TITRE: Développeur Web et Web Mobile

POREZ CYRIL

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 afin de valider l'unit PHP. J'ai réalisé le projet Boutique en ligne. Avec mes coéquipiers, nous avons choisi de créer une boutique spécialisée dans les vêtements de sport de combat. Pour ce faire nous nous sommes inspirés de marques spécialisées dans ce domaine.

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

Nous nous sommes déjà inspirés de sites existants . Nous avons donc convenu lors de nos réunions d’équipe non seulement quels étaient les besoins de l'entreprise, les fonctionnalités qui seraient développées 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 de l’entreprise Carnage s’adresse à des particuliers et plus exactement des passionnés de sports de combat.

Arborescence du site

L’arborescence du site se décline comme suit :

- Page d’accueil
- Page connexion
- Page inscription
- Page tous les produits
- Page produit
- Page sélection de produit
- Page mon compte
- Page détail commande
- Page panier
- Page envie (wishlist)
- Page paiement
- Page de confirmation de commande

Une partie back-office est également prévue afin de permettre la gestion du site.

1.Authentification

il a été convenu de pouvoir s'inscrire et se connecter de manière classique via un formulaire d'inscription/connexion.

2.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 la taille du vêtement mais également en fonction du prix.

3.Fiche produit

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

- Une ou plusieurs photos du produit
- Le nom du produit
- Le prix
- La description du produit
- La note du produit ou j'aime/j'aime pas
- Les commentaires clients
- Affichage des différentes tailles disponibles

5. Évaluation des produits et commentaires clients.

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.

6.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
- La taille sélectionnée 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 du panier.

7. 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.

8.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 mais également son adresse postale.
Il aura également la capacité de consulter ses précédentes commandes.

9.Back office

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

9.1 Ajout des catégories produit

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

9.2 Gestion des produits

L'administrateur aura également la capacité de créer des produits et de les supprimer.

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

- Donner un titre au produit
- Définir le prix du produit et même un prix promo
- Décrire le produit
- Uploader une ou plusieurs images du produit
- Établir les stocks du produit en fonction des tailles des articles.
- Décider si ce produit doit être mis en avant ou non.

10.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.

Spécifications 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 potentiellement pour le router

Technologies utilisées pour la partie front-end:

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

L'environnement de développement est le suivant:
- Editeur de code: Visual Studio Code
- 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 bundle easy Admin qui permet de mettre en place un panel administrateur simple et facile d'utilisation.

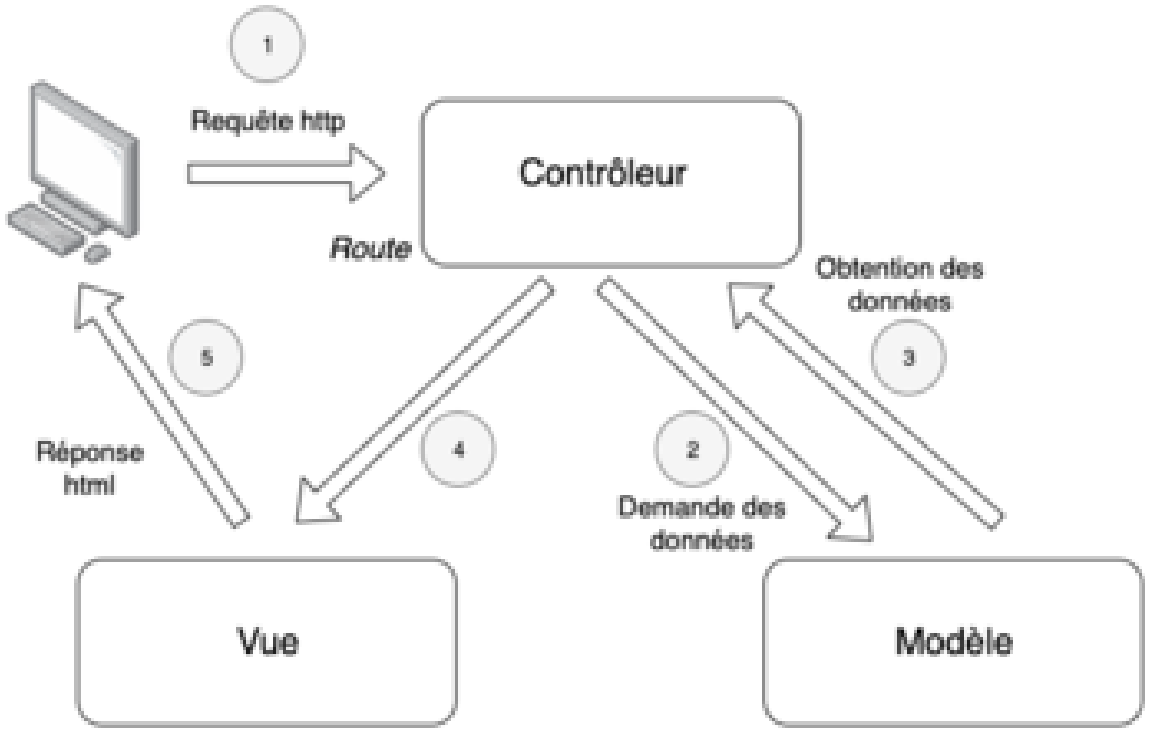
Architecture du projet
Le projet est développé sous PHP, l'utilisation d'un design pattern de type MVC (Model-View-Controller) sera mis en place.
L'architecture MVC est l'une des architectures les plus utilisées pour les applications Web, elle se compose de 3 modules:

- 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

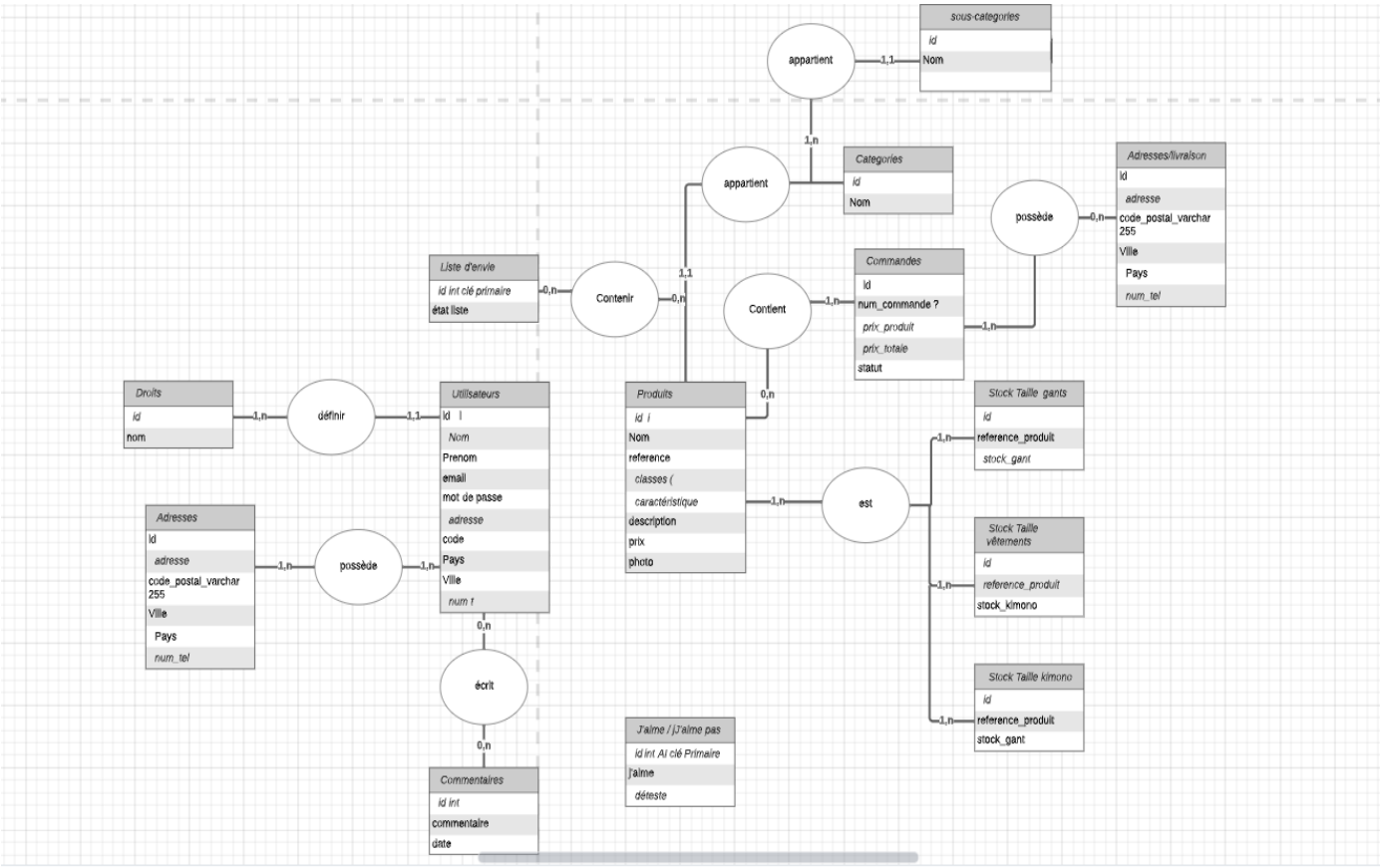
Les polices d'écriture sont les suivantes: Calibri.
La couleur dominante est la suivante : #FFFFFF, #FF0000, #000000

2.Maquette

La maquette a été réalisée avec le logiciel gratuit Figma
Nous avons réalisé le design de la boutique. Nous avons fait des recherches sur internet pour obtenir un design sympas. Nous nous sommes donc inspirée de ces modèles pour réaliser la maquette de la future application. Vous trouverez la maquette dans les annexes du dossier.

3.Conception de la base de données

Au regard des fonctionnalités demandées par l'école, j'ai développé la base de données suivante :



Vous trouverez également dans les annexes le modèle conceptuel 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 utilisateurs qui va permettre d'identifier les clients. Cette table est liées à différentes tables dont la table address, la table favorite 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 stock, catégorie, commentaire, commandes. 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 Commandes 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.

4.Extrait de code significatif

4.1 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 **Paniers** qui étend mon modèle générique **"Model"** lui donnant ainsi accès la connexion à la base de données.

Dans ma classe Paniers j'ai créé 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 placé en paramètre de la méthode représente ici 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.

4.2 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 j'ai utilisé la checkout session

de Stripe. Pour ce faire j'ai créé un fichier paiement.php dans mon dossier controller qui va gérer le paiement du client.

```php
re_once('../vendor/autoload.php');
on_start();



  // Nous appelons l'autoloader pour avoir accès à Stripe
  $prix = (float)$_SESSION['prixTotal'];

  // Nous instancions Stripe en indiquant la clé privée, pour prouver que nous sommes bien à l'origine de cette demande
  \Stripe\Stripe::setApiKey('sk_test_51KMXz2L8eUNbBHo6sgsAeEAIaLa7YN4KePAc6VyIzBD6vYPobi6nvhiIZc2i7IwDQ6mY7st3C9SGpQ8EqT

  // Nous créons l'intention de paiement et stockons la réponse dans la variable $intent
  $intent = \Stripe\PaymentIntent::create([
      'amount' => $prix*100, // Le prix doit être transmis en centimes
      'currency' => 'eur',
  ]);
```

Côté front stripe s'occupe du formulaire de paiment

```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>
```

Côté client, je définis dans un premier temps, la clé publique stripe. Ensuite, on va déclencher au clic sur le bouton checkout une requête qui va rechercher l'id de la checkout session. Le client sera ensuite redirigé vers la page de paiement Stripe où ce dernier va pouvoir renseigner ses informations de paiement.

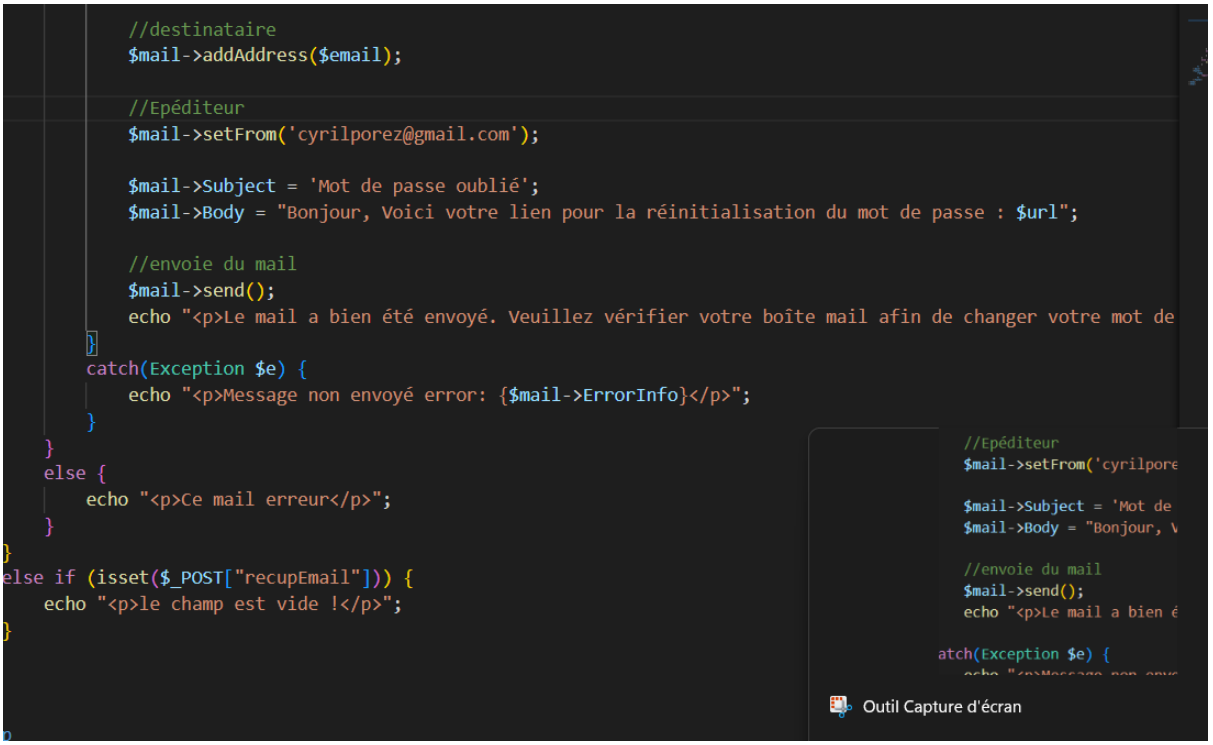
```
window.onload = () => {
    // On instancie Stripe et on lui passe notre clé publique
    let stripe = Stripe('pk_test_51KMXz2L8eUNbBHo6cVGTDBHw40tJQIKldLKtSbz8QGybAfA

    // Initialise les éléments du formulaire
    let elements = stripe.elements();

    // Récupère l'élément qui contiendra le nom du titulaire de la carte
    let cardholderName = document.getElementById('cardholder-name')

    // Récupère l'élement button
    let cardButton = document.getElementById('card-button');

    // Récupère l'attribut data-secret du bouton
    let clientSecret = cardButton.dataset.secret;


    // Crée les éléments de carte et les stocke dans la variable card
    let card = elements.create("card");

    card.mount("#card-elements");

    card.addEventListener('change', function(event) {
        // On récupère l'élément qui permet d'afficher les erreurs de saisie
        let displayError = document.getElementById("card-errors")

        // Si il y a une erreur
        if (event.error) {
            // On l'affiche
            displayError.textContent = event.error.message;
        } else {
            // Sinon on l'efface
            displayError.textContent = "";
        }
    });
```

4.3 PHP Mailer

PHPMailer est une bibliothèque d'envoi d'e-mails en PHP. Je m'en sers pour l'oublis du mot de passe de la part de l'utilisateur. Je l'installe à l'aide de composer. Je créer un token token unique à l'aide de la fonction uniqid(). jJe définis l'url que mon utilisateur recevra afin d'être redirigé vers vers la modification du mot de passe.

```php
use PHPMAILER\PHPMAILER\PHPMAILER;
use PHPMAILER\PHPMAILER\EXCEPTION;
use PHPMailer\PHPMailer\SMTP;

require_once "vendor/phpmailer/phpmailer/src/Exception.php";
require_once "vendor/phpmailer/phpmailer/src/PHPMailer.php";
require_once "vendor/phpmailer/phpmailer/src/SMTP.php";

//require 'PHPMailerAutoload.php';
require 'vendor/autoload.php';
$mail = new PHPMAILER();
```

```php
php
if (!empty($_POST['recupEmail'])) {
    $token = uniqid();
    $url = "http://localhost/boutique-en-ligne/libraries/nouveauMotdepasse?token=$token";
    $email = $_POST['recupEmail'];
    $recupEmail = $bdd->prepare("SELECT email from utilisateurs WHERE email = '$email'");
    $recupEmail->execute();
    $existe = $recupEmail->rowcount();

    if ($existe > 0) {

        try {

            $modifToken = $bdd->prepare("UPDATE utilisateurs SET token = '$token', date = '$date' WHERE email =
            $modifToken->execute();


            //Configuration
            //$mail->SMTPDebug = SMTP::DEBUG_SERVER; // information de debug

            //Configuration SMTP
            $mail->isSMTP();
            $mail->Host = "smtp.gmail.com";
            $mail->SMTPAuth = "true";
            $mail->Username = 'cyrilporez@gmail.com';   //SMTP username
            $mail->Password = 'Hokuto1989@';            //SMTP mot de passe
            $mail->SMTPSecure = 'tls';
            $mail->Port = 587;

            //Charset
            $mail->CharSet = "utf-8";

            //destinataire
            $mail->addAddress($email);
```



5. Veille sur les vulnérabilités de sécurité.

5.1 Exposition des données sensibles

Lorsque vous surfez sur Internet, votre navigateur utilise le protocole HTTP (Hypertext Transfer Protocol) pour afficher les pages web, et le protocole Transmission Control Protocol/Internet Protocol (TCP/IP) pour les transmettre.

Si le serveur web établit la connexion TCP avec le navigateur, une réponse avec le code status et le fichier demandé (généralement le fichier index.html pour la page web) sera transmise. Mais dans notre cas, les données transitent en HTTP et pas en HTTPS…

Les données transitant en HTTP peuvent être interceptées, car elles transitent en clair.

Comment éviter d'exposer les données sensibles en transit ?

- Utilisez le HTTPS pour l'ensemble de votre site, même s'il ne contient pas de données sensibles.
- Utilisez les requêtes GET pour récupérer les informations et POST pour modifier les informations.
- Sécurisez vos cookies pour qu'ils soient transmis par l'en-tête et via HTTPS.
- Sécurisez vos sessions en ajoutant une date d'expiration, en sécurisant l'ID et en ne mettant pas cet ID dans l'URL.

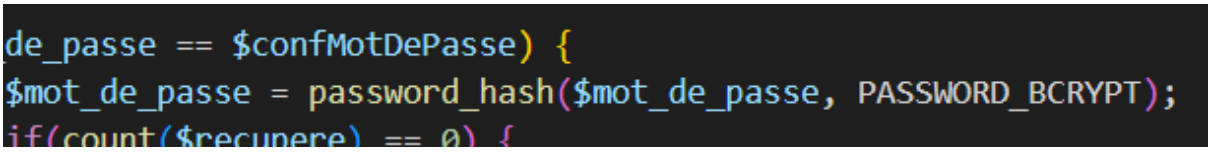
Les données sensibles ne sont pas seulement en transit, elles sont aussi stockées en base de données. Pour protéger certaines données stockées sur une application, il est possible d'utiliser des algorithmes de hachage.

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 cas du projet c'est la fonction password_hash qui s'occupe du hachage:



5.2 Injection SQL

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.

Comment s'en prémunir ?

- **Préparez les requêtes SQL**

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 ce projet les méthodes des différentes class du dossier Models ont des requêtes préparées puis exécutées. exemple ci-dessous.

```php
public function Commentaires($commentaire, $idUtilisateur, $idProduit) {
    $date = date('Y-m-d H:i:s');
    echo "ccccccccc";
    $sql = "INSERT INTO `commentaires`(commentaire, date, id_utilisateur, id_produit) VALUES
    (:commentaire, :date, :id_utilisateur, :id_produit)";
    $commentaires = $this->bdd->prepare($sql);
    $commentaires->bindValue(':commentaire', $commentaire, \PDO::PARAM_STR);
    $commentaires->bindValue(':date', $date, \PDO::PARAM_STR);
    $commentaires->bindValue(':id_utilisateur', $idUtilisateur, \PDO::PARAM_INT);
    $commentaires->bindValue('id_produit', $idProduit, \PDO::PARAM_INT);
    $commentaires->execute();

}
```

5.3 Faille XSS (Cross Site Scripting)

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.

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.
- **trim(),** cette fonction supprime les espaces inutiles

Dans le projet j'ai créer la fonction **protectionDonnees()** pour prévenir les **faille XSS**

Exemple:

```php
function protectionDonnées($données){
    //trim permet de supprimer les espaces inutiles
    $données = trim($données);
    //stripslashes supprimes les antishlashs
    $données = stripslashes($données);
    //htmlspecialchars permet d'échapper certains caractéres spéciaux et les transforment en entité HTML
    $données = htmlspecialchars($données);
    //strip_tags supprimer les balises HTML en autorisant éventuellement certaines d'entre elles
    $données= strip_tags($données);

    return $données;
}
```

5.4 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 passe.

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.

5.5 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.

5.6 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.

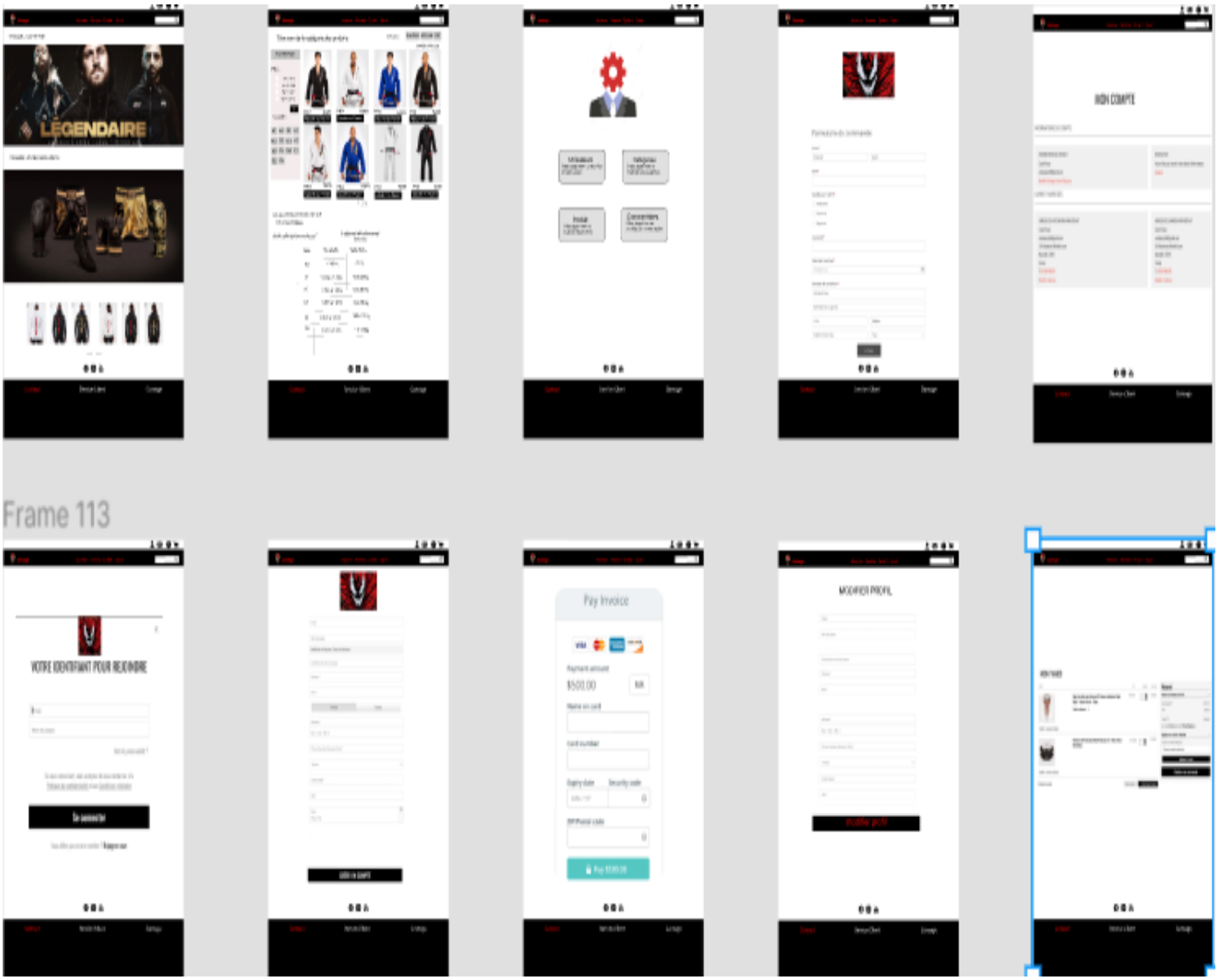
Conclusion

ce projet est l'aboutissement de toute les connaissances que j'ai emmagasiné durant cette année, plein de projet m'ont aidé à en arriver là, comme par exemple faire un module de connexion, de la gestion de projet pour connaître la démarche à adopter et quel procédé suivre, faire du maquettage, du css, du responsive, du JS....

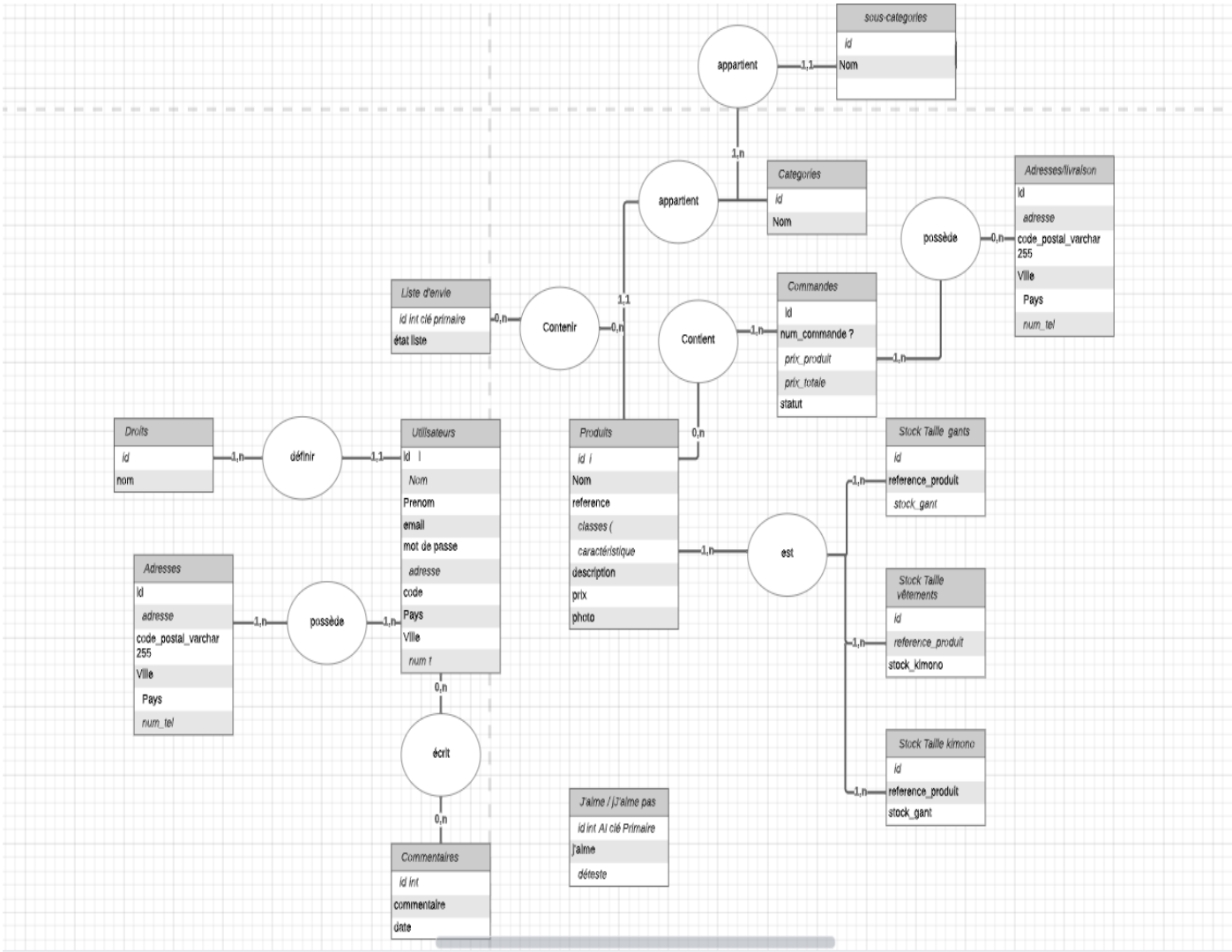
merci pour votre attention.

ANNEXES

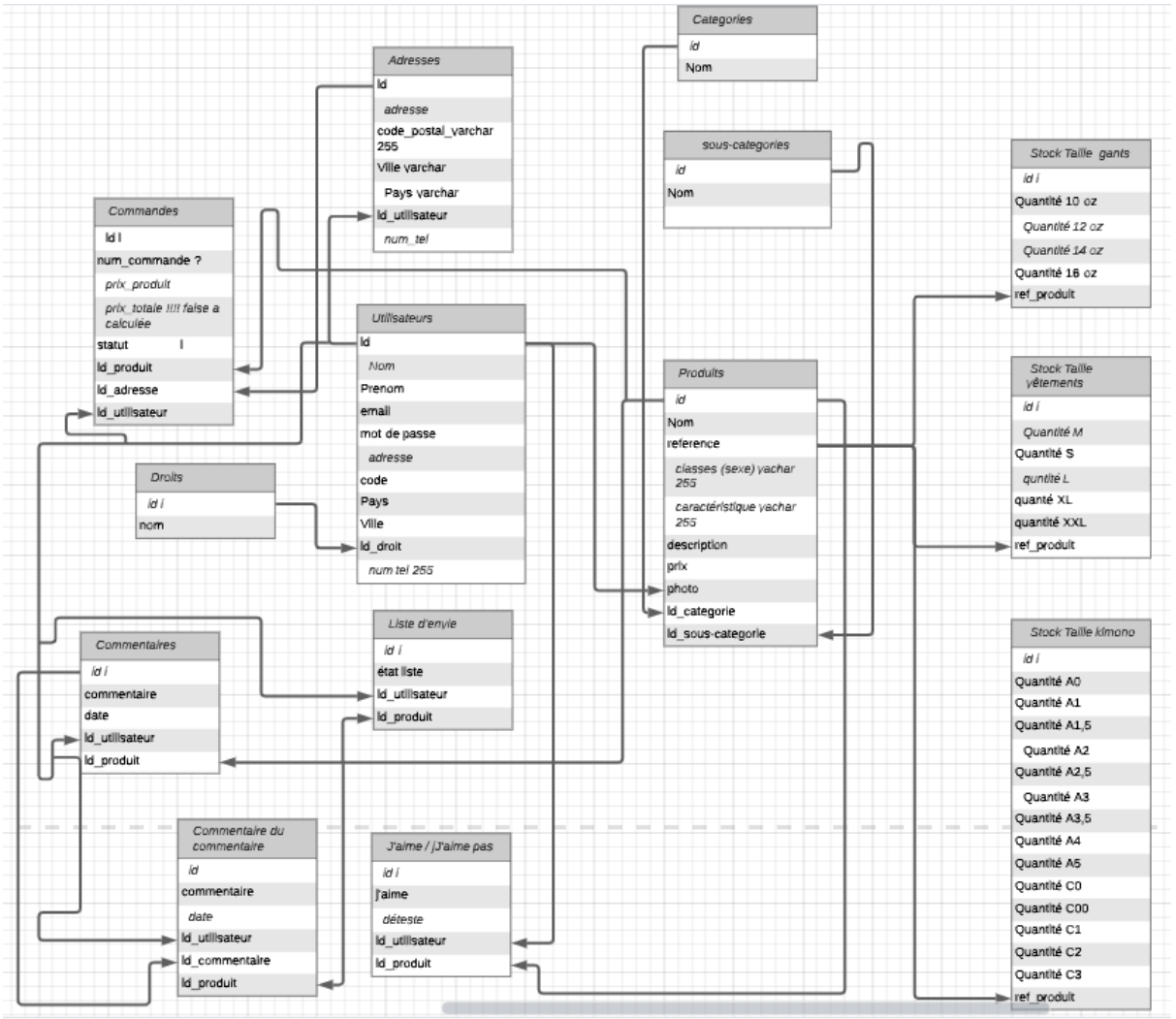
Maquette



MCD


sous-categories
id
Nom
appartient
1,1
1,n
Categories
id
Nom
appartient
Adresses/livraison
Id
adresse
code_postal_varchar 255
Ville
Pays
num_tel
possède
0,n
Liste d'envie
id int clé primaire
état liste
0,n
Contenir
0,n
1,1
Contient
1,n
Commandes
Id
num_commande ?
prix_produit
prix_totale
statut
1,n
Droits
id
nom
1,n
définir
1,1
Utilisateurs
Id I
Nom
Prenom
email
mot de passe
adresse
code
Pays
Ville
num t
Produits
id i
Nom
reference
classes (
caractéristique
description
prix
photo
0,n
1,n
est
Stock Taille gants
id
reference_produit
stock_gant
Stock Taille vêtements
id
reference_produit
stock_kimono
Stock Taille kimono
id
reference_produit
stock_gant
Adresses
Id
adresse
code_postal_varchar 255
Ville
Pays
num_tel
1,n
possède
1,n
0,n
écrit
0,n
Commentaires
id int
commentaire
date
J'aime / J'aime pas
id int AI clé Primaire
j'aime
déteste

MLD


Categories
id
Nom
Adresses
Id
adresse
code_postal_varchar 255
Ville varchar
Pays varchar
Id_utilisateur
num_tel
sous-categories
id
Nom
Stock Taille gants
id i
Quantité 10 oz
Quantité 12 oz
Quantité 14 oz
Quantité 16 oz
ref_produit
Commandes
Id I
num_commande ?
prix_produit
prix_totale !!!! false a calculée
statut I
Id_produit
Id_adresse
Id_utilisateur
Utilisateurs
Id
Nom
Prenom
email
mot de passe
adresse
code
Pays
Ville
Id_droit
num tel 255
Produits
id
Nom
reference
classes (sexe) vachar 255
caractéristique vachar 255
description
prix
photo
Id_categorie
Id_sous-categorie
Stock Taille vêtements
id i
Quantité M
Quantité S
quntité L
quanté XL
quantité XXL
ref_produit
Droits
id i
nom
Liste d'envie
id i
état liste
Id_utilisateur
Id_produit
Commentaires
id i
commentaire
date
Id_utilisateur
Id_produit
Stock Taille kimono
id i
Quantité A0
Quantité A1
Quantité A1,5
Quantité A2
Quantité A2,5
Quantité A3
Quantité A3,5
Quantité A4
Quantité A5
Quantité C0
Quantité C00
Quantité C1
Quantité C2
Quantité C3
ref_produit
Commentaire du commentaire
id
commentaire
date
Id_utilisateur
Id_commentaire
Id_produit
J'aime / J'aime pas
id i
j'aime
déteste
Id_utilisateur
Id_produit