

Département d'Informatique  
Université de Fribourg, Suisse  
<http://diuf.unifr.ch>



UNIVERSITÉ DE FRIBOURG  
UNIVERSITÄT FREIBURG

# **Gestion de recettes de cuisine**

## **Un prototype écrit avec PHP/MySQL**

**SALVADORI ARNAUD**

**No étudiant : [16-211-377]**

**Travail de séminaire en Informatique de Gestion**

**Encadré par :**  
**Prof. Dr. Jacques Pasquier – Rocha**  
**Jobin Johan**



**Fribourg, Septembre 2018**

# Table des matières

<b>1</b>	<b>Introduction.....</b>	<b>1</b>
<b>2</b>	<b>Domaine d'application .....</b>	<b>2</b>
<b>3</b>	<b>Modélisation des données.....</b>	<b>3</b>
3.1	Modèle entité-relation.....	3
3.2	Transformation en tables relationnelles.....	4
3.3	Description des tables .....	4
<b>4</b>	<b>Structure et rendu visuel .....</b>	<b>6</b>
4.1	Structure du site .....	6
4.2	Les différentes pages.....	7
4.2.1	Index.html.....	7
4.2.2	Ajouter_recette.html.....	7
4.2.3	Ajouter_recette.php .....	9
4.2.4	Choix_recette.php.....	9
4.2.5	Lire_recette.php .....	10
<b>5</b>	<b>Éléments de programmation .....</b>	<b>11</b>
5.1	Technologie utilisée.....	11
5.2	Connexion à la base de donnée.....	11
5.3	Exemples de code .....	12
5.3.1	Ajouter_recette.php .....	12
5.3.2	Lire_recette.php .....	14
5.3.3	Choix_recette.php.....	15
5.3.4	Fonctions PHP .....	16
<b>6</b>	<b>Conclusion .....</b>	<b>19</b>
	<b>Bibliographie.....</b>	<b>20</b>

# Liste des figures

<b>Figure 1</b> : Modèle entité-relation .....	3
<b>Figure 2</b> : Diagramme relationnel .....	4
<b>Figure 3</b> : Exemple de table : recette_ingredients.....	5
<b>Figure 4</b> : Structure du site.....	6
<b>Figure 5</b> : Index.html.....	7
<b>Figure 6</b> : Ajouter_recette.html.....	8
<b>Figure 7</b> : Ajouter_recette.php : option 1 .....	9
<b>Figure 8</b> : Ajouter_recette.php : option 2.....	9
<b>Figure 9</b> : Choix_recette.php.....	9
<b>Figure 10</b> : Lire_recette.php.....	10

# Liste des tableaux

<b>Tableau 1</b> : Tables.....	5
--------------------------------	---

# Liste des codes

<b>Code 1</b> : connection.php .....	11
<b>Code 2</b> : addslashes .....	12
<b>Code 3</b> : récolte données.....	13
<b>Code 4</b> : compareWithRecette.....	13
<b>Code 5</b> : ajout recette.....	14
<b>Code 6</b> : lire_recette.php.....	15
<b>Code 7</b> : choix_recette.php .....	16
<b>Code 8</b> : addIngredient .....	17
<b>Code 9</b> : addRecetteIngredients.....	18

# 1

## Introduction

Ce qui aurait pu paraître encore saugrenu il y a quelques années ne l'est désormais plus du tout : à l'heure des réseaux sociaux et des plateformes spécialisées dans le partage d'images, la culture culinaire a maintenant été érigée au rang de véritable mode de vie.

Si pour beaucoup la « boîte à recettes » dissimulée dans un recoin de la cuisine fait encore et toujours des miracles, pour d'autres, l'inspiration passe désormais aussi par l'examen minutieux de sites web dédiés à la « food culture », comme l'ont surnommée nos voisins anglo-saxons.

Le présent travail se veut ainsi un lien direct entre ces deux écoles. Proposant un outil de gestion de recettes en ligne, il permet d'une part de conserver toutes ses recettes au même endroit sans peur d'en égarer une, mais également d'utiliser un système élaboré d'archivage disponible en ligne. Les recettes peuvent en effet être réparties selon des catégories choisies, et même reliées entre elles selon leur liste d'ingrédients.

La première partie de ce travail est ainsi consacrée à la présentation de la modélisation des données, puis seront décrits la structure et le rendu visuel du site web. Enfin, nous reviendrons plus en détail sur les éléments de programmation en lien avec notre démarche.

# 2

## Domaine d'application

Le but de ce travail consistait en la création de plusieurs pages web, toutes connectées entre elles et donnant la possibilité de créer une recette ou de consulter celles déjà créées.

L'enjeu majeur reposait ainsi sur l'élaboration d'une base de données relationnelle, c'est-à-dire dont tous les éléments sont structurés entre eux selon des clés étrangères ou autres relations.

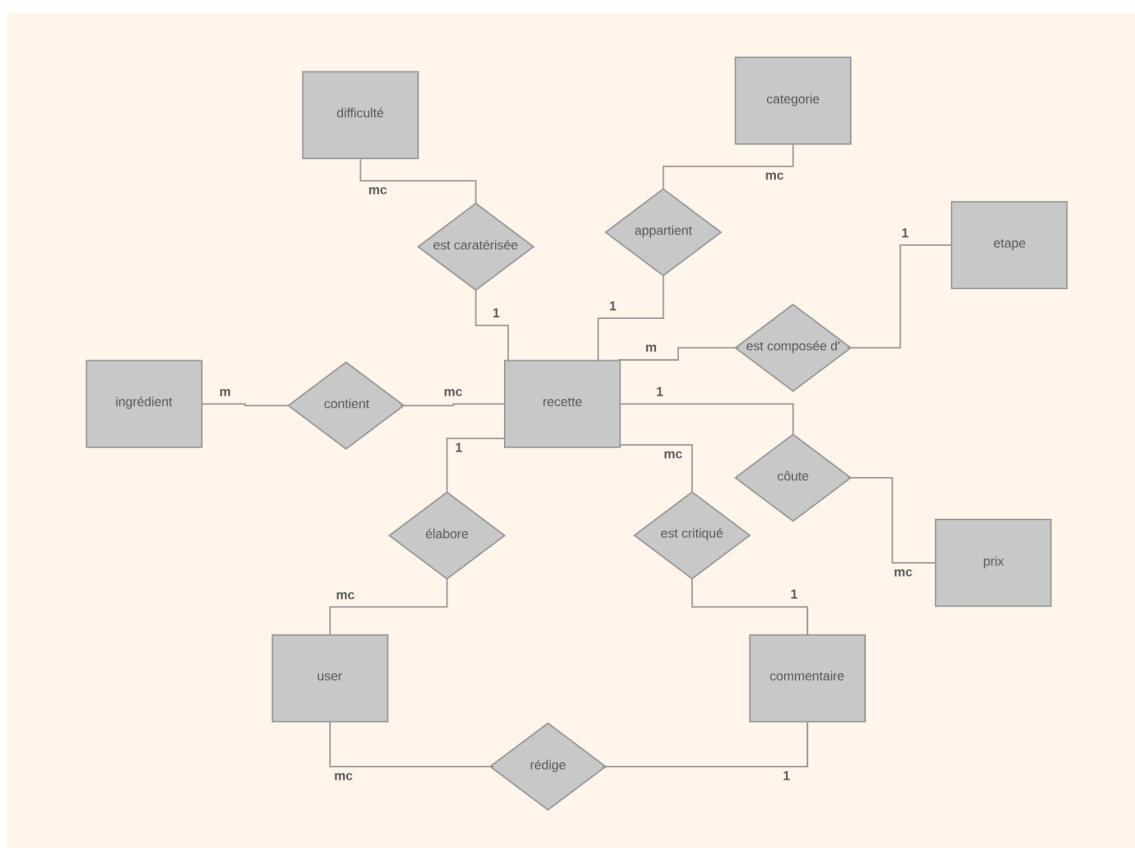
En outre, il fallait encore garantir l'accès en ligne au contenu de la base de données afin que l'utilisateur puisse créer et stocker ses recettes directement depuis le site sans devoir les ajouter manuellement, et qu'il puisse également les consulter à tout moment selon ses besoins. Tout cela, à l'aide d'une interface agréable et simple d'utilisation.

# 3

## Modélisation des données

Ce chapitre présente la structure de la base de données. Dans un premier temps, nous allons ainsi nous intéresser aux différentes relations qui existent entre les tables. Puis, dans un deuxième temps, nous allons également détailler la structure de quelques-unes de ces tables et examiner la façon dont elles sont connectées entre elles.

### 3.1 Modèle entité-relation



**Figure 1 :** *Modèle entité-relation*

Le schéma ci-dessus (fig. 1), réalisé grâce au site [lucidchart.com] est indispensable à la réalisation d'une base de données structurée. En effet, il expose toutes les relations existant entre les tables. Par exemple, la relation entre « recette » et « ingrédient » est une relation multiple-multiple, car une recette contient toujours un ou plusieurs ingrédient(s) (caractérisé



par un m), et un ingrédient est inclus dans 0, 1 ou plusieurs recettes (caractérisé par un mc). Ainsi, il sera nécessaire de créer une table « recette\_ingrédients » qui recensera quels ingrédients sont présents dans quelles recettes.

### 3.2 Transformation en tables relationnelles

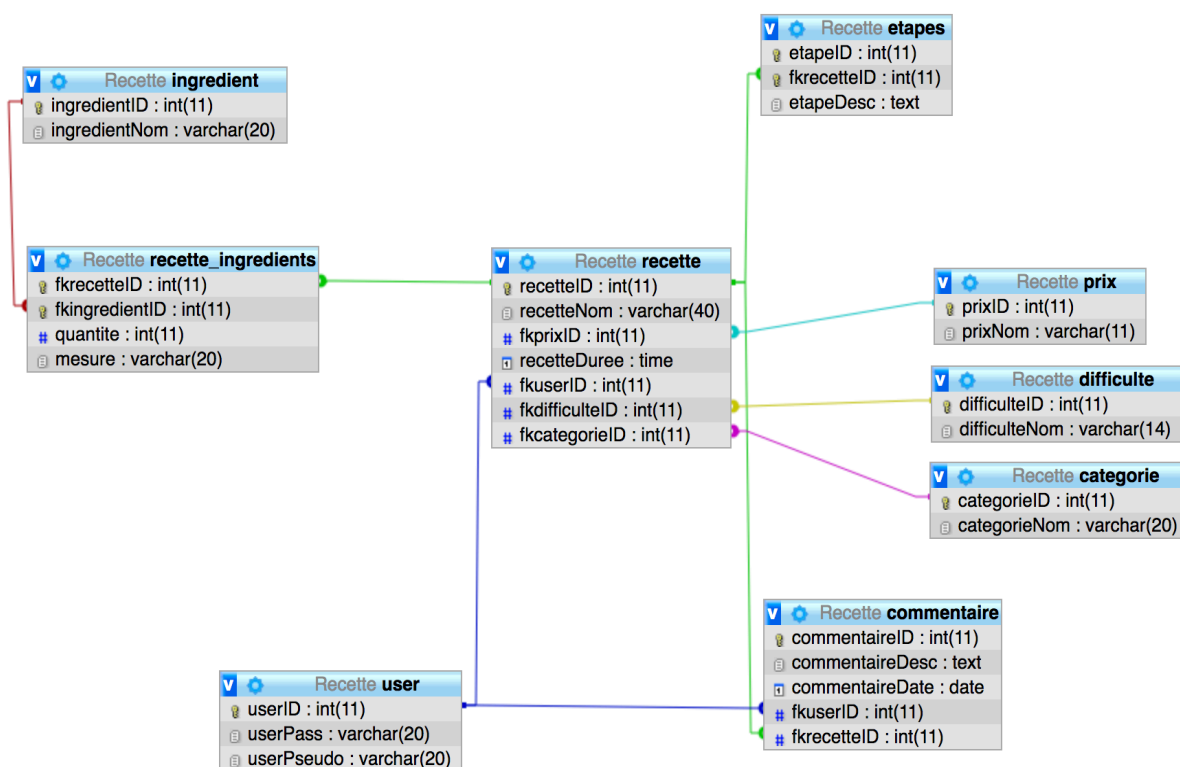


Figure 2 : Diagramme relationnel

Ce diagramme relationnel (fig. 2), réalisé directement dans PHP MySQL, permet de représenter de manière plus détaillée chaque table de la base de données. On peut y voir les différents attributs qui composent les tables. Ces dernières comprennent également des clés primaires et de clés étrangères qui permettent la connexion entre-elles.

### 3.3 Description des tables

Nom de la table	Description
<b>Recette</b>	Table centrale de la base de données. Récolte les caractéristiques des recettes : prix, difficulté, catégorie, user, durée, nom.
<b>Étapes</b>	Table qui recense les étapes d'une recette. Prend recetteID comme clé étrangère.

<b>Prix</b>	Classe les recettes selon le prix : bon marché, coût moyen, assez cher, cher.
<b>Difficulté</b>	Classe les recettes selon la difficulté : très facile, facile, niveau moyen, difficile.
<b>Catégorie</b>	Classe les recettes selon le type de plat : apéritif, entrée, plat principal, dessert.
<b>Ingrédient</b>	Table qui recense tous les ingrédients utilisés dans les recettes.
<b>Recette_ingrédients</b>	Table qui fait le lien entre la table « ingrédient » et la table « recette ». Indique quels ingrédients composent une recette.
<b>User</b>	Table qui recense les utilisateurs qui ont créé les différentes recettes.
<b>Commentaire</b>	Table qui recense les commentaires des utilisateurs. Prend fkrecetteID et userID comme clés étrangères.

Tableau 1 : Tables

Il est à préciser ici que la dernière table, « commentaire », ne sera pas utilisée par la suite.



	#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires
<input type="checkbox"/>	1	fkrecetteID 	int(11)			Non	Aucun(e)	
<input type="checkbox"/>	2	fkingredientID 	int(11)			Non	Aucun(e)	
<input type="checkbox"/>	3	quantite	int(11)			Non	Aucun(e)	
<input type="checkbox"/>	4	mesure	varchar(20)	utf8_general_ci		Non	Aucun(e)	

Figure 3 : Exemple de table : recette\_ingredients

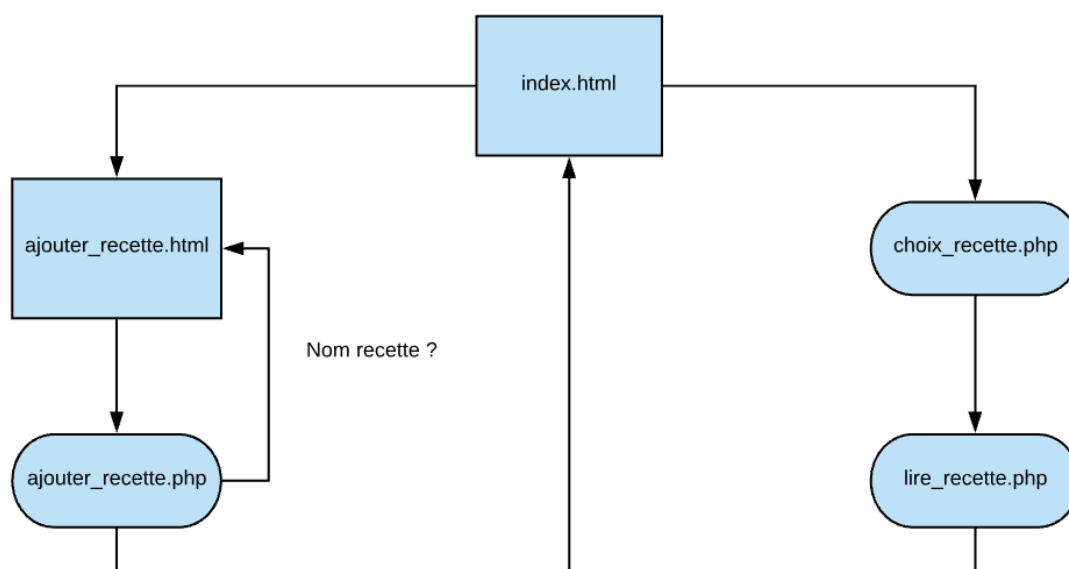
Si nous prenons l'exemple ci-dessus, tiré de la table « recette\_ingredients », nous pouvons voir que les deux clés primaires sont toutes deux des clés étrangères : « fkrecetteID » et « fkingredientID ». Il est nécessaire dans ce cas précis d'avoir deux clés primaires afin d'éviter la redondance, car un ingrédient peut être présent dans plusieurs recettes. Par ailleurs, on ajoute encore les attributs « quantité » et « mesure » qui permettent d'indiquer dans quelle quantité et avec quelle mesure l'ingrédient sera utilisé dans la recette.

# 4

## Structure et rendu visuel

Le chapitre 4 présente la structure et le rendu visuel final du site du point de vue d'un utilisateur lambda.

### 4.1 Structure du site



**Figure 4 :** *Structure du site*

Nous pouvons voir ci-dessus (fig. 4) de quelle façon est structuré le site et, plus précisément, comment les pages sont connectées entre elles. J'ai décidé de séparer les pages PHP et HTML, car il faut préciser ici qu'une page PHP peut contenir du code HTML ou CSS afin de structurer et d'améliorer le rendu visuel. Mes pages HTML n'utilisent quant à elles pas de PHP.

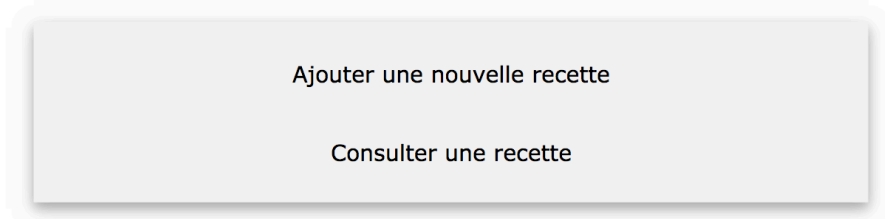
J'ai également choisi, en plus de ces 5 pages de rendu visuel, de créer un fichier PHP qui regroupe différentes fonctions utiles au bon fonctionnement de mon site : *fonctions.php*, ainsi qu'un autre fichier, *connect.php*, qui établit la connexion entre la base de données et le site. Ces deux fichiers feront objet d'une étude plus approfondie dans la suite de mon travail, au chapitre 5.

## 4.2 Les différentes pages

### 4.2.1 Index.html

*Index.html* est la page d'accueil du site, grâce à laquelle l'utilisateur peut choisir s'il veut ajouter une nouvelle recette ou en consulter une.

#### Ma base de donnée de recettes de cuisine



**Figure 5 :** *index.html*

### 4.2.2 Ajouter\_recette.html

En cliquant sur « ajouter une nouvelle recette », l'utilisateur est redirigé vers un formulaire d'ajout de recette (voir la fig. 6 à la page suivante). Ce formulaire présente uniquement du HTML/CSS et se contente d'envoyer les informations à un autre fichier PHP : *ajouter\_recette.php*.

Nouvelle recette

Votre pseudo :

Arnaud

Nom de la recette :

Fondant au chocolat maison

Difficulté :

très facile

Catégorie :

apéritif

Prix :

bon marché

Temps de préparation :

00:50

Ingrédients :

Quantité :	Mesure :	Ingrédient :
130	g de	chocolat
130	g de	beurre
130	g de	sucre
60	g de	farine

Etapes :

Faire fondre le chocolat dans un peu d'eau ou de lait. Une fois le chocolat fondu, ajoutez le beurre.

Une fois le tout fondu et mélangé, ajoutez le sucre.

Lorsque le mélange est homogène, ajoutez les jaunes des 3 oeufs un à un, puis enfin la farine.

Laissez reposer un peu, puis ajouter les blancs des 3 oeufs montés en neige.

Faites cuire à 165 degrés (thermostat 5/6) pendant 35 mn.

Ajouter

Figure 6 : *ajouter\_recette.html*

### 4.2.3 Ajouter\_recette.php

*Ajouter\_recette.php* est une page PHP permettant de stocker les différents éléments récoltés dans *ajouter\_recette.html*. Cette page inclut un autre fichier PHP, *fonction.php*, qui permet d'accéder à diverses fonctions utiles, comme par exemple *addIngredient* qui vérifie si l'ingrédient est déjà dans la base de données avant de l'ajouter. Certaines de ces fonctions seront étudiées dans le prochain chapitre.

Cette page contrôle si la recette ajoutée par l'utilisateur existe déjà dans la base de donnée. Elle génère, en fonction de l'action de l'utilisateur, deux types de messages (fig. 7 et 8) :

Cette recette existe déjà, veuillez choisir un autre nom ou en créer une autre.

Retour

Figure 7 : *ajouter\_recette.php* : option 1

Merci, votre recette a bien été ajoutée. Vous pouvez maintenant consulter vos recettes.

Retour à l'accueil

Figure 8 : *ajouter\_recette.php* : option 2

### 4.2.4 Choix\_recette.php

Cette page permet simplement de choisir la recette que l'on veut consulter. Elle génère un menu déroulant avec toutes les recettes de la BDD. Lorsque l'utilisateur clique sur « consulter », le formulaire est envoyé à *lire\_recette.php* qui affichera la recette.

Choix de ma recette :

Fondant au chocolat maison ▼

Consulter

Figure 9 : *choix\_recette.php*

#### 4.2.5 Lire\_recette.php

Ce fichier PHP génère une page HTML correspondant à la recette sélectionnée dans la page précédente. Il va directement rechercher les informations dans la base de données, puis les affiche sous une forme structurée. Après avoir consulté la page, l'utilisateur peut revenir sur la page d'accueil afin de consulter une nouvelle recette ou afin d'en ajouter une autre.

## Fondant au chocolat maison

- **Crée par :** Arnaud
- **Catégorie :** apéritif
- **Difficulté :** très facile
- **Coût :** bon marché
- **Duree :** 00:50:00
- **Ingrédients nécessaires :**
  - 130 g de chocolat
  - 130 g de beurre
  - 130 g de sucre
  - 60 g de farine
- **Etapes à suivre :**
  1. Faire fondre le chocolat dans un peu d'eau ou de lait. Une fois le chocolat fondu, ajoutez le beurre.
  2. Une fois le tout fondu et mélangé, ajoutez le sucre.
  3. Lorsque le mélange est homogène, ajoutez les jaunes des 3 oeufs un à un, puis enfin la farine.
  4. Laissez reposer un peu, puis ajouter les blancs des 3 oeufs montés en neige.
  5. Faites cuire à 165°C (thermostat 5/6) pendant 35 mn.

[Retour à l'accueil](#)

Figure 10 : *lire\_recette.php*

# 5

## Éléments de programmation

### 5.1 Technologie utilisée

En ce qui concerne les technologies et langage utilisés, nous pouvons regrouper dans un premier temps le CSS/HTML, qui permet de s'occuper de toute la partie graphique du site web, le HTML pour la forme, et le CSS pour le style et le fond.

Ensuite, nous avons utilisé PHP myadmin afin de créer la structure de la base de données et définir les relations entre les différentes tables dans le but de stocker les éléments des recettes.

Nous avons en outre utilisé le langage PHP pour gérer la connexion entre le site et la BDD, et finalement SQL pour interagir avec la base de données.

### 5.2 Connexion à la base de données

Afin que l'utilisateur puisse consulter des recettes ou en insérer des nouvelles en ligne, il est nécessaire que le site soit connecté directement à la base de données.

Il existe pour ce faire plusieurs façons d'opérer. Dans le cadre de ce travail, j'ai choisi d'utiliser la méthode PDO (PHP Data Object) qui, en plus d'être une méthode en pleine expansion, demeure plus flexible que les autres puisqu'elle s'adapte à n'importe quel type de SGBD (MySQL, Oracle, etc.) et est orientée objet (évite la redondance, présente une meilleure structure, etc.).

```
1  <?php
2  try{
3      $bdd = new
        PDO('mysql:host=localhost;dbname=Recette','root','root');
4
5      $bdd->setAttribute(PDO::ATTR_ERRMODE,
        PDO::ERRMODE_EXCEPTION);
6
7  } catch (PDOException $e){
8      echo 'Problème de connexion';
9      die();
10 }
```

**Code 1:** *connection.php*

L'une des manières possibles d'effectuer cette connexion est présentée ci-dessus (code 1). Cette connexion PDO demande 4 arguments :



- 
- Nom de l'hôte : localhost
  - Nom de la BDD : Recette
  - Nom de l'utilisateur : root
  - Mot de passe : root

Toutes ces informations sont directement visibles depuis la page d'accueil de PHP myadmin.

## 5.3 Exemples de code

Dans cette section, nous allons passer en revue quelques morceaux de code qui nous semblent particulièrement importants. Pour commencer, nous verrons 2 pages PHP : *ajouter\_recette.php* (insertion de données) et *lire\_recette.php* (lecture de données). Puis, dans un deuxième temps, nous nous pencherons sur 3 fonctions intéressantes du fichier annexe *fonctions.php*.

### 5.3.1 Ajouter\_recette.php

*Ajouter\_recette.php* accède premièrement aux différentes valeurs envoyées par le formulaire rempli par l'utilisateur, puis les stocke dans la BDD.

Avant de récupérer les valeurs du formulaire, par précaution, on applique la fonction `addslashes($array)` (code 2) à toutes les valeurs récupérées. Cette fonction ajoute des slashes devant tous les caractères spéciaux afin d'éviter de faire une erreur lorsque l'on en utilise.

```
1  <?php
2  include ("fonctions.php");
3  $input_arr = array();
4  foreach ($_POST as $key => $input_arr) {
5  $_POST[$key] = addslashes($input_arr);
6  }
```

**Code 2 : *addslashes***

Ensuite, on prend chaque élément du formulaire, puis on transfère ces éléments dans une nouvelle variable afin de pouvoir les utiliser plus tard. On accède aux données du formulaire avec `$_POST['nom_ID']` (code 3).

```

7     $recetteNom = $_POST['recetteNom'];
8     $prixID = $_POST['prixID'];
9     $categorieID = $_POST['categorieID'];
10    $recetteDuree = $_POST['recetteDuree'];
11    $difficulteID = $_POST['difficulteID'];
12    $etapes=array($_POST['etape1'],$_POST['etape2'],$_POST['etape3'],
$_POST['etape4'],$_POST['etape5'],$_POST['etape6'],$_POST['etape7'],
$_POST['etape8'],$_POST['etape9']);
13    $userPseudo = $_POST['userPseudo'];
14    $ingrédients =
array($_POST['ingr1'],$_POST['ingr2'],$_POST['ingr3'],$_POST['ingr4'],
$_POST['ingr5'],$_POST['ingr6'],$_POST['ingr7'],$_POST['ingr8']);
15    $quantite=
array($_POST['quantite1'],$_POST['quantite2'],$_POST['quantite3'],
$_POST['quantite4'],$_POST['quantite5'],$_POST['quantite6'],
$_POST['quantite7'],$_POST['quantite8']);
16    $mesure=
array($_POST['mesure1'],$_POST['mesure2'],$_POST['mesure3'],
$_POST['mesure4'],$_POST['mesure5'],$_POST['mesure6'],$_POST['mesure7'],
$_POST['mesure8']);

```

**Code 3 : récolte données**

On s'assure ensuite que le nom de la nouvelle recette n'existe pas encore avec la fonction `compareWithRecette($nom_de_la_recette,$bdd` (code 4). Si la fonction « return true », la page PHP envoie un bouton pour revenir à l'écran d'ajout de recette.

```

17    <?php
18
19    if (compareWithRecette($recetteNom,$bdd)) {
20        ?>
21        <form action="ajouterRecette.html" >
22            <p>Cette recette existe déjà, veuillez choisir un autre
nom ou en créer une autre.</p>
23            <p><button class="w3-button w3-hover-red"
type="submit">Retour</button></p>
24            </form>
25            <?php
26                return;
27        }

```

**Code 4 : compareWithRecette**

Comme la recette n'existe pas encore, plusieurs fonctions comme `addUser` ou `addEtapes` vont ajouter les différents éléments du formulaire dans la BDD.

Finalement, si tout s'est correctement déroulé, l'utilisateur peut choisir de revenir à l'index pour consulter les recettes où il peut en ajouter d'autres (code 5).

```

28     addUser($userPseudo,$bdd);
29     addrecette($recetteNom,$prixID,$recetteDuree,$difficulteID,
    $categorieID,$bdd);
30     $count = $bdd->query("SELECT COUNT(recetteID) FROM recette")->fetchColumn();
31     addEtapes($etapes,$count,$bdd);
32     addRecetteIngredients($ingredients,$quantite,$mesure,$bdd);
33
34     ?>
35     <form action="index.html">
36     <p>Merci, votre recette a bien été ajoutée. Vous pouvez maintenant consulter vos recettes.</p>
37     <p><button class="w3-button w3-hover-red" type="submit">Retour à l'accueil</button></p>
38     </form>
39
40 </div>
41 </body>

```

**Code 5 : ajout recette**

### 5.3.2 Lire\_recette.php

On récupère tout d'abord chaque élément de la recette que l'utilisateur veut consulter à l'aide de requêtes SQL, puis on les stocke dans des variables. On utilise ensuite un mélange de HTML/CSS pour organiser et inscrire à l'écran tout ce qui ne se rapporte pas directement aux variables stockées, et de PHP pour invoquer les différents éléments qui dépendent des variables (code 6).

```

1     $recetteID = $_POST['recetteID'];
2     $getRecetteNom = $bdd->query("SELECT recetteNom FROM recette WHERE recetteID = $recetteID")->fetchColumn();
3     $getIngredients = $bdd->query("SELECT ingredientNom, quantite, mesure FROM ingredient, recette, recette_ingredients WHERE (ingredientID = fkingredientID) AND (recetteID = fkrecetteID) AND (recetteID = $recetteID)");
4     $getCaract = $bdd->query("SELECT difficulteNom, categorieNom, userPseudo, prixNom, recetteDuree FROM recette, difficulte, prix, categorie, user WHERE categorieID = fkcategorieID AND userID = fkuserID AND difficulteID = fkdifficulteID AND prixID = fkprixID AND recetteID = $recetteID");
5     $caract = $getCaract->fetch();
6     $getEtapes = $bdd->query("SELECT etapeDesc FROM etapes, recette WHERE fkrecetteID = recetteID AND recetteID = $recetteID");

```

```

7      <h2 class = "w3-center" style="text-shadow:1px 1px 0 #444">
      <?php echo $getRecetteNom; ?> </h2>
8      <ul>
9          <li><strong> Crée par : </strong> <?php echo
          $caract['userPseudo']; ?> </li>
10         <li><strong> Catégorie : </strong> <?php echo
          $caract['categorieNom']; ?> </li>
11         <li><strong> Difficulte : </strong> <?php echo
          $caract['difficulteNom']; ?> </li>
12         <li><strong> Coût : </strong> <?php echo
          $caract['prixNom']; ?> </li>
13         <li><strong> Duree : </strong> <?php echo
          $caract['recetteDuree']; ?> </li><br>
14
15
16         <li><strong> Ingrédients nécessaires :</strong></li>
17         <ul>
18             <?php while ($r = $getIngredients->fetch()) { ?>
19                 <li> <?php echo $r['quantite'].
                ".$r['mesure']. " ".$r['ingredientNom']; ?> </li>
20                 <?php } ?>
21             </ul><br>
22         <li><strong> Etapes à suivre : </strong> </li>
23         <ol>
24             <?php while ($r = $getEtapes->fetch()){ ?>
25                 <li> <?php echo $r['etapeDesc']; ?> </li>
26                 <?php } ?>
27             </ol>
28     </ul>
29     <form method ="POST" action="index.html" style = "margin-
    top: 40px;">
30         <button class="w3-button w3-hover-red" type
        ="submit">Retour à l'accueil</button>
31     </form>

```

**Code 6 : lire\_recette.php**

### 5.3.3 choix\_recette.php

Cette page, comme mentionné au chapitre 4, permet de choisir la recette que l'on veut consulter.

On commence par inclure la connexion à la base de donnée avec le mot `include()`. Dans un deuxième temps, on récupère toutes les recettes dans la variable `$getRecettes`. Ensuite, sous la forme d'une liste déroulante, on va récupérer chaque valeur de la variable `$getRecettes` en utilisant une boucle `while` (code 7). A noter qu'à chaque fois que l'on veut utiliser une variable directement en lien avec la base de données, il faut passer par le langage PHP.

```

1  <?php
2  include ("connect.php");
3
4  $getRecettes = $bdd->query("SELECT recetteID,recetteNom
5  FROM recette");
6
7  ?>
8
9  <link rel="stylesheet"
10 href="https://www.w3schools.com/w3css/4/w3.css">
11
12 <body class="w3-container w3-card-4 w3-light-grey w3-
13 display-middle w3-center">
14 <form action ="lire_recette.php" method ="post">
15 <h4><label for="recette">Choix de ma recette :</label></h4>
16 <select id="recetteID" name="recetteID">
17 <?php while ($r = $getRecettes->fetch()) { ?>
18 <option value="<?php echo $r['recetteID']; ?>"> <?php echo
19 $r['recetteNom']; ?> </option>
20 <?php } ?>
21 </select>
22 <button type="submit">Consulter </button>
23
24 </form>
25 </body>

```

**Code 7 : choix\_recette.php**

### 5.3.4 Fonctions PHP

#### Ajouter un ingrédient

La fonction `addIngredient`, qui permet d'ajouter un ingrédient, est elle-même basée sur une autre fonction nommée `compareWithIngredients($ingredientNom, $bdd)` qui se charge de regarder si l'ingrédient existe déjà. Pour ce faire, elle stocke tous les ingrédients sous la forme d'un tableau, puis compare chaque nom avec celui que l'on veut ajouter grâce à la méthode `strcasecmp`. Cette dernière permet la comparaison insensible à la casse de chaînes binaires, c'est-à-dire qu'elle ne différencie pas les mots en majuscules des mots sans majuscules. Elle retourne 0 si les deux noms sont égaux. Dans notre cas, si la méthode `compareWithIngredients` « return true », l'ingrédient n'est pas ajouté une deuxième fois.

Si l'ingrédient n'existe pas encore, on crée une requête SQL, qui insère dans « ingredient » le nouvel ingrédient. A noter que pour insérer un nouvel élément, on utilise le mot `exec` (ligne 89 code 8).

```

1  function addIngredient($ingredientNom,$bdd) {
2      if (compareWithIngredients($ingredientNom,$bdd)) {
3          return;
4      }
5      $query = $bdd->query("SELECT COUNT(ingredientID) FROM
6      ingredient")->fetchColumn();
7      $sql = "INSERT INTO
8      ingredient(ingredientID,ingredientNom) VALUES
9      (($query+1), '$ingredientNom')";
10     $bdd->exec($sql);
11 }

```

```

9     function compareWithIngredients($ingredientNom,$bdd)
10    {
11        $query = $bdd->query("SELECT * FROM ingredient");
12        foreach ($query as $r) {
13            if(strcasecmp($r['ingredientNom'],$ingredientNom)
14            == 0){
15                return TRUE;
16            }
17        }
18        return FALSE;
19    }

```

**Code 8 :** *addIngredient*

### Associer des ingrédients à une recette

Afin d'associer des ingrédients à une recette, il est tout d'abord important d'avoir une fonction qui ajoute un seul ingrédient à une recette. Cette fonction prend en arguments à la fois le nom de l'ingrédient, la quantité utilisée, la mesure et finalement, la base de données.

Cette fonction ajoute l'ingrédient à la dernière recette créée. Elle commence par compter le nombre de recettes, puis stocke ce nombre dans `$recetteID`. Elle ajoute ensuite l'ingrédient à la base de données avec la méthode `addIngredient`. Puis, une recette SQL cherche l'ID du dernier ingrédient ajouté (c'est-à-dire celui qui vient d'être ajouté). On s'assure ensuite avec la méthode `ingredientAlreadyInRecette` que l'ingrédient n'est pas ajouté deux fois dans une recette. Finalement, on ajoute l'ingrédient et la recette dans la table `recette_ingredients` avec la quantité et la mesure choisies.

Cependant, invoquer cette méthode x fois le nombre d'ingrédients que l'on veut insérer un élément ne serait pas élégante. Meilleure, une méthode `addRecetteIngredients` permet d'éviter cette redondance. Cette dernière prend cette fois-ci 3 « arrays » en arguments : ingrédients, quantités et mesures, ainsi que la base de données.

La méthode itère selon le nombre d'ingrédients qu'il y a dans le tableau d'ingrédients donné en paramètre. A l'intérieur de la boucle `for`, on contrôle que chaque ingrédient a un nom et qu'à chaque fois une quantité a été inscrite. La mesure n'est pas contrôlée, car un ingrédient peut ne pas avoir de mesure (ex : une pomme). Si tout est respecté, la méthode `addRecetteIngrédient` est invoquée (code 9).

---

```

1  function
   addRecetteIngredient($ingrNom,$quantite,$measure,$bdd){
2
3      $recetteID = $bdd->query("SELECT COUNT(recetteID) FROM
recette")->fetchColumn();
4      addingredient($ingrNom,$bdd);
5      $ingredientID = $bdd->query("SELECT ingredientID from
ingredient WHERE ingredientNom = '$ingrNom'")-
>fetchColumn();
6
   if(ingredientAlreadyInRecette($ingredientID,$recetteID,$bdd
   )){
7          return;
8      }
9      $bdd->exec("INSERT INTO
recette_ingredients(fkrecetteID,fkingredientID,quantite,mes
ure) VALUES
($recetteID,$ingredientID,$quantite,'$measure')");
10
11 }
12 function
   addRecetteIngredients($ingredients,$quantites,$mesures,$bdd
   ){
13     for($i = 0;$i<sizeof($ingredients);$i++){
14         if ($ingredients[$i]==""){return;}
15         if ($quantites[$i]==null){return;}
16
17         addRecetteIngredient($ingredients[$i],$quantites[$i],$mesur
es[$i],$bdd);
18     }

```

**Code 9 :** *addRecetteIngredients*

# 6

## Conclusion

Ce travail m'a premièrement permis de créer un programme opérationnel et utile à mes proches et amis, qui peuvent en effet dorénavant stocker et consulter leurs propres recettes. Surtout, j'ai grâce à lui eu l'occasion d'approfondir mes connaissances en matière de langages de programmation, tout en me familiarisant aussi le fonctionnement d'un site internet et la connexion entre pages.

Si ce travail constitue une bonne entrée en matière, plusieurs fonctions pourraient par la suite être également ajoutées au programme : (1) un système de login « (user,password) » avec un onglet permettant de consulter ses propres recettes et de choisir d'en partager avec d'autres utilisateurs ; (2) un système de notes et de commentaires, afin de proposer, par exemple, des améliorations aux autres utilisateurs ; (3) la mise en place d'un système de calcul des quantités en fonction du nombre de personnes choisi.

Enfin, une autre extension possible de ce travail pourrait consister en la création d'un système de recherche de recettes en fonction des ingrédients. Il deviendrait ainsi imaginable de pouvoir trier ses recettes selon son régime alimentaire (par exemple, sans gluten ou sans viande). Finalement, il serait également intéressant de pouvoir envisager l'ajout d'images et/ou de vidéos dans les recettes.



# Bibliographie

## **[Openclassrooms, 2018]**

Concevez votre site web avec PHP et MySQL,

<https://openclassrooms.com/fr/courses/918836-concevez-votre-site-web-avec-php-et-mysql>, [Dernière visite : 20.06.2018]

Apprenez à créer votre site web avec html5 et css3,

<https://openclassrooms.com/fr/courses/1603881-apprenez-a-creer-votre-site-web-avec-html5-et-css3>, [Dernière visite : 19.07.2018]

Avantages de pdo, <https://openclassrooms.com/forum/sujet/avantages-de-pdo-12464>, [Dernière visite : 05.08.2018]

## **[Php, 2018]**

PHP manual, <http://php.net/manual/en>, [Dernière visite : 25.07.2018]

## **[w3schools, 2018]**

CSS style, <https://www.w3schools.com/Css/default.asp>, [Dernière visite : 18.07.2018]

## **[Marmiton, 2018]**

Exemples recettes, <https://www.marmiton.org/recettes/>, [Dernière visite : 10.05.2018]

## **[Commentcamarche, 2018]**

Récupérer valeurs formulaire, <https://www.commentcamarche.net/forum/affich-8720699-if-isset-ou-post>, [Dernière visite: 20.06.2018]

## **[lucidchart, 2018]**

Schéma entité-relations, <https://www.lucidchart.com/>, [Dernière visite : 12.07.2018]

## **[Snipplr, 2018]**

Add slashes to post variables, <https://snipplr.com/view/7872/addslashes-automatically-to-post-variables/>, [Dernière visite : 28.06.2018]