

```
1  <?php
2
3  /*
4   Fichier: function.php
5   Auteur: Kevin Zaffino
6   Date: 15/06/2016
7   Version:1.10
8   Description: Contient les différentes fonctions de traitement
9   Copyright (TPI 2016 - Kevin Zaffino © 2016)
10  */
11
12  require_once './phpScript/mysql.inc.php';
13
14  /**
15   * Créé un pointeur sur la base
16   * @staticvar type $dbc
17   * @return PDO
18   */
19  function &myDatabase() {
20      static $dbc = null;
21
22      if ($dbc == null) {
23          try {
24              $dbc = new PDO('mysql:host=' . DB_HOST . ';dbname=' . DB_NAME, DB_USER,
25                  DB_PASSWORD, array(PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8", PDO
26                      ::ATTR_PERSISTENT => true));
27          } catch (Exception $e) {
28              echo 'Erreur : ' . $e->getMessage() . '<br />';
29              echo 'N : ' . $e->getCode();
30              die('Could not connect to MySQL');
31          }
32      }
33      return $dbc;
34  }
35
36  //Gestion des Utilisateurs
37
38  /**
39   * Ajoute une adresse dans la base
40   * @staticvar type $ps_adress
41   * @staticvar type $ps_id
42   * @param string $country
43   * @param string $city
44   * @param string $street
45   * @return int => le dernier id ajouté dans la base
46   */
47  function addAddress($country, $city, $street) {
48      //Prepared statements
49      static $ps_adress = null;
50      static $ps_id = null;
51
52      //query
53      $sql_adress = 'insert into adress (adress.city,adress.street,adress.country_iso)
54          values (:city,:street,:iso)';
```

```

55     if ($ps_adress == null) {
56         $ps_adress = myDatabase()->prepare($sql_adress);
57     }
58     if ($ps_id == null) {
59         $ps_id = myDatabase()->prepare('SELECT LAST_INSERT_ID()');
60     }
61
62     try {
63         $ps_adress->bindParam(':city', $city, PDO::PARAM_STR);
64         $ps_adress->bindParam(':street', $street, PDO::PARAM_STR);
65         $ps_adress->bindParam(':iso', $country, PDO::PARAM_STR);
66         $ps_adress->execute();
67
68         $ps_id->execute();
69         $last_id = $ps_id->fetchAll(PDO::FETCH_NUM);
70
71         return $last_id[0][0];
72     } catch (PDOException $e) {
73         $isok = null;
74     }
75     return $isok;
76 }
77
78 /**
79  * Insere un nouvel utilisateur dans la base de données
80  * @param string $lastName      Le nom de famille
81  * @param string $firstName     Le prénom
82  * @param string $gender        Le genre
83  * @param string $mail          Le email
84  * @param string $pwd           Le mot de passe pas encore crypté
85  * @param string $phone         Le numero de telephone
86  * @param string $country       Le code ISO du pays
87  * @param string $city          Le nom de la ville
88  * @param string $street        Le nom de la rue
89  * @param int $image            L'id de l'image
90  * @return boolean True si correctement ajouté, autrement False.
91  */
92 function insertUser($lastName, $firstName, $gender, $mail, $pwd, $phone, $country,
93 $city, $street, $image = DEFAULT_IMAGE_ID) {
94     //Prepared statements
95     static $ps_user = null;
96
97     //Query
98     $sql_user = 'insert into users
99 (users.firstname,users.lastname,users.gender,users.mail,users.phone,users.banned,u
100 sers.password,users.id_Images,users.id_Adress, users.privilege) values
101 (:firstname,:lastname,:gender,:mail,:phone,0,:password,:idimage,:idaddress,
102 :privilege)';
103
104     if ($ps_user == null) {
105         $ps_user = myDatabase()->prepare($sql_user);
106     }
107
108     //ajout et récupération de l'identifiant de l'adresse
109     $id_adress = intval(addAddress($country, $city, $street));

```

```

107     //cryptage du mot de passe
108     $pwd_shal = sha1($pwd);
109
110     $priv = PRIV_USER;
111
112     try {
113
114         $sps_user->bindParam(':firstname', $firstName, PDO::PARAM_STR);
115         $sps_user->bindParam(':lastname', $lastName, PDO::PARAM_STR);
116         $sps_user->bindParam(':gender', $gender, PDO::PARAM_INT);
117         $sps_user->bindParam(':mail', $mail, PDO::PARAM_STR);
118         $sps_user->bindParam(':phone', $phone, PDO::PARAM_STR);
119         $sps_user->bindParam(':password', $pwd_shal, PDO::PARAM_STR);
120         $sps_user->bindParam(':idimage', $image, PDO::PARAM_INT);
121         $sps_user->bindParam(':idadress', $id_adress, PDO::PARAM_INT);
122         $sps_user->bindParam(':privilege', $priv, PDO::PARAM_INT);
123
124         $isok = $sps_user->execute();
125     } catch (PDOException $e) {
126         $isok = false;
127     }
128     return $isok;
129 }
130
131 /**
132  * Verification des identifiants de l'utilisateur
133  * @staticvar type $ps
134  * @param string $mail      L'adresse email
135  * @param string $pwd        Le mot de passe pas encore crypté
136  * @return boolean
137  */
138 function login($mail, $pwd) {
139
140     //Prepared statement
141     static $ps = null;
142
143     //Query
144     $sql = 'select users.id, users.password from users where users.mail = :mail';
145
146     if ($ps == null) {
147         $ps = myDatabase()->prepare($sql);
148     }
149
150     //cryptage du mot de passe
151     $pwd_shal = sha1($pwd);
152
153     try {
154         $ps->bindParam(':mail', $mail, PDO::PARAM_STR);
155         $isok = $ps->execute();
156         $isok = $ps->fetchAll(PDO::FETCH_ASSOC);
157         if (isset($isok[0])) {
158             if ($isok[0]['password'] == $pwd_shal) {
159                 $isok = intval($isok[0]['id']);
160             } else {
161                 $isok = FALSE;
162             }
163         } else {

```

```

164         $isok = FALSE;
165     }
166 } catch (PDOException $e) {
167     $isok = false;
168 }
169 return $isok;
170 }
171
172 /**
173  * Deconnecte l'utilisateur
174  */
175 function logout() {
176     destroySession();
177     header('Location: index.php');
178 }
179
180 /**
181  * Retourne un tableau avec les infos de l'utilisateur
182  * @staticvar type $ps
183  * @param int $id    Id de l'utilisateur
184  * @return Array     Tableau associatif contenant les données de l'utilisateur
185  */
186 function getUserInfo($id) {
187     //Prepared statement
188     static $ps = null;
189
190     //Query
191     $sql = 'select * from user_info where user_info.id = :id';
192
193     if ($ps == null) {
194         $ps = myDatabase()->prepare($sql);
195     }
196
197     try {
198         $ps->bindParam(':id', $id, PDO::PARAM_INT);
199
200         $isok = $ps->execute();
201         $isok = $ps->fetchAll(PDO::FETCH_ASSOC);
202         $isok = $isok[0];
203     } catch (PDOException $e) {
204         $isok = false;
205     }
206     return $isok;
207 }
208
209 /**
210  * Inverse l'etat de bannissement de l'utilisateur
211  * @staticvar type $ps
212  * @param int $uid    Identifiant de l'utilisateur
213  * @return boolean
214  */
215 function banUnbanUser($uid) {
216     $banstate = getUserInfo($uid)['banned'];
217     $finalestate = ($banstate == "0") ? 1 : 0;
218
219     //Prepared statement
220     static $ps = null;

```

```

221
222     //Query
223     $sql = 'UPDATE `users` SET `banned` = :ban WHERE `id` = :id';
224     if ($ps == null) {
225         $ps = myDatabase()->prepare($sql);
226     }
227     try {
228         $ps->bindParam(':ban', $finalestate, PDO::PARAM_INT);
229         $ps->bindParam(':id', $uid, PDO::PARAM_INT);
230         $isok = $ps->execute();
231     } catch (PDOException $e) {
232         $isok = false;
233     }
234     return $isok;
235 }
236
237 /**
238  * Modifie les informations de l'utilisateur
239  * @staticvar type $ps
240  * @param string $lastName      Nom de l'utilisateur
241  * @param string $firstName     Prename de l'utilisateur
242  * @param Bool $gender          Genre de l'utilisateur (0 pour Femme et 1 pour Homme)
243  * @param string $mail          Email de l'utilisateur
244  * @param string $phone         Numero de telephone de l'utilisateur
245  * @param int $id               Identifiant d el'utilisateur
246  * @return boolean
247  */
248 function updateUserInfo($lastName, $firstName, $gender, $mail, $phone, $id) {
249
250     //Prepared statement
251     static $ps = null;
252
253     //Query
254     $sql = "update users set users.firstname = :firstname ,users.lastname =
:lastname ,users.gender = :gender ,users.mail = :mail ,users.phone = :phone
where users.id = :id";
255
256     if ($ps == null) {
257         $ps = myDatabase()->prepare($sql);
258     }
259
260     try {
261         $ps->bindParam(':firstname', $firstName, PDO::PARAM_STR);
262         $ps->bindParam(':lastname', $lastName, PDO::PARAM_STR);
263         $ps->bindParam(':gender', $gender, PDO::PARAM_INT);
264         $ps->bindParam(':mail', $mail, PDO::PARAM_STR);
265         $ps->bindParam(':phone', $phone, PDO::PARAM_STR);
266
267         $ps->bindParam(':id', $id, PDO::PARAM_INT);
268
269         $isok = $ps->execute();
270     } catch (PDOException $e) {
271         $isok = false;
272     }
273     return $isok;
274 }
275

```

```
276  /**
277  * Modifie le chemin de l'avatar de l'utilisateur
278  * @staticvar type $ps
279  * @param int $id
280  * @param type $path
281  * @return boolean
282  */
283  function updateUserImage($id, $path) {
284      static $ps = null;
285
286      $sql = "update images set images.path = :path where images.id = :id";
287
288      if ($ps == null) {
289          $ps = myDatabase()->prepare($sql);
290      }
291
292      try {
293          $ps_user->bindParam(':path', $city, PDO::PARAM_STR);
294          $ps_user->bindParam(':id', $id, PDO::PARAM_INT);
295
296          $isok = $ps_user->execute();
297      } catch (PDOException $e) {
298          $isok = false;
299      }
300      return $isok;
301  }
302
303  /**
304  * Modifie l'adresse d'un utilisateur
305  * @staticvar type $ps
306  * @param int $id                Identifiant de l'utilisateur
307  * @param string $countryiso      Code ISO du pays
308  * @param string $city            Nom de la ville
309  * @param string $street          Nom de la rue
310  * @return boolean
311  */
312  function updateUserAdress($id, $countryiso, $city, $street) {
313
314      //Prepared stateent
315      static $ps = null;
316
317      //Query
318      $sql = "update adress set adress.city = :city ,adress.street = :street
319      ,adress.country_iso = :iso where adress.id = :id";
320
321      if ($ps == null) {
322          $ps = myDatabase()->prepare($sql);
323      }
324
325      try {
326
327          $ps->bindParam(':city', $city, PDO::PARAM_STR);
328          $ps->bindParam(':street', $street, PDO::PARAM_STR);
329          $ps->bindParam(':iso', $countryiso, PDO::PARAM_STR);
330          $ps->bindParam(':id', $id, PDO::PARAM_INT);
331
332          $isok = $ps->execute();
```

```
332     } catch (PDOException $e) {
333         $isok = false;
334     }
335     return $isok;
336 }
337
338 /**
339  * Retourne un tableau avec tous les utilisateurs
340  * @staticvar type $ps
341  * @return array      Tableau associatif contenant tous les utilisateurs
342  */
343 function getAllUser() {
344     static $ps = null;
345
346     $sql = "SELECT id, firstname as Prenom, lastname as Nom, mail, banned as Bannis
347     FROM user_info WHERE privilege=1;";
348
349     if ($ps == null) {
350         $ps = myDatabase()->prepare($sql);
351     }
352
353     try {
354         $isok = $ps->execute();
355         $isok = $isok = $ps->fetchAll(PDO::FETCH_ASSOC);
356     } catch (PDOException $e) {
357         $isok = false;
358     }
359
360     return $isok;
361 }
362
363 /**
364  * Modifie l'id de l'image d'un utilisateur avec un id donné
365  * @staticvar type $ps
366  * @param int $user_id      Identifiant d el'utilisateur
367  * @param int $image_id      Identifiant d el'image a attribuer
368  * @return boolean
369  */
370 function changeUserImage($user_id, $image_id) {
371     static $ps = null;
372
373     $sql = "update users set users.id_Images = :iid where users.id = :uid";
374
375     if ($ps == null) {
376         $ps = myDatabase()->prepare($sql);
377     }
378
379     try {
380         $ps->bindParam(':iid', $image_id, PDO::PARAM_INT);
381         $ps->bindParam(':uid', $user_id, PDO::PARAM_INT);
382
383         $isok = $ps->execute();
384     } catch (PDOException $ex) {
385         $isok = false;
386     }
387
388     return $isok;
389 }
```

```

388 }
389
390 /**
391  * Récupere tous les pays
392  * @staticvar type $ps
393  * @return boolean
394  */
395 function getAllCountry() {
396
397     //Prepared statement
398     static $ps = null;
399
400     //Query
401     $sql = 'select * from country order by name';
402
403     if ($ps == null) {
404         $ps = myDatabase()->prepare($sql);
405     }
406
407     try {
408         $isok = $ps->execute();
409         $isok = $isok = $ps->fetchAll(PDO::FETCH_ASSOC);
410     } catch (PDOException $e) {
411         $isok = false;
412     }
413     return $isok;
414 }
415
416 //Gestion des annonces
417
418 /**
419  * Insert une nouvelle annonce dans la base de donnée
420  * @staticvar type $ps
421  * @param string $name          libelle de l'annonce
422  * @param string $description    Description de l'annonce
423  * @param string $price          Prix de l'annonce
424  * @param string $date           Date de creation de l'annonce
425  * @param string $uid            Identifiant du créateur de l'annonce
426  * @param bool $mailvisible      Email visible ou non
427  * @param bool $phonevisible     Numero de telephone visible ou non
428  * @param bool $adressvisible    Adresse visible ou non
429  * @return boolean
430  */
431 function insertArticle($name, $description, $price, $date, $uid, $mailvisible,
432 $phonevisible, $adressvisible) {
433
434     //Prepared statements
435     static $ps = null;
436     static $ps_id = null;
437
438     //Query
439     $sql = 'insert into articles
440 (name,description,price,state,creationdate,banned,id_Users,mailvisible,phonevisibl
441 e,adressvisible) values
442 (:name,:description,:price,1,:date,0,:uid,:mvis,:pvis,:avis)';

```



```

441     if ($ps == null) {
442         $ps = myDatabase()->prepare($sql);
443     }
444
445     if ($ps_id == null) {
446         $ps_id = myDatabase()->prepare('SELECT LAST_INSERT_ID()');
447     }
448
449
450     try {
451         $ps->bindParam(':name', $name, PDO::PARAM_STR);
452         $ps->bindParam(':description', $description, PDO::PARAM_STR);
453         $ps->bindParam(':price', $price, PDO::PARAM_STR);
454         $ps->bindParam(':date', $date, PDO::PARAM_STR);
455         $ps->bindParam(':uid', $uid, PDO::PARAM_INT);
456         $ps->bindParam(':mvis', $mailvisible, PDO::PARAM_INT);
457         $ps->bindParam(':pvis', $phonevisible, PDO::PARAM_INT);
458         $ps->bindParam(':avis', $adressvisible, PDO::PARAM_INT);
459
460
461         $isok = $ps->execute();
462
463         $ps_id->execute();
464         $last_id = $ps_id->fetchAll(PDO::FETCH_NUM);
465
466         return intval($last_id[0][0]);
467     } catch (PDOException $e) {
468         $isok = false;
469     }
470     return $isok;
471 }
472
473 /**
474  * Edition des informations d'une annonce
475  * @staticvar type $ps
476  * @param int $idarticle      Identifiant de l'annonce a editer
477  * @param string $name        Libelle del'annonce
478  * @param string $description  Description del'annonce
479  * @param string $price       Prix de l'annonce
480  * @param bool $mailvisible   Email visible ou non
481  * @param bool $phonevisible  Numero de telephone visible ou non
482  * @param bool $adressvisible  Adresse visible ou non
483  * @return boolean
484  */
485 function editArticleInfo($idarticle, $name, $description, $price, $mailvisible,
486 $phonevisible, $adressvisible) {
487     //Prepared statement
488     static $ps = null;
489
490     //Query
491     $sql = 'update articles set name=:name ,description=:description ,price=:price
492 ,mailvisible=:mvis, phonevisible=:pvis ,adressvisible =:avis where id=:id';
493
494     if ($ps == null) {
495         $ps = myDatabase()->prepare($sql);
496     }

```

```
496
497     try {
498
499         $ps->bindParam(':name', $name, PDO::PARAM_STR);
500         $ps->bindParam(':description', $description, PDO::PARAM_STR);
501         $ps->bindParam(':price', $price, PDO::PARAM_STR);
502         $ps->bindParam(':mvis', $mailvisible, PDO::PARAM_INT);
503         $ps->bindParam(':pvis', $phonevisible, PDO::PARAM_INT);
504         $ps->bindParam(':avis', $adressvisible, PDO::PARAM_INT);
505         $ps->bindParam(':id', $idarticle, PDO::PARAM_INT);
506
507
508         $isok = $ps->execute();
509
510         return $isok;
511     } catch (PDOException $e) {
512         $isok = false;
513     }
514     return $isok;
515 }
516
517 /**
518  * Insert une image de l'article dans la base de données
519  * @staticvar type $ps
520  * @param int $aid      Identifiant de l'article
521  * @param int $iid      Identifiant de l'image
522  * @return boolean
523  */
524 function insertArticleImage($aid, $iid) {
525     //Prepared statement
526     static $ps = null;
527
528     //Query
529     $sql = "update images set images.id_articles = :articleid where images.id =
530         :imageid";
531
532     if ($ps == null) {
533         $ps = myDatabase()->prepare($sql);
534     }
535
536     try {
537         $ps->bindParam(':articleid', $aid, PDO::PARAM_INT);
538         $ps->bindParam(':imageid', $iid, PDO::PARAM_INT);
539         $isok = $ps->execute();
540     } catch (PDOException $e) {
541         $isok = false;
542     }
543
544     return $isok;
545 }
546
547 /**
548  * Retourne les images des articles en fonction de l'id de l'article
549  * @staticvar type $ps
550  * @param int $idarticle      Identifiant de l'article
551  * @return boolean
552  */
```

```

552 function articleImages($idarticle) {
553     //Prepared statement
554     static $ps = null;
555
556     //Query
557     $sql = 'SELECT id, path FROM images where images.id_articles = :id';
558
559     if ($ps == null) {
560         $ps = myDatabase()->prepare($sql);
561     }
562
563     try {
564         $ps->bindParam(':id', $idarticle, PDO::PARAM_INT);
565         $isok = $ps->execute();
566         $isok = $isok = $ps->fetchAll(PDO::FETCH_ASSOC);
567     } catch (PDOException $e) {
568         $isok = false;
569     }
570     return $isok;
571 }
572
573 /**
574  * Ouvre/Ferme un article
575  * @staticvar type $ps
576  * @param int $aid      Identifiant del'article
577  * @return boolean
578  */
579 function openCloseArticle($aid) {
580     //Recuperation des infos de l'article
581     $banstate = articleInfo($aid)['state'];
582     $finalestate = ($banstate == "0") ? 1 : 0;
583
584     //prepared statement
585     static $ps = null;
586
587     //Query
588     $sql = 'UPDATE `articles` SET `state` = :state WHERE `id` = :id';
589     if ($ps == null) {
590         $ps = myDatabase()->prepare($sql);
591     }
592     try {
593         $ps->bindParam(':state', $finalestate, PDO::PARAM_INT);
594         $ps->bindParam(':id', $aid, PDO::PARAM_INT);
595         $isok = $ps->execute();
596     } catch (PDOException $e) {
597         $isok = false;
598     }
599     return $isok;
600 }
601
602 /**
603  * Ban/Unban un article
604  * @staticvar type $ps
605  * @param int $aid      Identifiant de l'article
606  * @return boolean
607  */
608 function banunbanArticle($aid) {

```

```
609 //Recuperation de l'etat actuel
610 $banstate = articleInfo($aid)['banned'];
611
612 //inversement de l'etat
613 $finalestate = ($banstate == "0") ? 1 : 0;
614
615 //Prepared statement
616 static $ps = null;
617
618 //Query
619 $sql = "UPDATE `articles` SET `banned` = :ban WHERE `id` = :id";
620
621 if ($ps == null) {
622     $ps = myDatabase()->prepare($sql);
623 }
624 try {
625     $ps->bindParam(':ban', $finalestate, PDO::PARAM_INT);
626     $ps->bindParam(':id', $aid, PDO::PARAM_INT);
627     $isok = $ps->execute();
628 } catch (PDOException $e) {
629     $isok = false;
630 }
631 return $isok;
632 }
633
634 /**
635  * Recupere toutes les annonces de la base
636  * @staticvar type $ps
637  * @return boolean
638  */
639 function getAllArticles() {
640
641     //Prepared statement
642     static $ps = null;
643
644     //Query
645     $sql = 'select id, name as Libelle,description,price as Prix,state as
Etat,creationdate as "Date de creation",banned as Bannis from articles';
646
647     if ($ps == null) {
648         $ps = myDatabase()->prepare($sql);
649     }
650
651     try {
652         $isok = $ps->execute();
653         $isok = $isok = $ps->fetchAll(PDO::FETCH_ASSOC);
654     } catch (PDOException $e) {
655         $isok = false;
656     }
657
658     return $isok;
659 }
660
661 /**
662  * Recupere tous les articles qui sont actif et non bannis
663  * @staticvar type $ps
664  * @return boolean
```

```
665     */
666     function listArticles() {
667
668         //Prepared statement
669         static $ps = null;
670
671         //Query
672         $sql = 'SELECT * FROM articles where state = 1 and banned = 0 order by
        creationdate DESC';
673
674         if ($ps == null) {
675             $ps = myDatabase()->prepare($sql);
676         }
677
678         try {
679             $isok = $ps->execute();
680             $isok = $isok = $ps->fetchAll(PDO::FETCH_ASSOC);
681         } catch (PDOException $e) {
682             $isok = false;
683         }
684
685         return $isok;
686     }
687
688     /**
689     * Recupere toutes les annonces d'un utilisateur en particulier
690     * @staticvar type $ps
691     * @param int $uid          Identifiant dle'utilisateur
692     * @return boolean
693     */
694     function getUserArticles($uid) {
695
696         //Prepared statement
697         static $ps = null;
698
699         //Query
700         $sql = 'SELECT * FROM articles where banned = 0 and articles.id_Users = :id
        order by creationdate';
701
702         if ($ps == null) {
703             $ps = myDatabase()->prepare($sql);
704         }
705
706         try {
707             $ps->bindParam(':id', $uid, PDO::PARAM_INT);
708             $isok = $ps->execute();
709             $isok = $isok = $ps->fetchAll(PDO::FETCH_ASSOC);
710         } catch (PDOException $e) {
711             $isok = false;
712         }
713
714         return $isok;
715     }
716
717     /**
718     * Retourne les informations d'un article
719     * @staticvar type $ps
```

```

720  * @param int $id          $identifiant de l'article
721  * @return boolean
722  */
723  function articleInfo($id) {
724
725      //Prepared statement
726      static $ps = null;
727
728      //Query
729      $sql = 'select * from articles where id = :id';
730
731      if ($ps == null) {
732          $ps = myDatabase()->prepare($sql);
733      }
734
735      try {
736          $ps->bindParam(':id', $id, PDO::PARAM_INT);
737
738          $isok = $ps->execute();
739          $isok = $ps->fetchAll(PDO::FETCH_ASSOC);
740          $isok = $isok[0];
741      } catch (PDOException $e) {
742          $isok = false;
743      }
744
745      return $isok;
746  }
747
748  /**
749   * Supprime toutes les images d'un article dans la base de donnée
750   * @staticvar type $ps
751   * @param int $id_article      Identifiant del'article
752   */
753  function deleteArticleImages($id_article) {
754      $articleimage = articleImages($id_article);
755
756      foreach ($articleimage as $value) {
757          $id = intval($value['id']);
758          deleteFile(deleteImageEntry($id));
759      }
760  }
761
762  //Gestion des commentaires
763
764  /**
765   * Ajoute un commentaire dans la base
766   * @staticvar type $ps
767   * @param int $uid            Identifiant de l'utilisateur
768   * @param int $aid            Identifiant de l'article
769   * @param string $date        Date du commentaire
770   * @param string $com         Texte du commentaire
771   * @return boolean
772   */
773  function addComment($uid, $aid, $date, $com) {
774
775      //Prepared statement
776      static $ps = null;

```

```

777
778     //Query
779     $sql = "insert into comments
            (comments.id_Users,comments.id_articles,comments.date_com,comments.comm,comments.s
            tate,comments.banned) values (:uid,:aid,:date,:com,1,0)";

780     if ($ps == null) {
781         $ps = myDatabase()->prepare($sql);
782     }
783
784     try {
785         $ps->bindParam(':uid', $uid, PDO::PARAM_STR);
786         $ps->bindParam(':aid', $aid, PDO::PARAM_STR);
787         $ps->bindParam(':date', $date, PDO::PARAM_STR);
788         $ps->bindParam(':com', $com, PDO::PARAM_STR);
789
790         $isok = $ps->execute();
791     } catch (PDOException $e) {
792         $isok = false;
793     }
794     return $isok;
795 }
796
797 /**
798  * Recupere les commentaires d'une annonce
799  * @staticvar type $ps
800  * @param int $aid          Identifiant de l'annonce
801  * @return boolean
802  */
803 function getArticleComments($aid) {
804
805     //Prepared statement
806     static $ps = null;
807
808     //Query
809     $sql = "select * from comments where id_articles = :id and comments.banned = 0";
810     if ($ps == null) {
811         $ps = myDatabase()->prepare($sql);
812     }
813
814     try {
815         $ps->bindParam(':id', $aid, PDO::PARAM_INT);
816         $isok = $ps->execute();
817         $isok = $isok = $ps->fetchAll(PDO::FETCH_ASSOC);
818     } catch (PDOException $e) {
819         $isok = false;
820     }
821
822     return $isok;
823 }
824
825 /**
826  * met a jour l'etat du commentaire (en attente de moderation ou non)
827  * @staticvar type $ps
828  * @param int $idcomment    Identifiant du commentaire
829  * @param bool $state        Valeur a assigner
830  * @return boolean
831  */

```

```
832 function changeComState($idcomment, $state) {
833
834     //Prepared statement
835     static $ps = null;
836
837     //Query
838     $sql = 'update comments set comments.state = :state where comments.id = :id';
839
840     if ($ps == null) {
841         $ps = myDatabase()->prepare($sql);
842     }
843
844     try {
845         $ps->bindParam(':id', $idcomment, PDO::PARAM_INT);
846         $ps->bindParam(':state', $state, PDO::PARAM_INT);
847         $isok = $ps->execute();
848     } catch (PDOException $e) {
849         $isok = false;
850     }
851
852     return $isok;
853 }
854
855 /**
856  * Bannis un commentaire
857  * @staticvar type $ps
858  * @param int $idcomment      Identifiant du commentaire
859  * @return boolean
860  */
861 function banComment($idcomment) {
862
863     //Prepared statement
864     static $ps = null;
865
866     //Query
867     $sql = 'update comments set comments.banned = 1 where comments.id = :id';
868
869     if ($ps == null) {
870         $ps = myDatabase()->prepare($sql);
871     }
872
873     try {
874         $ps->bindParam(':id', $idcomment, PDO::PARAM_INT);
875         $isok = $ps->execute();
876     } catch (PDOException $e) {
877         $isok = false;
878     }
879
880     return $isok;
881 }
882
883 //Gestion des images
884
885 /**
886  * Insert une image dans la base de données et retourne son id
887  * @staticvar type $ps_image
888  * @staticvar type $ps_id
```



```
889  * @param string $path          Chemin de l'image
890  * @param int $idarticle        (a specifier si on souhaite assigner l'image a un
    article)
891  * @return int                  Identifiant de l'image
892  */
893  function insertImage($path, $idarticle = FALSE) {
894
895      //Prepared statements
896      static $ps_image = null;
897      static $ps_id = null;
898
899
900      //Query
901      if ($idarticle) {
902          $sql_image = 'insert into images (path,images.id_articles) values
    (:path,:id)';
903      } else {
904          $sql_image = 'insert into images (path) values (:path)';
905      }
906
907      $sql_id = 'select id from images where path = :path';
908
909      if ($ps_image == NULL) {
910          $ps_image = myDatabase()->prepare($sql_image);
911      }
912
913      if ($ps_id == NULL) {
914          $ps_id = myDatabase()->prepare($sql_id);
915      }
916
917      try {
918          $ps_image->bindParam(':path', $path, PDO::PARAM_STR);
919          if ($idarticle) {
920              $ps_image->bindParam(':id', $idarticle, PDO::PARAM_INT);
921          }
922
923          $ps_image->execute();
924
925          $ps_id->bindParam(':path', $path, PDO::PARAM_STR);
926
927          $isok = $ps_id->execute();
928          $isok = $ps_id->fetchAll(PDO::FETCH_NUM);
929      } catch (PDOException $ex) {
930          $isok = false;
931      }
932
933      return $isok[0][0];
934  }
935
936  /**
937   * Upload une image
938   * @return type
939   */
940  function imageUpload() {
941
942      //Extensions autorisées
943      $extArray = array('jpg', 'gif', 'png', 'jpeg');
```

```
944
945     $ext = '';
946     $error = '';
947     $imageName = '';
948
949 //Si le dossier cible n'existe pas, alors on essaye de le créer
950     if (!is_dir(TARGET)) {
951         if (!mkdir(TARGET, 0755)) {
952             exit('Erreur : le répertoire cible ne peut-être créé ! Vérifiez que vous
                diposiez des droits suffisants pour le faire ou créez le manuellement !');
953         }
954     }
955
956 //Verification si le champ n'est pas vide
957     if (!empty($_FILES[INPUT]['name'])) {
958
959 //recupère l'extension du fichier
960         $ext = pathinfo($_FILES[INPUT]['name'], PATHINFO_EXTENSION);
961
962 //verification de l'extension
963         if (in_array(strtolower($ext), $extArray)) {
964             $infosImg = getimagesize($_FILES[INPUT]['tmp_name']);
965
966
967 //Verification du type de l'image
968             if ($infosImg[2] >= 1 && $infosImg[2] <= 14) {
969
970
971 //Verification des dimensions et de la taille
972                 if (($infosImg[0] <= WIDTH_MAX) && ($infosImg[1] <= HEIGHT_MAX) && (
                    filesize($_FILES[INPUT]['tmp_name']) <= MAX_SIZE)) {
973
974                     if (isset($_FILES[INPUT]['error']) && UPLOAD_ERR_OK === $_FILES[
                        INPUT]['error']) {
975
976                         $imageName = uniqid() . '.' . $ext;
977
978
979                     if (move_uploaded_file($_FILES[INPUT]['tmp_name'], TARGET .
                        $imageName)) {
980                         $path = TARGET . $imageName;
981
982                         //insertion dans la base de donnée et retour de l'id
983                         return insertImage($path);
984                     } else {
985                         return 'Problème lors de l\'upload !';
986                     }
987                 } else {
988                     return 'Une erreur interne a empêché l\'upload de l\'image';
989                 }
990             } else {
991                 return 'Erreur dans les dimensions de l\'image !';
992             }
993         } else {
994             return 'Le fichier à uploader n\'est pas une image !';
995         }
996     } else {
```

```

997         return 'L\'extension du fichier est incorrecte !';
998     }
999 } else {
1000     return 'Image non renseignée!';
1001 }
1002 }
1003
1004 /**
1005  * Parcours le $_FILES pour envoyer toutes les images dans un article
1006  * @param int $id_article      Identifiant de l'article
1007  * @return boolean
1008  */
1009 function multiUpload($id_article) {
1010
1011     //Nombre d'image
1012     $nb = count($_FILES[INPUT]['name']);
1013     for ($i = 0; $i < $nb; $i++) {
1014         $name = uniqid();
1015         $extension_upload = strtolower(substr(strrchr($_FILES[INPUT]['name'][$i], '.'), 1));
1016         $destination = TARGET . $name . "." . $extension_upload;
1017         //deplacement des fichiers
1018         move_uploaded_file($_FILES[INPUT]['tmp_name'][$i], $destination);
1019         //insertion de l'image dans la base
1020         insertImage($destination, $id_article);
1021     }
1022     return TRUE;
1023 }
1024
1025 /**
1026  * Modifie le chemin d'une image dans la base d edonnée
1027  * @staticvar type $ps
1028  * @param type $id_image      Identifiant de l'image a modifier
1029  * @param type $newpath       Nouveau chemin
1030  * @return boolean
1031  */
1032 function editImagePath($id_image, $newpath) {
1033     static $ps = null;
1034
1035     $sql = "update images set images.path = :path where images.id = :id";
1036
1037     if ($ps == null) {
1038         $ps = myDatabase()->prepare($sql);
1039     }
1040
1041     try {
1042         $ps->bindParam(':path', $newpath, PDO::PARAM_STR);
1043         $ps->bindParam(':id', $id_image, PDO::PARAM_INT);
1044
1045         $isok = $ps->execute();
1046     } catch (PDOException $ex) {
1047         $isok = false;
1048     }
1049
1050     return $isok;
1051 }
1052

```

```
1053  /**
1054  * Supprime une image de la base de données
1055  * @staticvar type $ps_path
1056  * @staticvar type $ps_delete
1057  * @param int $image_id      Identifiant de l'image
1058  * @return boolean
1059  */
1060  function deleteImageEntry($image_id) {
1061
1062      static $ps_path = null;
1063      static $ps_delete = null;
1064
1065      $sql_path = "SELECT path FROM images where id=:id;";
1066      $sql_delete = "DELETE from images WHERE id=:id";
1067
1068
1069      if ($ps_path == null) {
1070          $ps_path = myDatabase()->prepare($sql_path);
1071      }
1072
1073      if ($ps_delete == null) {
1074          $ps_delete = myDatabase()->prepare($sql_delete);
1075      }
1076
1077      try {
1078          $ps_path->bindParam(':id', $image_id, PDO::PARAM_INT);
1079          $path = $ps_path->execute();
1080          $path = $ps_path->fetchAll(PDO::FETCH_ASSOC);
1081
1082
1083          $ps_delete->bindParam(':id', $image_id, PDO::PARAM_INT);
1084          $delete = $ps_delete->execute();
1085      } catch (PDOException $ex) {
1086          $path = false;
1087      }
1088
1089      return $path[0]['path'];
1090  }
1091
1092  /**
1093  * Supprime un fichier
1094  * @param string $path      Emplacement du fichier a supprimer
1095  */
1096  function deleteFile($path) {
1097      if (file_exists($path)) {
1098          unlink($path);
1099      }
1100  }
1101
```

```

1  <?php
2
3  /*
4      Fichier: fonctiondisp.php
5      Auteur: Kevin Zaffino
6      Date: 15/06/2016
7      Version:1.10
8      Description: Contient les fonction d'affichage du site
9      Copyright (TPI 2016 - Kevin Zaffino © 2016)
10  */
11
12  /**
13   * Retourne le nom et le prenom avec le format "Prenom N."
14   * @param string $firstname      Nom
15   * @param string $lastname       Prename
16   * @return (string) format Prename N.
17   */
18  function formatUserName($firstname, $lastname) {
19      return ucfirst($firstname) . " " . substr(ucfirst($lastname), 0, 1) . ".";
20  }
21
22  /**
23   * Retourne du code html pour afficher un article sur la page d'accueil
24   * @param Array $data            Donnée de l'utilisateur (id,name,description,date de
25   *                               creation, price)
26   * @param string $imgpath        Chemin de l'image à afficher
27   * @return string
28   */
29  function articleFormat($data, $imgpath) {
30
31      $creator_id = intval($data['id_Users']);
32      $creator_firstname = getUserInfo($creator_id)['firstname'];
33      $creator_lastname = getUserInfo($creator_id)['lastname'];
34      $creatorname = formatUserName($creator_firstname, $creator_lastname);
35
36      $output = "\n<li class=\"media well\">";
37      $output .= "\n<div class=\"pull-left col-lg-2\">";
38      $output .= "\n<a href=\"articles.php?idarticle=" . $data['id'] . "\"
39      class=\"thumbnail\">";
40      $output .= "<img alt=\"Image\" src=\"" . $imgpath . "\">";
41      $output .= "\n</a>";
42      $output .= "\n</div>";
43      $output .= "<div class=\"col-lg-6\">";
44      $output .= "\n<a href=\"articles.php?idarticle=" . $data['id'] . "\">";
45      $output .= "\n<b> " . $data['name'] . "</b>";
46      $output .= "\n</a>";
47      $output .= "<br/><br/>";
48      $output .= "\n<p> " . descriptionSize($data['description']) . "</p>";
49      $output .= "\n</div>";
50      $output .= "\n<div class=\"col-lg-4\">";
51      $output .= "\nCréateur de l'annonce : " . $creatorname;
52      $output .= "\n<br/><br/>";
53      $output .= "\nLe " . $data['creationdate'];
54      $output .= "\n<br/><br/>";
55      $output .= "\nPrix : " . $data['price'];
56      $output .= "\n</div>";
57      $output .= "\n</li>";

```

```

56     $output .= "\n</li>";
57
58     return $output;
59 }
60
61 /**
62  * Retourne un string d'un certain nombre de caractère avec "..." a la fin
63  * @param string $desc      Description à raccourcir
64  * @return string
65  */
66 function descriptionSize($desc) {
67     return substr($desc, 0, 50) . "...";
68 }
69
70 /**
71  * Retourne du code html pour afficher le commentaire
72  * @param Array $data        Données du commentaire
73  * @param int $creatorid     Identifiant du createur de l'article
74  * @return string            Code HTML
75  */
76 function commentFormat($data, $creatorid) {
77     $uid = intval($data['id_Users']);
78     $userinfo = getUserInfo($uid);
79
80     if ($userinfo['privilege'] == "2") {
81         $priv = "Admin";
82     } else {
83         $priv = "Membre";
84     }
85
86     if ($data['state'] == "1") {
87         $comment = $data['comm'];
88     } else {
89         $comment = "Commentaire en attente de modération";
90     }
91
92
93     $btn = "";
94
95     //Seulement le créateur ou les admin
96     if (getUserID() == $creatorid || getPrivilege() == PRIV_ADMIN) {
97         $btn .= "<input type=\"submit\" name=\"state\" class=\"btn btn-warning\"
98             value=\"!\" />";
99
100
101     //seulement les admin
102     if (getPrivilege() == PRIV_ADMIN) {
103         $btn .= "<input type=\"submit\" name=\"ban\" class=\"btn btn-danger\"
104             value=\"X\" />";
105
106     }
107
108     $output = "        \n<li class=\"media well\">
109         \n<div class=\"pull-left user-info col-lg-1\">
110             \n <img class=\"avatar img-circle img-thumbnail\"
111                 src=\"\" . $userinfo['path'] . "\""
112                 \nwidth=\"64\" alt=\"Generic placeholder image\">
113             \n<br/>

```

```

110         \n<strong><a href=\"user.html\">\" . formatUserName(
            $userinfo['firstname'], $userinfo['lastname']) .
            "</a></strong>
111     \n <br/><small>\" . $priv . "</small>
112     \n <br>
113
114         \n</div>
115     \n <div class=\"media-body\">\" .
116     $comment
117     . "\n </div>
118         \n<div id='postOptions' class=\"media-right\">
119         \" . $data['date_com'] . \"
120         \n<br/>\"
121     .
122     "\n<form action=\"#\" method=\"post\">
123         <input type=\"hidden\" name=\"comstate\"
            value=\"\" . $data['state'] . "\"/>
124         <input type=\"hidden\" name=\"idcom\" value=\"\" .
            $data['id'] . "\"/>
125     . $btn .
126     "</form>\"
127     .
128     //\n<a href=\"#\"><span class=\"input-group-addon\"><i class=\"glyphicon
        glyphicon-exclamation-sign\"></i></span></a>
129     //\n<br/>
130     //\n<a href=\"#\"><span class=\"input-group-addon\"><i class=\"glyphicon
        glyphicon-remove-sign\"></i></span></a>
131
132
133
134     "\n</div>
135         \n</li>\";
136
137     return $output;
138 }
139
140 /**
141  * Converti un tableau php en table html.
142  * Le paramètre facultatif $assoc permet d'afficher une ligne d'entete si le tableau
    est associatif.
143  * @param array $anArray Un tableau php (indexé ou associatif)
144  * @param boolean $assoc (true si le tableau est associatif)
145  * @return string Une chaîne qui contient un tableau html indenté.
146  */
147 function Array2Html($anArray, $assoc = false) {
148     $output = "\n<table class=\"table table-bordered\">\";
149     $lig = 0;
150
151     // Si le tableau est associatif, la première ligne contient le nom des colonnes
152     if ($assoc) {
153         $firstLine = true;
154         foreach ($anArray as $line) {
155             if ($firstLine) {
156                 $output .= "\n <tr>\";
157                 // affiche les entetes en th
158                 foreach ($line as $key => $value) {
159                     $output .= "\n <th>\" . $key . "</th>\";

```

```

160         }
161         $output .= "\n </tr>";
162         $output .= "\n <tr>";
163         // affiche les entetes en th
164         foreach ($line as $key => $value) {
165             $output .= "\n <td>" . $value . "</td>";
166         }
167         $output .= "\n </tr>";
168         $firstLine = false;
169     } else {
170         if (($lig++ % 2) == 1)
171             $output .= "\n <tr>";
172         else
173             $output .= "\n <tr class = \"odd\">";
174     // affiche les entetes en td
175         foreach ($line as $key => $value) {
176             $output .= "\n      <td>" . $value . "</td>";
177         }
178         $output .= "\n </tr>";
179     }
180 }
181 } else {
182 // Tableau indexé
183     for ($lig = 0; $lig < count($anArray); $lig++) {
184         if (($lig % 2) == 0)
185             $output .= "\n <tr>";
186         else
187             $output .= "\n <tr class=\"odd\">";
188     // Compte le nombre de colonnes
189         for ($col = 0; $col < count($anArray[$lig]); $col++) {
190             $output .= "\n      <td>" . $anArray[$lig][$col] . "</td>";
191         }
192         $output .= "\n </tr>";
193     }
194 }
195
196 // Compte le nombre de lignes
197 $output .= "\n</table>";
198 return $output;
199 }
200
201 /**
202  * Retourne du code html pour faire un select avec tous les pays
203  * @param string $selectedcountry      Nom du pays (a specifier pour definir
204  l'attribut selected)
205  * @return string      Code HTML
206  */
207 function selectCountry($selectedcountry = null) {
208     $output = "\n<select name=\"country\" class=\"form-control\" id=\"country\">";
209     foreach (getAllCountry() as $value) {
210         if ($selectedcountry == $value['name']) {
211             $selected = "selected";
212         } else {
213             $selected = "";
214         }
215     }

```



```
216
217
218     $output .= "\n<option value=\"\" . $value['iso'] . "\"" . $selected . ">" .
        $value['name'] . "</option>";
219 }
220
221 $output .= "\n</select>";
222
223 return $output;
224 }
225
226
```

```
1  <?php
2
3  /*
4   Fichier: controller.php
5   Auteur: Kevin Zaffino
6   Date: 15/06/2016
7   Version:1.10
8   Description: Gere les input du site
9   Copyright (TPI 2016 - Kevin Zaffino © 2016)
10  */
11
12  /**
13   * Recupere le mail et le mot de passe et effectue la verification
14   * @param string $mail
15   * @param string $pwd
16   * @return string
17   */
18  function checkLogin($mail, $pwd) {
19      $msg = TRUE;
20      if (login($mail, $pwd)) {
21          $userinfo = getUserInfo(login($mail, $pwd));
22          if ($userinfo['banned'] == '1') {
23              return "error";
24          }
25          $address = $userinfo['street'] . " " . $userinfo['city'] . ", " . $userinfo[
            'country'];
26
27          setPrivilege(intval($userinfo['privilege']));
28          setImagePath($userinfo['path']);
29          setUserID(intval($userinfo['id']));
30          setUsername(formatUserName($userinfo['firstname'], $userinfo['lastname']));
31          setUserMail($mail);
32          setUserTel($userinfo['phone']);
33          setUserAddress($address);
34          setLogged();
35
36          header('Location: index.php');
37      } else {
38          $msg = "Identifiants incorrect";
39      }
40      return $msg;
41  }
42
43  /**
44   * Recupere les informations d'enregistrement et effectue le traitement
45   * @param string $lastName
46   * @param string $firstName
47   * @param int $gender
48   * @param string $mail
49   * @param string $pwd
50   * @param string $phone
51   * @param string $country
52   * @param string $city
53   * @param string $street
54   * @param int $id_image
55   * @return boolean
56   */
```

```
57 function checkRegister($lastName, $firstName, $gender, $mail, $pwd, $phone, $country,
58     $city, $street, $id_image) {
59     if (!insertUser($lastName, $firstName, $gender, $mail, $pwd, $phone, $country,
60         $city, $street, $id_image)) {
61         return FALSE;
62     } else {
63         header('Location: login.php');
64     }
65 }
66 /**
67  * Récupere les information de création d'annonce et effectue le traitement
68  * @param string $name
69  * @param string $description
70  * @param string $price
71  * @param string $date
72  * @param bool $mvis
73  * @param bool $pvis
74  * @param bool $avis
75  * @return bool
76  */
77 function checkNewArticle($name, $description, $price, $date, $mvis, $pvis, $avis) {
78     $uid = getUserID();
79     return insertArticle($name, $description, $price, $date, $uid, $mvis, $pvis,
80         $avis);
81 }
82 /**
83  * Récupere les informations d'edition d'annonce et effectue le traitement
84  * @param int $id
85  * @param string $name
86  * @param string $description
87  * @param string $price
88  * @param bool $mvis
89  * @param bool $pvis
90  * @param bool $avis
91  */
92 function checkEditArticle($id, $name, $description, $price, $mvis, $pvis, $avis) {
93     editArticleInfo($id, $name, $description, $price, $mvis, $pvis, $avis);
94     header('Location: articles.php?idarticle=' . $id);
95 }
96 /**
97  * Récupere les informations d'edition du compte et effectue le traitement
98  * @param string $country
99  * @param string $city
100  * @param string $street
101  * @param int $adressid
102  * @param string $lastName
103  * @param string $firstName
104  * @param int $gender
105  * @param string $mail
106  * @param string $phone
107  */
108 function checkAccountEdit($country, $city, $street, $adressid, $lastName, $firstName,
109     $gender, $mail, $phone) {
```

```
110     updateUserAdress($adressid, $country, $city, $street);
111     updateUserInfo($lastName, $firstName, $gender, $mail, $phone, getUserID());
112     header('Location: userPage.php');
113 }
114
115
```

```
1  <?php
2  /*
3  Fichier: sessions.php
4  Auteur: Kevin Zaffino
5  Date: 15/06/2016
6  Version:1.10
7  Description: Gere les sessions
8  Copyright (TPI 2016 - Kevin Zaffino © 2016)
9  */
10
11 session_start();
12 ?>
13
14 <?php
15 require_once './phpScript/constants.php';
16
17 /**
18  * Est-ce que l'utilisateur est loggé
19  * @return {boolean} True is loggé, autrement false
20  */
21 function isLoggedIn() {
22     return (isset($_SESSION["logged"])) ? $_SESSION["logged"] : false;
23 }
24
25 /**
26  * Définis l'etat "logged"
27  */
28 function setLogged() {
29     $_SESSION['logged'] = true;
30 }
31
32 /**
33  * Recupere le niveau de privilège de l'utilisateur
34  * @return (int)
35  */
36 function getPrivilege() {
37     return (isset($_SESSION["privilege"])) ? $_SESSION["privilege"] : PRIV_UNKNOWN;
38 }
39
40 /**
41  * Assigne le niveau de privilege en session
42  * @param (int) $value
43  */
44 function setPrivilege($value) {
45     $_SESSION["privilege"] = $value;
46 }
47
48 /**
49  * Assigne le chemin de l'avatar de l'utilisateur en session
50  * @param (string) $path
51  */
52 function setImagePath($path) {
53     $_SESSION['image'] = $path;
54 }
55
56 /**
57  * Retourne le chemin de l'avatar de l'utilisateur contenu en session
```

```
58     * @return (string)
59     */
60 function getImagePath() {
61     return (isset($_SESSION["image"])) ? $_SESSION["image"] : NULL;
62 }
63
64 /**
65  * Est-ce que l'utilisateur est un admin
66  * @return (bool) TRUE si il est admin, sinon FALSE
67  */
68 function isAdmin() {
69     return getPrivilege() == PRIV_ADMIN;
70 }
71
72 /**
73  * Est-ce que l'utilisateur est un simple utilisateur
74  * @return (bool) TRUE si oui, sinon FALSE
75  */
76 function isUser() {
77     return getPrivilege() == PRIV_USER;
78 }
79
80 /**
81  * Recupere l'id utilisateur contenu en session
82  * @return (int)
83  */
84 function getUserID() {
85     return (isset($_SESSION["uid"])) ? $_SESSION["uid"] : -1;
86 }
87
88 /**
89  * Assigne l'id de l'utilisateur en session
90  * @param type $uid
91  */
92 function setUserID($uid) {
93     $_SESSION["uid"] = $uid;
94 }
95
96 /**
97  * Retourne le nom de l'utilisateur enregistré en session
98  * @return type
99  */
100 function getUsername() {
101     return (isset($_SESSION["uname"])) ? $_SESSION["uname"] : "";
102 }
103
104 /**
105  * Assigne le nom d'utilisateur a une variable de session
106  * @param type $name
107  */
108 function setUsername($name) {
109     $_SESSION["uname"] = $name;
110 }
111
112 /**
113  * Retourne le mail de l'utilisateur
114  * @return type
```

```
115     */
116 function getUserMail() {
117     return (isset($_SESSION["umail"])) ? $_SESSION["umail"] : "";
118 }
119
120 /**
121  * Assigne le mails de l'utilisateur dans une variable de session
122  * @param type $mail
123  */
124 function setUserMail($mail) {
125     $_SESSION['umail'] = $mail;
126 }
127
128 /**
129  * Recupere le numero de l'utilisateur
130  * @return type
131  */
132 function getUserTel() {
133     return (isset($_SESSION["utel"])) ? $_SESSION["utel"] : "";
134 }
135
136 /**
137  * Assigne le numero de l'utilisateur dans une variable de session
138  * @param type $tel
139  */
140 function setUserTel($tel) {
141     $_SESSION['utel'] = $tel;
142 }
143
144 /**
145  * Recupere l'adresse de l'utilisateur
146  * @return type
147  */
148 function getUserAdress() {
149     return (isset($_SESSION["uadress"])) ? $_SESSION["uadress"] : "";
150 }
151
152 /**
153  * Assigne l'adresse de l'utilisateur dans une variable de session
154  * @param type $mail
155  */
156 function setUserAdress($mail) {
157     $_SESSION['uadress'] = $mail;
158 }
159
160 /**
161  * Detruit la session
162  */
163 function destroySession(){
164     session_destroy();
165 }
166
```

```
1  <?php
2  /*
3  Fichier: constants.php
4  Auteur: Kevin Zaffino
5  Date: 15/06/2016
6  Version:1.10
7  Description: Contient les constantes du site
8  Copyright (TPI 2016 - Kevin Zaffino © 2016)
9  */
10
11 //constantes pour les privilèges
12 define('PRIV_ADMIN', 2);
13 define('PRIV_USER', 1);
14 define('PRIV_UNKNOWN', 0);
15
16 //image par défaut de l'utilisateur
17 define('DEFAULT_IMAGE_ID', 19);
18
19 //Upload d'image
20 define('INPUT', 'imgupload'); // Nom de l'input du fichier
21 define('TARGET', './img/'); // Repertoire cible
22 define('MAX_SIZE', 10000000); // Taille max en octets du fichier
23 define('WIDTH_MAX', 5000); // Largeur max de l'image en pixels
24 define('HEIGHT_MAX', 5000); // Hauteur max de l'image en pixels
25
```



```
1  <?php
2  /*
3  Fichier: inc.all.php
4  Auteur: Kevin Zaffino
5  Date: 15/06/2016
6  Version:1.10
7  Description: Gere les require utilisé sur les différentes page
8  Copyright (TPI 2016 - Kevin Zaffino © 2016)
9  */
10
11 // Les constantes
12 require_once './phpScript/constants.php';
13 // Les sessions
14 require_once './phpScript/sessions.php';
15 // Les fonctions
16 require_once './phpScript/function.php';
17 //Les fonctions d'affichage
18 require_once './phpScript/functiondisp.php';
19 //Le controleur
20 require_once './phpScript/controller.php';
21
22
```

```
1  <?php
2  /*
3  Fichier: mysql.inc.php
4  Auteur: Kevin Zaffino
5  Date: 15/06/2016
6  Version:1.10
7  Description: Contient les constantes de connexion a la base de données
8  Copyright (TPI 2016 - Kevin Zaffino © 2016)
9  */
10
11 //constantes de base de données
12 define('DB_HOST', 'localhost');
13 define('DB_NAME', 'jannonce_db');
14 define('DB_USER', 'jannonce_admin');
15 define('DB_PASSWORD', 'Superadmin');
16
17
18
19
20
```

```
1  <?php
2
3  /*
4   Fichier: showmenu.php
5   Auteur: Kevin Zaffino
6   Date: 15/06/2016
7   Version:1.10
8   Description: Gere l'affichage des menus
9   Copyright (TPI 2016 - Kevin Zaffino © 2016)
10  */
11
12  if (isset($_REQUEST['logout'])) {
13      logOut();
14  }
15
16  switch (getPrivilege()) {
17      case 0:
18          include './menu/defaultMenu.html';
19          break;
20      case 1:
21          include './menu/authMenu.html';
22          break;
23      case 2:
24          include './menu/adminMenu.html';
25          break;
26  }
27
```

```
1  <?php
2  /*
3   Fichier: index.php
4   Auteur: Kevin Zaffino
5   Date: 15/06/2016
6   Version:1.10
7   Description: Page d'accueil du site
8   Copyright (TPI 2016 - Kevin Zaffino © 2016)
9   */
10
11 require_once './phpScript/inc.all.php';
12 ?>
13 <!DOCTYPE html>
14 <html>
15     <head>
16         <meta charset="UTF-8">
17         <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
18
19         <title>Accueil</title>
20     </head>
21     <body>
22         <?php
23         include './menu/showmenu.php';
24         ?>
25         <div class="container">
26             <?php if (isLoggedIn()) { ?>
27                 <div>
28                     <a href="newArticle.php" class="btn btn-info" role="button">
29                         Nouvelle annonce</a>
30                     <br/><br/>
31                 </div>
32             <?php } ?>
33             <div id="articles">
34                 <ul class="media-list forum">
35                     <?php
36                     foreach (listArticles() as $value) {
37                         $path = articleImages(intval($value['id']))[0]['path'];
38                         print articleFormat($value, $path);
39                     }
40                     ?>
41                 </ul>
42             </div>
43         </body>
44 </html>
45
```

```
1  <?php
2  /*
3      Fichier: about.php
4      Auteur: Kevin Zaffino
5      Date: 15/06/2016
6      Version:1.10
7      Description: Page "a propos" du site
8      Copyright (TPI 2016 - Kevin Zaffino © 2016)
9  */
10
11  require_once './phpScript/inc.all.php';
12  ?>
13  <!DOCTYPE html>
14  <html>
15      <head>
16          <meta charset="UTF-8">
17          <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
18
19          <title>about</title>
20      </head>
21      <body>
22          <?php
23          include './menu/showmenu.php';
24          ?>
25
26          <div class="panel-info col-lg-3 pull-left">
27              <div class = "panel-heading">
28                  <h3 class = "panel-title">A propos du site</h3>
29              </div>
30              <div class="panel-body">
31                  <p>Le site permettra aux utilisateurs de poster diverses annonces
32                      (ventes, services, ...). Les utilisateurs pourront voir les annonces
33                      existantes et auront ainsi la possibilité de contacter la personne
34                      concernée.</p>
35              </div>
36          </div>
37
38          <div class="panel-info col-lg-9 pull-right">
39              <div class = "panel-heading">
40                  <h3 class = "panel-title">F.A.Q</h3>
41              </div>
42              <div class="panel-body">
43                  <h2>
44                      Comment se connecter sur le site ?
45                  </h2>
46                  <p>
47                      Il suffit de cliquer sur l'onglet « Connexion/Inscription » dans
48                      le menu du site.
49                  </p>
50                  <h2>
51                      Comment ajouter une nouvelle annonce ?
52                  </h2>
53                  <p>
54                      Une fois connecté sur le site, il suffit de cliquer sur «
55                      Nouvelle annonce » dans la page d'accueil.
```

```
53         </p>
54         <h2>
55             Je possède une annonce qui est terminée, comment le signaler ?
56         </h2>
57         <p>
58             Il suffit de cliquer sur « Ouvrir/Fermer » dans la page de votre
             annonce.
59         </p>
60     </div>
61 </div>
62
63 <div class="panel-info col-lg-3 pull-left">
64     <div class = "panel-heading">
65         <h3 class = "panel-title">Auteur</h3>
66     </div>
67     <div class="panel-body">
68         Nom      : ZAFFINO Kevin
69         <br/>
70         email    : kevin.zffn@eduge.ch
71     </div>
72 </div>
73 </body>
74 </html>
75
```

```
1  <?php
2  /*
3      Fichier: administration.php
4      Auteur: Kevin Zaffino
5      Date: 15/06/2016
6      Version: 1.10
7      Description: Page d'administration du site
8      Copyright (TPI 2016 - Kevin Zaffino © 2016)
9  */
10
11  require_once './phpScript/inc.all.php';
12
13  //Acces unique aux administrateurs
14  if (getPrivilege() != PRIV_ADMIN) {
15      header('Location: index.php');
16  }
17  ?>
18  <!DOCTYPE html>
19  <html>
20      <head>
21          <meta charset="UTF-8">
22          <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
23
24          <title>Administration</title>
25      </head>
26      <body>
27          <?php
28              include './menu/showmenu.php';
29              ?>
30
31          <?php
32              if (isset($_REQUEST['banuser'])) {
33                  $uid = intval($_REQUEST['uid']);
34                  banUnbanUser($uid);
35                  header('Location: administration.php');
36              }
37
38              if (isset($_REQUEST['banarticle'])) {
39                  $aid = intval($_REQUEST['aid']);
40                  banunbanArticle($aid);
41                  header('Location: administration.php');
42              }
43              ?>
44
45          <div id="articles" class = "panel panel-primary">
46              <div class = "panel-heading">
47                  <h3 class = "panel-title">Utilisateurs</h3>
48              </div>
49              <div class = "panel-body">
50                  <ul class="media-list forum">
51                      <?php
52                          $users = getAllUser();
53
54                          for ($i = 0; $i < count($users); $i++) {
55                              $edit = '<form action="#" method="post">
56                                  <input type="hidden" name="uid" value="' . $users[$i][
57                                      'id' . '"/>
```

```
57         <input type="submit" name="banuser" class="btn
           btn-primary" value="Ban/Unban"/>
58     </form>';
59
60     $users[$i]['Edition'] = $edit;
61 }
62
63     print Array2Html($users, true);
64     ?>
65
66     </ul>
67     </div>
68 </div>
69
70 <div id="articles" class = "panel panel-primary">
71     <div class = "panel-heading">
72         <h3 class = "panel-title">Annonces</h3>
73     </div>
74     <div class = "panel-body">
75         <ul class="media-list forum">
76             <?php
77                 $articles = getAllArticles();
78                 for ($i = 0; $i < count($articles); $i++) {
79                     $edit = '<form action="#" method="post">
80                         <input type="hidden" name="aid" value="' . $articles[
81                             $i]['id'] . '"/>
82                         <input type="submit" name="banarticle" class="btn
83                             btn-primary" value="Ban/Unban"/>
84                     </form>';
85                     $articles[$i]['Edition'] = $edit;
86                 }
87                 print Array2Html($articles, true);
88             </ul>
89         </div>
90     </div>
91 </body>
92 </html>
93
94
```



```
1  <!--
2  Fichier: adminMenu.html
3  Auteur: Kevin Zaffino
4  Date: 15/06/2016
5  Version:1.10
6  Description: Affiche le menu administrateur
7  Copyright (TPI 2016 - Kevin Zaffino © 2016)
8  -->
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>J'annonce</b></a></li>
14         </ul>
15         <ul class="nav navbar-nav navbar-right">
16             <li><a href="./administration.php" >Administration</a></li>
17             <li><a href="./userPage.php" >Mon compte</a></li>
18             <li><a href="./about.php" >A propos</a></li>
19             <li><a href="index.php?logout=true" >Deconnexion</a></li>
20         </ul>
21     </div>
22 </nav>
```

```
1  <?php
2  /*
3      Fichier: articleImages.php
4      Auteur: Kevin Zaffino
5      Date: 15/06/2016
6      Version:1.10
7      Description: Page d'affichage des images d'annonce
8      Copyright (TPI 2016 - Kevin Zaffino © 2016)
9  */
10 require_once './phpScript/inc.all.php';
11
12 //Uniquement si le parametre "id" existe
13 if (!isset($_REQUEST['id'])) {
14     header('Location: index.php');
15 }
16 ?>
17 <!DOCTYPE html>
18 <html>
19     <head>
20         <meta charset="UTF-8">
21         <title>articleImages</title>
22         <base target="_parent" />
23     </head>
24     <body>
25         <?php
26             $id = intval($_REQUEST['id']);
27             $img = articleImages($id);
28
29             //Affichage des images
30             foreach ($img as $value) {
31                 echo '<a href="' . $value['path'] . '"></a>';
33             }
34         ?>
35     </body>
36 </html>
```

```
1  <?php
2  /*
3      Fichier: articles.php
4      Auteur: Kevin Zaffino
5      Date: 15/06/2016
6      Version:1.10
7      Description: Page d'annonce du site
8      Copyright (TPI 2016 - Kevin Zaffino © 2016)
9  */
10
11  require_once './phpScript/inc.all.php';
12
13  if (isset($_REQUEST['idarticle'])) {
14      $said = intval($_REQUEST['idarticle']);
15  } else {
16      header('Location: index.php');
17  }
18  $articleinfo = articleInfo($said);
19  $banned = intval($articleinfo['banned']);
20
21  if ($banned) {
22      header('Location: index.php');
23  }
24  ?>
25  <!DOCTYPE html>
26  <html>
27      <head>
28          <meta charset="UTF-8">
29          <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
30          <title>Annonce</title>
31      </head>
32      <body>
33
34          <style>
35              .media-right {
36                  width: 400px ;
37              }
38          </style>
39
40          <?php
41          include './menu/showmenu.php';
42
43          $said = intval($_REQUEST['idarticle']);
44          $articlecreator = intval(articleInfo($said)['id_Users']);
45          $articleinfo = articleInfo($said);
46          $userinfo = getUserInfo(intval($articleinfo['id_Users']));
47
48          $state = intval($articleinfo['state']);
49
50
51          $vmail = intval($articleinfo['mailvisible']);
52          $vphone = intval($articleinfo['phonevisible']);
53          $vadress = intval($articleinfo['adressvisible']);
54
55          if (isset($_REQUEST['changestate'])) {
56              openCloseArticle($said);
57              header('Location: #');
```

```

58     }
59
60
61     if (isset($_REQUEST['post'])) {
62         $com = filter_input(INPUT_POST, 'txt');
63         $date = date("Y-m-d H:i:s");
64         addComment(getUserID(), $aid, $date, $com);
65         header("refresh:0");
66     }
67
68     if (isset($_REQUEST['state'])) {
69         $idcom = intval($_REQUEST['idcom']);
70         $comstate = intval($_REQUEST['comstate']);
71
72
73         if ($comstate == 1) {
74             $reversestate = 0;
75         } else {
76             $reversestate = 1;
77         }
78
79         changeComState($idcom, $reversestate);
80         header("refresh:0");
81     }
82
83     if (isset($_REQUEST['ban'])) {
84         $idcom = intval($_REQUEST['idcom']);
85         banComment($idcom);
86         header("refresh:0");
87     }
88     ?>
89     <div class="panel" id='article'>
90         <div class="panel-body">
91             <div class="pull-left col-lg-4">
92                 <iframe src="articleImages.php?id=<?php echo $aid; ?>" width=
93                     "100%" height="400">
94                 </iframe>
95             </div>
96             <div class="col-lg-4">
97                 <label for="description"><?php echo $articleinfo['name']; ?>
98                 </label>
99
100                 <div id='description'>
101                     <?php echo $articleinfo['description']; ?>
102                 </div>
103
104                 <br/><br/>
105                 <label for="description">Prix : <?php echo $articleinfo['price'];
106                     ?></label>
107             </div>
108             <div id="info" class="pull-right col-lg-4">
109                 Createur del'annonce : <?php echo getUserInfo(intval($articleinfo
110                     ['id_Users']))['firstname']; ?>
111                 <br/>
112                 Le : <?php echo $articleinfo['creationdate']; ?>
113                 <br/><br/>
114                 <?php

```

```

111         if ($vphone) {
112             echo 'Tel. : ' . $userinfo['phone'] . '<br/>';
113         }
114
115         if ($vmail) {
116             echo 'E-mail : ' . $userinfo['mail'] . '<br/>';
117         }
118
119         if ($vaddress) {
120             echo 'Adresse : ' . $userinfo['country'] . '<br/>';
121         }
122     }
123
124     <br/><br/><br/>
125
126     Etat de l'annonce : <?php
127     if ($state) {
128         echo 'Ouvert';
129     } else {
130         echo 'Fermé';
131     }
132     }
133
134     <?php if (getUserID() == $articlecreator || getPrivilege() ==
PRIV_ADMIN) { ?>
135         <form action="#" method="post">
136             <input type="hidden" name="id" value="<?php echo $aid; ?>
"/>
137             <button type="submit" class="btn btn-success" name=
"changestate">Ouvrir/Fermer</button>
138             <a href="./editArticle.php?aid=<?php echo $aid; ?>" class
="btn btn-warning" role="button">Editer l'annonce</a>
139         </form>
140         <?php } ?>
141     </div>
142
143 </div>
144 </div>
145
146 <div id="articles" class = "panel panel-primary">
147     <div class = "panel-heading">
148         <h3 class = "panel-title">Commentaire(s)</h3>
149     </div>
150     <div class = "panel-body">
151         <div id='comments' class="col-lg-12">
152             <ul class="media-list forum">
153                 <?php
154                 foreach (getArticleComments($aid) as $value) {
155                     print commentFormat($value, $articlecreator);
156                 }
157             }
158
159             <?php if (getPrivilege() != PRIV_UNKNOWN && $state == 1) { ?>
160                 <li class="media well">
161                     <div class="pull-left user-info col-lg-1" href="#">
162                         <img class="avatar img-circle img-thumbnail" src=
"<?php echo getImagePath(); ?>"

```

```
163         width="64" alt="Generic placeholder image">
164     <br/>
165     <strong><a href="user.html"><?php echo $_SESSION[
166         'uname' ] . "<br/>"; ?></a></strong>
167     <br>
168
169 </div>
170 <div class="media-right">
171     <form role="form" action="#" method="post">
172
173         <textarea name='txt' rows="5" cols="100"
174             maxlength="500" required></textarea>
175         <br/>
176         <button type="submit" class="btn btn-success"
177             name="post">Envoyer</button>
178     </form>
179 </div>
180 </li>
181 <?php } ?>
182 </ul>
183 </div>
184 </div>
185 </body>
186 </html>
```

```
1  <!--
2  Fichier: authMenu.html
3  Auteur: Kevin Zaffino
4  Date: 15/06/2016
5  Version:1.10
6  Description: Affiche le menu d'utilisateur authentifié
7  Copyright (TPI 2016 - Kevin Zaffino © 2016)
8  -->
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>J'annonce</b></a></li>
14         </ul>
15         <ul class="nav navbar-nav navbar-right">
16             <li><a href="./userPage.php" >Mon compte</a></li>
17             <li><a href="./about.php" >A propos</a></li>
18             <li><a href="index.php?logout=true" >Deconnexion</a></li>
19         </ul>
20     </div>
21 </nav>
```

```
1  <!--
2  Fichier: defaultMenu.html
3  Auteur: Kevin Zaffino
4  Date: 15/06/2016
5  Version:1.10
6  Description: Affiche le menu par défaut
7  Copyright (TPI 2016 - Kevin Zaffino © 2016)
8  -->
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>J'annonce</b></a></li>
14         </ul>
15         <ul class="nav navbar-nav navbar-right">
16             <li><a href="./login.php" >Connexion/Inscription</a></li>
17             <li><a href="./about.php" >A propos</a></li>
18         </ul>
19     </div>
20 </nav>
```



```
1  <?php
2  /*
3      Fichier: editArticle.php
4      Auteur: Kevin Zaffino
5      Date: 15/06/2016
6      Version:1.10
7      Description: Page d'edition d'annonce
8      Copyright (TPI 2016 - Kevin Zaffino © 2016)
9  */
10
11  require './phpScript/inc.all.php';
12
13  $aid = intval($_REQUEST['aid']);
14  $uidarticle = intval(articleInfo($aid)['id_Users']);
15
16  //Acces unique au createur de l'annonce et à l'admin
17  if (getUserID() != $uidarticle && getPrivilege() != PRIV_ADMIN) {
18      header('Location: index.php');
19  }
20  ?>
21  <!DOCTYPE html>
22  <html>
23      <head>
24          <meta charset="UTF-8">
25          <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
26
27          <title>Edition de l'annonce</title>
28      </head>
29      <body>
30          <?php
31              include './menu/showmenu.php';
32
33
34              $article = articleInfo($aid);
35
36              if (isset($_REQUEST['submit'])) {
37                  $name = filter_var($_REQUEST['name'], FILTER_SANITIZE_SPECIAL_CHARS);
38                  $description = filter_var($_REQUEST['description'],
39                      FILTER_SANITIZE_SPECIAL_CHARS);
40                  $price = filter_var($_REQUEST['price'], FILTER_SANITIZE_SPECIAL_CHARS);
41
42                  $date = date('Y-m-d H:i:s');
43                  // $uid = getUserID();
44
45                  $mvis = (isset($_POST["mailVisible"])) ? TRUE : FALSE;
46                  $pvis = (isset($_POST["phoneVisible"])) ? TRUE : FALSE;
47                  $avis = (isset($_POST["adressVisible"])) ? TRUE : FALSE;
48
49                  if ($_FILES[INPUT]['name'][0] != '') {
50                      deleteArticleImages($aid);
51                      multiUpload($aid);
52                  }
53
54                  checkEditArticle($aid, $name, $description, $price, $mvis, $pvis, $avis);
55              }
56          ?>
```

```

57     <div class="col-md-5 col-md-offset-4">
58         <div class="panel panel-default">
59             <div class = "panel-heading">
60                 <h3 class = "panel-title">Edite l'annonce</h3>
61             </div>
62             <div class="panel-body">
63                 <form action="#" method="post" enctype="multipart/form-data">
64                     <label for="title">Libelle :
65                         <input type="text" class="form-control" name='name' value
66                             = "<?php echo $article['name']; ?>" required/>
67                     </label>
68                     <br/>
69                     <label for="description">Description :<br/>
70                         <textarea name='description' rows="10" cols="50"
71                             maxlength="500" required><?php echo $article[
72                                 'description']; ?></textarea>
73                     </label>
74                     <br/>
75                     <label for="price">Prix :
76                         <input type="text" name='price' value="<?php echo
77                             $article['price']; ?>" required/>
78                     </label>
79                     <br/>
80                     <label for="image">Image(s) (Les anciennes images seront
81                         supprimées) :
82                         <input type="file" name="<?php echo INPUT; ?>[]" multiple
83                             accept="image/*"/>
84                     </label>
85                     <br/>
86                     <label for='mailVisible'>
87                         E-mail visible : <input type="checkbox" name=
88                             'mailVisible' <?php
89                             if ($article['mailvisible'] == 1) {
90                                 echo 'checked' ;
91                             }
92                             ?>/>
93                     </label>
94
95                     <label for='phoneVisible'>
96                         Numero de Tel. visible : <input type="checkbox" name=
97                             'phoneVisible' <?php
98                             if ($article['phonevisible'] == 1) {
99                                 echo 'checked' ;
100                             }
101                             ?>/>
102                     </label>
103
104                     <label for='adressVisible'>
105                         Adresse visible : <input type="checkbox" name=
106                             'adressVisible' <?php
107                             if ($article['adressvisible'] == 1) {
108                                 echo 'checked' ;
109                             }
110                             ?>/>
111                     </label>
112                     <br/>

```

```
105         <button type="submit" class="btn btn-success" name="submit">
            Envoyer</button>
106     </form>
107 </div>
108 </div>
109 </div>
110 </body>
111 </html>
112
```

```
1  <?php
2  /*
3   Fichier: login.php
4   Auteur: Kevin Zaffino
5   Date: 15/06/2016
6   Version:1.10
7   Description: Page de login du site
8   Copyright (TPI 2016 - Kevin Zaffino © 2016)
9   */
10
11 require_once './phpScript/inc.all.php';
12
13 //Acces unique aux utilisateurs authentifiés
14 if (isLoggedIn()) {
15     header('Location: index.php');
16 }
17 ?>
18 <!DOCTYPE html>
19 <html>
20     <head>
21         <meta charset="UTF-8">
22         <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
23         <title>login</title>
24     </head>
25     <body>
26         <?php
27         include './menu/showmenu.php';
28
29         $error = "";
30
31         if (isset($_REQUEST['login'])) {
32
33             $mail = filter_var($_REQUEST['mail'], FILTER_SANITIZE_EMAIL);
34             $pwd = filter_var($_REQUEST['pwd'], FILTER_SANITIZE_SPECIAL_CHARS);
35
36             $error = checkLogin($mail, $pwd);
37         }
38         ?>
39         <div class="container" style="margin-top:30px">
40             <div class="col-md-4 col-md-offset-4">
41                 <div class="panel panel-default">
42                     <div class="panel-body">
43                         <form method="post" action="#">
44                             <div class="form-group">
45                                 <label for="mail" class="sr-only">Email address
46                                 </label>
47                                 <div class="input-group">
48                                     <span class="input-group-addon"><i class=
49                                     "glyphicon glyphicon-user"></i></span>
50                                     <input type="email" class="form-control" id="mail"
51                                     placeholder="Entrez votre adresse email"
52                                     name="mail" required>
53                                 </div>
54                             </div>
55                             <div class="form-group">
56                                 <label for="pwd" class="sr-only">Password</label>
57                                 <div class="input-group">
```

```
55         <span class="input-group-addon"><i class=
56         "glyphicon glyphicon-lock"></i></span>
57         <input type="password" class="form-control" id=
58         "pwd"
59         placeholder="Mot de passe" name="pwd"
60         required>
61     </div>
62 </div>
63 <button type="submit" class="btn btn-success" name=
64 "login">Connexion</button>
65 <a href="register.php" class="btn btn-primary" role=
66 "button">Pas encore inscrit ?</a>
67 </form>
68 </div>
69 </div>
70 <?php if ($error) { ?>
71     <div class="alert alert-warning">
72         <strong>Attention!</strong> Identifiants incorrects.
73     </div>
74 <?php } ?>
75 </div>
76 </div>
77 </body>
78 </html>
```

```
1  <?php
2  /*
3      Fichier: modification.php
4      Auteur: Kevin Zaffino
5      Date: 15/06/2016
6      Version:1.10
7      Description: Page d'edition des informations utilisateur
8      Copyright (TPI 2016 - Kevin Zaffino © 2016)
9  */
10
11  require_once './phpScript/inc.all.php';
12
13  //Acces unique aux utilisateurs authentifiés
14  if (!isLoggedIn()) {
15      header('Location: index.php');
16  }
17  ?>
18
19  <!DOCTYPE html>
20  <html>
21      <head>
22          <meta charset="UTF-8">
23          <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
24          <link rel="stylesheet" type="text/css" href=
25              "CSS/font-awesome/css/font-awesome.css">
26          <title>Modification</title>
27      </head>
28      <body>
29          <?php
30              include './menu/showmenu.php';
31
32              $data = getUserInfo(getUserID());
33
34              // Si on a appuyé sur le bouton Valider
35              if (isset($_REQUEST['change'])) {
36
37                  $lastName = filter_var($_REQUEST['lastname'],
38                      FILTER_SANITIZE_SPECIAL_CHARS);
39                  $firstName = filter_var($_REQUEST['firstname'],
40                      FILTER_SANITIZE_SPECIAL_CHARS);
41                  $mail = filter_var($_REQUEST['mail'], FILTER_SANITIZE_SPECIAL_CHARS);
42                  // $pwd = filter_var($_REQUEST['pwd'], FILTER_SANITIZE_SPECIAL_CHARS);
43                  $phone = filter_var($_REQUEST['phone'], FILTER_SANITIZE_SPECIAL_CHARS);
44                  $country = filter_var($_REQUEST['country'], FILTER_SANITIZE_SPECIAL_CHARS);
45
46                  $city = filter_var($_REQUEST['city'], FILTER_SANITIZE_SPECIAL_CHARS);
47                  $street = filter_var($_REQUEST['street'], FILTER_SANITIZE_SPECIAL_CHARS);
48                  $gender = filter_var($_REQUEST['gender'], FILTER_SANITIZE_SPECIAL_CHARS);
49
50                  if ($_FILES[INPUT]['name'] != '') {
51                      deleteFile(getImagePath());
52                      $id_image = intval(imageUpload());
53                      changeUserInfo(getUserID(), $id_image);
54                      $userinfo = getUserInfo(getUserID());
55                      setImagePath($userinfo['path']);
56                  }
57              }
```

```

54
55         checkAccountEdit($country, $city, $street, $data['id_Adress'], $lastName,
56             $firstName, $gender, $mail, $phone);
57     }
58     <div class="container" style="margin-top:30px">
59         <div class="col-md-5 col-md-offset-3">
60             <div class="panel panel-default">
61                 <div class="panel-heading">
62                     <h3 class="panel-title">Modification</h3>
63                 </div>
64                 <div class="panel-body">
65                     <form role="form" method="post" action="#" enctype=
66                         "multipart/form-data">
67                         <label for="lastname">Nom :
68                             <input type="text" class="form-control" name=
69                                 "lastname" id='lastname' value="<?php echo $data[
70                                     'lastname']; ?>" required/>
71                         </label>
72                         <label for="firstname">Prenom :
73                             <input type="text" class="form-control" name=
74                                 "firstname" id='firstname' value="<?php echo $data[
75                                     'firstname']; ?>" required/>
76                         </label>
77                         <label for="mail">E-mail :
78                             <input type="mail" class="form-control" name="mail"
79                                 id='mail' value="<?php echo $data['mail']; ?>"
80                                 required />
81                         </label>
82                         <label for="phone">Tel. :
83                             <input type="text" class="form-control" name="phone"
84                                 id='phone' value="<?php echo $data['phone']; ?>">
85                         </label>
86                         <label for="adress">Adresse :
87                             <div id='adress'>
88                                 <?php
89                                 echo selectCountry($data['country']);
90                                 ?>
91                             <label for="city">Ville :
92                                 <input type="text" class="form-control" name=
93                                     "city" id='city' value="<?php echo $data[
94                                         'city']; ?>">
95                                 </label>
96                                 <label for="street">Rue :
97                                     <input type="text" class="form-control" name=
98                                         "street" id='street' value="<?php echo $data[
99                                             'street']; ?>">
100                                </label>
101                            </div>
102                        </label>
103                        <label for="gender">Genre :
104                            <select name="gender" class="form-control" name=
105                                "gender" id="gender">

```

```
97         <option value="1" <?php
98         if ($data['gender'] == "1") {
99             echo 'selected';
100         }
101         ?>>Homme</option>
102         <option value="0" <?php
103         if ($data['gender'] == "0") {
104             echo 'selected';
105         }
106         ?>>Femme</option>
107     </select>
108 </label>
109 <br/>
110
111
112     <label for="image">Image de profil :
113         <input type="file" class="form-control" name="<?php
114         echo INPUT; ?>" id='image' accept="image/*"/>
115     </label>
116     <br/>
117     <button type="submit" class="btn btn-success" name=
118     "change">Valider</button>
119 </form>
120 </div>
121 </div>
122 </div>
123 </body>
124 </html>
125
```



```
1  <?php
2  /*
3      Fichier: newArticle.php
4      Auteur: Kevin Zaffino
5      Date: 15/06/2016
6      Version:1.10
7      Description: Page de creation d'annonce
8      Copyright (TPI 2016 - Kevin Zaffino © 2016)
9  */
10 require './phpScript/inc.all.php';
11
12 //Acces unique aux utilisateurs authentifiés
13 if (!isLoggedIn()) {
14     header('Location: index.php');
15 }
16 ?>
17 <!DOCTYPE html>
18 <html>
19     <head>
20         <meta charset="UTF-8">
21         <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
22
23         <title>Nouvelle annonce</title>
24     </head>
25     <style>
26         .panel{
27             width: 75%;
28
29         }
30     </style>
31     <body>
32         <?php
33         include './menu/showmenu.php';
34
35         if (isset($_REQUEST['submit'])) {
36
37             $name = filter_var($_REQUEST['name'], FILTER_SANITIZE_SPECIAL_CHARS);
38             $description = filter_var($_REQUEST['description'],
39                                     FILTER_SANITIZE_SPECIAL_CHARS);
40             $price = filter_var($_REQUEST['price'], FILTER_SANITIZE_SPECIAL_CHARS);
41
42             $date = date('Y-m-d H:i:s');
43
44             $mvis = (isset($_POST["mailVisible"])) ? TRUE : FALSE;
45             $pvis = (isset($_POST["phoneVisible"])) ? TRUE : FALSE;
46             $avis = (isset($_POST["adressVisible"])) ? TRUE : FALSE;
47
48             $article_id = checkNewArticle($name, $description, $price, $date, $mvis,
49                                     $pvis, $avis);
50
51             multiUpload($article_id);
52
53             header('Location: articles.php?idarticle=' . $article_id);
54         }
55         ?>
56         <div class="col-md-5 col-md-offset-4">
57             <div class="panel panel-default">
```

```

56         <div class = "panel-heading">
57             <h3 class = "panel-title">Nouvelle annonce</h3>
58         </div>
59         <div class="panel-body">
60             <form action="#" method="post" enctype="multipart/form-data">
61                 <label for="title">Libelle :
62                     <input type="text" class="form-control" name='name'
63                         required/>
64                 </label>
65                 <br/>
66                 <label for="description">Description :<br/>
67                     <textarea name='description' rows="10" cols="50"
68                         maxlength="500" required></textarea>
69                 </label>
70                 <br/>
71                 <label for="price">Prix :
72                     <input type="text" name='price' required/>
73                 </label>
74                 <br/>
75                 <label for="image">Image(s) :
76                     <input type="file" name="<?php echo INPUT; ?>[]" multiple
77                         accept="image/*" required/>
78                 </label>
79                 <br/>
80                 <label for='mailVisible'>
81                     E-mail visible : <input type="checkbox" name=
82                         'mailVisible'/>
83                 </label>
84                 <br/>
85                 <label for='phoneVisible'>
86                     Numero de Tel. visible : <input type="checkbox" name=
87                         'phoneVisible'/>
88                 </label>
89                 <br/>
90                 <label for='adressVisible'>
91                     Adresse visible : <input type="checkbox" name=
92                         'adressVisible'/>
93                 </label>
94                 <br/>
95                 <button type="submit" class="btn btn-success" name="submit">
96                     Envoyer</button>
97             </form>
98         </div>
99     </div>
100 </body>
101 </html>

```

```
1  <?php
2  /*
3      Fichier: register.php
4      Auteur: Kevin Zaffino
5      Date: 15/06/2016
6      Version:1.10
7      Description: Page d'enregistrement
8      Copyright (TPI 2016 - Kevin Zaffino © 2016)
9  */
10
11  require_once './phpScript/inc.all.php';
12
13  //Acces unique aux utilisateurs authentifiés
14  if (isLoggedIn()) {
15      header('Location: index.php');
16  }
17  ?>
18
19  <!DOCTYPE html>
20  <html>
21      <head>
22          <meta charset="UTF-8">
23          <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
24          <link rel="stylesheet" type="text/css" href=
25              "CSS/font-awesome/css/font-awesome.css">
26          <title>register</title>
27      </head>
28      <body>
29          <?php
30              include './menu/showmenu.php';
31
32              $mailverif = TRUE;
33              $error = "";
34
35              // Si on a appuyé sur le bouton Valider
36              if (isset($_REQUEST['register'])) {
37                  $lastName = filter_var($_REQUEST['lastname'],
38                      FILTER_SANITIZE_SPECIAL_CHARS);
39                  $firstName = filter_var($_REQUEST['firstname'],
40                      FILTER_SANITIZE_SPECIAL_CHARS);
41                  $mail = filter_var($_REQUEST['mail'], FILTER_SANITIZE_EMAIL);
42                  $pwd = filter_var($_REQUEST['pwd'], FILTER_SANITIZE_SPECIAL_CHARS);
43                  $phone = filter_var($_REQUEST['phone'], FILTER_SANITIZE_SPECIAL_CHARS);
44                  $country = filter_var($_REQUEST['country'], FILTER_SANITIZE_SPECIAL_CHARS);
45
46                  $city = filter_var($_REQUEST['city'], FILTER_SANITIZE_SPECIAL_CHARS);
47                  $street = filter_var($_REQUEST['street'], FILTER_SANITIZE_SPECIAL_CHARS);
48                  $gender = filter_var($_REQUEST['gender'], FILTER_SANITIZE_SPECIAL_CHARS);
49
50                  // $id_image = "qq";
51                  if ($_FILES[INPUT]['name'] != '') {
52                      $id_image = intval(imageUpload());
53                  } else {
54                      $id_image = DEFAULT_IMAGE_ID;
55                  }
56              }
57          ?>
58      </body>
59  </html>
```

```

54         $mailverif = checkRegister($lastName, $firstName, $gender, $mail, $pwd,
55         $phone, $country, $city, $street, $id_image);
56     }
57     <div class="container" style="margin-top:30px">
58         <div class="col-md-5 col-md-offset-3">
59             <div class="panel panel-default">
60                 <div class = "panel-heading">
61                     <h3 class = "panel-title">Inscription</h3>
62                 </div>
63                 <div class="panel-body">
64                     <form method="post" action="#" enctype="multipart/form-data">
65                         <label for="lastname">Nom* :
66                             <input type="text" class="form-control" name=
67                             "lastname" id='lastname' required>
68                         </label>
69                         <label for="firstname">Prenom* :
70                             <input type="text" class="form-control" name=
71                             "firstname" id='firstname' required>
72                         </label>
73                         <label for="mail">E-mail* :
74                             <input type="email" class="form-control" name="mail"
75                             id='mail' required>
76                         </label>
77                         <?php if (!$mailverif) { ?>
78                             <div class="alert alert-warning">
79                                 <strong>Attention!</strong> Adresse email déjà
80                                 utilisée.
81                             </div>
82                         <?php } ?>
83                         <label for="pwd">mdp* :
84                             <input type="password" class="form-control" name=
85                             "pwd" id='pwd' pattern=".{3,}" required title="3
86                             characters minimum">
87                         </label>
88                         <label for="phone">Tel. :
89                             <input type="text" class="form-control" name="phone"
90                             id='phone'>
91                         </label>
92                         <label for="adress">Adresse :
93                             <div id='adress'>
94                                 <?php
95                                 echo selectCountry();
96                                 ?>
97                             <label for="city">Ville :
98                                 <input type="text" class="form-control" name=
99                                 "city" id='city'>
100                             </label>
101                             <label for="street">Rue :
102                                 <input type="text" class="form-control" name=
103                                 "street" id='street'>
104                             </label>
105                         </div>

```

```
101                                     </label>
102
103                                     <label for="gender">Genre :
104                                     <select name="gender" class="form-control" id=
105                                     "gender">
106                                     <option value="1">Homme</option>
107                                     <option value="0">Femme</option>
108                                     </select>
109                                     </label>
110                                     <br/>
111
112                                     <label for="image">Image de profil :
113                                     <input type="file" class="form-control" name="<?php
114                                     echo INPUT; ?>" id='image' accept="image/*"/>
115                                     </label>
116                                     <br/>
117                                     <button type="submit" class="btn btn-success" name=
118                                     "register">Valider</button>
119
120                                     </form>
121                                     </div>
122                                     </div>
123                                     </div>
124                                     </body>
</html>
```

```
1  <?php
2  /*
3      Fichier: userPage.php
4      Auteur: Kevin Zaffino
5      Date: 15/06/2016
6      Version:1.10
7      Description: Page d'information personnel
8      Copyright (TPI 2016 - Kevin Zaffino © 2016)
9  */
10 require_once './phpScript/inc.all.php';
11
12 //Acces unique aux utilisateurs
13 if (!isLoggedIn()) {
14     header('Location: index.php');
15 }
16 ?>
17 <!DOCTYPE html>
18 <html>
19     <head>
20         <meta charset="UTF-8">
21         <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
22
23         <title>User page</title>
24     </head>
25     <body>
26         <?php
27         include './menu/showmenu.php';
28         $userinfo = getUserInfo($_SESSION['uid']);
29         ?>
30
31         <div class="panel-info" id='information'>
32             <div class = "panel-heading">
33                 <h3 class = "panel-title">Information personnelle</h3>
34             </div>
35             <div class="panel-body">
36                 <div class="pull-left col-lg-2">
37                     <a href="#" class="thumbnail">
38                         
40                     </a>
41                 </div>
42                 <div class="col-lg-4">
43                     <div id='description'>
44                         <?php
45                         $gender = ($userinfo['gender'] == "0") ? "Femme" : "Homme";
46
47                         echo 'Nom: ' . $userinfo['lastname'];
48                         echo '<br/>';
49                         echo 'Prenom: ' . $userinfo['firstname'];
50                         echo '<br/>';
51                         echo 'Genre : ' . $gender;
52                         echo '<br/><br/><br/>';
53                         ?>
54
55                     <br/><br/><br/>
56                     <a href="modification.php" class="btn btn-warning" role=
57                         "button">Modifier les informations</a>
```

```
56         </div>
57     </div>
58
59     <div id="info" class="pull-right col-lg-4">
60         <div class="panel-default">
61             <div class="panel-heading">
62                 <h3 class="panel-title">Contact</h3>
63             </div>
64             <div class="panel-body">
65
66                 Tel. : <?php echo $userinfo['phone']; ?>
67                 <br/>
68                 E-mail : <?php echo $userinfo['mail']; ?>
69                 <br/>
70                 Adresse : <?php echo '<br/>' . $userinfo['street'] . " "
71                           . $userinfo['city'] . ", " . $userinfo['country']; ?>
72                 <br/><br/><br/>
73             </div>
74         </div>
75     </div>
76 </div>
77
78 <div id="articles" class="panel panel-primary">
79     <div class="panel-heading">
80         <h3 class="panel-title">Mes annonce(s)</h3>
81     </div>
82     <div class="panel-body">
83         <ul class="media-list forum">
84             <?php
85                 foreach (getUserArticles(getUserID()) as $value) {
86                     $path = articleImages(intval($value['id']))[0]['path'];
87                     print articleFormat($value, $path);
88                 }
89             ?>
90         </ul>
91     </div>
92 </div>
93
94 </body>
95 </html>
96
```