

```
1 <?php
2 if (empty($_SESSION)) {
3     session_start();
4 }
5 ?>
6 <nav class="navbar navbar-expand-lg navbar-dark bg-light" id="navbarMain">
7     <div class="container-fluid">
8         <a class="navbar-brand" href="index.php">
9             
10        </a>
11        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-
bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
12            <span class="navbar-toggler-icon"></span>
13        </button>
14        <div class="collapse navbar-collapse" id="navbarSupportedContent">
15            <ul class="navbar-nav me-auto mb-2 mb-lg-0">
16                <li class="nav-item">
17                    <a class="nav-link active" aria-current="page"
href="index.php">Accueil</a>
18                </li>
19                <?php
20                if (isset($_SESSION["nomLog"])) {
21                    echo '<li class="nav-item">
22                        <a class="nav-link active" aria-current="page"
href="rechercheJeu.php">Jeux</a>
23                    </li>;
24                    echo '<li class="nav-item">
25                        <a class="nav-link active" aria-current="page"
href="creerJeu.php">Ajouter un jeu</a>
26                    </li>;
27                }
28                if ($_SESSION["statutLog"] == "6") {
29                    echo '
30                    <li class="nav-item">
31                        <a class="nav-link active" aria-current="page"
href="admin.php">Gérer les utilisateurs</a>
32                    </li>
33                    ';
34                }
35                ?>
36
37                <li class="nav-item dropdown">
38                    <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown"
role="button" data-bs-toggle="dropdown" aria-expanded="false">
39                        <?php //Mon compte
40                        //$_SESSION["nomLog"]="Bar";
41                        if (isset($_SESSION["nomLog"])) {
42                            echo $_SESSION["nomLog"];
43                        } else {
44                            echo 'Login';
45                        }
46                        ?>
47                    </a>
48                    <ul class="dropdown-menu" aria-labelledby="navbarDropdown">
49                        <?php
50                        if (isset($_SESSION["nomLog"])) {
51                            // echo '<li class="nav-item">
```

```
52 //      <a class="nav-link active" aria-current="page"
href="creerJeu.php">Ajouter un jeu</a>
53 // </li>';
54 echo '<li><a class="dropdown-item"
href="pageUtilisateur.php">Mon compte</a></li>';
55 echo '<li>
56 <hr class="dropdown-divider">
57 </li>';
58 echo '<li><a class="dropdown-item"
href="deconnexion.php">Déconnexion</a></li>';
59 } else {
60 echo '<li><a class="dropdown-item" href="login.php">Se
connecter</a></li>';
61 echo '<li><a class="dropdown-item"
href="enregistrer.php">Créer un compte</a></li>';
62 }
63 ?>
64 </ul>
65 </li>
66 </ul>
67 </div>
68 </div>
69 </nav>
```

```
1 <?php
2 if (empty($_SESSION)) {
3     session_start();
4 }?>
5 <!DOCTYPE html>
6 <html lang="fr">
7
8 <head>
9     <meta charset="UTF-8">
10    <meta http-equiv="X-UA-Compatible" content="IE=edge">
11    <meta name="viewport" content="width=device-width, initial-scale=1.0">
12    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
13    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYs0g+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>
14    <link rel="stylesheet" href="style/style.css">
15    <link rel="icon" href="image/favicon.ico" type="image/x-icon">
16    <title>PNJ</title>
17
18 </head>
19
20 <body>
21     <?php include("navbar.php");
22     if (isset($_SESSION['statutLog'])) {
23         require_once("../fonctionPHP/fonctionJeu.php");
24         echo '<ul class="list-group">';
25         echo '<h1>Tout mes jeux</h1>';
26         $data = listeJeuUtilisateur($_SESSION['nomLog']);
27         echo '</ul>';
28         <form action="jeuInfo.php" method="GET">
29             ';
30         for ($i = 0; $i < sizeof($data); $i++) {
31             echo '
32                 <div class="card">
33                 <div class="card-header">
34                     ' . $data[$i]['titre'] . '
35                 </div>
36                 <div class="card-body">
37                 
38                 <p class="card-text"> date de sortie : ' . $data[$i]['dateSortie'] .
'<br> catégorie : ' . $data[$i]['categorie'] . '</p>
39                 <button class="btn btn-primary" name="nomJeu" value="' . $data[$i]
['titre'] . ''' type="submit">Voir</button>
40                 </div>
41                 </div>
42             ';
43         }
44     } else {
45         ?>
46         <div class="container" id="menu">
47             
48             <div class="row align-items-start">
49
```

```
50         <div class="col">
51             <a href="login.php" class="btn btn-primary btn-lg">Se
connecter</a>
52         </div>
53         <div class="col">
54             <a href="enregistrer.php" class="btn btn-primary btn-lg">Créer un
compte</a>
55         </div>
56     </div>
57     <?php
58     }
59     ?>
60 </body>
61
62 </html>
```

```

1 <?php
2 if(isset($_POST['log'])){
3     $mail = filter_input(INPUT_POST, "mail", FILTER_VALIDATE_EMAIL);
4     $mdp = htmlspecialchars($_POST['mdp'], ENT_NOQUOTES);
5     require("../fonctionPHP/fonctionUtilisateur.php");
6     $mdp=hash('sha256',$mdp);
7     if(connecterUtilisateur($mail,$mdp)==true){
8         header("Location: index.php");
9     }else{
10         echo"Le compte n'existe pas ou n'est pas activé";
11     }
12 }
13 ?>
14 <!DOCTYPE html>
15 <html lang="en">
16 <head>
17     <meta charset="UTF-8">
18     <meta http-equiv="X-UA-Compatible" content="IE=edge">
19     <meta name="viewport" content="width=device-width, initial-scale=1.0">
20     <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
21     <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>
22     <link rel="stylesheet" href="style/style.css">
23     <title>Se connecter</title>
24 </head>
25 <body>
26 <?php include("navbar.php") ?>
27 <form action="#" method="post" id="login">
28     <div class="form-group">
29         <label for="mail">Adresse Email</label>
30         <input type="email" class="form-control" id="mail" name="mail" aria-
describedby="emailHelp" placeholder="Mon adresse Email">
31         <!-- <small id="emailHelp" class="form-text text-muted">We'll never share your
email with anyone else.</small> -->
32     </div>
33     <div class="form-group">
34         <label for="mdp">Mot de passe</label>
35         <input type="password" class="form-control" id="mdp" name="mdp"
placeholder="Mon mot de passe">
36     </div>
37     <div class="form-group">
38         <button type="submit" name="log" class="btn btn-primary">Se connecter</button>
<br>
39         <a href="enregistrer.php">Je n'ai pas de compte</a>
40     </div>
41 </form>
42 </body>
43 </html>

```

```

1 <?php
2 if (isset($_POST['Creer'])) {
3     $prenom = htmlspecialchars($_POST["prenom"], ENT_NOQUOTES);
4     $nom = htmlspecialchars($_POST["nom"], ENT_NOQUOTES);
5     $pseudo = htmlspecialchars($_POST["pseudo"], ENT_NOQUOTES);
6     $mail = htmlspecialchars($_POST["mail"], ENT_NOQUOTES);
7     $mdp = htmlspecialchars($_POST["mdp"], ENT_NOQUOTES);
8     $m2p = htmlspecialchars($_POST["m2p"], ENT_NOQUOTES);
9     //$mdp = htmlspecialchars($_POST['mdp'], ENT_NOQUOTES);
10
11     if ($mdp == $m2p) {
12         $mdp = hash('sha256', $mdp);
13         //image
14         $data = file_get_contents($_FILES['img']['tmp_name']);
15         $finfo = new finfo(FILEINFO_MIME_TYPE);
16         $mime = $finfo->file($_FILES['img']['tmp_name']);
17         $src = 'data:' . $mime . ';base64,' . base64_encode($data);
18
19         require("../fonctionPHP/fonctionUtilisateur.php");
20         if (creerUtilisateur($pseudo, $mdp, $prenom, $nom, $mail, $src) == true) {
21             header("Location: login.php");
22         }
23     }
24 }
25 ?>
26 <!DOCTYPE html>
27 <html lang="en">
28
29 <head>
30     <meta charset="UTF-8">
31     <meta http-equiv="X-UA-Compatible" content="IE=edge">
32     <meta name="viewport" content="width=device-width, initial-scale=1.0">
33     <link
34 href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
35 rel="stylesheet" integrity="sha384-
36 1BmE4kWBq78iYhF1dvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
37 crossorigin="anonymous">
38     <script
39 src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
40 integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYs0g+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
41 crossorigin="anonymous"></script>
42     <link rel="stylesheet" href="style/style.css">
43     <script src="./js/script.js"></script>
44     <title>Créer un compte</title>
45 </head>
46
47 <body>
48     <?php include("navbar.php") ?>
49     <form action="#" method="post" class="creer" id="login" enctype="multipart/form-
50 data">
51         <div class="form-group">
52             <label for="prenom">Prénom</label>
53             <input type="text" class="form-control" id="prenom" name="prenom"
54 placeholder="Mike" required>
55         </div>
56         <div class="form-group">
57             <label for="nom">Nom</label>
58             <input type="text" class="form-control" id="nom" name="nom"
59 placeholder="Oxlong" required>

```

```
50     </div>
51     <div class="form-group">
52         <label for="pseudo">Pseudo</label>
53         <input type="text" class="form-control" id="pseudo" name="pseudo"
placeholder="sylvain_Dudrift" required>
54     </div>
55     <div class="form-group">
56         <label for="mail">Adresse Email</label>
57         <input type="email" class="form-control" id="mail" name="mail" aria-
describedby="emailHelp" placeholder="Mike.oxlong@chungus.com" required>
58     </div>
59     <div class="form-group">
60         <label for="mdp">Mot de passe</label>
61         <input type="password" class="form-control" id="mdp" name="mdp"
placeholder="Mot de passe" required>
62     </div>
63     <div class="form-group">
64         <label for="m2p">Répéter le mot de passe</label>
65         <input type="password" class="form-control" id="m2p" name="m2p" required>
66     </div>
67     <div class="form-group">
68         <label for="img">Photo de profile</label>
69         <input type="file" class="form-control" name="img" accept="image/png,
image/gif, image/jpeg" id="img">
70     </div>
71     <button type="submit" name="Creer" class="btn btn-primary">Créer mon
compte</button>
72     <button class="btn btn-primary" onclick="effacerForm()">Effacer</button>
73     <a href="login.php">J'ai déjà un compte</a>
74 </form>
75 </body>
76
77 </html>
```

```
1 <?php
2 require_once("../fonctionPHP/fonctionUtilisateur.php");
3 require_once("../fonctionPHP/fonctionCommentaire.php");
4 if (isset($_POST['bannirUtilisateur'])) {
5     bannirUtilisateur($_POST['bannirUtilisateur']);
6 } else if (isset($_POST['valider'])) {
7     validerUtilisateur($_POST['valider']);
8 } else if (isset($_POST['refuser'])) {
9     refuserUtilisateur($_POST['refuser']);
10 } else if (isset($_POST['validerCom'])) {
11     validerCommentaire($_POST['validerCom']);
12 }
13 ?>
14 <!DOCTYPE html>
15 <html lang="en">
16
17 <head>
18     <meta charset="UTF-8">
19     <meta http-equiv="X-UA-Compatible" content="IE=edge">
20     <meta name="viewport" content="width=device-width, initial-scale=1.0">
21     <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
22     <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYs0g+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>
23     <link rel="stylesheet" href="style/style.css">
24     <title>Administration</title>
25 </head>
26
27 <body>
28     <?php
29     include("navbar.php");
30     $nonInscrit = infoNonVerifier();
31     $toutUtilisateur = infoToutUtilisateur();
32     $toutCommentaire = infoCommentaire();
33     ?>
34     <div class="container">
35         <div class="row align-items-start">
36             <div class="col">
37                 <h2>les non validés</h2>
38                 <form action="#" name="testBan" method="post">
39                     <ul class="list-group">
40                         <?php for ($i = 0; $i < sizeof($nonInscrit); $i++) {
41                             echo '
42                             <li class="list-group-item">
43                                 <button type="submit" value="' . $nonInscrit[$i]["pseudo"] . '"
name="valider">Valider le compte de ' . $nonInscrit[$i]["pseudo"] . '</button>
44                                 <button type="submit" value="' . $nonInscrit[$i]["pseudo"] . '"
name="refuser">Refuser le compte de ' . $nonInscrit[$i]["pseudo"] . '</button>
45                             </li>';
46                         }
47                         ?>
48                     </ul>
49                 </form>
50             </div>
```



```
51     <div class="col">
52         <h2>tout les utilisateurs</h2>
53         <form action="#" name="testValider" method="post">
54             <ul class="list-group">
55                 <?php for ($i = 0; $i < sizeof($toutUtilisateur); $i++) {
56                     echo '
57                 <li class="list-group-item">
58                     <button type="submit" value="" . $toutUtilisateur[$i]["pseudo"] .
59                     '"" name="bannirUtilisateur">Bannir le compte de ' . $toutUtilisateur[$i]["pseudo"] .
60                     '</button>
61                 </li>';
62             }
63             <?>
64         </ul>
65     </form>
66 </div>
67 <div class="col">
68     <h2>critiques non validés</h2>
69     <form action="#" name="testValider" method="post">
70         <ul class="list-group">
71             <?php for ($i = 0; $i < sizeof($toutCommentaire); $i++) {
72                 echo '
73                 <div data-bs-spy="scroll" data-bs-target="#navbar-example2" data-bs-
74                 offset="0" id="scrollComAdmin" class="scrollspy-example" tabindex="0">
75                 <li class="list-group-item">
76                     <p>Auteur : ' . $toutCommentaire[$i]["pseudo"] . '</p>
77                     <p>Jeux : ' . $toutCommentaire[$i]["titre"] . '</p>
78                     <p>Commentaire : ' . $toutCommentaire[$i]["commentaire"] . '</p>
79                     <button type="submit" value="" . $toutCommentaire[$i]
80                     ["idCommentaire"] . '"" name="validerCom">Valider le commentaire de ' .
81                     $toutCommentaire[$i]["pseudo"] . ' sur ' . $toutCommentaire[$i]["titre"] . '</button>
82                 </li> </div>';
83             }
84             <?>
85         </ul>
86     </form>
87 </div>
88 </div>
89 </body>
90 </html>
```

```

1
2
3 <?php
4 if (empty($_SESSION)) {
5     session_start();
6 }
7 require_once("../fonctionPHP/fonctionJeu.php");
8 if (isset($_POST['CreerJeu'])) {
9     $titre = htmlspecialchars($_POST['titre'], ENT_NOQUOTES);
10    $editeur = htmlspecialchars($_POST["editeur"], ENT_NOQUOTES);
11    $dateSortie = htmlspecialchars($_POST["dateSortie"], ENT_NOQUOTES);
12    $description = htmlspecialchars($_POST["description"], ENT_NOQUOTES);
13    $categorie = htmlspecialchars($_POST["categorie"], ENT_NOQUOTES);
14    $public = true;
15    if ($_POST['public'] == "on") {
16        $public = false;
17    }
18    $data = file_get_contents($_FILES['photo']['tmp_name']);
19    $finfo = new finfo(FILEINFO_MIME_TYPE);
20    $mime = $finfo->file($_FILES['photo']['tmp_name']);
21    $src = 'data:' . $mime . ';base64,' . base64_encode($data);
22    creerJeu($titre, $dateSortie, $editeur, $description, $public, $categorie, $src,
    $_SESSION['nomLog']);
23 }
24 ?>
25 <!DOCTYPE html>
26 <html lang="en">
27
28 <head>
29     <meta charset="UTF-8">
30     <meta http-equiv="X-UA-Compatible" content="IE=edge">
31     <meta name="viewport" content="width=device-width, initial-scale=1.0">
32     <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhF1dvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
33     <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYs0g+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>
34     <link rel="stylesheet" href="style/style.css">
35     <script src="./js/script.js"></script>
36     <title>Ajouter un jeu</title>
37 </head>
38
39 <body>
40     <?php
41     include("navbar.php");
42     ?>
43     <form action="#" method="post" id="login" enctype="multipart/form-data">
44         <div class="form-group">
45             <label for="titre">Titre</label>
46             <input type="text" class="form-control" id="titre" name="titre"
placeholder="Stardew Valley" required>
47         </div>
48
49         <div class="form-group">
50             <label for="editeur">Editeur</label>

```

```
51         <input type="text" class="form-control" id="editeur" name="editeur"
placeholder="ConcernedApe" required>
52     </div>
53     <div class="form-group">
54         <label for="dateSortie">Date de sortie</label>
55         <input type="date" class="form-control" id="dateSortie" name="dateSortie"
required>
56     </div>
57
58     <div class="form-floating">
59         <textarea class="form-control" name="description" id="floatingTextarea"
required></textarea>
60         <label for="floatingTextarea">Résumé du jeu</label>
61     </div>
62     <select class="form-select" name="categorie" aria-label="Default select
example" required>
63         <?php
64             $data = infoCategorie();
65             for ($i = 0; $i < sizeof($data); $i++) {
66                 echo '<option value="' . $data[$i]["idCategorie"] . '" selected>' .
$data[$i]["categorie"] . '</option>';
67             }
68             ?>
69             <!-- // récupérer les catégories sur la base -->
70     </select>
71     <div class="form-group">
72         <label for="photo">Image du jeu</label>
73         <input type="file" class="form-control" name="photo" accept="image/*/"
id="photo">
74     </div>
75
76     <input class="form-check-input" name="public" type="checkbox"
id="flexCheckDefault">
77     <label class="form-check-label" for="flexCheckDefault">
78         Rendre le jeu privé ?
79     </label>
80
81     <div class="form-group">
82         <button type="submit" name="CreerJeu" class="btn btn-primary">Créer le
jeu</button>
83         <button class="btn btn-primary" onclick="effacerForm()">Effacer</button>
84     </div>
85
86 </form>
87 </body>
88
89 </html>
```

```

1 <?php
2 require_once("../fonctionPHP/fonctionJeux.php");
3
4 if (isset($_GET['recherche'])) {
5     $titre = htmlspecialchars($_GET['titreJeu'], ENT_NOQUOTES);
6     $categorie = htmlspecialchars($_GET['categorie'], ENT_NOQUOTES);
7     $annee = htmlspecialchars($_GET['anneeJeu'], ENT_NOQUOTES);
8     $moyenne = htmlspecialchars($_GET['noteJeu'], ENT_NOQUOTES);
9     $auteur = htmlspecialchars($_GET['auteur'], ENT_NOQUOTES);
10
11     $data = rechercheJeu($titre, $categorie, $annee, $moyenne, $auteur);
12     $legend="Résultat de la recherche";
13 } else {
14     $data = recupererJeux();
15     $legend="Derniers jeux ajoutés";
16 }
17 ?>
18 <!DOCTYPE html>
19 <html lang="en">
20
21 <head>
22     <meta charset="UTF-8">
23     <meta http-equiv="X-UA-Compatible" content="IE=edge">
24     <meta name="viewport" content="width=device-width, initial-scale=1.0">
25     <link
26 href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
27 rel="stylesheet" integrity="sha384-
28 1BmE4kWBq78iYhFtdvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
29 crossorigin="anonymous">
30     <script
31 src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
32 integrity="sha384-ka7Sk0GlIn4gmtz2MlQnikT1wXgYs0g+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
33 crossorigin="anonymous"></script>
34     <link rel="stylesheet" href="style/style.css">
35     <script src="./js/script.js"></script>
36     <title>Recherche</title>
37 </head>
38
39 <body>
40     <?php include("navbar.php") ?>
41     <div class="container">
42         <div class="row align-items-start">
43             <div class="col">
44                 <legend><?=$legend?></legend>
45                 <div data-bs-spy="scroll" data-bs-target="#navbar-example2" data-bs-
46 offset="0" id="scrollTest" class="scrollspy-example" tabindex="0">
47
48                 <?php
49
50                     for ($i = 0; $i < sizeof($data); $i++) {
51                         echo '
52                             <div class="card" id="carteRecherche" style="width: 30rem;">
53                                 <img src="" . $data[$i]['imageJeu'] . "" id="imgrecherche"
54 class="card-img-top" alt="...">
55                                 <div class="card-body">
56                                     <p class="card-text">' . $data[$i]['titre'] . '</p>
57                                     <p class="card-text"> Note des utilisateurs : ' . $data[$i]
58 ['appreciation'] . '/10</p>
59                                     <form action="jeuInfo.php" method="GET">

```

```

50         <button class="btn btn-primary" name="nomJeu" value="" .
$data[$i]['titre'] . ' type="submit">Voir</button>
51     </form>
52 </div>
53 </div>
54     ';
55     }
56
57     ?>
58 </div>
59 <!-- <div class="card" style="width: 18rem;">
60     
61     <div class="card-body">
62         <p class="card-text">Some quick example text to build on the
card title and make up the bulk of the card's content.</p>
63         <form action="jeuInfo.php" method="GET">
64             <button class="btn btn-primary" name="nomJeu"
value="Minecraft" type="submit">Voir</button>
65         </form>
66     </div>
67 </div> -->
68 </div>
69 <div class="col">
70     <fieldset>
71         <legend>Options de recherche</legend>
72
73         <form action="" method="GET" id="recherche">
74             <div class="row g-3 align-items-center">
75                 <div class="col-auto">
76                     <label for="titreJeu" class="col-form-label">Titre
du Jeu :</label>
77                 </div>
78                 <div class="col-auto">
79                     <input type="text" id="titreJeu" name="titreJeu"
class="form-control">
80                 </div>
81             </div>
82             <div class="row g-3 align-items-center">
83                 <div class="col-auto">
84                     <label for="auteur" class="col-form-label">Auteur :
</label>
85                 </div>
86                 <div class="col-auto">
87                     <input type="text" id="auteur" name="auteur"
class="form-control">
88                 </div>
89             </div>
90
91             <div class="row g-3 align-items-center">
92                 <div class="col-auto">
93                     <label for="categorie" class="col-form-
label">Catégorie :</label>
94                 </div>
95                 <div class="col-auto">
96                     <select class="form-select" name="categorie" aria-
label="Default select example">
97                         <option value="" selected>Tout</option>
98                     <?php

```

```
99         $data = infoCategorie();
100         for ($i = 0; $i < sizeof($data); $i++) {
101             echo '<option value="' . $data[$i]
["idCategorie"] . '>' . $data[$i]["categorie"] . '</option>';
102         }
103     ?>
104     </select>
105     </div>
106 </div>
107 <div class="row g-3 align-items-center">
108     <div class="col-auto">
109         <label for="anneeJeu" class="col-form-label">Année
de sortie :</label>
110     </div>
111     <div class="col-auto">
112         <input type="number" name="anneeJeu" id="anneeJeu"
class="form-control">
113     </div>
114 </div>
115 <div class="row g-3 align-items-center">
116     <div class="col-auto">
117         <label for="noteJeu" class="col-form-label">Note
minimal :</label>
118     </div>
119     <div class="col-auto">
120         <input type="number" id="noteJeu" name="noteJeu"
class="form-control">
121     </div>
122 </div>
123 <button type="submit" name="recherche" class="btn btn-
primary">Rechercher</button>
124 </form>
125 <button class="btn btn-primary"
onclick="effacerRecherche()">Effacer</button>
126 </fieldset>
127 </div>
128 </div>
129 </div>
130 </body>
131
132 </html>
```

```
1 <?php
2 if (empty($_SESSION)) {
3     session_start();
4 }
5 if (isset($_GET['nomJeu'])) {
6     require_once('./fonctionPHP/fonctionJeu.php');
7     require_once('./fonctionPHP/fonctionNote.php');
8     require_once("./fonctionPHP/fonctionCommentaire.php");
9     $nomJeu = htmlspecialchars($_GET['nomJeu'], ENT_NOQUOTES);
10
11
12     if (isset($_POST['commentaire'])) {
13         $critique = htmlspecialchars($_POST['critique'], ENT_NOQUOTES);
14         creerCommentaire($_SESSION['nomLog'], $nomJeu, $critique, false);
15     }
16     if (isset($_POST['note'])) {
17         if (ajouterNote($_POST['note'], $_SESSION['nomLog'], $nomJeu) != false) {
18             $moyenne = calculerNoteMoyenne($nomJeu);
19             attribuerMoyenne($moyenne, $nomJeu);
20         }
21     }
22     if(isset($_POST["btn_modif"])){
23         $n_critique = htmlspecialchars($_POST['comm_modif'], ENT_NOQUOTES);
24         modifierCommentaire($n_critique,$_POST["btn_modif"]);
25     }
26     if(isset($_POST["btn_suppr"])){
27         supprimerCommentaire($_POST["btn_suppr"]);
28     }
29
30     $data = infoJeu($nomJeu);
31 } else {
32     header("Location: index.php");
33 }
34 $test = dejaNote($_SESSION['nomLog'], $nomJeu);
35 ?>
36 <!DOCTYPE html>
37 <html lang="en">
38
39 <head>
40     <meta charset="UTF-8">
41     <meta http-equiv="X-UA-Compatible" content="IE=edge">
42     <meta name="viewport" content="width=device-width, initial-scale=1.0">
43     <link
44 href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
45 rel="stylesheet" integrity="sha384-
46 1BmE4kWbQ78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
47 crossorigin="anonymous">
48     <script
49 src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
50 integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYs0g+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
51 crossorigin="anonymous"></script>
52     <link rel="stylesheet" href="style/style.css">
53     <title>Document</title>
54 </head>
55
56 <body>
57     <?php include("navbar.php");
58     ?>
59     <div class="container py-4">
```

```

53     <!-- <header class="pb-3 mb-4 border-bottom">
54     <a href="index.php" class="d-flex align-items-center text-dark text-
decoration-none">
55         <span class="fs-4">Retour au Menu</span>
56     </a> -->
57     </header>
58     <div class="p-5 mb-4 bg-light rounded-3">
59         <div class="container-fluid py-5">
60             <h1 class="display-5 fw-bold"><?php echo ($nomJeu) ?></h1>
61             <p class="col-md-8 fs-4">
62                 <?= "développé par : " . $data[0]['editeur'] ?>
63             </p>
64             <p class="col-md-8 fs-4">
65                 <?= "Sortie le : " . $data[0]['dateSortie'] ?>
66             </p>
67             <p class="col-md-8 fs-4">
68                 <?= "Note des utilisateur : " . $data[0]['appreciation'] .
'/10' ?>
69             </p>
70             <div class="row align-items-md-stretch">
71                 <?php echo '
72 <img class="imageJeu" src="" . ($data[0]['imageJeu']) . "alt=...">
73 ' ?>
74                 <span class="desc"><?= $data[0]['description'] ?></span>
75                 <!-- <div class="box">
76                     
77                     <span style=""></span>
78                 </div> -->
79             </div>
80         </div>
81     </div>
82     <div class="row align-items-md-stretch">
83         <div class="col-md-6">
84             <div class="h-100 p-5 text-white bg-dark rounded-3">
85                 <h2>Genres</h2>
86                 <ul class="listeP">
87                     <li><?= $data[0]['categorie'] ?></li>
88                 </ul>
89                 <!--<button class="btn btn-outline-secondary"
type="button">Example button</button>-->
90             </div>
91         </div>
92         <div class="col-md-6">
93             <?php
94             if ($test == true) {
95                 <?>
96                 <form action="" method="POST">
97                     <input type="number" name="note" id="note" min="0" max="10"
value="5">
98                     <button type="submit" name="noteJeu">Noter le jeu</button>
99                 </form>
100             <?php } else {
101                 echo 'Vous avez déjà attribué une note à ce jeu';
102             } ?>
103         </div>
104     <form action="#" method="POST">
105         <div class="card">
106             <div class="card-body">
107                 <div class="form-floating">

```



```

108         <p>Rédiger votre critique</p>
109         <textarea class="form-control" name="critique"
placeholder="Leave a comment here" id="floatingTextarea"></textarea>
110     </div>
111     <button type="submit" name="commentaire" class="btn btn-
primary">Envoyer le commentaire</button>
112 </div>
113 </div>
114
115 </form>
116
117 </div>
118 <div class="row align-items-md-stretch">
119     <?php
120     $commentaires = infoJeuCommentaire($nomJeu);
121     for ($i = 0; $i < sizeof($commentaires); $i++) {
122         if ($commentaires[$i]['pseudo'] == $_SESSION['nomLog']) {
123             echo '<li class="list-group-item" id="topique">
124                 <div class="card-body">
125                     <h5>' . $commentaires[$i]['pseudo'] . '</h5>
126                     <h5>' . explode(' ', $commentaires[$i]['dateCommentaire'])
[0] . '</h5>
127                     <img src="" . $commentaires[$i]['photoDeProfil'] . '' alt=""
class="imgP">
128                     <form action="" method="POST">
129                         <input type="hidden" value="" . $commentaires[$i]
["commentaire"] . '' id="commValue">
130                         <textarea class="form-control" name="comm_modif"
id="comm_modif" id="floatingTextarea" ></textarea> <br>
131                         <button class="btn btn-primary" name="btn_modif"
value="" . $commentaires[$i]["idCommentaire"] . '' type="submit">Modifier mon
commentaire</button>
132                         <button class="btn btn-primary" name="btn_supp"
value="" . $commentaires[$i]["idCommentaire"] . '' type="submit">Supprimer mon
commentaire</button>
133                     </form>
134                 </div>
135             </li>';
136         } else {
137             echo '<li class="list-group-item" id="topique">
138                 <div class="card-body">
139                     <h5>' . $commentaires[$i]['pseudo'] . '</h5>
140                     <h5>' . explode(' ', $commentaires[$i]['dateCommentaire'])
[0] . '</h5>
141                     <img src="" . $commentaires[$i]['photoDeProfil'] . '' alt=""
class="imgP">
142                     <p>' . $commentaires[$i]['commentaire'] . '</p>
143                 </div>
144             </li>';
145         }
146     }
147     ?>
148 </div>
149 <script>
150     document.getElementById("comm_modif").value=document.getElementById("commValue").va
lue
151 </script>
152 </body>
153

```

154 </html>

```
1 <?php
2 if (empty($_SESSION)) {
3     session_start();
4 }
5 require("../fonctionPHP/fonctionUtilisateur.php");
6 require("../fonctionPHP/fonctionCommentaire.php");
7 require("../fonctionPHP/fonctionJeux.php");
8 if (isset($_POST['supCom'])) {
9     supprimerCommentaire($_POST['supCom']);
10 }
11 if (isset($_POST['supProfile'])) {
12     supprimerUtilisateur($_SESSION['nomLog']);
13     header('Location: deconnexion.php');
14 }
15 if (isset($_POST['passerPublic'])) {
16     passerJeuPublic($_POST['passerPublic']);
17 }
18 if (isset($_POST['changerProfile'])) {
19     $n_pseudo = htmlspecialchars($_POST["pseudo"], ENT_NOQUOTES);
20     $n_mail = htmlspecialchars($_POST["mail"], ENT_NOQUOTES);
21     $n_mdp = htmlspecialchars($_POST["mdp"], ENT_NOQUOTES);
22     if($n_mdp!=""){
23         $n_mdp=hash('sha256',$n_mdp);
24     }
25
26     if(!modifierUtilisateur($_SESSION['nomLog'],$n_pseudo, $n_mail, $n_mdp)){
27         echo 'Il y a eu une erreur lors de la modification du compte';
28     }
29 }
30 }
31 $data = infoUtilisateur($_SESSION['nomLog']);
32 $comms = voirCommentaireUtilisateur($_SESSION['nomLog']);
33 $jeux = listeJeuxUtilisateur($_SESSION['nomLog']);
34 ?>
35 <!DOCTYPE html>
36 <html lang="en">
37
38 <head>
39     <meta charset="UTF-8">
40     <meta http-equiv="X-UA-Compatible" content="IE=edge">
41     <meta name="viewport" content="width=device-width, initial-scale=1.0">
42     <link
43 href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
44 rel="stylesheet" integrity="sha384-
45 1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
46 crossorigin="anonymous">
47 <script
48 src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
49 integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYs0g+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
50 crossorigin="anonymous"></script>
51 <link rel="stylesheet" href="./style/style.css">
52 <title>Mon compte</title>
53 </head>
54 <!-- Email,Pseudo,PhotoProfil -->
55 <body>
56
57     <?php include("navbar.php"); ?>
58     <div class="container" id="pp">
```

```

53     
54     <p> Pseudo : <?= $data[0]["pseudo"] ?></p>
55     <p> Email : <?= $data[0]["email"] ?></p>
56     <form action="" method="POST">
57         <button type="submit" class="btn btn-danger" name="supProfile"
value="oui">Supprimer ce profile</button>
58     </form>
59 </div>
60 <div class="container" id="pp">
61     <legend>Modifier son profil</legend>
62     <form action="#" method="POST">
63         <div class="form-group">
64             <label for="mail">Nouveau Pseudo</label>
65             <input type="text" class="form-control" id="pseudo" name="pseudo"
placeholder="Mon nouveau Pseudo">
66             <!-- <small id="emailHelp" class="form-text text-muted">We'll never
share your email with anyone else.</small> -->
67         </div>
68         <div class="form-group">
69             <label for="mail">Nouvelle Adresse Email</label>
70             <input type="email" class="form-control" id="mail" name="mail" aria-
describedby="emailHelp" placeholder="mon nouveau Mail">
71             <!-- <small id="emailHelp" class="form-text text-muted">We'll never
share your email with anyone else.</small> -->
72         </div>
73         <div class="form-group">
74             <label for="mdp">Nouveau Mot de passe</label>
75             <input type="password" class="form-control" id="mdp" name="mdp"
placeholder="Mon nouveau mot de passe">
76         </div>
77         <div class="form-group">
78             <button type="submit" name="changerProfile" class="btn btn-
primary">Faire un demande de modification</button><br>
79         </div>
80     </form>
81 </div>
82 <div class="container" id="pp">
83     <form action="" method="post">
84     <legend>tout mes commentaires</legend>
85     <ul class="list-group">
86         <?php
87         for ($i = 0; $i < sizeof($comms); $i++) {
88             echo ' <li class="list-group-item">
89             Jeux : ' . $comms[$i]['titre'] . '<br>
90             Date : ' . explode(' ', $comms[$i]["dateCommentaire"])[0] . '<br>
91             Commentaire : ' . $comms[$i]['commentaire'] . '<br>
92             <button type="submit" class="btn btn-primary" name="supCom" value=""
. $comms[$i]['idCommentaire'] . "'>Supprimer ce commentaire</button>
93             </li>
94             '
95             }
96         ?>
97     </ul>
98     </form>
99 </div>
100 <div class="container" id="pp">
101     <form action="" method="post">
102     <legend>Liste de mes Jeux</legend>
103     <ul class="list-group">

```

```
104         <?php
105         for ($i = 0; $i < sizeof($jeux); $i++) {
106             echo ' <li class="list-group-item">
107             Titre : ' . $jeux[$i]['titre'] . '<br>
108             
110             '
111             if ($jeux[$i]['public'] == false) {
112                 echo '
113                 <button type="submit" class="btn btn-primary" name="passerPublic"
114                 value="' . $jeux[$i]['idJeu'] . '">Passer le jeu en public</button>
115                 </li>
116                 '
117                 ;
118             }
119         }
120     </form>
121 </div>
122 </body>
123 </html>
```

```
1 <?php
2
3 define("HOST", "localhost");
4 define("DBNAME", "db_PNJ");
5 define("DBUSER", "TPI_USER");
6 define("DBPWD", "Finish_him");
7
8 function connexionBDD()
9 {
10     static $dbc = null;
11
12     // Première visite de la fonction
13     if ($dbc == null)
14     {
15         // Essaie le code ci-dessous
16         try
17         {
18             $dbc = new PDO('mysql:host=' . HOST . ';dbname=' . DBNAME, DBUSER, DBPWD,
array(
19                 PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8",
20                 PDO::ATTR_PERSISTENT => true
21             ));
22         }
23         // Si une exception est arrivée
24         catch (Exception $e)
25         {
26             echo 'Erreur : ' . $e->getMessage() . '<br />';
27             echo 'N° : ' . $e->getCode();
28             // Quitte le script et meurt
29             die('Could not connect to MySQL');
30         }
31     }
32     // Pas d'erreur, retourne un connecteur
33     return $dbc;
34 }
```

localhost:4649/?mode=php

```

59     }
60     if ($n_mdp != "") {
61         $sql .= ", motDePasse = :MDP ";
62     }
63 } elseif ($n_mail != "") {
64     $sql .= "email = :EMAIL ";
65     if ($n_mdp != "") {
66         $sql .= ", motDePasse = :MDP ";
67     }
68 } elseif ($n_mdp != "") {
69     $sql .= "motDePasse = :MDP ";
70 }
71 $sql .= "WHERE pseudo = :PSEUDO";
72
73 if ($ps == null) {
74     $ps = connexionBDD()->prepare($sql);
75 }
76 $reponse = false;
77
78 try {
79     $ps->bindParam(":PSEUDO", $pseudo, PDO::PARAM_STR);
80     if ($n_pseudo != "") {$ps->bindParam(":NPSEUDO", $n_pseudo,
PDO::PARAM_STR);}
81     if ($n_mail != "") {$ps->bindParam(":EMAIL", $n_mail, PDO::PARAM_STR);}
82     if ($n_mdp != "") {$ps->bindParam(":MDP", $n_mdp, PDO::PARAM_STR);}
83     //$ps->bindParam(":PSEUDO", $pseudo, PDO::PARAM_STR);
84
85     $reponse = $ps->execute();
86     if($reponse==true){
87         if (empty($_SESSION)) {
88             session_start();
89         }
90         $_SESSION['nomLog']=$n_pseudo;
91     }
92 } catch (PDOException $e) {
93     echo ($e->getMessage());
94 }
95 return $reponse;
96 }
97 /**
98  * vérifie si le mot de passe et l'email de l'utilisateur sont correctes
99  *
100  * @param [string] $email
101  * @param [string] $mdp
102  * @return bool
103  */
104 function connecterUtilisateur($email, $mdp)
105 {
106     static $ps = null;
107     //$sql
108     $sql = "SELECT email,motDePasse,pseudo,idStatut FROM UTILISATEUR WHERE email =
:EMAIL";
109
110     if ($ps == null) {
111         $ps = connexionBDD()->prepare($sql);
112         $ps->bindParam(":EMAIL", $email, PDO::PARAM_STR);
113     }
114     $reponse = false;
115
116     try {

```



```

117         if ($ps->execute())
118             $reponse = $ps->fetchAll(PDO::FETCH_ASSOC);
119     } catch (PDOException $e) {
120         echo $e->getMessage();
121     }
122     if (count($reponse) > 0) {
123         if ($email == $reponse[0]["email"] && $mdp == $reponse[0]["motDePasse"]) {
124             if ($reponse[0]["idStatut"] == 2 || $reponse[0]["idStatut"] == 6) {
125                 if (empty($_SESSION)) {
126                     session_start();
127                 }
128                 $_SESSION['nomLog'] = $reponse[0]["pseudo"];
129                 $_SESSION['statutLog'] = $reponse[0]["idStatut"];
130                 $reponse = true;
131             } else {
132                 $reponse = false;
133             }
134             //le compte existe mais soit il est pas vérifier , banni, refusé
            ,supprimé
135         }else{
136             $reponse = false;
137         }
138     }
139     return $reponse;
140 }
141 /**
142  * modifie le statut d'utilisateur à valider
143  *
144  * @param [string] $pseudo
145  * @return void
146  */
147 function validerUtilisateur($pseudo)
148 {
149     static $ps = null;
150     $sql = "UPDATE UTILISATEUR SET idStatut = 2 ";
151     $sql .= "WHERE pseudo = :PSEUDO";
152
153     if ($ps == null) {
154         $ps = connexionBDD()->prepare($sql);
155     }
156     $reponse = false;
157
158     try {
159         $ps->bindParam(":PSEUDO", $pseudo, PDO::PARAM_STR);
160
161         $reponse = $ps->execute();
162     } catch (PDOException $e) {
163         echo ($e->getMessage());
164     }
165     return $reponse;
166 }
167 /**
168  * modifie le statut d'utilisateur à refuse
169  *
170  * @param [string] $pseudo
171  * @return void
172  */
173 function refuserUtilisateur($pseudo)
174 {
175     static $ps = null;

```

```
176     $sql = "UPDATE UTILISATEUR SET idStatut = 3 ";
177     $sql .= "WHERE pseudo = :PSEUDO";
178
179     if ($ps == null) {
180         $ps = connexionBDD()->prepare($sql);
181     }
182     $reponse = false;
183
184     try {
185         $ps->bindParam(":PSEUDO", $pseudo, PDO::PARAM_STR);
186
187         $reponse = $ps->execute();
188     } catch (PDOException $e) {
189         echo ($e->getMessage());
190     }
191     return $reponse;
192 }
193 /**
194  * modifie le statut d'utilisateur à banni
195  *
196  * @param [string] $pseudo
197  * @return void
198  */
199 function bannirUtilisateur($pseudo)
200 {
201     static $ps = null;
202     $sql = "UPDATE UTILISATEUR SET idStatut = 4 ";
203     $sql .= "WHERE pseudo = :PSEUDO";
204
205     if ($ps == null) {
206         $ps = connexionBDD()->prepare($sql);
207     }
208     $reponse = false;
209
210     try {
211         $ps->bindParam(":PSEUDO", $pseudo, PDO::PARAM_STR);
212
213         $reponse = $ps->execute();
214     } catch (PDOException $e) {
215         echo ($e->getMessage());
216     }
217     return $reponse;
218 }
219 /**
220  * modifie le statut d'utilisateur à supprimer
221  *
222  * @param [string] $pseudo
223  * @return void
224  */
225 function supprimerUtilisateur($pseudo)
226 {
227     static $ps = null;
228     $sql = "UPDATE UTILISATEUR SET idStatut = 5 ";
229     $sql .= "WHERE pseudo = :PSEUDO";
230
231     if ($ps == null) {
232         $ps = connexionBDD()->prepare($sql);
233     }
234     $reponse = false;
235
```

```
236     try {
237         $ps->bindParam(":PSEUDO", $pseudo, PDO::PARAM_STR);
238
239         $reponse = $ps->execute();
240     } catch (PDOException $e) {
241         echo ($e->getMessage());
242     }
243     return $reponse;
244 }
245 /**
246  * retourne un tableau de tout les utilisateur qui ne sont pas vérifier
247  *
248  * @return array
249  */
250 function infoNonVerifier()
251 {
252     static $ps = null;
253     $sql = "SELECT pseudo,idUtilisateur from UTILISATEUR";
254     $sql .= " WHERE idStatut =1";
255
256     if ($ps == null) {
257         $ps = connexionBDD()->prepare($sql);
258     }
259     $reponse = false;
260
261     try {
262         if ($ps->execute())
263             $reponse = $ps->fetchAll(PDO::FETCH_ASSOC);
264     } catch (PDOException $e) {
265         echo $e->getMessage();
266     }
267     return $reponse;
268 }
269 /**
270  * retourne un tableau de tout les utilisateur qui ne sont pas admins
271  *
272  * @return array
273  */
274 function infoToutUtilisateur()
275 {
276     static $ps = null;
277     $sql = "SELECT pseudo,idUtilisateur from UTILISATEUR WHERE idStatut = 2";
278
279     if ($ps == null) {
280         $ps = connexionBDD()->prepare($sql);
281     }
282     $reponse = false;
283
284     try {
285         if ($ps->execute())
286             $reponse = $ps->fetchAll(PDO::FETCH_ASSOC);
287     } catch (PDOException $e) {
288         echo $e->getMessage();
289     }
290     return $reponse;
291 }
292 /**
293  * retourne toute les information sur un utilisateur
294  *
295  * @param [string] $pseudo
```

```
296  * @return array
297  */
298  function infoUtilisateur($pseudo)
299  {
300      static $ps = null;
301      $sql = "SELECT * from UTILISATEUR WHERE pseudo = :PSEUDO";
302
303      if ($ps == null) {
304          $ps = connexionBDD()->prepare($sql);
305      }
306      $reponse = false;
307
308      try {
309          $ps->bindParam(":PSEUDO", $pseudo, PDO::PARAM_STR);
310          if ($ps->execute())
311              $reponse = $ps->fetchAll(PDO::FETCH_ASSOC);
312      } catch (PDOException $e) {
313          echo $e->getMessage();
314      }
315      return $reponse;
316  }
317
```

```
1 <?php
2 require_once("../bdd/connexionBdd.php");
3 /**
4  * ajoute un jeu dans la base de données
5  *
6  * @param [string] $titre
7  * @param [string] $date
8  * @param [string] $editeur
9  * @param [string] $description
10 * @param [bool] $public
11 * @param [int] $idCategorie
12 * @param [string] $photo
13 * @param [string] $pseudo
14 * @return void
15 */
16 function creerJeu($titre, $date, $editeur, $description, $public, $idCategorie,
17 $photo,$pseudo)
18 {
19     static $ps = null;
20     $sql = 'INSERT INTO JEU
21 (titre,dateSorite,editeur,description,public,idCategorie,imageJeu,idUtilisateur) ' ;
22     $sql .= 'SELECT :TITRE, :DATE, :EDITEUR, :DESCRIPTION, :PUBLIC, :IDCATEGORIE,
23 :IMAGEJEU, UTILISATEUR.idUtilisateur FROM UTILISATEUR ' ;
24     $sql .= 'WHERE UTILISATEUR.pseudo = :PSEUDO ' ;
25
26     if ($ps == null) {
27         $ps = connexionBDD()->prepare($sql);
28     }
29     $reponse = false;
30
31     try {
32         $ps->bindParam(":TITRE", $titre, PDO::PARAM_STR);
33         $ps->bindParam(":DATE", $date, PDO::PARAM_STR);
34         $ps->bindParam(":EDITEUR", $editeur, PDO::PARAM_STR);
35         $ps->bindParam(":DESCRIPTION", $description, PDO::PARAM_STR);
36         $ps->bindParam(":PUBLIC", $public, PDO::PARAM_BOOL);
37         $ps->bindParam(":IDCATEGORIE", $idCategorie, PDO::PARAM_INT);
38         $ps->bindParam(":IMAGEJEU", $photo, PDO::PARAM_STR);
39         $ps->bindParam(":PSEUDO", $pseudo, PDO::PARAM_STR);
40
41         $reponse = $ps->execute();
42     } catch (PDOException $e) {
43         echo ($e->getMessage());
44         $reponse = $e->getMessage();
45     }
46     return $reponse;
47 }
48 /**
49 * récupère tout les jeux public de la base de données
50 *
51 * @return array
52 */
53 function recupererJeu()
54 {
55     static $ps = null;
56     $sql = "SELECT j.titre,j.imageJeu,j.dateSorite,j.appreciation,c.categorie FROM
57 JEU j , CATEGORIE c ";
58     $sql .= "WHERE j.idCategorie =c.idCategorie AND j.public = true ";
59     $sql .= "ORDER BY j.idJeu desc";
```

```
56
57     if ($ps == null) {
58         $ps = connexionBDD()->prepare($sql);
59     }
60     $reponse = false;
61
62     try {
63         if ($ps->execute())
64             $reponse = $ps->fetchAll(PDO::FETCH_ASSOC);
65     } catch (PDOException $e) {
66         echo $e->getMessage();
67     }
68     return $reponse;
69 }
70 /**
71  * retourne un tableau avec toutes les catégories de jeu
72  *
73  * @return array
74  */
75 function infoCategorie()
76 {
77     static $ps = null;
78     $sql = "SELECT * from CATEGORIE";
79
80     if ($ps == null) {
81         $ps = connexionBDD()->prepare($sql);
82     }
83     $reponse = false;
84
85     try {
86         if ($ps->execute())
87             $reponse = $ps->fetchAll(PDO::FETCH_ASSOC);
88     } catch (PDOException $e) {
89         echo $e->getMessage();
90     }
91     return $reponse;
92 }
93 /**
94  * retourne un tableau avec tout les information d'un jeu
95  *
96  * @param [string] $nomJeu
97  * @return array
98  */
99 function infoJeu($nomJeu)
100 {
101     static $ps = null;
102     $sql = "SELECT j.titre ,j.imageJeu ,j.editeur ,j.dateSortie
,j.appreciation,j.description ,c.categorie ";
103     $sql .= "FROM JEU j , CATEGORIE c ";
104     $sql .= "WHERE c.idCategorie =j.idCategorie ";
105     $sql .= "AND j.titre = :TITRE ";
106
107     if ($ps == null) {
108         $ps = connexionBDD()->prepare($sql);
109     }
110     $reponse = false;
111
112     try {
113         $ps->bindParam(":TITRE", $nomJeu, PDO::PARAM_STR);
114         if ($ps->execute())
```

```
115         $reponse = $ps->fetchAll(PDO::FETCH_ASSOC);
116     } catch (PDOException $e) {
117         echo $e->getMessage();
118     }
119     return $reponse;
120 }
121 /**
122  * retourne un tableau de jeux filter par multiple critères
123  *
124  * @param [string] $titre
125  * @param [int] $idCategorie
126  * @param [int] $annee
127  * @param [int] $moyenne
128  * @param [string] $utilisateur
129  * @return array
130  */
131 function rechercheJeu($titre, $idCategorie, $annee, $moyenne,$utilisateur)
132 {
133     static $ps = null;
134
135     $sql = "SELECT j.titre, j.appreciation, j.imageJeu FROM JEU j, UTILISATEUR u,
136     CATEGORIE c WHERE j.public=true AND c.idCategorie=j.idCategorie AND u.idUtilisateur
137     =j.idUtilisateur ";
138     if ($titre != "") {
139         $sql .= "AND titre LIKE :TITRE ";
140     }
141     if ($idCategorie != "") {
142         $sql .= "AND j.idCategorie = :IDCATEGORIE ";
143     }
144     if ($moyenne != "") {
145         $sql .= "AND j.appreciation >= :MOYENNE ";
146     }
147     if ($annee != "") {
148         $sql .= "AND j.dateSorite >= :ANNEE ";
149     }
150     if ($utilisateur != "") {
151         $sql .= "AND j.idUtilisateur = u.idUtilisateur AND u.pseudo = :UTILISATEUR";
152     }
153     // $sql = "SELECT j.titre,j.imageJeu,j.dateSorite ,c.categorie FROM JEU j ,
154     CATEGORIE c ";
155     // $sql .= "WHERE j.idCategorie =c.idCategorie ";
156     if ($ps == null) {
157         $ps = connexionBDD()->prepare($sql);
158     }
159     $reponse = false;
160
161     try {
162         if ($titre != "") {
163             $titre = "%" . $titre . "%";
164             $ps->bindParam(":TITRE", $titre, PDO::PARAM_STR);
165         }
166         if ($idCategorie != "") {
167             $ps->bindParam(":IDCATEGORIE", $idCategorie, PDO::PARAM_INT);
168         }
169         if ($annee != "") {
170             $annee = $annee . "-01-01";
171             $ps->bindParam(":ANNEE", $annee, PDO::PARAM_STR);
```

```
172     }
173     if ($moyenne != "") {
174
175         $ps->bindParam(":MOYENNE", $moyenne, PDO::PARAM_INT);
176     }
177     if ($utilisateur != "") {
178         $ps->bindParam(":UTILISATEUR", $utilisateur, PDO::PARAM_STR);
179     }
180     if ($ps->execute())
181         $reponse = $ps->fetchAll(PDO::FETCH_ASSOC);
182 } catch (PDOException $e) {
183     echo $e->getMessage();
184 }
185 return $reponse;
186 }
187 /**
188  * retourne un tableau avec tout les jeux créer par un utilisateur
189  *
190  * @param [string] $pseudo
191  * @return array
192  */
193 function listeJeuUtilisateur($pseudo){
194     static $ps = null;
195     $sql = "SELECT j.titre,j.idJeu,j.dateSortie,j.imageJeu, j.public, c.categorie
196 FROM JEU j , CATEGORIE c, UTILISATEUR u ";
197     $sql .= "WHERE j.idCategorie =c.idCategorie ";
198     $sql .= "AND j.idUtilisateur = u.idUtilisateur ";
199     $sql .= "AND u.pseudo = :PSEUDO ";
200     $sql .= "ORDER BY j.titre ASC";
201
202     if ($ps == null) {
203         $ps = connexionBDD()->prepare($sql);
204     }
205     $reponse = false;
206
207     try {
208         $ps->bindParam(":PSEUDO",$pseudo,PDO::PARAM_STR);
209         if ($ps->execute())
210             $reponse = $ps->fetchAll(PDO::FETCH_ASSOC);
211     } catch (PDOException $e) {
212         echo $e->getMessage();
213     }
214     return $reponse;
215 }
216 /**
217  * change le statut d'un jeu de privé à public
218  *
219  * @param [int] $idJeu
220  * @return void
221  */
222 function passerJeuPublic($idJeu){
223     static $ps = null;
224     $sql = "UPDATE JEU SET public = true ";
225     $sql .= "WHERE JEU.idJeu = :IDJEU";
226
227     if ($ps == null) {
228         $ps = connexionBDD()->prepare($sql);
229     }
230     $reponse = false;
```



```
231     try {
232         $ps->bindParam(":IDJEU", $idJeu, PDO::PARAM_INT);
233
234         $reponse = $ps->execute();
235     } catch (PDOException $e) {
236         echo ($e->getMessage());
237     }
238     return $reponse;
239 }
240
```

```
1 <?php
2
3 require_once("../bdd/connexionBdd.php");
4 /**
5  * ajoute un commentaire dans la base de données
6  *
7  * @param [string] $pseudo
8  * @param [string] $nomJeu
9  * @param [string] $commentaire
10 * @param [bool] $valide
11 * @return void
12 */
13 function creerCommentaire($pseudo, $nomJeu, $commentaire, $valide)
14 {
15     static $ps = null;
16     $statut = 1;
17     $sql = 'INSERT INTO COMMENTAIRE
18 (idUtilisateur,idJeu,accepte,dateCommentaire,commentaire) ' ;
19     $sql .= 'SELECT UTILISATEUR.idUtilisateur,JEU.idJeu
20 ,:ACCEPTE,:DATECOMMENTAIRE,:COMMENTAIRE FROM UTILISATEUR,JEU ' ;
21     $sql .= 'WHERE JEU.titre= :TITRE ' ;
22     $sql .= 'AND UTILISATEUR.pseudo= :PSEUDO ' ;
23
24     if ($ps == null) {
25         $ps = connexionBDD()->prepare($sql);
26     }
27     $reponse = false;
28
29     try {
30         $ps->bindParam(":PSEUDO", $pseudo, PDO::PARAM_STR);
31         $ps->bindParam(":TITRE", $nomJeu, PDO::PARAM_STR);
32         $ps->bindParam(":ACCEPTE", $valide, PDO::PARAM_BOOL);
33         $ps->bindParam(":COMMENTAIRE", $commentaire, PDO::PARAM_STR);
34         $ps->bindParam(":DATECOMMENTAIRE", date('Y-m-d'), PDO::PARAM_STR);
35
36         $reponse = $ps->execute();
37     } catch (PDOException $e) {
38         echo ($e->getMessage());
39     }
40     return $reponse;
41 }
42 /**
43  * modifie le contenu d'un commentaire
44  *
45  * @param [string] $n_comm
46  * @param [string] $idCom
47  * @return void
48  */
49 function modifierCommentaire($n_comm,$idCom){
50     static $ps = null;
51     $sql = "UPDATE COMMENTAIRE SET commentaire = :COMM ";
52     $sql .= "WHERE idCommentaire = :IDCOM";
53
54     if ($ps == null) {
55         $ps = connexionBDD()->prepare($sql);
56     }
57     $reponse = false;
58
59     try {
```

```

58     $ps->bindParam(":COMM",$n_comm,PDO::PARAM_STR);
59     $ps->bindParam(":IDCOM", $idCom, PDO::PARAM_INT);
60
61     $reponse = $ps->execute();
62 } catch (PDOException $e) {
63     echo ($e->getMessage());
64 }
65 return $reponse;
66 }
67 /**
68  * retourne un tableau de tout les commentaire non-validés
69  *
70  * @return array
71  */
72 function infoCommentaire()
73 {
74     static $ps = null;
75     $sql = "SELECT c.idCommentaire,c.commentaire ,u.pseudo, j.titre FROM
COMMENTAIRE c ,UTILISATEUR u ,JEU j ";
76     $sql .= "WHERE c.idUtilisateur =u.idUtilisateur ";
77     $sql .= "AND c.idJeu =j.idJeu ";
78     $sql .= "AND c.accepte =FALSE ";
79
80     if ($ps == null) {
81         $ps = connexionBDD()->prepare($sql);
82     }
83     $reponse = false;
84
85     try {
86         if ($ps->execute())
87             $reponse = $ps->fetchAll(PDO::FETCH_ASSOC);
88     } catch (PDOException $e) {
89         echo $e->getMessage();
90     }
91     return $reponse;
92 }
93 /**
94  * change le status d'un commentaire de non-validé à validé
95  *
96  * @param [int] $idCommentaire
97  * @return void
98  */
99 function validerCommentaire($idCommentaire)
100 {
101     static $ps = null;
102     $sql = "UPDATE COMMENTAIRE SET accepte = true ";
103     $sql .= "WHERE idCommentaire = :IDCOM";
104
105     if ($ps == null) {
106         $ps = connexionBDD()->prepare($sql);
107     }
108     $reponse = false;
109
110     try {
111         $ps->bindParam(":IDCOM", $idCommentaire, PDO::PARAM_INT);
112
113         $reponse = $ps->execute();
114     } catch (PDOException $e) {
115         echo ($e->getMessage());
116     }

```

```
117     return $reponse;
118 }
119 /**
120  * retourne tout les commentaires d'un jeu
121  *
122  * @param [string] $titre
123  * @return array
124  */
125 function infoJeuCommentaire($titre)
126 {
127     static $ps = null;
128     $sql = "SELECT c.idCommentaire,c.commentaire ,c.dateCommentaire ,u.pseudo
,u.photoDeProfil FROM COMMENTAIRE c ,UTILISATEUR u,JEU j ";
129     $sql .= "WHERE c.idUtilisateur = u.idUtilisateur ";
130     $sql .= "AND c.idJeu =j.idJeu ";
131     $sql .= "AND j.titre =:TITRE ";
132     $sql .= "AND c.accepte =TRUE ";
133
134     if ($ps == null) {
135         $ps = connexionBDD()->prepare($sql);
136     }
137     $reponse = false;
138
139     try {
140         $ps->bindParam(":TITRE", $titre, PDO::PARAM_STR);
141         if ($ps->execute())
142             $reponse = $ps->fetchAll(PDO::FETCH_ASSOC);
143     } catch (PDOException $e) {
144         echo $e->getMessage();
145     }
146     return $reponse;
147 }
148 /**
149  * retourne tout les commentaires d'un utilisateur
150  *
151  * @param [string] $pseudo
152  * @return array
153  */
154 function voirCommentaireUtilisateur($pseudo)
155 {
156     static $ps = null;
157     $sql = "SELECT j.titre, c.idCommentaire ,c.dateCommentaire ,c.commentaire FROM
COMMENTAIRE c ,JEU j,UTILISATEUR u ";
158     $sql .= "WHERE j.idJeu =c.idJeu ";
159     $sql .= "AND u.idUtilisateur =c.idUtilisateur ";
160     $sql .= "AND u.pseudo = :PSEUDO ";
161     $sql .= "AND c.accepte =true ";
162
163     if ($ps == null) {
164         $ps = connexionBDD()->prepare($sql);
165     }
166     $reponse = false;
167
168     try {
169         $ps->bindParam(":PSEUDO", $pseudo, PDO::PARAM_STR);
170         if ($ps->execute())
171             $reponse = $ps->fetchAll(PDO::FETCH_ASSOC);
172     } catch (PDOException $e) {
173         echo $e->getMessage();
174     }
175 }
```

```
175     return $reponse;
176 }
177 /**
178  * passe le statut d'un commentaire de validé à non-validé
179  *
180  * @param [int] $idCommentaire
181  * @return void
182  */
183 function supprimerCommentaire($idCommentaire)
184 {
185     static $ps = null;
186     $sql = "DELETE FROM COMMENTAIRE ";
187     $sql .= "WHERE idCommentaire = :IDCOM";
188
189     if ($ps == null) {
190         $ps = connexionBDD()->prepare($sql);
191     }
192     $reponse = false;
193
194     try {
195         $ps->bindParam(":IDCOM", $idCommentaire, PDO::PARAM_INT);
196
197         $reponse = $ps->execute();
198     } catch (PDOException $e) {
199         echo ($e->getMessage());
200     }
201     return $reponse;
202 }
203
```

```
1 <?php
2 require_once("../bdd/connexionBdd.php");
3 /**
4  * retourne la moyenne d'un jeu
5  *
6  * @param [string] $nomJeu
7  * @return int
8  */
9 function calculerNoteMoyenne($nomJeu)
10 {
11     static $ps = null;
12     $sql = "SELECT n.note FROM NOTE n, JEU j ";
13     $sql .= "WHERE n.idJeu = j.idJeu ";
14     $sql .= "AND j.titre=:TITRE";
15
16     if ($ps == null) {
17         $ps = connexionBDD()->prepare($sql);
18     }
19     $reponse = false;
20
21     try {
22         $ps->bindParam(":TITRE", $nomJeu, PDO::PARAM_STR);
23         if ($ps->execute())
24             $reponse = $ps->fetchAll(PDO::FETCH_ASSOC);
25         $moyenne = 0;
26         for ($i = 0; $i < sizeof($reponse); $i++) {
27             $moyenne += $reponse[$i]['note'];
28         }
29         $moyenne = $moyenne / sizeof($reponse);
30     } catch (PDOException $e) {
31         echo $e->getMessage();
32     }
33     return $moyenne;
34 }
35 /**
36  * donne une moyenne à un jeu
37  *
38  * @param [int] $moyenne
39  * @param [string] $nomJeu
40  * @return void
41  */
42 function attribuerMoyenne($moyenne, $nomJeu)
43 {
44     static $ps = null;
45     $sql = "UPDATE JEU SET appreciation = :NOTE ";
46     $sql .= "WHERE titre = :TITRE";
47
48     if ($ps == null) {
49         $ps = connexionBDD()->prepare($sql);
50     }
51     $reponse = false;
52
53     try {
54         $ps->bindParam(":NOTE", $moyenne, PDO::PARAM_STR);
55         $ps->bindParam(":TITRE", $nomJeu, PDO::PARAM_STR);
56         $reponse = $ps->execute();
57     } catch (PDOException $e) {
58         echo ($e->getMessage());
59     }
60 }
```

```
60     return $reponse;
61 }
62 /**
63  * ajoute une note à un jeu dans la base de données
64  *
65  * @param [int] $note
66  * @param [string] $pseudo
67  * @param [string] $titre
68  * @return bool
69  */
70 function ajouterNote($note,$pseudo,$titre){
71     static $ps = null;
72     $sql = "INSERT INTO NOTE (note,idUtilisateur,idJeu) ";
73     $sql .= "SELECT :NOTE ,u.idUtilisateur ,j.idJeu ";
74     $sql .= "FROM JEU j, UTILISATEUR u ";
75     $sql .= "WHERE u.pseudo =:PSEUDO ";
76     $sql .= "AND j.titre =:TITRE ";
77
78     if ($ps == null) {
79         $ps = connexionBDD()->prepare($sql);
80     }
81     $reponse = false;
82
83     try {
84         $ps->bindParam(":NOTE", $note, PDO::PARAM_INT);
85         $ps->bindParam(":TITRE", $titre, PDO::PARAM_STR);
86         $ps->bindParam(":PSEUDO", $pseudo, PDO::PARAM_STR);
87         $reponse = $ps->execute();
88     } catch (PDOException $e) {
89         echo ($e->getMessage());
90     }
91     return $reponse;
92 }
93 /**
94  * regarde si un utilisateur a déjà commenter un jeux
95  *
96  * @param [string] $pseudo
97  * @param [string] $titre
98  * @return bool
99  */
100 function dejaNote($pseudo,$titre){
101     static $ps = null;
102     $sql = "SELECT u.pseudo FROM UTILISATEUR u , JEU j ,NOTE n ";
103     $sql .= "WHERE u.idUtilisateur =n.idUtilisateur ";
104     $sql .= "AND j.idJeu =n.idJeu ";
105     $sql .= "AND j.titre = :TITRE ";
106     $sql .= "AND u.pseudo = :PSEUDO ";
107
108     if ($ps == null) {
109         $ps = connexionBDD()->prepare($sql);
110     }
111     $reponse = false;
112
113     try {
114         $ps->bindParam(":TITRE", $titre, PDO::PARAM_STR);
115         $ps->bindParam(":PSEUDO", $pseudo, PDO::PARAM_STR);
116         if ($ps->execute())
117             $reponse = $ps->fetchAll(PDO::FETCH_ASSOC);
118         if(sizeof($reponse)==0){
119             //il peut mettre une note
```

```
120         $reponse= true;
121     }else{
122         $reponse=false;
123     }
124 } catch (PDOException $e) {
125     echo $e->getMessage();
126 }
127 return $reponse;
128 }
129
```



```
1 function effacerForm(){
2     document.getElementById("login").reset();
3     console.log("efface")
4 }
5 function effacerRecherche(){
6     document.getElementById("recherche").reset();
7     console.log("efface")
8 }
```

```
1 #navbarMain {
2     background-color: indigo !important;
3 }
4
5 #btnSearch {
6     color: yellow;
7     outline-style: outset;
8 }
9
10 #btnSearch:hover {
11     color: indigo;
12     background-color: yellow;
13 }
14
15 #login {
16     margin-top: 10%;
17     margin-left: 35%;
18     margin-right: 35%;
19     width: 30%;
20 }
21
22 #test {
23     /* background-color: aquamarine; */
24     margin-left: 5%;
25     margin-right: 5%;
26     width: 90%;
27 }
28
29 #carte {
30     margin: auto;
31     margin-bottom: 2.5%;
32     margin-top: 2.5%;
33 }
34
35 #pp {
36     border: 2px solid rgb(0, 0, 0);
37 }
38
39 #info {
40     float: right;
41 }
42
43 #topique {
44     margin-top: 3%;
45     margin-right: 3%;
46 }
47
48 .imgP {
49     height: 50px;
50     border-radius: 50%;
51     float: right;
52 }
53
54 .form-control {
55     margin-bottom: 2.5% !important;
56     margin-top: 2.5% !important;
57 }
58 }
59
```

```
60 .creer {
61     margin-top: 10% !important;
62 }
63
64 #Picture {
65     /* height: 1000px;
66     width: 1000px; */
67     max-width: 150px;
68     max-height: 150px;
69     border-radius: 50%;
70 }
71
72 #testImg {
73     max-width: 200px;
74     max-height: 200px;
75     vertical-align: middle;
76     display: flex;
77 }
78
79 .imageJeu {
80     max-width: 500px;
81 }
82
83 #menu {
84     /* margin-left: 20%; */
85     padding-left: 20%;
86     width: 100%;
87 }
88
89
90 #menuLogo {
91     height: 70%;
92     width: 70%;
93 }
94
95 #navLogo {
96     max-height: 30px;
97 }
98
99 #scrollTest {
100     height: 48rem !important;
101     overflow-y: scroll;
102     border: solid !important;
103 }
104 }
105
106 #scrollComAdmin {
107     height: 10rem !important;
108     overflow-y: scroll;
109     border: solid !important;
110 }
111
112 #carteRecherche {
113     margin-left: 5%;
114     margin-top: 2%;
115 }
116
117 #imgrecherche {
118     max-height: 200;
119     width: auto;
```

120 }

```
1 <?php
2 session_start();
3 session_destroy();
4 header("Location: index.php");
5
```