

```

1 <?php
2 /*****
3 AUTEUR      : Constantin Herrmann
4 LIEU       : CFPT Informatique Genève
5 DATE      : Avril 2020
6 TITRE PROJET: RESA
7 VERSION    : 1.0
8 *****/
9
10 // On test si il faut ajouter le PDO (dans le cas ou le fichier à été include dans un fichier
    qui ne l'as pas)
11 $pdo_path = "../..../pdo.php";
12 if(file_exists($pdo_path)){
13     include_once $pdo_path;
14 }
15
16 /*
17 * Lie une image à un établissement
18 * Params:
19 * - idEtablissement : l'id de l'établissement à lier
20 * - idUploader : l'id de l'utilisateur qui met en ligne la photo
21 * - file : la photo à mettre en ligne
22 * - target_dir : le dossier dans lequel stocker la photo
23 * - Le nombre de répertoire en arrière ou en avant pour atteindre "images/..."
24 */
25 function SaveImageEtablissement($idEtablissement, $idUploader, $file_name, $tmp){
26
27     // Création de l'id unique pour renommer la photo
28     $id = uniqid();
29
30     // On déplace l'image dans le dossier de destination
31     $target_file = UploadImage("images/restaurant/", $file_name, $tmp, $id, "../..../");
32
33     if($target_file != -1){
34
35         // On ajoute l'image dans la base de données
36         $finalePath = $target_file;
37         AddImageToDatabase($id, $finalePath, $idUploader);
38
39         static $query2 = null;
40
41         if ($query2 == null) {
42             $req = 'INSERT INTO `etablissement_has_image`(`idEtablissement`, `idImages`) VALUES
(:e , :i)';
43             $query2 = database()->prepare($req);
44         }
45
46         try {
47             $query2->bindParam(':e', $idEtablissement, PDO::PARAM_STR);
48             $query2->bindParam(':i', $id, PDO::PARAM_STR);
49
50             $query2->execute();
51         }
52         catch (Exception $e) {
53             error_log($e->getMessage());
54         }
55     }
56 }
57
58 /*
59 * Lie une image à un plat
60 * Params:
61 * - idDish : l'id du plat à lier
62 * - idUploader : l'id de l'utilisateur qui met en ligne la photo
63 * - file : la photo à mettre en ligne
64 * - target_dir : le dossier dans lequel stocker la photo
65 * - Le nombre de répertoire en arrière ou en avant pour atteindre "images/..."
66 */
67 function SaveImageDish($idDish, $idUploader, $file, $target_dir, $toMove){
68
69     // Création de l'id unique pour renommer la photo
70     $id = uniqid();
71
72     // On déplace l'image dans le dossier de destination
73     $target_file = UploadImage($toMove, $target_dir, $file, $id);
74
75     // Déclaration d'une variable avec le dossier initial du dossier image
76     $base_dir = "images/";
77
78     // On vérifie si le nom n'est pas égal à -1 (REF: UploadImage())
79     if($target_file != -1){
80
81         // On ajoute l'image dans la base de données

```

```

82     $finalePath = $base_dir.$target_file;
83     AddImageToDatabase($id, $finalePath, $idUploader);
84
85     // On ajoute le lien entre l'image et le plat dans la base de données
86     static $query2 = null;
87
88     if ($query2 == null) {
89         $req = 'INSERT INTO `dish_has_image`(`idDish`, `idImages`) VALUES (:d , :i)';
90         $query2 = database()->prepare($req);
91     }
92
93     try {
94         $query2->bindParam(':d', $idDish, PDO::PARAM_STR);
95         $query2->bindParam(':i', $id, PDO::PARAM_STR);
96
97         $query2->execute();
98     }
99     catch (Exception $e) {
100         error_log($e->getMessage());
101     }
102 }
103 }
104
105 /*
106 * Lie une image à un utilisateur
107 * Params:
108 * - idUser : l'id de l'utilisateur
109 * - idUploader : l'id de l'utilisateur qui met en ligne la photo
110 * - file : la photo à mettre en ligne
111 * - target_dir : le dossier dans lequel stocker la photo
112 * - Le nombre de répertoire en arrière ou en avant pour atteindre "images/..."
113 */
114 function SaveImageUser($idUser, $idUploader, $file, $target_dir, $toMove){
115     // Création de l'id unique pour renommer la photo
116     $id = uniqid();
117
118     // On déplace l'image dans le dossier de destination
119     $target_file = UploadImage($toMove, $target_dir, $file, $id);
120
121     // Déclaration d'une variable avec le dossier initial du dossier image
122     $base_dir = "images/";
123
124     // On vérifie si le nom n'est pas égal à -1 (REF: UploadImage())
125     if($target_file != -1){
126
127         // On ajoute l'image dans la base de données
128         $finalePath = $base_dir.$target_file;
129         AddImageToDatabase($id, $finalePath, $idUploader);
130
131         // On ajoute le lien entre l'image et le plat dans la base de données
132         static $query2 = null;
133
134         if ($query2 == null) {
135             $req = 'INSERT INTO `user_has_image`(`idUser`, `idImage`) VALUES (:d, :i)';
136             $query2 = database()->prepare($req);
137         }
138
139         try {
140             $query2->bindParam(':d', $idUser, PDO::PARAM_STR);
141             $query2->bindParam(':i', $id, PDO::PARAM_STR);
142
143             $query2->execute();
144         }
145         catch (Exception $e) {
146             error_log($e->getMessage());
147         }
148     }
149 }
150
151 /*
152 * Enregistre une images dans un dossier
153 * Params:
154 * - ToMove : Le nombre de répertoire en arrière ou en avant pour atteindre "images/..."
155 * - target_dir : le dossier dans le répertoire "images" dans lequel il faut mettre l'image
156 * - file : la photo à ajouter
157 * - file_name : le nom que l'on souhaite donner à l'image
158 */
159 function UploadImage($target_dir, $file_name, $tmp, $new_file_name, $toMove){
160
161     // On récupère le type de la photo (JPG, PNG, ETC.)
162     $file_type = strtolower(pathinfo($file_name, PATHINFO_EXTENSION));
163
164     // On fusionne le nouveau nom du fichier avec l'extension (mon_nouveau_nom.l'extension)

```

```
165     $myFile = $new_file_name." ".$file_type;
166
167     // On ajoute le dossier de destination
168     $target_file = $target_dir.$myFile;
169
170     // On essaie de déplacer la photo dans le dossier désiré
171     if (move_uploaded_file($tmp, $toMove.$target_file)) {
172         // Retourne le nom du fichier si OK
173         return $target_file;
174     } else {
175         // Retourne -1 si il n'as pas réussi à transférer le fichier
176         return -1;
177     }
178 }
179
180 /*
181 * Ajout le path de l'image dans la base de données
182 * Params:
183 *   - id : l'id de l'image
184 *   - finalePath : Son emplacement final
185 *   - idUploader : l'id du user qui met en ligne la photo
186 */
187 function AddImageToDatabase($id, $finalePath, $idUploader){
188     static $query = null;
189
190     if ($query == null) {
191         $req = 'INSERT INTO `images`(`id`, `path`, `postedby`) VALUES (:i, :p, :u)';
192         $query = database()->prepare($req);
193     }
194
195     try {
196
197         $query->bindParam(':i', $id, PDO::PARAM_STR);
198         $query->bindParam(':p', $finalePath, PDO::PARAM_STR);
199         $query->bindParam(':u', $idUploader, PDO::PARAM_STR);
200
201         $query->execute();
202     }
203     catch (Exception $e) {
204         error_log($e->getMessage());
205     }
206 }
207
208
```