

```
1 <?php
2 header('Location: server/controllers/home.php');
3
```

```
1  /*
2  Titre : Contrôleur de l'administrateur
3  Date : Jeudi, 16 mai 2019
4  Auteur : Romain Peretti
5  Version : 1.0
6  Description : Cette page inclut la vue.
7  */
8
9  body {
10     background-color: #373738;
11     color: lightgray;
12 }
13
14 .my-btn {
15     margin: 2px;
16     color: lightgray;
17 }
18
19 .my-ad {
20     margin: 5px;
21 }
22
23 .my-ad:hover {
24     box-shadow: 2px 2px 18px 2px rgb(128, 240, 99);
25     transition: box-shadow 0.3s ease-in-out;
26 }
27
28 .my-ad:not(:hover) {
29     transition: box-shadow 1s ease-out;
30 }
31
32 .my-profil {
33     margin: 5px;
34 }
35
36 .my-navbar {
37     background-color: #4A4A4F;
38 }
39
40 hr {
41     background-color: white;
42 }
43
44 .my-tr:hover {
45     color: white;
46 }
47
48 .my-input {
49     background-color: rgb(230, 230, 230);
50 }
```

```
1 <!--
2 Titre : Élement "head"
3 Date : Mercredi 15 mai 2019
4 Auteur : Romain Peretti
5 Version : 1.0
6 -->
7 <title><?= $pageName ?></title>
8 <meta charset='utf-8'>
9 <meta name='viewport' content='width=device-width, initial-scale=1'>
10 <!-- Bootstrap CSS & JS -->
11 <link href='../css/bootstrap/css/bootstrap.min.css' rel='stylesheet' type='text
    /css' />
12 <script src='https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js'
    ></script>
13 <script src='../css/bootstrap/js/bootstrap.min.js' type='text/javascript'></
    script>
14 <!-- Font Awesome CDN -->
15 <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.8.2/css/
    all.css" integrity="sha384-oS3vJWv+0UjzBfQzYUhtDYW+Pj2yciDJxpsK1OYPAYjqT085Qq/
    1cq5FLXAZQ7Ay" crossorigin="anonymous">
16 <!-- SweetAlert CDN -->
17 <script src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
18 <!-- Personnel CSS -->
19 <link href='../css/personnal.css' rel='stylesheet' type='text/css'>
```

```
1 <!--
2 Titre : Pied de page
3 Date : 9 mai 2019
4 Auteur : Romain Peretti
5 Version : 1.0
6 Description : Cette page permet l'affichage du pied de page.
7 -->
8 <div class='col-md-12'>
9     <footer class='text-center'>
10         <hr>
11         <p>Romain Peretti - IDA-P4A - 4ème année</p>
12         <p>CFPT Informatique - Travail Pratique Individuel</p>
13         <a href='#'>Retour en haut</a>
14     </footer>
15 </div>
```

```

1 <!--
2 Titre : Barre de navigation
3 Date : 9 mai 2019
4 Auteur : Romain Peretti
5 Version : 1.0
6 Description : Cette page permet l'affichage de la barre de navigation avec les
  possibilités de se connecter, se déconnecter et s'inscrire qui apparaissent ou
  disparaissent suivant si on est connecté ou pas.
7 -->
8 <div class='col-md-12 p-0'>
9     <nav class='navbar navbar-expand-lg navbar-dark my-navbar'>
10         <a class='navbar-brand' href='../controllers/home.php'>Direct Prod</a>
11         <button class='navbar-toggler' type='button' data-toggle='collapse' data-
  -target='#navbarText' aria-controls='navbarText' aria-expanded='false' aria-
  label='Toggle navigation'>
12             <span class='navbar-toggler-icon'></span>
13         </button>
14         <div class='collapse navbar-collapse' id='navbarText'>
15             <ul class='navbar-nav mr-auto'>
16                 <li class='nav-item'>
17                     <a class='nav-link' href='../controllers/home.php'><span
  class='fa fa-home'></span> Accueil</a>
18                 </li>
19                 <?php if (isset($_SESSION['email'])) : ?>
20                     <li class='nav-item'>
21                         <a class='nav-link' href='../controllers/profil.php?
  email=<?= $_SESSION['email'] ?>'><span class='fa fa-user'></span> Profil</a>
22                     </li>
23                 <?php endif; ?>
24                 <?php if (isset($_SESSION['role']) && $_SESSION['role'] == '2')
  : ?>
25                     <li class='nav-item'>
26                         <a class='nav-link' href='../controllers/admin.php'><
  span class='fas fa-clipboard-list'></span> Administrateur</a>
27                     </li>
28                 <?php endif; ?>
29             </ul>
30             <?php if (isset($_SESSION['email'])) : ?>
31                 <span class='navbar-text'><a href='../controllers/logout.php'
  class='btn btn-outline-danger my-btn'><span class='fas fa-power-off'></span></a>
32             </span>
33             <?php else : ?>
34                 <span class='navbar-text'><a href='../controllers/signIn.php'
  class='btn btn-outline-primary my-btn'>Inscription</a></span>
35                 <span class='navbar-text'><a href='../controllers/login.php'
  class='btn btn-outline-primary my-btn'>Connexion</a></span>
36             <?php endif; ?>
37         </div>
38     </nav>
39 </div>

```

```
1 <?php
2 /*
3 Titre : Contrôleur du profil
4 Date : Lundi, 13 mai 2019
5 Auteur : Romain Peretti
6 Version : 1.0
7 Description : Cette page récupère tous les fichiers nécessaire pour le site
8 */
9
10 //Démarrage de la session
11 session_start();
12
13 //Variables de session pour éviter de se reconnecter tous le temps
14 //$_SESSION['email'] = 'romain.prтт@eduge.ch';
15 //$_SESSION['role'] = '2';
16
17 // Fichier de connexion à la base
18 require_once '../database/database.php';
19
20 // Fichier de constantes
21 require_once '../constants/constants.php';
22
23 // Classes
24 require_once '../classes/Advertisement.php';
25 require_once '../classes/User.php';
26 require_once '../classes/Rating.php';
27 require_once '../classes/Picture.php';
28
29 // Manageurs
30 require_once '../managers/advertisementManager.php';
31 require_once '../managers/userManager.php';
32 require_once '../managers/ratingManager.php';
33 require_once '../managers/pictureManager.php';
```

```
1 <?php
2 /*
3  Titre : Page de validation de la connexion
4  Date : Mardi, 14 mai 2019 / modifier le jeudi, 22 mai 2019
5  Auteur : Romain Peretti
6  Version : 2.0
7  Description : Cette page récupère les champs envoyer par la page de connexion
   et test la connexion. Elle retourne un code en json
8 */
9
10 require_once '../inc/inc.all.php';
11
12 header('Content-Type: application/json');
13
14 // Initialisation des variables
15 $email = '';
16 $password = '';
17
18 // Récupération des champs
19 if (isset($_POST['email'])) {
20     $email = filter_input(INPUT_POST, 'email', FILTER_SANITIZE_STRING);
21 }
22 if (isset($_POST['password'])) {
23     $password = filter_input(INPUT_POST, 'password', FILTER_SANITIZE_STRING);
24 }
25
26 // Récupération de l'utilisateur
27 $result = UserManager::Login($email, $password);
28
29 if ($result === null) {
30     // Utilisateur inconnue
31     echo '{"ReturnCode": 1, "Message": "Email inconnu."}';
32     exit();
33 }
34 if ($result) {
35     $u = UserManager::GetUserByEmail($email);
36     $_SESSION['email'] = $email;
37     $_SESSION['role'] = $u->roleCode;
38     echo '{"ReturnCode": 2, "Message": "Email et mot de passe valide"}';
39     exit();
40 }
41
42 // Email ou mot de passe invalide
43 echo '{"ReturnCode": 0, "Message": "Email ou mot de passe invalide"}';
44
```

```
1 <?php
2 /*
3 Titre : Page de test du manager "userManager"
4 Date : 13 mai 2019
5 Auteur : Romain Peretti
6 Version : 1.0
7 Description : Tests sur chacune des méthodes du manager.
8 */
9
10 require_once '../managers/userManager.php';
11
12 /**
13  * Test 1 - GetUserByEmail()
14  */
15 echo '<h3>Test 1 - GetUserByEmail()</h3>';
16
17 echo '<h4>1.1 Utilisateur existant</h4>';
18 // Récupération d'un utilisateur existant
19 $u = UserManager::GetUserByEmail('romain.prtt@eduge.ch');
20 if ($u === false) {
21     echo 'Problème lors de la récupération';
22 }
23 if ($u === null) {
24     echo 'L\'utilisateur recherché n\'existe pas';
25 }
26 if ($u) {
27     echo '<pre>' . var_dump($u) . '</pre>';
28 }
29
30 echo '<h4>1.2 Utilisateur inexistant</h4>';
31 // Récupération d'un utilisateur inexistant
32 $u = UserManager::GetUserByEmail('a');
33 if ($u === false) {
34     echo 'Problème lors de la récupération';
35 }
36 if ($u === null) {
37     echo 'L\'utilisateur recherché n\'existe pas';
38 }
39 if ($u) {
40     echo '<pre>' . var_dump($u) . '</pre>';
41 }
42
43 /**
44  * Test 2 - CreateUser()
45  */
46 echo '<h3>Test 2 - CreateUser()</h3>';
47
48 echo '<h4>2.1 Utilisateur valide</h4>';
49 // Ajout d'un utilisateur
50 $u = new User('claude@gmail.com', '123', 'Lausanne', 'Vaud', '1000', '4 Rue des
    sapins', 'J\'aime marcher', 'iahdguihvncniwherhsfsdfus', '1');
51
52 $result = UserManager::CreateUser($u);
53 if ($result === false) {
54     echo 'Problème lors de la création';
55 }
56 if ($result === null) {
57     echo 'L\'utilisateur existe déjà';
58 }
59 if ($result === true) {
```



```
60     echo 'Création d\'un utilisateur réussi';
61 }
62
63 echo '<h4>2.2 Utilisateur invalide (email déjà utilisé)</h4>';
64 // Ajout d'un utilisateur avec un email déjà utilisé
65 $u = new User('romain.prtt@eduge.ch', '123', 'Lausanne', 'Vaud', '1000', '4 Rue
    des sapins', 'J\'aime marcher', 'iahdhguihvcniuwhersfsdfus', '1');
66
67 $result = UserManager::CreateUser($u);
68 if ($result === false) {
69     echo 'Problème lors de la création';
70 }
71 if ($result === null) {
72     echo 'L\'utilisateur existe déjà';
73 }
74 if ($result === true) {
75     echo 'Création d\'un utilisateur réussi';
76 }
77
78 /**
79  * Test 3 - UpdateUser()
80  */
81 echo '<h3>Test 3 - UpdateUser()</h3>';
82
83 // Modification d'un utilisateur
84 $u = new User('claud@gmail.com', '123', 'Genève', 'Genève', '1243', '1 Rue des
    Bouleaux', 'J\'aime danser', 'iahdhguihvcniuwhersfsdfus', '1');
85
86 $result = UserManager::UpdateUser($u);
87 if ($result === false) {
88     echo 'Problème lors de la modification';
89 }
90 if ($result) {
91     echo 'Modification d\'un utilisateur réussi';
92 }
93
94 /**
95  * Test 4 - DeleteUser()
96  */
97 echo '<h3>Test 4 - DeleteUser()</h3>';
98
99 // Suppression d'un utilisateur
100 $result = UserManager::DeleteUser('claud@gmail.com');
101 if ($result === false) {
102     echo 'Problème lors de la suppression';
103 }
104 if ($result) {
105     echo 'Suppression d\'un utilisateur réussi';
106 }
107
108 /**
109  * Test 5 - UserExist()
110  */
111 echo '<h3>Test 5 - UserExist()</h3>';
112
113 echo '<h4>5.1 Utilisateur existant</h4>';
114 // Essaie avec un utilisateur existant
115 $result = UserManager::UserExist('romain.prtt@eduge.ch');
116 if ($result === false) {
117     echo 'Problème lors de la récupération';
```

```
118 }
119 if ($result === null) {
120     echo 'L\'utilisateur n\'existe pas';
121 }
122 if ($result) {
123     echo 'L\'utilisateur existe';
124 }
125
126 echo '<h4>5.2 Utilisateur inexistant</h4>';
127 // Essaie avec un utilisateur inexistant
128 $result = UserManager::UserExist('avcfdadsa@gmail.com');
129 if ($result === false) {
130     echo 'Problème lors de la récupération';
131 }
132 if ($result === null) {
133     echo 'L\'utilisateur n\'existe pas';
134 }
135 if ($result) {
136     echo 'L\'utilisateur existe';
137 }
138
139 /**
140  * Test 6 - Login()
141  */
142 echo '<h3>Test 6 - Login()</h3>';
143
144 $result = UserManager::Login('romain.prтт@eduge.ch', '123');
145 if ($result === false) {
146     echo 'Problème lors de la connexion';
147 }
148 if ($result === null) {
149     echo 'Mauvais mot de passe ou email';
150 }
151 if ($result) {
152     echo 'Connexion réussi';
153 }
154
155 /**
156  * Test 7 - UpdateAdminToUser()
157  */
158 echo '<h3>Test 5 - UpdateAdminToUser()</h3>';
159
160 $result = UserManager::UpdateAdminToUser('romain.prтт@eduge.ch');
161 if ($result === false) {
162     echo 'Problème lors de la modification';
163 }
164 if ($result === null) {
165     echo 'L\'administrateur a été modifié';
166 }
167
168 /**
169  * Test 8 - UpdateUserToAdmin()
170  */
171 echo '<h3>Test 5 - UpdateUserToAdmin()</h3>';
172
173 $result = UserManager::UpdateUserToAdmin('romain.prтт@eduge.ch');
174 if ($result === false) {
175     echo 'Problème lors de la modification';
176 }
177 if ($result) {
```

```
178     echo 'L\'utilisateur a été modifié';  
179 }
```

```
1 <?php
2 /*
3  Titre : Page de test du manager "ratingManager"
4  Date : 13 mai 2019
5  Auteur : Romain Peretti
6  Version : 1.0
7  Description : Tests sur chacune des méthodes du manager.
8  */
9
10 require_once '../managers/ratingManager.php';
11
12 // Id de l'annonce sur laquelle sera effectué les tests
13 $idAd = 42;
14
15 /**
16  * Test 1 - CreateRating()
17  */
18 echo '<h3>Test 1 - CreateRating()</h3>';
19
20 //Ajout d'une évaluation
21 $r = new Rating(null, 3, 'Très bonne farine pour faire du pain', null, 'romain.
prtt@eduge.ch', $idAd);
22
23 $result = RatingManager::CreateRating($r);
24 if ($result === false) {
25     echo 'Problème lors de la création';
26 }
27 if ($result) {
28     echo 'Création d\'une évaluation réussi';
29 }
30
31 /**
32  * Test 2 - GetRatingsOfAnAd()
33  */
34 echo '<h3>Test 2 - GetRatingsOfAnAd()</h3>';
35
36 $results = RatingManager::GetRatingsOfAnAd($idAd);
37 if ($results === false) {
38     echo 'Problème lors de la récupération';
39 }
40 if ($results) {
41     echo '<pre>' . var_dump($results[0]) . '</pre>';
42 }
43
44 /**
45  * Test 3 - GetScoreOfAnAd()
46  */
47 echo '<h3>Test 3 - GetScoreOfAnAd()</h3>';
48
49 $result = RatingManager::GetScoreOfAnAd($idAd);
50 if ($result === false) {
51     echo 'Problème lors de la récupération';
52 }
53 if ($result) {
54     echo '<pre>' . var_dump($result) . '</pre>';
55 }
56
57 /**
58  * Test 4 - DeleteRatingsOfAnAd()
59  */
```

```
60 echo '<h3>Test 4 - DeleteRatingsOfAnAd()</h3>';
61
62 $result = RatingManager::DeleteRatingsOfAnAd($idAd);
63 if ($result === false) {
64     echo 'Problème lors de la suppression';
65 }
66 if ($result) {
67     echo 'Suppression réussi';
68 }
69
70 /**
71  * Test 5 - DeleteRatingsOfUser()
72  */
73 echo '<h3>Test 5 - DeleteRatingsOfUser()</h3>';
74
75 $result = RatingManager::DeleteRatingsOfUser('jack@gmail.com');
76 if ($result === false) {
77     echo 'Problème lors de la suppression';
78 }
79 if ($result) {
80     echo 'Suppression réussi';
81 }
```

```
1 <?php
2 /*
3  Titre : Page de test du manageur "pictureManager"
4  Date : 13 mai 2019
5  Auteur : Romain Peretti
6  Version : 1.0
7  Description : Tests sur chacune des méthodes du manageur.
8  */
9
10 require_once '../managers/pictureManager.php';
11
12 // Id de l'annonce qui sera utilisée pour ces tests
13 $idAd = 34;
14
15 /**
16  * Test 1 - CreatePicture()
17  */
18 echo '<h3>Test 1 - CreatePicture()</h3>';
19 /*
20 $p = new Picture(null, base64_encode('Picture.jpg'), $idAd);
21
22 $result = PictureManager::CreatePicture($p);
23 if ($result === false) {
24     echo 'Problème lors de la création';
25 }
26 if ($result) {
27     echo 'Création d\'une image réussie';
28 }*/
29
30 /**
31  * Test 2 - GetPictureById()
32  */
33 echo '<h3>Test 2 - GetPictureById()</h3>';
34
35 $p = PictureManager::GetPictureById(1);
36 if ($p === false) {
37     echo 'Problème lors de la récupération';
38 }
39 if ($p) {
40     echo '<pre>' . var_dump($p) . '</pre>';
41     echo '';
42 }
43
44 /**
45  * Test 3 - GetPicturesByAdId()
46  */
47 echo '<h3>Test 3 - GetPicturesByAdId()</h3>';
48
49 $results = PictureManager::GetPicturesByAdId($idAd);
50 if ($results === false) {
51     echo 'Problème lors de la récupération';
52 }
53 if ($results) {
54     echo '<pre>' . var_dump($results[0]) . '</pre>';
55     echo '';
56 }
57
58 /**
59  * Test 4 - DeletePicture()
60  */
```

```
61 echo '<h3>Test 4 - DeletePicture()</h3>';
62
63 $result = PictureManager::DeletePicture(1);
64 if ($result === false) {
65     echo 'Problème lors de la suppression';
66 }
67 if ($result) {
68     echo 'Suppression d\'une image réussi';
69 }
70
71 /**
72  * Test 5 - DeletePicturesOfAnAd()
73  */
74 echo '<h3>Test 5 - DeletePicturesOfAnAd()</h3>';
75
76 $result = PictureManager::DeletePicturesOfAnAd($idAd);
77 if ($result === false) {
78     echo 'Problème lors de la suppression';
79 }
80 if ($result) {
81     echo 'Suppression des images réussi';
82 }
```

```
1 <?php
2 /*
3  Titre : Page de test du manager "advertisementManager"
4  Date : 13 mai 2019
5  Auteur : Romain Peretti
6  Version : 1.0
7  Description : Tests sur chacune des méthodes du manager.
8  */
9
10 require_once '../managers/advertisementManager.php';
11
12 /**
13  * Test 1 - GetValidatedAds()
14  */
15 echo '<h3>Test 1 - GetValidatedAds()</h3>';
16
17 // Récupération des annonces validées
18 $results = AdvertisementManager::GetValidatedAds();
19 if ($results === false) {
20     echo 'Problème lors de la récupération';
21 }
22 if ($results) {
23     echo '<pre>' . var_dump($results[0]) . '</pre>';
24 } else {
25     echo 'Aucune annonces';
26 }
27
28 /**
29  * Test 2 - GetInvalidAds()
30  */
31 echo '<h3>Test 2 - GetInvalidAds()</h3>';
32
33 // Récupération des annonces non validées
34 $results = AdvertisementManager::GetInvalidAds();
35 if ($results === false) {
36     echo 'Problème lors de la récupération';
37 }
38 if ($results) {
39     echo '<pre>' . var_dump($results[0]) . '</pre>';
40 } else {
41     echo 'Aucune annonces';
42 }
43
44 /**
45  * Test 3 - GetAdById()
46  */
47 echo '<h3>Test 3 - GetAdById()</h3>';
48
49 echo '<h4>3.1 Annonce existante</h4>';
50 // Récupération d'une annonce existante
51 $a = AdvertisementManager::GetAdById(7);
52 if ($a === false) {
53     echo 'Problème lors de la récupération';
54 }
55 if ($a === null) {
56     echo 'Annonce introuvable';
57 }
58 if ($a) {
59     echo '<pre>' . var_dump($a) . '</pre>';
60 }
```



```
61
62 echo '<h4>3.1 Annonce inexistante</h4>';
63 // Récupération d'une annonce inexistante
64 $a = AdvertisementManager::GetAdById(0);
65 if ($a === false) {
66     echo 'Problème lors de la récupération';
67 }
68 if ($a === null) {
69     echo 'Annonce introuvable';
70 }
71 if ($a) {
72     echo '<pre>' . var_dump($a) . '</pre>';
73 }
74
75 /**
76  * Test 4 - GetAdsByUserEmail()
77  */
78 echo '<h3>Test 4 - GetAdsByUserEmail()</h3>';
79
80 // Récupération des annonces d'un utilisateur
81 $results = AdvertisementManager::GetAdsByUserEmail('romain.prтт@eduge.ch');
82 if ($results === false) {
83     echo 'Problème lors de la récupération';
84 }
85 if ($results) {
86     echo '<pre>' . var_dump($results[0]) . '</pre>';
87 }
88
89 /**
90  * Test 5 - CreateAd()
91  */
92 echo '<h3>Test 5 - CreateAd()</h3>';
93
94 // Ajout d'une annonce
95 $a = new Advertisement(null, 'Farine de seigle 4', 'Farine complète', 0, null,
96     null, 'jack@gmail.com');
97 $result = AdvertisementManager::CreateAd($a);
98 if ($a === false) {
99     echo 'Problème lors de la création';
100 }
101 if ($a) {
102     echo 'Création d\'une annonce réussi';
103 }
104 //Id de l'annonce qu'on vient de créer, il est nécessaire de le modifier à
105 //chaque fois. Utiliser pour les trois derniers tests
106 $idAd = 50;
107
108 /**
109  * Test 6 - UpdateAd()
110  */
111 echo '<h3>Test 6 - UpdateAd()</h3>';
112
113 // Modification d'une annonce
114 $a = new Advertisement($idAd, 'Farine de seigle 4 A+', 'Farine doublement
115     complète', 1, null, null, 'jack@gmail.com');
116 $result = AdvertisementManager::UpdateAd($a);
117 if ($a === false) {
118     echo 'Problème lors de la modification';
119 }
```

```
118 if ($a) {
119     echo 'Modification d\'une annonce réussie';
120 }
121
122 /**
123  * Test 7 - UpdateAdToValid()
124  */
125 echo '<h3>Test 7 - UpdateAdToValid()</h3>';
126
127 //Validation de l'annonce
128 $result = AdvertisementManager::UpdateAdToValid($idAd);
129 if ($result === false) {
130     echo 'Problème lors de la validation';
131 }
132 if ($result) {
133     echo 'Annonce validée';
134 }
135
136 /**
137  * Test 8 - DeleteAd()
138  */
139 echo '<h3>Test 8 - DeleteAd()</h3>';
140
141 // Suppression d'une annonce
142 $result = AdvertisementManager::DeleteAd($idAd);
143 if ($result === false) {
144     echo 'Problème lors de la suppression';
145 }
146 if ($result) {
147     echo 'Annonce supprimée';
148 }
149
150 /**
151  * Test 9 - DeleteAdsOfUser()
152  */
153 echo '<h3>Test 9 - DeleteAdsOfUser()</h3>';
154 $result = AdvertisementManager::DeleteAdsOfUser('jack@gmail.com');
155 if ($result === false) {
156     echo 'Problème lors de la suppression';
157 }
158 if ($result) {
159     echo 'Annonces supprimées';
160 }
161
162 /**
163  * Test 10 - Research()
164  */
165 echo '<h3>Test 10 - Research()</h3>';
166 $results = AdvertisementManager::Research('geneve', false, 'city');
167 if ($results === false) {
168     echo 'Problème lors de la recherche';
169 }
170 if ($results) {
171     if ($results != array()) {
172         echo '<pre>' . var_dump($results[0]) . '</pre>';
173     } else {
174         echo '<p>Aucunes annonces associées à cette recherche</p>';
175     }
176 }
177
```

```

1  <!--
2  Titre : Page d'accueil
3  Date : 8 mai 2019
4  Auteur : Romain Peretti
5  Version : 1.0
6  Description : La vue de la page d'accueil contenant toutes les annonces valides
7  -->
8  <!DOCTYPE html>
9  <html>
10
11  <head>
12      <?php include_once '../inc/head.php'; ?>
13  </head>
14
15  <body>
16      <div class='container col-md-12'>
17          <div class='row'>
18              <?php include_once '../inc/navbar.php'; ?>
19          </div>
20          <div class='row justify-content-center'>
21              <div class='col-md-7'>
22                  <form method='POST' class='m-1'>
23                      <div class='form-row'>
24                          <div class='form-group col-3'>
25                              <select class="form-control m-1" name="searchOption
26                                  ">
27                                  <option value='city'>Ville</option>
28                                  <option value='canton'>Canton</option>
29                                  <option value='postCode'>Code postal</option>
30                                  <option value='title'>Titre de l'annonce</
31                                  option>
32                                  <option value='description'>Description de l'
33                                  annonce</option>
34                                  <option value='score'>Évaluation de l'annonce</
35                                  option>
36                              </select>
37                          </div>
38                          <div class='form-group col-6'>
39                              <input type='text' class='form-control m-1' name='
40                                  searchContent' placeholder="Recherche">
41                          </div>
42                          <div class='form-group col'>
43                              <button type='submit' class='form-control btn btn-
44                                  outline-primary my-btn m-1' name='search'><span class='fas fa-search'></span></
45                                  button>
46                          </div>
47                          <?php if (isset($_SESSION['email'])) : ?>
48                              <div class='form-group col'>
49                                  <a href='../controllers/createAd.php' class='
50                                      form-control btn btn-outline-success my-btn m-1'><span class='fas fa-plus-
51                                      circle'></span></a>
52                              </div>
53                          <?php endif; ?>
54                      </div>
55                      <div class='form-row m-1'>
56                          <div class='form-group'>
57                              <input type='checkbox' id='bio' name='searchOrganic
58                              '>
59                              <label for='bio'>Bio <span class="fa fa-seedling"
60                                  style='color: rgb(57, 192, 39)'></span></label>

```

```

50         </div>
51     </div>
52 </form>
53 </div>
54 </div>
55
56 <div class='row justify-content-center'>
57     <?php if (count($ads) > 0) :
58         foreach ($ads as $ad) :
59             $u = UserManager::getUserByEmail($ad->userEmail);
60             $score = RatingManager::getScoreOfAnAd($ad->
idAdvertisement);
61             ??
62             <div class='media border rounded col-md-10 my-ad'>
63                 <div class='media-body'>
64                     <h4><?=$ad->title ?></h4>
65                     <p class='text-justify'><?=$ad->description ?></p>
66
67                     <?php if ($ad->organic == 1) : ??>
68                         <span class="fa fa-seedling" style='color: rgb
(57, 192, 39)'></span>
69                         <label>Produit bio</label>
70                     <?php else : ??>
71                         <label>Produit non bio</label>
72                     <?php endif; ??>
73                     <p><?=$ad->streetAndNumber . ' ' . $u
->city . ', ' . $u->postCode . ' ' . $u->canton ?></p>
74                     <p><?=$ad->date_create($ad->
creationDate), 'd M Y \à H:i:s' ?></p>
75                     <p>Annonceur : <?=$ad->userEmail ?></p>
76                     <?php if ($score == 0) : ??>
77                         <p>Il n'y a aucune évaluation de cette annonce
</p>
78                     <?php else : ??>
79                         <p>Évaluation de l'annonce par les autres
utilisateurs :
80
81                         <?php for ($i = 1; $i <= MAX_STARS_RATING;
$i++) : ??>
82                             <?php if ($i <= $score) : ??>
83                                 <span class='fas fa-star' style='
color:gold'></span>
84                             <?php else : ??>
85                                 <span class='far fa-star'></span>
86                             <?php endif; ??>
87                         <?php endfor; ??>
88                     </p>
89                     <div class='row justify-content-end'>
90                         <a class='btn btn-outline-primary my-btn' href
='../controllers/adDetails.php?idAd=<?=$ad->idAdvertisement ?>'><span class='
fas fa-stream'></span></a>
91                         <?php if (isset($_SESSION['email']) && $ad->
userEmail == $_SESSION['email']) : ??>
92                             <a class='btn btn-outline-warning my-btn'
href='../controllers/updateAd.php?idAd=<?=$ad->idAdvertisement ?>'><span
class='fas fa-pencil-alt'></span></a>
93                             <a class='btn btn-outline-danger my-btn'
href='../controllers/deleteAd.php?idAd=<?=$ad->idAdvertisement ?>'><span
class='fas fa-trash-alt'></span></a>
94                         <?php endif; ??>

```

```
94         </div>
95     </div>
96 </div>
97     <?php endforeach; ?>
98     <?php else : ?>
99         <h2>Aucune annonce avec cette recherche</h2>
100     <?php endif; ?>
101 </div>
102 <div class='row'>
103     <?php include_once '../inc/footer.php'; ?>
104 </div>
105 </div>
106 </body>
107
108 </html>
```

```

1  <!--
2  Titre : Page d'administration des nouvelles annonces
3  Date : 9 mai 2019
4  Auteur : Romain Peretti
5  Version : 1.0
6  Description : La vue de la page d'administration contenant un tableaux avec
   toutes les nouvelles annonces en attente d'être validée
7  -->
8  <!DOCTYPE html>
9  <html>
10
11  <head>
12      <?php include_once '../inc/head.php'; ?>
13  </head>
14
15  <body>
16      <div class='container col-md-12'>
17          <div class='row'>
18              <?php include_once '../inc/navbar.php'; ?>
19          </div>
20          <div class='row justify-content-center'>
21              <div class='col-md-10 mb-5'>
22                  <?php if (isset($ads)) : ?>
23                      <div class='text-center'>
24                          <h2>Annonce en attente de validation</h2>
25                      </div>
26                      <table class='table table-striped table-responsive m-2'>
27                          <thead>
28                              <th scope='col'>#</th>
29                              <th scope='col'>Titre</th>
30                              <th scope='col'>Description</th>
31                              <th scope='col'>Date</th>
32                              <th scope='col'>Annonceur</th>
33                              <th scope='col'></span></th>
34                          </thead>
35                          <tbody>
36                              <?php foreach ($ads as $ad) :
37                                  $u = UserManager::getUserByEmail($ad->userEmail
38                                  );
39                                  $adsNumber++;
40                                  <?>
41                                  <tr class='my-tr'>
42                                      <th scope='row'><?= $adsNumber ?></th>
43                                      <td><?= $ad->title ?></td>
44                                      <td><?= $ad->description ?></td>
45                                      <td><?= date_format(date_create($ad->
46                                      creationDate), 'd M Y \à H:i:s') ?></td>
47                                      <td><?= $u->email ?></td>
48                                      <td class='text-center'>
49                                          <a class='btn btn-outline-primary my-
50                                          btn' href='../controllers/adDetails.php?idAd=<?= $ad->idAdvertisement ?>'><span
51                                          class='fas fa-stream'></span></a>
52                                          <a class='btn btn-outline-success my-
53                                          btn' href='../controllers/admin.php?idAd=<?= $ad->idAdvertisement ?>'><span
54                                          class='fas fa-check'></span></a>
55                                          <!-- Button trigger modal -->
56                                          <button type="button" class="btn btn-
57                                          outline-danger my-btn" data-toggle="modal" data-target="#exampleModal<?= $ad->
58                                          idAdvertisement ?>"><span class='fas fa-trash-alt'></span></button>
59                                          <!-- Modal -->

```

```

52         <div class="modal fade" id="
exampleModal<?= $ad->idAdvertisement ?>" tabindex="-1" role="dialog" aria-
labelledby="exampleModalLabel" aria-hidden="true">
53         <div class="modal-dialog" role="
document">
54         <div class="modal-content">
55         <div class="modal-header">
56         <h5 class="modal-title
" id="exampleModalLabel" style='color:black;'>Êtes-vous sur de vouloir
supprimer cette annonce ?</h5>
57         <button type="button"
class="close" data-dismiss="modal" aria-label="Close">
58         <span aria-hidden=
"true">&times;</span>
59         </button>
60         </div>
61         <div class="modal-footer">
62         <form method='POST'>
63         <button type="
button" class="btn btn-outline-secondary " data-dismiss="modal">Fermer</button>
64         <input type="
submit" class="btn btn-outline-danger" value='Confirmer' name='delete'>
65         <input type='
hidden' name='idAd' value='<?= $ad->idAdvertisement ?>'>
66         </form>
67         </div>
68         </div>
69         </div>
70         </div>
71         </td>
72         </tr>
73         <?php endforeach; ?>
74         </tbody>
75         </table>
76
77         <?php else : ?>
78         <h1 class='text-center'>Aucune nouvelles annonces</h1>
79         <?php endif; ?>
80         </div>
81     </div>
82     <div class='row justify-content-center'>
83         <div class='col-md-10'>
84             <?php if (isset($users)) : ?>
85                 <div class='text-center'>
86                     <h2>Liste des utilisateurs enregistrés</h2>
87                 </div>
88                 <table class='table table-striped m-2'>
89                     <thead>
90                         <th scope='col'>#</th>
91                         <th scope='col'>Email</th>
92                         <th scope='col'>Ville</th>
93                         <th scope='col'>Canton</th>
94                         <th scope='col'>Rue et N°</th>
95                         <th scope='col'>Description</th>
96                         <th scope='col' class='text-center'>Admin ?</th>
97                     </thead>
98                     <tbody>
99                         <?php foreach ($users as $u) :
100                             $usersNumber++;
101                             ?>

```

```

102         <tr class='my-tr'>
103             <th scope='row'><?= $usersNumber ?></th>
104             <td><?= $u->email ?></td>
105             <td><?= $u->city ?></td>
106             <td><?= $u->canton ?></td>
107             <td><?= $u->streetAndNumber ?></td>
108             <td><?= $u->description ?></td>
109             <td class='text-center'>
110                 <form method='POST'>
111                     <?php if ($u->roleCode ==
USER_CODE) : ?>
112                         <button type='submit' name='
updateToAdmin' class='btn btn-outline-success my-btn'>
113                             <span class='fas fa-check'
></span>
114                         </button>
115                         <input type='hidden' name='
userEmail' value='<?= $u->email ?>'>
116                     <?php else : ?>
117                         <button type='submit' name='
updateToUser' class='btn btn-outline-danger my-btn'>
118                             <span class='fas fa-times'
></span>
119                         </button>
120                         <input type='hidden' name='
userEmail' value='<?= $u->email ?>'>
121                     <?php endif; ?>
122                 </form>
123             </td>
124         <?php endforeach; ?>
125     </tbody>
126 </table>
127 <?php endif; ?>
128 </div>
129 </div>
130 <div class='row'>
131     <?php include_once '../inc/footer.php'; ?>
132 </div>
133 </div>
134 </body>
135 <script>
136     $('#myModal').on('shown.bs.modal', function() {
137         $('#myInput').trigger('focus')
138     });
139 </script>
140 </html>

```



```

1  <!--
2  Titre : Page de connexion
3  Date : 8 mai 2019
4  Auteur : Romain Peretti
5  Version : 1.0
6  Description : La vue de la page connexion contenant un formulaire
7  -->
8  <!DOCTYPE html>
9  <html>
10
11  <head>
12      <?php include_once '../inc/head.php'; ?>
13  </head>
14
15  <body>
16      <div class='container col-md-12'>
17          <div class='row'>
18              <?php include_once '../inc/navbar.php'; ?>
19          </div>
20          <div class='row justify-content-center'>
21              <div class='col-md-6'>
22                  <form class='m-2'>
23                      <div class='form-group'>
24                          <label for='email'>Adresse email</label>
25                          <input type='email' class='form-control my-input' id='
email' placeholder='Entrez un email valide'>
26                      </div>
27                      <div class='form-group'>
28                          <label for='password'>Mot de passe</label>
29                          <input type='password' class='form-control my-input' id
='password'>
30                      </div>
31                      <div class='form-group'>
32                          <small><a href='../controllers/signIn.php'>Pas encore
de compte ?</a></small>
33                      </div>
34                      <button type='submit' class='btn btn-outline-primary my-btn
' id='send'>Connexion</button>
35                  </form>
36              </div>
37          </div>
38          <div class='row'>
39              <?php include_once '../inc/footer.php'; ?>
40          </div>
41      </div>
42
43  </body>
44  <script>
45      const VALIDATE_URL = '../ajax/validateLogin.php';
46      const ON_SUCCESS_URL = '../controllers/home.php';
47      $(document).ready(function() {
48          $('#send').click(validateLogin)
49      });
50
51      /**
52       * Call-back lorsqu'on clique sur le bouton de connexion
53       */
54      function validateLogin(event) {
55          // Prevent form submission
56          if (event) {

```

```
57         event.preventDefault();
58     }
59     //Récupération de l'email
60     var email = $('#email').val();
61
62     //Récupération du mot de passe
63     var password = $('#password').val();
64
65     // Tests sur les champs du formulaire
66     if (email.length == 0) {
67         $('#email').focus(); // Met le focus sur le champ email
68
69         return;
70     }
71     if (password.length == 0) {
72         $('#password').focus(); // Met le focus sur le champ password
73
74         return;
75     }
76
77     $.ajax({
78         method: 'POST',
79         url: VALIDATE_URL,
80         data: {
81             'email': email,
82             'password': password
83         },
84         dataType: 'json',
85         success: function(data) {
86             switch (data.ReturnCode) {
87                 case 0:
88                     swal('Mauvais email ou mot de passe', '', 'error')
89                     break;
90                 case 1:
91                     swal('Mauvais email ou mot de passe', '', 'error')
92                     break;
93                 case 2:
94                     location.href = ON_SUCCESS_URL;
95                     break;
96             }
97         }, // #end success
98         error: function(jqXHR) {
99             msg = 'Une erreur est survenue : ';
100             switch (jqXHR.status) {
101                 case 200:
102                     msg = msg + jqXHR.status + ' json invalide.'
103                     break;
104                 case 404:
105                     msg = msg + jqXHR.status + ' page introuvable.'
106                     break;
107             }
108         } // #end error
109     });
110 }
111 </script>
112
113 </html>
```

```

1  <!--
2  Titre : Page de détails de l'utilisateur
3  Date : 9 mai 2019
4  Auteur : Romain Peretti
5  Version : 1.0
6  Description : La vue de la page connexion contenant les détails de l'
   utilisateur ainsi que toutes ses annonces
7  -->
8  <!DOCTYPE html>
9  <html>
10
11  <head>
12      <?php include_once '../inc/head.php'; ?>
13  </head>
14
15  <body>
16      <div class='container col-md-12'>
17          <div class='row'>
18              <?php include_once '../inc/navbar.php'; ?>
19          </div>
20          <div class='row justify-content-center'>
21              <div class='media border col-md-10 my-profil'>
22                  <div class='media-body'>
23                      <h2><?= $u->email ?></h2>
24                      <p>Ville : <?= $u->city ?></p>
25                      <p>Canton : <?= $u->canton ?></p>
26                      <p>Code postal : <?= $u->postCode ?></p>
27                      <p>Rue et n° : <?= $u->streetAndNumber ?></p>
28                      <p class='text-justify'>Description : <?= $u->description ?>
29
30                      <div class='row justify-content-end'>
31                          <a class='btn btn-outline-warning my-btn' href='../
   controllers/updateProfil.php?email=<?= $u->email ?>'><span class='fas fa-pencil
   -alt'></span></a>
32                          <a class='btn btn-outline-danger my-btn' href='../
   controllers/deleteProfil.php?email=<?= $u->email ?>'><span class='fas fa-trash-
   alt'></span></a>
33                      </div>
34                  </div>
35              </div>
36              <div class='row justify-content-center'>
37                  <?php foreach ($ads as $ad) :
38                      $u = UserManager::getUserByEmail($ad->userEmail);
39                      $score = RatingManager::getScoreOfAnAd($ad->idAdvertisement)
40                      ?>
41                      <div class='media border rounded col-md-5 my-ad'>
42                          <div class='media-body'>
43                              <h4><?= $ad->title ?></h4>
44                              <p class='text-justify'><?= $ad->description ?></p>
45                              <?php if ($ad->organic == 1) : ?>
46                                  <span class="fa fa-seedling" style='color: rgb(57,
   192, 39)'></span>
47                              <label>Produit bio</label>
48                              <?php else : ?>
49                                  <label>Produit non bio</label>
50                              <?php endif; ?>
51                              <p><?= 'Vente au ' . $u->streetAndNumber . ' ' . $u->
   city . ', ' . $u->postCode . ' ' . $u->canton ?></p>
52                              <p><?= 'Posté le ' . date_format(date_create($ad->

```

```

52 creationDate), 'd M Y \à H:i:s') ?></p>
53         <p>Annonce : <?= ($ad->valid == 1 ? 'Validée' : 'Pas
validée') ?></p>
54         <?php if ($score == 0) : ?>
55             <p>Il n'y a aucune évaluation de cette annonce</p>
56         <?php else : ?>
57             <p>Évaluation de l'annonce par les autres
utilisateurs :
58                 <?php for ($i = 1; $i <= MAX_STARS_RATING; $i
++): ?>
59                     <?php if ($i <= $score) : ?>
60                         <span class='fas fa-star' style='color
:gold'></span>
61                     <?php else : ?>
62                         <span class='far fa-star'></span>
63                     <?php endif; ?>
64                 <?php endfor; ?>
65             </p>
66         <?php endif; ?>
67         <div class='row justify-content-end'>
68             <a class='btn btn-outline-primary my-btn' href='
../controllers/adDetails.php?idAd=<?= $ad->idAdvertisement ?>'><span class='
fas fa-stream'></span></a>
69             <a class='btn btn-outline-warning my-btn' href='
../controllers/updateAd.php?idAd=<?= $ad->idAdvertisement ?>'><span class='fas
fa-pencil-alt'></span></a>
70             <a class='btn btn-outline-danger my-btn' href='../
controllers/deleteAd.php?idAd=<?= $ad->idAdvertisement ?>'><span class='fas fa
-trash-alt'></span></a>
71         </div>
72     </div>
73 </div>
74 <?php endforeach; ?>
75 </div>
76 <div class='row'>
77     <?php include_once '../inc/footer.php'; ?>
78 </div>
79 </div>
80 </body>
81
82 </html>

```

```

1  <!--
2  Titre : Page d'inscription
3  Date : 8 mai 2019
4  Auteur : Romain Peretti
5  Version : 1.0
6  Description : La vue de la page d'inscription contenant un formulaire
7  -->
8  <!DOCTYPE html>
9  <html>
10
11  <head>
12      <?php include_once '../inc/head.php'; ?>
13  </head>
14
15  <body>
16      <div class='container col-md-12'>
17          <div class='row'>
18              <?php include_once '../inc/navbar.php'; ?>
19          </div>
20          <div class='row justify-content-center'>
21              <div class="col-md-8">
22                  <div class="media-body">
23                      <form method="POST" class="m-2">
24                          <div class="form-group">
25                              <label for="email">Adresse email</label>
26                              <input type="email" class="form-control my-input"
name="email" id="email" placeholder="Entrez un email valide" required value="<?=
(isset($_POST['email']) ? $_POST['email'] : '') ?>">
27                          </div>
28                          <div class="form-group">
29                              <label for="password">Mot de passe</label>
30                              <input type="password" class="form-control my-input
" name="password" id="password" required>
31                          </div>
32                          <div class="form-group">
33                              <label for="repeatPassword">Confirmer le mot de
passe</label>
34                              <input type="password" class="form-control my-input
" name="repeatPassword" id="repeatPassword" required>
35                          </div>
36                          <div class="form-group">
37                              <div class="row">
38                                  <div class="col-md-6">
39                                      <label for="canton">Canton</label>
40                                      <input type="text" class="form-control my-
input" name="canton" id="canton" placeholder="Vaud" required value="<?= (isset(
$_POST['canton']) ? $_POST['canton'] : '') ?>">
41                                  </div>
42                                  <div class="col-md-6">
43                                      <label for="city">Ville</label>
44                                      <input type="text" class="form-control my-
input" name="city" id="city" placeholder="Lausanne" required value="<?= (isset(
$_POST['city']) ? $_POST['city'] : '') ?>">
45                                  </div>
46                              </div>
47                          </div>
48                          <div class="form-group">
49                              <div class="row">
50                                  <div class="col-md-3">
51                                      <label for="postCode">Code postal</label>

```

```

52         <input type="text" class="form-control my-
input" name="postCode" id="postCode" placeholder="1007" required value="<?=(
isset($_POST['postCode']) ? $_POST['postCode'] : '') ?>">
53     </div>
54     <div class="col-md-9">
55         <label for="streetAndNumber">Rue et numéro
</label>
56         <input type="text" class="form-control my-
input" name="streetAndNumber" id="streetAndNumber" placeholder="14 Rue Jean-
Dujardin" required value="<?=(isset($_POST['streetAndNumber']) ? $_POST['
streetAndNumber'] : '') ?>">
57     </div>
58 </div>
59 </div>
60 <div class="form-group">
61     <label for="description">Description</label>
62     <textarea name="description" class="form-control
my-input" id="description"><?=(isset($_POST['description']) ? $_POST['
description'] : '') ?></textarea>
63 </div>
64 <div class="form-group">
65     <small><a href="../controllers/login.php">Déjà un
compte ?</a></small>
66 </div>
67     <button type="submit" class="btn btn-outline-primary
my-btn" name="send">Inscription</button>
68 </form>
69 </div>
70 </div>
71 </div>
72 <div class="row">
73     <?php include_once '../inc/footer.php'; ?>
74 </div>
75 </div>
76 </body>
77
78 </html>

```

```

1  <!--
2  Titre : Page de création d'une annonce
3  Date : 8 mai 2019
4  Auteur : Romain Peretti
5  Version : 1.0
6  Description : La vue de la page de création d'une annonce contenant le
   formulaire à remplir
7  -->
8  <!DOCTYPE html>
9  <html>
10
11  <head>
12      <?php include_once '../inc/head.php'; ?>
13  </head>
14
15  <body>
16      <div class='container col-md-12'>
17          <div class='row'>
18              <?php include_once '../inc/navbar.php'; ?>
19          </div>
20          <div class='row justify-content-center'>
21              <div class='col-md-6'>
22                  <form method='POST' enctype='multipart/form-data' class='m-4'>
23                      <div class="form-group">
24                          <label for="title">Titre</label>
25                          <input type="text" class="form-control my-input" name="
title" id="title" placeholder="Salade Romaine" required>
26                      </div>
27                      <div class="form-group">
28                          <label for="description">Description</label>
29                          <textarea name="description" class="form-control my-
input" id="description" required></textarea>
30                      </div>
31                      <div class="form-group">
32                          <input type="checkbox" name="organic" id="organic"
value="true">
33                          <label for="organic">Produit bio</label>
34                      </div>
35                      <div class="form-group">
36                          <label for="file">Image(s) du produit</label>
37                          <input type="file" class="form-control-file" id="file"
name='pictures[]' accept='image/jpeg, image/png, image/jpg' multiple>
38                      </div>
39                      <button type="submit" class="btn btn-outline-primary my-btn
" name="send">Créer</button>
40                  </form>
41              </div>
42          </div>
43          <div class='row'>
44              <?php include_once '../inc/footer.php'; ?>
45          </div>
46      </div>
47  </body>
48
49  </html>

```

```
1 <!--
2 Titre : Page de suppression d'une annonce
3 Date : 8 mai 2019
4 Auteur : Romain Peretti
5 Version : 1.0
6 Description : La vue de la page de suppression d'une annonce contenant un
  formulaire de validation
7 -->
8 <!DOCTYPE html>
9 <html>
10
11 <head>
12     <?php include_once '../inc/head.php'; ?>
13 </head>
14
15 <body>
16     <div class='container col-md-12'>
17         <div class='row'>
18             <?php include_once '../inc/navbar.php'; ?>
19         </div>
20         <div class='row justify-content-center'>
21             <div class='col-md-6'>
22                 <form method='POST' class='m-4'>
23                     <fieldset class='form-group'>
24                         <legend class='col-form-label'>Voulez-vous vraiment
25 supprimer l'annonce intitulée '<?= $a->title ?>' ?</legend>
26                         <div class="form-row">
27                             <a href='../controllers/adDetails.php?idAd=<?= $a->
28 idAdvertisement ?>' class='btn btn-outline-primary my-btn'>Annuler</a>
29                             <input type='submit' class='btn btn-outline-danger
30 my-btn' name='send' value='Confirmer'>
31                         </div>
32                     </fieldset>
33                 </form>
34             </div>
35         </div>
36     </div>
37 </body>
38
39 </html>
```



```

1  <!--
2  Titre : Page de modification d'une annonce
3  Date : 8 mai 2019
4  Auteur : Romain Peretti
5  Version : 1.0
6  Description : La vue de la page de modification d'une annonce contenant le
   formulaire rempli avec les données de la base
7  -->
8  <!DOCTYPE html>
9  <html>
10
11  <head>
12      <?php include_once '../inc/head.php'; ?>
13  </head>
14
15  <body>
16      <div class='container col-md-12'>
17          <div class='row'>
18              <?php include_once '../inc/navbar.php'; ?>
19          </div>
20          <div class='row justify-content-center'>
21              <div class='col-md-6'>
22                  <form method='POST' enctype='multipart/form-data' class='m-4'>
23                      <div class="form-group">
24                          <label for="title">Titre</label>
25                          <input type="text" class="form-control my-input" name="
title" id="title" required value="<?= $ad->title ?>">
26                      </div>
27                      <div class="form-group">
28                          <label for="description">Description</label>
29                          <textarea name="description" class="form-control my-
input" id="description" required><?= $ad->description ?></textarea>
30                      </div>
31                      <div class="form-group">
32                          <input type="checkbox" name="organic" id="organic"
value="true" <?= ($ad->organic == ORGANIC ? 'checked' : '') ?>>
33                          <label for="organic">Produit bio</label>
34                      </div>
35                      <div class="form-group">
36                          <label for="file">Image(s) du produit</label>
37                          <input type="file" class="form-control-file" id="file"
name='pictures[]' accept='image/jpeg, image/png, image/jpg' multiple>
38                      </div>
39                      <div class='form-group'>
40                          <input type="submit" class="btn btn-outline-primary my-
btn" name="send" value='Modifier'>
41                      </div>
42                  </form>
43              </div>
44          </div>
45          <div class='row justify-content-center'>
46              <div class='row col-md-6'>
47                  <?php foreach ($pictures as $p) : ?>
48                      <div class='text-center col-md-4'>
49                          <img src='<?= $p->picture ?>' class='img-fluid' alt='
imgProduct'>
50
51                          <!-- Button trigger modal -->
51                          <button type="button" class="btn btn-outline-danger my-
btn" data-toggle="modal" data-target="#exampleModal<?= $p->idPicture ?>">
52

```

X

```

53         </button>
54         <!-- Modal -->
55         <div class="modal fade" id="exampleModal"= $p-&gt;
idPicture ?&gt;" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel"
aria-hidden="true"&gt;
56             &lt;div class="modal-dialog" role="document"&gt;
57                 &lt;div class="modal-content"&gt;
58                     &lt;div class="modal-header"&gt;
59                         &lt;h5 class="modal-title" id="
exampleModalLabel" style='color:black;'&gt;Êtes-vous sur de vouloir supprimer
cette image ?&lt;/h5&gt;
60                         &lt;button type="button" class="close"
data-dismiss="modal" aria-label="Close"&gt;
61                             &lt;span aria-hidden="true"&gt;&amp;times;&lt;/
span&gt;
62                         &lt;/button&gt;
63                     &lt;/div&gt;
64                     &lt;div class="modal-footer"&gt;
65                         &lt;form method='POST'&gt;
66                             &lt;button type="button" class="btn
btn-outline-secondary" data-dismiss="modal"&gt;Fermer&lt;/button&gt;
67                             &lt;input type="submit" class="btn
btn-outline-danger" value='Confirmer' name='delete'&gt;
68                             &lt;input type='hidden' name='idPic'
value=&lt;<?= $p-&gt;idPicture ?&gt;'&gt;
69                         &lt;/form&gt;
70                     &lt;/div&gt;
71                 &lt;/div&gt;
72             &lt;/div&gt;
73         &lt;/div&gt;
74     &lt;/div&gt;
75     &lt;?php endforeach; ?&gt;
76 &lt;/div&gt;
77 &lt;/div&gt;
78 &lt;div class='row justify-content-center'&gt;
79     &lt;p&gt;&lt;i&gt;* Attention, la suppression des images est définitive&lt;/i&gt;&lt;/p&gt;
80 &lt;/div&gt;
81 &lt;div class='row'&gt;
82     &lt;?php include_once '../inc/footer.php'; ?&gt;
83 &lt;/div&gt;
84 &lt;/div&gt;
85 &lt;/body&gt;
86 &lt;script&gt;
87     $('#myModal').on('shown.bs.modal', function() {
88         $('#myInput').trigger('focus')
89     });
90 &lt;/script&gt;
91
92
93 &lt;/html&gt;
</pre

```

```

1  <!--
2  Titre : Page de détails d'une annonce
3  Date : 8 mai 2019
4  Auteur : Romain Peretti
5  Version : 1.0
6  Description : La vue de la page de détails d'une annonce contenant un les
7                informations et les images de l'annonce
8                ainsi que les commentaires des autres utilisateurs.
9  -->
10 <!DOCTYPE html>
11 <html>
12 <head>
13     <?php include_once '../inc/head.php'; ?>
14 </head>
15
16 <body>
17     <div class='container col-md-12'>
18         <div class='row'>
19             <?php include_once '../inc/navbar.php'; ?>
20         </div>
21         <div class='row justify-content-center'>
22             <div class='media border rounded col-md-10 m-1'>
23                 <div class='media-body'>
24                     <h2><?= $ad->title ?></h2>
25                     <p class='text-justify'><?= $ad->description ?></p>
26                     <?php if ($ad->organic == 1) : ?>
27                         <span class="fa fa-seedling" style='color: rgb(57, 192,
39);'></span>
28                         <label>Produit bio</label>
29                     <?php else : ?>
30                         <label>Produit non bio</label>
31                     <?php endif; ?>
32                     <p>Vente au <?= $u->streetAndNumber . ', ' . $u->postCode .
33                         ' ' . $u->canton ?></p>
34                     <p>Posté le <?= date_format(date_create($ad->creationDate),
35                         'd M Y \à H:i:s') ?></p>
36                     <p>Annonceur : <?= $ad->userEmail ?></p>
37                     <?php if ($score == 0) : ?>
38                         <p>Il n'y a aucune évaluation de cette annonce</p>
39                     <?php else : ?>
40                         <p>Évaluation de l'annonce par les autres utilisateurs
41                         :
42                         <?php for ($i = 1; $i <= 5; $i++) : ?>
43                             <?php if ($i <= $score) : ?>
44                                 <span class='fas fa-star' style='color:gold
45                                     '></span>
46                             <?php else : ?>
47                                 <span class='far fa-star'></span>
48                             <?php endif; ?>
49                         <?php endfor; ?>
50                     </p>
51                     <?php endif; ?>
52                     <div class='row'>
53                         <?php foreach ($pics as $p) : ?>
54                             <div class=' text-center col-md-4'>
55                                 <a href='../views/showFullPicture.php?id=<?= $p
56                                     ->idPicture ?>'>
57                                     <img src='<?= $p->picture ?>' class='img-
58                                     fluid'>

```

```

53         </a>
54     </div>
55     <?php endforeach; ?>
56 </div>
57 <?php if (isset($_SESSION['email']) && $_SESSION['email']
== $ad->userEmail) : ?>
58     <div class='row justify-content-end'>
59         <a class='btn btn-outline-warning my-btn' href='
../controllers/updateAd.php?idAd=?= $ad->idAdvertisement ?>'><span class='fas
fa-pencil-alt'></span></a>
60         <a class='btn btn-outline-danger my-btn' href='../
controllers/deleteAd.php?idAd=?= $ad->idAdvertisement ?>'><span class='fas fa
-trash-alt'></span></a>
61     </div>
62 <?php endif; ?>
63 </div>
64 </div>
65 </div>
66 <div class='row justify-content-center'>
67     <?php foreach ($rates as $rate) : ?>
68         <div class='col-md-5 border m-1'>
69             <h4>?<= $rate->userEmail ?</h4>
70             <p class='text-justify'>?<= $rate->comment ?</p>
71             <p>
72                 <?php for ($i = 1; $i <= 5; $i++) : ?>
73                     <?php if ($i <= $rate->rating) : ?>
74                         <span class='fas fa-star' style='color:gold'>
</span>
75                     <?php else : ?>
76                         <span class='far fa-star'></span>
77                     <?php endif; ?>
78                 <?php endfor; ?>
79             </p>
80         </div>
81     <?php endforeach; ?>
82 </div>
83 <?php if (isset($_SESSION['email'])) : ?>
84     <div class='row justify-content-center'>
85         <div class='col-md-5 border m-1'>
86             <h3>Poster un commentaire</h3>
87             <form method='POST'>
88                 <div class='form-group row'>
89                     <label for='comment' class='col-sm-2 col-form-
label'>Commentaire</label>
90                     <div class='col-sm-10'>
91                         <textarea class='form-control' id='comment'
name='comment' required></textarea>
92                     </div>
93                 </div>
94                 <fieldset class='form-group'>
95                     <div class='row'>
96                         <legend class='col-sm-2 col-form-label pt-0'>
Note</legend>
97                         <div class='col-sm-10'>
98                             <div class='form-check form-check-inline'>
99                                 <input class='form-check-input' type='
radio' name='grade' id='radio1' value='1' checked>
100                                 <label class='form-check-label' for='
radio1'>1</label>
101                         </div>

```

```
102         <div class='form-check form-check-inline'>
103             <input class='form-check-input' type='
radio' name='grade' id='radio2' value='2'>
104             <label class='form-check-label' for='
radio1'>2</label>
105         </div>
106         <div class='form-check form-check-inline'>
107             <input class='form-check-input' type='
radio' name='grade' id='radio3' value='3'>
108             <label class='form-check-label' for='
radio1'>3</label>
109         </div>
110         <div class='form-check form-check-inline'>
111             <input class='form-check-input' type='
radio' name='grade' id='radio4' value='4'>
112             <label class='form-check-label' for='
radio1'>4</label>
113         </div>
114         <div class='form-check form-check-inline'>
115             <input class='form-check-input' type='
radio' name='grade' id='radio5' value='5'>
116             <label class='form-check-label' for='
radio1'>5</label>
117         </div>
118     </div>
119 </div>
120 </fieldset>
121 <div class="form-group row">
122     <div class="col-sm-10">
123         <button type="submit" class="btn btn-outline-
primary my-btn" name='send'>Envoyer</button>
124     </div>
125 </div>
126 </form>
127 </div>
128 </div>
129 <?php endif; ?>
130 <div class='row'>
131     <?php include_once '../inc/footer.php'; ?>
132 </div>
133 </div>
134 </body>
135
136 </html>
```

```
1 <?php
2 require_once '../inc/inc.all.php';
3
4 $idPic = filter_input(INPUT_GET, 'id', FILTER_SANITIZE_NUMBER_INT);
5 $p = PictureManager::GetPictureById($idPic);
6 ?>
7
8 <!DOCTYPE html>
9 <html>
10 <head>
11     <?php
12     $pageName = 'Image';
13     include_once '../inc/head.php';
14     ?>
15 </head>
16 <body>
17     <div class='row col-12 justify-content-center'>
18         <div style='margin-top: 50vh;transform: translateY(-50%);'>
19             <img src='<?= $p->picture ?>' class='mx-auto d-block'>
20         </div>
21     </div>
22 </body>
23
24 </html>
```

```
1 <!--
2 Titre : Page de suppression d'un utilisateur
3 Date : 9 mai 2019
4 Auteur : Romain Peretti
5 Version : 1.0
6 Description : La vue de la page de suppression d'une annonce contenant un
  formulaire de validation
7 -->
8 <!DOCTYPE html>
9 <html>
10
11 <head>
12     <?php include_once '../inc/head.php'; ?>
13 </head>
14
15 <body>
16     <div class='container col-md-12'>
17         <div class='row'>
18             <?php include_once '../inc/navbar.php'; ?>
19         </div>
20         <div class='row justify-content-center'>
21             <div class='col-md-6'>
22                 <form method='POST' class='m-4'>
23                     <fieldset class='form-group'>
24                         <legend class='col-form-label'>Voulez-vous vraiment
supprimer votre compte (<?= $u->email ?>) ?</legend>
25                         <div class="form-row">
26                             <a href='../controllers/profil.php?email=<?=
$_SESSION['email'] ?>' class='btn btn-outline-primary my-btn'>Annuler</a>
27                             <input type='submit' class='btn btn-outline-danger
my-btn' name='send' value='Confirmer'>
28                         </div>
29                     </fieldset>
30                 </form>
31             </div>
32         </div>
33         <div class='row'>
34             <?php include_once '../inc/footer.php'; ?>
35         </div>
36     </div>
37 </body>
38
39 </html>
```

```

1  <!--
2  Titre : Page de modification de son profil
3  Date : 9 mai 2019
4  Auteur : Romain Peretti
5  Version : 1.0
6  Description : La vue de la page de modification de son profil contenant le
   formulaire rempli avec les données de la base
7  -->
8  <!DOCTYPE html>
9  <html>
10
11  <head>
12      <?php include_once '../inc/head.php'; ?>
13  </head>
14
15  <body>
16      <div class='container col-md-12'>
17          <div class='row'>
18              <?php include_once '../inc/navbar.php'; ?>
19          </div>
20          <div class='row justify-content-center'>
21              <div class="col-md-8">
22                  <div class="media-body">
23                      <form method="POST" class="m-4">
24                          <div class="form-group">
25                              <label for="newPassword">Nouveau mot de passe</
label>
26                              <input type="password" class="form-control my-input
" name="newPassword" id="newPassword">
27                          </div>
28                          <div class="form-group">
29                              <label for="repeatNewPassword">Confirmer le mot de
passe</label>
30                              <input type="password" class="form-control my-input
" name="repeatNewPassword" id="repeatNewPassword">
31                          </div>
32                          <div class="form-group">
33                              <div class="row">
34                                  <div class="col-md-6">
35                                      <label for="canton">Canton</label>
36                                      <input type="text" class="form-control my-
input" name="canton" id="canton" required value='<?= (isset($_POST['canton']) ?
$_POST['canton'] : $u->canton) ?>'>
37                                  </div>
38                                  <div class="col-md-6">
39                                      <label for="city">Ville</label>
40                                      <input type="text" class="form-control my-
input" name="city" id="city" required value='<?= (isset($_POST['city']) ? $_POST
['city'] : $u->city) ?>'>
41                                  </div>
42                              </div>
43                          </div>
44                          <div class="form-group">
45                              <div class="row">
46                                  <div class="col-md-3">
47                                      <label for="postCode">Code postal</label>
48                                      <input type="text" class="form-control my-
input" name="postCode" id="postCode" required value='<?= (isset($_POST['postCode
']) ? $_POST['postCode'] : $u->postCode) ?>'>
49                                  </div>

```



```
50         <div class="col-md-9">
51             <label for="streetAndNumber">Rue et numéro
52         </label>
53         <input type="text" class="form-control my-
54         input" name="streetAndNumber" id="streetAndNumber" required value='<?=(isset(
55         $_POST['streetAndNumber']) ? $_POST['streetAndNumber'] : $u->streetAndNumber)
56         ?>'>
57     </div>
58 </div>
59 <div class="form-group">
60     <label for="description">Description</label>
61     <textarea name="description" class="form-control
62     my-input" id="description"><?=(isset($_POST['description']) ? $_POST['
63     description'] : $u->description) ?></textarea>
64 </div>
65 <div class="form-group">
66     <label for="password">Entrez votre mot de passe
67     actuel pour appliquer les modifications</label>
68     <input type="password" class="form-control my-
69     input" name="password" id="password" required>
70 </div>
71 <button type="submit" class="btn btn-outline-primary
72     my-btn" name="send">Modifier</button>
73 </form>
74 </div>
75 </div>
76 <div class="row">
77     <?php include_once '../inc/footer.php'; ?>
78 </div>
79 </div>
80 </body>
81 </html>
```

```
1 <?php
2 /*
3  Titre : Constantes de connexion à la base de donnée
4  Date : Mercredi, 8 mai 2019
5  Auteur : Romain Peretti
6  Version : 1.0
7  Description : Cette page contient les constantes pour la connexion à la base de
   données
8  */
9
10 DEFINE('DB_USER', 'adminDirectProd');
11 DEFINE('DB_PASSWORD', 'SuperTPI');
12 DEFINE('DB_PORT', '3306');
13 DEFINE('DB_HOST', 'localhost');
14 DEFINE('DB_NAME', 'direct_prod');
15 DEFINE('DB_TYPE', 'mysql');
```

```

1  <?php
2
3  /* Titre : Classe "User"
4   * Date : Jeudi, 09.05.2019
5   * Auteurs : Romain Peretti
6   * Version : 1.0
7   * Description : Création de la classe "User" et initialisation des principaux
   champs de cette classe
8   */
9
10 /**
11  * @brief    Objet User
12  * @remark  Cet objet est utilisé comme conteneur en référence avec la table "
   users"
13  *
14  *      Exemple d'utilisation 1
15  *      $u = new User();
16  *      $u->email = "romain.prтт@eduge.ch";
17  *      $u->password = SHA1("1234");
18  *      $u->city = "Genève";
19  *      $u->canton = "Genève";
20  *      $u->postCode = "1232";
21  *      $u->streetAndNumber = "Joseph-Berthet 14";
22  *      $u->description = "J'aime la photographie";
23  *      $u->salt = "aocnNfNfgoitrosOAIrERXccpvw$w";
24  *      $u->roles_code = 1;
25  *
26  *
27  *      Exemple d'utilisation 2
28  *      $u = new User("romain.prтт@eduge.ch", SHA1("1234"), "Genève", "
   Genève", "1232", "Joseph-Berthet 14", "J'aime la photographie", "
   aocnNfNfgoitrosOAIrERXccpvw$w", 1);
29  */
30 class User
31 {
32
33     /**
34      * @brief    Le Constructor appelé au moment de la création de l'objet. Ie.
   new User();
35      * @param InEmail    L'email de l'utilisateur. Default ""
36      * @param InPassword  Le mot de passe de l'utilisateur. Default ""
37      * @param InCity     La ville de l'utilisateur. Default ""
38      * @param InCanton   Le canton de l'utilisateur. Default ""
39      * @param InPostCode  Le code postal de l'utilisateur. Default ""
40      * @param InStreetAndNumber  Le numéro et le nom de la rue de l'
   utilisateur. Default ""
41      * @param InDescription  La description de l'utilisateur. Default ""
42      * @param InSalt     Le salt du mot de passe de l'utilisateur. Default ""
43      * @param InRoleCode  Le code du rôle de l'utilisateur. Default ""
44      */
45     public function __construct($InEmail = "", $InPassword = "", $InCity = "",
   $InCanton = "", $InPostCode = "", $InStreetAndNumber = "", $InDescription = "",
   $InSalt = "", $InRoleCode = "")
46     {
47         $this->email = $InEmail;
48         $this->password = $InPassword;
49         $this->city = $InCity;
50         $this->canton = $InCanton;
51         $this->postCode = $InPostCode;
52         $this->streetAndNumber = $InStreetAndNumber;

```

```
53     $this->description = $InDescription;
54     $this->salt = $InSalt;
55     $this->roleCode = $InRoleCode;
56 }
57
58 /** @var string L'email de l'utilisateur */
59 public $email;
60
61 /** @var string Le mot de passe encrypté de l'utilisateur */
62 public $password;
63
64 /** @var string La ville de l'utilisateur */
65 public $city;
66
67 /** @var string Le canton de l'utilisateur */
68 public $canton;
69
70 /** @var string Le code postal de l'utilisateur */
71 public $postCode;
72
73 /** @var string La rue et le numérc de l'utilisateur */
74 public $streetAndNumber;
75
76 /** @var string La description de l'utilisateur */
77 public $description;
78
79 /** @var string Le hash utilisé pour le mdp de l'utilisateur */
80 public $salt;
81
82 /** @var int Code du role de l'utilisateur */
83 public $roleCode;
84 }
85
```

```

1  <?php
2
3  /* Titre : Classe "Rating"
4   * Date : Jeudi, 09.05.2019
5   * Auteurs : Romain Peretti
6   * Version : 1.0
7   * Description : Création de la classe "Rating" et initialisation des
   principaux champs de cette classe
8   */
9
10 /**
11  * @brief   Objet Rating
12  * @remark  Cet objet est utilisé comme conteneur en référence avec la table "
   rates"
13  *
14  *           Exemple d'utilisation 1
15  *           $r = new Rating();
16  *           $r->idRate = null;
17  *           $r->rating = 4;
18  *           $r->comment = "Ce blé est vraiment idéal pour faire du pain";
19  *           $r->postDate = "2019-05-13 10:00:00";
20  *           $r->userEmail = "romain.prтт@eduge.ch";
21  *           $r->idAdvertisement = 1;
22  *
23  *
24  *           Exemple d'utilisation 2
25  *           $r = new Rating(null, 4, "Ce blé est vraiment idéal pour faire du
   pain", "2019-05-13 10:00:00", "romain.prтт@eduge.ch", 1);
26  */
27 class Rating
28 {
29     /**
30      * @brief   Le Constructor appelé au moment de la création de l'objet. Ie.
   new Rating();
31      * @param InIdRate   L'id de l'évaluation. Default null
32      * @param InRating    La note donné par l'utilisateur. Default ""
33      * @param InComment   Le commentaire. Default ""
34      * @param InPostDate  La date de création de l'évaluation. Default ""
35      * @param InUserEmail L'email de l'utilisateur ayant mis un commentaire.
   Default ""
36      * @param InIdAdvertisement L'id de l'annonce auquel l'évaluation est
   assignée. Default ""
37      */
38     public function __construct($InIdRate = null, $InRating = "", $InComment = "
   ", $InPostDate = "", $InUserEmail = "", $InIdAdvertisement = "")
39     {
40         $this->idRate = $InIdRate;
41         $this->rating = $InRating;
42         $this->comment = $InComment;
43         $this->postDate = $InPostDate;
44         $this->userEmail = $InUserEmail;
45         $this->idAdvertisement = $InIdAdvertisement;
46     }
47
48     /** @var int L'id de l'évaluation */
49     public $idRate;
50
51     /** @var int La note donné par l'utilisateur */
52     public $rating;
53

```

```
54     /** @var string Le commentaire de l'utilisateur */
55     public $comment;
56
57     /** @var int La date de création du commentaire */
58     public $postDate;
59
60     /** @var string L'email de l'utilisateur */
61     public $userEmail;
62
63     /** @var int L'id de l'annonce */
64     public $idAdVERTISEMENT;
65 }
66
```

```
1 <?php
2
3 /* Titre : Classe "Picture"
4  * Date : Lundi, 13.05.2019
5  * Auteurs : Romain Peretti
6  * Version : 1.0
7  * Description : Création de la classe "Picture" et initialisation des
   principaux champs de cette classe
8  */
9
10 /**
11  * @brief   Objet Picture
12  * @remark Cet objet est utilisé comme conteneur en référence avec la table "
   pictures"
13  *
14  *   Exemple d'utilisation 1
15  *   $p = new Picture();
16  *   $p->idPicture = 1;
17  *   $p->picture = BASE64(myImage.png);
18  *   $p->idAdvertisement = 1;
19  *
20  *
21  *   Exemple d'utilisation 2
22  *   $p = new Picture(1, BASE64(myImage.png), 1);
23  */
24 class Picture
25 {
26     /**
27     * @brief   Le Constructor appelé au moment de la création de l'objet. Ie.
   new Rating();
28     * @param InIdPicture   L'id de l'image. Default ""
29     * @param InPicture     L'image. Default ""
30     * @param InIdAdvertisement L'id de l'annonce auquel l'image est
   assignée. Default ""
31     */
32     public function __construct($InIdPicture = "", $InPicture = "",
   $InIdAdvertisement = "")
33     {
34         $this->idPicture = $InIdPicture;
35         $this->picture = $InPicture;
36         $this->idAdvertisement = $InIdAdvertisement;
37     }
38
39     /** @var int L'id de l'image */
40     public $idPicture;
41
42     /** @var string L'image encodée en base 64 */
43     public $picture;
44
45     /** @var int L'id de l'annonce auquel est assignée l'image */
46     public $idAdvertisement;
47 }
48
```

```

1  <?php
2
3  /* Titre : Classe "Advertisement"
4   * Date : Jeudi, 09.05.2019
5   * Auteurs : Romain Peretti
6   * Version : 1.0
7   * Description : Création de la classe "Advertisement" et initialisation des
   principaux champs de cette classe
8   */
9
10 /**
11  * @brief   Objet Advertisement
12  * @remark  Cet objet est utilisé comme conteneur en référence avec la table "
   advertisements"
13  *
14  *           Exemple d'utilisation 1
15  *           $a = new Advertisement();
16  *           $a->idAdvertisement = null;
17  *           $a->title = "Farine de blé";
18  *           $a->description = "Farine de blé complet";
19  *           $a->organic = true;
20  *           $a->valid = true;
21  *           $a->creationDate = "2019-05-13 10:00:00";
22  *           $a->userEmail = "romain.prtt@eduge.ch";
23  *
24  *
25  *           Exemple d'utilisation 2
26  *           $a = new Advertisement(null, "Farine de blé", "Farine de blé
   complet", true, true, "2019-05-13 10:00:00", "romain.prtt@eduge.ch");
27  */
28 class Advertisement
29 {
30
31     /**
32     * @brief   Le Constructor appelé au moment de la création de l'objet. Ie.
   new Advertisement();
33     * @param InIdAdvertisement      L'id de l'annonce. Default null
34     * @param InTitle                Le Titre de l'annonce. Default ""
35     * @param InDescription          La description de l'annonce. Default ""
36     * @param InOrganic              Un bool définissant si l'annonce est organique ou
   non. Default ""
37     * @param InValid                Un bool définissant si l'annonce est validé par l'
   admin. Default null
38     * @param InCreationDate         La date de création de l'annonce. Default null
39     * @param InUserEmail            L'email de l'utilisateur auquel l'annonce est
   assigné. Default ""
40     */
41     public function __construct($InIdAdvertisement = null, $InTitle = "",
   $InDescription = "", $InOrganic = "", $InValid = null, $InCreationDate = null,
   $InUserEmail = "")
42     {
43         $this->idAdvertisement = $InIdAdvertisement;
44         $this->title = $InTitle;
45         $this->description = $InDescription;
46         $this->organic = $InOrganic;
47         $this->valid = $InValid;
48         $this->creationDate = $InCreationDate;
49         $this->userEmail = $InUserEmail;
50     }
51

```



```
52     /** @var int L'id de l'annonce */
53     public $idAdVERTISEMENT;
54
55     /** @var string Le titre de l'annonce */
56     public $title;
57
58     /** @var string La description de l'annonce */
59     public $description;
60
61     /** @var bool Si l'annonce est organique ou pas */
62     public $organic;
63
64     /** @var bool Si l'annonce est toujours valide */
65     public $valid;
66
67     /** @var string La date de création de l'annonce */
68     public $creationDate;
69
70     /** @var int L'email de l'utilisateur ayant créé l'annonce */
71     public $userEmail;
72 }
73
```

```

1  <?php
2  /*
3  Auteur : M. Aigroz
4  Date : 3 avril 2019
5  */
6
7
8  require_once '../config/config.php';
9
10 /**
11  * Retourne un objet PDO connecté à la base de données
12  * @return \PDO
13  */
14 class Database
15 {
16
17     private static $pdoInstance;
18
19     /**
20      * @brief Class Constructor - Créer une nouvelle connexion à la database
21      * si la connexion n'existe pas
22      * On la met en privé pour que personne puisse créer une nouvelle
23      * instance via ' = new Database();'
24      */
25     private function __construct()
26     { }
27
28     /**
29      * @brief Comme le constructeur, on rend __clone privé pour que personne
30      * ne puisse cloner l'instance
31      */
32     private function __clone()
33     { }
34
35     /**
36      * @brief Retourne l'instance de la Database ou créer une connexion
37      * initiale
38      * @return $objInstance;
39      */
40     public static function getInstance()
41     {
42         if (self::$pdoInstance == null) {
43             try {
44                 $dsn = DB_TYPE . ' :host=' . DB_HOST . ' ;port=' . DB_PORT . ' ;
45                 dbname=' . DB_NAME . ' ;charset=utf8;';
46                 self::$pdoInstance = new PDO($dsn, DB_USER, DB_PASSWORD, null);
47                 self::$pdoInstance->setAttribute(PDO::ATTR_ERRMODE, PDO::
48                 ERRMODE_EXCEPTION);
49             } catch (PDOException $e) {
50                 echo "EDatabase Error: " . $e->getMessage();
51             }
52         }
53         return self::$pdoInstance;
54     }
55
56     # end method
57
58     /**
59      * @brief Passes on any static calls to this class onto the singleton PDO
60      * instance
61      * @param $chrMethod The method to call
62

```

```
54      * @param    $arrArguments    The method's parameters
55      * @return   $mix              The method's return value
56      */
57
58      final public static function __callStatic($chrMethod, $arrArguments)
59      {
60          $pdo = self::getInstance();
61          return call_user_func_array(array($pdo, $chrMethod), $arrArguments);
62      }
63
64      # end method
65 }
66
```

```
1 <?php
2 /*
3 Titre : Manageur des utilisateurs
4 Date : 9 mai 2019
5 Auteur : Romain Peretti
6 Version : 1.0
7 Description : Cette classe contient toutes les fonctions en rapport avec les
   utilisateur utilisées sur le site.
8             Chaque fonctions envoie une requête SQL
9 */
10
11 require_once '../database/database.php';
12 require_once '../classes/User.php';
13 require_once '../managers/AdvertismentManager.php';
14
15 class UserManager
16 {
17
18     /**
19      * @brief Retourne un utilisateur en fonction de son email
20      * @param string email L'email de l'utilisateur qu'on recherche.
21      * @return [User] L'utilisateur de type User.
22      *         False Une erreur est survenue.
23      *         Null L'utilisateur n'est pas trouvé.
24      */
25     public static function GetUserByEmail($email)
26     {
27         $u = null;
28
29         //Initialisation de la requête
30         $req = 'SELECT email, password, city, canton, postCode, streetAndNumber
   , description, salt, roleCode FROM users WHERE email = :e';
31         $statement = Database::prepare($req);
32
33         try {
34             $statement->execute(array(':e' => $email));
35         } catch (PDOException $e) {
36             return false;
37         }
38
39         //On récupère le premier élément
40         if ($row = $statement->fetch(PDO::FETCH_ASSOC, PDO::FETCH_ORI_NEXT)) {
41             $u = new User($row['email'], $row['password'], $row['city'], $row['
   canton'], $row['postCode'], $row['streetAndNumber'], $row['description'], $row['
   salt'], $row['roleCode']);
42         }
43
44         return $u;
45     }
46
47     /**
48      * @brief Retourne tous les utilisateurs (non admin)
49      * @return array Un tableau contenant tous les utilisateurs de type "User"
50      *         False Une erreur est survenue.
51      */
52     public static function GetUsers()
53     {
54         $arr = array();
55
56         //Initialisation de la requête
```

```

57     $req = 'SELECT email, password, city, canton, postCode,
streetAndNumber, description, salt, roleCode FROM users';
58     $statement = Database::prepare($req);
59
60     try {
61         $statement->execute();
62     } catch (PDOException $e) {
63         return false;
64     }
65
66     while ($row = $statement->fetch(PDO::FETCH_ASSOC)) {
67         //On crée une variable de la classe User
68         $u = new User($row['email'], $row['password'], $row['city'], $row['
canton'], $row['postCode'], $row['streetAndNumber'], $row['description'], $row[
'salt'], $row['roleCode']);
69         //Et on l'ajoute dans le tableau qu'on retourne à la fin
70         array_push($arr, $u);
71     }
72
73     return $arr;
74 }
75
76 /**
77  * @brief Ajoute un utilisateur
78  * @param [User] L'objet User que l'on veut insérer.
79  * @return boolean True Ok.
80  *          False Une erreur est survenue.
81  *          Null L'utilisateur existe déjà.
82  */
83 public static function CreateUser($u)
84 {
85     //Vérification que l'email n'est pas déjà utilisé
86     if (userManager::UserExist($u->email)) {
87         return null;
88     }
89
90     //Initialisation de la requête
91     $req = 'INSERT INTO users (email, password, city, canton, postCode,
streetAndNumber, description, salt, roleCode) VALUES (:e, :pwd, :ci, :ca, :po
, :st, :d, :sa, :r)';
92     $statement = Database::prepare($req);
93
94     //Création d'un hash
95     $salt = uniqid(mt_rand(), true);
96     //Encryption du mot de passe et du hash en sha1
97     $encryptedPassword = sha1($u->password . $salt);
98     $roles = USER_CODE;
99
100    try {
101        $statement->execute(array(
102            ':e' => $u->email,
103            ':pwd' => $encryptedPassword,
104            ':ci' => $u->city,
105            ':ca' => $u->canton,
106            ':po' => $u->postCode,
107            ':st' => $u->streetAndNumber,
108            ':d' => $u->description,
109            ':sa' => $salt,
110            ':r' => $roles
111        ));

```

```

112     } catch (PDOException $e) {
113         return false;
114     }
115
116     return true;
117 }
118
119 /**
120  * @brief Modifie un utilisateur
121  * @param [User] L'objet User que l'on veut modifier.
122  * @return boolean True Ok.
123  *          False Une erreur est survenue.
124  */
125 public static function updateUser($u)
126 {
127     //Initialisation de la requête
128     $req = 'UPDATE users SET password = :pwd, city = :ci, canton = :ca,
postCode = :po, streetAndNumber = :st, description = :d, salt = :s WHERE email
= :e';
129     $statement = Database::prepare($req);
130
131     //Création d'un nouveau hash
132     $salt = uniqid(mt_rand(), true);
133     //Encryption du mot de passe et du nouveau hash en sha1
134     $encryptedPassword = sha1($u->password . $salt);
135
136     try {
137         $statement->execute(array(
138             ':pwd' => $encryptedPassword,
139             ':ci' => $u->city,
140             ':ca' => $u->canton,
141             ':po' => $u->postCode,
142             ':st' => $u->streetAndNumber,
143             ':d' => $u->description,
144             ':s' => $salt,
145             ':e' => $u->email
146         ));
147     } catch (PDOException $e) {
148         return false;
149     }
150
151     return true;
152 }
153
154 /**
155  * @brief Modifie un utilisateur en administrateur
156  * @param string email L'email de l'utilisateur que l'on veut modifier.
157  * @return boolean True Ok.
158  *          False Une erreur est survenue.
159  */
160 public static function updateUserToAdmin($email)
161 {
162     //Initialisation de la requête
163     $req = 'UPDATE users SET roleCode = ' . ADMINISTRATOR_CODE . ' WHERE
email = :e';
164     $statement = Database::prepare($req);
165
166     try {
167         $statement->execute(array(':e' => $email));
168     } catch (PDOException $e) {

```

```

169         return false;
170     }
171
172     return true;
173 }
174
175 /**
176  * @brief Modifie un administrateur en utilisateur
177  * @param string email L'email de l'utilisateur que l'on veut modifier.
178  * @return boolean True Ok.
179  *          False Une erreur est survenue.
180  */
181 public static function UpdateAdminToUser($email)
182 {
183     //Initialisation de la requête
184     $req = 'UPDATE users SET roleCode = ' . USER_CODE . ' WHERE email = :e';
185
186     $statement = Database::prepare($req);
187
188     try {
189         $statement->execute(array(':e' => $email));
190     } catch (PDOException $e) {
191         return false;
192     }
193
194     return true;
195 }
196
197 /**
198  * @brief Supprime un utilisateur en fonction de son email
199  * @param string email L'email de l'utilisateur qu'on veut supprimer.
200  * @return boolean True Ok.
201  *          False Une erreur est survenue.
202  */
203 public static function DeleteUser($email)
204 {
205     //Initialisation de la requête
206     $req = 'DELETE FROM users WHERE email = :e';
207     $statement = Database::prepare($req);
208
209     Database::beginTransaction();
210
211     try {
212         // Suppression des annonces et des évaluations de l'utilisateur
213         if (AdvertisementManager::DeleteAdsOfUser($email) == false) {
214             Database::rollBack();
215             return false;
216         }
217         $statement->execute(array(':e' => $email));
218         Database::commit();
219         return true;
220     } catch (PDOException $e) {
221         Database::rollBack();
222     }
223
224     return false;
225 }
226
227 /**
228  * @brief Vérifie si un utilisateur existe déjà avec l'email spécifié
229  * @param string email L'email de l'utilisateur qu'on recherche.

```

```
228      * @return boolean True L'utilisateur existe.
229      *                  False Une erreur est survenue.
230      *                  Null L'utilisateur n'existe pas.
231      */
232      public static function UserExist($email)
233      {
234          //Initialisation de la requête
235          $req = 'SELECT * FROM users WHERE email = :e';
236          $statement = Database::prepare($req);
237
238          try {
239              $statement->execute(array(':e' => $email));
240          } catch (PDOException $e) {
241              return false;
242          }
243
244          //On vérifie si la requête retourne quelque chose
245          if (count($statement->fetchAll()) > 0) {
246              return true;
247          } else {
248              return false;
249          }
250      }
251
252      /**
253      * @brief Essaie de connecter l'utilisateur
254      * @param string email L'email de l'utilisateur.
255      * @param string password Le mot de passe de l'utilisateur.
256      * @return boolean True La connexion a réussi.
257      *                  False Une erreur est survenue.
258      *                  Null L'utilisateur n'existe pas
259      */
260      public static function Login($email, $password)
261      {
262          // Vérification que l'email est utilisé
263          if (!UserManager::UserExist($email)) {
264              return null;
265          }
266
267          //Initialisation de la requête
268          $req = 'SELECT password, salt FROM users WHERE email = :e';
269          $statement = Database::prepare($req);
270
271          try {
272              $statement->execute(array(':e' => $email));
273          } catch (PDOException $e) {
274              return false;
275          }
276
277          if ($row = $statement->fetch(PDO::FETCH_ASSOC)) {
278              //Vérification que le mot de passe entré correspond bien
279              if (sha1($password . $row['salt']) == $row['password']) {
280                  return true;
281              }
282          }
283
284          return false;
285      }
286  }
287
```



```

1  <?php
2  /*
3  Titre : Manageur des évaluations
4  Date : 9 mai 2019
5  Auteur : Romain Peretti
6  Version : 1.0
7  Description : Cette classe contient toutes les fonctions en rapport avec les
   commentaires et les notes utilisées sur le site.
8      Chaque fonctions envoie une requête SQL
9  */
10
11  require_once '../database/database.php';
12  require_once '../classes/Rating.php';
13
14  class RatingManager
15  {
16      /**
17       * @brief Retourne toutes les commentaires et notes en rapport avec l'
   annonce spécifié
18       * @param int idAd L'id de l'annonce qu'on recherche
19       * @return array Un tableau contenant toutes les commentaires et notes de
   type "Rating".
20       *      False Une erreur est survenue.
21       */
22       public static function GetRatingsOfAnAd($idAd)
23       {
24           $arr = array();
25
26           //Initialisation de la requête
27           $req = 'SELECT idRate, rating, comment, postDate, email,
   idAdvertisement FROM rates WHERE idAdvertisement = :idAd';
28           $statement = Database::prepare($req);
29
30           try {
31               $statement->execute(array(':idAd' => $idAd));
32           } catch (PDOException $e) {
33               return false;
34           }
35
36           //Tant qu'il y a des entrées
37           while ($row = $statement->fetch(PDO::FETCH_ASSOC)) {
38               //On crée une variable de la classe Rating
39               $r = new Rating(null, $row['rating'], $row['comment'], $row['
   postDate'], $row['email'], $row['idAdvertisement']);
40               //Et on l'ajoute dans le tableau qu'on retourne à la fin
41               array_push($arr, $r);
42           }
43
44           return $arr;
45       }
46
47       /**
48       * @brief Retourne le score donné par tous les commentaires des utilisateurs
49       * @param int idAd L'id de l'annonce qu'on recherche
50       * @return float Un nombre avec un chiffre derrière la virgule étant la
   moyenne de tous les commentaires.
51       *      False Une erreur est survenue.
52       */
53       public static function GetScoreOfAnAd($idAd)
54       {

```

```

55     $score = 0;
56
57     //Initialisation de la requête
58     $req = 'SELECT AVG(rating) AS score FROM rates WHERE idAdvertisement
= :idAd';
59     $statement = Database::prepare($req);
60
61     try {
62         $statement->execute(array(':idAd' => $idAd));
63     } catch (PDOException $e) {
64         return false;
65     }
66
67     if ($row = $statement->fetch(PDO::FETCH_ASSOC, PDO::FETCH_ORI_NEXT)) {
68         // On garde qu'une décimale
69         $score = number_format($row['score'], 0);
70     }
71
72     return $score;
73 }
74
75 /**
76  * @brief Ajout d'un commentaire et d'une note
77  * @param [Rating] Le commentaire et la note de type "Rating"
78  * @return boolean True Ok.
79  *         False Une erreur est survenue.
80  */
81 public static function CreateRating($r)
82 {
83     //Initialisation de la requête
84     $req = 'INSERT INTO rates (rating, comment, email, idAdvertisement)
VALUES (:r, :c, :e, :idAd)';
85     $statement = Database::prepare($req);
86
87     try {
88         $statement->execute(array(
89             ':e' => $r->userEmail,
90             ':idAd' => $r->idAdvertisement,
91             ':r' => $r->rating,
92             ':c' => $r->comment
93         ));
94     } catch (PDOException $e) {
95         return false;
96     }
97
98     return true;
99 }
100
101 /**
102  * @brief Suppression de toutes les évaluations d'une annonce
103  * @param int idAd L'id de l'annonce.
104  * @return boolean True Ok.
105  *         False Une erreur est survenue.
106  */
107 public static function DeleteRatingsOfAnAd($idAd)
108 {
109     //Initialisation de la requête
110     $req = 'DELETE FROM rates WHERE idAdvertisement = :id';
111     $statement = Database::prepare($req);
112     try {

```

```
113         $statement->execute(array(':id' => $idAd));
114     } catch (PDOException $e) {
115         return false;
116     }
117     return true;
118 }
119
120 /**
121  * @brief Suppression de toutes les évaluations d'un utilisateur
122  * @param string email L'email de l'utilisateur.
123  * @return boolean True Ok.
124  *          False Une erreur est survenue.
125  */
126 public static function DeleteRatingsOfUser($email)
127 {
128     //Initialisation de la requête
129     $req = 'DELETE FROM rates WHERE email = :e';
130     $statement = Database::prepare($req);
131
132     try {
133         $statement->execute(array(':e' => $email));
134     } catch (PDOException $e) {
135         return false;
136     }
137
138     return true;
139 }
140 }
141
```

```

1  <?php
2  /*
3  Titre : Manageur des images
4  Date : Lundi, 13 mai 2019
5  Auteur : Romain Peretti
6  Version : 1.0
7  Description : Cette classe contient toutes les fonctions en rapport avec les
   images utilisées sur le site.
8               Chaque fonctions envoie une requête SQL
9  */
10
11  require_once '../database/database.php';
12  require_once '../classes/Picture.php';
13
14  class PictureManager
15  {
16      /**
17       * @brief Retourne l'annonce que l'on recherche par l'identifiant de son
   annonceur
18       * @param int idPicture L'id de l'image qu'on recherche.
19       * @return [Picture] L'image de type "Picture".
20       *          False Une erreur est survenue.
21       *          Null Aucune image.
22       */
23       public static function GetPictureById($idPicture)
24       {
25           $p = null;
26
27           $req = 'SELECT idPicture, picture, idAdvertisement FROM pictures WHERE
   idPicture = :id';
28           $statement = Database::prepare($req);
29
30           try {
31               $statement->execute(array(':id' => $idPicture));
32           } catch (PDOException $e) {
33               return false;
34           }
35
36           //Récupération le premier élément
37           if ($row = $statement->fetch(PDO::FETCH_ASSOC, PDO::FETCH_ORI_NEXT)) {
38               $p = new Picture($row['idPicture'], $row['picture'], $row['
   idAdvertisement']);
39           }
40
41           return $p;
42       }
43
44       /**
45       * @brief Retourne les images en rapport avec l'annonce spécifié
46       * @param int idPicture L'id de l'annonce qu'on recherche.
47       * @return array Un tableau contenant toutes les images.
48       *          False Une erreur est survenue.
49       */
50       public static function GetPicturesByAdId($idAd)
51       {
52           $arr = array();
53
54           //Initialisation de la requête
55           $req = 'SELECT idPicture, picture, idAdvertisement FROM pictures WHERE
   idAdvertisement = :id';

```

```

56     $statement = Database::prepare($req);
57
58     try {
59         $statement->execute(array(':id' => $idAd));
60     } catch (PDOException $e) {
61         return false;
62     }
63
64     //Tant qu'il y a des entrées
65     while ($row = $statement->fetch(PDO::FETCH_ASSOC)) {
66         //On crée une variable de la classe Picture
67         $p = new Picture($row['idPicture'], $row['picture'], $row['
idAdvertisement']);
68         //Et on l'ajoute dans le tableau qu'on retourne à la fin
69         array_push($arr, $p);
70     }
71
72     return $arr;
73 }
74
75 /**
76  * @brief Ajout d'une image
77  * @param int idAd L'id de l'annonce auquel est lié l'image.
78  * @return boolean True Ok.
79  *          False Une erreur est survenue.
80  */
81 public static function CreatePicture($idAd)
82 {
83     for ($i = 1; $i <= count($_FILES['pictures']['name']); $i++) {
84
85         // Index pour la superglobal $_FILES car on commence avec $i = 1
86         $index = $i - 1;
87
88         // Si la taille de fichier est inférieure à 3 Mo et que son
extension fait partie des extensions autorisés
89         if ($_FILES['pictures']['size'][$index] < MAX_FILE_SIZE && in_array
($_FILES['pictures']['type'][$index], EXTENSION_AUTHORIZED)) {
90
91             // vérification que le transfert c'est bien déroulé
92             if (!isset($_FILES['pictures']['name'][$index]) || !
is_uploaded_file($_FILES['pictures']['tmp_name'][$index])) {
93                 echo ('<div class="alert alert-danger mb-0" role="alert">
Probleme de transfert - fichier(s) invalide(s)</div>');
94                 return false;
95             }
96             // Récupération du fichier temporaire
97             $data = file_get_contents($_FILES['pictures']['tmp_name'][$
$index]);
98             // Récupération du MIME
99             $finfo = new finfo(FILEINFO_MIME_TYPE); //Pas oublier dans php
.ini de décommenter "extension=php_fileinfo.dll"
100             $mime = $finfo->file($_FILES['pictures']['tmp_name'][$index]);
101             // Création de la chaîne en base 64
102             $src = 'data:' . $mime . ';base64,' . base64_encode($data);
103
104             // Initialisation de la requête
105             $req = 'INSERT INTO pictures(picture, idAdvertisement) VALUES
(:p, :id)';
106             $statement = Database::prepare($req);
107

```

```

108         try {
109             if ($statement->execute(array(':p' => $src, ':id' => $idAd
))) {
110                 if (count($_FILES['pictures']['name']) == $i) {
111                     return true;
112                 }
113             }
114         } catch (PDOException $e) {
115             return false;
116         }
117     }
118 }
119 }
120
121 /**
122  * @brief Suppression d'une image
123  * @param int idPictu re L'id de l'image à supprimer.
124  * @return boolean True Ok.
125  *          False Une erreur est survenue.
126  */
127 public static function DeletePicture($idPicture)
128 {
129     //Initialisation de la requête
130     $req = 'DELETE FROM pictures WHERE idPicture = :id';
131     $statement = Database::prepare($req);
132
133     try {
134         $statement->execute(array(':id' => $idPicture));
135     } catch (PDOException $e) {
136         return false;
137     }
138
139     return true;
140 }
141
142 /**
143  * @brief Suppression des images lié à une annonce
144  * @param int idAd L'id de l'annonce.
145  * @return boolean True Ok.
146  *          False Une erreur est survenue.
147  */
148 public static function DeletePicturesOfAnAd($idAd)
149 {
150     //Initialisation de la requête
151     $req = 'DELETE FROM pictures WHERE idAdvertisement = :id';
152     $statement = Database::prepare($req);
153     try {
154         $statement->execute(array(':id' => $idAd));
155     } catch (PDOException $e) {
156         return false;
157     }
158     return true;
159 }
160 }
161

```

```

1  <?php
2  /*
3  Titre : Manageur des annonces
4  Date : 9 mai 2019
5  Auteur : Romain Peretti
6  Version : 1.0
7  Description : Cette classe contient toutes les fonctions en rapport avec les
   annonces utilisées sur le site.
8      Chaque fonctions envoie une requête SQL
9  */
10
11 require_once '../database/database.php';
12 require_once '../classes/Advertisement.php';
13 require_once '../managers/pictureManager.php';
14 require_once '../managers/ratingManager.php';
15
16 class AdvertisementManager
17 {
18     /**
19      * @brief Retourne toutes les annonces validées par l'administrateur
20      * @return array Un tableau contenant toutes les annonces de type "
   Advertisement".
21      * False Une erreur est survenue.
22      */
23     public static function GetValidatedAds()
24     {
25         $arr = array();
26
27         //Initialisation de la requête
28         $req = 'SELECT idAdvertisement, title, description, organic, valid,
   creationDate, email FROM advertisements WHERE valid = 1 ORDER BY creationDate
   DESC';
29         $statement = Database::prepare($req);
30
31         try {
32             $statement->execute();
33         } catch (PDOException $e) {
34             return false;
35         }
36
37         //Tant qu'il y a des entrées
38         while ($row = $statement->fetch(PDO::FETCH_ASSOC)) {
39             //On crée une variable de la classe Advertisement
40             $a = new Advertisement($row['idAdvertisement'], $row['title'], $row[
   'description'], $row['organic'], $row['valid'], $row['creationDate'], $row['
   email']);
41             //Et on l'ajoute dans le tableau qu'on retourne à la fin
42             array_push($arr, $a);
43         }
44
45         return $arr;
46     }
47
48     /**
49      * @brief Retourne toutes les annonces non validées par l'administrateur
50      * @return array Un tableau contenant toutes les annonces de type "
   Advertisement".
51      * False Une erreur est survenue.
52      */
53     public static function GetInvalidAds()

```

```

54     {
55         $arr = array();
56
57         //Initialisation de la requête
58         $req = 'SELECT idAdvertisement, title, description, organic, valid,
creationDate, email FROM advertisements WHERE valid = 0 ORDER BY creationDate
DESC';
59         $statement = Database::prepare($req);
60
61         try {
62             $statement->execute();
63         } catch (PDOException $e) {
64             return false;
65         }
66
67         //Tant qu'il y a des entrées
68         while ($row = $statement->fetch(PDO::FETCH_ASSOC)) {
69             //On crée une variable de la classe Advertisement
70             $a = new Advertisement($row['idAdvertisement'], $row['title'], $row
['description'], $row['organic'], $row['valid'], $row['creationDate'], $row['
email']);
71             //Et on l'ajoute dans le tableau qu'on retourne à la fin
72             array_push($arr, $a);
73         }
74
75         return $arr;
76     }
77
78     /**
79      * @brief Retourne l'annonce que l'on recherche par son identifiant
80      * @param int idAd L'id de l'annonce que l'on recherche.
81      * @return [Advertisement] L'annonce de type "Advertisement".
82      *         False Une erreur est survenue.
83      *         Null L'annonce n'est pas trouvée.
84      */
85     public static function GetAdById($idAd)
86     {
87         $a = null;
88
89         //Initialisation de la requête
90         $req = 'SELECT idAdvertisement, title, description, organic, valid,
creationDate, email FROM advertisements WHERE idAdvertisement = :idAd';
91         $statement = Database::prepare($req);
92
93         try {
94             $statement->execute(array(':idAd' => $idAd));
95         } catch (PDOException $e) {
96             return false;
97         }
98
99         //On récupère le premier élément
100         if ($row = $statement->fetch(PDO::FETCH_ASSOC, PDO::FETCH_ORI_NEXT)) {
101             $a = new Advertisement($row['idAdvertisement'], $row['title'], $row
['description'], $row['organic'], $row['valid'], $row['creationDate'], $row['
email']);
102         }
103
104         return $a;
105     }
106

```



```

107  /**
108   * @brief Retourne l'annonce que l'on recherche par l'identifiant de son
    annonceur
109   * @param string email L'email de l'annonceur.
110   * @return array Un tableau contenant toutes les annonces de type "
    Advertisement".
111   *          False Une erreur est survenue.
112   */
113   public static function GetAdsByUserEmail($email)
114   {
115       $arr = array();
116
117       //Initialisation de la requête
118       $req = 'SELECT idAdvertisement, title, description, organic, valid,
    creationDate, email FROM advertisements WHERE email = :e';
119       $statement = Database::prepare($req);
120
121       try {
122           $statement->execute(array(':e' => $email));
123       } catch (PDOException $e) {
124           return false;
125       }
126
127       // Tant qu'il y a des entrées
128       while ($row = $statement->fetch(PDO::FETCH_ASSOC, PDO::FETCH_ORI_NEXT)
    ) {
129           $a = new Advertisement($row['idAdvertisement'], $row['title'], $row
    ['description'], $row['organic'], $row['valid'], $row['creationDate'], $row['
    email']);
130
131           array_push($arr, $a);
132       }
133
134       return $arr;
135   }
136
137  /**
138   * @brief Ajout d'une annonce
139   * @param [Advertisement] L'objet "Advertisement" qu'on veut insérer dans
    la base.
140   * @return boolean True Ok.
141   *          False Une erreur est survenue.
142   */
143   public static function CreateAd($a)
144   {
145       //Initialisation de la requête
146       $req = 'INSERT INTO advertisements (title, description, organic, valid
    , email) VALUES (:t, :d, :o, :v, :e)';
147       $statement = Database::prepare($req);
148       Database::beginTransaction();
149       try {
150           if ($statement->execute(array(
151               ':t' => $a->title,
152               ':d' => $a->description,
153               ':o' => $a->organic,
154               ':v' => INVALID,
155               ':e' => $a->userEmail
156           ))) {
157               //Nécessaire pour les tests
158               if ($_FILES != array()) {

```

```

159         //Vérification qu'il y ait bien des éléments dans la
    tableau
160         if ($_FILES['pictures']['error'][0] != 4) {
161             if (PictureManager::CreatePicture(Database::
lastInsertId()) == false) {
162                 Database::rollBack();
163                 return false;
164             }
165         }
166     }
167     Database::commit();
168     return true;
169 }
170 } catch (PDOException $e) {
171     Database::rollBack();
172 }
173
174     return false;
175 }
176
177 /**
178  * @brief Modification d'une annonce
179  * @param [Advertisement] L'objet "Advertisement" qu'on veut modifier dans
la base.
180  * @return boolean True Ok.
181  *          False Une erreur est survenue.
182  */
183     public static function UpdateAd($a)
184     {
185         //Initialisation de la requête
186         $req = 'UPDATE advertisements SET title = :t, description = :d,
organic = :o WHERE idAdvertisement = :id';
187         $statement = Database::prepare($req);
188         Database::beginTransaction();
189         try {
190             if ($statement->execute(array(
191                 ':t' => $a->title,
192                 ':d' => $a->description,
193                 ':o' => $a->organic,
194                 ':id' => $a->idAdvertisement
195             ))) {
196                 if ($_FILES != array()) {
197                     // Si des fichiers ont été télécharger
198                     if ($_FILES['pictures']['error'][0] != 4) {
199                         if (PictureManager::CreatePicture($a->idAdvertisement)
== false) {
200                             Database::rollBack();
201                             return false;
202                         }
203                     }
204                 }
205                 Database::commit();
206                 return true;
207             }
208         } catch (PDOException $e) {
209             Database::rollBack();
210         }
211         return false;
212     }
213 }

```

```

214  /**
215   * @brief Valide une annonce
216   * @param int idAd L'id de l'annonce que l'on veut validée.
217   * @return boolean True Ok.
218   *          False Une erreur est survenue.
219   */
220  public static function UpdateAdToValid($idAd)
221  {
222      //Initialisation de la requête
223      $req = 'UPDATE advertisements SET valid = :v WHERE idAdvertisement = :
idAd';
224      $statement = Database::prepare($req);
225
226      try {
227          $statement->execute(array(
228              ':v' => VALID,
229              ':idAd' => $idAd
230          ));
231      } catch (PDOException $e) {
232          return false;
233      }
234
235      return true;
236  }
237
238  /**
239   * @brief Suppression d'une annonce
240   * @param int idAd L'id de l'annonce qu'on veut supprimer.
241   * @return boolean True Ok.
242   *          False Une erreur est survenue.
243   */
244  public static function DeleteAd($idAd)
245  {
246      //Initialisation de la requête
247      $req = 'DELETE FROM advertisements WHERE idAdvertisement = :idAd';
248      $statement = Database::prepare($req);
249
250      // Suppression des images liées avec cette annonce
251      if (PictureManager::DeletePicturesOfAnAd($idAd) == false) {
252          return false;
253      }
254
255      // Suppression des évaluations liées avec cette annonce
256      if (RatingManager::DeleteRatingsOfAnAd($idAd) == false) {
257          return false;
258      }
259
260      try {
261          $statement->execute(array(':idAd' => $idAd));
262      } catch (PDOException $e) {
263          return false;
264      }
265
266      return true;
267  }
268
269  /**
270   * @brief Suppression de toutes les annonces d'un utilisateur
271   * @param string email L'email de l'utilisateur.
272   * @return boolean True Ok.

```

```

273      *                               False Une erreur est survenue.
274      */
275      public static function DeleteAdsOfUser($email)
276      {
277          $ads = AdvertisementManager::GetAdsByUserEmail($email);
278
279          foreach ($ads as $ad) {
280              if (AdvertisementManager::DeleteAd($ad->idAdvertisement) == false)
281              {
282                  return false;
283              }
284          }
285          return true;
286      }
287
288
289      /**
290       * @brief Retourne les annonces en rapport avec la recherche
291       * @param string searchContent La recherche de l'utilisateur.
292       * @param boolean organic Si l'utilisateur recherche que des annonces bio.
293       * @param string searchOption L'option de recherche.
294       * @return array Un tableau contenant toutes les annonces de type "
295       Advertisement".
296       *                               False Une erreur est survenue.
297       */
298      public static function Research($searchContent, $organic, $searchOption)
299      {
300          $arr = array();
301
302          $baseReq = "SELECT DISTINCT a.idAdvertisement, a.title, a.description
303          , a.organic, a.valid, a.creationDate, a.email FROM advertisements AS a, users
304          AS u WHERE a.valid = 1 AND a.email = u.email ";
305
306          $withOrganic = "AND a.organic = 1 ";
307
308          switch ($searchOption) {
309              case 'city':
310                  $withContent = "AND u.city LIKE :sc ";
311                  $searchContent = "%$searchContent%";
312                  break;
313              case 'canton':
314                  $withContent = "AND u.canton LIKE :sc ";
315                  $searchContent = "%$searchContent%";
316                  break;
317              case 'postCode':
318                  $withContent = "AND u.postCode LIKE :sc ";
319                  $searchContent = "%$searchContent%";
320                  break;
321              case 'title':
322                  $withContent = "AND a.title LIKE :sc ";
323                  $searchContent = "%$searchContent%";
324                  break;
325              case 'description':
326                  $withContent = "AND a.description LIKE :sc ";
327                  $searchContent = "%$searchContent%";
328                  break;
329              case 'score':
330                  $withContent = "AND :sc = (SELECT FORMAT(AVG(r.rating), 'N')
331          FROM rates AS r WHERE r.idAdvertisement = a.idAdvertisement)";

```

```
328         break;
329     }
330
331     if ($searchContent != "" && $organic) {
332         $finalReq = $baseReq . $withContent . $withOrganic;
333     } else if ($searchContent != "" && !$organic) {
334         $finalReq = $baseReq . $withContent;
335     } else if ($searchContent == "" && $organic) {
336         $finalReq = $baseReq . $withOrganic;
337     } else {
338         return false;
339     }
340
341     $statement = Database::prepare($finalReq);
342
343     try {
344         $statement->execute(array(
345             ':sc' => $searchContent
346         ));
347     } catch (PDOException $e) {
348         return false;
349     }
350
351     // Tant qu'il y a des entrées
352     while ($row = $statement->fetch(PDO::FETCH_ASSOC)) {
353         $a = new Advertisement($row['idAdvertisement'], $row['title'], $row
['description'], $row['organic'], $row['valid'], $row['creationDate'], $row['
email']);
354
355         array_push($arr, $a);
356     }
357
358     return $arr;
359 }
360 }
361
```

```
1 <?php
2 /*
3  Titre : Constantes du site
4  Date : Mardi, 21 mai 2019
5  Auteur : Romain Peretti
6  Version : 1.0
7  Description : Cette page contient toutes les constantes
8  */
9
10 define('MAX_STARS_RATING', 5);
11
12 define('USER_CODE', 1);
13 define('ADMINISTRATOR_CODE', 2);
14
15 define('VALID', 1);
16 define('INVALID', 0);
17
18 define('ORGANIC', 1);
19 define('NOT_ORGANIC', 0);
20
21 define('MAX_FILE_SIZE', 3000000);
22
23 define('EXTENSION_AUTHORIZED', array('image/gif', 'image/png', 'image/jpg', '
    image/pjpeg', 'image/bmp', 'image/jfif', 'image/jpeg', 'image/pjp'));
24
25 define('ERROR_MAX_FILE_SIZE', 2);
26 define('NO_DOWNLOADED_PICTURE', 4);
27
28 define('OK', 0);
29 define('PWD_NOT_VALID', 1);
30 define('NEW_PWD_NOT_EQUAL', 2);
```

```
1 <?php
2 /*
3 Titre : Contrôleur de l'accueil
4 Date : Lundi, 13 mai 2019
5 Auteur : Romain Peretti
6 Version : 1.0
7 Description : Cette page inclut la recherche d'annonces et la vue
8 */
9
10 require_once '../inc/inc.all.php';
11
12 $ads = AdvertisementManager::GetValidatedAds();
13
14 if (isset($_POST['search'])) {
15     $searchContent = filter_input(INPUT_POST, 'searchContent',
16     FILTER_SANITIZE_STRING);
17     $searchOrganic = filter_input(INPUT_POST, 'searchOrganic',
18     FILTER_SANITIZE_STRING);
19     $searchOption = filter_input(INPUT_POST, 'searchOption',
20     FILTER_SANITIZE_STRING);
21
22     if ($searchContent != "" || isset($searchOrganic)) {
23         $results = AdvertisementManager::Research($searchContent, (isset(
24         $searchOrganic) ? true : false), $searchOption);
25         if (isset($results)) {
26             $ads = $results;
27         } else if ($results === false) {
28             echo '<div class="alert alert-danger mb-0" role="alert">Une erreur
29             est survenue lors de la recherche</div>';
30         } else {
31             echo '<div class="alert alert-warning mb-0" role="alert">Veuillez
32             entrer une recherche</div>';
33         }
34     }
35 }
36
37 $pageName = 'Accueil';
38
39 include_once '../views/showHome.php';
40
```

```
1 <?php
2 /*
3 Titre : Contrôleur de l'administrateur
4 Date : Mardi, 14 mai 2019 / modifier le jeudi, 22 mai 2019
5 Auteur : Romain Peretti
6 Version : 2.0
7 Description : Cette page inclut la vue.
8 */
9
10 require_once '../inc/inc.all.php';
11
12 if (!isset($_SESSION['email']) || $_SESSION['role'] != ADMINISTRATOR_CODE) {
13     header('Location: home.php');
14 }
15
16 $idAdToValid = filter_input(INPUT_GET, 'idAd', FILTER_SANITIZE_NUMBER_INT);
17
18 $ads = AdvertisementManager::GetInvalidAds();
19 $adsNumber = 0;
20
21 $users = UserManager::GetUsers();
22 $usersNumber = 0;
23
24 if (isset($idAdToValid)) {
25     if (AdvertisementManager::UpdateAdToValid($idAdToValid)) {
26         echo '<div class="alert alert-success mb-0" role="alert">Annonce
validée</div>';
27         echo '<meta http-equiv="refresh" content="1;URL=admin.php">';
28     }
29 }
30
31 if (isset($_POST['delete'])) {
32     $idAdToDelete = filter_input(INPUT_POST, 'idAd', FILTER_SANITIZE_NUMBER_INT
);
33
34     if (AdvertisementManager::DeleteAd($idAdToDelete)) {
35         echo '<div class="alert alert-success mb-0" role="alert">Annonce
supprimée</div>';
36         echo '<meta http-equiv="refresh" content="1;URL=admin.php">';
37     } else {
38         echo '<div class="alert alert-success mb-0" role="alert">Erreur lors de
la suppression</div>';
39     }
40 }
41
42 if (isset($_POST['updateToAdmin'])) {
43     $userEmail = filter_input(INPUT_POST, 'userEmail', FILTER_SANITIZE_STRING);
44
45     if (UserManager::UpdateUserToAdmin($userEmail)) {
46         echo '<div class="alert alert-success mb-0" role="alert">Utilisateur
mit à jour</div>';
47         echo '<meta http-equiv="refresh" content="1;URL=admin.php">';
48     } else {
49         echo '<div class="alert alert-success mb-0" role="alert">Erreur lors de
la modification</div>';
50     }
51 }
52
53 if (isset($_POST['updateToUser'])) {
54     $userEmail = filter_input(INPUT_POST, 'userEmail', FILTER_SANITIZE_STRING);
```



```
55
56     if (UserManager::UpdateAdminToUser($userEmail)) {
57         if ($userEmail == $_SESSION['email']) {
58             $_SESSION['role'] = USER_CODE;
59         }
60         echo '<div class="alert alert-success mb-0" role="alert">Utilisateur
mit à jour</div>';
61         echo '<meta http-equiv="refresh" content="1;URL=admin.php">';
62     } else {
63         echo '<div class="alert alert-success mb-0" role="alert">Erreur lors
de la modification</div>';
64     }
65 }
66
67 $pageName = 'Administrateur';
68
69 include_once '../views/showAdmin.php';
70
```

```
1 <?php
2 /*
3  Titre : Controlleur de la connexion
4  Date : Mardi, 14 mai 2019
5  Auteur : Romain Peretti
6  Version : 1.0
7  Description : Cette page inclut la vue
8  */
9
10 require_once '../inc/inc.all.php';
11
12 $pageName = 'Connexion';
13
14 include_once '../views/showLogin.php';
```

```
1 <?php
2 /*
3 Auteur : php.net
4 Lien : https://www.php.net/manual/fr/function.session-destroy.php
5 Date : Mardi, 14 mai 2019
6 */
7
8 // Initialisation de la session.
9 // Si vous utilisez un autre nom
10 // session_name("autrenom")
11 session_start();
12
13 // D truit toutes les variables de session
14 $_SESSION = array();
15
16 // Si vous voulez d truire compl tement la session, effacez  galement
17 // le cookie de session.
18 // Note : cela d truirait la session et pas seulement les donn es de session !
19 if (ini_get("session.use_cookies")) {
20     $params = session_get_cookie_params();
21     setcookie(session_name(), '', time() - 42000,
22         $params["path"], $params["domain"],
23         $params["secure"], $params["httponly"]
24     );
25 }
26
27 // Finalement, on d truit la session.
28 session_destroy();
29
30 header("Location: home.php");
```

```
1 <?php
2 /*
3  Titre : Contrôleur du profil
4  Date : Mardi, 14 mai 2019
5  Auteur : Romain Peretti
6  Version : 1.0
7  Description : Cette page inclut les détails de l'utilisateur et ses annonces
8  */
9
10 require_once '../inc/inc.all.php';
11
12 $email = filter_input(INPUT_GET, 'email', FILTER_SANITIZE_STRING);
13
14 if (isset($email) && $email != $_SESSION['email']) {
15     header('Location: home.php');
16 }
17
18 $u = UserManager::getUserByEmail($email);
19 $ads = AdvertisementManager::getAdsByUserEmail($email);
20
21 $pageName = 'Détails de votre compte';
22
23 include_once '../views/showProfil.php';
24
```

```
1 <?php
2 /*
3  Titre : Contrôleur de l'inscription
4  Date : Lundi, 13 mai 2019
5  Auteur : Romain Peretti
6  Version : 1.0
7  Description : Cette page inclut la création d'une annonce avec le formulaire et
   la vue
8 */
9
10 require_once '../inc/inc.all.php';
11
12 if (isset($_POST['send'])) {
13
14     $password = filter_input(INPUT_POST, 'password', FILTER_SANITIZE_STRING);
15     $repeatPassword = filter_input(INPUT_POST, 'repeatPassword',
16     FILTER_SANITIZE_STRING);
17     if ($password == $repeatPassword) {
18
19         $email = filter_input(INPUT_POST, 'email', FILTER_SANITIZE_EMAIL);
20         if (filter_var($email, FILTER_VALIDATE_EMAIL)) {
21             $canton = filter_input(INPUT_POST, 'canton', FILTER_SANITIZE_STRING
22 );
23             $city = filter_input(INPUT_POST, 'city', FILTER_SANITIZE_STRING);
24             $postCode = filter_input(INPUT_POST, 'postCode',
25     FILTER_SANITIZE_STRING);
26             $streetAndNumber = filter_input(INPUT_POST, 'streetAndNumber',
27     FILTER_SANITIZE_STRING);
28             $description = filter_input(INPUT_POST, 'description',
29     FILTER_SANITIZE_STRING);
30
31             $u = new User($email, $password, $city, $canton, $postCode,
32     $streetAndNumber, $description, null, null);
33             $result = UserManager::CreateUser($u);
34
35             if ($result === false) {
36                 echo '<div class="alert alert-danger mb-0" role="alert">Message
37 d\'erreur : "Erreur lors de la création"</div>';
38             }
39             if ($result === null) {
40                 echo '<div class="alert alert-danger mb-0" role="alert">Cet
41 email est déjà utilisé</div>';
42             }
43             if ($result) {
44                 echo '<div class="alert alert-success mb-0" role="alert">Compte
45 créé, en attente de redirection</div>';
46                 echo '<meta http-equiv="refresh" content="1;URL=login.php">';
47             }
48             } else {
49                 echo '<div class="alert alert-danger mb-0" role="alert">Veuillez
50 rentrer un email valide</div>';
51             }
52             } else {
53                 echo '<div class="alert alert-danger mb-0" role="alert">Les mots de
54 passe ne sont pas pareil</div>';
55             }
56         }
57     }
58
59     $pageName = 'Inscription';
60 }
```

```
49 include_once '../views/showSignIn.php';  
50
```

```
1 <?php
2 /*
3 Titre : Contrôleur de création d'une annonce
4 Date : Mardi, 15 mai 2019
5 Auteur : Romain Peretti
6 Version : 1.0
7 Description : Cette page inclut la vue.
8 */
9
10 require_once '../inc/inc.all.php';
11
12 if (isset($_POST['send'])) {
13     for ($i = 1; $i <= count($_FILES['pictures']['name']); $i++) {
14
15         $index = $i - 1;
16
17         // Mauvaise extension
18         if (in_array($_FILES['pictures']['type'][$index], EXTENSION_AUTHORIZED)
19             || $_FILES['pictures']['error'][$index] == NO_DOWNLOADED_PICTURE) {
20
21             //Fichier trop gros
22             if ($_FILES['pictures']['size'][$index] < MAX_FILE_SIZE) {
23
24                 // Si toutes les images ont été parcouru
25                 if ($i == count($_FILES['pictures']['name'])) {
26
27                     // Récupération des champs
28                     $title = filter_input(INPUT_POST, 'title',
29 FILTER_SANITIZE_STRING);
30                     $description = filter_input(INPUT_POST, 'description',
31 FILTER_SANITIZE_STRING);
32                     $organic = filter_input(INPUT_POST, 'organic',
33 FILTER_SANITIZE_STRING);
34
35                     // Création de l'annonce
36                     $a = new Advertisement(null, $title, $description, (isset(
37 $organic) ? ORGANIC : NOT_ORGANIC), null, null, $_SESSION['email']);
38                     $result = AdvertisementManager::CreateAd($a);
39
40                     // Affichage du résultat
41                     if ($result == false) {
42                         echo '<div class="alert alert-danger mb-0" role="alert
43 ">Message d\'erreur : "Erreur lors de la création"</div>';
44                     }
45                     if ($result) {
46                         echo '<div class="alert alert-success mb-0" role="alert
47 ">Création d\'une annonce réussi, en attente de redirection</div>';
48                         echo '<meta http-equiv="refresh" content="2;URL=home.
49 php">';
50                     }
51                 }
52             } else {
53                 echo '<div class="alert alert-warning mb-0" role="alert">
54 Fichier proposé (' . $_FILES['pictures']['name'][$index] . ') trop volumineux.
55 Taille max 3 Mo</div>';
56             }
57         } else {
58             echo '<div class="alert alert-warning mb-0" role="alert">Mauvaise
59 extension de fichier pour (' . $_FILES['pictures']['name'][$index] . ')</div>';
60         }
61     }
62 }
```

```
50     }
51 }
52
53 $pageName = 'Création d\'une annonce';
54
55 include_once '../views/showCreateAd.php';
56
```



```
1 <?php
2 /*
3 Titre : Contrôleur de la suppression d'une annonce
4 Date : Mardi, 14 mai 2019
5 Auteur : Romain Peretti
6 Version : 1.0
7 Description : Cette page inclut la recherche d'annonces et la vue
8 */
9
10 require_once '../inc/inc.all.php';
11
12 $idAd = filter_input(INPUT_GET, 'idAd', FILTER_SANITIZE_NUMBER_INT);
13
14 $a = AdvertisementManager::GetAdById($idAd);
15
16 if ($_SESSION['email'] != $a->userEmail) {
17     header('Location: home.php');
18 }
19
20 if (isset($_POST['send'])) {
21
22     $result = AdvertisementManager::DeleteAd($idAd);
23     if ($result === false) {
24         echo '<div class="alert alert-danger mb-0" role="alert">Message d\'
erreur : "Erreur lors de la suppression"</div>';
25     }
26     if ($result) {
27         echo '<div class="alert alert-success mb-0" role="alert">Suppression
réussi, en attente de redirection</div>';
28         echo '<meta http-equiv="refresh" content="1;URL=home.php">';
29     }
30 }
31
32 $pageName = 'Suppression d\'une annonce';
33
34 include_once '../views/showDeleteAd.php';
35
```

```
1 <?php
2 /*
3 Titre : Contrôleur de la modification d'une annonce
4 Date : Mardi, 14 mai 2019
5 Auteur : Romain Peretti
6 Version : 1.0
7 Description : Cette page inclut la vue.
8 */
9
10 require_once '../inc/inc.all.php';
11
12 $idAd = null;
13
14 if (isset($_GET['idAd'])) {
15     $idAd = filter_input(INPUT_GET, 'idAd', FILTER_SANITIZE_NUMBER_INT);
16 }
17
18 $ad = AdvertisementManager::GetAdById($idAd);
19 $pictures = PictureManager::GetPicturesByAdId($idAd);
20
21 if ($_SESSION['email'] != $ad->userEmail) {
22     header('Location: home.php');
23 }
24
25 if (isset($_POST['send'])) {
26     for ($i = 1; $i <= count($_FILES['pictures']['name']); $i++) {
27
28         $index = $i - 1;
29
30         // Mauvaise extension
31         if (in_array($_FILES['pictures']['type'][$index], EXTENSION_AUTHORIZED)
32 || $_FILES['pictures']['error'][$index] == NO_DOWNLOADED_PICTURE) {
33
34             //Fichier trop gros
35             if ($_FILES['pictures']['size'][$index] < MAX_FILE_SIZE) {
36
37                 // Si toutes les images ont été parcouru
38                 if ($i == count($_FILES['pictures']['name'])) {
39
40                     // Récupération des champs
41                     $title = filter_input(INPUT_POST, 'title',
42 FILTER_SANITIZE_STRING);
43                     $description = filter_input(INPUT_POST, 'description',
44 FILTER_SANITIZE_STRING);
45                     $organic = filter_input(INPUT_POST, 'organic');
46
47                     // Création de l'annonce
48                     $a = new Advertisement($ad->idAdvertisement, $title,
49 $description, (isset($organic) ? ORGANIC : NOT_ORGANIC), null, null, $ad->
50 userEmail);
51                     $result = AdvertisementManager::UpdateAd($a);
52
53                     // Affichage du résultat
54                     if ($result === false) {
55                         echo '<div class="alert alert-danger mb-0" role="alert
56 ">Message d\'erreur : "Erreur lors de la modification"</div>';
57                     }
58                     if ($result) {
59                         echo '<div class="alert alert-success mb-0" role="alert
60 ">Modification d\'une annonce réussi, en attente de redirection</div>';
61                     }
62                 }
63             }
64         }
65     }
66 }
```

```
54         echo '<meta http-equiv="refresh" content="1;URL=home.
php">';
55     }
56 }
57 } else {
58     echo '<div class="alert alert-warning mb-0" role="alert">
Fichier proposé (' . $_FILES['pictures']['name'][$index] . ') trop volumineux.
Taille max 3 Mo</div>';
59 }
60 } else {
61     echo '<div class="alert alert-warning mb-0" role="alert">Mauvaise
extension de fichier pour (' . $_FILES['pictures']['name'][$index] . ')</div>';
62 }
63 }
64 }
65 if (isset($_POST['delete'])) {
66     if (isset($_POST['idPic'])) {
67
68         $idPic = filter_input(INPUT_POST, 'idPic', FILTER_SANITIZE_NUMBER_INT);
69
70         PictureManager::DeletePicture($idPic);
71
72         header('Location: updateAd.php?idAd=' . $idAd);
73     }
74 }
75
76 $pageName = 'Modification d\'une annonce';
77
78 include_once '../views/showUpdateAd.php';
79
```

```
1 <?php
2 /*
3 Titre : Contrôleur des détails d'une annonce
4 Date : Lundi, 13 mai 2019
5 Auteur : Romain Peretti
6 Version : 1.0
7 Description : Cette page inclut la vue.
8 */
9
10 require_once '../inc/inc.all.php';
11
12 $idAd = filter_input(INPUT_GET, 'idAd', FILTER_SANITIZE_STRING);
13
14 $ad = AdvertisementManager::GetAdById($idAd);
15 $u = UserManager::GetUserByEmail($ad->userEmail);
16 $rates = RatingManager::GetRatingsOfAnAd($idAd);
17 $score = RatingManager::GetScoreOfAnAd($idAd);
18 $pics = PictureManager::GetPicturesByAdId($idAd);
19
20 if (isset($_POST['send'])) {
21     $comment = filter_input(INPUT_POST, 'comment', FILTER_SANITIZE_STRING);
22     $rating = filter_input(INPUT_POST, 'grade', FILTER_SANITIZE_STRING);
23
24     $r = new Rating(null, $rating, $comment, null, $_SESSION['email'], $ad->
        idAdvertisement);
25     $result = RatingManager::CreateRating($r);
26
27     if ($result === false) {
28         echo '<div class="alert alert-danger mb-0" role="alert">Message d\'
        erreur : "Erreur lors de la création"</div>';
29     }
30     if ($result) {
31         echo '<div class="alert alert-success mb-0" role="alert">Création du
        commentaire réussi</div>';
32         echo '<meta http-equiv="refresh" content="2;URL=#">';
33     }
34 }
35
36 $pageName = 'Détails d\'une annonce';
37
38 include_once '../views/showAdDetails.php';
```

```
1 <?php
2 /*
3 Titre : Contrôleur de la suppression du profil
4 Date : Mardi, 14 mai 2019
5 Auteur : Romain Peretti
6 Version : 1.0
7 Description : Cette page inclut la vue.
8 */
9
10 require_once '../inc/inc.all.php';
11
12 $email = filter_input(INPUT_GET, 'email', FILTER_SANITIZE_STRING);
13
14 $u = UserManager::getUserByEmail($email);
15
16 if ($_SESSION['email'] != $email) {
17     header('Location: home.php');
18 }
19
20 if (isset($_POST['send'])) {
21
22     $result = UserManager::DeleteUser($email);
23     if ($result === false) {
24         echo '<div class="alert alert-danger mb-0" role="alert">Message d\'
erreur : "Erreur lors de la suppression"</div>';
25     }
26     if ($result) {
27         echo '<div class="alert alert-success mb-0" role="alert">Suppression
réussi, en attente de redirection</div>';
28         echo '<meta http-equiv="refresh" content="1;URL=logout.php">';
29     }
30 }
31
32 $pageName = 'Suppression de votre compte';
33
34 include_once '../views/showDeleteProfil.php';
35
```

```
1 <?php
2 /*
3  Titre : Contrôleur de la modification du profil
4  Date : Mardi, 14 mai 2019
5  Auteur : Romain Peretti
6  Version : 1.0
7  Description : Cette page inclut la vue.
8  */
9
10 require_once '../inc/inc.all.php';
11
12 $email = filter_input(INPUT_GET, 'email', FILTER_SANITIZE_STRING);
13
14 $u = UserManager::getUserByEmail($email);
15
16 if ($_SESSION['email'] != $email) {
17     header('Location: home.php');
18 }
19
20 if (isset($_POST['send'])) {
21
22     $state;
23     $finalPwd;
24
25     $pwd = filter_input(INPUT_POST, 'password', FILTER_SANITIZE_STRING);
26
27     if (sha1($pwd . $u->salt) == $u->password) {
28         $newPwd = filter_input(INPUT_POST, 'newPassword',
29 FILTER_SANITIZE_STRING);
30         $repeatNewPwd = filter_input(INPUT_POST, 'repeatNewPassword',
31 FILTER_SANITIZE_STRING);
32
33         if ($newPwd != "" && $repeatNewPwd != "") {
34             if ($newPwd == $repeatNewPwd) {
35                 // Changement du MDP
36                 $finalPwd = $newPwd;
37                 $state = OK;
38             } else {
39                 // Erreur répétition du nouveau MDP
40                 $state = NEW_PWD_NOT_EQUAL;
41             }
42         } else {
43             // Pas de changement du MDP
44             $finalPwd = $pwd;
45             $state = OK;
46         }
47     } else {
48         // Erreur de MDP
49         $state = PWD_NOT_VALID;
50     }
51
52     switch ($state) {
53         case OK:
54             $canton = filter_input(INPUT_POST, 'canton',
55 FILTER_SANITIZE_STRING);
56             $city = filter_input(INPUT_POST, 'city', FILTER_SANITIZE_STRING);
57             $postCode = filter_input(INPUT_POST, 'postCode',
58 FILTER_SANITIZE_STRING);
59             $streetAndNumber = filter_input(INPUT_POST, 'streetAndNumber',
60 FILTER_SANITIZE_STRING);
```

```
56     $description = filter_input(INPUT_POST, 'description',  
    FILTER_SANITIZE_STRING);  
57  
58     $u = new User($_SESSION['email'], $finalPwd, $city, $canton,  
    $postCode, $streetAndNumber, $description, null, null);  
59  
60     if (UserManager::UpdateUser($u)) {  
61         echo '<div class="alert alert-success mb-0" role="alert">  
Modification effectuée, en attente de redirection</div>';  
62         echo '<meta http-equiv="refresh" content="1;URL=profil.php?  
email=' . $_SESSION['email'] . '">';  
63     } else {  
64         echo '<div class="alert alert-danger mb-0" role="alert">  
Message d\'erreur : "Erreur lors de la modification"</div>';  
65     }  
66     break;  
67     case PWD_NOT_VALID:  
68         echo '<div class="alert alert-danger mb-0" role="alert">Message d  
' . 'erreur : "Le mot de passe actuel est faux"</div>';  
69         break;  
70     case NEW_PWD_NOT_EQUAL:  
71         echo '<div class="alert alert-danger mb-0" role="alert">Message d  
' . 'erreur : "Les nouveaux mot de passe ne sont pas les mêmes"</div>';  
72         break;  
73     }  
74 }  
75  
76 $pageName = 'Modification de votre profil';  
77  
78 include_once '../views/showUpdateProfil.php';  
79
```