# TITRE: Développeur Web et Web Mobile


MAGASSOUBA AHMED SEKOU

# Table des matières

# Compétences du référentiel couvertes par le projet

Le projet couvre les compétences énoncées ci-dessous.

Pour l'activité 1, **"Développer la partie front-end d'une application web et web mobile en intégrant les recommandations de sécurité"**:

- **Maquetter une application**
- **Réaliser une interface utilisateur web ou mobile statique et adaptable**
- **Développer une interface utilisateur web dynamique**
- **Réaliser une interface utilisateur avec une solution de gestion de contenu ou e_commerce .**

Pour l'activité 2, **"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 la partie back-end d'une application web ou web mobile**
- **Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e_commerce**

# Résumé

Dans le cadre de ma formation pour valider la **Project Pool** 2, j'ai réalisé le projet Boutique en ligne.

Le projet consistait à réaliser un site de ecommerce avec une liste de fonctionnalités minimum

- ❖ Créer une maquette graphique
- ❖ Conception de la base donnée
- ❖ Mise en avant des produits phares / derniers produits mis en ligne
- ❖ Chaque produit doit avoir une page complète générée dynamiquement (nom, image, prix, description, ajouter au panier...)
- ❖ Création de comptes d'utilisateurs
- ❖ Gestion du profil utilisateur (informations, historique d'achat,consultation du panier...)
- ❖ Gestion des produits à l'aide de back office pour les admins
- ❖ Gestion des catégories et des sous catégories de produits
- ❖ Barre de recherche de produits
- ❖ Validation du panier (simulation du paiement)
- ❖ Design contemporain et respectant la charte graphique de votre entreprise

# Spécifications fonctionnelles

## ● Description de l'existant

Aucun site n'existait auparavant .Nous avons convenu pendant nos entretiens avec les collègues du groupe pour le projet quel genre de site nous allons développer , quels seraient les besoins , les fonctionnalités qui seront développées mais , la structure de la base de données mais également l'aspect graphique de la future application.

## ● Périmètre du projet

Le site sera réalisé en français et ce dernier devra être accessible sur différents supports, à savoir mobile,tablette et ordinateur.

## Cible adressée par le site internet

Le site s'adresse à des particuliers qui voudraient s'acheter une trottinette ou les accessoires qui vont avec.

## Arborescence du site

L'arborescence du site se décline comme suit:

- Page d'accueil
- Page connexion
- Page inscription
- Page tous les produits
- Page détail de produit
- Page sélection de produit
- Page mon compte
- Page détail commande
- Page panier
- Page choix de la livraison
- Page paiement
- Page de confirmation de commande
- Une partie back-office (Administration) est également prévue afin de permettre la gestion du site

# Description des fonctionnalités

## 1. Page d'accueil

On affiche les produits en promotion , les meilleur vente et des nouveaux arrivages

## 2. Authentification

Nous avons convenu de mettre en place un système de formulaire " inscription/connexion" qui permet aux utilisateurs du site de s'authentifier et de sauvegarder leurs données.

## 3. Récupération du mot de passe en cas d'oubli

L'utilisateur devrait être en mesure de récupérer son compte en cas d'oubli de son mot de passe . En cliquant sur le lien mot de passe oublié il sera demandé de rentrer son mail , et un lien lui sera envoyé pour pour qu'il définisse un nouveau mot de passe.

## 4. Catalogue produit et filtre

Une page doit permettre d'afficher l'ensemble des produits disponibles. cet affichage devra comprendre la photo du produit , le nom du produit ainsi que son prix.

Un filtre doit être implémenté afin de trier les articles en fonction de leur catégorie  et sous catégorie

## 5. Fiche produit

L'utilisateur devra être en mesure d'accéder à une fiche produit .Cette dernière devra comprendre:

- ❖ Une ou plusieur photos du produit
- ❖ Le nom du produit
- ❖ Le prix
- ❖ La description du produit
- ❖ La note du produit
- ❖ Les commentaires clients
- ❖ La possibilité de commenter le produit si on est connecté
- ❖ La possibilité d'ajouter le produit au panier

## 6. Evaluation des produits et commentaires client

L'utilisateur devra être en mesure d'évaluer le produit grâce à un système de notation sur 5 étoiles. 5 étoiles signifiant "Très satisfait du produit" et 1 étoile "Pas du tout satisfait du produit".

L'utilisateur pourra également laisser un commentaire sur le produit afin de faire part de ses appréciations sur ce dernier.

## 7. Panier client

L'utilisateur devra être en mesure de sélectionner un article et de le mettre dans son

panier dans le but final de passer commande sur le site.

Ce panier devra afficher les éléments suivants:

- ❖ Une photo de l'article
- ❖ Le prix de l'article
- ❖ La quantité demandée par le client
- ❖ Le total des articles sélectionnés
- ❖ Un bouton de validation du panier débouchant sur le processus de commande.

Le client devra être en mesure d'augmenter ou diminuer la quantité demandée. Il aura aussi la possibilité de supprimer le un article du panier, voire l'intégralité .

## 8. Fonctionnalité de recherche produit

Le client devra être en mesure d'effectuer une recherche de produit ou de catégorie dans un champ prévu à cet effet. Afin de faciliter la recherche, un système de recherche avec autocompletion doit être mis en place.

## 9. Espace client

L'utilisateur aura accès à un espace client dans lequel il lui sera possible de consulter et modifier ses informations. A savoir son nom, son prénom, son adresse email, son mot de passe (hors authentification facebook) mais également son adresse postale.

Il aura également la capacité de consulter ses précédentes commandes et de voir

leur statut (annulée, en cours de traitement, expédiée).

## 10. Back office

Le gérant du site devra avoir accès à un espace sécurisé lui permettant d'administrer le site.

A. Ajouter un produit

L'administrateur sera en mesure d'ajouter un nouveau produit sur le site en entrant toutes les informations concernant ce dernier

B. Gérer les produits

L'administrateur sera en mesure de gérer les produits c'est-à-dire modifier les informations d'un produit , le désactiver pour qu'il ne soit plus visible sur le site , l'activer, le mettre en promotion , modifier le stock ou le supprimer.

Lors de la création du produit, ce dernier sera en mesure de:

- ❖ Donner un titre au produit
- ❖ Définir le prix du produit
- ❖ Décrire le produit
- ❖ Uploader une image du produit
- ❖ Établir le stock du produit.
- ❖ Décider si ce produit doit être mis en avant ou non.

C. Gérer les utilisateurs

L'administrateur sera en mesure de gérer les utilisateurs en changeant leur droit d'utilisateur à modérateur ou a administrateur, il peut aussi les interdire l'accès au site

D. Gérer les droits utilisateurs

L'administrateur sera en mesure de gérer les droit de produit c'est-à-dire créer

des droits, les modifier et les supprimer.

E. Gérer les catégories

L'administrateur sera en mesure de gérer les catégories de produit c'est-à-dire créer

des catégories, les modifier et les supprimer.

F. Gérer les sous-catégories

L'administrateur sera en mesure de gérer les sous-catégories de produit c'est-à-dire créer des sous-catégories , les modifier et les supprimer.

G. Gérer les commandes

L'administrateur sera en mesure de gérer les commandes de produit c'est-à-dire , une fois que la commande sera enregistrée , il pourra gérer l'état de la commande de la préparation à l'expédition.

Il aura la possibilité de visualiser les commandes qui ont été effectuées sur le site.

Ceci signifie que l'administrateur aura accès aux informations

suivantes:

- La date de la commande
- Le nom et le prénom de la personne ayant passé commande
- Son adresse
- Le point relais choisis
- Les différents articles commandés: quantité, taille etc...
- L' administrateur pourra également changer le statut de la commande, par exemple passer la commande en commande expédiée lorsque celle-ci a été remise au prestataire de livraison.

## 11. Livraison

## 12. Solution de paiement

La solution de paiement choisie est celle proposée par Stripe. Cette solution permettra au client de procéder au paiement de manière sécurisée par carte bancaire.

## 13. Notification du client

Un mail devra être envoyé au client afin de lui confirmer que sa commande a bien été passée. Ce mail devra contenir un récapitulatif de la commande et l'adresse de livraison qui a été choisi.

# Spécification techniques

## Choix techniques et environnement de travail

Technologies utilisées pour la partie back-end :

- Le projet sera réalisé avec le langage PHP ( Hypertext Preprocessor)
- Base de données SQL
- Gestionnaire de dépendance: Composer

Technologies utilisées pour la partie front-end:

- Le projet sera réalisé avec du HTML , CSS ,Bootstrap.
- Et enfin Javascript afin de dynamiser le site et d'améliorer l'expérience utilisateur.

L'environnement de développement est le suivant:

- Editeur de code: VSCode
- Outil de versioning: GIT, Github.
- Maquettage: Figma

Du point de vue de l'organisation, j'ai utilisé Trello afin de découper le projet en une

multitude de tâches à réaliser et de définir leur ordre de priorité.

Concernant le back-office, j'ai utilisé le composant bootstrap-table pour les tableaux, me permettant de faire des tris , du filtrage,des recherches et de la pagination de façon relativement simple.

# Architecture du projet

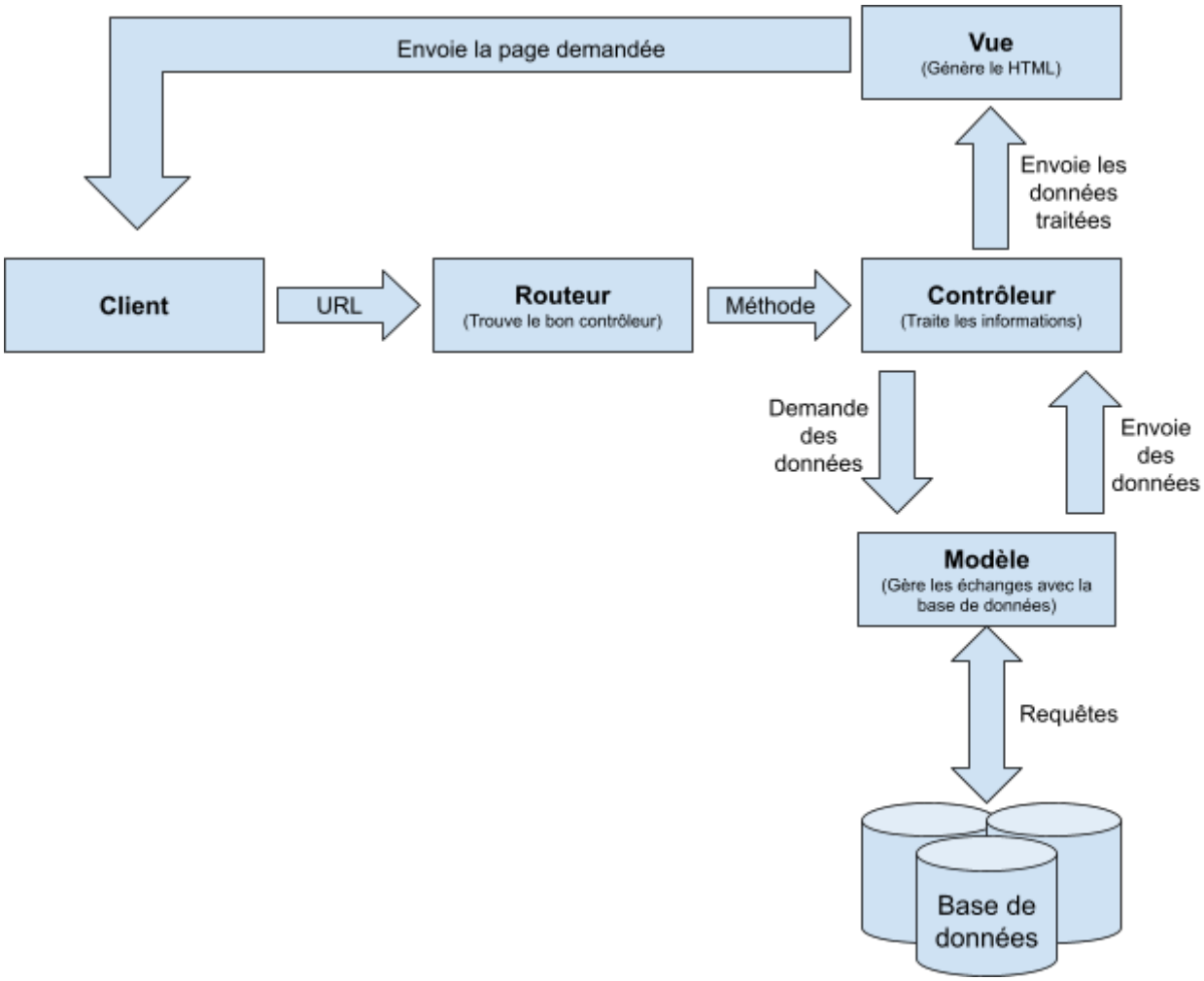
Le projet est développé sous **PHP** avec l'utilisation des classes et d'un design pattern de type **MVC(Model-View-Controller).**

- ## L'architecture MVC

  **L'architecture MVC est l'une des architectures les plus utilisées pour les applications Web, elle se compose de 3 modules:**

  - **Modèle:** noyau de l'application qui gère les données, permet de récupérer les informations dans la base de données, de les organiser pour qu'elles puissent ensuite être traitées par le contrôleur.
  - **Vue:** composant graphique de l'interface qui permet de présenter les données du modèle à l'utilisateur.
  - **Contrôleur:** composant responsable des prises de décisions, gère la logique du code, il est l'intermédiaire entre le modèle et la vue.

## Schéma illustrant le design pattern MVC

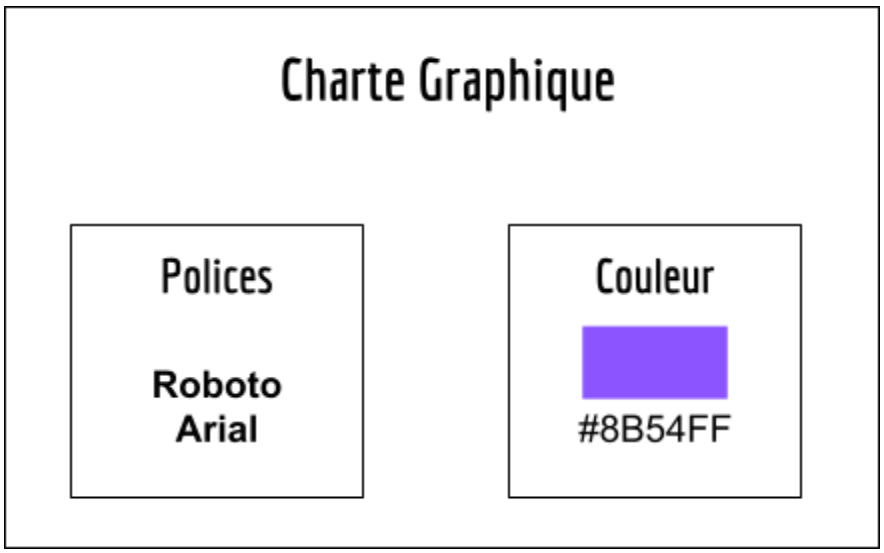


# Réalisations

## 1. Charte graphique

Les polices d'écriture sont les suivantes: Roboto et Arial

La couleur dominante est la suivante : `#8B54FF`




## 2. Maquette

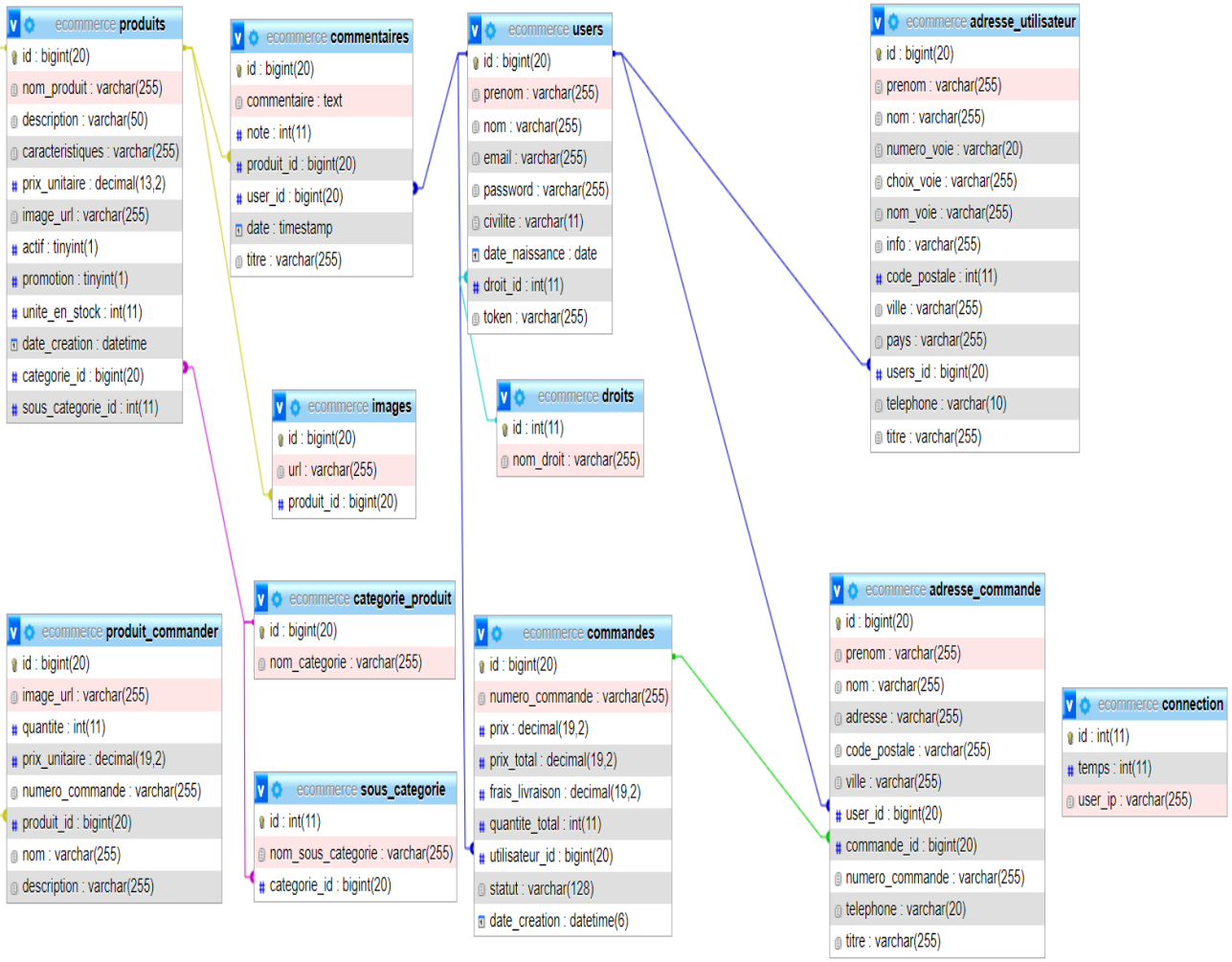
La maquette a été réalisée avec le logiciel gratuit Figma. J'ai fait le design en m'inspirant des exemples sur le net pour avoir un site au design contemporain et moderne.

Vous trouverez la maquette dans les annexes du dossier.

## 3. Conception de la base de données

Au regard de fonctionnalités qui seront mis en place , j'ai développé la base de données suivantes :

**Modèle physique de données**




Vous trouverez également dans les annexes **le modèle conceptuel de données** ainsi que le **modèle logique de données**.

Comme illustré ci dessus, on peut voir que la base de données s'articule autour de 3 tables principales. Tout d'abord , la table **users** qui va permettre d'identifier les clients. Cette table est liée à différentes tables dont la table **address**, la table **comment** et également la table **order** qui va me permettre de par exemple de relier une commande au client concerné.

Il y a ensuite la table **product** qui va être reliée à la table **category, comment**, **order_details**. Toutes ces liaisons vont permettre de déterminer le stock du produit, à quelle catégorie il appartient, de voir les commentaires et lorsqu'une commande

sera passée à savoir de quels produits se compose cette commande.

Enfin, il y a la table **order** qui permet de recueillir les informations relatives aux

commandes passées. Cette dernière est reliée à la table users de manière à pouvoir

identifier qui a passé commande. Elle est également reliée à la table **order_detail** afin d'obtenir les détails de la commande et à la table delivery qui permet de récupérer les informations relatives à la livraison de la commande

# 4. Extrait de code significatif

## A. Panier Client

Pour la gestion du panier client j'ai choisi de stocker ce dernier en session.

Pour ce faire j'ai tout d'abord créé une classe **PaniersModel** qui étend mon modèle générique **"Model"** lui donnant ainsi accès aux méthode du Modèle générique et la connexion a la base de données.

Dans ma classe PaniersModel j'ai implémenter différentes méthodes pour **ajouter un article** , **le supprimer** , **modifier les quantités.**

Pour le stockage en session, j'ai utilisé la global **"$_SESSION"** et j'ai stocké dans **"$_SESSION['panier']"**

Dans le **constructeur** de ma classe **"PaniersModel "** j'instentie $_SESSION["panier"] a un tableau vide "**$_SESSION["panier"] = aray()**"

**Voici un exemple de méthode pour ajouter un article au panier:**

```php
/**
 * Méthode pour ajouter un produit au panier
 *
 * prend en paramètre l'id du produit
 */
public function ajouter_au_panier($id)
{
    // Si le produit est déja dan le panier on l'incremente
    if (isset($_SESSION['panier'][$id])) {
        $_SESSION['panier'][$id]++;

        // sinon on l'initialise a 1
    } else {
        $_SESSION['panier'][$id] = 1;
    }
}
```

**$id** représente l'id du produit qu'on ajoute au panier

Si **$_SESSION["panier"][$id]** existe cela signifie qu'on a déjà ajouté le produit dans le panier donc au prochain ajout on va incrémenter la quantité et ajouter +1 a sa valeur

Sinon on va ajouter le produit au panier avec une quantité égale à 1.

Le panier n'étant pas sauvegardé dans la base de données , j'ai par la suite créé une méthode qui va récupérer toutes les informations liées aux produits présents dans le afin de les afficher ainsi que les quantités et le prix total des articles.

## B. Intégration stripe

Stripe est une solution de paiement qui va permettre à au client de régler sa commande en carte bancaire Visa ou MasterCard.

Son intégration est organisée autour d'une API de type REST.

Afin de réaliser l'intégration de la solution de paiement de **Stripe** , j'ai créé un controller "**PayementController**" qui va gérer le paiement du client .

J'ai installer le gestionnaire de package composer, et par la suite grâce à composer j'ai installer la librairie de stripe **composer require stripe/stripe-php**

Dans mon controlleur **"PayementController"** j'ai créé une méthode dans laquelle j'utilise ma **secret key de test fourni par stripe après la création de mon compte** pour envoyer une **intention de paiement a stripe** avec le **montant à régler.**

## Voici le code utilisant stripe  dans mon controller

```php
/**
 * Méthode de payement avec stripe
 */
public function index()
{

    if (isset($_SESSION['user']) && isset($_SESSION['user']['id']) && !empty($_SESSION['user']['id'])
     && isset($_SESSION['totalPayer']) && !empty($_SESSION['totalPayer'])) {

        require_once('vendor/autoload.php');

        $prix = (float)strip_tags($_SESSION['totalPayer']);

        $key = 'sk_test_51KcitxK6jN51enOQnkpkyLyEjYsjgQ20BNiTbIpQJYloKdtKk6rzaRy5TGiLeAt6QQ1ZHMjnrvbrLehwxmSx98zD00jFa3M9Hu';

        // On instantie stripe
        \Stripe\Stripe::setApiKey($key);

        // Un crée l'intention
        $intent = \Stripe\PaymentIntent::create([
            'amount' => $prix * 100,
            'currency' => 'eur'
        ]);


        $this->render('payement/index', ['intent' => $intent]);
    } else {
        header('Location: /boutique-en-ligne/main');
    }
}
```

## Coté front stripe s'occupe de la génération d'un formulaire de paiement

```php
<section class="payement">


    <form action="" method="post">
        <div id="errors"></div> <!-- Contiendra les messages d'erreur au payement -->
        <label for="nomCarte">Titulaire de la carte</label>
        <input type="text" name="nomCarte" id="nomCarte" placeholder="Saisissez le nom sur la carte">
        <hr>

        <div id="card-elements"></div><!-- Contiendra les information de la carte -->
        <hr>
        <div id="card-errors" role="alert"></div><!-- Contiendra les erreurs relatif a la carte -->
        <div>
            <button id="card-button" type="button" data-secret="<?= $intent['client_secret'] ?>">Payer</button>
        </div>
    </form>

</section>
```

### C. Php mailler

## 5. Veille sur les vulnérabilités de sécurité

### A. Exposition des données sensibles

Les données sensibles ne sont pas seulement en transit, elles sont aussi stockées en base de données. Pour protéger certaines données stockées sur une application,il est possible d'utiliser des algorithmes de hachage.

L'intérêt des algorithmes de hachage est qu'ils permettent de calculer une empreinte(ou ``hash'') d'une chaîne de caractères, par exemple. Cette empreinte est utile pour éviter de stocker en clair le mot de passe dans la base de données.

**Comment éviter d'exposer les données stockées ?**

- Sécurisez votre base de données avec le chiffrement.
- Utilisez des algorithmes de hachage sécurisés tels que **Argon5**, **Scrypt**, **Bcrypt** et **PBKDF2**.
- Le masquage des données peut être utilisé pour sécuriser les données sensibles d'une base de données.

Dans le projet est utilisé algorithmes de hachage Aragon pour haché les données sensible comme le mot de passe

```php
//  On chiffre le mot de passe
$pass = password_hash($_POST['password'], PASSWORD_ARGON2I);
```

## B. Valider les entrées

Cela consiste à limiter ce que l'utilisateur peut mettre dans la zone de texte.Cela n'empêchera pas l'injection, mais c'est une mesure que vous pouvez mettre en place pour limiter des attaques de base. En effet, les caractères spéciaux spécifiques à certains langages ne pourront pas être utilisés.

### ❖ RegEx(expressions régulières)

Les expressions régulières sont un outil très utile pour les développeurs. Ils permettent de trouver, d'identifier ou de remplacer un mot, un caractère ou tout autre type de chaîne.

Dans le projet j'ai utilisé la fonction **preg_match()** de php et les expressions régulières **regex** pour valider les entrées , j'ai ainsi traité l'ensemble des champ avec lesquels l'utilisateur interagit pour avoir le type de données attendu pour chaque champ.

## C. File upload

❖ Explication

Le script Upload permet le transfert des fichiers depuis votre machine qui est le client vers le site qui est le serveur, mais souvent le script d'upload contient des vulnérabilités.

La **faille upload** est une faille permettant d'uploader des fichiers avec une extension non autorisée, cette faille est due à la mauvaise configuration du script d'upload ou à l'absence complète de sécurité. Celle-ci est généralement présente dans les scripts d'upload d'images. C'est une des failles les plus dangereuses.

Le but de cette faille est d'uploader un fichier avec une extension non autorisée. (Par exemple un code php) de façon à avoir un accès au serveur cible. Si le formulaire d'upload de votre site n'est pas sécurisé, alors un pirate informatique pourrait sans problème s'amuser à uploader un fichier PHP malveillant (web shell par exemple) qui lui permettrait de prendre le contrôle total de votre application web, et de votre serveur.

❖ Solution

Comment s'en prémunir ?

➔ Vérifier le type MIME

Certains scripts Upload ne font que vérifier si le MIME correspond aux types de fichiers autorisés par contre une vérification de mime n'est pas suffisante parce qu'un pirate peut contourner cette vérification

➔ Vérifier la taille du fichier

➔ Interdire les doubles extensions

➔ Générer un nom aléatoire pour le fichier uploadé et enregistré le nom dans la base de donnée

➔ Ne pas permettre l'écrasement de fichier

➔ Assigner les bonnes permissions au répertoire

Dans le projet pour l'upload des images du produit j'ai configurer le .htaccess d'apache pour éviter les doubles extensions et dans le controlleur produit j'ai effectué les vérifications nécessaire pour éviter une faille upload

**.htaccess avec les condition et règle concernant le dossier d'upload d'images**

```
1   RewriteEngine On
2
3   RewriteCond %{REQUEST_URI} !\.(png|jpg|jpeg|svg)$
4
5   RewriteRule .*$ - [F]
```

```php
// Traitement de l'image
// On verifi qu'on a une image
if (isset($_FILES['image']) && $_FILES['image']['error'] === 0) {

    // On met les conditions d'acceptation des images
    //On met en place un tableau d'extension et de type mime "extension" => "type mime"
    $extensionsTab = [
        "jpg"  => "image/jpeg",
        "jpeg" => "image/jpeg",
        "png" => "image/png"
    ];

    // On recupère le informations de l'image dans la glebale $_FILES
    $nomImage = $_FILES['image']['name'];
    $typeImage = $_FILES['image']['type'];
    $tailleImage = $_FILES['image']['size'];

    // On recupère l'extension a l'aide de la fonction pathinfo()
    $extension = strtolower(pathinfo($nomImage, PATHINFO_EXTENSION));

    // On verifie l'abscence de l'extension ou du type mime dans notre tableau d'extensions
    if (!array_key_exists($extension, $extensionsTab) || !in_array($typeImage, $extensionsTab)) {
        $_SESSION['erreur'] = "Le format d'image n'est pas pris en compte";
        echo "Le format d'image n'est pas pris en compte";
    }

    // l' image est correct
    // On va limité la taille de
    if ($tailleImage > 1024 * 1024) {
        $_SESSION['erreur'] = "L'image est trop volumineux, veuillez choisir une image de moins de 1Mo";
    }

    // On génère un nom unique pour l'image avec la fonction md5(uniqid())
    // uniqid est un timestamp
    // md5 permet de le chiffrer pourqu'on ai une chaine de carractère alléatoire
    $nouveauNom = md5(uniqid());

    $image = $nouveauNom . '.' . $extension;
    echo $image;
    // On génère un chemin complet d'accès
    //  __DIR__ => chemin complet vers les dossier dans lequel on se trouve
    $nouveauChemin =    __DIR__ . "/uploads/$image";
```

```php
//pour supriner  une image  unlink(__DIR__."nom de l'image"image
// On deplace le fichier du dossier temporaire dans lequel il se trouve pour
//Rle mettre dans le dossier uploads avec le nouveau non
if (!move_uploaded_file($_FILES["image"]["tmp_name"], $nouveauChemin)) {
    $_SESSION['erreur'] = "le changement d'image a echoué";
}

// on va protéger le fichier contre l'ecriture et l'execution' avec le chmod
// le premier chiffre après le 0 represente le propriétaire
// le deuxième le groupe
// troisième c'est le visiteur

chmod($nouveauChemin, 0644);
```

## D. Injection SQL

❖ Explication

Cette vulnérabilité permet à un attaquant d'injecter des données non maîtrisées qui seront exécutées par l'application et qui permettent d'effectuer des actions qui ne sont normalement pas autorisées.

Ce type d'attaque s'effectue généralement grâce aux champs présents dans les formulaires.

Dans le cas d'une attaque par injection SQL, au lieu de mettre un nom d'utilisateur et un mot de passe sur une page de connexion, un utilisateur malveillant entrera des données directement interprétées par le moteur SQL, ce qui lui permettra de modifier le comportement de votre application.

❖ Solution

Comment s'en prémunir ?

Pour prévenir les **injections SQL**, il faut faire appel aux **requêtes préparées**. Ce sont des requêtes dans lesquelles les paramètres sont interprétés indépendamment de la requête elle-même. De cette manière, il est impossible d'effectuer des injections. Dans tous les systèmes de gestion de bases de données, deux méthodes sont utilisées : **"prepare()"** qui prépare la requête et **"execute()"** qui exécute la requête avec les paramètres.

Dans le projet j'ai créé une méthode **"requete()"** qui prépare et exécute toutes mes requêtes pour éviter les injections SQL.

### Méthode requte()

```php
/**
 *  //Methode pour preparer et exécuter une requete
 *
 * @param string $sql LA REQUETE A EXECUTE
 * @param array|null $attributs UN TABLEAU DE VALEURS
 * @return $query
 */
public function requete(string $sql, array $attributs = null)
{

    //On recupère l'instance de Db
    $this->db = Db::getInstance();

    //On verifie si on a des attributs
    if ($attributs !== null) {
        //Requete préparée
        $query = $this->db->prepare($sql);
        $query->execute($attributs);
        return $query;
    } else {
        //Requete simple
        return $this->db->query($sql);
    }
}
```

## E. Faille XSS (Cross Site Scripting)

❖ Explication

La faille **XSS**, a l'origine CSS (Cross Site Scripting) changé pour ne pas confondre avec le CSS des feuilles de style (Cascading Style Sheet), est un type de faille de sécurité des sites Web, que l'on trouve dans les applications Web mal sécurisées.

Le  principe de cette faille est d'injecter un code malveillant en langage de javascript dans un site web vulnérable. Par exemple en déposant un message

dans un forum qui redirige l'internaute vers un faux site (**phishing**) ou qui vole ses informations (cookies).

La **faille XSS** permet d'exécuter des scripts du côté client. Ceci signifie que vous ne pouvez exécuter que du JavaScript, HTML et d'autres langages qui ne vont s'exécuter que chez celui qui lance le script et pas sur le serveur directement.

❖ Solution

Comment s'en prémunir?

Plusieurs techniques permettent de se protéger de la faille XSS :

➔ La fonction **htmlspecialchars()** convertit les caractères spéciaux en entités HTML.

➔ **htmlentities()** qui est identique à **htmlspecialChars()** sauf qu'elle filtre tous les caractères équivalents aux codage html ou javascript.

➔ **strip-tags()**, cette fonction supprime toutes les balise.

Dans le projet j'ai utilisé la fonction **strip-tags** pour prévenir les **faille XSS**

**Exemple:**

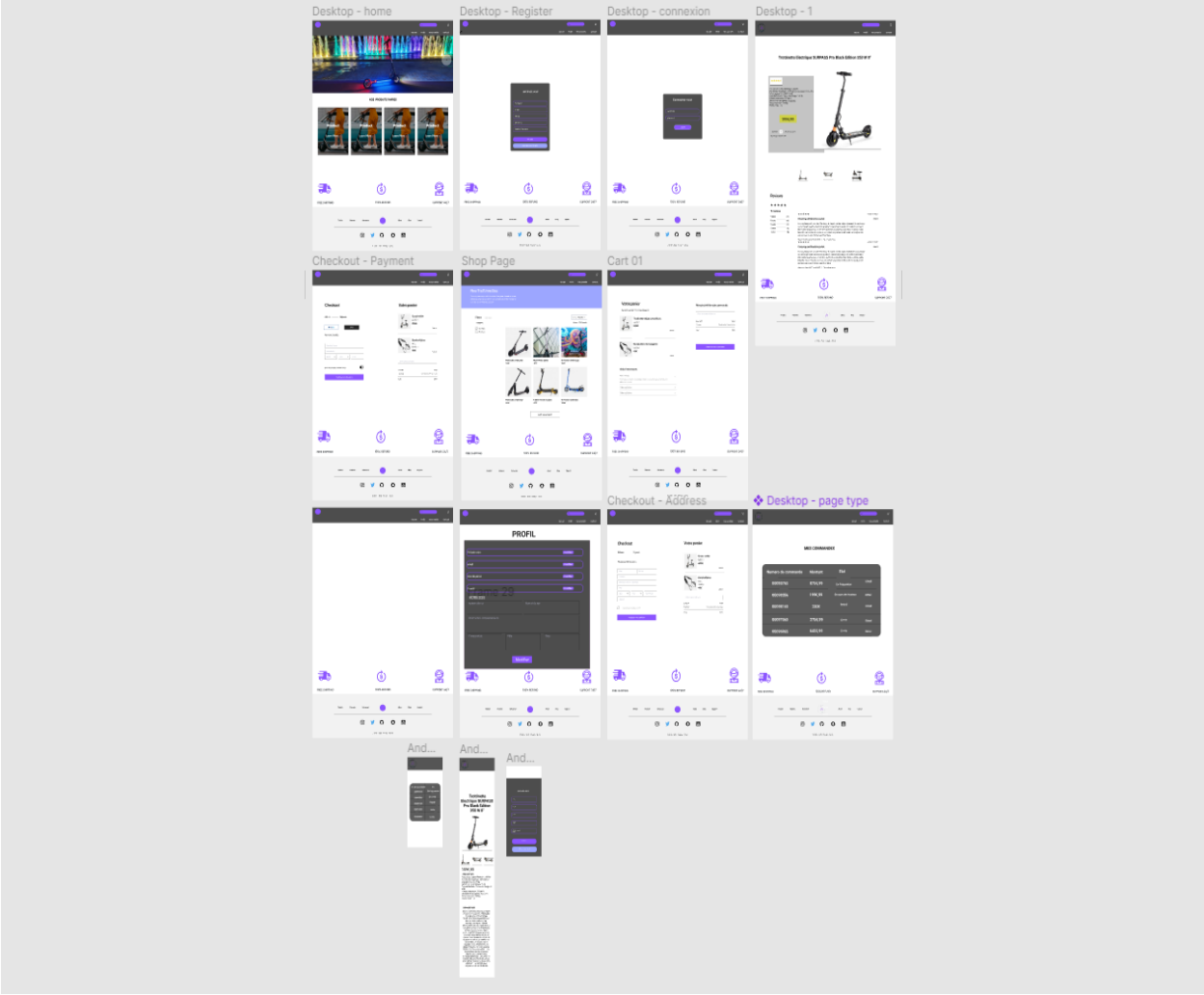
```
$prenom = strip_tags($_POST['prenom']);
$nom = strip_tags($_POST['nom']);
$email = strip_tags($_POST['email']);
$email = filter_var($email, FILTER_VALIDATE_EMAIL);
$civilite = strip_tags($_POST['civilite']);
$passConfirm = strip_tags($_POST['confirmPassword']);
```

## 6. Recherche effectuées à partir d'un site anglophone

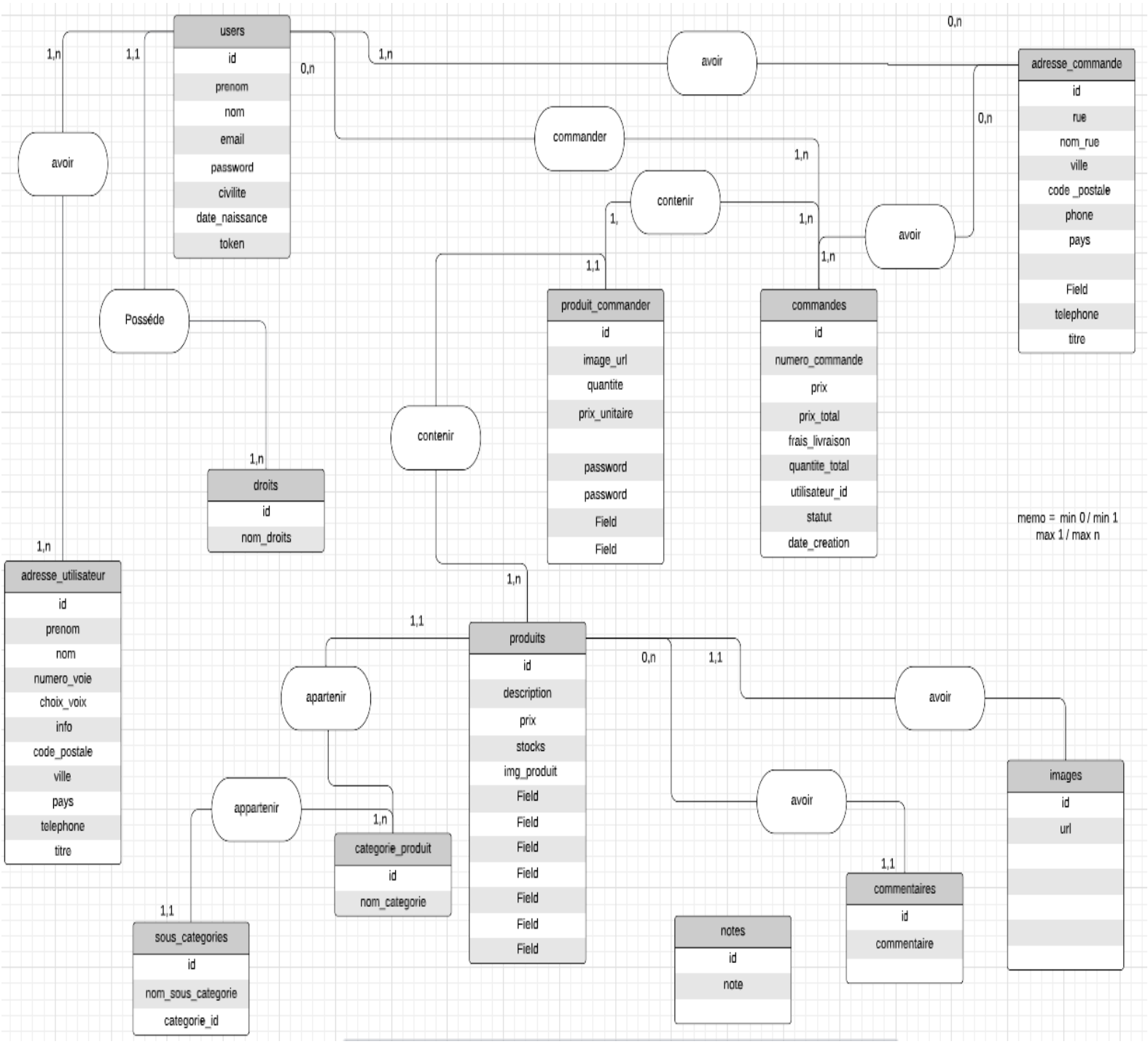
Pendant le développement du projet j'ai eu plusieurs erreurs php et javascript . Certaines erreurs ont été faciles à gérer et d'autres plus compliquées. Pour ces derniers les recherches sur google m'ont souvent amené sur stackoverflow ou je trouvais  assez souvent la solution a mon problème.

# ANNEXES

- Maquette



- Modèle conceptuel des données



- Modèle logique des données