D&CODE

Nom : BENRABAH

Prénom : Samy

Introduction

Compétences couvertes par le projet

Le projet que je vais présenter est un site e-commerce réalisé en binôme, je vais vous présenter les compétences du référentiel que j'ai couvert avec ma partie du développement.

- *Activité type 1 :*

"Développer la partie front-end d'une application web ou web-mobile en intégrant les recommandations de sécurité. "

- Maquetter une application
- Réaliser une interface web statique et adaptable
- Développer une interface utilisateur web dynamique

Les compétences liées à l'activité de type 1 sont abordées au travers de :

- La réalisation d'une maquette et d'un prototype
- L'intégration de la maquette dans le projet

- *Activité type 1 :*

"Développer la partie back-end d'une application web ou web-mobile en intégrant les recommandations de sécurité. "

- Créer une base de données
- Développer les composants d'accès aux données
- Développer le back-end d'une application web / web mobile

Les compétences liées à l'activité de type 2 sont abordées au travers de :

- La conception et la modélisation de base de données utilisant la méthode MERISE.

- L’utilisation de la POO permettant l’interaction avec les données présentes en BDD pour l’utilisateur

Résumé du dossier de projet

Ce dossier présente une partie du travail que j’ai effectué, à savoir la création du site web D&Code, une boutique en ligne proposant des produits d’intérieur.

Le site web D&Code a pour objectif de permettre à ses utilisateurs d'avoir accès aux produits et les différentes catégories de produits, commander différents produits,elle permet aussi aux utilisateurs de s’inscrire, se connecter.
Le site pourra être utilisé par les utilisateurs inscrits ou non mais uniquement les utilisateurs inscrits pourront commander.
Il y a également une partie administrateur qui sert au management du site, de ses produits, des offres misent en avant et un suivi d’activité et des commandes ainsi que des promotions.

Ce site web comptant comme l’un des deux plus grands projets de l’année pour le passage du titre professionnel de développeur web et web mobile, a été réalisé en binôme avec Morad Labrid au cours de ma formation de développeur web à l’école La Plateforme à Marseille.

Cahier des charges

Présentation de l’entreprise

C’est une entreprise fictive que j’ai dû créer pour la réalisation de ce projet de site e-commerce. J’ai décidé de choisir comme domaine d’activité la vente de produits de design d’intérieurs.

Les besoins clients

L’objectif

L'objectif est de mettre en place une marketplace permettant à des particuliers d’acheter des produits de décoration ou de meubler leur intérieur.
Prévue pour être utilisée uniquement sur le web, un travail en amont sera réalisé pour que le site soit compatible sur tous les types d’appareils que ce soit mobile ou ordinateur.

Les cibles

Sur le site il y aura principalement trois types de cibles :

Les visiteurs : ce qui comprend les utilisateurs non inscrits sur le site, qui pourront consulter les produits et potentiellement devenir des nouveaux inscrits, voir qui nous sommes, prendre contact avec le fournisseur.

Les inscrits : ce qui comprend les utilisateurs qui pourront consulter les produits, voir qui nous sommes, avoir accès au panier, prendre commande auprès du site et avoir un historique des commandes passées.

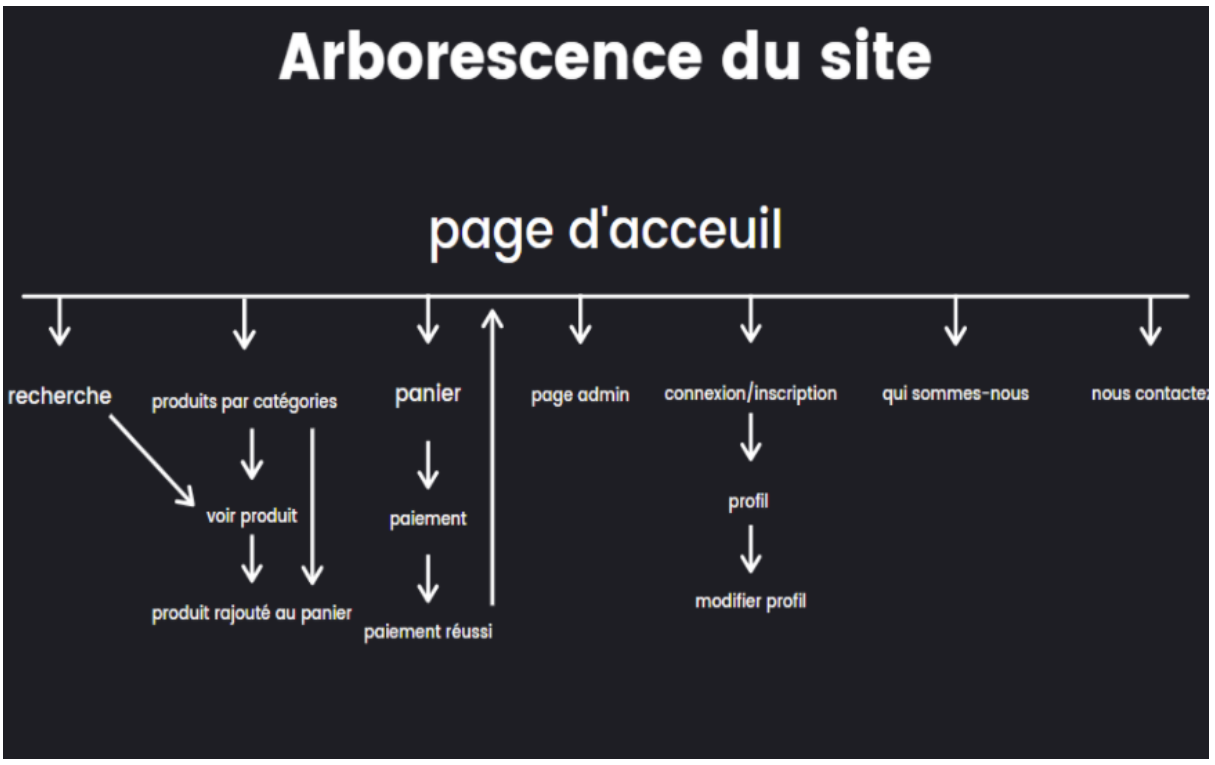
les administrateurs : ce qui comprend les utilisateurs avec des droit d’accès spéciaux au site pour le manager, c'est-à-dire avoir un suivi des commandes réalisées, changer les produits mis en avant sur le site, ajouter, supprimer, modifier un produit, changer les droits d’accès d’un utilisateur.

Le périmètre du projet

Le site sera disponible uniquement en français.
Par ailleurs, il devra être adapté à tout les supports (ordinateurs, tablettes, mobiles) pour permettre à tout les utilisateurs d'avoir une expérience de navigation optimale.

L'arborescence

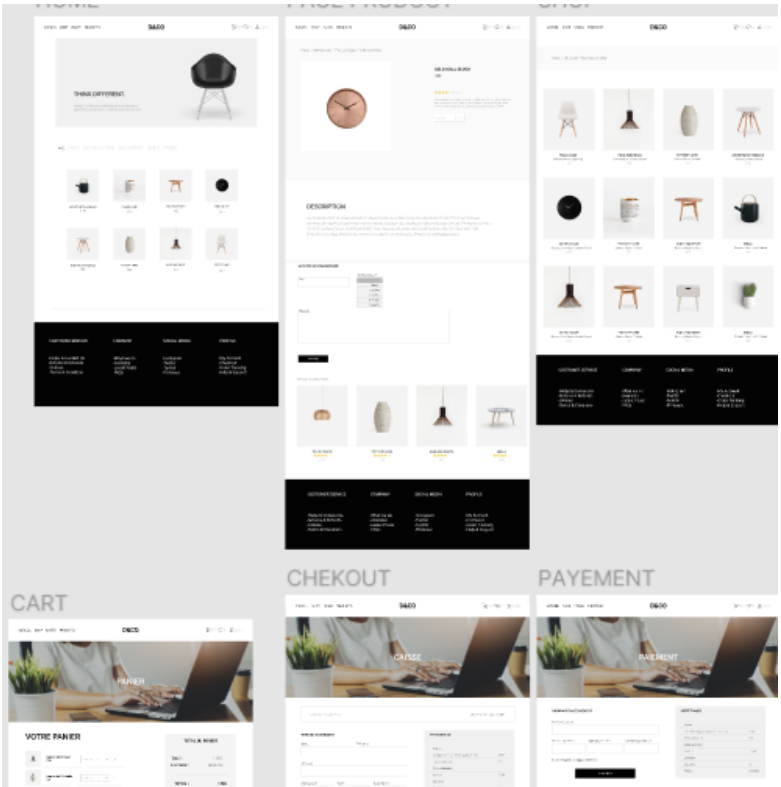
j'ai défini le parcours utilisateur, c'est-à-dire identifier comment se déroule le parcours et toutes les possibilités d'actions que l'utilisateur aura sur le site, je l'ai modélisé sous forme d'arborescence.



La charte graphique

Pour ce qu'il s'agit de la charte graphique, je défini entre autres les couleurs, les logos et les polices utilisées pour la typographie, pour que le site ai une identité visuelle, qu'il soit attrayant visuellement pour l'utilisateur. Puisque, l'utilisateur ne voit et ne se fie qu'à l'apparence du site et ne peut pas voir le Back-office.

Voici la maquette du site E-commerce partie client:



Spécificités techniques : Backend

Environnement de développement et organisation

Technologies :

Pour mener à bien le développement de ce projet, j'ai opté pour un environnement de développement sous la stack technique XAMPSERVER. WAMP est un acronyme faisant référence à:

- **X**:Linux,Unix,**Windows**,Mac le système d'exploitation utilisé
- **A**pache, logiciel permettant de créer un serveur HTTP
- **M**ySQL, le SGBD
- **P**HP le langage utilisé pour gérer les requêtes vers le serveur HTTP et générer des pages web dynamiques.

Workflow :

Pour la gestion de projet et le travail d'équipe avec mon binôme, nous avons utilisé divers outils permettant de se répartir les tâches de développement et d'avoir toutes les versions de notre site a chaque modification apportées.

Le travail sur GitHub se fait autour de deux branches principales, la branche "Master" et la branche "Dev".
d'autres branches sont créées pour chaque nouvelle fonctionnalité qu'il faut développer. Une fois le développement de ces autres branches terminées, il faut les mergées sur la branche de référence Dev pour que mon binôme puisse y avoir accès et vérifier si le code ne comporte pas de bug.

Modélisation de la base de données

Modèle Logique de données (MLD)

Le modèle logique de données (MLD), est une étape intermédiaire entre le modèle conceptuel de données et le modèle physique de données.

Le passage d'un MCD en MLD s'effectue selon quelques règles de conversion précise :

Une entité du MCD devient une relation, c'est-à-dire une table. Dans un SGBD de type relationnel, une table est une structure tabulaire dont chaque ligne correspond aux données d'un objet enregistré et où chaque colonne correspond à une propriété de cet objet.
Ces colonnes font notamment référence aux caractéristiques définies
dans le dictionnaire de données du MCD.

Les identifiants respectifs de chaque entité deviennent des clefs primaires et toutes les autres propriétés définies dans le MCD deviennent des attributs. Les clefs primaires permettent d'identifier de façon unique chaque enregistrement dans une table et ne peuvent pas avoir de valeur nulle.

Les cardinalités de type "0:n" / "1:n" sont représentées à travers le référencement de la clef primaire de la table qui la possède, en clef étrangère au sein de la table auquelle elle est liée. Si deux tables liées possèdent une cardinalité de type "0:n / 1:n", la relation sera traduite par la création d'une table de jonction, dont la clef primaire de chaque table deviendra une clef étrangère au sein de la table de jonction.

Le MLD de D&Code

Ici, nous voyons bien que les clés primaires de chaque table associée deviennent des clé étrangère dans chacune des tables qui reçoivent les données.

Extrait de code

Toutes les requêtes en base de données sont gérées par la classe **Database**.
Cette classe permet de se connecter à la base de données.

```php
p

s Database

private $_localhost = "localhost";
private $_dbname = "boutique";
private $_username = "root";
private $_password ="";
protected $pdo;

public function __construct(){

    try {
        $this->pdo = new PDO('mysql:host='.$this->_localhost.';dbname='.$this->_dbname.'', $this->_username, $this->_password);

    } catch (PDOException $Exception) {

        throw new ErrorException($Exception->getMessage(), (int) $Exception->getCode());
    }



}

public function __destruct(){
    $this->pdo;
}
```

L'utilisation de requêtes préparées, rendues possibles grâce à l'extension PDO, présentent un double avantage par rapport à l'exécution directe de requêtes SQL. Premièrement, peu importe le nombre d'exécution des requêtes, nous n'avons besoin de les préparer qu'une seule fois pour qu'elles soient réutilisables à volonté avec des paramètres identiques ou différents.
En un second temps, elles présentent un avantage majeur en termes de sécurité notamment pour prévenir des injections SQL.
Nos requêtes étant pré-formatées, elles empêchent le passage de code malicieux en paramètre, nous n'avons donc pas besoin de protéger nos paramètres ou valeurs manuellement.

Dans un premier temps, je commence par créer ma classe et y ajouter des attributs si on le souhaite.

un attribut est une variable qui est partagée par toutes les instances de la classe, on peut y stocker par exemples les paramètres de la connexion à la base de données pour ne pas avoir à les recopier à chaque fois que l'on a besoin de faire un appel à la base de données.

Pour que je puisse faire appel à mes fonctions sur cette page, il faut que je j'inclus ma page de fonction dans la page ou je souhaite utiliser ces fonctions,faire hériter ma class **Database** et ensuite instancier la classe que je veux instancier, cela ce fait de cette manière :

```php
php

quire 'database.php';
ass User extends Database

protected $pdo;
private $username;
private $password;
private $email;


public function register($firstName, $lastName, $userName, $email, $phone, $adress, $zip, $city, $country, $password)
{
    $stmt_select = $this->pdo->prepare("SELECT id FROM users WHERE email = ? OR username=?");
    $stmt_select->bindParam(1, $email);
    $stmt_select->bindParam(2, $userName);
    $stmt_select->execute();
    $stmt_select->fetchAll(PDO::FETCH_OBJ);
    $row = $stmt_select->rowCount();
```

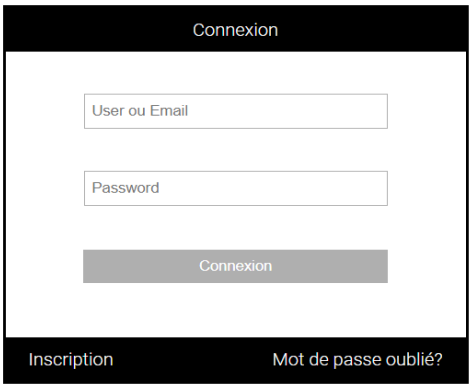
```php
<?php
$password
```

Après avoir fait cela, on peut s'attaquer à l'écriture de toutes nos fonctions. Les quatre opérations de base pour la persistance des données, en particulier le stockage d'informations en base de données est nommée le CRUD (pour *create*, *read*, *update*, *delete*).

Je vais vous présenter l'ensemble des requêtes couvertes par le CRUD :

En ce qui concerne le create, cette opération est gérée par la requête "INSERT", pour envoyer des éléments en base de données.
Dans le cadre de ma boutique en ligne, je m'en suis servi pour envoyer en base de données les clients connectés.

Je commence déjà par récupérer les données en méthode 'POST' dans mon formulaire ci dessous :



```html
<form class="connexion">
    <p>Inscription</p>
    <br>
    <input type="text"  id="first_name" placeholder="Votre nom">
    <br>
    <input type="text"  id="last_name" placeholder="Votre prénom">
    <br>
    <input type="text"  id="username" placeholder="Username">
    <br>
    <input type="email"  id="email_inscription" placeholder="Votre adresse email">
    <br>
    <input type="text"  id="phone" placeholder="06 34 56 78 90" pattern="[0-9]{2} [0-9]{2} [0-9]{2} [0-9]{2} [0-9]{2}"
    <br>
    <input type="text"  id="adress" placeholder="Votre adresse">
    <br>
    <input type="number"  id="zip" placeholder="Code postal">
    <br>
    <input type="text"  id="city" placeholder="Ville">
    <br>
    <input type="text"  id="country" placeholder="Pays">
    <br>
    <input type="password"  id="password" placeholder="Password">
    <br>
    <input type="submit" id="valider_register" value="Inscription">
    <div class="block_bottom">
        <a href="connexion.php">Connexion</a>
        <a href="connexion.php?block=Reinitialisation">Mot de passe oublié?</a>
    </div>
```

Ici le code HTML de mon formulaire :

Donc pour envoyer des données en BDD, la méthode est la suivante, je commence par rédiger ma fonction que j'ai appelé ici “connexion”,

```php
public function register($firstName, $lastName, $userName, $email, $phone, $adress, $zip, $city, $country, $password)
{
    $stmt_select = $this->pdo->prepare("SELECT id FROM users WHERE email = ? OR username=?");
    $stmt_select->bindParam(1, $email);
    $stmt_select->bindParam(2, $userName);
    $stmt_select->execute();
    $stmt_select->fetchAll(PDO::FETCH_OBJ);
    $row = $stmt_select->rowCount();
```

```php
if (isset($_POST["valider_connexion"])) {
    $username = htmlspecialchars(trim($_POST["userPHP"]));
    $email = filter_var(trim($_POST["userPHP"]), FILTER_VALIDATE_EMAIL);
    $password = htmlspecialchars(trim($_POST["passwordPHP"]));
    echo $user->connexion($username, $email, $password);
}
```

En ce qui concerne le read, cette opération est gérée par la requête SELECT, pour récupérer des éléments en base de données.

Dans le cadre de ma boutique en ligne, je m'en suis servi pour afficher les articles présents en base de données.

Je commence par rédiger la fonction qui est destinée à récupérer ces données et les afficher.

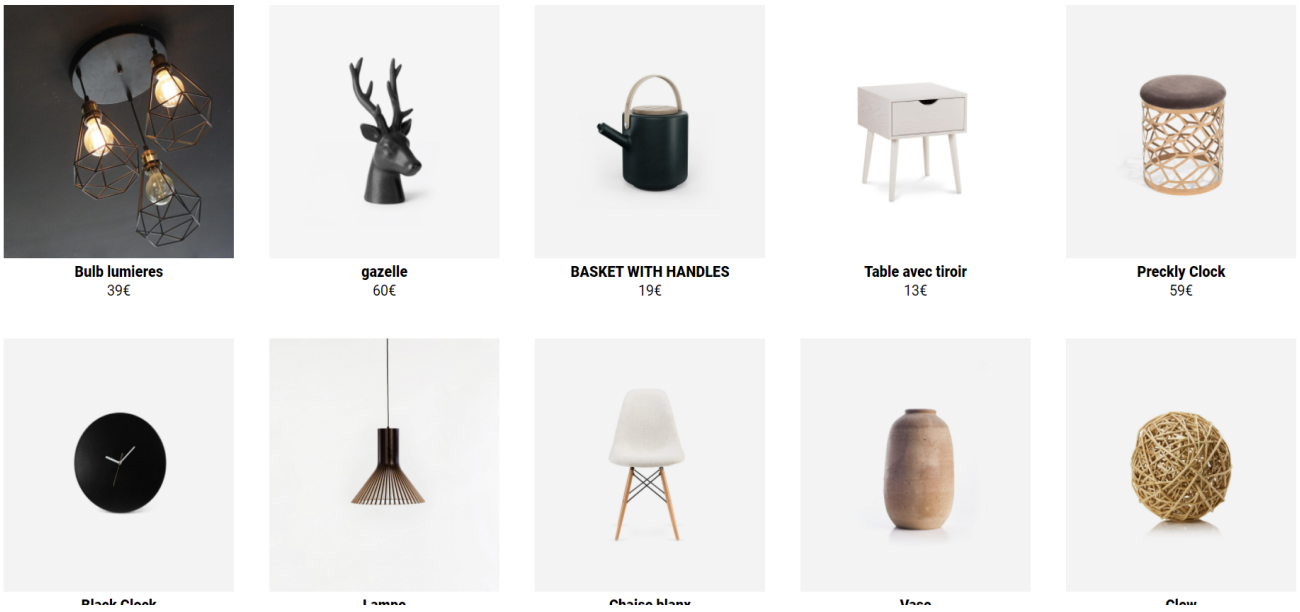
```php
public function showProduct()
{
    $stmt = $this->pdo->prepare("SELECT * FROM products ORDER BY product_date DESC");
    $stmt->execute();
    $tab = $stmt->fetchALL(PDO::FETCH_OBJ);
    return $tab;
}
```

je commence par récupérer l'attribut de ma connexion à la base de données, après avoir fait ça j'utilise la fonction prepare() qui est une fonction intégré par php qui est utilisée pour préparer une instruction SQL à exécuter.

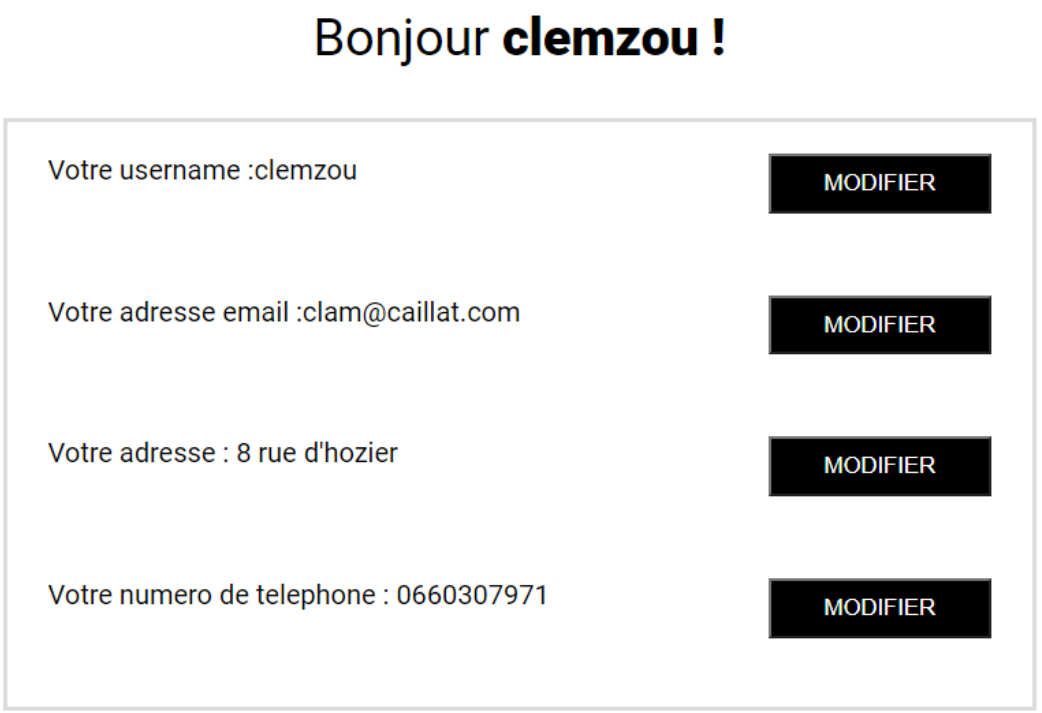
Après avoir fait cela je rédige ma fonction, qui est un SELECT, je l'exécute ensuite pour que cette requête soit traité par la BDD, et pour finir je fait un fetch de cette requête qui permet de rendre exploitable l'objet récupéré lors de la connexion après lui avoir passé ma requête SQL, je fais une boucle foreach() qui fournit une façon simple de parcourir des tableaux, cette boucle ne fonctionne que pour les tableaux et les objets, elle réitère chaque valeur du tableau une par une jusqu'à qu'il soit parcouru en entier.

à chaque itération pour chaque valeur, je récupère dans des variables les index qui m'intéressent et que je voudrais afficher, la en l'occurrence je récupère mes article en BDD et je décide choisir uniquement en index la description, l'image, le prix, le nom et l'id de l'article.

et ensuite je les affiche à mon bon vouloir, en rajouter des balises HTML et un code CSS pour mettre en forme l'affichage du tout.



En ce qui concerne “l’update”, cette opération est gérée par la requête qui porte le même nom UPDATE, pour modifier des éléments en base de données.
Dans le cadre de ma boutique en ligne, je m’en suis servi pour donner la possibilité à chaque utilisateur de mettre à jour leurs information personnelles (là en l'occurrence leurs nom et prénom) présents en base de données.



Je commence par rédiger la fonction qui est destinée à modifier ces données en BDD.

```php
public function updateProfilUsername ($newUsername,$id)
{

                $stmt_update=$this->pdo->prepare("UPDATE users SET username = ? WHERE id = ? ");
                $stmt_update->execute([$newUsername,$id]);


                $stmt_select=$this->pdo->prepare("SELECT * FROM users WHERE username = ?");
                $stmt_select->execute([$newUsername]);
                $row_username=$stmt_select->fetch(PDO::FETCH_OBJ);
                $_SESSION['user']=$row_username;



}
```

je commence par récupérer l'attribut qui contient la connexion à la BDD pour que je puisse communiquer avec et lui donner des instructions, après avoir fait ça je récupère l'id de l'utilisateur connecté qui est stocké dans une variable dite superglobale ce qui signifie qu'elles sont disponibles quel que soit le contexte du script, la en l'occurrence c'est la variable "$_SESSION" qui est utilisée pour stocker des tableaux associatif.

après avoir fait ça je rédige ma requête préparé, je dis de mettre à jour le nom et le prénom et je lui dis par quelles valeurs les remplacés et quelle élément ciblé avec l'id, après avoir fait ça je l'exécute et je remplace les valeurs stockés ma variable superglobale par les nouvelles que l'utilisateur à entrer tout en retournant un message comme quoi les informations ont bien été mises à jour.

l'utilisateur pourra ensuite voir sur son profil que les informations ont bien été modifiées.

Intégration

L'intégration de la maquette à été faite en utilisant le langage de balisage HTML. HTML étant limité à la création de simples représentations structurées des pages, c'est avec le langage CSS que je défini le style désiré en accord avec la charte graphique.
Je le fais par le biais de feuilles de style personnalisées propres à chaque page. Ce choix à l'avantage d'offrir une maintenabilité du code plus aisé car les modifications apportées impactent seulement la page lié à la feuille de style concernée.

pour ajouter du CSS à une balise HTML il suffit de lui donner une "class" que l'on nommera comme on le souhaite (de préférence avec un nom bien explicit pour pouvoir mieux s'y retrouver après), avec cela je peux ciblé cette balise via le nom qu'elle porte directement dans mon fichier CSS.

je commence par "linké" mon fichier CSS sur la page dans laquelle je désire ajouter du style, cela se fait avec une balise <link> à l'intérieur de la balise <head>

```
<link rel="stylesheet" href="../style/css/boutique.css">
```

Après avoir fait cela, je donne une classe à n'importe quelle balise et c'est comme ça que je lui attribue du style.

```
<div class='titrecaserouge'>
    Ajouter un article
</div>
```

```
.titrecaserouge {
    color: #F7EBE8;
    font-size: 20PX;
    font-family: 'Poppins', sans-seri
    font-weight: bolder;
}
```

Responsive design

une grande importance est portée sur l'adaptation de l'interface pour que la plateforme puisse être utilisée à partir de tous types de support.
Ces derniers regroupent principalement les ordinateurs, tablettes et smartphones.
Pour ce faire, j'utilise des Media Queries. En appelant les classes que je veux cibler et en définissant les largeurs d'écran maximal ou minimal, je peux facilement adapter le contenu affiché en changeant sa taille, masquer ou afficher du contenu ciblé, voici un exemple des media queries que j'ai fais sur RECON :

```css
/* ////////////////////////////////////////////////////// RESPONSIVE PAGE CONNEXION INSCRIPTION */

@media screen and (max-width: 860px) {
    .caseinscription {
        display: flex;
        align-items: center;
        justify-content: center;
    }
    .caseconnexion {
        display: flex;
        align-items: center;
        justify-content: center;
    }
    .buttonconnexion {
        margin-bottom: 100px;
    }
    .caseconnexioninscription {
        flex-direction: column;
    }
    .connexionsepareinscription {
        display: none;
    }
```

je défini que la largeur maximale que je souhaite est de 860px donc ces paramètres seront affectés aux classes ciblé qu'une fois que la largeur d'écran passe en dessous de 860px, voici un exemple :

Plus de 1060 px

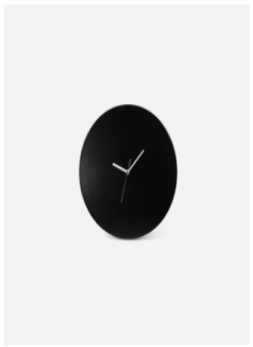
Bulb lumieres
39€

gazelle
60€

BASKET WITH HANDLES
19€

Table avec tiroir
13€

Preckly Clock
59€



Black Clock
35€

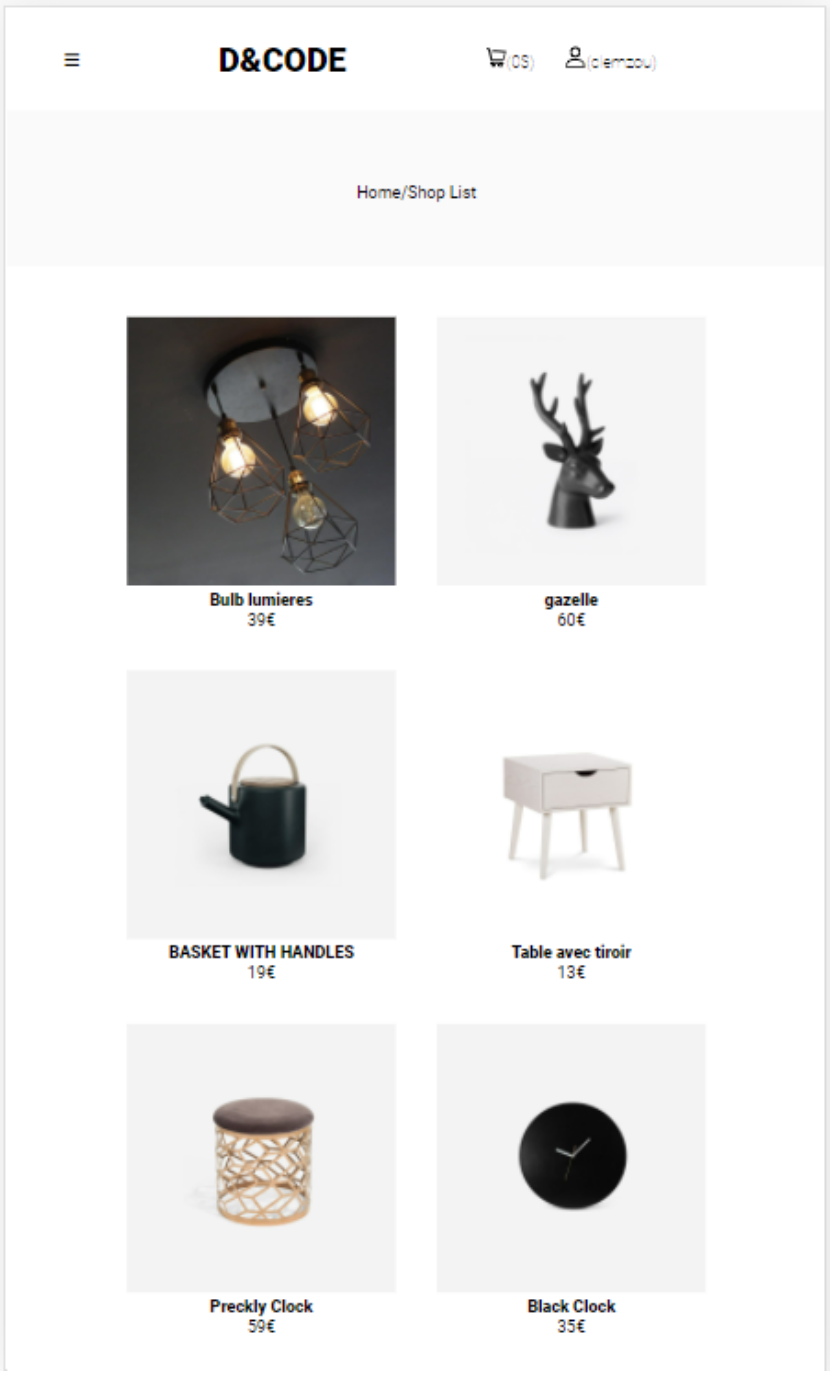
Lampe
35€

Chaise blanx
28€

Vase
29.89€

Clew
12€

Moins de 1060 px



Javascript et AJAX

Dans le but d'agrémenter l'expérience utilisateur et d'apporter du dynamisme au site, nous avons eu recours au langage Javascript.
Utilisé côté client il nous permet de manipuler le Document Object Model (DOM), une représentation structurée sous forme d'arborescence de nœuds, de tout le contenu d'un document HTML. Grâce à cette interface, nous pouvons exécuter du code en réponse à un événement déclenché de la part du l'utilisateur, agir sur les différents éléments d'une page et les modifier, ou récupérer des données d'un serveur distant en effectuant des requêtes sans que cela n'interfère avec l'affichage et le comportement de la page existante.

Dans le cas de ma boutique en ligne je me suis servi de la librairie JQUERY qui est une bibliothèque JavaScript libre et multiplateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web.

Dans ma boutique, j'ai utilisé JS pour ne pas avoir à rafraîchir la page a chaque communication à ma base de données, j'ai utilisé la méthode AJAX.

Tout d'abord, je commence par créer mon fichier JS que je vais "linké" avec une balise <script> sur la page où je souhaite ajouter du JS ainsi que pour mon cas la librairie JQUERY pour pouvoir utiliser ces fonctions :

```
</ul>
<script src="https://code.jquery.com/jquery-3.6.0.min.js" integrity="sha256-/xUj+3OJU5yExlq6GSYGSHk7tPXikynS7ogEvD
<script src="../../Script/user.js"></script>
```

j'ai utiliser AJAX pour qu'un utilisateur puisse se connecter au site :

Tout d'abord, j'attribue des "id" aux balises que je souhaite cibler dans mon fichier JS.

```
<input class='inpt' id="mail" type = 'mail' name = 'e-mail' >
```

dans le cas de cette <input>, je lui attribue comme "id" mail comme on peut voir.

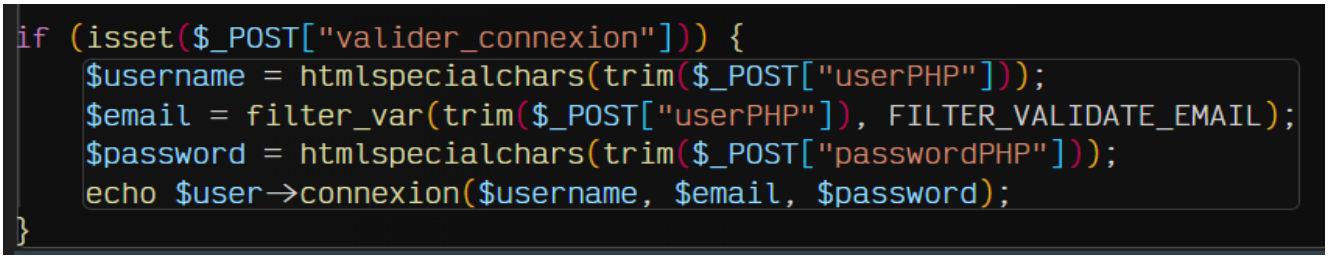
Pour ce qu'il s'agit de la partie JS :

ici dans le JS je défini que dès que l'utilisateur cliquera sur la balise qui a pour "id" connexion, il récupérera la valeur des input qui ont pour "id" mail et password avec la fonction de JQUERY qui permet de récupérer les valeurs des balises ciblés la syntaxe est la suivante : $("#id").val().

après avoir fait cela je vérifie que les input ont bien été rempli par l'utilisateur en posant la condition que les valeurs sont différentes de nul,

Si cette condition est bien vérifiée, je lance le AJAX.
Je spécifie en URL la page sur laquelle je veux que l'AJAX envoie les données, là en l'occurrence "action.php" qui est une page intermédiaire qui

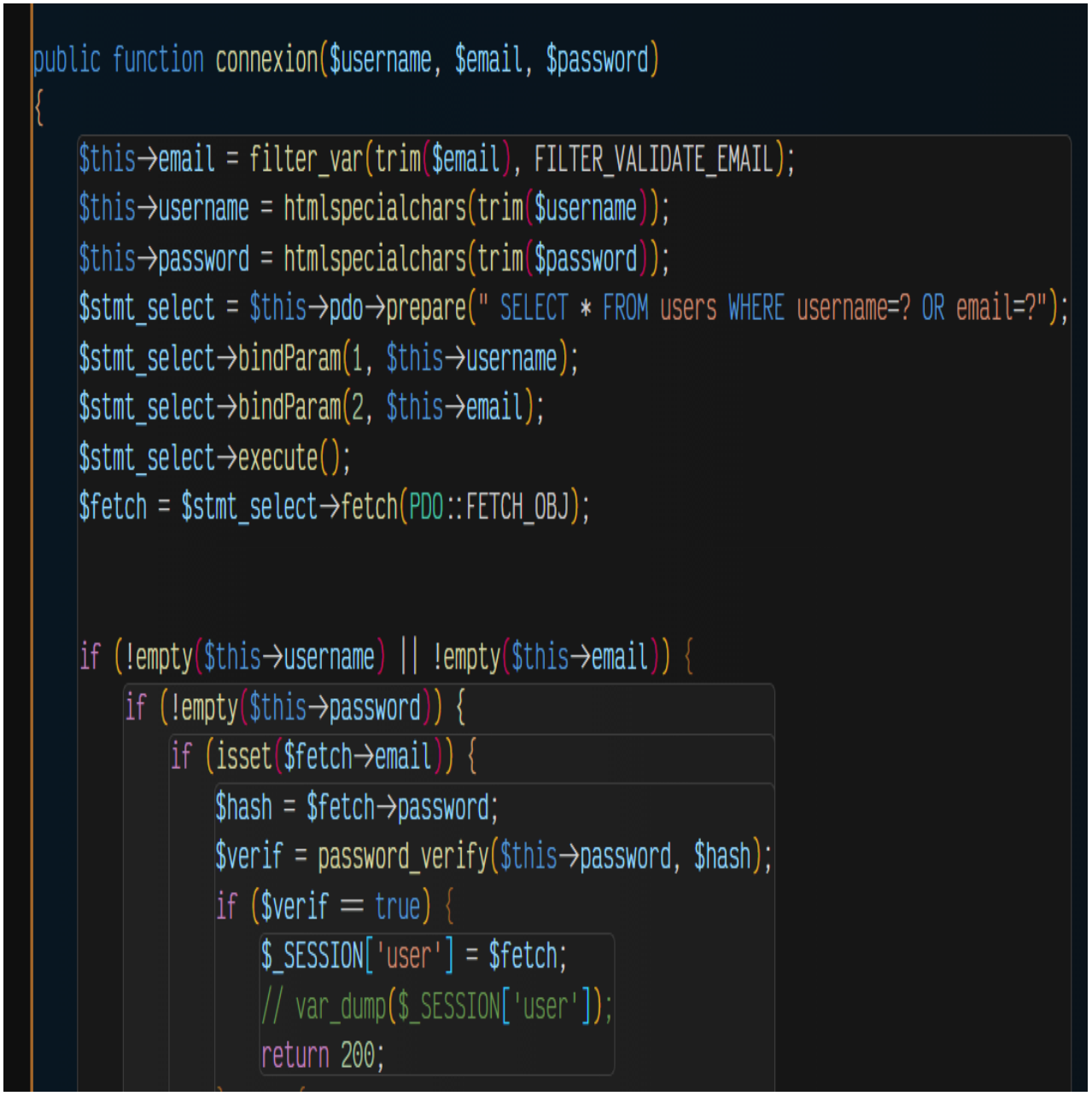
traitera les données pour les envoyer à ma fonction PHP dédiée à la connexion, avec la “method“ je défini si je souhaite les envoyer en ‘POST’ ou en ‘GET’.
Après cela je défini ce que je vais stocké dans ma variable superglobale “POST” et comment je vais l’appelé.



```php
if (isset($_POST["valider_connexion"])) {
    $username = htmlspecialchars(trim($_POST["userPHP"]));
    $email = filter_var(trim($_POST["userPHP"]), FILTER_VALIDATE_EMAIL);
    $password = htmlspecialchars(trim($_POST["passwordPHP"]));
    echo $user->connexion($username, $email, $password);
}
```

après avoir fait cela je rédige sur ma page action une condition pour vérifier si les données en bien étaient envoyer avec le booléen qui a pour index ‘connexion’, et ensuite je les passent en paramètre de ma fonction connexion.

Ensuite ces données parcourt ma fonction php :



```php
public function connexion($username, $email, $password)
{
    $this->email = filter_var(trim($email), FILTER_VALIDATE_EMAIL);
    $this->username = htmlspecialchars(trim($username));
    $this->password = htmlspecialchars(trim($password));
    $stmt_select = $this->pdo->prepare(" SELECT * FROM users WHERE username=? OR email=?");
    $stmt_select->bindParam(1, $this->username);
    $stmt_select->bindParam(2, $this->email);
    $stmt_select->execute();
    $fetch = $stmt_select->fetch(PDO::FETCH_OBJ);


    if (!empty($this->username) || !empty($this->email)) {
        if (!empty($this->password)) {
            if (isset($fetch->email)) {
                $hash = $fetch->password;
                $verif = password_verify($this->password, $hash);
                if ($verif == true) {
                    $_SESSION['user'] = $fetch;
                    // var_dump($_SESSION['user']);
                    return 200;
```

Si toutes les vérifications ont été effectuées, je stocke les informations dans des variables superglobales '$_SESSION' et pour finir envoyer l'utilisateur sur une autre page, dans notre cas, la page des produits.

Sécurité

Une veille concernant les failles de sécurité web a été faite durant la réalisation du site. J'ai parcouru le web à travers différents sites internet pour recouper les informations sur les failles web les plus connues.

Voici les sources dont je me suis servis :

https://www.cert.ssi.gouv.fr/

https://vigilance.fr/?langue=1

https://security.stackexchange.com/

https://owasp.org/

Faille INCLUDE

Il s'agit d'une faille très dangereuse. Comme son nom l'indique, elle exploite une mauvaise utilisation de la fonction include.

La plupart du temps, cette fonction est utilisée pour exécuter du code PHP qui se situe dans une autre page, permettant de se connecter à une base de données.

Il existe deux types de failles include :

A distance : il s'agit de la faille include par excellence. C'est à la fois la plus courante et la plus facilement exploitable.

En local : cela revient à inclure des fichiers qui se trouvent sur le serveur du site. Une personne mal intentionnée pourra s'emparer facilement de votre fichier contenant vos mots de passes.

Pour se protéger de cette faille, rien de mieux que de la tester ! Il vous suffit d'inclure une page qui
n'existe pas. Si l'URL de celle-ci est vulnérable, un message d'erreur vous sera transmis venant de PHP.

Injection SQL

Cette faille survient lors de la modification d'une requête SQL et consiste à injecter des morceaux de code non filtrés, généralement par le biais d'un formulaire. Cela revient à détourner la requête et lui faire faire autre chose que ce pour quoi elle a été conçue. Cette manipulation donne donc accès à vos données telles que les login, mots de passe ou adresses e-mail.

Faille UPLOAD

Cette faille peut apparaître lors de l'upload de fichiers sur un site : photo de profil, document pdf, image dans un message, etc. Elle profite de l'action effectuée pour mettre en ligne des fichiers malveillants PHP qui vont permettre au « hacker » de prendre le contrôle total de notre site.

Pour éviter cette vulnérabilité, il est important de :

Empêcher les utilisateurs d'envoyer des fichiers lorsque cela n'est pas une fonction primordiale pour votre site ou application

Interdire l'exécution de code depuis le dossier dans lequel sont stockés les fichiers uploadés sur votre site

Vérifier et autoriser l'extension des fichiers que vous tolérez via une liste blanche.

Faille XSS

Une faille XSS consiste à injecter du code qui pourra être interprété directement par le navigateur Web. Ce dernier ne fera ainsi aucune différence entre le code du site et celui injecté par le pirate.
Les effets sont bien entendu assez embêtants puisque vous risquez de faire face à des redirections vers un autre site, du vol de cookies ou encore une modification du code de votre page.

Pour vous protéger des XSS, vous devez remplacer les caractères pouvant être compris par le navigateur comme des balises par leur entité HTML. En procédant ainsi, le navigateur affichera mot à mot le caractère et ne cherchera plus à l'interpréter. En PHP, vous pouvez utiliser les fonctions
htmlentities ou htmlspecialchar.

ATTAQUE PAR FORCE BRUTE

Cette méthode consiste à trouver le mot de passe ou la clé cryptographique d'une personne afin de pouvoir accéder à un service en ligne, à des données personnelles, voire à un ordinateur.
Il est donc indispensable d'utiliser des mots de passe forts pour vos sites et comptes utilisateurs afin de rendre complexe l'attaque par une personne tiers.

3 conseils utiles pour renforcer vos mots de passe :

Utiliser des lettres minuscules, des majuscules, des chiffres et des caractères spéciaux (notez qu'il existe des générateurs automatiques de mots de passe sur internet)

Renouveler souvent ses mots de passe.

Utiliser des mots de passe différents pour chaque site.

Merci pour votre attention.