

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <?php
19 //Démarrage de la session
20 session_start();
21 ini_set('display_errors', 1);
22 if (!isset($_SESSION['connectedUser'])) {
23     $_SESSION['connectedUser'] = [
24         'isConnected' => false,
25         'idUser' => null,
26         'email' => null,
27         'isAdmin' => null
28     ];
29     $_SESSION['idEditPost'] = null;
30
31     $_SESSION['AlertMessage'] = [
32         'type' => null,
33         'message' => null
34     ];
35 }
36
37 //Include des fichiers nécessaires
38 require("models/db_connect.php");
39 require("models/Production.php");
40 require("models/User.php");
41 require("models/Categorie.php");
42 require("models/like_unlike.php");
43
44 $uc = filter_input(INPUT_GET, 'uc') == null ? "home" : filter_input(INPUT_GET, 'uc'); //Page à
d'accueil par défaut
45
46 if (!isset($_SESSION['orderProduction'])) {
47     $_SESSION['orderProduction'] = "date_soumission";
48 }
49
50 if (!isset($_SESSION['filterCategorie'])) {
51     $_SESSION['filterCategorie'] = 0;
52 }
53
54 //Visuel du header
55 if ($uc != "getAllProductions") {
56     include 'vue/header.php';
57 }
58
59 // Gestion des affichages
60 switch ($uc) {
61     // Affichage de la page d'accueil
```

```
62     case 'home':
63         $production = Production::getAllProductions();
64         include "vue/home.php"; //Visuel de la page d'accueil
65         break;
66     case 'login':
67         include 'controllers/login_controller.php';
68         break;
69     case 'production':
70         include 'controllers/production_controller.php';
71         break;
72     case 'trier':
73         include 'controllers/tri_controller.php';
74         break;
75     case 'admin':
76         include 'controllers/admin_controller.php';
77         break;
78
79     case 'getAllPosts':
80         echo Production::CountAllProduction();
81         break;
82 }
83
84 //Visuel du footer
85 if ($uc != "getAllPosts") {
86     include 'vue/footer.php';
87 }
88 error_reporting(E_ALL);
89
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <div class="container" style="display: block; margin-top: 3vh; position: relative;">
19     <?php
20
21     //Si l'utilisateur accède à cette page par erreur ou bien volontairement, il est
22     //instantanément redirigé vers l'accueil (index.php)
23     if ($_SESSION['connectedUser']['isAdmin'] == 2) {
24         } else {
25             //header('Location: index.php');
26             echo '<script>window.location.href = "index.php";</script>';
27         }
28
29         if ($_SESSION['AlertMessage']['type'] != null) { ?>
30             <div class="alert alert-<?= $_SESSION['AlertMessage']['type'] ?> alert-dismissible
31                 show" role="alert">
32                 <?= $_SESSION['AlertMessage']['message'] ?>
33                 <button type="button" class="close" data-dismiss="alert" aria-label="Close">
34                     <span aria-hidden="true">&times;</span>
35                 </button>
36             </div>
37         <?php
38             $_SESSION['AlertMessage'] = [
39                 'type' => null,
40                 'message' => null
41             ];
42         ?>
43         <div class="panel panel-default" style="width: 100%;">
44             <div class="panel-heading">
45                 <h2>Administration des catégories</h2>
46             </div>
47             <div class="panel-body">
48                 <p>Vous pouvez ici gérer les catégories disponibles. Ajoutez/Modifiez/Supprimez
49                     une catégorie en 1 clique.</p>
50             </div>
51             <div class="panel-heading">
52                 <form method="POST" action="index.php?uc=admin&action=AddCategorie">
53                     <h2>Ajouter une catégorie : </h2>
54                     <input type="text" name="nomCategorie" placeholder="Entrez la catégorie"
55                         required>
56                     <input type="submit" class="btn btn-info" value="Ajouter">
57                 </form>
58             </div>
59             <div class="row">
```

```
59         <table class="table">
60             <thead>
61                 <tr>
62                     <th scope="col">idCategorie</th>
63                     <th scope="col">Nom</th>
64                     <th scope="col">Modifier</th>
65                     <th scope="col">Supprimer</th>
66
67                 </tr>
68             </thead>
69             <tbody>
70                 <?php
71                     foreach (Categorie:: GetAllCategories() as $categorie) {
72                         ?
73                         <tr>
74                             <td style="padding-left: 40px;"><?= $categorie->getIdCategorie() ?></?
75                             td>
76                             <td value="<?= $categorie->getIdCategorie() ?>"><?= $categorie-
77                             >getNom() ?></td>
78                             <td style="padding-left: 20px;"><a class="btn btn-success glyphicon
79                             glyphicon-edit" href="index.php?
80                             uc=admin&action>ShowEditCategorie&idCategorie=<?= $categorie-
81                             >getIdCategorie() . '&nameCategorie=' . $categorie-
82                             >GetCategorieNameById($categorie->getIdCategorie()) ?>"> </a></td>
83                             <td style="padding-left: 32px;"><a class="btn btn-danger glyphicon
84                             glyphicon-remove" href="index.php?
85                             uc=admin&action=deleteCategorie&idCategorie=<?= $categorie-
86                             >getIdCategorie() ?>"> </a></td>
87                         </tr>
88                     <?php } ?>
89                 </tbody>
90             </table>
91         </div>
92     </div>
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <div class="container" style="display: block; margin-top: 3vh;">
19
20     <h1>Mot De Passe</h1>
21     <form method='post' action="index.php?uc=login&action=UpdateUserPassword">
22         <div class="form-group">
23             <label>Entrez votre nouveau mot de passe : </label>
24             <input type="password" class="form-control" name ="newPassword" required>
25         </div>
26         <div class="form-group">
27             <label>Confirmez le nouveau mot de passe : </label>
28             <input type="password" class="form-control" name="confirmNewPassword" required>
29         </div>
30         <div class="form-group">
31
32             <a href="index.php?uc=login&action>ShowForgotPassword">Mot de passe oublié ?
33                 Cliquez ici</a><br>&ampnbsp&ampnbsp
34         </div>
35         <button type="submit" class="btn btn-success pull-left" name='updatePassword'>Mettre
36             à jour le mot de passe</button><br><br>
37         <a class="btn btn-default" href="index.php?uc=login&action>ShowProfile">Retour à la
38             page profil</a>&ampnbsp&ampnbsp
39     </form>
40 </div>
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <div class="container">
19     <?php
20         if ($_SESSION['AlertMessage']['type'] != null) { ?>
21             <div class="alert alert-<?= $_SESSION['AlertMessage']['type'] ?> alert-dismissible" style="width: 100%; margin-bottom: 10px;">
22                 <?= $_SESSION['AlertMessage']['message'] ?>
23                 <button type="button" class="close" data-dismiss="alert" aria-label="Close">
24                     <span aria-hidden="true">&times;</span>
25                 </button>
26             </div>
27         <?php
28             $_SESSION['AlertMessage'] = [
29                 'type' => null,
30                 'message' => null
31             ];
32         }
33     ?>
34     <div class="panel panel-default" style="width: 100%;">
35         <div class="panel-heading">
36             <h2>Modification d'une catégorie</h2>
37         </div>
38         <div class="panel-body">
39             <p>Pour modifier la catégorie sélectionnée, il suffit de changer le nom de celle-ci et de valider en cliquant sur le bouton "Valider les changements".</p>
40         </div>
41     </div>
42     <div class="row">
43         <form method="POST" action="index.php?uc=admin&action=UpdateCategorie&idCategorie=<?= $_GET['idCategorie'] ?>">
44             <h2>Modifier une catégorie : </h2>
45             <input type="text" name="nomCategorie" value="<?= $_GET['nameCategorie'] ?>" required>
46             <input type="submit" class="btn btn-info" value="Modifier">
47         </form>
48         <a class="btn btn-default" href="index.php?uc=admin&action>ShowAdminPannel">Retour</a>
49     </div>
50 </div>
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <div class="container" style="display: block; margin-top: 3vh;">
19     <?php
20         if ($_SESSION['message']['type'] != null) { ?>
21             <div class="alert alert-<?= $_SESSION['message']['type'] ?> alert-dismissible" role="alert">
22                 <button type="button" class="close" data-dismiss="alert" aria-label="Close"><span >&times;</span></button>
23                 <?= $_SESSION['message']['content'] ?>
24             </div>
25         <?php
26             $_SESSION['message'] = [
27                 'type' => null,
28                 'content' => null
29             ];
30         }
31     ?>
32     <h1>Modification d'une production</h1>
33
34     <!-- Formulaire de modification d'une production -->
35     <form method="POST" action="index.php?uc=production&action=validateEdit" enctype="multipart/form-data">
36         <div class="form-group">
37             <label for="idTitreProduction">Titre : </label>
38             <textarea class="form-control" id="idTitreProduction" rows="1" name="titreProduction"><?= $post->getTitreProduction() ?></textarea>
39         </div>
40         <div class="form-group">
41             <label for="idDescriptionProduction">Description : </label>
42             <textarea class="form-control" id="idDescriptionProduction" rows="4" name="descriptionProduction"><?= $post->getDescriptionProduction() ?></textarea>
43         </div>
44
45         <div class="form-group">
46             <label>Fichier contenu dans la production : </label>
47             <div style="display: flex; flex-direction: row; flex-wrap: wrap; flex-flow :center; width: 100%">
48                 <?php
49                 foreach (Media::getAllMediasByPostId($idPost) as $media) {
50                     switch (explode("/", $media->getTypeMedia())[0]) {
51                         case 'image':
52                     ?>
53                         <div style="border: 1px solid black; width: 33.3%">
54                             
55                             <a class="btn btn-danger" href="index.php?
```

```
56         uc=production&action=deleteMedia&idMedia=<?= $media->getIdMedia() ? >
57             >">X</a>
58             </div>
59             <?php
60                 break;
61             case 'video':
62             ?>
63             <div style="border: 1px solid black; width: 33.3%" >
64                 <video controls autoplay loop muted width="100%">
65                     <source src=".//assets/medias/<?= $media->getNomFichierMedia() ?>" type="<?= $media->getTypeMedia() ?>">
66                 </video>
67                 <a class="btn btn-danger" href="index.php?
68 uc=production&action=deleteMedia&idMedia=<?= $media->getIdMedia() ? >">X</a>
69             </div>
70             <?php
71                 break;
72             case 'audio':
73             ?>
74                 <div style="border: 1px solid black; width: 33.3%">
75                     <audio controls src=".//assets/medias/<?= $media-
76 >getNomFichierMedia() ?>" style="width: 50%; margin-left: 20%"></
77 audio>
78                     <a class="btn btn-danger" href="index.php?
79 uc=production&action=deleteMedia&idMedia=<?= $media->getIdMedia() ? >">X</a>
80                 </div>
81             <?php
82                 break;
83
84             }
85             echo '<br>';
86         }
87     ?>
88     </div>
89
90     </div>
91     <div class="form-group">
92         <label for="idFile">Changer l'image de la production : </label>
93         <input type="file" class="form-control-file" id="idFile" accept="image"
94             name="filesProduction[]">
95     </div>
96
97     <input class="btn btn-success" type="submit" value="Confirmer la modification">
98     <hr>
99     </form>
100 </div>
101 <?php Media::CountAllMediasPerPost($_SESSION['idEditPost']) ?>
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <div class="container" style="display: block; margin-top: 3vh; position: relative;">
19     <div class="form-group">
20         <div class="col-xs-3">
21             <form method="POST" action="index.php?uc=production&action=UpdateProductionInfos&idProduction=<?= $production->getIdProduction() ?>">
22                 <h4>Titre de la production : </h4>
23                 <input type="text" id="titre" name="titre" value="<?= $production->getTitreProduction() ?>"><br>
24                 <h4>Description : </h4>
25                 <input type="text" id="description" name="description" value="<?= $production->getDescriptionProduction() ?>"><br>
26                 <h4 for="select-categorie">Catégorie : </h4>
27                 <select name="categorieProduction" id="select-categorie" required>
28                     <?php
29                         foreach (Categorie:: GetAllCategories() as $categorie) {
30                             ?
31                             <option value="<?= $categorie->getIdCategorie() ?>" <?= $production->getCategories_idCategorie() == $categorie->getIdCategorie() ? "selected" : "" ?><?= $categorie->getNom() ?></option>
32                     <?php
33                         }
34                         ?
35                     </select><br><br>
36                     <input type="submit" class="btn btn-success pull-left" value="Mettre à jour la production"><br><br>
37                     <a href="index.php?uc=production&action>ShowMyProductions" class="btn btn-default pull-left"><i>Annuler</i></a>
38             </form>
39         </div>
40     </div>
41 </div>
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <div class="container" style="margin-top: 3vh;">
19     <h1>Erreur 404</h1>
20     <p>Veuillez réesayer plus tard.</p>
21 </div>
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 </div>
19 <!-- /main -->
20
21 </div>
22 </div>
23 </div>
24
25 <script type="text/javascript" src="assets/js/jquery.js"></script>
26 <script type="text/javascript" src="assets/js/bootstrap.js"></script>
27 <script type="text/javascript" src="assets/js/main.js"></script>
28 </body>
29
30 </html>
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <div class="container" style="display: block; margin-top: 3vh;">
19
20     <h1>Récupération De Mot De Passe</h1>
21     <div class="form-group">
22         <h4>Votre nouveau mot de passe est le suivant :</h4>
23         <input type="text" class="form-control" value="<?= $password ?>" readonly>
24         <label><i>Vous pouvez le copier-coller pour le modifier sur la page de
25             modification.</i></label>
26     </div>
27     <a class="btn btn-info" href="index.php">Retour à la page d'accueil</a>
28 </div>
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <div class="container" style="display: block; margin-top: 3vh; position: relative;">
19     <?php
20
21     //Si l'utilisateur accède à cette page par erreur ou bien volontairement, il est
22     //instantanément redirigé vers l'accueil (index.php)
23     if ($_SESSION['connectedUser']['isAdmin'] == 2) {
24         } else {
25             //header('Location: index.php');
26             echo '<script>window.location.href = "index.php";</script>';
27         }
28
29         if ($_SESSION['AlertMessage']['type'] != null) { ?>
30             <div class="alert alert-<?= $_SESSION['AlertMessage']['type'] ?> alert-dismissible
31                 show" role="alert">
32                 <?= $_SESSION['AlertMessage']['message'] ?>
33                 <button type="button" class="close" data-dismiss="alert" aria-label="Close">
34                     <span aria-hidden="true">&times;</span>
35                 </button>
36             </div>
37         <?php
38             $_SESSION['AlertMessage'] = [
39                 'type' => null,
40                 'message' => null
41             ];
42         }
43     ?>
44     <div class="panel panel-default" style="width: 100%;">
45         <div class="panel-heading">
46             <!--Message de bienvenue-->
47             <h2>Administration des utilisateurs</h2>
48         </div>
49         <div class="panel-body">
50             <p>Vous pouvez ici gérer les droits donnés aux utilisateurs, c'est à dire que
51                 vous pouvez donner les droits administrateur à n'importe quel utilisateur en 1
52                 click.</p>
53         </div>
54
55     </div>
56     <div class="row">
57         <br>
58
59         <table class="table" style="background-color: #F0F0F0;">
60             <thead>
61                 <tr>
62                     <!--Haut du tableau-->
```

```
59             <th scope="col">idUser</th>
60             <th scope="col">Username</th>
61             <th scope="col">Firstname</th>
62             <th scope="col">Lastname</th>
63             <th scope="col">Email</th>
64             <th scope="col">isAdmin</th>
65             <th scope="col">Valider les changements</th>
66         </tr>
67     </thead>
68     <tbody>
69         <?php
70         foreach (User:: GetAllUsers() as $user) {
71             ?>
72             <tr>
73                 <form method="POST" action="index.php?
uc=admin&action=changeAdminRights&idUser=<?= $user->getIdUser() ?>">
74                     <td style="padding-left: 30px;"><?= $user->getIdUser() ?></td>
75                     <td style="padding-left: 40px;"><?= $user->getUserName() ?></td>
76                     <td style="padding-left: 40px;"><?= $user->getFirstName() ?></td>
77                     <td style="padding-left: 40px;"><?= $user->getLastName() ?></td>
78                     <td style="padding-left: 10px;"><?= $user->getEmail() ?></td>
79                     <td style="padding-left: 35px;"><input type="checkbox"
name="isAdmin" value="checkbox<?= $user->getIdUser() ?>" <?= $user-
>getIsAdmin() == 2 ? "checked" : "" ?>></td>
80                     <td style="padding-left: 70px;"><button type="submit" class="btn
btn-success">Confirmer</button> </td>
81                 </form>
82             </tr>
83         <?php } ?>
84     </tbody>
85 </table>
86
87     </div>
88 </div>
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <!DOCTYPE html>
19 <html lang="fr">
20
21 <head>
22   <meta http-equiv="content-type" content="text/html; charset=UTF-8">
23   <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
24   <meta charset="utf-8">
25   <title>Foto'Gal</title>
26   <!--Link des css-->
27   <link href="assets/css/bootstrap.css" rel="stylesheet">
28   <link href="assets/css/FotoGal.css" rel="stylesheet">
29   <link href=".//assets/css/custom.css" rel="stylesheet">
30 </head>
31
32 <body>
33   <div class="wrapper">
34     <div class="box">
35       <div class="row row-offcanvas row-offcanvas-left">
36         <!-- main right col -->
37         <div class="column col-sm-12 col-xs-12" id="main">
38           <!-- top nav -->
39           <div class="navbar navbar-blue navbar-static-top">
40             <div class="navbar-header">
41               <a href="index.php" class="navbar-brand logo">F </a>
42             </div>
43             <div>
44               <nav class="navbar-collapse" role="navigation">
45                 <ul class="nav navbar-nav">
46                   <li>
47                     <a href="index.php"><i class="glyphicon glyphicon-home"></i> Accueil</a>
48                   </li>
49
50                   <?php
51                     if ($_SESSION['connectedUser']['isConnected'] == false) {
52                   ?>
53                     <li>
54                       <a href="index.php?uc=login&action>ShowLoginForm"><i class="glyphicon glyphicon-plus"></i> Connecter-vous</a>
55                     </li>
56                   <?php
57                     } else {
58                   ?>
59                     <li>
60                       <a href="index.php?uc=production&action=show"><i class="glyphicon glyphicon-cloud-upload"></i> Poster</a>
```

```
61          </li>
62          <li>
63              <a href="index.php?uc=trier&action>ShowTriForm"><i class="glyphicon glyphicon-list-alt"></i> Trier</a>
64          </li>
65          <?php
66      }
67      ?>
68      <li>
69          <a href="index.php?uc=login&action>ShowPageFAQ"><i class="glyphicon glyphicon-question-sign"></i> Aide</a>
70      </li>
71  </ul>
72  <ul class="nav navbar-nav navbar-right">
73      <?php
74      if ($_SESSION["connectedUser"]["isConnected"] == true) {
75
76      ?>
77      <li class="dropdown">
78          <!--Icône Compte-->
79          <a href="#" class="dropdown-toggle" data-toggle="dropdown">
80
81              <?php
82                  //Si l'utilisateur connecté n'est pas un admin alors on affiche pas le texte "Administrateur" ni le logo admin
83                  if ($_SESSION['connectedUser']['isAdmin'] == 2) {
84
85                      <i><b>Administrateur </b></i>
86                      <i class="glyphicon glyphicon-cog"></i>
87
88                  } else {
89                      //Sinon on affiche juste le logo
90
91                      <i class="glyphicon glyphicon-cog"></i>
92
93                  }
94
95
96          </a>
97          <ul class="dropdown-menu">
98              <li><a href="index.php?uc=login&action>ShowProfile">Mon Profil</a></li>
99
100             <li><a href="index.php?uc=production&action>ShowMyProductions">Mes
101
102
103                 Productions</a></li>
104
105                 <?php
106                 if ($_SESSION['connectedUser']['isAdmin'] == 2) {
107                     //Si c'est un administrateur, on affiche les 2 pages administratives, sinon on affiche rien.
108
109                     <li><a href="index.php?
110
111                         uc=admin&action>ShowCategorieAdminPannel">CRUD Des Catégories</a></li>
112
113                     <li><a href="index.php?
114
115                         uc=admin&action>ShowUserAdminPannel">Gestion Des Droits</a></li>
116
117                     <?php
118                     }
119
120                     ?>
121                     <li><a href="index.php?uc=login&action=disconnect">Se Déconnecter</a></li>
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
968
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1197
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
```

```
113          </ul>
114      </nav>
115      </div>
116  </div>
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <div class="container" style="display: block; margin-top: 3vh; position: relative;">
19     <?php
20         if ($_SESSION['AlertMessage']['type'] != null) { ?>
21             <div class="alert alert-<?= $_SESSION['AlertMessage']['type'] ?> alert-dismissible" ?>
22                 <?= $_SESSION['AlertMessage']['message'] ?>
23                     <button type="button" class="close" data-dismiss="alert" aria-label="Close">
24                         <span aria-hidden="true">&times;;</span>
25                     </button>
26             </div>
27         <?php
28             $_SESSION['AlertMessage'] = [
29                 'type' => null,
30                 'message' => null
31             ];
32         }
33     ?>
34     <div class="panel panel-default" style="width: 100%;">
35         <div class="panel-heading">
36             <!-- Message de bienvenue-->
37             <h2>Bienvenue sur Foto'Gal</h2>
38         </div>
39         <div class="panel-body">
40             <h4>Affichage des 10 dernières productions par ordre chronologique.</h4>
41         </div>
42     </div>
43 </div>
44 <div class="container" style="margin-left: 250px;">
45     <div class="row">
46         <table class="table" style="background-color: #F0F0F0;">
47             <thead>
48                 <tr>
49                     <!--Haut du tableau-->
50                     <th scope="col" style="padding-left: 60px; padding-right: 60px;">Titre</th>
51                     <th scope="col" style="padding-left: 50px; padding-right: 50px;">Description</th>
52                     <th scope="col" style="padding-left: 50px; padding-right: 50px;">Catégorie</th>
53                     <th scope="col" style="padding-left: 70px; padding-right: 70px;">Image</th>
54                     <th scope="col" style="padding-left: 50px; padding-right: 50px;">Date de publication</th>
55                     <th scope="col" style="padding-left: 40px; padding-right: 40px;">Date de modification</th>
```

C:\Users\elias\Desktop\TPI_ZaiemElias\CodeSource\vue\home.php

```
56             <th scope="col" style="padding-left: 50px; padding-right: 50px;">Like&ampnbsp/&ampnbspDislike</th>
57             <th scope="col" style="padding-left: 20px; padding-right: 20px;">Détails</th>
58         </tr>
59     </thead>
60     <tbody>
61         <?php
62             //Affiche chaque production et affiche les données dans un tableau
63             foreach (Production::getAllProductionAccueil() as $production) {
64                 ?
65                 <tr>
66                     <td style="padding-left: 30px;"><?= $production->getTitreProduction() ?></td>
67                     <td style="padding-left: 30px;"><?= $production->getDescriptionProduction() ?></td>
68                     <!--Si le nom de la catégorie est null, alors on affiche Non définie, sinon on affiche le nom de la catégorie.-->
69                     <td style="padding-left: 58px;"><?= $production->getCategories_idCategorie() == null ? "Non définie" : Categorie::GetCategorieNameById($production->getCategories_idCategorie()) ?></td>
70                     <td style="max-width: 400px;"></td>
71                     <td><?= $production->getDateSoumission() ?></td>
72                     <td><?= $production->getDateModification() ?></td>
73                     <?php
74                         //Si l'utilisateur est connecté, alors on affiche rien, sinon on affiche le message suivant : "Connectez-vous pour pouvoir liker/disliker"
75                         if ($_SESSION['connectedUser']['isConnected'] == false) {
76                             ?
77                             <td>
78                                 <a>Connectez-vous pour pouvoir liker/disliker.</a>
79                             </td>
80                         <?php
81                     } else {
82                         ?
83                         <td>
84                             <?php
85                                 //Si l'utilisateur n'a pas liker/unliker alors on affiche les bouton, sinon on affiche juste si il a liker ou non.
86                                 $likeUnlike = LikeUnlike::GetLikeUnlike($_SESSION['connectedUser']['idUser'], $production->getIdProduction());
87                                 if ($likeUnlike == false) {
88                                     ?
89                                     <a class="btn btn-success" href="index.php?uc=production&action=like&id= <?= $production->getIdProduction() ?>">Like</a>&ampnbsp&ampnbsp;
90                                     <a class="btn btn-danger" href="index.php?uc=production&action=dislike&id= <?= $production->getIdProduction() ?>">Dislike</a><br>
91                                     <?php
92                                         } else {
93                                             ?
94                                             <a>Vous avez <?= $likeUnlike == 1 ? "liké" : "disliké" ?> cette publication.</a><br>
95                                         <?php
96                                         }
97                                         ?
98                                         <br>
99                                         <!-- Affichage du nombre de like/unlike de cette publication-->
```

```
99          <?= "Like : " . LikeUnlike::GetCompteurLikeUnlike
    ($production->getIdProduction(), 1) . "| Dislike : " .
    LikeUnlike::GetCompteurLikeUnlike($production->getIdProduction(),
2) ?>
100         </td>
101     <?php
102     }
103     ?>
104     <!--Bouton pour accéder à la page détails de la production-->
105     <td style="padding-left: 20px;"><a class="btn btn-info"
106 href="index.php?uc=production&action=showDetail&id=<?= $production-
    >getIdProduction() ?>">Voir</a></td>
107     </tr>
108     <?php } ?>
109   </tbody>
110 </table>
111 </div>
112 </div>
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <div class="container">
19     <?php
20         if ($_SESSION['AlertMessage']['type'] != null) {
21             ?>
22                 <div class="alert alert-<?= $_SESSION['AlertMessage']['type'] ?> alert-dismissible show" role="alert">
23                     <?= $_SESSION['AlertMessage']['message'] ?>
24                         <button type="button" class="close" data-dismiss="alert" aria-label="Close">
25                             <span aria-hidden="true">&times;</span>
26                         </button>
27                 </div>
28             <?php
29                 $_SESSION['AlertMessage'] = [
30                     "type" => null,
31                     "message" => null
32                 ];
33             ?>
34             ?>
35             <form method="post" action="index.php?uc=login&action=validateLogin">
36                 <div class="form-floating mb-3">
37                     <label for="inputEmail" id="idLabelEmail">Votre adresse mail :</label>
38                     <input class="form-control" name="Email" type="email"
39                         placeholder="example@example.com" required />
40                 </div>
41                 <br>
42                 <div class="form-floating mb-3">
43                     <label for="inputPassword">Votre mot de passe :</label>
44                     <input class="form-control" name="Password" type="password" placeholder="Entrez
45                         votre mot de passe ici" required />
46                 </div>
47                 <br>
48                 <div class="d-flex align-items-center justify-content-between mt-4 mb-0">
49                     <a href="index.php?uc=login&action>ShowRegisterForm"><b>Pas de compte ?
50                         S'inscrire ici</b></a><br>
51                     <a href="index.php?uc=login&action>ShowForgotPassword"><i>Mot de passe oublié ?</i>
52                         </a><br><br>
53                     <button class="btn btn-info" name="login" type="submit">Se connecter</button>
54                 </div>
55             </form>
56         </div>
57     </?php>
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <div class="container">
19     <?php
20         if ($_SESSION['AlertMessage']['type'] != null) {
21             ?>
22                 <div class="alert alert-<?= $_SESSION['AlertMessage']['type'] ?> alert-dismissible show" role="alert">
23                     <?= $_SESSION['AlertMessage']['message'] ?>
24                         <button type="button" class="close" data-dismiss="alert" aria-label="Close">
25                             <span aria-hidden="true">&times;</span>
26                         </button>
27                 </div>
28             <?php
29                 $_SESSION['AlertMessage'] = [
30                     "type" => null,
31                     "message" => null
32                 ];
33             ?>
34             <form method="post" action="index.php?uc=login&action=randomPasswordGeneration">
35                 <div class="form-floating mb-3">
36                     <label for="inputEmail" id="idLabelEmail">Votre adresse mail :</label>
37                     <input class="form-control" name="EmailRecovery" type="email"
38                         placeholder="example@example.com" required />
39                 </div>
40                 <br>
41
42                 <br>
43                 <div class="d-flex align-items-center justify-content-between mt-4 mb-0">
44                     <button class="btn btn-success" name="recoverPassword" type="submit">Récupérer le ↵
45                         mdp</button>
46                 </div>
47             </form>
48         </div>
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <div class="container" style="display: block; margin-top: 3vh; position: relative;">
19     <?php
20         if ($_SESSION['AlertMessage']['type'] != null) { ?>
21             <div class="alert alert-<?= $_SESSION['AlertMessage']['type'] ?> alert-dismissible show" role="alert">
22                 <?= $_SESSION['AlertMessage']['message'] ?>
23                 <button type="button" class="close" data-dismiss="alert" aria-label="Close">
24                     <span aria-hidden="true">&times;</span>
25                 </button>
26             </div>
27         <?php
28             $_SESSION['AlertMessage'] = [
29                 'type' => null,
30                 'message' => null
31             ];
32         }
33     ?>
34     <div class="panel panel-default" style="width: 100%;">
35         <div class="panel-heading">
36             <!--Message de bienvenue-->
37             <h1>Mes Productions : </h1>
38             <h4>Vous pouvez supprimer / modifier vos productions en 1 click !</h4>
39         </div>
40     </div>
41     <br><br>
42     <div class="row">
43         <table class="table" style="background-color: #F0F0F0;">
44             <thead>
45                 <tr>
46                     <!--Haut du tableau-->
47                     <th scope="col">Titre</th>
48                     <th scope="col">Description</th>
49                     <th scope="col">Catégorie</th>
50                     <th scope="col">Image</th>
51                     <th scope="col">Date de publication</th>
52                     <th scope="col">Date de modification</th>
53                     <th scope="col">Like&nbsp;/&nbsp;Dislike</th>
54                     <th scope="col">Modifier</th>
55                     <th scope="col">Supprimer</th>
56                 </tr>
57             </thead>
58             <tbody>
59                 <?php
60                     //Affiche chaque production et affiche les données dans un tableau
61                     foreach (Production::getAllUserProductions($_SESSION['connectedUser'])
```

```
62     ['idUser']) as $production) {  
63         ?>  
64             <tr>  
65                 <!--Récupère les données avec les gettersetter-->  
66                 <td><?= $production->getTitreProduction() ?></td>  
67                 <td><?= $production->getDescriptionProduction() ?></td>  
68                 <td><?= $production->getCategories_idCategorie() == null ? "Non  
69                     définie" : Categorie::GetCategorieNameById($production-  
70                     >getCategories_idCategorie()) ?></td>  
71                     <td style="max-width: 500px;"></td>  
73                     <td><?= $production->getDate_soumission() ?></td>  
74                     <td><?= $production->getDate_modification() ?></td>  
75                     <td><?= "Like : " . LikeUnlike::GetCompteurLikeUnlike($production-  
76                         >getIdProduction(), 1) . " | Dislike : " .  
77                         LikeUnlike::GetCompteurLikeUnlike($production->getIdProduction(), 2) ?>  
78                     ></td>  
79                     <td>&nbsp;&nbsp;<a class="btn btn-success glyphicon glyphicon-edit"  
80                         href="index.php?  
81                             uc=production&action>ShowEditProductionPage&idProduction=<?=  
82                             $production->getIdProduction() ?>"> </a></td>  
83                     <td>&nbsp;&nbsp;&nbsp;<a class="btn btn-danger glyphicon  
84                         glyphicon-remove" href="index.php?  
85                             uc=production&action=deleteProduction&idProduction=<?= $production-  
86                             >getIdProduction() ?>"> </a></td>  
87                     </tr>  
88             <?php } ?>  
89         </tbody>  
90     </div>  
91 </div>
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <div class="container">
19     <div class="panel-heading">
20         <!-- Manuel d'utilisateur inclu dans le site : -->
21         <h1><b>Manuel d'utilisateur : </b></h1>
22         <h4>Ici vous allez retrouver toutes les réponses à vos questions. Cette page a été
23             créé dans le but de vous aider à utiliser le site Foto'Gal.</h4>
24     </div>
25     <div class="panel panel-default" style="width: 100%;">
26
27         <div class="panel-heading">
28             <h3>Comment puis-je me connecter ?</h3>
29             <p>Rendez-vous sur la page connexion, puis connectez-vous. Si vous n'avez pas de
30                 compte, vous pouvez vous inscrire en cliquant sur "Pas de compte ? s'inscrire
31                 ici"</p>
32         </div>
33         <div class="panel-heading">
34             <h3>Je veux poster une Production, comment dois-je faire ?</h3>
35             <p>Cliquez sur le bouton "Post" en haut de la page puis remplissez tout les
36                 champs et cliquez sur "Publier".</p>
37         </div>
38         <div class="panel-heading">
39             <h3>Je souhaite trier les Productions, comment faire ?</h3>
40             <p>Cliquez sur le bouton "Trier" en haut de la page et sélectionnez la catégorie
41                 que vous voulez trier, puis appuyez sur le bouton envoyer. Voilà ! Vous avez
42                 trié les productions.</p>
43         </div>
44         <div class="panel-heading">
45             <h3>Comment dois-je faire pour modifier mon profil ?</h3>
46             <p>Cliquez sur le menu déroulant en haut à droite de la page, puis cliquez sur
47                 "Mon Profil", une fois fait il vous suffit juste de modifier vos informations
48                 puis de cliquer sur le bouton "Mettre à jour son profil".</p>
49         </div>
50         <div class="panel-heading">
51             <h3>Je souhaite changer le like/unlike d'une production, comment dois-je faire ?
52                 </h3>
53             <p>Tout d'abord rendez-vous sur la page de tri et retrouvez la production dont
54                 vous souhaitez changer le like/unlike. Une fois cela fait, cliquez sur le bouton
55                 "Voir" qui est à droite du tableau (bouton bleu). Maintenant que vous êtes sur
56                 cette page de détails, vous pouvez changer de like/unlike en 1 clique.</p>
57         </div>
58         <div class="panel-heading">
59             <h3>Vous êtes un administrateur et vous désirez vous connecter ?</h3>
60             <p>Rien de plus simple, allez sur la page connexion. Connectez-vous avec votre
61                 compte, si vous êtes un administrateur vous aurez un petit texte en haut à
62                 droite de la page qui dit "Administrateur". Si ce n'est pas le cas, vous n'êtes
63                 pas administrateur.</p>
64         </div>
```

tout simplement pas administrateur, vous pouvez demander à un admin de vous donner les droits.</p>

49 </div>

50 </div>

51 </div>

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <div class="container" style="display: block; margin-top: 3vh;">
19     <?php
20         if ($_SESSION['AlertMessage']['type'] != null) { ?>
21             <div class="alert alert-<?= $_SESSION['AlertMessage']['type'] ?> alert-dismissible show" role="alert">
22                 <?= $_SESSION['AlertMessage']['message'] ?>
23                 <button type="button" class="close" data-dismiss="alert" aria-label="Close">
24                     <span aria-hidden="true">&times;</span>
25                 </button>
26             </div>
27         <?php
28             $_SESSION['AlertMessage'] = [
29                 'type' => null,
30                 'message' => null
31             ];
32         }
33     ?>
34     <!--Visuel du formulaire de post-->
35     <h1>Publiez une production sur Foto'Gal !</h1>
36     <form method="POST" action="index.php?uc=production&action=validate" enctype="multipart/form-data">
37         <!--Formulaire d'envoi du post-->
38         <div class="form-group">
39             <label for="idTitreProduction">Titre de la production : </label>
40             <textarea class="form-control" id="idTitreProduction" rows="1" name="titreProduction" required></textarea>
41         </div>
42         <div class="form-group">
43             <label for="idDescriptionProduction">Description de la production : </label>
44             <textarea class="form-control" id="idDescriptionProduction" rows="4" name="descriptionProduction" required></textarea>
45         </div>
46         <div class="form-group">
47             <label for="select-categorie">Catégorie : </label>
48
49             <select name="categorieProduction" id="select-categorie" required>
50                 <?php
51                     foreach (Categorie:: GetAllCategories() as $categorie) {
52                         ?>
53                         <option value="= $categorie-&gt;getIdCategorie() ?&gt;&lt;?= $categorie-&gt;getNom() ?&gt;"&gt;&lt;/option&gt;
54                 &lt;?php
55                 }
56             ?&gt;
57             &lt;/select&gt;
58         &lt;/div&gt;
59     &lt;/form&gt;
60 </pre
```

```
58      </div>
59
60      <div class="form-group">
61          <label for="idFile">Image : </label>
62          <input type="file" class="form-control-file" id="idFile" accept=".jpg,.png" name="filesPost[]" required>
63      </div>
64      <input class="btn btn-success" type="submit" value="Publier">
65  </form>
66 </div>
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12
13 Auteur : Elias Zaiem
14 Date : 18.05.2022
15 Projet : TPI Elias Zaiem Mai-2022
16 Prof De TPI : Mr.Garchery
17 Classe : I.DA-P4A
18 --><div class="container" style="display: block; margin-top: 3vh; position: relative;">
19     getTitreProduction() ?>" width="50%"><br>
20     <?php
21     $likeUnlike = LikeUnlike::GetLikeUnlike($_SESSION['connectedUser']['idUser'],
22         $production->getIdProduction());
23     if ($likeUnlike == false) { ?>
24         <a class="btn btn-success" href="index.php?uc=production&action=like&id= <?= $production->getIdProduction() ?>">>Like</a>&ampnbsp&ampnbsp;
25         <a class="btn btn-danger" href="index.php?uc=production&action=dislike&id= <?= $production->getIdProduction() ?>">>Dislike</a><br>
26     <?php
27     } else {
28     ?>
29         <a>Vous avez <?= $likeUnlike == 1 ? "liké" : "disliké" ?> cette publication.</a>
30     <?php
31     }
32     ?>
33     <div class="form-group">
34         <div class="col-xs-3">
35             <h4>Titre de la production : </h4>
36             <input type="text" id="titre" name="titre" value="<?= $production->getTitreProduction() ?>" readonly><br>
37             <h4>Description : </h4>
38             <input type="text" id="description" name="description" value="<?= $production->getDescriptionProduction() ?>" readonly><br>
39             <h4>Catégorie : </h4>
40             <input type="text" id="categorie" name="categorie" value="<?= $production->getCategories_idCategorie() == null ? "Non définie" : Categorie::GetCategorieNameById($production->getCategories_idCategorie()) ?>" readonly><br>
41             <h4>Nom du fichier d'originie : </h4>
42             <input type="text" id="filename" name="filename" value="<?= $production->getFilename() ?>" readonly><br>
43         </div>
44         <div class="form-group">
45             <h4>Date de soumission : </h4>
46             <input type="text" id="date_soumission" name="date_soumission" value="<?= $production->getDate_soumission() ?>" readonly><br>
47             <h4>Date de modification : </h4>
48             <input type="text" id="date_modification" name="date_modification" value="<?= $production->getDate_modification() ?>" readonly><br>
49             <h4>Publiée par :</h4>
50             <input type="text" id="publier_par" name="publier_par" value="<?= User::GetUsernameById($production->getUser_idUser()) ?>" readonly><br>
```

```
50      </div>
51      <div class="container">
52          <a class="btn btn-success" href="index.php?
53              uc=production&action=editLikePost&idProduction= <?= $production->getIdProduction?>
54              () ?>&like=1">Like</a>&ampnbsp&ampnbsp;
55
56          <a class="btn btn-danger" href="index.php?
57              uc=production&action=editLikePost&idProduction= <?= $production->getIdProduction?>
58              () ?>&like=2">Dislike</a><br>
59      </div>
60
61  </div>
62
63  </div>
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <div class="container" style="display: block; margin-top: 3vh;">
19     <h1>Modification Du Profil</h1>
20     <form method="POST" action="index.php?uc=login&action=UpdateUserInfo">
21         <div class="form-group">
22             <label>Nom d'utilisateur : </label><br>
23             <input type="text" name="username" value="<?= $_SESSION['connectedUser'] ?>['username'] ?>" required>
24         </div>
25         <div class="form-group">
26             <label>Prenom : </label><br>
27             <input type="text" name="firstname" value="<?= $_SESSION['connectedUser'] ?>['firstname'] ?>" required>
28         </div>
29         <div class="form-group">
30             <label>Nom : </label><br>
31             <input type="text" name="lastname" value="<?= $_SESSION['connectedUser'] ?>['lastname'] ?>" required>
32         </div>
33         <div class="form-group">
34             <label>Email : </label><br>
35             <input type="email" name="email" value="<?= $_SESSION['connectedUser']['email'] ?>" required>
36         </div>
37         <input type="submit" class="btn btn-success pull-left" value="Mettre à jour son profil"><br>
38     </form><br>
39     <div class="form-group">
40         <label>Modifier son mot de passe : </label>
41         <a class="btn btn-info" href="index.php?uc=login&action=changePassword">Changer</a>&ampnbsp&ampnbsp;
42     </div>
43 </div>
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <div class="container">
19     <?php
20         if ($_SESSION['AlertMessage']['type'] != null) {
21             ?>
22                 <div class="alert alert-<?= $_SESSION['AlertMessage']['type'] ?> alert-dismissible show" role="alert">
23                     <?= $_SESSION['AlertMessage']['message'] ?>
24                         <button type="button" class="close" data-dismiss="alert" aria-label="Close">
25                             <span aria-hidden="true">&times;</span>
26                         </button>
27                 </div>
28             <?php
29                 $_SESSION['AlertMessage'] = [
30                     "type" => null,
31                     "message" => null
32                 ];
33             ?>
34             <form method="POST" action="index.php?uc=login&action=validateRegister">
35                 <div class="form-floating mb-3">
36                     <div class="form-floating mb-3 mb-md-0">
37                         <label id="idLabelPrenom" for="inputFirstName">Prénom</label>
38                         <input class="form-control" id="fname" name="FirstName" type="text" value=<?=
39                             = $_SESSION['UserInfos']['firstname'] ?>" placeholder="Entrez votre prénom" required />
40                     </div>
41                 </div>
42                 <div class="form-floating mb-3">
43                     <div class="form-floating">
44                         <label for="inputLastName">Nom</label>
45                         <input class="form-control" id="lname" name="LastName" type="text" value=<?=
46                             $SESSION['UserInfos']['lastname'] ?>" placeholder="Entrez votre nom" required />
47                     </div>
48                 </div>
49                 <div class="form-floating mb-3">
50                     <div class="form-floating">
51                         <label for="inputUserName">Pseudo</label>
52                         <input class="form-control" id="uname" name="UserName" type="text" value=<?=
53                             $SESSION['UserInfos']['username'] ?>" placeholder="Entrez votre nom d'utilisateur" required />
54                     </div>
55                 </div>
56                 <div class="form-floating mb-3">
57                     <div class="form-floating">
```

```
56          <label for="inputEmail">Email</label>
57          <input class="form-control" id="email" name="Email" type="email" value="= $_SESSION['UserInfos']['email'] ?" placeholder="example@gmail.com" required />
58      </div>
59  </div>
60  <div class="form-floating mb-3">
61      <div class="form-floating mb-3">
62          <div class="form-floating mb-3 mb-md-0">
63              <label for="inputPassword">Mot de passe</label>
64              <input class="form-control" id="password" name="Password" type="password" value="= $_SESSION['UserInfos']['password'] ?" placeholder="Entrez un mot de passe" required />
65          </div>
66      </div>
67      <div class="form-floating mb-3">
68          <div class="form-floating mb-3 mb-md-0">
69              <label for="inputPasswordConfirm">Confirmer le mot de passe</label>
70              <input class="form-control" id="confirmpassword" name="ConfirmPassword" type="password" placeholder="Confirmez le mot de passe" required />
71          </div>
72      </div>
73  </div>
74  <br>
75  <div class="d-flex align-items-center justify-content-between mt-4 mb-0">
76      <a href="index.php?uc=login&action>ShowLoginForm"><b>Déjà un compte ? Se connecter ici</b></a><br><br>
77      <button class="btn btn-info" name="submit" type="submit">S'inscrire</button>
78  </div>
79  </form>
80 </div>
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <div class="container" style="display: block; margin-top: 3vh; position: relative;">
19     <div class="panel panel-default" style="width: 100%;">
20         <div class="panel-heading">
21             <h3>Choisissez la catégorie que vous souhaitez trier : </h3>
22             <form method="post" action="index.php?uc=trier&action=filterCategorie">
23                 <select name="categorieProduction" id="select-categorie" required>
24                     <option selected>Toutes Catégories</option>
25                     <?php
26                         // Va chercher les catégories dans la base de données et les affiche
27                         // ensuite dans la liste déroulante.
28                         foreach (Categorie:: GetAllCategories() as $categorie) {
29                             ?>
29                             <option name="<?= $categorie->getIdCategorie() ?>" value="<?=
29                             $categorie->getIdCategorie() ?>><?= $categorie->getNom() ?></option>
30                         <?php
31                         }
32                         ?>
33                     </select>
34                     <input type="submit" value="Trier" />
35                 </form>
36             </div>
37         </div>
38     </div>
39     <div class="container" style="margin-left: 250px;">
40         <table class="table" style="background-color: #F0F0F0;">
41             <thead>
42                 <tr>
43                     <!--Haut du tableau-->
44                     <th scope="col" style="padding-left: 60px; padding-right: 60px;">Titre</th>
45                     <th scope="col" style="padding-left: 50px; padding-right:
45                         50px;">Description</th>
46                     <th scope="col" style="padding-left: 50px; padding-right: 50px;">Catégorie</
46                         th>
47                     <th scope="col" style="padding-left: 70px; padding-right: 70px;">Image</th>
48                     <th scope="col" style="padding-left: 50px; padding-right: 50px;">Date de
48                         publication</th>
49                     <th scope="col" style="padding-left: 40px; padding-right: 40px;">Date de
49                         modification</th>
50                     <th scope="col" style="padding-left: 50px; padding-right: 50px;">Like&nbsp;/
50                         &nbsp;Dislike</th>
51                     <th scope="col" style="padding-left: 20px; padding-right: 20px;">Détails</th>
52                 </tr>
53             </thead>
54             <tbody>
55                 <?php
```

```
56         //Affiche chaque production et affiche les données dans un tableau
57         foreach (Production::getAllProductions($_SESSION['filterCategorie']), $_SESSION
58             ['orderProduction']) as $production) {
59             ?>
60             <tr>
61                 <td style="padding-left: 30px;"><?= $production->getTitreProduction() ?>
62                     </td>
63                 <td style="padding-left: 35px;"><?= $production->getDescriptionProduction()
64                     () ?></td>
65                     <!--Si le nom de la catégorie est null, alors on affiche Non définie,
66                         sinon on affiche le nom de la catégorie.-->
67                 <td style="padding-left: 58px;"><?= $production-
68                     >getCategories_idCategorie() == null ? "Non définie" :
69                         Categorie::GetCategorieNameById($production->getCategories_idCategorie
70                             ()) ?></td>
71                 <td style="max-width: 400px;"></td>
73                 <td><?= $production->getDate_soumission() ?></td>
74                 <td><?= $production->getDate_modification() ?></td>
75                 <td style="padding-left: 40px;"><?= "Like : " .
76                     LikeUnlike::GetCompteurLikeUnlike($production->getIdProduction(), 1) .
77                     " | Dislike : " . LikeUnlike::GetCompteurLikeUnlike($production-
78                         >getIdProduction(), 2) ?></td>
79                     <!--Bouton pour accéder à la page détails de la production-->
80                     <td style="padding-left: 20px;"><a class="btn btn-info" href="index.php?
81                         uc=production&action=showDetail&id=<?= $production->getIdProduction() ?>
82                         ">Voir</a></td>
83             </tr>
84             <?php } ?>
85         </tbody>
86     </table>
87 </div>
88 </div>
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <?php
19 class Categorie
20 {
21     private $idCategorie;
22     private $nom;
23
24     /**
25      * Get the value of idCategorie
26      */
27     public function getIdCategorie()
28     {
29         return $this->idCategorie;
30     }
31
32     /**
33      * Set the value of idCategorie
34      *
35      * @return self
36      */
37     public function setIdCategorie($idCategorie)
38     {
39         $this->idCategorie = $idCategorie;
40
41         return $this;
42     }
43
44     /**
45      * Get the value of nom
46      */
47     public function getNom()
48     {
49         return $this->nom;
50     }
51
52     /**
53      * Set the value of nom
54      *
55      * @return self
56      */
57     public function setNom($nom)
58     {
59         $this->nom = $nom;
60
61         return $this;
62     }
```

```
63
64
65     public static function GetCategorieNameById($idCategorie)
66     {
67         $req = MonPdo::getInstance()->prepare("SELECT * FROM categories WHERE idCategorie
68             = :idCategorie");
69         $req->setFetchMode(PDO::FETCH_CLASS | PDO::FETCH_PROPS_LATE, 'Categorie');
70         $req->bindParam(":idCategorie", $idCategorie);
71         $req->execute();
72         $result = $req->fetch();
73
74         return $result->getNom();
75     }
76
77     public static function GetAllCategories()
78     {
79         $req = MonPdo::getInstance()->prepare("SELECT * FROM categories");
80         $req->setFetchMode(PDO::FETCH_CLASS | PDO::FETCH_PROPS_LATE, 'Categorie');
81         $req->execute();
82         $result = $req->fetchAll();
83
84         return $result;
85     }
86
87 //Supprime la catégorie selon son Id
88     public static function DeleteCategorie($idCategorie)
89     {
90         $req = MonPdo::getInstance()->prepare("DELETE FROM categories WHERE idCategorie
91             = :idCategorie");
92         $req->bindParam(":idCategorie", $idCategorie);
93         $req->execute();
94
95 //Ajouter une catégorie depuis le pannel admin
96     public static function AddNewCategorie($CategorieName)
97     {
98         $req = MonPdo::getInstance()->prepare("INSERT INTO categories(nom) VALUES (:nom)");
99         $req->bindParam(":nom", $CategorieName);
100        $req->execute();
101    }
102
103    public static function UpdateCategorie($idCategorie, $nameCategorie)
104    {
105        $req = MonPdo::getInstance()->prepare("UPDATE categories SET nom =:nom WHERE
106            idCategorie =:idCategorie");
107        $req->bindParam(":nom", $nameCategorie);
108        $req->bindParam(":idCategorie", $idCategorie);
109        $req->execute();
110    }
111 }
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <?php // Template du pdo
19
20 /**
21 * Casse d'accès aux données Utilise les services de la classe PDO
22 *
23 */
24
25 class MonPdo
26 {
27     private static $serveur = 'mysql:host=localhost';
28     private static $bdd = 'dbname=fotogal';
29     private static $user = 'root';
30     private static $mdp = '2404';
31     private static $monPdo;
32     private static $unPdo = null;
33
34     // Constructeur privé, crée l'instance de PDO qui sera sollicité
35     // pour toutes les méthodes de la classe
36
37     private function __construct()
38     {
39         MonPdo::$unPdo = new PDO(MonPdo::$serveur . ";" . MonPdo::$bdd, MonPdo::$user,
40             MonPdo::$mdp);
41         MonPdo::$unPdo->query('SET names utf8');
42         MonPdo::$unPdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
43     }
44
45     public function __destruct()
46     {
47         MonPdo::$unPdo = null;
48     }
49
50     /**
51      * Fonction statique qui crée l'unique instance de la classe
52      * Appel : $instanceMonPdo = MonPdo::getInstance();
53      *
54      * @return l'unique objet de la classe MonPdo
55      */
56     public static function getInstance()
57     {
58         if (self::$unPdo == null) {
59             self::$monPdo = new MonPdo();
60         }
61         return self::$unPdo;
62     }
63 }
```

62 ?>

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <?php
19 class LikeUnlike
20 {
21     private $like;
22     private $date;
23     private $utilisateurs_idUser;
24     private $production_idProduction;
25
26
27     /**
28      * Get the value of like
29      */
30     public function getLike()
31     {
32         return $this->like;
33     }
34     /**
35      * Set the value of like
36      *
37      * @return self
38      */
39
40     public function setLike($like)
41     {
42         $this->like = $like;
43
44         return $this;
45     }
46
47
48     /**
49      * Get the value of date
50      */
51     public function getDate()
52     {
53         return $this->date;
54     }
55
56
57     /**
58      * Set the value of date
59      *
60      * @return self
61      */
62     public function setDate($date)
```

```
63     {
64         $this->date = $date;
65
66         return $this;
67     }
68
69
70     /**
71      * Get the value of utilisateurs_idUser
72      */
73     public function getUtilisateurs_idUser()
74     {
75         return $this->utilisateurs_idUser;
76     }
77
78     /**
79      * Set the value of utilisateurs_idUser
80      *
81      * @return self
82      */
83     public function setUtilisateurs_idUser($utilisateurs_idUser)
84     {
85         $this->utilisateurs_idUser = $utilisateurs_idUser;
86
87         return $this;
88     }
89     /**
90      * Get the value of production_idProduction
91      */
92     public function getProduction_idProduction()
93     {
94         return $this->production_idProduction;
95     }
96
97
98     /**
99      * Set the value of production_idProduction
100     *
101     * @return self
102     */
103    public function setProduction_idProduction($production_idProduction)
104    {
105        $this->production_idProduction = $production_idProduction;
106
107        return $this;
108    }
109
110    //Fonction pour liker une production
111    public function LikePost()
112    {
113        $like = $this->getLike();
114        $date = date("Y-m-d H:i:s");
115        $idUser = $this->getUtilisateurs_idUser();
116        $idProduction = $this->getProduction_idProduction();
117
118        //Insert dans la base de données like_unlike
119        $req = MonPdo::getInstance()->prepare("INSERT INTO `like_unlike` (`like`, `date`,
120                                              `utilisateurs_idUser`, `production_idProduction`) VALUES
121                                              (:like, :date, :idUser, :idProduction)");
122        $req->bindParam(":like", $like);
123        $req->bindParam(":date", $date);
124        $req->bindParam(":idUser", $idUser);
```

```
123     $req->bindParam(":idProduction", $idProduction);
124     $req->execute();
125 }
126
127 //Fonction de modification de post
128 public function EditLikePost()
129 {
130     $like = $this->getLike();
131     $idUser = $this->getUtilisateurs_idUser();
132     $idProduction = $this->getProduction_idProduction();
133
134     $req = MonPdo::getInstance()->prepare("UPDATE like_unlike SET `like` =
135         :likeProduction WHERE production_idProduction = :idProduction AND
136         utilisateurs_idUser = :idUser");
137     $req->bindParam(":likeProduction", $like);
138     $req->bindParam(":idUser", $idUser);
139     $req->bindParam(":idProduction", $idProduction);
140     $req->execute();
141 }
142
143 public static function GetLikeUnlike($idUser, $idPost)
144 {
145     $req = MonPdo::getInstance()->prepare("SELECT * FROM like_unlike WHERE
146         utilisateurs_idUser = :idUser AND production_idProduction = :idProduction;");
147     $req->setFetchMode(PDO::FETCH_CLASS | PDO::FETCH_PROPS_LATE, 'LikeUnlike'); // 
148     methode de fetch
149     $req->bindParam(":idUser", $idUser);
150     $req->bindParam(":idProduction", $idPost);
151     $req->execute(); // executer la requette
152     $result = $req->fetch();
153
154     if ($result != false) {
155         return $result->getLike();
156     } else {
157         return false;
158     }
159 }
160
161 // 1 = like, 2 = dislike
162 public static function GetCompteurLikeUnlike($idProduction, $likeUnlike)
163 {
164     $req = MonPdo::getInstance()->prepare("SELECT COUNT(idLikeUnlike) as 'compteur' FROM
165         like_unlike WHERE production_idProduction = :idProduction AND like_unlike.like =
166         :likeUnlike");
167     $req->bindParam(':idProduction', $idProduction);
168     $req->bindParam(':likeUnlike', $likeUnlike);
169     $req->execute();
170
171     /*SELECT productions.*, like_unlike.*
172      FROM productions
173      INNER JOIN like_unlike ON productons.idProduction =
174          like_unlike.production_idProduction
175      WHERE like_unlike.like = 0;*/
176
177     $result = $req->Fetch();
178     return $result['compteur'];
179 }
180
181 }
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <?php
19 class Production
20 {
21     //Déclaration de toutes les variables
22     private $idProduction;
23     private $titre;
24     private $description;
25     private $date_soumission;
26     private $date_modification;
27     private $filename;
28     private $categories_idCategorie;
29     private $utilisateurs_idUser;
30     private $compteurCategorie;
31
32
33     //Récup la valeur du idPost
34     public function getIdProduction()
35     {
36         return $this->idProduction;
37     }
38
39     /**
40      * Set the value of idPost
41      *
42      * @return self
43      */
44     public function setIdProduction($idProduction)
45     {
46         $this->idProduction = $idProduction;
47
48         return $this;
49     }
50
51     //Récup la valeur du titre
52     public function getTitreProduction()
53     {
54         return $this->titre;
55     }
56
57     /**
58      * Set the value of titre
59      *
60      * @return self
61      */
62     public function setTitreProduction($titre)
```

```
63     {
64         $this->titre = $titre;
65
66         return $this;
67     }
68
69     //Récup la valeur du description
70     public function getDescriptionProduction()
71     {
72         return $this->description;
73     }
74
75     /**
76      * Set the value of description
77      *
78      * @return self
79      */
80     public function setDescriptionProduction($description)
81     {
82         $this->description = $description;
83
84         return $this;
85     }
86
87     /**
88      * Get the value of compteurPost
89      */
90     public function getCompteurProduction()
91     {
92         return $this->compteurPost;
93     }
94
95     /**
96      * Set the value of compteurPost
97      *
98      * @return self
99      */
100    public function setCompteurProduction($compteurPost)
101    {
102        $this->compteurPost = $compteurPost;
103
104        return $this;
105    }
106
107
108
109    //Récup la valeur de la creationDate
110    public function getDate_soumission()
111    {
112        return $this->date_soumission;
113    }
114
115    /**
116      * Set the value of creationDate
117      *
118      * @return self
119      */
120    public function setDate_soumission($date_soumission)
121    {
122        $this->date_soumission = $date_soumission;
123
124        return $this;
```

```
125     }
126
127     //Récup la valeur de la modificationDate
128     public function getDate_modification()
129     {
130         return $this->date_modification;
131     }
132
133     /**
134      * Set the value of modificationDate
135      *
136      * @return self
137      */
138     public function setDate_modification($date_modification)
139     {
140         $this->date_modification = $date_modification;
141
142         return $this;
143     }
144
145
146     /**
147      * Get the value of categories_idCategorie
148      */
149     public function getCategories_idCategorie()
150     {
151         return $this->categories_idCategorie;
152     }
153
154     /**
155      * Set the value of categories_idCategorie
156      *
157      * @return self
158      */
159     public function setCategories_idCategorie($categories_idCategorie)
160     {
161         $this->categories_idCategorie = $categories_idCategorie;
162
163         return $this;
164     }
165
166     /**
167      * Get the value of user_idUser
168      */
169     public function getUser_idUser()
170     {
171         return $this->utilisateurs_idUser;
172     }
173
174     /**
175      * Set the value of user_idUser
176      *
177      * @return self
178      */
179     public function setUser_idUser($utilisateurs_idUser)
180     {
181         $this->utilisateurs_idUser = $utilisateurs_idUser;
182
183         return $this;
184     }
185
186     /**
```

```
187     * Get the value of filename
188     */
189     public function getFilename()
190     {
191         return $this->filename;
192     }
193
194 /**
195 * Set the value of filename
196 *
197 * @return self
198 */
199 public function setFilename($filename)
200 {
201     $this->filename = $filename;
202
203     return $this;
204 }
205
206
207 //*****
208 //Fonctions//
209
210 // Ajout de la production en base de données
211 public static function AddProduction(Production $production)
212 {
213
214     $titre = $production->getTitreProduction();
215     $description = $production->getDescriptionProduction();
216     $date_soumission = $production->getDate_soumission();
217     $date_modification = $production->getDate_modification();
218     $filename = $production->getFilename();
219     $idCategorie = $production->getCategories_idCategorie();
220     $idUser = $production->getUser_idUser();
221
222
223     $req = MonPdo::getInstance()->prepare("INSERT INTO productions(titre, description,
224                                         date_soumission, date_modification, filename, categories_idCategorie ,
225                                         utilisateurs_idUser) VALUES
226                                         (:titre, :descriptionPost, :date_soumission, :date_modification, :filenamePost, :i
227                                         dCategorie, :idUser);");
228     $req->bindParam(":titre", $titre);
229     $req->bindParam(":descriptionPost", $description);
230     $req->bindParam(":date_soumission", $date_soumission);
231     $req->bindParam(":date_modification", $date_modification);
232     $req->bindParam(":filenamePost", $filename);
233     $req->bindParam(":idCategorie", $idCategorie);
234     $req->bindParam(":idUser", $idUser);
235
236
237     $req->execute(); // executer la requette
238
239     return MonPdo::getInstance()->lastInsertId();
240 }
241
242
243 //Récup toutes les productions par date de publication
244 public static function getAllProductionByDate()
245 {
246     $req = MonPdo::getInstance()->prepare("SELECT * FROM productions ORDER BY
247                                         date_soumission DESC;");
248     $req->setFetchMode(PDO::FETCH_CLASS | PDO::FETCH_PROPS_LATE, 'Production'); // methode de fetch
```

```
243     $req->execute(); // executer la requette
244
245     $lesResultats = $req->fetchAll();
246     return $lesResultats;
247 }
248
249 //Récup toutes les productions
250 public static function getAllProductions($idCategorie = 0, $order = "date_soumission")
251 {
252
253     if($idCategorie == 0){
254         $req = MonPdo::getInstance()->prepare("SELECT * FROM productions ORDER
255                                         BY :ORDERPROD DESC;");
256     }
257     else{
258         $req = MonPdo::getInstance()->prepare("SELECT * FROM productions WHERE
259                                         categories_idCategorie = :IDCATEGORIE ORDER BY :ORDERPROD DESC;");
260         $req->bindParam(':IDCATEGORIE', $idCategorie);
261     }
262     $req->bindParam(':ORDERPROD', $order);
263     $req->setFetchMode(PDO::FETCH_CLASS | PDO::FETCH_PROPS_LATE, 'Production'); //
264         methode de fetch
265     $req->execute(); // executer la requette
266
267
268 //Récup toutes les productions
269 public static function getAllProductionAccueil()
270 {
271
272     $req = MonPdo::getInstance()->prepare("SELECT * FROM productions ORDER BY
273                                         date_soumission DESC LIMIT 10;");
274
275     $req->setFetchMode(PDO::FETCH_CLASS | PDO::FETCH_PROPS_LATE, 'Production'); //
276         methode de fetch
277     $req->execute(); // executer la requette
278
279
280 //Récup toutes les productions
281 public static function getAllUserProductions($idUser)
282 {
283
284     $req = MonPdo::getInstance()->prepare("SELECT * FROM productions WHERE
285                                         utilisateurs_idUser = :idUser ORDER BY date_soumission DESC;");
286     $req->setFetchMode(PDO::FETCH_CLASS | PDO::FETCH_PROPS_LATE, 'Production'); //
287         methode de fetch
288     $req->bindParam(':idUser', $idUser);
289     $req->execute(); // executer la requette
290
291
292 //Compteur de production(s)
293 public static function CountAllProduction()
294 {
295
296     $req = MonPdo::getInstance()->prepare("SELECT COUNT(idProduction) as
297                                         'compteurProduction' FROM productions;");
298     $req->setFetchMode(PDO::FETCH_CLASS | PDO::FETCH_PROPS_LATE, 'Production');
```

```
297     $req->execute();
298     $result = $req->fetch();
299     return $result->getCompteurPost();
300 }
301
302
303
304 //Récupère les productions
305 public static function GetProdutionById($idProduction)
306 {
307     $req = MonPdo::getInstance()->prepare("SELECT * FROM productions WHERE idProduction = :idProduction");
308     $req->setFetchMode(PDO::FETCH_CLASS | PDO::FETCH_PROPS_LATE, 'Production');
309     $req->bindParam(":idProduction", $idProduction);
310     $req->execute();
311
312     $result = $req->fetch();
313     return $result;
314 }
315
316 //Fonction qui converti en Mo les Octets
317 public static function ConvertOctetsToMo($octets)
318 {
319     // 1mo = 1 048 576 octets
320     return $octets / 1000000;
321 }
322
323 //Génère un nom d'image aléatoire pour ne pas avoir les mêmes noms dans les médias.
324 // (stockés sur le serveur)
325 public static function GenerateRandomImageName()
326 {
327     $alphabet = range('a', 'z');
328     $newImageName = "";
329     for ($i = 0; $i < 26; $i++) {
330         $newImageName .= $alphabet[rand(0, 25)];
331     }
332     return $newImageName;
333 }
334
335 public static function RemoveCategorie($idCategorie)
336 {
337     $req = MonPdo::getInstance()->prepare("UPDATE productions SET categories_idCategorie = NULL WHERE categories_idCategorie =:idCategorie");
338     $req->bindParam(":idCategorie", $idCategorie);
339     $req->execute();
340 }
341
342 //Fonction qui met à jour les informations d'une production
343 public static function UpdateProductionInfos($titre, $description, $categorie,
344     $date_modification, $idProduction)
345 {
346     //Requête sql pour update l'utilisateur
347     $req = MonPdo::getInstance()->prepare("UPDATE productions SET idProduction = :idProduction, titre = :titre, description = :descriptionModif,
348     categories_idCategorie = :idCategorie, date_modification = :dateModif WHERE
349     idProduction = :idProduction");
350     $req->bindParam(":titre", $titre);
351     $req->bindParam(":descriptionModif", $description);
352     $req->bindParam(":idCategorie", $categorie);
353     $req->bindParam(":dateModif", $date_modification);
354     $req->bindParam(":idProduction", $idProduction);
```

```
352     $req->execute();
353 
354 }
355 
356 }
357 
358 // supprime la production et les likes selon son id
359 public static function DeleteProduction($idProduction)
360 {
361     $req = MonPdo::getInstance()->prepare("DELETE FROM like_unlike WHERE
362         production_idProduction = :idProduction; DELETE FROM productions WHERE
363         idProduction = :idProduction");
364     $req->bindParam(":idProduction", $idProduction);
365     $req->execute();
366 }
367 
368 public static function GetMediaNameByProductionId($idProduction)
369 {
370     $req = MonPdo::getInstance()->prepare("SELECT * FROM productions WHERE idProduction =
371         :idProduction");
372     $req->setFetchMode(PDO::FETCH_CLASS | PDO::FETCH_PROPS_LATE, 'Production');
373     $req->bindParam(":idProduction", $idProduction);
374     $req->execute();
375 
376     $result = $req->fetch();
377     return $result->getFilename();
378 }
379 }
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <?php
19 class User
20 {
21     // **** variables ****
22     private $idUser;
23     private $FirstName;
24     private $LastName;
25     private $Email;
26     private $PasswordChiffer;
27     private $UserName;
28     private $isAdmin;
29
30
31     // **** Get Set ****
32
33     /**
34      * Get the value of idUser
35      */
36     public function getIdUser()
37     {
38         return $this->idUser;
39     }
40
41     /**
42      * Set the value of idUser
43      *
44      * @return self
45      */
46     public function setIdUser($idUser)
47     {
48         $this->idUser = $idUser;
49
50         return $this;
51     }
52
53     /**
54      * Get the value of FirstName
55      */
56     public function getFirstName()
57     {
58         return $this->FirstName;
59     }
60
61     /**
62      * Set the value of FirstName
```

```
63     *
64     * @return self
65     */
66     public function setFirstName($FirstName)
67     {
68         $this->FirstName = $FirstName;
69
70         return $this;
71     }
72
73 /**
74 * Get the value of FirstName
75 */
76     public function getUserNome()
77     {
78         return $this->UserName;
79     }
80
81 /**
82 * Set the value of FirstName
83 *
84 * @return self
85 */
86     public function setUserName($UserName)
87     {
88         $this->UserName = $UserName;
89
90         return $this;
91     }
92
93 /**
94 * Get the value of LastName
95 */
96     public function getLastNome()
97     {
98         return $this->LastName;
99     }
100
101 /**
102 * Set the value of LastName
103 *
104 * @return self
105 */
106     public function setLastName($LastName)
107     {
108         $this->LastName = $LastName;
109
110         return $this;
111     }
112
113 /**
114 * Get the value of Email
115 */
116     public function getEmail()
117     {
118         return $this->Email;
119     }
120
121 /**
122 * Set the value of Email
123 *
124 * @return self
```

```
125     */
126     public function setEmail($Email)
127     {
128         $this->Email = $Email;
129
130         return $this;
131     }
132
133 /**
134 * Get the value of isAdmin
135 */
136 public function getIsAdmin()
137 {
138     return $this->isAdmin;
139 }
140
141 /**
142 * Set the value of Email
143 *
144 * @return self
145 */
146 public function setIsAdmin($isAdmin)
147 {
148     $this->isAdmin = $isAdmin;
149
150     return $this;
151 }
152
153 /**
154 * Get the value of PasswordChiffre
155 */
156 public function getPasswordChiffre()
157 {
158     return $this->PasswordChiffre;
159 }
160
161 /**
162 * Set the value of PasswordChiffre
163 *
164 * @return self
165 */
166 public function setPasswordChiffre($PasswordChiffre)
167 {
168     $this->PasswordChiffre = $PasswordChiffre;
169
170     return $this;
171 }
172
173 //Fonction qui chiffre le mot de passe en sha256
174 public static function ChiffrePassword($passwordClair)
175 {
176     return hash('sha256', $passwordClair);
177 }
178
179 //Fonction qui connecte l'utilisateur à la base de données
180 public static function ConnectUser(User $user)
181 {
182     //Récupère le mail et le mot de passe
183     $email = $user->getEmail();
184     $passwordHash = User::ChiffrePassword($user->getPasswordChiffre());
185
186     //Selectionne l'utilisateur qui va se connecter selon le mail
```

```
187     $req = MonPdo::getInstance()->prepare("SELECT * FROM utilisateurs WHERE Email
188         = :email");
189     $req->setFetchMode(PDO::FETCH_CLASS | PDO::FETCH_PROPS_LATE, 'User');
190     $req->bindParam(':email', $email);
191     $req->execute();
192     $result = $req->fetch();
193
194     if ($result == false) {
195         return false;
196     }
197
198     //Vérifie le mot de passe si c'est le même que celui entrer dans le formulaire de connexion (Compare les 2 hash, si c'est le même alors c'est bon)
199     if ($result->getPasswordChiffrer() == $passwordHash) {
200         return $result;
201     } else {
202         return false;
203     }
204
205     //Fonction qui créer un nouvel utilisateur
206     public static function CreateUser(User $user)
207     {
208         //Champs du formulaire
209         $UserName = $user->getUserName();
210         $FirstName = $user->getFirstName();
211         $LastName = $user->getLastName();
212         $Email = $user->getEmail();
213
214         //Chiffre le mot de passe en sha256
215         $PasswordChiffrer = User::ChiffrerPassword($user->getPasswordChiffrer());
216         //N'est pas administrateur par défaut (1 = User | 2 = Admin)
217         $isAdmin = 1;
218
219         //Requête pour insérer un nouvel utilisateur dans la base de données
220         $req = MonPdo::getInstance()->prepare("INSERT INTO utilisateurs(FirstName, LastName,
221             Email, PasswordChiffrer, UserName, isAdmin) VALUES
222             (:FIRSTNAME, :LASTNAME, :EMAIL, :PASSWORDCHIFFRER, :USERNAME, :ISADMIN)");
223         $req->bindParam(':FIRSTNAME', $FirstName);
224         $req->bindParam(':LASTNAME', $LastName);
225         $req->bindParam(':EMAIL', $Email);
226         $req->bindParam(':PASSWORDCHIFFRER', $PasswordChiffrer);
227         $req->bindParam(':USERNAME', $UserName);
228         $req->bindParam(':ISADMIN', $isAdmin);
229         $req->execute();
230
231     //Fonction qui vérifie la disponibilité de l'email (lors de l'inscription d'un utilisateur)
232     public static function IsEmailAvailable($email)
233     {
234         //Sélectionne les mails dans la table *Utilisateurs*
235         $req = MonPdo::getInstance()->prepare("SELECT Email FROM utilisateurs WHERE Email
236             = :email");
237         $req->setFetchMode(PDO::FETCH_CLASS | PDO::FETCH_PROPS_LATE, 'User');
238         $req->bindParam(':email', $email);
239         $req->execute();
240         $result = $req->fetch();
241
242         if ($result == false) {
243             // Si le mail existe pas, return true car il est libre
244             return true;
245         } else {
```

```
243         //Sinon, dit qu'il existe déjà
244         return false;
245     }
246 }
247
248 //Fonction qui permet de générer un nouveau mot de passe en cas d'oubli.
249 public static function RecoverPassword(User $user)
250 {
251     //Créer un mot de passe uniq (généré aléatoirement)
252     $password = uniqid();
253     //set le nouveau mot de passe générée
254     $user->setPasswordChiffrer($password);
255     //Chiffre (en SHA256) le nouveau mot de passe avant de le mettre dans la base de
256     //données
257     $PasswordChiffrer = User::ChiffrerPassword($user->getPasswordChiffrer());
258     $Id = $user->getIdUser();
259     $Email = $user->getEmail();
260
261     //Insertion dans la base du nouveau mdp
262     $req = MonPdo::getInstance()->prepare("UPDATE utilisateurs SET
263         `PasswordChiffrer`=:PASSWORDSHA256 WHERE `Email`=:EMAIL");
264     $req->bindParam(':PASSWORDSHA256', $PasswordChiffrer);
265     $req->bindParam(':EMAIL', $Email);
266     $req->execute();
267
268     return $password;
269 }
270
271 //Fonction qui trouve un utilisateur selon son id
272 public static function GetUsernameById($idUser)
273 {
274     //Sélectionne l'utilisateur selon son id
275     $req = MonPdo::getInstance()->prepare("SELECT * FROM utilisateurs WHERE idUser
276         = :idUser");
277     $req->setFetchMode(PDO::FETCH_CLASS | PDO::FETCH_PROPS_LATE, 'User');
278     $req->bindParam(":idUser", $idUser);
279     $req->execute();
280     $result = $req->fetch();
281
282     return $result->getUserName();
283 }
284
285 //Fonction qui trouve un utilisateur selon son email (Utilisé dans la page profil)
286 public static function GetUsernameByEmail($email)
287 {
288     //Sélectionne l'utilisateur selon son email
289     $req = MonPdo::getInstance()->prepare("SELECT * FROM utilisateurs WHERE `Email`=
290         :Email");
291     $req->setFetchMode(PDO::FETCH_CLASS | PDO::FETCH_PROPS_LATE, 'User');
292     $req->bindParam(":Email", $email);
293     $req->execute();
294     $result = $req->fetch();
295
296     //Retourne l'username
297     return $result->getUserName();
298 }
299
300 //Récupère tout les utilisateurs
301 public static function GetAllUsers()
302 {
303     $req = MonPdo::getInstance()->prepare("SELECT * FROM utilisateurs");
304     $req->setFetchMode(PDO::FETCH_CLASS | PDO::FETCH_PROPS_LATE, 'User');
```

```
301     $req->execute();
302     $result = $req->fetchAll();
303 
304     return $result;
305 }
306 
307 //Fonction qui met à jour les informations de l'utilisateur (page profil)
308 public static function UpdateUserInfo($UserName, $FirstName, $LastName, $Email)
309 {
310     //Récupère en session le idUser et si il est admin ou non pour ne pas écraser ce
311     //champs dans la base de données
312     $isAdmin = $_SESSION['connectedUser']['isAdmin'];
313     $idUser = $_SESSION['connectedUser']['idUser'];
314 
315     //Requête sql pour update l'utilisateur
316     $req = MonPdo::getInstance()->prepare("UPDATE utilisateurs SET username =:USERNAME,
317                                                 firstname =:FIRSTNAME, lastname =:LASTNAME, email=:EMAIL WHERE email =:EMAIL");
318     $req->bindParam(":USERNAME", $UserName);
319     $req->bindParam(":FIRSTNAME", $FirstName);
320     $req->bindParam(":LASTNAME", $LastName);
321     $req->bindParam(":EMAIL", $Email);
322 
323     if ($req->execute()) {
324         $_SESSION['connectedUser'] = [
325             'isConnected' => true,
326             'isAdmin' => $isAdmin,
327             'idUser' => $idUser,
328             'email' => $Email,
329             'firstname' => $FirstName,
330             'lastname' => $LastName,
331             'username' => $UserName,
332         ];
333     }
334 
335 //Fonction qui permet de générer un nouveau mot de passe en cas d'oubli.
336 public static function UpdateUserPassword($newPassword)
337 {
338     //Récupère le user par email
339     $email = $_SESSION['connectedUser']['email'];
340 
341     //Chiffre (en SHA256) le nouveau mot de passe avant de le mettre dans la base de
342     //données
343     $PasswordChiffrer = User::ChiffrerPassword($newPassword);
344 
345     //Insertion dans la base du nouveau mdp
346     $req = MonPdo::getInstance()->prepare("UPDATE utilisateurs SET
347         `PasswordChiffrer`=:PASSWORDSHA256 WHERE `Email`=:EMAIL");
348     $req->bindParam(':PASSWORDSHA256', $PasswordChiffrer);
349     $req->bindParam(':EMAIL', $email);
350     $req->execute();
351 
352 //Fonction qui permet de générer un nouveau mot de passe en cas d'oubli.
353 public static function UpdateUserRole($idUser, $statut)
354 {
355     //Insertion dans la base du nouveau mdp
356     $req = MonPdo::getInstance()->prepare("UPDATE utilisateurs SET `isAdmin`=:isAdmin
357         WHERE `idUser`=:idUser");
358     $req->bindParam(':isAdmin', $statut);
```

```
358     $req->bindParam(':idUser', $idUser);
359     $req->execute();
360
361 }
362 }
363 }
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <?php
19 // Controlleur admin
20
21 $action = filter_input(INPUT_GET, 'action');
22 switch ($action) {
23     //Visuel de la page de gestion des catégories
24     case 'ShowCategorieAdminPannel':
25         include 'vue/adminPannel.php';
26         break;
27
28     //Visuel de la page de gestion des utilisateurs (isAdmin)
29     case 'ShowUserAdminPannel':
30         include 'vue/gestionUserAdmin.php';
31         break;
32
33     //Affiche la page de modification de la catégorie sélectionnée.
34     case 'ShowEditCategorie':
35         include 'vue/editCategorie.php';
36         break;
37
38     //Changements des droits de l'utilisateur
39     case 'changeAdminRights':
40         $idUser = filter_input(INPUT_GET, 'idUser', FILTER_SANITIZE_NUMBER_INT);
41         $username;
42         $checkboxAdmin = filter_input(INPUT_POST, 'isAdmin');
43         if ($checkboxAdmin != null) {
44             User::UpdateUserRole($idUser, 2);
45             $_SESSION['AlertMessage'] = [
46                 'type' => "success",
47                 'message' => "Les droits d'administrateur ont bien été donnés"
48             ];
49         } else {
50             User::UpdateUserRole($idUser, 1);
51             $_SESSION['AlertMessage'] = [
52                 'type' => "success",
53                 'message' => "Les droits ont été retirés"
54             ];
55         }
56
57     //header('Location: index.php?uc=admin&action=ShowUserAdminPannel');
58     echo '<script>window.location.href = "index.php?
59         uc=admin&action=ShowUserAdminPannel";</script>';
60         break;
61
62     //Suppression d'une catégorie depuis la page admin
```

```
62     case 'deleteCategorie':
63         $idCategorie = filter_input(INPUT_GET, 'idCategorie', FILTER_SANITIZE_NUMBER_INT);
64         $nom = Categorie::GetCategorieNameById($idCategorie);
65         Production::RemoveCategorie($idCategorie);
66         Categorie::DeleteCategorie($idCategorie);
67         //Une fois la catégorie supprimée, un message de réussite s'affiche
68         $_SESSION['AlertMessage'] = [
69             'type' => "success",
70             'message' => "La Catégorie " . $nom . " a bien été supprimé de la base de
71             données."
72         ];
73         //Redirige l'administrateur sur la page de gestion
74         //header('Location: index.php?uc=admin&action=ShowCategorieAdminPannel');
75         echo '<script>window.location.href = "index.php?
76             uc=admin&action=ShowCategorieAdminPannel";</script>';
77         break;
78
79         //Ajoute une catégorie dans la base de données
80     case 'AddCategorie':
81         $CategorieName = filter_input(INPUT_POST, 'nomCategorie', FILTER_SANITIZE_STRING);
82         //Appel de la fonction d'ajout
83         Categorie::AddNewCategorie($CategorieName);
84         //Message de réussite
85         $_SESSION['AlertMessage'] = [
86             'type' => "success",
87             'message' => "La nouvelle catégorie " . $CategorieName . " a bien été ajouté
88             dans la base de données."
89         ];
89         //Redirection de l'administrateur
90         //header('Location: index.php?uc=admin&action=ShowCategorieAdminPannel');
91         echo '<script>window.location.href = "index.php?
92             uc=admin&action=ShowCategorieAdminPannel";</script>';
93         break;
94
95         //Met à jour une catégorie (modification du nom de la catégorie)
96     case 'UpdateCategorie':
97         $idCategorie = filter_input(INPUT_GET, 'idCategorie', FILTER_SANITIZE_NUMBER_INT);
98         $nomCategorie = filter_input(INPUT_POST, 'nomCategorie', FILTER_SANITIZE_STRING);
99         //Appel de la fonction de modification
100        Categorie::UpdateCategorie($idCategorie, $nomCategorie);
101        //Message de réussite
102        $_SESSION['AlertMessage'] = [
103            'type' => "success",
104            'message' => "La catégorie " . $nomCategorie . " a bien été modifiée dans la base
105            de données."
106        ];
107        //Redirection de l'administrateur
108        //header('Location: index.php?uc=admin&action=ShowCategorieAdminPannel');
109        echo '<script>window.location.href = "index.php?
110             uc=admin&action=ShowCategorieAdminPannel";</script>';
111        break;
112    }
113 }
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <?php
19 /* Controleur si tout va bien lors du post*/
20
21 $action = filter_input(INPUT_GET, 'action');
22
23 switch ($action) {
24     //Visuel de la page de Post
25     case 'ShowLoginForm':
26
27         if ($_SESSION['connectedUser']['isConnected'] == false) {
28             //Affiche le formulaire de post
29             include 'vue/login_form.php';
30         } else {
31             //header('Location: index.php');
32             echo '<script>window.location.href = "index.php";</script>';
33         }
34     break;
35
36     case 'ShowRegisterForm':
37
38         if (!isset($_SESSION["UserInfos"])) {
39             $_SESSION["UserInfos"] = [
40                 'firstname' => "",
41                 'lastname' => "",
42                 'email' => "",
43                 'password' => "",
44                 'username' => ""
45             ];
46         }
47         if ($_SESSION['connectedUser']['isConnected'] == false) {
48             //Affiche le formulaire de post
49             include 'vue/register_form.php';
50         } else {
51             //header('Location: index.php');
52             echo '<script>window.location.href = "index.php";</script>';
53         }
54     break;
55
56     case 'disconnect':
57         //Affiche le formulaire de post
58         session_destroy();
59         session_unset();
60         //header("Location:index.php");
61         echo '<script>window.location.href = "index.php";</script>';
62         break;
```

```
63
64     case 'validateLogin':
65
66         $email = filter_input(INPUT_POST, 'Email', FILTER_SANITIZE_EMAIL);
67         $password = filter_input(INPUT_POST, 'Password', FILTER_DEFAULT);
68
69         if ($email != "" && $password != "") {
70             $user = new User();
71             $user->setEmail($email)
72                 ->setPasswordChiffrer($password)
73                 ->GetUsernameByEmail($email);
74
75             $result = User::ConnectUser($user);
76             if ($result != false) {
77                 $_SESSION['connectedUser'] = [
78                     'isConnected' => true,
79                     'idUser' => $result->getIdUser(),
80                     'email' => $result->getEmail(),
81                     'firstname' => $result->getFirstName(),
82                     'lastname' => $result->getLastName(),
83                     'username' => $result->getUsername(),
84                     'password' => $result->getPasswordChiffrer(),
85                     'isAdmin' => $result->getIsAdmin()
86                 ];
87                 //header('Location: index.php');
88                 echo '<script>window.location.href = "index.php";</script>';
89             } else {
90                 $_SESSION['AlertMessage'] = [
91                     "type" => "danger",
92                     "message" => "Un des champs n'est pas correct, veuillez réessayer..."
93                 ];
94                 //header("Location: index.php?uc=login&action>ShowLoginForm");
95                 echo '<script>window.location.href = "index.php?
96                     uc=login&action>ShowLoginForm";</script>';
97                 exit();
98             }
99         } else {
100             $_SESSION['AlertMessage'] = [
101                 "type" => "danger",
102                 "message" => "Merci de remplir tout les champs."
103             ];
104             //header("Location : index.php?uc=login&action>ShowLoginForm");
105             echo '<script>window.location.href = "index.php?
106                 uc=login&action>ShowLoginForm";</script>';
107             exit();
108         }
109     break;
110
111     case 'validateRegister':
112
113         //Filtrage des données entrées dans le formulaire
114         $firstname = filter_input(INPUT_POST, 'FirstName', FILTER_SANITIZE_STRING);
115         $lastname = filter_input(INPUT_POST, 'LastName', FILTER_SANITIZE_STRING);
116         $email = filter_input(INPUT_POST, 'Email', FILTER_SANITIZE_EMAIL);
117         $password = filter_input(INPUT_POST, 'Password', FILTER_DEFAULT);
118         $username = filter_input(INPUT_POST, 'UserName', FILTER_SANITIZE_STRING);
119         $confirmpassword = filter_input(INPUT_POST, 'ConfirmPassword', FILTER_DEFAULT);
120
121         //Stock en session les infos pour faire en sorte que le formulaire soit sticky
122         $_SESSION["UserInfo"] = [
123             'firstname' => $firstname,
124             'lastname' => $lastname,
```

```
123         'email' => $email,
124         'password' => $password,
125         'username' => $username
126     ];
127
128     //Si tout les champs sont remplis alors :
129     if ($firstname != "" && $lastname != "" && $email != "" && $password != "" &&
130         $confirmPassword != "" && $username != "") {
131         if ($password == $confirmPassword) {
132             //Si le mail n'est pas déjà pris alors :
133             if (User::IsEmailAvailable($email)) {
134                 $user = new User();
135                 //Set des informations
136                 $user->setFirstName($firstname)->setLastName($lastname)->setEmail
137                     ($email)->setPasswordChiffrer($password)->setUserName($username);
138                 //Appel la fonction createUser (qui ajoute l'utilisateur en base de
139                     données)
140                 User::CreateUser($user);
141                 //header("Location:index.php?uc=login&action>ShowLoginForm");
142                 echo '<script>window.location.href = "index.php?
143                     uc=login&action>ShowLoginForm";</script>';
144                 $_SESSION['AlertMessage'] = [
145                     "type" => "success",
146                     "message" => "Vous avez bien créer votre compte, vous pouvez
147                         désormais vous connecter !"
148                 ];
149                 exit;
150             } else {
151                 //Si l'email est déjà pris alors on affiche un message d'erreur
152                 //header("Location:index.php?uc=login&action>ShowRegisterForm");
153                 echo '<script>window.location.href = "index.php?
154                     uc=login&action>ShowRegisterForm";</script>';
155                 $_SESSION['AlertMessage'] = [
156                     "type" => "danger",
157                     "message" => "L'email existe déjà ! Veuillez en choisir un autre."
158                 ];
159             }
160         } else {
161             //Si les mot de passe sont différents alors on affiche un message d'erreur
162             //header("Location:index.php?uc=login&action>ShowRegisterForm");
163             echo '<script>window.location.href = "index.php?
164                     uc=login&action>ShowRegisterForm";</script>';
165             $_SESSION['AlertMessage'] = [
166                 "type" => "danger",
167                 "message" => "Les mots de passes ne correspondent pas !"
168             ];
169             exit();
170         }
171     } else {
172         //Si un ou plusieurs champs sont vides, on affiche un message d'erreur
173         //header("Location:index.php?uc=login&action>ShowRegisterForm");
174         echo '<script>window.location.href = "index.php?
175                     uc=login&action>ShowRegisterForm";</script>';
176         $_SESSION['AlertMessage'] = [
177             "type" => "danger",
178             "message" => "Merci de remplir tout les champs."
179         ];
180         exit();
181     }
182     break;
183
184     case 'ShowProfile':
```

```
177     include 'vue/profile.php';
178
179     break;
180
181 case 'changePassword':
182     include 'vue/changePassword.php';
183     break;
184
185 case 'UpdateUserPassword':
186     //Filtrage des données qui vont être modifiées
187     $newPassword = filter_input(INPUT_POST, 'newPassword', FILTER_SANITIZE_STRING);
188     $confirmNewPassword = filter_input(INPUT_POST, 'confirmNewPassword',
189         FILTER_SANITIZE_STRING);
190     if ($newPassword == $confirmNewPassword) {
191         User::UpdateUserPassword($newPassword);
192
193         $_SESSION['AlertMessage'] = [
194             'type' => "success",
195             'message' => "Le mot de passe de " . $username . " a bien été mis à jour"
196         ];
197     } else {
198         $_SESSION['AlertMessage'] = [
199             'type' => "success",
200             'message' => "Les mot de passe ne correspondent pas !"
201         ];
202     }
203
204     //header("Location:index.php?uc=login&action>ShowProfile");
205     echo '<script>window.location.href = "index.php?uc=login&action>ShowProfile";</
206         script>';
207     exit;
208
209     break;
210
211 case 'UpdateUserInfo':
212     //Filtrage des données qui vont être modifiées
213     $firstname = filter_input(INPUT_POST, 'firstname', FILTER_SANITIZE_STRING);
214     $lastname = filter_input(INPUT_POST, 'lastname', FILTER_SANITIZE_STRING);
215     $email = filter_input(INPUT_POST, 'email', FILTER_SANITIZE_EMAIL);
216     $username = filter_input(INPUT_POST, 'username', FILTER_SANITIZE_STRING);
217     User::UpdateUserInfo($username, $firstname, $lastname, $email);
218
219     $_SESSION['AlertMessage'] = [
220         'type' => "success",
221         'message' => "Le profil de " . $username . " a bien été mis à jour"
222     ];
223     //header('Location: index.php');
224     echo '<script>window.location.href = "index.php";</script>';
225     break;
226
227 case 'randomPasswordGeneration':
228
229     $user = new User();
230     $user->setEmail($_POST['EmailRecovery']);
231     $password = User::RecoverPassword($user);
232     include 'vue/generationNouveauMdp.php';
233     break;
234
235 case 'ShowForgotPassword':
236     //Affiche la page de récupération de mot de passe
237     include 'vue/mpdOublier.php';
238     break;
```

```
237
238
239
240     case 'ShowPageFAQ':
241         //Affiche la page de FAQ (c'est le manuel utilisateur, il est inclu dans le site ↵
242         //web)
243         include 'vue/pageFAQ.php';
244         break;
245
246     default:
247         //Page d'erreur
248         include 'vue/erreur404.php';
249         break;
250 }
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <?php
19 /* Controleur si tout va bien lors du post*/
20
21 $action = filter_input(INPUT_GET, 'action');
22 switch ($action) {
23     //Visuel de la page de Post
24     case 'show':
25         //Affiche le formulaire de post
26         include 'vue/post_form.php';
27         break;
28     case 'validate':
29         //Récupération de la description et récupération du/des fichier(s)
30         $fichier = $_FILES["filesPost"];
31         $titreProduction = filter_input(INPUT_POST, 'titreProduction',
32             FILTER_SANITIZE_STRING);
32         $descriptionProduction = filter_input(INPUT_POST, 'descriptionProduction',
33             FILTER_SANITIZE_STRING);
33         $categorieProduction = filter_input(INPUT_POST, 'categorieProduction',
34             FILTER_SANITIZE_STRING);
34
35
36         //Si tout les champs sont remplis alors :
37         if ($titreProduction != "" && $descriptionProduction != "" && $categorieProduction ! =
38             "" && $fichier['name'][0] != "") {
39
40             //Vérification du type de fichier (si c'est bien une image ou non)
41             if (explode("/", $fichier['type'][0])[0] != "image") {
42                 $_SESSION['AlertMessage'] = [
43                     'type' => "danger",
44                     'message' => "Seulement les images sont supportées !"
45                 ];
46                 //header('Location: index.php?uc=post&action=show');
47                 echo '<script>window.location.href = "index.php?uc=post&action=show";</
48                     script>';
49                 exit;
50             }
51
52             $fileMo = Production::ConvertOctetsToMO($fichier['size'][0]);
53             // vérifie la taille de chaque image afin de ne pas dépasser 5 Mo
54             if ($fileMo > 5) {
55                 $_SESSION['AlertMessage'] = [
56                     'type' => "danger",
57                     'message' => "L'image ne doit pas dépasser les 5 Mo !"
58                 ];
59         }
60     }
61 }
```

```
...op\TPI_ZaiemElias\CodeSource\controllers\production_controller.php 2
58         //header('Location: index.php?uc=post&action=show');
59         echo '<script>window.location.href = "index.php?uc=post&action=show";</
60             script>';
61         exit;
62     }
63
64     //Récupère la date du jour
65     $currentDate = date("Y-m-d H:i:s");
66     $dir = "./assets/medias/";
67
68     // génère un nom aléatoire
69     $randomName = Production::GenerateRandomImageName() . "." . explode("/", 
70         $fichier['type'][0])[1];
71
72     // si le nom aléatoire existe déjà, on en regénère un
73     while (file_exists($dir . $randomName)) {
74         $randomName = Production::GenerateRandomImageName() . "." . explode("/", 
75             $fichier['type'][0])[1];
76     }
77
78     if (move_uploaded_file($fichier['tmp_name'][0], $dir . $randomName)) {
79         // on crée la production dans la base de données
80         $production = new Production();
81         $production->setTitreProduction($titreProduction)
82             ->setDescriptionProduction($descriptionProduction)
83             ->setDateSoumission($currentDate)
84             ->setDateModification($currentDate)
85             ->setFilename($randomName)
86             ->setCategories_idCategorie($categorieProduction)
87             ->setUser_idUser($_SESSION['connectedUser']['idUser']);
88         Production::AddProduction($production);
89
90         //Si le post à été créer, afficher un message de réussite.
91         $_SESSION['AlertMessage'] = [
92             'type' => "success",
93             'message' => "Vous avez posté avec succès !"
94         ];
95         //header('Location: index.php');
96         echo '<script>window.location.href = "index.php";</script>';
97         exit;
98     } else {
99         // retourne un message d'erreur si les champs ne sont pas remplis
100        $_SESSION['AlertMessage'] = [
101            'type' => "danger",
102            'message' => "L'image n'a pas pu être publié !"
103        ];
104        //header('Location: index.php?uc=post&action=show');
105        echo '<script>window.location.href = "index.php?uc=post&action=show";</
106             script>';
107        exit;
108    }
109    // retourne un message d'erreur si les champs ne sont pas remplis
110    $_SESSION['AlertMessage'] = [
111        'type' => "danger",
112        'message' => "Tout les champs doivent être remplis !"
113    ];
114    //header('Location: index.php?uc=post&action=show');
115    echo '<script>window.location.href = "index.php?uc=post&action=show";</script>';
116    exit;
117
118}
119break;
```

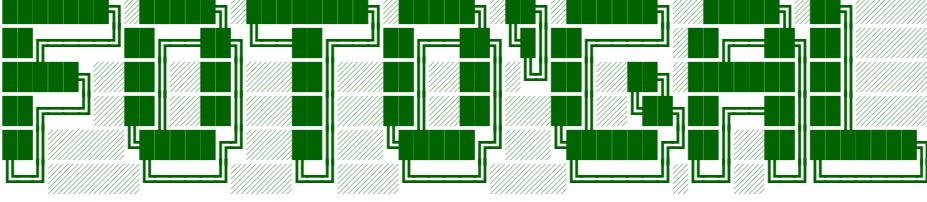
```
116
117     case 'showDetail':
118         $idPost = filter_input(INPUT_GET, 'id', FILTER_SANITIZE_NUMBER_INT);
119         $production = Production::GetProductionById($idPost);
120         include 'vue/postDetail.php';
121         break;
122
123     case 'like':
124         $idPost = filter_input(INPUT_GET, 'id', FILTER_SANITIZE_NUMBER_INT);
125         $like = new LikeUnlike();
126         $like->setLike(1)
127             ->setUtilisateurs_idUser($_SESSION['connectedUser']['idUser'])
128             ->setProduction_idProduction($idPost);
129         $like->LikePost();
130
131         $_SESSION['AlertMessage'] = [
132             'type' => "success",
133             'message' => "Vous avez liké le post"
134         ];
135         //Redirige l'utilisateur vers l'index
136         //header('Location: index.php');
137         echo '<script>window.location.href = "index.php";</script>';
138         exit;
139         break;
140
141     case 'dislike':
142         $idPost = filter_input(INPUT_GET, 'id', FILTER_SANITIZE_NUMBER_INT);
143         $like = new LikeUnlike();
144         $like->setLike(2)
145             ->setUtilisateurs_idUser($_SESSION['connectedUser']['idUser'])
146             ->setProduction_idProduction($idPost);
147         $like->LikePost();
148
149         $_SESSION['AlertMessage'] = [
150             'type' => "danger",
151             'message' => "Vous avez disliked le post"
152         ];
153         //header('Location: index.php');
154         echo '<script>window.location.href = "index.php";</script>';
155         exit;
156         break;
157
158
159     case 'editLikePost':
160         $idProduction = filter_input(INPUT_GET, 'idProduction', FILTER_SANITIZE_NUMBER_INT);
161         $numberLike = filter_input(INPUT_GET, 'like', FILTER_SANITIZE_NUMBER_INT);
162
163
164         $like = new LikeUnlike();
165         $like->setUtilisateurs_idUser($_SESSION['connectedUser']['idUser'])
166             ->setProduction_idProduction($idProduction)
167             ->setLike($numberLike);
168         $like->>EditLikePost();
169
170         //header('Location: index.php');
171         echo '<script>window.location.href = "index.php";</script>';
172         exit;
173         break;
174
175     case 'ShowMyProductions':
176         //Affiche la page des productions publiées par l'utilisateur
177         include 'vue/myProductions.php';
```

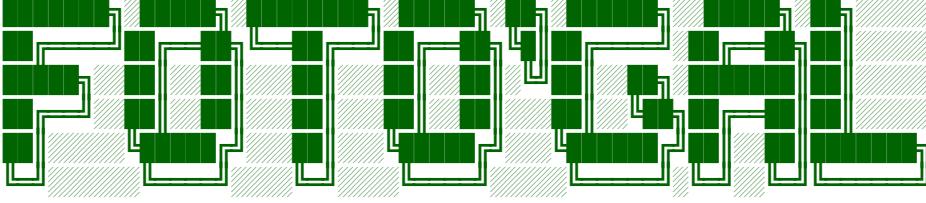
```
178         break;
179
180     //Suppression d'une production et de son média (image stockée en locale)
181     case 'deleteProduction':
182         $idProduction = filter_input(INPUT_GET, 'idProduction', FILTER_SANITIZE_NUMBER_INT);
183         //Va chercher le média dans les assets et le supprime puis redirige vers la page
184         Myproduction
185         if (unlink("./assets/medias/" . Production::GetMediaNameByProductionId
186             ($idProduction))) {
187             Production::DeleteProduction($idProduction);
188         }
189         //Message de réussie
190         $_SESSION['AlertMessage'] = [
191             'type' => "success",
192             'message' => "La production a bien été supprimée de la base de données !"
193         ];
194         //Redirige l'utilisateur puis affiche le message de réussite
195         //header('Location: index.php?uc=production&action>ShowMyProductions');
196         echo '<script>window.location.href = "index.php?
197             uc=production&action>ShowMyProductions";</script>';
198         exit;
199         break;
200
201     case 'ShowEditProductionPage':
202         $idProduction = filter_input(INPUT_GET, 'idProduction', FILTER_SANITIZE_NUMBER_INT);
203         $production = Production::GetProdutionById($idProduction);
204         //Affiche la page des productions publiées par l'utilisateur
205         include 'vue/editProduction.php';
206         break;
207
208
209     case 'editProduction':
210         $idProduction = filter_input(INPUT_GET, 'idProduction', FILTER_SANITIZE_NUMBER_INT);
211         $numberLike = filter_input(INPUT_GET, 'like', FILTER_SANITIZE_NUMBER_INT);
212
213         $like = new LikeUnlike();
214         $like->setUtilisateurs_idUser($_SESSION['connectedUser']['idUser'])
215             ->setProduction_idProduction($idProduction)
216             ->setLike($numberLike);
217         $like->>EditLikePost();
218
219         //header('Location: index.php');
220         echo '<script>window.location.href = "index.php";</script>';
221         exit;
222         break;
223
224     case 'UpdateProductionInfos':
225         //Filtrage des données qui vont être modifiées
226         $titre = filter_input(INPUT_POST, 'titre', FILTER_SANITIZE_STRING);
227         $description = filter_input(INPUT_POST, 'description', FILTER_SANITIZE_STRING);
228         $categorie = filter_input(INPUT_POST, 'categorieProduction', FILTER_SANITIZE_EMAIL);
229         $idProduction = $_GET['idProduction'];
230         $date_modification = date("Y-m-d H:i:s");
231         Production::UpdateProductionInfos($titre, $description, $categorie,
232             $date_modification, $idProduction);
233         //Message de réussie
234         $_SESSION['AlertMessage'] = [
235             'type' => "success",
236             'message' => "La production " . $titre . " a bien été modifié dans la base de
237                 données."
238         ];
239         //Redirection de l'utilisateur
```

```
235
236     //Suite a une erreur que j'ai jamais eu de ma vie, je ne peux plus utiliser les
237     //header location pour regiriger vers une page... je dois donc utiliser une autre
238     //technique vraiment moche mais qui fonctionne. Désole pour le code de mauvaise
239     //qualité
240
241     //header('Location: index.php?uc=production&action>ShowMyProductions');
242     echo '<script>window.location.href = "index.php?
243         uc=production&action>ShowMyProductions";</script>';
244     break;
245
246     default:
247         include 'vue/erreur404.php'; // affiche la page d'erreur 404 si le lien n'est pas
248         valide
249     break;
250
251 }
```

```
1 <!--
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 -->
18 <?php
19 /* Controleur si tout va bien lors du tri*/
20
21 $action = filter_input(INPUT_GET, 'action');
22 switch ($action) {
23     //Visuel de la page de tri
24     case 'ShowTriForm':
25         //Affiche la page trier.
26         include 'vue/trier_form.php';
27         break;
28
29     case 'filterCategorie':
30
31         $_SESSION['filterCategorie'] = $_POST['categorieProduction'];
32         $_SESSION['orderProduction'] = $_POST['typeTri'];
33
34         //header('Location: index.php?uc=trier&action>ShowTriForm');
35         echo '<script>window.location.href = "index.php?uc=trier&action>ShowTriForm";</
36             script>';
37         exit;
38         break;
39
40     default:
41         include 'vue/erreur404.php';
42         break;
43 }
```

```
1  /*
2
3
4
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 */
18
19 if (document.getElementById('idNumberOfPosts') != null) {
20     // appel ajax
21     var requestOptions = {
22         method: 'GET',
23         redirect: 'follow'
24     };
25     fetch("index.php?uc=getAllProductions", requestOptions).then(function(response) {
26         return response.json();
27     })
28     .then(function(myDatas) {
29         document.getElementById('idNumberOfPosts').innerText = myDatas;
30     })
31
32 }
```

```
1 /*  
2  
3  
4   
5  
6  
7  
8  
9  
10  
11  
12 Auteur : Elias Zaiem  
13 Date : 18.05.2022  
14 Projet : TPI Elias Zaiem Mai-2022  
15 Prof De TPI : Mr.Garchery  
16 Classe : I.DA-P4A  
17 */  
18  
19 .container {  
20     padding-top: 60px;  
21 }  
22  
23 @media (max-width: 979px) {  
24     .container {  
25         padding-top: 0px;  
26     }  
27 }
```

```
1  /*
2
3
4  
5
6
7
8
9
10
11
12 Auteur : Elias Zaiem
13 Date : 18.05.2022
14 Projet : TPI Elias Zaiem Mai-2022
15 Prof De TPI : Mr.Garchery
16 Classe : I.DA-P4A
17 */
18
19 html,
20 body {
21     height: 100%;
22     font-family: verdana, arial, sans-serif;
23     color: #555555;
24 }
25
26 .nav {
27     font-family: Arial, sans-serif;
28     font-size: 13px;
29 }
30
31 a {
32     color: #222222;
33 }
34
35 a:hover {
36     text-decoration: none;
37 }
38
39 hr {
40     border-color: #dedede;
41 }
42
43 .wrapper,
44 .row {
45     height: 100%;
46     margin-left: 0;
47     margin-right: 0;
48 }
49
50 .wrapper:before,
51 .wrapper:after,
52 .column:before,
53 .column:after {
54     content: "";
55     display: table;
56 }
57
58 .wrapper:after,
59 .column:after {
60     clear: both;
61 }
62
```

```
63 .column {  
64     height: 100%;  
65     overflow: auto;  
66     *zoom: 1;  
67 }  
68  
69 .column .padding {  
70     padding: 20px;  
71 }  
72  
73 .full {  
74     padding-top: 70px;  
75 }  
76  
77 .box {  
78     bottom: 0;  
79     /* increase for footer use */  
80     left: 0;  
81     position: absolute;  
82     right: 0;  
83     top: 0;  
84     background-color: #444444;  
85     /*  
86     background-image:url('/assets/example/bg_suburb.jpg');  
87     background-size:cover;  
88     background-attachment:fixed;  
89     */  
90 }  
91  
92 .divider {  
93     margin-top: 32px;  
94 }  
95  
96 .navbar-blue {  
97     border-width: 0;  
98     background-color: #3d3d3d;  
99     color: #ffffff;  
100    font-family: arial, sans-serif;  
101    top: 0;  
102    position: fixed;  
103    width: inherit;  
104 }  
105  
106 .navbar-blue li>a,  
107 .navbar-toggle {  
108     color: #efefef;  
109 }  
110  
111 .navbar-blue .dropdown-menu li a {  
112     color: #3d3d3d;  
113 }  
114  
115 .navbar-blue .dropdown-menu li>a {  
116     padding-left: 30px;  
117 }  
118  
119 .navbar-blue li>a:hover,  
120 .navbar-blue li>a:focus,  
121 .navbar-blue .open,  
122 .navbar-blue .open>a,  
123 .navbar-blue .open>a:hover,  
124 .navbar-blue .open>a:focus {
```

```
125     background-color: #666666;
126     color: #fff;
127 }
128
129 #main {
130     background-color: #e9eaed;
131     padding-left: 0;
132     padding-right: 0;
133 }
134
135 #main .img-circle {
136     margin-top: 18px;
137     height: 70px;
138     width: 70px;
139 }
140
141 #sidebar {
142     padding: 0px;
143     padding-top: 15px;
144 }
145
146 #sidebar,
147 #sidebar a,
148 #sidebar-footer a {
149     color: #ffffff;
150     background-color: transparent;
151     text-shadow: 0 0 2px #000000;
152     padding-left: 5px;
153 }
154
155 #sidebar .nav li>a:hover {
156     background-color: #7c7c7c;
157 }
158
159 .logo {
160     display: block;
161     padding: 3px;
162     background-color: #fff;
163     color: #3d3d3d;
164     height: 28px;
165     width: 28px;
166     margin: 9px;
167     margin-right: 2px;
168     margin-left: 15px;
169     font-size: 20px;
170     font-weight: 700;
171     text-align: center;
172     text-decoration: none;
173     text-shadow: 0 0 1px;
174     border-radius: 2px;
175 }
176
177 #sidebar-footer {
178     background-color: #444;
179     position: absolute;
180     bottom: 5px;
181     padding: 5px;
182 }
183
184 #footer {
185     margin-bottom: 20px;
186 }
```

```
187
188
189 /* bootstrap overrides */
190
191 h1,
192 h2,
193 h3 {
194     font-weight: 800;
195 }
196
197 .navbar-toggle,
198 .close {
199     outline: 0;
200 }
201
202 .navbar-toggle .icon-bar {
203     background-color: #fff;
204 }
205
206 .btn-primary,
207 .label-primary,
208 .list-group-item.active,
209 .list-group-item.active:hover,
210 .list-group-item.active:focus {
211     background-color: #6e6e6e;
212     color: #fffffe;
213 }
214
215 .btn-default {
216     color: #666666;
217     text-shadow: 0 0 1px rgba(0, 0, 0, .3);
218 }
219
220 .form-control {}
221
222 .panel textarea,
223 .well textarea,
224 textarea.form-control {
225     resize: none;
226 }
227
228 .badge {
229     color: #a0a0a0;
230     background-color: #fff;
231 }
232
233 .badge:hover,
234 .badge-inverse {
235     background-color: #a0a0a0;
236     color: #fff;
237 }
238
239 .jumbotron {
240     background-color: transparent;
241 }
242
243 .label-default {
244     background-color: #dddddd;
245 }
246
247 .page-header {
248     margin-top: 55px;
```

```
249     padding-top: 9px;
250     border-top: 1px solid #eeeeee;
251     font-weight: 700;
252     text-transform: uppercase;
253     letter-spacing: 2px;
254 }
255
256 .panel-default .panel-heading {
257     background-color: #f9fafb;
258     color: #555555;
259 }
260
261 .col-sm-9.full {
262     width: 100%;
263 }
264
265 .modal-header,
266 .modal-footer {
267     background-color: #f2f2f2;
268     font-weight: 800;
269     font-size: 12px;
270 }
271
272 .modal-footer i,
273 .well i {
274     font-size: 20px;
275     color: #c0c0c0;
276 }
277
278 .modal-body {
279     padding: 0px;
280 }
281
282 .modal-body textarea.form-control {
283     resize: none;
284     border: 0;
285     box-shadow: 0 0 0;
286 }
287
288 small.text-muted {
289     font-family: courier, courier-new, monospace;
290 }
291
292
293 /* adjust the contents on smaller devices */
294
295 @media (max-width: 768px) {
296     .column .padding {
297         padding: 7px;
298     }
299     .full {
300         padding-top: 20px;
301     }
302     .navbar-blue {
303         background-color: #8b8b8b;
304         top: 0;
305         width: 100%;
306         position: relative;
307     }
308 }
309
310 @media (max-width: 1000px) {
```

```
311     #idLabelPrenom {  
312         margin-top: 50px;  
313     }  
314     #idLabelEmail {  
315         margin-top: 50px;  
316     }  
317     #ImageDetails {  
318         margin-top: 50px;  
319     }  
320 }  
321  
322  
323 /*  
324 * off canvas sidebar  
325 * -----  
326 */  
327  
328 @media screen and (max-width: 768px) {  
329     .row-offcanvas {  
330         position: relative;  
331         -webkit-transition: all 0.25s ease-out;  
332         -moz-transition: all 0.25s ease-out;  
333         transition: all 0.25s ease-out;  
334     }  
335     .row-offcanvas-left.active {  
336         left: 33%;  
337     }  
338     .row-offcanvas-left.active .sidebar-offcanvas {  
339         left: -33%;  
340         position: absolute;  
341         top: 0;  
342         width: 33%;  
343         margin-left: 5px;  
344     }  
345     #sidebar,  
346     #sidebar a,  
347     #sidebar-footer a {  
348         padding-left: 3px;  
349     }  
350 }
```