

## Développement web

# Projet PHP

Année Académique 2019-2020

Marchand Valentin

# 1. Table des matières

1.	Table des matières .....	2
2.	Introduction .....	3
3.	Présentation générale du projet .....	3
4.	Base de données .....	4
4.1.	Schéma.....	4
4.2.	Description.....	4
5.	Explication des scripts .....	5
5.1.	Index.php .....	5
5.2.	Model.....	5
5.2.1.	Connect_data.php et db_config.php .....	5
5.2.2.	Verif_data.php.....	5
5.2.3.	Insert_data.php .....	6
5.2.4.	Delete_data.php.....	6
5.2.5.	Colect_data.php .....	6
5.3.	View .....	7
5.3.1.	Template.php .....	7
5.3.2.	Header.php.....	8
5.3.3.	Navbar.php .....	8
5.3.4.	Content.php .....	8
5.3.5.	Add_form.php .....	8
5.3.6.	Collection.php .....	9
5.3.7.	Form_inscription.php .....	9
5.3.8.	Form_modal.php .....	9
5.3.9.	Footer.html.....	9
5.4.	Controller.....	9
5.4.1.	Verif_inscription.php.....	9
5.4.2.	Verif_login.php .....	10
5.4.3.	Verif_add.php.....	10
5.4.4.	LogOut.php.....	10
5.4.5.	Add_to_col.php.....	11
5.4.6.	Del_to_col.php .....	11
6.	Problèmes rencontrés.....	12
7.	Améliorations et conclusions.....	12

## 2. Introduction

Dans le cadre du cour de programmation web nous avons dû créer un site lié à une base de données avec des fonctionnalités codées en PHP. Pour ma part, aimant beaucoup le montage pc, mon choix s'est porté sur la création d'un site permettant de choisir les différent composant à mettre dans son ordinateur.

## 3. Présentation générale du projet

Ce site contient différent composant de tous type visible par tout le monde mais sélectionnable uniquement par les personnes inscrites afin de donner une raison à l'utilisateur de s'inscrire.

Tous les membre ont accès à une collection où tous les composant choisi viennent se mettre afin que l'utilisateur puisse voir sa sélection et est aussi la possibilité de supprimer les composant qu'il ne veut pas garder.

La puissance consommée par les composants est également indiquée afin que l'utilisateur puisse voir si l'alimentation choisi est assez puissante pour tout alimenter.

L'administrateur, quant à lui, a accès à une page permettant d'ajouter un composant au site avec son image, son nom, sa marque, sa description, sa puissance et sa référence.

### Compte Admin :

Mail : [marchandvalentin.7@outlook.fr](mailto:marchandvalentin.7@outlook.fr)

MDP : ValentinMarchand@22

Mail : [antoine.malaise@heh.be](mailto:antoine.malaise@heh.be)

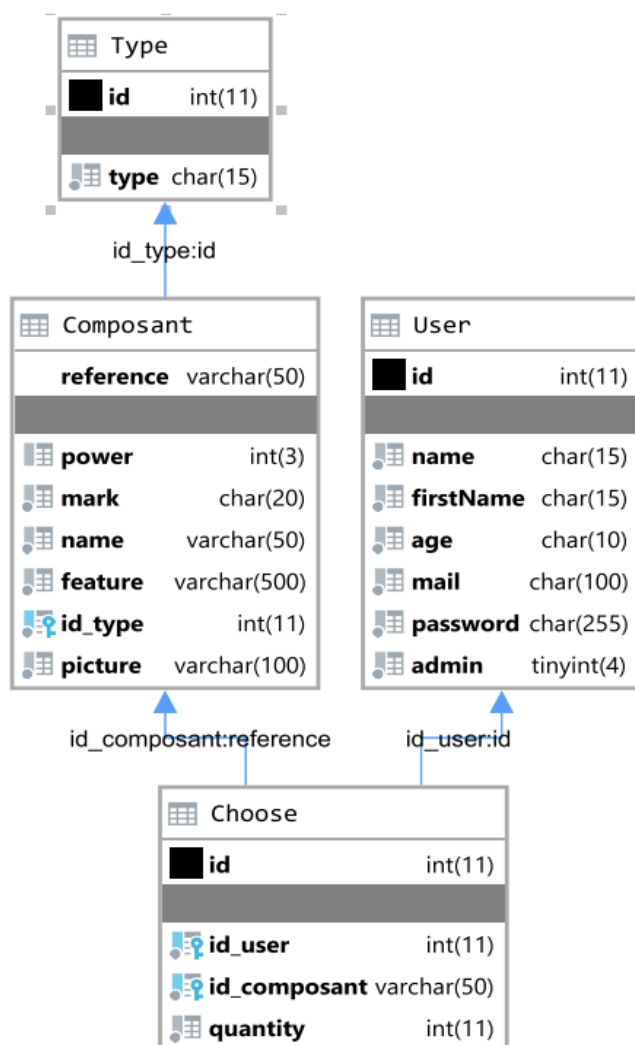
MDP : AntoineMalaise@1

Mail : [fabrice.scopel@heh.be](mailto:fabrice.scopel@heh.be)

MDP : FabriceScopel@1

## 4. Base de données

### 4.1. Schéma



### 4.2. Description

Il y a 4 entités, Type, Composant, User et Choose ;

- Type : permet d'identifier les différents types de composant en associant un id avec le type.
- Composant : Liste tous les composants, la référence est la clé primaire, ils font référence à un Type.
- User : Liste des utilisateurs étant inscrit au site.
- Choose : Liste des composants choisi par les utilisateurs, associe id de l'utilisateur avec la référence du composant, les clés étrangères sont donc id\_composant et id\_user.

## 5. Explication des scripts

Les scripts sont séparés en 3 catégories :

- Model : lien avec la base de données
- View : affichage des éléments à l'utilisateur
- Controller : teste des entrées utilisateur

### 5.1. *Index.php*

Dans aucune des catégories mentionnées, sert de point d'entrée du site en redirigeant vers la page principale template.php.

```
8 header('Location: view/template.php?value=home');
```

### 5.2. *Model*

#### 5.2.1. *Connect\_data.php et db\_config.php*

db\_config.php : Option de connexion, par exemple le nom de la base de données et le mot de passe.

```
3 $DB_DSN = 'mysql:host=' . $_ENV['HOST'] . ';dbname=' . $_ENV['HOSTNAME'];
4 $DB_USER = 'root';
5 $DB_PASS = $_ENV['PASSWORD'];
```

connect\_data.php : Gère la connexion à la base de données et affiche un message d'erreur si celle-ci échoue.

```
5 try
6 {
7     $PDO = new PDO($DB_DSN,$DB_USER,$DB_PASS,$options);
8     return $PDO;
9 }
10 catch (PDOException $e)
11 {
12     echo "Erreur : " . $e->getMessage();
13     return null;
14 }
```

#### 5.2.2. *Verif\_data.php*

Possède 4 fonctions permettant de vérifier si des éléments existe.

- verifMailUser : vérifie si le mail n'est pas déjà inscrit.

```
4 function verifMailUser($mail) : bool
```

- verifPasswordUser : vérifie si le mot de passe est correct par rapport au mail donné.

```
23      function verifPasswordUser($password,$mail,&$result) : bool
```

- verifReferenceComposant : vérifie dans la table Composant si le composant entré n'existe pas déjà.

```
47      function verifReferenceComposant($reference) : bool
```

- verifReferenceChoose : vérifie dans la table Choose si l'utilisateur n'a pas déjà sélectionné cette élément.

```
66      function verifReferenceChoose($reference,$user,&$quantity = 0) : bool
```

### 5.2.3. *Insert\_data.php*

Possède 4 fonctions permettant d'insérer des données.

- insertUser : Ajoute un nouvel utilisateur.

```
3      function insertUser($user)
```

- insertComposant : Ajoute un nouveaux composant.

```
24     function insertComposant(array $composant)
```

- insertChoose : crée une nouvelle liaison entre un composant et un utilisateur.

```
42     function insertChooce($reference,$user)
```

- changeQuantity : Change l'attribue quantity d'un élément dans le tableau Choose.

```
53     function changeQuantity($reference,$user,$quantity)
```

### 5.2.4. *Delete\_data.php*

Possède 2 fonctions permettant de supprimer des données.

- Del\_choose : supprime un élément du tableau Choose.

```
4      function del_choose($reference,$user)
```

- Del\_all\_choose : supprime tous les éléments lier à un utilisateur du tableau Choose.

```
15     function del_all_choose($user)
```

### 5.2.5. *Colect\_data.php*

Possède 2 fonctions permettant de récupérer des données.

- get\_composants : Récupère tous les composants d'un type.

```
4      function get_composants($index) : array
```

- get\_choose : Récupère tous les composant choisi par un utilisateur.

```
16     function get_choose($user) : array
```

## 5.3. View

### 5.3.1. *Template.php*

Script principale chargeant les différentes parties de la page, en-tête, navigation, contenu et pied de page.

```
2 | include_once 'header.php';
3 | include_once 'form_modal.php';
4 | include_once 'navbar.php';
5 | include_once 'content.php';

32 | if ($_GET['value'] == 'mother' || $_GET['value'] == 'home')
33 | {
34 |     mother();
35 | }
36 | else if ($_SESSION['admin'] && $_GET['value'] == 'add')
37 | {
38 |     include_once 'add_form.php';
39 | }
40 | else if ($_GET['value'] == 'col')
41 | {
42 |     include_once 'collection.php';
43 | }
44 | else if ($ifExists)
45 | {
46 |     component();
47 | }
48 | else
49 | {
50 |     echo '<h1>Erreur : La page n\'existe pas</h1>';
51 | }

55 | if($ifExists)
56 |     affiche_composant($ty);
57 | include_once 'footer.html';
```

### 5.3.2. *Header.php*

En-tête html générique des pages.

```

8  <!doctype html>
9  <html lang="fr">
10   <head>
11     <title>Tester son PC</title>
12     <link rel="stylesheet" href="...">
13     <link rel="stylesheet" href="...">
14     <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
15     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js"></script>
16     <script src="https://stackblitz.com/..."></script>
17     <meta charset="utf-8">
18     <meta name="description" content="...">
19     <meta name="keywords" content="...">
20     <meta name="viewport" content="width=device-width, initial-scale=1">
21     <meta name="author" content="...">
22     <meta name="geo.placement" content="...">
23     <meta name="geo.region" content="...">
24   </head>
25   <body>

```

### 5.3.3. *Navbar.php*

Bar de navigation générique des différentes pages

```

6  <div class="collapse navbar-collapse" id="navbarSupportedContent">
7    <ul class="navbar-nav mr-auto">
8      <li class="nav-item <?php if($_GET['value'] == 'mother' || $_GET['value'] == 'home') echo 'active';?>">
9        <a class="nav-link" href="template.php?value=mother#nav">Carte mère<span class="sr-only">(current)</span></a>
10      </li>

```

### 5.3.4. *Content.php*

Différents contenus des différentes pages avec la description des composants ainsi que l'affichage de la liste des composants d'un type.

```

68  function affiche_composant($ty)
69  {
70    $list_composant = get_composants($ty);

```

### 5.3.5. *Add\_form.php*

Formulaire dédié aux administrateurs afin de pouvoir ajouter des composants à la base de données.

```

24  <form action="../../controller/verif_add.php" method="post" enctype="multipart/form-data">

```



### 5.3.6. *Collection.php*

Page affichant les composants choisis par l'utilisateur connecté.

```
13     require_once '../model/colect_data.php';
14     $choose_list = get_choose($_SESSION['id']);
```

### 5.3.7. *Form\_inscription.php*

Formulaire d'inscription.

```
6 <form id="corp" method="post" action="../controller/verif_inscription.php">
```

### 5.3.8. *Form\_modal.php*

Formulaire de connexion affiché avec un modal.

```
11 <form method="post" action="../controller/verif_logIn.php?value=<?=$_GET["value"];?>">
```

### 5.3.9. *Footer.html*

Pied de page générique des pages

## 5.4. *Controller*

### 5.4.1. *Verif\_inscription.php*

Vérifie les différents éléments entrés dans le formulaire d'inscription et si tout est correct appelle la fonction insertUser afin d'inscrire l'utilisateur dans la base de données.

```
2     require_once '../model/insert_data.php';
3     require_once '../model/verif_data.php';

91     if($is_correct)
92     {
93         insertUser($val);
94         verifPasswordUser($password,$mail,$result);
95         extract($result);
96         session_start();
97         $_SESSION["name"] = $name;
98         $_SESSION["firstName"] = $firstName;
99         $_SESSION["id"] = $id;
100        $_SESSION["admin"] = 0;
101        header("Location: ../view/template.php?value=home");
102    }
```

### 5.4.2. Verif\_login.php

Vérifie si le mail entré existe et que le mot de passe correspond en appelant les fonctions verifMailUser et verifPasswordUser afin de connecter l'utilisateur au site.

```

2      require_once '../model/verif_data.php';

13     if(verifPasswordUser($password,$mail,$result))
14     {
15         if(!session_start())
16         {
17             echo "la session na pas pu démarrer";
18         }
19         //echo "password check";
20         $_SESSION["name"] = $result["name"];
21         $_SESSION["firstName"] = $result["firstName"];
22         $_SESSION["id"] = $result["id"];
23         $_SESSION['admin'] = $result["admin"];
24         header("Location: ../view/template.php?value=$value");
25     }

```

### 5.4.3. Verif\_add.php

Vérifie les différents éléments entrés dans le formulaire d'ajout des composant et si tout est correct appel la fonction insertComposant afin d'ajouter le composant à la base de données.

```

2      require_once '../model/insert_data.php';
3      require_once '../model/verif_data.php';

156     if($is_correct)
157     {
158         insertComposant($value);
159
160         header("Location: ../view/template.php?value=add&c=1");
161     }

```

### 5.4.4. Logout.php

Permet de se déconnecter du site.

```

2      session_start();
3      session_unset();
4      session_destroy();
5      extract($_GET);
6      header ("Location: ../view/template.php?value=$value");

```

#### 5.4.5. *Add\_to\_col.php*

Vérifie si l'utilisateur est connecté et si la référence du composant que l'utilisateur veut ajouter existe avant d'appeler la fonction insertChoose pour l'ajouter à la base de données.

```
2      require_once '../model/verif_data.php';
3      require_once '../model/insert_data.php';

11     if(verifReferenceChoose($_GET['ref'],$_SESSION['id'],$quantity))
12     {
13         changeQuantity($_GET['ref'],$_SESSION['id'],$quantity);
14     }
15     else
16     {
17         insertChoose($_GET['ref'],$_SESSION['id']);
18     }
19 }
20 header("Location: ../view/template.php?value=col");
```

#### 5.4.6. *Del\_to\_col.php*

Vérifie si l'utilisateur est connecté et si la référence du composant que l'utilisateur veut supprimer existe avant d'appeler la fonction del\_choose pour le supprimer de la base de données ou del\_all\_choose pour supprimer tous les composants de l'utilisateur.

```
8      if($_GET['mode'] == 0)
9      {
10         if(verifReferenceChoose($_GET['ref'],$_SESSION['id']))
11         {
12             del_choose($_GET['ref'],$_SESSION['id']);
13         }
14     }
15     else if($_GET['mode'] == 1)
16     {
17         del_all_choose($_SESSION['id']);
18     }
19 }
20
21 header("Location: ../view/template.php?value=col");
```

## 6. Problèmes rencontrés

- Problème de connexion à la base de données.
- Problème avec la connexion de l'utilisateur, variable session.
- Problème de retour à une page correct après inscription.

## 7. Améliorations et conclusions

Pour améliorer l'état actuel du site j'aurais pu :

- Ajouter la possibilité de supprimer son compte ou de le modifier.
- Ajouter la possibilité de supprimer un composant.
- Améliorer la sécurité.
- Laisser la possibilité à un administrateur de supprimer un compte utilisateur.
- Clarté dans le code.

Actuellement le site est capable de :

- Permettre aux utilisateurs de s'inscrire avec sécurité.
- Permettre aux utilisateurs de se connecter et déconnecter.
- Ajouter et afficher les composants.
- Ajouter et supprimer des composants à notre collection en tant qu'utilisateur.
- Cliquer sur une image pour la zoomé.