

Rapport de Projet en Php : Site Web e-commerce

1. Introduction :

Pour ce projet de création d'une application Web en PHP j'ai décidé de partir sur site de vente en ligne, c'est un projet que j'avais depuis un certain moment en tête de faire, que ce soit pour moi et pour montrer mes compétences.

Vendant déjà des produits fait en impression 3D principalement sur des sites de ventes tel que Etsy et Ebay, j'ai depuis un moment pensé à vouloir faire mon propre site pour pouvoir mettre en vente mes produits.

2. Description du projet et fonctionnalité attendues :

2.1 Catalogue de produit :

Le but est de proposer plusieurs produits, chaque produits ayant leur propres pages avec une description et plusieurs photos du produit, avec des options de couleur, de design, ou selon une compatibilité du produit.

2.2 Inscription et Connexion :

L'utilisateur pourra explorer le site sans se connecter, mais si il le veut, il pourra se créer un compte si il n'en n'a pas déjà un, puis se connecter pour pouvoir passer commande.

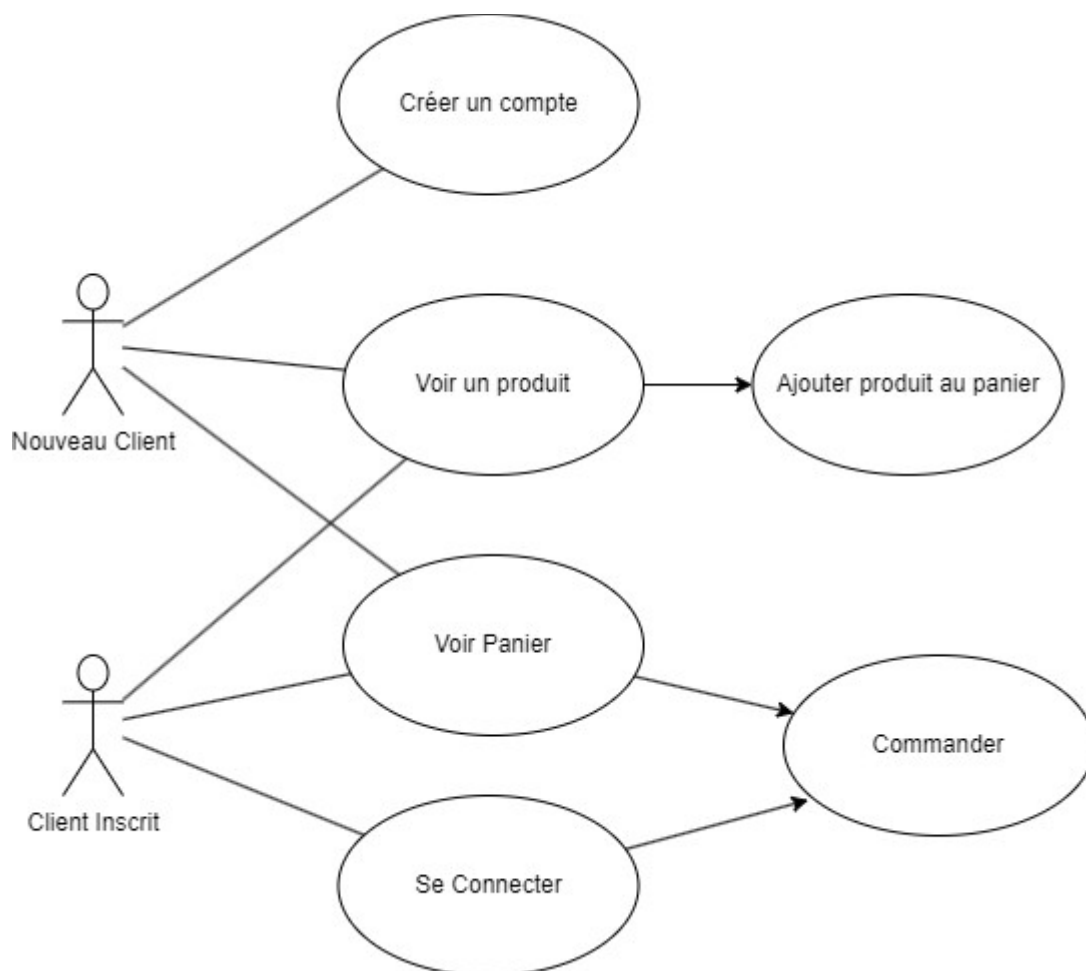
2.3 Ajout au panier :

Une fonctionnalité attendue serait de pouvoir ajouter un produit au panier selon une disponibilité (en vérifiant en communiquant avec la base de donnée), puis de voir les produits qu'il a ajouté au panier.

2.4 Compte utilisateur :

Une autre fonctionnalité serait de pouvoir se diriger vers une page utilisateur si l'utilisateur est déjà connecté pour pouvoir ainsi modifier ses informations personnelles, que ce soit adresse ou mot de passe.

Diagramme de cas d'utilisation pour un site de vente en ligne :



3. Architecture et structure du projet :

3.1 Décomposition des pages

J'ai voulu partir rapidement sur une décompositions des pages, c'est à dire de profiter de l'utilisation d'include pour séparer chaque pages en blocs, et ainsi gagner de la lisibilité et de la flexibilité lors de modifications

Le code qui permet cela est le suivant :

```
41 </div>
42 <!-- Web pages sorting navigation -->
43 <?php
44
45 $allowed = array('slider', 'fn_paddles', 'tm_shifter', 'lg_shifter', 'register', 'login', 'cart');
46 $page = ( isset($_GET['page']) ) ? $_GET['page'] : 'slider';
47 if ( in_array($page, $allowed) ){
48     include("$page.php");
49 }
50 else {
51     include("404.php");
52 }
53 ?>
```

Essentiellement on va stocker tous les noms des pages que l'on a dans notre projet dans l'array \$allowed, et lors d'un appel de ce type :

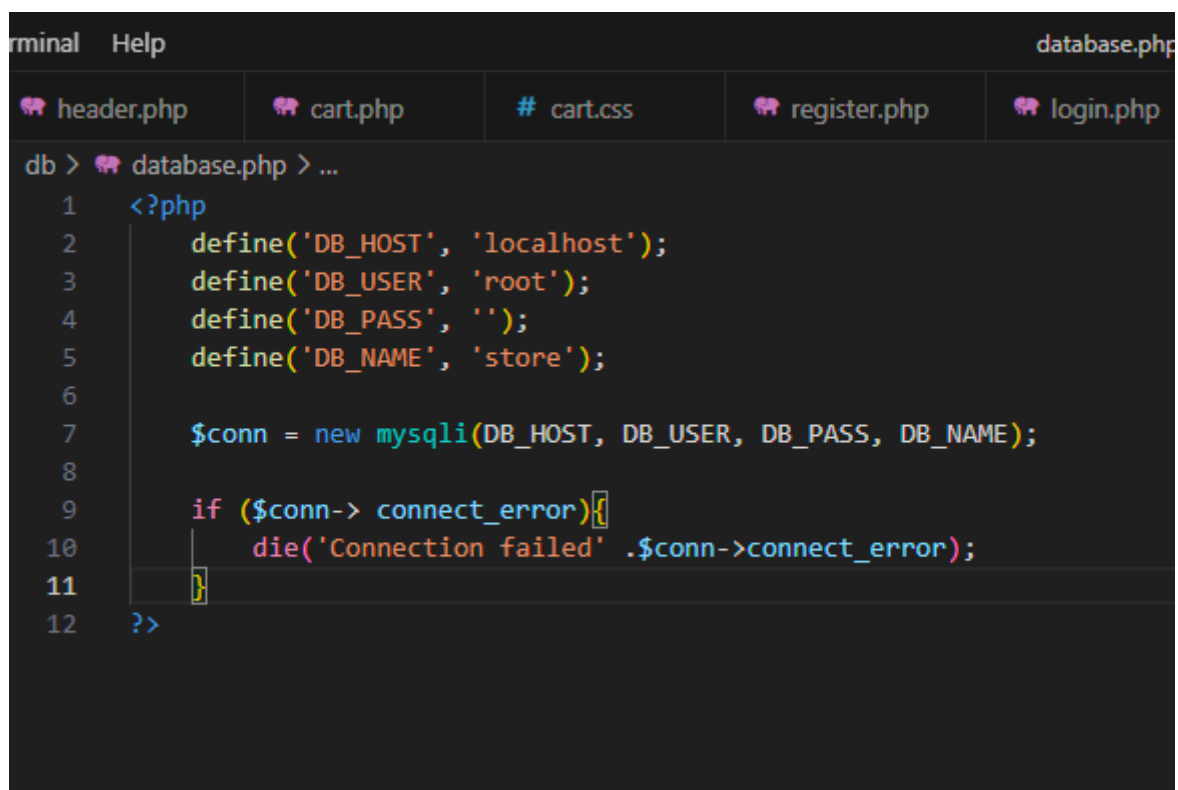
```
3 <!-- Navigation bar -->
4 <div class="navbar">
5     <a href="header.php?page=slider">Home</a>
6     <a href="header.php?page=fn_paddles">Fanatec</a>
7     <a href="header.php?page=lg_shifter">Logitech</a>
8     <div class="tm_dropdown">
9         <button class="tm_button_dropdown">Thrustmaster</button>
10        <div class="tm_dropdown_content">
11            <a href="header.php?page=tm_shifter">Thrustmaster shifter</a>
12            <a href="#">Thrustmaster Quick Release</a>
```

on vérifie que le nom de cette page est bien présente dans notre array, le cas échéant on pourra la charger.

Le header.php est donc la base du site, et l'on vient changer le contenu intérieur selon les pages sur lesquelles on navigue.

3.2 Connexion base de donnée

Le code de connexion à la base de donnée se fait sur un fichier à part pour plus de lisibilité et l'on pourra l'inclure dans l'header.php, pour n'avoir ainsi à l'inclure qu'ici, car toutes les autres pages sont un bloc de l'header.php

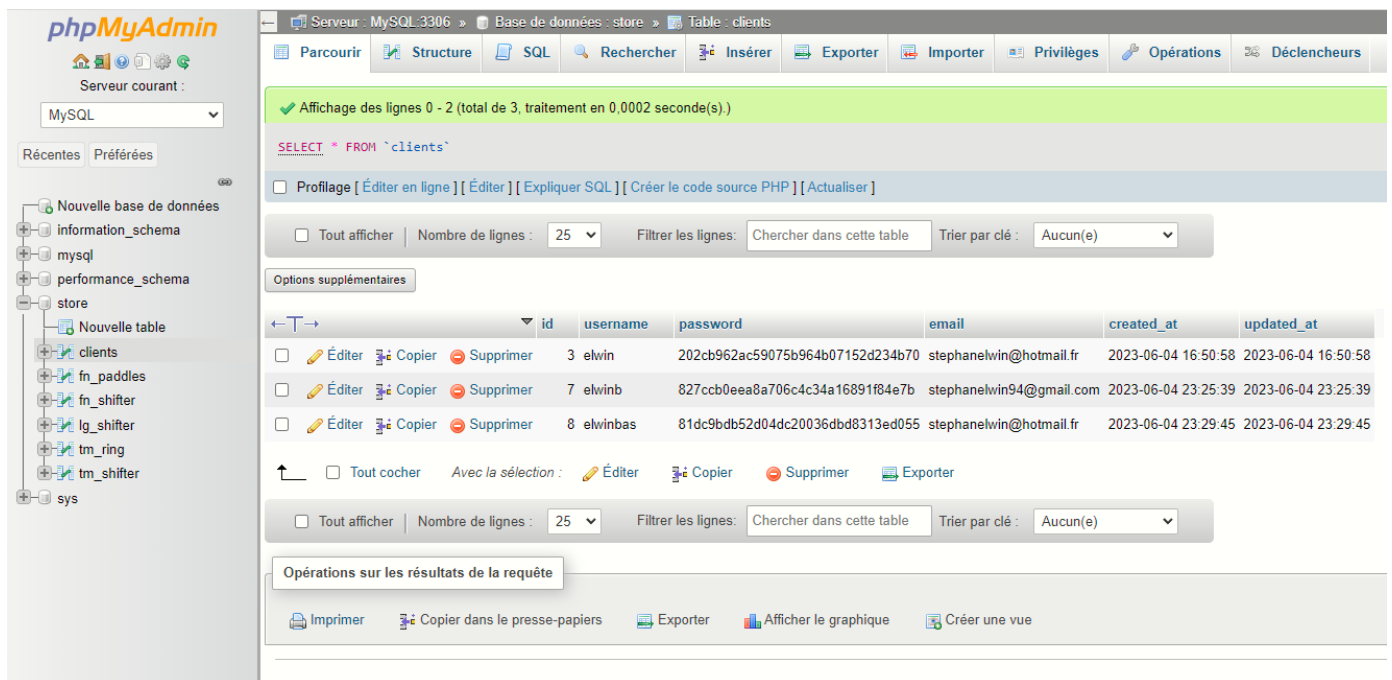


```
terminal  Help                                     database.php
header.php  cart.php  # cart.css  register.php  login.php

db > database.php > ...
1  <?php
2      define('DB_HOST', 'localhost');
3      define('DB_USER', 'root');
4      define('DB_PASS', '');
5      define('DB_NAME', 'store');
6
7      $conn = new mysqli(DB_HOST, DB_USER, DB_PASS, DB_NAME);
8
9      if ($conn->connect_error){
10         die('Connection failed' . $conn->connect_error);
11     }
12  ?>
```

3.3 Création de compte, et connexion


Tout d'abord, la structure de ma base de donnée et la table pour les clients est la suivante :



The screenshot shows the phpMyAdmin interface for a MySQL server. The left sidebar displays the database structure, including the 'store' database and the 'clients' table. The main panel shows the 'clients' table structure and data. The table has columns: id, username, password, email, created_at, and updated_at. The data is displayed in a table with 3 rows.

	id	username	password	email	created_at	updated_at
<input type="checkbox"/>	3	elwin	202cb962ac59075b964b07152d234b70	stephanelwin@hotmail.fr	2023-06-04 16:50:58	2023-06-04 16:50:58
<input type="checkbox"/>	7	elwinb	827ccb0eea8a706c4c34a16891f84e7b	stephanelwin94@gmail.com	2023-06-04 23:25:39	2023-06-04 23:25:39
<input type="checkbox"/>	8	elwinbas	81dc9bdb52d04dc20036dbd8313ed055	stephanelwin@hotmail.fr	2023-06-04 23:29:45	2023-06-04 23:29:45

La création d'un compte se fait via cette page

[Home](#) [Fanatec](#) [Logitech](#) [Thrustmaster](#) [Welcome, elwinbas](#) 

REGISTER

Register Now

Already have an account? [Log in here](#)

© 2023 Sim Racing Homepage. All rights reserved.

Avec le code mis en place, si le mot de passe est différent il y aura une erreur, et si le nom d'utilisateur est déjà pris il y aura aussi un message d'erreur.

Cette vérification est réalisé en comparant sur notre base de donnée un nom d'utilisateur similaire, si il existe on l'inclue dans une array, ce qui nous permet de savoir si ce nom d'utilisateur existe déjà. C'est d'ailleurs avec cette logique que l'on va se connecter.

Pour se connecter, on va entrer notre nom d'utilisateur et mot de passe. La logique pour vérifier l'utilisateur est la suivante :

```
1
2 <?php
3
4 if (isset($_POST['Submit'])) {
5     $Username = filter_input(INPUT_POST, 'Username', FILTER_SANITIZE_SPECIAL_CHARS);
6     $Password = md5($_POST['Password']);
7
8     $select = "SELECT * FROM clients WHERE username = ? AND password = ?";
9     $stmt = mysqli_prepare($conn, $select);
10    mysqli_stmt_bind_param($stmt, 'ss', $Username, $Password);
11    mysqli_stmt_execute($stmt);
12    $result = mysqli_stmt_get_result($stmt);
13
14    if (mysqli_num_rows($result) > 0) {
15
16        $_SESSION['Username'] = $Username;
17        $_SESSION['isLoggedIn'] = true;
18
19        header('Location: header.php?page=slider');
20        exit();
21    } else {
22        $error[] = 'Incorrect password or username';
23    }
24 }
25
```

Comme pour la création de compte, mais cette fois on compare un nom d'utilisateur avec un mot de passe pour vérifier l'existence de celui-ci. Si c'est trouvé, on aura une session d'utilisateur avec le nom de client.

3.4 Session utilisateur

Cette fonctionnalité n'est malheureusement pas abouti, mais le but étant d'utiliser `session_start()` pour sauver des informations, tel que le nom de l'utilisateur, pour par exemple l'afficher une fois qu'il est connecté. Mon but était aussi de pouvoir sauver les articles que l'utilisateur mets dans son panier pour qu'il puisse ainsi les retrouver.

Bien sûr j'aurais voulu aussi implémenter la possibilité de se déconnecter.

Conclusion :

Ce projet m'a permis de faire un première essai sur quelque chose que j'ai voulu faire depuis un très long moment. Ce n'est pas en Php que j'aurais voulu le faire, mais après m'y être mit je vois définitivement des points intéressants à l'utiliser. J'aurais aimé développer bien d'avantages ce site web, mais c'est déjà une bonne base car c'est comme cela que j'envisagerais de le concevoir. C'est une base solide pour des améliorations futures, même si je devais changer de framework pour le continuer.

Ce qui à prend une très grosse partie de mon temps n'était pas tellement la logique, mais surtout la partie design UX, surtout lorsqu'on veut mêler à ça un changement dynamique en utilisant des données de notre base de donnée.