload the image in the correct folder.
45          uploadEventImage();
46
47          // Getting the datas from the image.
48          $datas['event_image'] = $_SESSION['imageevent_name'];
49
50          // Add the event into the database.
51          insertNewEvent($datas);
52
53          // Redirect to the index page.
54          header('location: index.php');
55      } else {
56          $date_error = '<div class="alert alert-warning"><strong>ATTENTION
57          !</strong>La date de fin doit être supérieur ou égale à celle de début
```

```

        !</div>';
50     }
51 }
52
53
54 // Checks if the insert user button is pressed.
55 if (isset($_POST['insert_user'])) {
56     // Getting the datas with security check from the form field.
57     $datas['user_email'] = filter_var($_POST['user_email'],
58     FILTER_SANITIZE_SPECIAL_CHARS);
59     $datas['user_nickname'] = filter_var($_POST['user_nickname'],
60     FILTER_SANITIZE_SPECIAL_CHARS);
61     // Checks if the two passwords are matching.
62     if ($_POST['user_password'] == $_POST['user_checkpassword']) {
63         if (strlen($_POST['user_password']) >= 3) {
64             $datas['user_password'] = sha1($_POST['user_password']);
65             // Add the new user in the database.
66             insertNewUser($datas);
67             // Shows a succes message.
68             echo '<script type="text/javascript">
69                 alert("Votre compte à été crée !");
70                 document.location.href = "index.php";
71             </script>';
72         } else {
73             $passwordlength_error = '<div class="alert
74             alert-warning"><strong>ATTENTION !</strong> Le mot de passe doit faire
75             au minimum 3 caractères !</div>';
76         }
77     } else {
78         $passwordcheck_error = '<div class="alert alert-warning"><strong>ATTENTION
79         !</strong> Les mots de passe ne correspondent pas !</div>';
80     }
81 }
82
83 // Checks if the connect button is pressed.
84 if (isset($_POST['connect_user'])) {
85     // Getting the datas from the form field.
86     $datas['user_email'] = (isset($_POST['user_email']) ? $_POST['user_email'] : "");
87     $datas['user_password'] = (isset($_POST['user_password']) ? $_POST[
88     'user_password'] : "");
89
90     // Connect the user with the datas he gave us. If all the datas are true then
91     log him.
92     if (checkuser($datas)) {
93
94         $results = getUserInfo();
95         if ($results[0]['status'] == 1) {
96             // Sets the logged status to true.
97             setLoggedIn(true);
98             // Set his group.
99             setUserGroup(getUserGroupFromDatabase($datas));
100         } else {
101             $banconnect_error = '<div class="alert alert-warning"><strong>ATTENTION
102             !</strong> Votre compte a été banni veuillez contacter l\'administrateur
103             !</div>';
104         }
105     } else {

```

```
97         $signin_error = '<div class="alert alert-warning"><strong>ATTENTION
          !</strong> La combinaison email / mot de passe est incorrecte!</div>';
98     }
99 }
100
101 // Checks if the comment button is pressed.
102 if (isset($_POST['insert_comment'])) {
103     // Getting the datas from the form field.
104     $datas['comment_time'] = date('Y-m-d H:i:s');
105     $datas['comment_text'] = $_POST['comment_text'];
106     $datas['user_id'] = getUserId();
107     $datas['event_id'] = $_REQUEST['event_id'];
108
109     if ($_POST['comment_text'] != "") {
110         // Add the new comment into the database.
111         insertNewComment($datas);
112         // Redirect to the event detail page.
113         header('Location: event_details.php?event_id=' . $_REQUEST['event_id']);
114     } else {
115         $comment_error = '<div class="alert alert-warning"><strong>ATTENTION
          !</strong> Le texte du commentaire ne doit pas être vide !</div>';
116     }
117 }
118
119 // Checks if the logout button is pressed.
120 if (isset($_REQUEST['logout'])) {
121     logOutUser();
122     header('Location: index.php');
123 }
124
125 // Change the menubar depending if it's a visitor, member or admin that signin.
126 switch (getUserGroup()) {
127     case -1: require '/menu/default_menu.php';
128         break;
129     case 0: require '/menu/user_menu.php';
130         break;
131     case 1: require '/menu/admin_menu.php';
132         break;
133     default :require '/menu/default_menu.php';
134         break;
135 }
136
137 // Checks if the edit profile button is pressed.
138 if (isset($_POST['edit_profile'])) {
139     // Getting the datas (secured!) from the form fields.
140     $datas['user_name'] = filter_var($_POST['edit_name'],
        FILTER_SANITIZE_SPECIAL_CHARS);
141     $datas['user_firstname'] = filter_var($_POST['edit_firstname'],
        FILTER_SANITIZE_SPECIAL_CHARS);
142     $datas['user_organisation'] = filter_var($_POST['edit_organisation'],
        FILTER_SANITIZE_SPECIAL_CHARS);
143     $datas['user_adress'] = filter_var($_POST['edit_adress'],
        FILTER_SANITIZE_SPECIAL_CHARS);
144     $datas['user_desc'] = $_POST['edit_desc'];
145     $datas['user_id'] = getUserId();
146
147     // Upload the image in the correct file.
```



```
148     uploadAvatarImage();
149
150     // Getting the datas for the image.
151     $datas['user_avatar'] = $_SESSION['imageavatar_name'];
152
153     // Edit the new profile in the database.
154     editUserData($datas);
155
156     // Redirect to the index page.
157     header('Location: myaccount.php');
158 }
159
160 // Checks if the edit event button is pressed.
161 if (isset($_POST['edit_event'])) {
162     // Getting the datas (secured!) from the form fields.
163     $datas['event_title'] = filter_var($_POST['editevent_title'],
164     FILTER_SANITIZE_SPECIAL_CHARS);
165     $datas['event_street'] = filter_var($_POST['editevent_street'],
166     FILTER_SANITIZE_SPECIAL_CHARS);
167     $datas['event_city'] = filter_var($_POST['editevent_city'],
168     FILTER_SANITIZE_SPECIAL_CHARS);
169     $datas['event_country'] = filter_var($_POST['editevent_country'],
170     FILTER_SANITIZE_SPECIAL_CHARS);
171     $datas['event_datestart'] = filter_var($_POST['editevent_datestart'],
172     FILTER_SANITIZE_SPECIAL_CHARS);
173     $datas['event_dateend'] = filter_var($_POST['editevent_dateend'],
174     FILTER_SANITIZE_SPECIAL_CHARS);
175     $datas['event_desc'] = $_POST['editevent_desc'];
176     $datas['event_id'] = $_GET['event_id'];
177
178     if ($_POST['editevent_datestart'] <= $_POST['editevent_dateend']) {
179
180         // Upload the image in the correct folder.
181         uploadEventImage();
182         // Getting the datas for the image.
183         $datas['event_image'] = $_SESSION['imageevent_name'];
184
185         // Edit the new profile in the database.
186         editUserEvent($datas);
187
188         // Redirect to the index page.
189         header('Location: myaccount.php');
190     } else {
191         $editdate_error = '<div class="alert alert-warning"><strong>ATTENTION
192         !</strong>La date de fin doit être supérieur ou égale à celle de début
193         !</div>';
194     }
195 }
196
197 // Checks if the searchbox is not empty and displays the results of the search.
198 if (isset($_GET['search'])) {
199     $search_query = filter_var($_GET['user_query'], FILTER_SANITIZE_SPECIAL_CHARS);
200     if ($search_query != "") {
201         searchAnEvent($search_query);
202     } else {
203         echo '<div class="row col-md-offset-0"><h1>Veuillez entrer un élément de
204         recherche...</h1><br/>';
```

```
196     }
197 }
198
199 // Checks if the delete user event button is pressed.
200 if (isset($_POST['delete_userevent'])) {
201     $datas['event_id'] = $_POST['hidden_idevent'];
202     deleteEvent($datas);
203 }
204
205 // Checks if the ban admin event button is pressed.
206 if (isset($_POST['ban_adminevent'])) {
207     $datas['event_id'] = $_POST['hidden_ideventadmin'];
208     banEvent($datas);
209 }
210
211 // Checks if the unban event admin button is pressed.
212 if (isset($_POST['unban_adminevent'])) {
213     $datas['event_id'] = $_POST['hidden_ideventadmin'];
214     unbanEvent($datas);
215 }
216
217 // Checks if the delete admin event button is pressed
218 if (isset($_POST['delete_adminevent'])) {
219     $datas['event_id'] = $_POST['hidden_ideventadmin'];
220     deleteEvent($datas);
221 }
222
223 // Checks if the ban comment button is pressed.
224 if (isset($_POST['ban_comment'])) {
225     $datas['comment_id'] = $_POST['hidden_idcommentadmin'];
226     banComment($datas);
227 }
228
229 // Checks if the unban comment button is pressed.
230 if (isset($_POST['unban_comment'])) {
231     $datas['comment_id'] = $_POST['hidden_idcommentadmin'];
232     unbanComment($datas);
233 }
234
235 // Checks if the delete comment button is pressed.
236 if (isset($_POST['delete_comment'])) {
237     $datas['comment_id'] = $_POST['hidden_idcommentadmin'];
238     deleteComment($datas);
239 }
240
241 // Checks if the ban admin user button is pressed.
242 if (isset($_POST['ban_adminuser'])) {
243     $datas['user_id'] = $_POST['hidden_iduseradmin'];
244     banUser($datas);
245 }
246
247 // Checks if the unban admin user button is pressed.
248 if (isset($_POST['unban_adminuser'])) {
249     $datas['user_id'] = $_POST['hidden_iduseradmin'];
250     unbanUser($datas);
251 }
252
```

```
253 // Checks if the delete admin user button is pressed
254 if (isset($_POST['delete_adminuser'])) {
255     $datas['user_id'] = $_POST['hidden_iduseradmin'];
256     deleteUser($datas);
257 }
258
```

```
1  <!------->
2  * Author : Philippe Ku
3  * School / Class : CFPT Informatique / I.FA-P3B
4  * Date : 15.06.2016
5  * Programm : Event gestion website
6  * File : default_menu.php
7  * Description : This file is the menubar for the un-authenticated user (visitors)
8  * Version : 1.10
9  ----->
10 <nav class="navbar navbar-inverse">
11     <div class="container-fluid">
12         <ul class="nav navbar-nav">
13             <li class="active"><a href="index.php"><b>Events Manager</b></a></li>
14             <li><a href="index.php">Accueil</a></li>
15             <li><a href="about.php">A propos</a></li>
16         </ul>
17         <form class="navbar-form navbar-right inline-form" method="GET" action=
18             "results.php">
19             <input type="text" name="user_query" class="input-sm form-control"
20                 placeholder="Recherche...">
21             <button type="submit" name="search" style="margin-right: 10px;" class=
22                 "btn btn-primary"><span class="glyphicon glyphicon-search"
23                 ></span></button>
24             <a href="signup.php" style="margin-right: 10px;"><span class="glyphicon
25                 glyphicon-pencil"></span> S'inscrire </a>
26             <a href="signin.php" style="margin-right: 10px;"><span class="glyphicon
27                 glyphicon-log-in"></span> Se connecter </a>
28         </form>
29     </div>
30 </nav>
```

```
1  <!------->
2  * Author : Philippe Ku
3  * School / Class : CFPT Informatique / I.FA-P3B
4  * Date : 15.06.2016
5  * Programm : Event gestion website
6  * File : user_menu.php
7  * Description : This file is the menubar for the authenticated user (members)
8  * Version : 1.10
9  ----->
10 <nav class="navbar navbar-inverse">
11     <div class="container-fluid">
12         <ul class="nav navbar-nav">
13             <li class="active"><a href="index.php"><b>Events Manager</b></a></li>
14             <li><a href="index.php">Accueil</a></li>
15             <li><a href="insert_event.php">Ajouter un événement</a></li>
16             <li><a href="about.php">A propos</a></li>
17         </ul>
18         <form class="navbar-form navbar-right inline-form" method="GET" action=
19             "results.php">
20             <input type="text" name="user_query" class="input-sm form-control"
21                 placeholder="Recherche...">
22             <button type="submit" name="search" style="margin-right: 10px;" class=
23                 "btn btn-primary"><span class="glyphicon glyphicon-search"
24                 ></span></button>
25             <a href="myaccount.php" style="margin-right: 10px;"><span class=
26                 "glyphicon glyphicon-user"></span> Mon compte </a>
27             <a href="index.php?logout=true" style="margin-right: 10px;"><span class=
28                 "glyphicon glyphicon-off"></span> Se déconnecter </a>
29         </form>
30     </div>
31 </nav>
```

```
1  <!------->
2  * Author : Philippe Ku
3  * School / Class : CFPT Informatique / I.FA-P3B
4  * Date : 15.06.2016
5  * Programm : Event gestion website
6  * File : admin_menu.php
7  * Description : This file is the menubar for the authenticated users (administrators)
8  * Version : 1.10
9  ----->
10 <nav class="navbar navbar-inverse">
11     <div class="container-fluid">
12         <ul class="nav navbar-nav">
13             <li class="active"><a href="index.php"><b>Events Manager</b></a></li>
14             <li><a href="index.php">Accueil</a></li>
15             <li><a href="insert_event.php">Ajouter un événement</a></li>
16             <li><a href="about.php">A propos</a></li>
17         </ul>
18         <form class="navbar-form navbar-right inline-form" method="GET" action=
19             "results.php">
20             <input type="text" name="user_query" class="input-sm form-control"
21                 placeholder="Recherche...">
22             <button type="submit" name="search" id="navbarbutton" class="btn
23                 btn-primary"><span class="glyphicon glyphicon-search"></span></button>
24             <a href="administration.php" style="margin-right: 10px;"><span class=
25                 "glyphicon glyphicon-cog"></span> Administration </a>
26             <a href="myaccount.php" style="margin-right: 10px;"><span class=
27                 "glyphicon glyphicon-user"></span> Mon compte </a>
28             <a href="index.php?logout=true" style="margin-right: 10px;"><span class=
29                 "glyphicon glyphicon-off"></span> Se déconnecter </a>
30         </form>
31     </div>
32 </nav>
```

```

1  <!-------
2  * Author : Philippe Ku
3  * School / Class : CFPT Informatique / I.FA-P3B
4  * Date : 15.06.2016
5  * Programm : Event gestion website
6  * File : index.php
7  * Description : The index page of the website
8  * Version : 1.10
9  ----->
10 <?php
11 require 'functions/functions.php';
12 require 'controller.php';
13 ?>
14 <!DOCTYPE html>
15 <html>
16     <head>
17         <meta charset="UTF-8">
18         <meta http-equiv="X-UA-Compatible" content="IE=edge">
19         <meta name="viewport" content="width=device-width, initial-scale=1">
20         <title>Accueil</title>
21         <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
22         <script src="
23             https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
24         <script src="bootstrap/js/bootstrap.min.js"></script>
25     </head>
26     <body>
27         <div class="row col-md" style="padding-left: 20px;">
28             <form method="POST" action="">
29                 <select class="input-sm" name="event_filter">
30                     <option value="default">Sélectionner un filtre</option>
31                     <option value="day">Jour</option>
32                     <option value="week">Semaine</option>
33                     <option value="month">Mois</option>
34                 </select>
35                 <input class="btn btn-primary btn-sm" type="submit" name="filter"/>
36             </form>
37         </div>
38         <?php
39             // Changes the date to a french format.
40             setlocale(LC_TIME, 'fr', 'fr_FR', 'fr_FR.ISO8859-1');
41             $date = strftime("%A %d %B %Y.");
42             $month = strftime("%B %Y.");
43
44             // Displays the events depending on the filter (day, week, month).
45             if (isset($_POST['filter'])) {
46                 if ($_POST['event_filter'] == "day") {
47                     echo '<div class="row col-md-offset-0"><h1>Les événements du jour : '
48                         . $date . '</h1><br/>';
49                     getCurrentDayEvents();
50                 }
51                 if ($_POST['event_filter'] == "week") {
52                     echo '<div class="row col-md-offset-0"><h1>Les événements de la
53                         semaine</h1><br/>';
54                     getCurrentWeekEvents();
55                 }
56                 if ($_POST['event_filter'] == "month") {
57                     echo '<div class="row col-md-offset-0"><h1>Les événements du mois : '

```

```
        . $month . ' </h1><br/>';
55     getCurrentMonthEvents();
56     }
57     } else {
58         echo '<div class="row col-md-offset-0"><h1>Les événements du jour : ' .
            $date . ' </h1><br/>';
59         getCurrentDayEvents();
60     }
61     ?>
62     </body>
63 </html>
64
65
66
67
68
```



```
1  <!------->
2  * Author : Philippe Ku
3  * School / Class : CFPT Informatique / I.FA-P3B
4  * Date : 15.06.2016
5  * Programm : Event gestion website
6  * File : event_details.php
7  * Description : The event details page of the website
8  * Version : 1.10
9  ----->
10 <?php
11 require 'functions/functions.php';
12 require 'controller.php';
13 ?>
14 <!DOCTYPE html>
15 <html>
16     <head>
17         <meta charset="UTF-8">
18         <meta http-equiv="X-UA-Compatible" content="IE=edge">
19         <meta name="viewport" content="width=device-width, initial-scale=1">
20         <title>Détails de l'événement</title>
21         <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
22         <script src="
23             https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
24         <script src="bootstrap/js/bootstrap.min.js"></script>
25         <script type="text/javascript">
26             // Function that generate a message for the delete confirmation.
27             function confirmDeletion() {
28                 return (confirm("Voulez-vous supprimer cette entrée ?"));
29             }
30         </script>
31     </head>
32     <body>
33         <?php
34             // Checks if there is an event id and if so displays the event details and
35             // the comments.
36             if (isset($_GET['event_id'])) {
37                 $event_id = $_GET['event_id'];
38                 getEventDetail($event_id);
39                 if (isUserAdmin()) {
40                     displayCommentAdmin($event_id);
41                 } else {
42                     displayCommentUser($event_id);
43                 }
44             }
45         ?>
46     </body>
47 </html>
```

```
1  <!------->
2  * Author : Philippe Ku
3  * School / Class : CFPT Informatique / I.FA-P3B
4  * Date : 15.06.2016
5  * Programm : Event gestion website
6  * File : about.php
7  * Description : The about page of the website
8  * Version : 1.10
9  ----->
10 <?php
11 require 'functions/functions.php';
12 require 'controller.php';
13 ?>
14 <!DOCTYPE html>
15 <html>
16     <head>
17         <meta charset="UTF-8">
18         <meta http-equiv="X-UA-Compatible" content="IE=edge">
19         <meta name="viewport" content="width=device-width, initial-scale=1">
20         <title>A propos</title>
21         <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
22         <script src="
23             https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
24         <script src="bootstrap/js/bootstrap.min.js"></script>
25     </head>
26     <body>
27         <div class="container">
28             <div class="row col-md-offset-0">
29                 <h1>A propos</h1><br/>
30                 <div class="row">
31                     <h2>Le site :</h2><br/>
32                     <p>Sed ut perspiciatis unde omnis iste natus error sit
33 voluptatem accusantium doloremque laudantium,
34 totam rem aperiam, eaque ipsa quae ab illo inventore
35 veritatis et quasi architecto beatae vitae
36 dicta sunt explicabo. Nemo enim ipsam voluptatem quia
37 voluptas sit aspernatur aut odit aut fugit,
38 sed quia consequuntur magni dolores eos qui ratione
39 voluptatem sequi nesciunt. Neque porro quisquam
40 est, qui dolorem ipsum quia dolor sit amet, consectetur,
41 adipisci velit, sed quia non numquam eius modi
42 tempora incidunt ut labore et dolore magnam aliquam quaerat
43 voluptatem. Ut enim ad minima veniam, quis
44 nostrum exercitationem ullam corporis suscipit laboriosam,
45 nisi ut aliquid ex ea commodi consequatur? Quis
46 autem vel eum iure reprehenderit qui in ea voluptate velit
47 esse quam nihil molestiae consequatur, ve
48 l illum qui dolorem eum fugiat quo voluptas nulla pariatur?
49 </p>
50 </div>
51 <div class="row">
52     <h2>L'auteur :</h2><br/>
53     <p>Sed ut perspiciatis unde omnis iste natus error sit
54 voluptatem accusantium doloremque laudantium,
55 totam rem aperiam, eaque ipsa quae ab illo inventore
56 veritatis et quasi architecto beatae vitae
57 dicta sunt explicabo. Nemo enim ipsam voluptatem quia
```

```
46 voluptas sit aspernatur aut odit aut fugit,  
sed quia consequuntur magni dolores eos qui ratione  
47 voluptatem sequi nesciunt. Neque porro quisquam  
est, qui dolorem ipsum quia dolor sit amet, consectetur,  
48 adipisci velit, sed quia non numquam eius modi  
tempora incidunt ut labore et dolore magnam aliquam quaerat  
voluptatem. Ut enim ad minima veniam, quis  
49 nostrum exercitationem ullam corporis suscipit laboriosam,  
nisi ut aliquid ex ea commodi consequatur? Quis  
50 autem vel eum iure reprehenderit qui in ea voluptate velit  
esse quam nihil molestiae consequatur, ve  
51 l illum qui dolorem eum fugiat quo voluptas nulla pariat?  
</p>  
52 </div>  
53 <div class="row">  
54 <h2>F.A.Q :</h2><br/>  
55 <p>Sed ut perspiciatis unde omnis iste natus error sit  
voluptatem accusantium doloremque laudantium,  
56 totam rem aperiam, eaque ipsa quae ab illo inventore  
veritatis et quasi architecto beatae vitae  
57 dicta sunt explicabo. Nemo enim ipsam voluptatem quia  
voluptas sit aspernatur aut odit aut fugit,  
58 sed quia consequuntur magni dolores eos qui ratione  
voluptatem sequi nesciunt. Neque porro quisquam  
59 est, qui dolorem ipsum quia dolor sit amet, consectetur,  
adipisci velit, sed quia non numquam eius modi  
60 tempora incidunt ut labore et dolore magnam aliquam quaerat  
voluptatem. Ut enim ad minima veniam, quis  
61 nostrum exercitationem ullam corporis suscipit laboriosam,  
nisi ut aliquid ex ea commodi consequatur? Quis  
62 autem vel eum iure reprehenderit qui in ea voluptate velit  
esse quam nihil molestiae consequatur, ve  
63 l illum qui dolorem eum fugiat quo voluptas nulla pariat?  
</p>  
64 </div>  
65 </div>  
66 </div>  
67 </body>  
68 </html>
```

```

1  <!-------
2  * Author : Philippe Ku
3  * School / Class : CFPT Informatique / I.FA-P3B
4  * Date : 15.06.2016
5  * Programm : Event gestion website
6  * File : signin.php
7  * Description : The sign in page of the website
8  * Version : 1.10
9  ----->
10 <?php
11 require 'functions/functions.php';
12 require 'controller.php';
13 ?>
14 <!DOCTYPE html>
15 <html>
16     <head>
17         <meta charset="UTF-8">
18         <meta http-equiv="X-UA-Compatible" content="IE=edge">
19         <meta name="viewport" content="width=device-width, initial-scale=1">
20         <title>Connexion</title>
21         <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
22         <script src="
23             https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
24         <script src="bootstrap/js/bootstrap.min.js"></script>
25     </head>
26     <body>
27         <div class="container">
28             <?php
29             // If the user is already logged.
30             if (isLoggedIn()) {
31                 // Redirect to the index.
32                 header('location: index.php');
33             }
34             <?>
35             <div class="row col-md-offset-0">
36                 <h1>Connexion</h1>
37             </div>
38             <div class="row">
39                 <div class="col-md-4 col-md-offset-2 pull-left">
40                     <div class="panel panel-default" style="width: 800px;">
41                         <div class="panel-heading">
42                             <h3 class="panel-title">Connexion</h3>
43                         </div>
44                         <div class="panel-body" style="width: 800px;">
45                             <form method="POST" action="#">
46                                 <fieldset>
47                                     <div class="form-group">
48                                         <label>Email :</label>
49                                         <input class="form-control" placeholder="
50                                             Email..." name="user_email" type="email"
51                                             required="">
52                                     </div>
53                                     <div class="form-group">
54                                         <label>Mot de passe :</label>
55                                         <input class="form-control" placeholder="Mot
56                                             de passe..." name="user_password" type="
57                                             password" required="">

```

```
53         </div>
54         <input class="btn btn-lg btn-primary btn-block"
           type="submit" name="connect_user" value="Se
           connecter">
55     </fieldset>
56 </form>
57 <?php
58 // Displays the error messages.
59 echo $signin_error;
60 echo $banconnect_error;
61 ?>
62 </div>
63 </div>
64 </div>
65 </div>
66 </div>
67 </body>
68 </html>
69
70
71
72
73
74
75
```

```

1  <!-------
2  * Author : Philippe Ku
3  * School / Class : CFPT Informatique / I.FA-P3B
4  * Date : 15.06.2016
5  * Programm : Event gestion website
6  * File : signup.php
7  * Description : The sign up page of the website
8  * Version : 1.10
9  ----->
10 <?php
11 require 'functions/functions.php';
12 require 'controller.php';
13 ?>
14 <!DOCTYPE html>
15 <html>
16     <head>
17         <meta charset="UTF-8">
18         <meta http-equiv="X-UA-Compatible" content="IE=edge">
19         <meta name="viewport" content="width=device-width, initial-scale=1">
20         <title>Inscription</title>
21         <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
22         <script src="
23             https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
24         <script src="bootstrap/js/bootstrap.min.js"></script>
25     </head>
26     <body>
27         <div class="container">
28             <?php
29             // If the user is already logged.
30             if (isLoggedIn()) {
31                 // Redirect to the index.
32                 header('location: index.php');
33             }
34             <?>
35             <div class="row col-md-offset-0">
36                 <h1>Inscription</h1>
37             </div>
38             <div class="row">
39                 <div class="col-md-4 col-md-offset-2 pull-left">
40                     <div class="panel panel-default" style="width: 800px;">
41                         <div class="panel-heading">
42                             <h3 class="panel-title">Inscription</h3>
43                         </div>
44                         <div class="panel-body" style="width: 800px;">
45                             <form method="POST" action="#">
46                                 <fieldset>
47                                     <div class="form-group">
48                                         <label>Email :</label>
49                                         <input class="form-control" placeholder="
50                                             Email..." name="user_email" type="email"
51                                             required="">
52                                     </div>
53                                     <div class="form-group">
54                                         <label>Pseudo :</label>
55                                         <input class="form-control" placeholder="
56                                             Pseudo..." name="user_nickname" type="text"
57                                             required="">

```

```
53         </div>
54         <div class="form-group">
55             <label>Mot de passe :</label>
56             <input class="form-control" placeholder="Mot
de passe..." name="user_password" type=
"password" required="">
57         </div>
58         <div class="form-group">
59             <label>Vérification Mot de passe :</label>
60             <input class="form-control" placeholder=
"Vérification du mot de passe..." name=
"user_checkpassword" type="password" required
="">
61         </div>
62         <input class="btn btn-lg btn-primary btn-block"
type="submit" name="insert_user" value=
"S'inscrire">
63     </fieldset>
64 </form>
65 <?php
66     // Displays the error messages.
67     echo $passwordlength_error;
68     echo $passwordcheck_error;
69     ?>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </body>
76 </html>
77
78
79
80
81
82
83
```

```
1  <!------->
2  * Author : Philippe Ku
3  * School / Class : CFPT Informatique / I.FA-P3B
4  * Date : 15.06.2016
5  * Programm : Event gestion website
6  * File : results.php
7  * Description : The search results page of the website
8  * Version : 1.10
9  ----->
10 <?php
11 require 'functions/functions.php';
12 require 'controller.php';
13 ?>
14 <!DOCTYPE html>
15 <html>
16     <head>
17         <meta charset="UTF-8">
18         <meta http-equiv="X-UA-Compatible" content="IE=edge">
19         <meta name="viewport" content="width=device-width, initial-scale=1">
20         <title>Résultats de la recherche</title>
21         <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
22         <script src="
23             https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
24             <script src="bootstrap/js/bootstrap.min.js"></script>
25     </head>
26     <body>
27     </body>
28 </html>
29
30
31
32
```



```

1  <!-------
2  * Author : Philippe Ku
3  * School / Class : CFPT Informatique / I.FA-P3B
4  * Date : 15.06.2016
5  * Programm : Event gestion website
6  * File : insert_event.php
7  * Description : The insert event page of the website
8  * Version : 1.10
9  ----->
10 <?php
11 require 'functions/functions.php';
12 require 'controller.php';
13 ?>
14 <!DOCTYPE html>
15 <html>
16     <head>
17         <meta charset="UTF-8">
18         <meta http-equiv="X-UA-Compatible" content="IE=edge">
19         <meta name="viewport" content="width=device-width, initial-scale=1">
20         <title>Ajouter un événement</title>
21         <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
22         <script src="
23             https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
24         <script src="bootstrap/js/bootstrap.min.js"></script>
25         <!-- Script that adds TinyMCE, a JavaScript text editor for the textarea. -->
26         <script src="//cdn.tinymce.com/4/tinymce.min.js"></script>
27         <script>tinyMCE.init({selector: 'textarea'});</script>
28     </head>
29     <body>
30         <?php
31             // If the user is not a member or an admin.
32             if (!isUserAdmin() && !isUserMember()) {
33                 // Redirect to the index.
34                 header('location: index.php');
35             }
36         ?>
37         <div class="container">
38             <div class="row col-md-offset-0">
39                 <h1>Ajouter un événement</h1><br/>
40             </div>
41             <div class="row">
42                 <div class="col-md-4 col-md-offset-2 pull-left">
43                     <div class="panel panel-default" style="width: 800px;">
44                         <div class="panel-heading">
45                             <h3 class="panel-title">Ajouter un événement</h3>
46                         </div>
47                         <div class="panel-body" style="width: 800px;">
48                             <form method="POST" action="#" enctype=
49                                 "multipart/form-data">
50                                 <fieldset>
51                                     <div class="form-group">
52                                         <label>Titre :</label>
53                                         <input class="form-control" placeholder=
54                                             "Titre..." name="event_title" type="text"
55                                             required="">
56                                     </div>
57                                 </div>
58                             </div>
59                         </div>
60                     </div>
61                 </div>
62             </div>
63         </div>

```

```

54         <label>Rue :</label>
55         <input class="form-control" placeholder=
           "Rue..." name="event_street" type="text"
           required="">
56     </div>
57     <div class="form-group">
58         <label>Ville :</label>
59         <input class="form-control" placeholder=
           "Ville..." name="event_city" type="text"
           required="">
60     </div>
61     <div class="form-group">
62         <label>Pays :</label>
63         <select class="form-control" name=
           "event_country" required="">
64             <option>Sélectionnez un pays</option>
65             <?php getCountriesList(); // Gets the
               list of all the countries in the world. ?>
66         </select>
67     </div>
68     <div class="form-group">
69         <label>Date de début :</label>
70         <input class="form-control" name=
           "event_datestart" type="date" required="">
71     </div>
72     <div class="form-group">
73         <label>Date de fin :</label>
74         <input class="form-control" name=
           "event_dateend" type="date">
75     </div>
76     <div class="form-group">
77         <label>Description :</label>
78         <textarea class="form-control" name=
           "event_desc" cols="50" rows="10" style=
           "max-width: 100%;"></textarea>
79     </div>
80     <div class="form-group">
81         <label>Image :</label>
82         <input class="form-control" name="<?php echo
           INPUT; ?>" type="file" required="">
83     </div>
84     <input class="btn btn-lg btn-primary btn-block"
           type="submit" name="insert_event" value="Ajouter
           l'événement">
85 </fieldset>
86 </form>
87 <?php
88 // Displays the error messages.
89 echo $date_error;
90 ?>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 </body>
97 </html>

```

```

1  <!-------
2  * Author : Philippe Ku
3  * School / Class : CFPT Informatique / I.FA-P3B
4  * Date : 15.06.2016
5  * Programm : Event gestion website
6  * File : insert_comment.php
7  * Description : The insert comment page of the website
8  * Version : 1.10
9  ----->
10 <?php
11 require 'functions/functions.php';
12 require 'controller.php';
13 ?>
14 <!DOCTYPE html>
15 <html>
16     <head>
17         <meta charset="UTF-8">
18         <meta http-equiv="X-UA-Compatible" content="IE=edge">
19         <meta name="viewport" content="width=device-width, initial-scale=1">
20         <title>Ajouter un commentaire</title>
21         <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
22         <script src="
23             https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
24         <script src="bootstrap/js/bootstrap.min.js"></script>
25         <!-- Script that adds TinyMCE, a JavaScript text editor for the textarea. -->
26         <script src="//cdn.tinymce.com/4/tinymce.min.js"></script>
27         <script>tinyMCE.init({selector: 'textarea'});</script>
28     </head>
29     <body>
30         <?php
31             // If the user is not a member or an admin.
32             if (!isUserAdmin() && !isUserMember()) {
33                 // Redirect to the index.
34                 header('location: index.php');
35             }
36         ?>
37         <div class="container">
38             <div class="row col-md-offset-0">
39                 <h1>Ajouter un commentaire</h1><br/>
40             </div>
41             <div class="row">
42                 <div class="col-md-4 col-md-offset-2 pull-left">
43                     <div class="panel panel-default" style="width: 800px;">
44                         <div class="panel-heading">
45                             <h3 class="panel-title">Ajouter un commentaire</h3>
46                         </div>
47                         <div class="panel-body" style="width: 800px;">
48                             <form method="POST" action="#" enctype=
49                                 "multipart/form-data">
50                                 <fieldset>
51                                     <div class="form-group">
52                                         <label>Texte :</label>
53                                         <textarea class="form-control" name=
54                                             "comment_text" cols="50" rows="10" style=
55                                                 "max-width: 100%;"></textarea>
56                                     </div>
57                                     <input class="btn btn-lg btn-primary btn-block"

```

```
type="submit" name="insert_comment" value=
"Ajouter le commentaire">
```

```
</fieldset>
```

```
</form>
```

```
<?php
```

```
// Displays the error messages.
```

```
echo $comment_error;
```

```
?>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

```

1  <!-------
2  * Author : Philippe Ku
3  * School / Class : CFPT Informatique / I.FA-P3B
4  * Date : 15.06.2016
5  * Programm : Event gestion website
6  * File : myaccount.php
7  * Description : The personal account page of the website
8  * Version : 1.10
9  ----->
10 <?php
11 require 'functions/functions.php';
12 require 'controller.php';
13 // The user id.
14 $datas['user_id'] = getUserId();
15 ?>
16 <!DOCTYPE html>
17 <html>
18     <head>
19         <meta charset="UTF-8">
20         <meta http-equiv="X-UA-Compatible" content="IE=edge">
21         <meta name="viewport" content="width=device-width, initial-scale=1">
22         <title>Mon compte</title>
23         <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
24         <script src="
25             https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
26         <script src="bootstrap/js/bootstrap.min.js"></script>
27         <script type="text/javascript">
28             // Function in JavaScript that generate a message for the delete
29             confirmation.
30             function confirmDeletion() {
31                 return (confirm("Voulez-vous supprimer cette entrée ?"));
32             }
33         </script>
34     </head>
35     <body>
36         <?php
37             // If the user is not a member or an admin.
38             if (!isUserAdmin() && !isUserMember()) {
39                 // Redirect to the index.
40                 header('location: index.php');
41             }
42         ?>
43         <div class="container col-lg-11" style="padding-left: 150px;">
44             <div class="row col-md">
45                 <h1>Mon profil</h1><br/>
46             </div>
47             <div class="row">
48                 <table class="table table-striped table-bordered">
49                     <thead>
50                         <tr>
51                             <th>Email</th>
52                             <th>Pseudo</th>
53                             <th>Nom</th>
54                             <th>Prénom</th>
55                             <th>Organisation</th>
56                             <th>Adresse</th>
57                             <th>Avatar</th>

```

```
56         <th>Description</th>
57     </tr>
58 </thead>
59 <tbody>
60     <?php
61         // Displays the user's profile.
62         displayUserProfile($datas);
63     ?>
64 </tbody>
65 </table>
66 <?php echo ' <a href="edit_profile.php" class="btn btn-primary
pull-right">Modifier mon profil</a><br/>' ; ?>
67 </div>
68 <div class="row col-md">
69     <h1>Mes événements</h1><br/>
70 </div>
71 <div class="row">
72     <table class="table table-striped table-bordered">
73         <thead>
74             <tr>
75                 <th>Titre</th>
76                 <th>Rue</th>
77                 <th>Ville</th>
78                 <th>Pays</th>
79                 <th>Date de début</th>
80                 <th>Date de fin</th>
81                 <th>Description</th>
82                 <th>Image</th>
83                 <th>Gestion</th>
84             </tr>
85         </thead>
86         <tbody>
87             <?php
88                 // Displays all the user's events.
89                 displayAllUserEvents($datas);
90             ?>
91         </tbody>
92     </table>
93 </div>
94 </div>
95 </body>
96 </html>
```

```

1  <!-------
2  * Author : Philippe Ku
3  * School / Class : CFPT Informatique / I.FA-P3B
4  * Date : 15.06.2016
5  * Programm : Event gestion website
6  * File : edit_event.php
7  * Description : The edit event page of the website
8  * Version : 1.10
9  ----->
10 <?php
11 require 'functions/functions.php';
12 require 'controller.php';
13 // Get the id from the url.
14 $datas['event_id'] = $_GET['event_id'];
15 // Get the informations of the event from the dsatabase.
16 $results = getEventInfo($datas);
17 ?>
18 <!DOCTYPE html>
19 <html>
20     <head>
21         <meta charset="UTF-8">
22         <meta http-equiv="X-UA-Compatible" content="IE=edge">
23         <meta name="viewport" content="width=device-width, initial-scale=1">
24         <title>Modifier un événement</title>
25         <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
26         <script src="
27             https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
28         <script src="bootstrap/js/bootstrap.min.js"></script>
29         <!-- Script that adds TinyMCE, a JavaScript text editor for the textarea. -->
30         <script src="//cdn.tinymce.com/4/tinymce.min.js"></script>
31         <script>tinymce.init({selector: 'textarea'});</script>
32     </head>
33     <body>
34         <?php
35         // If the user is not a member or an admin.
36         if (!isUserAdmin() && !isUserMember()) {
37             // Redirect to the index.
38             header('location: index.php');
39         }
40         <?>
41         <div class="container">
42             <div class="row col-md-offset-0">
43                 <h1>Modifier un événement</h1><br/>
44             </div>
45             <div class="row">
46                 <div class="col-md-4 col-md-offset-2 pull-left">
47                     <div class="panel panel-default" style="width: 800px;">
48                         <div class="panel-heading">
49                             <h3 class="panel-title">Modifier un événement</h3>
50                         </div>
51                         <div class="panel-body" style="width: 800px;">
52                             <form method="POST" action="#" enctype=
53                             "multipart/form-data">
54                                 <fieldset>
55                                     <div class="form-group">
56                                         <label>Titre </label>
57                                         <input class="form-control" placeholder=

```

```

56         "Titre..." name="editevent_title" type="text"
57         value="<?php echo $results[0]['title']; ?>"
58         required=""
59     </div>
60     <div class="form-group">
61         <label>Rue :</label>
62         <input class="form-control" placeholder=
63         "Rue..." name="editevent_street" type="text"
64         value="<?php echo $results[0]['street']; ?>"
65         required=""
66     </div>
67     <div class="form-group">
68         <label>Ville :</label>
69         <input class="form-control" placeholder=
70         "Ville..." name="editevent_city" type="text"
71         value="<?php echo $results[0]['city']; ?>"
72         required=""
73     </div>
74     <div class="form-group">
75         <label>Pays :</label>
76         <select class="form-control" name=
77         "editevent_country" required=""
78             <option>Sélectionnez un pays</option>
79             <?php getCountriesList(); ?>
80         </select>
81     </div>
82     <div class="form-group">
83         <label>Date de début :</label>
84         <input class="form-control" name=
85         "editevent_datestart" type="date" value="
86         <?php echo $results[0]['dateStart']; ?>"
87         required=""
88     </div>
89     <div class="form-group">
90         <label>Date de fin :</label>
91         <input class="form-control" name=
92         "editevent_dateend" value="<?php echo
93         $results[0]['dateEnd']; ?>" type="date">
94     </div>
95     <div class="form-group">
96         <label>Description :</label>
97         <textarea class="form-control" name=
98         "editevent_desc" cols="50" rows="10" style=
99         "max-width: 100%;"><?php echo $results[0][
100         'description']; ?></textarea>
101     </div>
102     <div class="form-group">
103         <label>Image :</label>
104         <input class="form-control" name="<?php echo
105         INPUT; ?>" type="file" required="">
106     </div>
107     <input class="btn btn-lg btn-primary btn-block"
108     type="submit" name="edit_event" value="Modifier
109     l'événement">
110 </fieldset>
111 </form>
112 <?php

```



```
92         // Displays the error messages.
93         echo $editdate_error;
94     ?>
95     </div>
96 </div>
97 </div>
98 </div>
99 </div>
100 </body>
101 </html>
```

```

1  <!-------
2  * Author : Philippe Ku
3  * School / Class : CFPT Informatique / I.FA-P3B
4  * Date : 15.06.2016
5  * Programm : Event gestion website
6  * File : edit_profile.php
7  * Description : The edit profile page of the website
8  * Version : 1.10
9  ----->
10 <?php
11 require 'functions/functions.php';
12 require 'controller.php';
13 // Gets the user infos from the database.
14 $results = getUserInfo();
15 ?>
16 <!DOCTYPE html>
17 <html>
18     <head>
19         <meta charset="UTF-8">
20         <meta http-equiv="X-UA-Compatible" content="IE=edge">
21         <meta name="viewport" content="width=device-width, initial-scale=1">
22         <title>Modifier mon profil</title>
23         <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
24         <script src="
25             https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
26         <script src="bootstrap/js/bootstrap.min.js"></script>
27         <!-- Script that adds TinyMCE, a JavaScript text editor for the textarea. -->
28         <script src="//cdn.tinymce.com/4/tinymce.min.js"></script>
29         <script>tinymce.init({selector: 'textarea'});</script>
30     </head>
31     <body>
32         <?php
33         // If the user is not a member or an admin.
34         if (!isUserAdmin() && !isUserMember()) {
35             // Redirect to the index.
36             header('location: index.php');
37         }
38         ?>
39         <div class="container">
40             <div class="row col-md-offset-0">
41                 <h1>Modifier mon profil</h1><br/>
42             </div>
43             <div class="row">
44                 <div class="col-md-4 col-md-offset-2 pull-left">
45                     <div class="panel panel-default" style="width: 800px;">
46                         <div class="panel-heading">
47                             <h3 class="panel-title">Modifier mon profil</h3>
48                         </div>
49                         <div class="panel-body" style="width: 800px;">
50                             <form method="POST" action="#" enctype=
51                             "multipart/form-data">
52                                 <fieldset>
53                                     <div class="form-group">
54                                         <label>Nom :</label>
55                                         <input class="form-control" placeholder=
56                                         "Nom..." name="edit_name" type="text" value="
57                                         <?php echo $results[0]['name']; ?>" required=

```

```
54         ">
55     </div>
56     <div class="form-group">
57         <label>Prénom :</label>
58         <input class="form-control" placeholder=
59             "Prénom..." name="edit_firstname" type="text"
60             value="<?php echo $results[0]['firstname'];
61             ?>" required="">
62     </div>
63     <div class="form-group">
64         <label>Organisation :</label>
65         <input class="form-control" placeholder=
66             "Organisation..." name="edit_organisation"
67             type="text" value="<?php echo $results[0][
68             'organisation']; ?>" required="">
69     </div>
70     <div class="form-group">
71         <label>Adresse :</label>
72         <input class="form-control" placeholder=
73             "Adresse..." name="edit_adress" type="text"
74             value="<?php echo $results[0]['adress']; ?>"
75             required="">
76     </div>
77     <div class="form-group">
78         <label>Avatar :</label>
79         <input class="form-control" name="<?php echo
80             INPUT; ?>" type="file" required="">
81     </div>
82     <div class="form-group">
83         <label>Description :</label>
84         <textarea class="form-control" name=
85             "edit_desc" cols="50" rows="10" style=
86             "max-width: 100%;"><?php echo $results[0][
87             'description']; ?></textarea>
88     </div>
89     <input class="btn btn-lg btn-primary btn-block"
90         type="submit" name="edit_profile" value=
91         "Modifier mon profil">
92 </fieldset>
93 </form>
94 </div>
95 </div>
96 </div>
97 </div>
98 </div>
99 </body>
100 </html>
```

```

1  <!-------
2  * Author : Philippe Ku
3  * School / Class : CFPT Informatique / I.FA-P3B
4  * Date : 15.06.2016
5  * Programm : Event gestion website
6  * File : administration.php
7  * Description : The administration page of the website (reserved to administrators)
8  * Version : 1.10
9  ----->
10 <?php
11 require 'functions/functions.php';
12 require 'controller.php';
13 ?>
14 <!DOCTYPE html>
15 <html>
16     <head>
17         <meta charset="UTF-8">
18         <meta http-equiv="X-UA-Compatible" content="IE=edge">
19         <meta name="viewport" content="width=device-width, initial-scale=1">
20         <title>Administration</title>
21         <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
22         <script src="
23             https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
24         <script src="bootstrap/js/bootstrap.min.js"></script>
25         <script type="text/javascript">
26             // Function that generate a message for the delete confirmation.
27             function confirmDeletion() {
28                 return (confirm("Voulez-vous supprimer cette entrée ?"));
29             }
30         </script>
31     </head>
32     <body>
33         <?php
34         // If the user is not an admin.
35         if (!isUserAdmin()) {
36             // Redirect to the index page.
37             header('location: index.php');
38         }
39         ?>
40         <div class="container col-lg-12">
41             <div class="row col-md">
42                 <h1>Mon profil</h1><br/>
43             </div>
44             <div class="row">
45                 <table class="table table-striped table-bordered">
46                     <thead>
47                         <tr>
48                             <th>Email</th>
49                             <th>Pseudo</th>
50                             <th>Nom</th>
51                             <th>Prénom</th>
52                             <th>Organisation</th>
53                             <th>Adresse</th>
54                             <th>avatar</th>
55                             <th>Description</th>
56                             <th>Gestion</th>

```

```
57         </thead>
58         <tbody>
59             <?php
60                 // Display all the users from the database.
61                 displayAllUsers();
62             ?>
63         </tbody>
64     </table>
65 </div>
66 <div class="row col-md">
67     <h1>Gérer les événements</h1><br/>
68 </div>
69 <div class="row">
70     <table class="table table-striped table-bordered">
71         <thead>
72             <tr>
73                 <th>Titre</th>
74                 <th>Rue</th>
75                 <th>Ville</th>
76                 <th>Pays</th>
77                 <th>Date de début</th>
78                 <th>Date de fin</th>
79                 <th>Description</th>
80                 <th>Image</th>
81                 <th>Gestion</th>
82             </tr>
83         </thead>
84         <tbody>
85             <?php
86                 // Displays all the events from the database.
87                 displayAllAdminEvents();
88             ?>
89         </tbody>
90     </table>
91 </div>
92 </div>
93 </body>
94 </html>
```