

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



# Site d'information concernant la médecine douce

## Un prototype écrit avec PHP/MySQL

**Jérémy Gaudin**

No étudiant : 14-413-116

**Travail de séminaire en Informatique de Gestion**

**Encadré par :**

**Prof. Dr. Jacques Pasquier – Rocha**

**Johan Jobin**



**Fribourg, octobre 2017**

# Table des matières

<b>Table des matières</b> .....	<b>i</b>
<b>Liste des figures</b> .....	<b>ii</b>
<b>Liste des codes</b> .....	<b>iii</b>
<b>1 Introduction</b> .....	<b>1</b>
<b>2 Domaine d'application</b> .....	<b>2</b>
<b>3 Modélisation des données</b> .....	<b>3</b>
3.1 Le schéma entité-association.....	3
3.2 Transformation en tables relationnelles .....	5
3.3 Les tables.....	6
3.3.1 Personne.....	6
3.3.2 Patient.....	6
3.3.3 Therapeute.....	6
3.3.4 Critique.....	7
3.3.5 Est_specialist.....	7
3.3.6 Specialite .....	7
3.3.7 Affectation.....	7
3.3.8 Cabinet .....	8
3.3.9 Adresse.....	8
3.3.10 Localite.....	8
<b>4 Le site d'un point de vue de l'utilisateur final et éléments de programmation</b> .....	<b>9</b>
4.1 L'apparence.....	9
4.2 Les fonctions .....	11
<b>5 Conclusion</b> .....	<b>17</b>
<b>A Fichier .zip du code source</b> .....	<b>18</b>
<b>Bibliographie</b> .....	<b>19</b>

# Liste des figures

Figure 1 : Modèle entité-association .....	3
Figure 2 : Diagramme relationnel .....	5
Figure 3 : La table Personne.....	6
Figure 4 : La table Patient .....	6
Figure 5 : La table Therapeute .....	6
Figure 6 : La table Critique .....	7
Figure 7: La table Est_specialist .....	7
Figure 8 : La table Specialite.....	7
Figure 9 : La table Affectation .....	7
Figure 10 : La table Cabinet.....	8
Figure 11 : La Table Adresse .....	8
Figure 12 : La table Localite .....	8
Figure 13 : Page 1.....	9
Figure 14 : Page 2.....	10
Figure 15 : Page 3.....	10
Figure 16 : Page 4.....	11

# Liste des codes

Code 1 : Code source de page2.php .....	12
Code 2 : Code source de page3.php .....	14
Code 3 : Code source de traitement.php .....	15
Code 4 : Code source de page4.php .....	16

# 1

## Introduction

A travers le monde, les gens sont de plus en plus connectés. Les technologies de l'information ont de plus en plus d'importance. Les entreprises doivent, afin d'être visible aux yeux du grand public, s'adapter à la numérisation des informations. Les clients utilisent internet comme canal d'information.

Ce travail de séminaire a pour but d'amener une vitrine plus moderne aux différentes médecines douces en répertoriant celles-ci dans un site. Le site permet de choisir plus facilement et en toute confiance un thérapeute dans notre région grâce à la possibilité de feedbacks des différents patients. Cela permettrait d'augmenter la confiance dans ces médecines moins traditionnelles.

Afin de construire le site, différentes étapes vont être accomplies. Le chapitre 2 présentera le champ d'application. Le chapitre 3 se concentrera sur la modélisation des données, il contiendra la présentation du modèle entité-association et la traduction de celui-ci en schéma relationnel. Le chapitre 4 fournira l'explication du site pour les personnes désirant s'informer sur les possibilités de médecine douce dans sa région.

# 2

## Domaine d'application

Le thème choisi pour ce travail est un site regroupant différents thérapeutes proposant la médecine douce. La volonté principale est celle de la transparence. Les médecines douces n'ayant pas toujours une patente légale, il est difficile pour le patient de s'y retrouver dans les différentes offres entre les thérapeutes professionnels et les charlatans.

La plateforme a pour but de permettre d'éliminer les thérapeutes de mauvaises qualités grâce au système de critique. Il réduirait les coûts de communication à travers le regroupement qui diviserait les coûts entre les différents professionnels. La communication moderne amènerait à réduire les coûts des communications plus anciennes et coûteuses tel que les journaux, la radio et la distribution de flyers. Non seulement les coûts vont baisser mais avec la révolution numérique, il découlerait une meilleure visibilité. Cependant, il ne faut pas complètement arrêter les moyens de communications habituels sous peine de perdre les patients non connectés.

# 3

## Modélisation des données

Dans ce chapitre, nous allons observer la structure de la base de données *therapeute*. Tout d'abord, je vais présenter le schéma entité-association qui est la première étape de la création d'une base de données. Nous allons par la suite détailler ce modèle afin de comprendre les différents attributs choisis. On prendra connaissance des clés primaires et étrangères afin de comprendre le lien entre les différentes tables

### 3.1 Le schéma entité-association

Le schéma entité-association, présenté à la figure 1, est la première étape pratique afin de créer une base de données

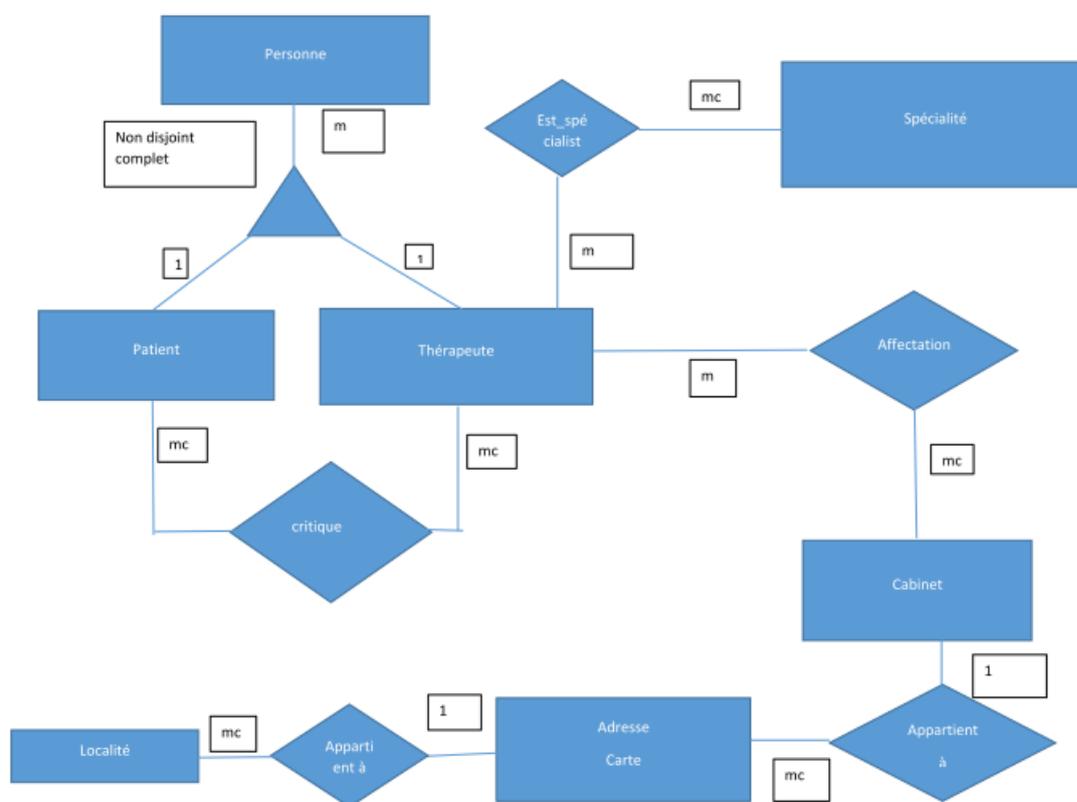


Figure 1 : Modèle entité-association

---

En me basant sur les standards présentés par [Andreas Meyer], j'ai construit le schéma avec Microsoft Word afin d'avoir une vue générale de ma base de données *therapeute*. Le reste du travail a dû être basé sur ce schéma afin de garder une même ligne directrice.

Afin de comprendre, nous allons interpréter de manière textuelle mon modèle :

- Une personne peut être soit un thérapeute, soit un patient ou les deux.
- Un patient peut faire plusieurs ou aucune critique.
- Un thérapeute peut recevoir plusieurs ou aucune critique.
- Un thérapeute a au moins une spécialité.
- Une spécialité peut avoir plusieurs ou aucun thérapeute.
- Un thérapeute a au moins un cabinet.
- Un cabinet a plusieurs ou aucun thérapeute.
- Un cabinet a une adresse.
- Une adresse peut avoir plusieurs ou aucun cabinet.
- Une adresse a une seule localité.
- Une localité peut avoir plusieurs ou aucune adresse.

## 3.2 Transformation en tables relationnelles

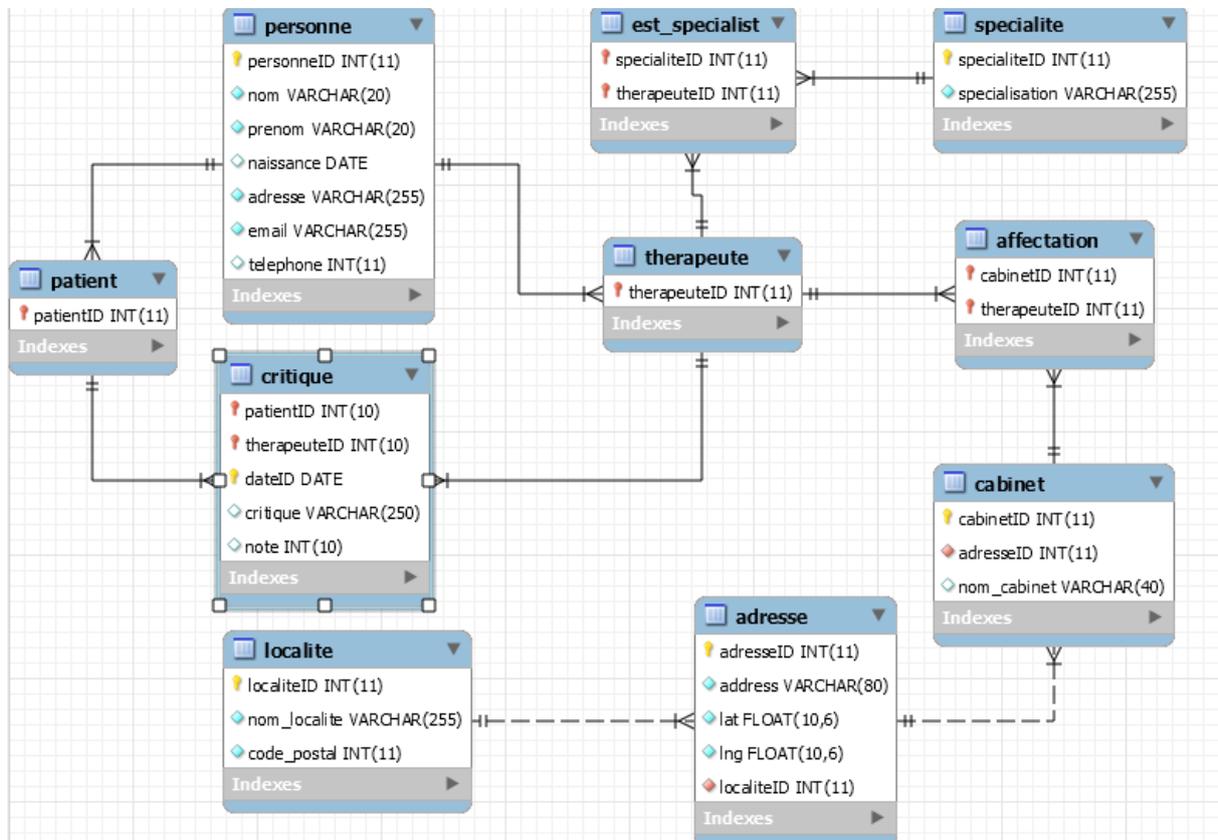


Figure 2 : Diagramme relationnel

Après avoir fait le schéma entité-relationnel (voir figure 1), il est nécessaire de créer des tables sur phpMYAdmin grâce à l'utilisation de [MAMP] en les liant à travers les clés primaires et les clés secondaires. Dans la figure 2, j'ai dû créer une table pour chaque entité et en rajouter 3 en plus lors des relations multiples-multiples.

Une clé primaire est la donnée qui permet d'identifier de manière unique un enregistrement dans une table. Une clé primaire peut être composée d'une ou de plusieurs colonnes de la table. Deux lignes distinctes de la table ne peuvent pas avoir les mêmes valeurs dans les colonnes définies comme clé primaire. Il est possible de définir pour une même table plusieurs contraintes d'unicité, mais au plus une seule clé primaire. Une clé primaire est choisie parmi les clés candidates. Suivant les cas il peut être nécessaire ou préférable d'utiliser une clé artificielle ajoutée aux données comme clé primaire. [Wikipedia a]

Une clé étrangère identifie une colonne ou un ensemble de colonnes d'une table comme référençant une colonne ou un ensemble de colonnes d'une autre table (la table référencée). Les colonnes de la table référencée doivent faire partie d'une contrainte de clé primaire ou d'une contrainte d'unicité. La contrainte de clé étrangère garantit que les valeurs de chaque ligne de la table référençant existent dans la table référencée : ainsi une ligne de la table référençant ne peut pas contenir un ensemble de valeurs qui n'existe pas dans la table référencée. [Wikipedia b]

Après la création des tables et la liaison entre elles, j'ai utilisé MySQLWorkbench [MySQL Workbench, 2017] afin de mieux visualiser la base de données.

### 3.3 Les tables

#### 3.3.1 Personne

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/> 1	personneID	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Primaire Plus
<input type="checkbox"/> 2	nom	varchar(20)	latin1_swedish_ci		Non	Aucun(e)			Modifier Supprimer Primaire Plus
<input type="checkbox"/> 3	prenom	varchar(20)	latin1_swedish_ci		Non	Aucun(e)			Modifier Supprimer Primaire Plus
<input type="checkbox"/> 4	naissance	date			Oui	NULL			Modifier Supprimer Primaire Plus
<input type="checkbox"/> 5	adresse	varchar(255)	latin1_swedish_ci		Non	Aucun(e)			Modifier Supprimer Primaire Plus
<input type="checkbox"/> 6	email	varchar(255)	latin1_swedish_ci		Non	Aucun(e)			Modifier Supprimer Primaire Plus
<input type="checkbox"/> 7	telephone	int(11)			Oui	NULL			Modifier Supprimer Primaire Plus

Figure 3 : La table Personne

La création de table personne, présentée dans la Figure 3, permet de regrouper tous les attributs en commun des tables *Patient* et *Therapeute*, afin de ne pas avoir à double les attributs communs. Le clé primaire *personneID* donne un identifiant unique grâce à l'auto incrémentation. Tous les autres attributs définissent les personnes à travers leur nom, prénom, naissance, adresse, email et téléphone.

#### 3.3.2 Patient

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/> 1	patientID	int(11)			Non	Aucun(e)			Modifier Supprimer Primaire Unique Index Plus

Figure 4 : La table Patient

Dans cette table *Patient*, je référence tous les patients. Nous pouvons voir que dans la Figure 4, la clé primaire et la clé étrangère sont dans ce cas la même. *PatientID* permet l'identification des personnes en tant que patients.

#### 3.3.3 Therapeute

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/> 1	therapeuteID	int(11)			Non	Aucun(e)			Modifier Supprimer Primaire Unique Index Plus

Figure 5 : La table Therapeute

Dans cette table *Therapeute*, présentée à la Figure 5, je référence tous les thérapeutes. La clé primaire et la clé étrangère sont dans ce cas la même. *TherapeuteID* permet l'identification des personnes en tant que thérapeutes. Il est important de souligner pour la table *Therapeute* et *Patient* qu'une personne peut être à la fois un patient et un thérapeute car c'est une spécialisation non disjoint complète.

### 3.3.4 Critique

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/> 1	patientID	int(10)			Non	Aucun(e)			Modifier Supprimer Primaire Unique Plus
<input type="checkbox"/> 2	therapeuteID	int(10)			Non	Aucun(e)			Modifier Supprimer Primaire Unique Plus
<input type="checkbox"/> 3	dateID	date			Non	Aucun(e)			Modifier Supprimer Primaire Unique Plus
<input type="checkbox"/> 4	critique	varchar(250)	latin1_swedish_ci		Oui	NULL			Modifier Supprimer Primaire Unique Plus
<input type="checkbox"/> 5	note	int(10)			Oui	NULL			Modifier Supprimer Primaire Unique Plus

Figure 6 : La table Critique

Dans la Figure 6 nous pouvons voir la table *critique*, qui a dû être créée pour satisfaire la relation multiples-multiples entre les tables *patient* et *therapeute*. Elle permet le retour des patients sur la qualité du soin prodigué par les thérapeutes. Pour cela, il a fallu avoir une clé primaire composite (*patientID*, *therapeuteID*, *dateID*) afin de garder une unicité par critique. La date permet aux patients de faire plusieurs critiques. L'attribut *critique* permet de donner son avis sous forme de texte. Il y a aussi la possibilité de mettre des notes aux thérapeutes à travers l'attribut *note*.

### 3.3.5 Est\_specialist

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/> 1	specialiteID	int(11)			Non	Aucun(e)			Modifier Supprimer Primaire Plus
<input type="checkbox"/> 2	therapeuteID	int(11)			Non	Aucun(e)			Modifier Supprimer Primaire Plus

Figure 7: La table Est\_specialist

La table *Est\_specialist*, présentée à la Figure 7, a dû être créée pour satisfaire la relation multiples-multiples entre les tables *specialite* et *therapeute*. Cette table permet de définir les spécialisations du thérapeute. La clé primaire est définie par deux clés étrangères *specialiteID* et *therapeuteID*.

### 3.3.6 Specialite

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/> 1	specialiteID	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
<input type="checkbox"/> 2	specialisation	varchar(255)	utf8_general_ci		Non	Aucun(e)			Modifier Supprimer Plus

Figure 8 : La table Specialite

La table *specialite*, présentée à la Figure 8 a pour but de définir les différentes spécialisations en médecine douce. Elle est définie par la clé primaire *specialiteID* et par l'attribut *specialisation* qui nous donne le nom de la spécialisation.

### 3.3.7 Affectation

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/> 1	cabinetID	int(11)			Non	Aucun(e)			Modifier Supprimer Primaire Plus
<input type="checkbox"/> 2	therapeuteID	int(11)			Non	Aucun(e)			Modifier Supprimer Primaire Plus

Figure 9 : La table Affectation

La table *affectation*, présentée à la Figure 9 a dû être créée pour satisfaire la relation multiples-multiples entre les tables *cabinet* et *therapeute*. Cette table permet de lier les thérapeutes à leurs cabinets. La clé primaire est définie par deux clés étrangères *cabinetID* et *therapeuteID*.

### 3.3.8 Cabinet

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/> 1	<b>cabinetID</b>	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Primaire Plus
<input type="checkbox"/> 2	<b>adresseID</b>	int(11)			Non	Aucun(e)			Modifier Supprimer Primaire Plus
<input type="checkbox"/> 3	<b>nom_cabinet</b>	varchar(40)	utf8_general_ci		Oui	NULL			Modifier Supprimer Primaire Plus

Figure 10 : La table Cabinet

La table *cabinet*, présentée à la Figure 10, permet de définir les cabinets de médecine douce. Elle est définie par la clé primaire *cabinetID*. La clé étrangère *adresseID* qui sert à référencer l'adresse au cabinet. L'attribut *nom\_cabinet* permet de nommer les cabinets.

### 3.3.9 Adresse

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/> 1	<b>adresseID</b>	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Primaire Unique Plus
<input type="checkbox"/> 2	<b>address</b>	varchar(80)	utf8_general_ci		Non	Aucun(e)			Modifier Supprimer Primaire Unique Plus
<input type="checkbox"/> 3	<b>lat</b>	float(10,6)			Non	Aucun(e)			Modifier Supprimer Primaire Unique Plus
<input type="checkbox"/> 4	<b>lng</b>	float(10,6)			Non	Aucun(e)			Modifier Supprimer Primaire Unique Plus
<input type="checkbox"/> 5	<b>localiteID</b>	int(11)			Non	Aucun(e)			Modifier Supprimer Primaire Unique Plus

Figure 11 : La Table Adresse

La table *adresse*, présentée à la Figure 11, a comme clé primaire *adresseID*. Les attributs *address*, *lat*, *ing* permettent de localiser l'adresse avec GoogleMap en liant l'adresse avec sa latitude et longitude. La clés étrangère *localiteID* sert à référencer les localités aux adresses.

### 3.3.10 Localité

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/> 1	<b>localiteID</b>	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Primaire Unique Plus
<input type="checkbox"/> 2	<b>nom_localite</b>	varchar(255)	utf8_general_ci		Non	Aucun(e)			Modifier Supprimer Primaire Unique Plus
<input type="checkbox"/> 3	<b>code_postal</b>	int(11)			Non	Aucun(e)			Modifier Supprimer Primaire Unique Plus

Figure 12 : La table Localite

La table *localite*, présentée à la Figure 12, est utilisée afin de définir la localité. Elle a *localiteID* comme clé primaire et l'attribut *nom\_localite* et *code\_postal* qui définissent une localité.

# 4

## Le site d'un point de vue de l'utilisateur final et éléments de programmation

Dans ce chapitre, nous allons expliquer l'interface du point de vue de l'utilisateur final. Il est important de simplifier la rentrée de données car tout le monde n'a pas les connaissances adéquates. Cela permet une utilisation ordinaire à travers une interface web sous forme HTML CSS et PHP.

### 4.1 L'apparence

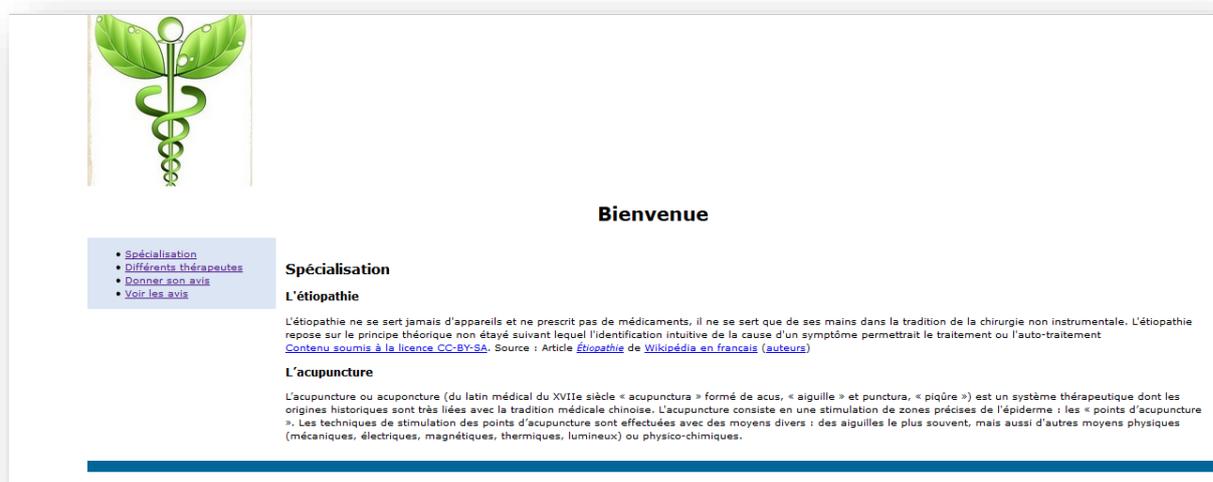
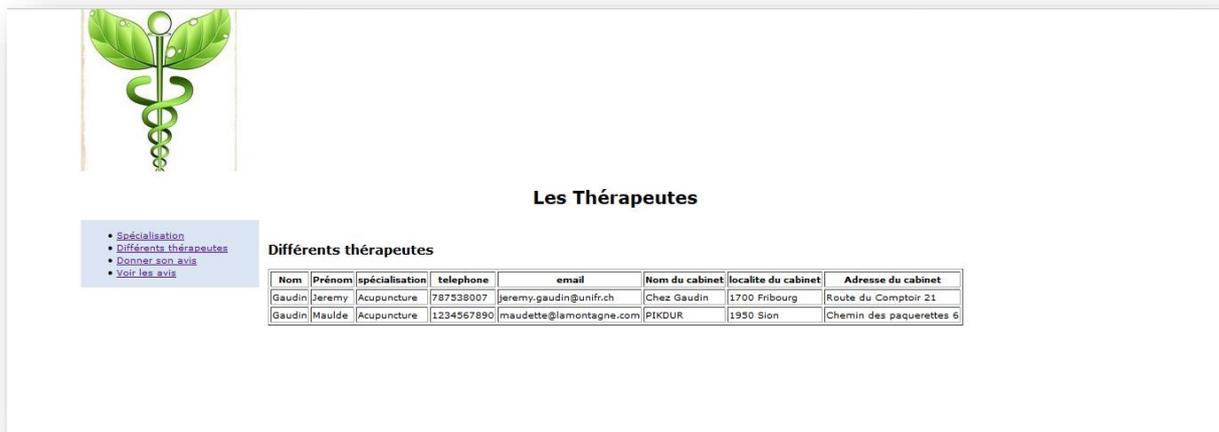


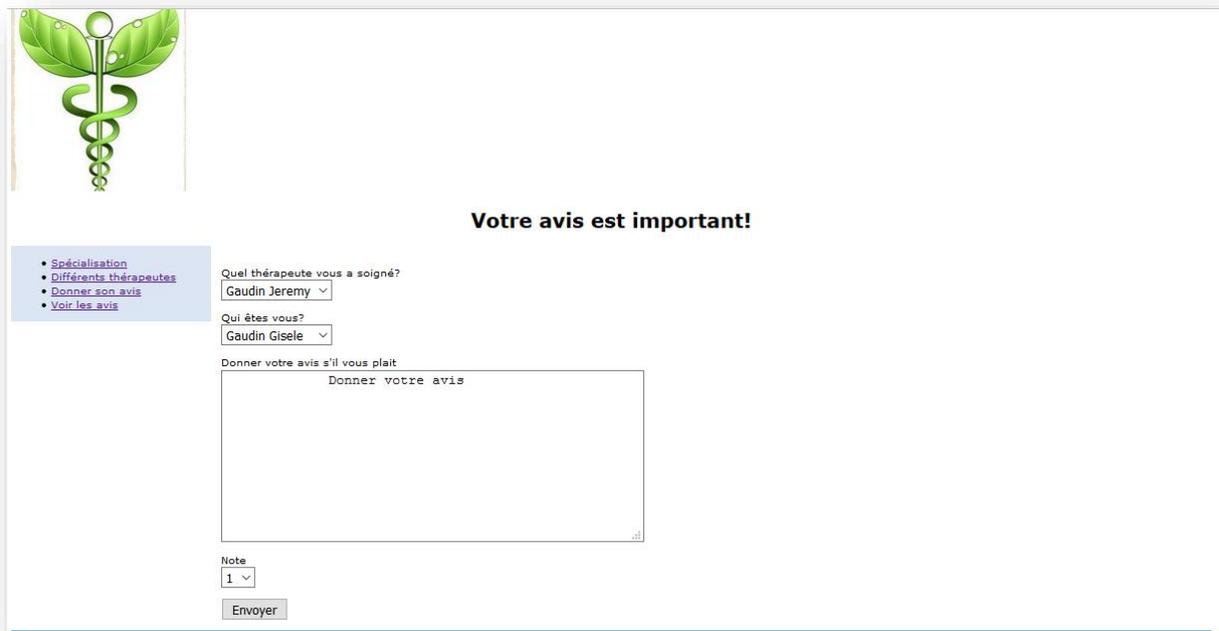
Figure 13 : Page 1

La première page d'accueil, illustrée à la Figure 13, introduit le site à travers la liste des différentes médecines existantes sur la plateforme. Elle sert surtout à rediriger, grâce au menu, l'utilisateur à travers les différentes options qu'offre le site.



**Figure 14 :** Page 2

La page 2, illustrée à la Figure 14, du site permet d’afficher les différents thérapeutes que le site répertorie. Ils nous donnent les informations qui sont le nom, le prénom, la spécialisation pratiquée, le numéro de téléphone, le nom du cabinet du praticien ainsi que l’adresse de celui-ci.



**Figure 15 :** Page 3

La page 3, illustrée à la Figure 15, permet de donner son avis. Il faut tout d’abord choisir le thérapeute qui nous a soigné dans une liste déroulante et par la suite il faut nous identifier en choisissant parmi les clients, notre nom et prénom. L’avis est donné sous forme de texte ouvert et sous forme de note allant de 1 à 5. Une fois le formulaire rempli, il faut cliquer sur le bouton envoyer afin que le formulaire soit écrit dans la base de données *therapeute*.

---

Une fois le formulaire envoyé une autre page s'ouvre avec un message signifiant que la critique a bien été prise en compte et un lien permettant de retourner au site principal.



**Figure 16 :** Page 4

La page 4, illustrée à la Figure 16, permet à l'utilisateur final de se renseigner sur les avis précédemment émis par les autres patients. Il nous est ainsi possible de choisir le thérapeute qui a les meilleures notes et qui nous convient le mieux.

## 4.2 Les fonctions

Les fonctions sont les outils qui permettent de simplifier l'utilisation des sites pour les personnes n'étant pas expert en informatique. Elles permettent d'effectuer du code SQL à travers des fonctions PHP sans que l'utilisateur sans rende compte.

```

1  <?php
2      $servername = "localhost";
3      $username = "root";
4      $password = "";
5      $dbname = "therapeute";
6
7      // Create connection
8      $conn = new mysqli($servername, $username, $password,
9      $dbname);
10     // Check connection
11     if ($conn->connect_error) {
12         die("Connection failed: " . $conn->connect_error);
13     }
14     $sql = "SELECT  personne.nom,  personne.prenom,
15     personne.email,  personne.telephone,  specialite.specialisation
16     ,  adresse.address,  cabinet.nom_cabinet,  localite.code_postal,
17     localite.nom_localite
18     from
19     specialite,personne,est_specialist,therapeute,affectation,cab
20     inet,adresse,localite
21     WHERE therapeute.therapeuteID = personne.personneID
22     and
23     therapeute.therapeuteID
24     =
25     est_specialist.therapeuteID
26     and
27     est_specialist.specialiteID
28     =
29     specialite.specialiteID
30     and affectation.therapeuteID = therapeute.therapeuteID
31     and affectation.cabinetID = cabinet.cabinetID
32     and cabinet.adresseID= adresse.adresseID
33     and localite.localiteID = adresse.localiteID ";
34     $result = $conn->query($sql);
35     if ($result->num_rows > 0) {
36         // output data of each row
37         while($row = $result->fetch_assoc()) {
38             //boucle crée grâce à fetch_assoc
39             echo "<tr> <td>". $row["nom"]."</td>",
40             "<td>" . $row["prenom"]. "</td>",
41             "<td>" . $row["specialisation"]. "</td>",
42             "<td>" . $row["telephone"]. "</td>",
43             "<td>" . $row["email"] . "</td>",
44             "<td>" . $row["nom_cabinet"]. "</td>",
45             "<td>" . $row["code_postal"] . " " . " " .
46             $row["nom_localite"] . "</td>",
47             "<td>" . $row["address"]."</td></tr>";
48         }
49     } else {
50         echo "0 results";
51     }
52     $conn->close();
53     ?>

```

**Code 1** : Code source de page2.php

Dans ce code source de la page2.php, présenté au Code, il a fallu aller chercher les différentes données concernant les thérapeutes dans les tables afin de pouvoir les afficher sous forme de tableau. Le code PHP permet d'exécuter le code SQL qui sélectionne les bonnes données avec le select. Par la suite, le ECHO écrit sous forme HTML les résultats trouvés.

```

1  <?php
2      $servername = "localhost";
3      $username = "root";
4      $password = "";
5      $dbname = "therapeute";
6
7      // Create connection
8      $conn = new mysqli($servername, $username, $password,
9      $dbname);
10     // Check connection
11     if ($conn->connect_error) {
12         die("Connection failed: " . $conn->connect_error);
13     }
14
15     $sqltherapeute = "SELECT personne.nom, personne.prenom,
16     therapeuteID
17     From personne, therapeute
18     WHERE personne.personneID = therapeute.therapeuteID";
19     $sqlpatient = "SELECT personne.nom, personne.prenom,
20     patientID
21     From personne, patient
22     WHERE personne.personneID = patient.patientID";
23
24     $result = $conn->query($sqltherapeute);
25     if ($result->num_rows > 0) {
26         // output data of each row
27
28         echo "<form method='post' action='traitement.php'>
29         <p>
30             <label for='pays'>Quel thérapeute vous a
31             soigné?</label><br />
32             <select name='therapeute' id='therapeute' >;
33
34                 while($row = $result->fetch_assoc()) {
35                     //boucle crée grâce à fetch_assoc
36
37                     echo "
38                         <option value=" .
39                         $row["therapeuteID"] . ">".
40                         $row["nom"] . "
41                         " .
42                         $row["prenom"] . "</option>";
43                 }
44
45                 echo "</select>
46
47                 </p>";
48
49     } else {
50         echo "0 results";
51     }
52
53     ?>
54
55     <?php
56     $result2 = $conn->query($sqlpatient);
57     if ($result2->num_rows > 0) {
58         // output data of each row
59
60         echo "
61         <p>
62             <label for='pays'>Qui êtes vous?</label><br />

```

```

56         <select name='patient' id='patient' >;
57
58         while($row = $result2->fetch_assoc()) {
59             //boucle crée grâce à fetch_assoc
60
61             echo          "          <option          value="".
        $row["patientID"].">".          $row["nom"].          "          "          ".
        $row["prenom"]."</option>";
62             }
63
64             echo          "</select>
65
66             </p>";
67
68
69         } else {
70             echo "0 results";
71         }
72
73         $conn->close();
74         ?>
75
76
77         <label for="ameliorer">
78         Donner votre avis s'il vous plait
79         </label>
80         <br />
81         <textarea name="critique" id="critique" rows="10"
        cols="50">
82         Donner votre avis
83         </textarea>
84
85         </p>
86
87
88
89         <p>
90         <label for='pays'>Note</label><br />
91         <select name='note' id='note' >
92             <option value="1">1</option>
93             <option value="2">2</option>
94             <option value="3">3</option>
95             <option value="4">4</option>
96             <option value="5">5</option>
97         </select>
98
99         </p>
100         <input type="submit" value="Envoyer" />
101     </form>

```

**Code 2** : Code source de page3.php

Pour créer la page3.php, l'utilisation du Code 2 a tout d'abord créé un questionnaire HTML grâce à la balise <label> et <option>. Les deux balises permettent de proposer des choix à sélectionner à l'utilisateur final. L'utilisation de la balise <textarea> a permis la création d'une zone texte.

---

Le PHP a permis d'avoir une boucle (*fetch.assoc()*) qui sélectionne les thérapeutes et les patients dans la base de données *therapeute* afin que le patient puisse se sélectionner et sélectionner la personne qui a fourni les soins.

Une fois les choix faits, l'utilisateur va utiliser le bouton Envoyer qui est de type *Submit*. Le *Submit* va permettre d'envoyer les données enregistrées à la page *traitement.php*.

```
1 <?php
2 if(isset($_POST['patient'])){
3     //construction de la resquete
4     $therapeute=$_POST['therapeute'];
5     $patient=$_POST['patient'];
6     $critique=$_POST['critique'];
7     $note=$_POST['note'];
8     $timezone = date_default_timezone_get();
9     $date = date('Y-m-d');
10    $sql = "INSERT INTO critique(patientID, therapeuteID,
11    critique, note, dateId)
12    VALUES ('".$_patient."', '".$_therapeute."',
13    '".$_critique."', '".$_note."', '".$_date."')";
14    //création de la connexion
15    $servername = "localhost";
16    $username = "root";
17    $password = "";
18    $dbname = "therapeute";
19    $conn = new mysqli($servername, $username, $password, $dbname);
20    // Check connection
21    if ($conn->connect_error) {
22        die("Connection failed: " . $conn->connect_error);
23    }
24    $result = $conn->query($sql);
25    $conn->close();
26    echo "Votre critique a bien été enregistrée<br>";
27    echo "<a href='page1.php'>Retour sur la page d'accueil</a>";
28 }
```

**Code 3** : Code source de *traitement.php*

Le code PHP de *traitement.php*, présenté au Code 3, n'est pas visible. Il est lié à la page 3 à travers la méthode `.post()` qui permet d'envoyer des données, dans notre cas celles du formulaire.

Une fois les informations reçues, elles sont insérées dans une requête SQL de type *INSERT TO* qui permet d'écrire dans la base de données.

Le code est fini par un *ECHO* qui permet d'écrire sous forme HTML un message à l'utilisateur afin de lui confirmer que la critique a bien été enregistrée et propose un retour à la page d'accueil à travers un référence à la page 1.

```

1      <?php
2          $servername = "localhost";
3          $username = "root";
4          $password = "";
5          $dbname = "therapeute";
6
7          // Create connection
8          $conn = new mysqli($servername, $username, $password,
9          $dbname);
10         // Check connection
11         if ($conn->connect_error) {
12             die("Connection failed: " . $conn->connect_error);
13         }
14
15         $sql = "SELECT p1.nom as nom_therapeute, p1.prenom as
16         prenom_therapeute, p2.nom as nom_patient, p2.prenom as
17         prenom_patient ,critique.critique, critique.note,
18         critique.dateID
19         FROM critique
20         INNER JOIN therapeute ON critique.therapeuteId =
21         therapeute.therapeuteId
22         INNER JOIN patient ON critique.patientID =
23         patient.patientID
24         INNER JOIN personne p1 ON p1.personneID =
25         therapeute.therapeuteID
26         INNER JOIN personne p2 ON p2.personneID =
27         patient.patientID ";
28         $result = $conn->query($sql);
29         if ($result->num_rows > 0) {
30             // output data of each row
31             while($row = $result->fetch_assoc()) {
32                 //boucle crée grâce à fetch_assoc
33                 echo "<tr> <td>" . $row["nom_therapeute"] . " "
34                 . $row["prenom_therapeute"] . "</td>",
35                 "<td>" . $row["nom_patient"] . " " .
36                 $row["prenom_patient"] . "</td>",
37                 "<td>" . $row["critique"] . "</td>",
38                 "<td>" . $row["note"] . "</td>",
39                 "<td>" . $row["dateID"] . "</td>";
40
41             }
42         } else {
43             echo "0 results";
44         }
45
46         $conn->close();
47     ?>

```

**Code 4** : Code source de *page4.php*

La page 4 est similaire à la *page1.php*. Le code PHP du Code 4 permet l'affichage des données enregistrées par les utilisateurs à travers la *page3.php*.

---

# 5

## Conclusion

A la fin de mon travail, mon site est muni de différentes possibilités fournis par le PHP et le SQL tels que l'option de voir les différentes thérapeutes référencées sur la plateforme, la possibilité d'avoir un retour de la part des patients afin d'améliorer la qualité des soins prodigués et la possibilité de voir les critiques émises permet d'augmenter la transparence des thérapeutes en médecine douce.

Cependant le site web reste tout de même sommaire pour un usage professionnel. Il n'y a pas de login ce qui permet à n'importe qui d'émettre de fausses critiques. De plus, la base de données n'est pas utilisée à 100%. Il y aurait possibilité d'utiliser les données de la table adresse afin d'utiliser une carte GoogleMap.

En résumé, ce travail permet de mieux comprendre la création d'une base de données et l'utilisation de celle-ci. Il me permet de mieux savoir ce qui est possible d'être réalisable par une équipe d'informaticien et me permettra, je l'espère une meilleure synergie avec ceux-ci.

# A

## Fichier .zip du code source

Le fichier .zip contient 9 fichiers

- Therapeute.sql : La base de données à ouvrir à l'aide de phpMYAdmin
- Page1.php : la page d'accueil
- Page2.php : Liste des thérapeutes
- Page3.php : Formulaire critique
- Page4.php : Liste des critiques
- Traitement.php : page de traitement des données envoyées
- Le fichier CSS fourni par l'exercice\_5a\_sol du cours d'informatique de gestion du Prof. Dr. Jacques Pasquier – Rocha
- Le fichier image qui contient l'image medecinedouce.jpg
- Rapport.pdf : le rapport au format PDF

# Bibliographie

**[Meier, 2006]**

A.Meier, *Introduction pratique aux bases de données relationnelles*, Springer, 2006

**[Pasquier, Albreshne, 2014]**

J.Pasquier, A.Albreshne, *Informatique de gestion I*, Université de Fribourg, 2014.

**[w3Schools.com, 2017]**

PHP Select Data From MySQL

[https://www.w3schools.com/php/php\\_mysql\\_select.asp](https://www.w3schools.com/php/php_mysql_select.asp), Dernière visite : 29.10.2017

**[MySQL Workbench, 2017]**

Download MySQLWorkbench, <https://dev.mysql.com/downloads/workbench>,

Dernière visite : 29.10.2017

**[MAMP, 2017]**

Download MAMP:My Apache – MySQL – PHP, <https://www.mamp.info/en/>,

Dernière visite :29.10.2017

**[form.guide, 2017]**

Using the POST methode in a PHP form

<http://form.guide/php-form/php-form-post.html> , Dernière visite : 29.10.2017

**[coordonnees-gps, 2017]**

Coordonnées GPS et Google Map

<https://www.coordonnees-gps.fr/>, Dernière visite : 29.10.2017

**[Google, 2017]**

Utiliser MySQL et PHP avec Google Maps

<https://developers.google.com/maps/documentation/javascript/mysql-to-maps?hl=fr> ,

Dernière visite : 29.10.2017

**[stackoverflow, 2017]**

<https://stackoverflow.com/> ,Dernière visite : 29.10.2017

**[Wikipedia a, 2017]**

Clé primaire

[https://fr.wikipedia.org/wiki/Cl%C3%A9\\_primaire](https://fr.wikipedia.org/wiki/Cl%C3%A9_primaire) , Dernière visite 26.10.2017

**[Wikipedia b, 2017]**

Clé étrangère

[https://fr.wikipedia.org/wiki/Cl%C3%A9\\_%C3%A9trang%C3%A8re](https://fr.wikipedia.org/wiki/Cl%C3%A9_%C3%A9trang%C3%A8re), dernière visite :  
26.10.2017