Titre : Développeur Web/Web Mobile

DOSSIER DE PROJET

Conception et réalisation d'un site e-commerce

SONDER Frédérick

TABLE DES MATIERES

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 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 e-commerce.

Résumé :

Freya Lab est une auto-entreprise créée en 2019 et basée à **Istres**, qui vend des articles pour **professionnel de l'onglerie**. Sa particularité est que la dirigeante, Mme **Carolyne SONDER**, est aussi prothésiste ongulaire, Elle utilise, met en valeurs et promeut ses articles. . N'ayant pas de site pour la vente de sa marque sur internet, la gérante m'a donc sollicité dans le but de réaliser une boutique en ligne.
En effet, son souhait étant que ses clientes puissent commander directement sur le site internet.
L'enjeu étant pour l'entreprise de **satisfaire** des besoins, d'**accroître** ses parts de marché ainsi que sa **visibilité** sur internet.

Spécifications fonctionnelles

Description de l'existant

Aucune boutique n'existait auparavant.
La cliente possédait déjà un salon d'onglerie « Freya le Salon », Freya Lab. en est la continuité.
La charte graphique a donc été reprise.
Nous avons déterminé lors de nos différents entretiens 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, : mobile, tablette et ordinateur.
Cible adressée par le site internet
Le site de l'entreprise « Freya Lab. » s'adresse à des professionnels de l'onglerie.

Arborescence du site

L'arborescence du site se décline comme suit :

- Page d'accueil
- Page connexion
- Page inscription
- Page Nos produits
- Page détail produit
- Page mon profil
- Page mes commandes
- Page détail commande
- Page panier
- Page choix de la livraison
- Page paiement
- Page de confirmation de commande

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

Description des fonctionnalités

1. Authentification classique

L'entreprise Freya Lab. Souhaite que ses clients puissent 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 et le nom du produit ainsi que son prix.
Un filtre doit être implémenté afin de trier les articles en fonction de leur catégorie.

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
- Les commentaires clients
- Affichage des différents contenants disponibles
- Un mode d'utilisation du produit

4. 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.

5. 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 un article du panier, voire son intégralité.

6. 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.

7. Espace client

L'utilisateur aura accès à un espace client dans lequel il lui sera possible de consulter ses informations.
Il pourra aussi modifier son nom, prénom, adresse email, date de naissance, mot de passe mais également ajouter ou modifier une adresse de livraison.
Il aura également la capacité de consulter ses précédentes commandes, de voir leur statut (annulée, en cours de traitement, expédiée).

8. Back office

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

9. 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.

9.1 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
- Décider si ce produit doit être mis en avant ou non.

9.2 Système de promotion

L'administrateur devra être en mesure de mettre en place un système de promotion de type code promo.
Ce code devra être renseigné par le client juste avant la validation de son panier.
Pour informer le client de la promotion en cours, un bandeau indiquant la promotion devra apparaître sur la page d'accueil du site.
A noter que l'administrateur sera en mesure de décider si ce bandeau est visible ou pas.

9.3 Suivi des commandes

L'administrateur 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.

10. Intégrer un prestataire de livraison

L'entreprise n'a pas de prestataire pour expédier ses commandes.
Le site devra donc intégrer un ou plusieurs plugin de prestataire afin de permettre aux clients de choisir le relais le plus proche de chez eux.

11.Solution de paiement

La solution de paiement choisie et qui devra être mise en place est celle proposée par Stripe. Cette solution permettra au client de procéder au paiement de manière sécurisée par carte bancaire.

12. 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 le point relais qui a été choisi.

Spécifications techniques

Technologies utilisées pour la partie back-end :

- Le projet sera réalisé avec le langage **PHP**
- Base de données **SQL**
- Gestionnaire de dépendance: **Composer**

Technologies utilisées pour la partie front-end:

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

L'environnement de développement est le suivant :

- Éditeur 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, pour la partie Admin, un système de grid qui permet de mettre en place un panel simple et facile d'utilisation.

Architecture du projet

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

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.

Réalisations

1. Charte graphique

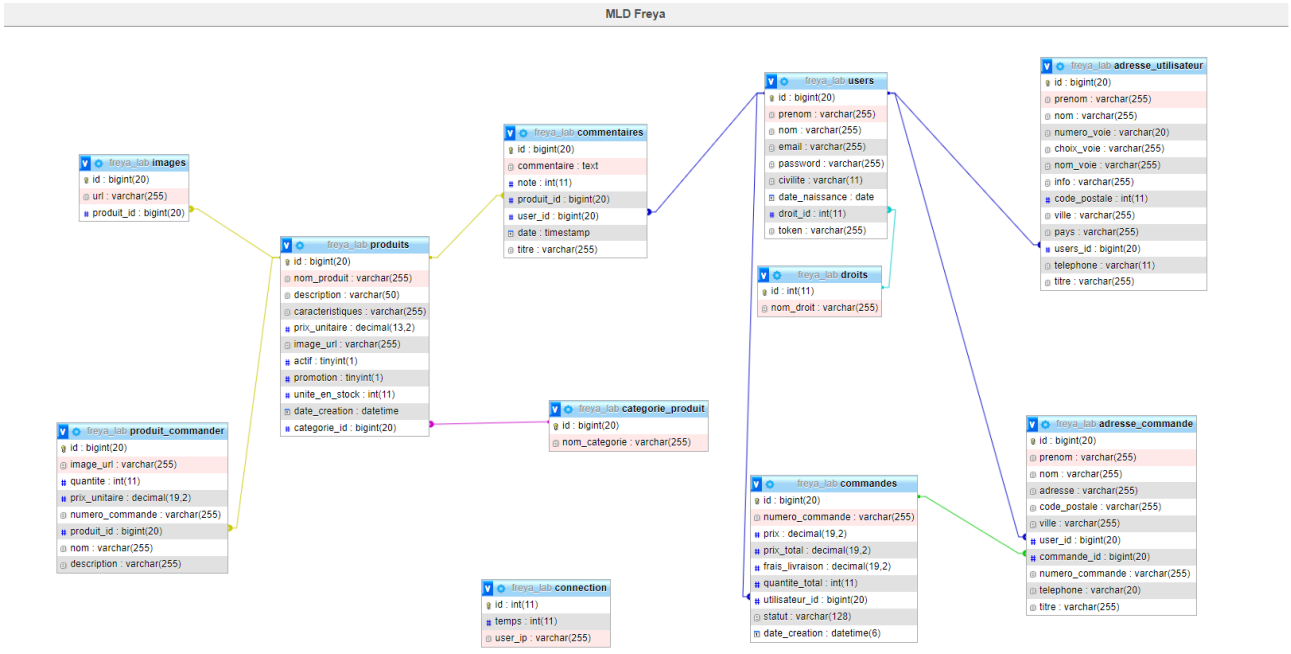
Les polices d'écriture sont les suivantes: Cinzel et Tan mon cheri.
La couleur dominante est la suivante : #5D13A8
Vous trouverez la charte complète dans les annexes du dossier.

2. Maquette :

La maquette a été réalisée avec le logiciel gratuit **Figma**
L'entreprise n'ayant pas de Web designer, j'ai réalisé moi même le design de la boutique.
La gérante m'a envoyé des captures d'écran de site dont le design lui plaisait.
Je me suis donc inspirée de ces captures 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'entreprise, j'ai développé la base de données suivante :



Comme illustré ci dessus, on peut voir que la base de données s'articule autour de **2** 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 **addresse_utilisateur**, la table **addresse_commande** et également la table **commandes** qui va me permettre de par exemple de relier une commande au client concerné.
Il y a ensuite la table **produits** qui va être reliée à la table **categorie**, **commentaires**, **produit_commander**.
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, de savoir de quels produits se compose cette commande.

4. Extraits de code significatifs

Méthode affichant une page listant tout les produits

```
public function index($page)
{
      // On instancie le model correspondant a la table produit
      $produitsModel = newProduitsModel;

      //On va chercher tous les produits
      $produits = $produitsModel->findAll();

      //  On recupère les catégories pour le fitre
      $categoriesModel = new CategoriesModel;
      $categories = $categoriesModel->findAll();

      // On génère la vue
      $this->render('produits/index', compact('produits','categories'));
}
```

Affichage de la vue

```
<section>
      <?php foreach ($categories as $categorie) :?>
            <ul class="list-group">
            <li class="list-group-item"><h2>
               <a href="/FreyaLab/produits/categorie_produit/<?=$categorie->id ?>">
            <div>
            <img src="/FreyaLab/Views/images/<?=$categorie->img_cat?>"alt="image<?=
$categorie->nom_categorie ?>">
            </div>
                                    <article>
                                          <p class="article_categorie_home">
                                                <b>
                                                <?= $categorie ->nom_categorie ?>
                                                </b>
                                          </p>
                                    </article>
                              </a>
                        </h2>
                  </li>
            </ul>
      <?php endforeach; ?>
</section>
```

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 **Argon2I**, **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 j'utilise **Argon2I**, qui s'occupe du hachage des mots de passes:

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

Cet extrait de code de signifie que le mot de passe inséré par l'utilisateur va être chiffrer par l'algorithme de hachage Argon2I pour le stocker en sécurité dans la base de donnée.

```
// On verifie si le mot de passe est correct
if (password_verify($_POST['password'], $user->getPassword())) {
// Le mot de passe est bon // On crée la session
$user->setSession();}
```

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 i**nterprétées par le moteur SQL**, ce qui lui permettra de modifier le comportement de votre application.

Comment s'en prémunir ?

- Validez 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. Par exemple, la contrainte `filter_var($email, FILTER_VALIDATE_EMAIL)` s'assure que la valeur est bien une adresse mail valide.

```
// On verifie si le formulaire est valide
if (Form::validate($_POST, ['prenom', 'nom', 'email', 'civilite', 'password',
'confirmPassword']))
{// Le formulaire est valide$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']);
```

Cet extrait de code de signifie que le formulaire rempli par l'utilisateur va être sécurisé par **strip_tags** qui supprime les balises **HTML** et **PHP** ainsi que **filter_validate_email** qui vérifiera si le format de l'e-mail est correct **selon la syntaxe de la RFC 822**.

- Préparez les requêtes SQL

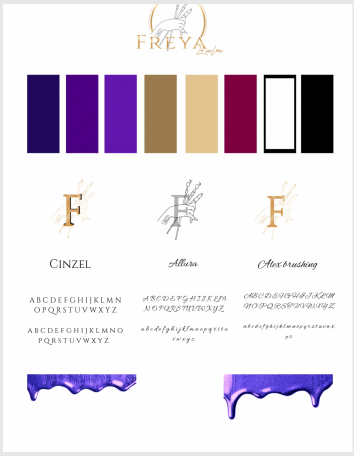
  Ce sont des requêtes dans lesquelles les paramètres sont interprétés indépendamment de la requête elle-même. De cette manière, il est impossible d'effectuer des injections.

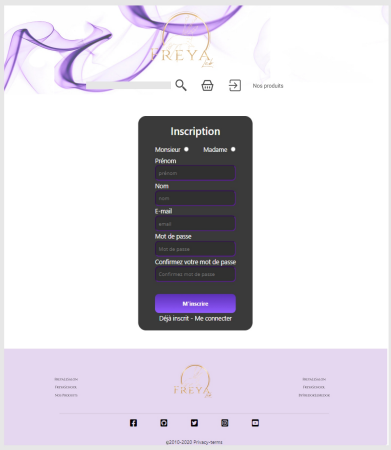
```
/**  //Methode pour preparer et exécuter une requete
public function requete(string $sql, array $attributs = null)
{
//On recupère l'instance de Db
$this->db = Db::getInstance();
//On verifie si on a des attributs
if ($attributs !== null) {
//Requete préparée
$query = $this->db->prepare($sql);
```

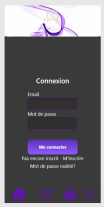
```
$query->execute($attributs);
return $query;
} else {
//Requete simple
return $this->db->query($sql);
}}
```

Chaque requête personnalisées réalisée dans ce projet, sera préparée et interprétée par cette méthode lorsque j'utilise ses requêtes, Cette méthode va me permettre d'interpréter les paramètres insérés sans mettre les valeurs en dur.

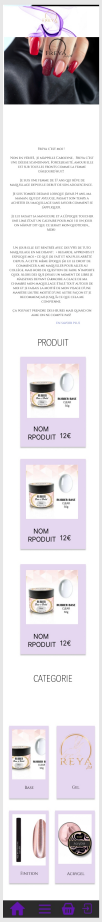
Annexes

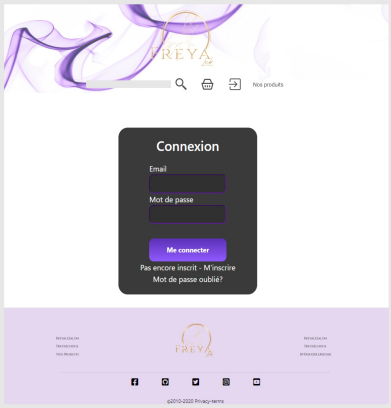

FREYA
CINZEL
Alliura
Alex brushing

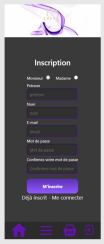

Nos produits
Inscription
Monsieur
Madame
Prénom
Nom
E-mail
Mot de passe
Confirmez votre mot de passe
M'inscrire
Déjà inscrit - Me connecter

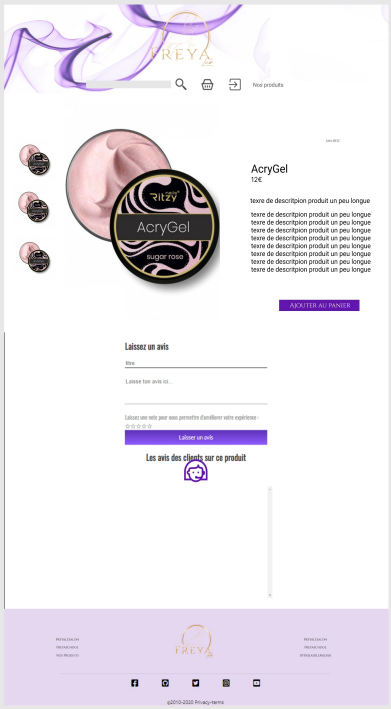

Connexion
Me connecter


Nos produits
QUI EST FREYA
MEILLEURES VENTES
CATEGORIE
RUBBER BASE CLEAR
Base
Gel
Finition
Acrygel


PRODUIT
NOM RPODUIT 12€
CATEGORIE


Nos produits
Connexion
Email
Mot de passe
Me connecter
Pas encore inscrit - M'inscrire
Mot de passe oublié?


Inscription
Déjà inscrit - Me connecter


Nos produits
AcryGel
12€
AJOUTER AU PANIER
Laissez un avis
Les avis des clients sur ce produit


AcryGel
12€
AJOUTER AU PANIER
Laissez un avis
Les avis des clients sur ce produit

Schéma illustrant le design pattern MVC