Dossier de projet titre développeur web et web mobile.

Site web : L’Ornithorynque Tattoo

Création d’un développement d’un site vitrine ainsi que d’une interface admin.

Titre présenté : Développeur web et web mobile

par Fabien Ponzio

En formation continue à la plateforme, 8 rue d’Hozier 13002 Marseille.

Sommaire :

Introduction 04
1.Présentation
1.1 Présentation personnelle 04
1.2 Présentation La Plateforme_ Coding School. 04
1.3 Présentation de la formation 04

2.Présentation du projet 05
2.1 Présentation de l'entreprise 05
2.2 Liste des compétence couvertes par le projet 06
2.3 Résumé du projet 06
2.4 Cahier des charges du projet 07
2.4.1 Les objectifs du site 07
2.4.2 Les fonctionnalités du site 07
2.4.3 Les cibles 08
2.4.4 Le périmètre du projet 08

3. Partie Front-End 09

3.1 Maquette 09
3.1.1 Arborescence 10
3.1.2 Charte Graphique 10
3.1.3 Prototype 11
3.1.4 Outils 12
3.1.5 Difficultés rencontrés et axes d'améliorations 12

3.2 Développer une l'interface utilisateur web dynamique 12
3.2.1 Intégration et contenu dynamique 12
3.2.2 Responsive design 14
3.3 Conclusion 14

4. Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité 15
4.1 Organisation 15
4.2 Création de la base de données 16
4.3 Composants d'accès aux données 17
4.4 Développement de la partie BackEnd d'une application mobile ou web mobile 18

4.4.1 Fonctionnalité significative : la page admin 18
4.4.2 Extrait de code 18
4.5 Veille et sécurité 23
4.5.1 Référencement des principales failles de sécurité existantes 23
4.5.2 Pratiques suivies pour sécuriser le site 24
4.5.3 Conclusion 27

4.6 Recherche Anglophone 28
4.6.1 Description d'une situation nécessitant une recherche sur un site anglophone 28
4.6.2 Extrait du site anglophone 28
4.6.3 Traduction en français du site 29

5. Conclusion 30
6.Annexes

Introduction

J’ai, en septembre 2020, intégré une formation dans le but de préparer mon passage au titre professionnel de développeur web et web mobile à LaPlateforme_ Marseille.

Dans le but de présenter le titre professionnel, de proposer un support de lecture complet et d’acquérir les compétences nécessaires à son obtention, j’ai travaillé seul au développement d’un site vitrine pour un tatoueur Marseillais.

Le projet que je vais vous présenter dans ce dossier est venu me confronter au monde professionnel dans la mesure où l’objectif du projet était de réaliser un site web dans son entièreté, de l’élaboration du cahier des charges, en passant par le maquettage, au code, jusqu'à sa mise en ligne…

1. Présentation

1.1 Présentation personnelle

Je m’appelle Fabien PONZIO et je suis en reconversion d’études. En effet après avoir étudié dans l’univers du commerce j’ai décidé d’intégrer l’univers du web qui se trouve être naturellement complémentaire. Mon orientation vers ce milieu s’est faite grâce à ma curiosité et l’envie de disposer d’une autonomie en termes d’informatique et de développement web.

1.2 Présentation La Plateforme_ Coding School

La Coding School est une formation web qui s’adresse à tous ceux qui souhaitent s’ouvrir les portes des métiers du numérique. Le modèle pédagogique unique de l’École la Plateforme, membre du label GEN, Grande École du Numérique, s’adapte aux besoins de chacun. Les évaluations se font par des contrôles continus sur des projets webs concrets réalisés seuls ou en groupes. L’intégration professionnelle est réalisée au travers de projets professionnels tutorés, portés par des entreprises. La coding school revendique une pédagogie active et inductive centrée sur l’apprenant et orientée projet.

1.3 Présentation de la formation

La formation (1200 h/an) s'effectue en présentiel, dans un lieu dédié spécifiquement pour catalyser l'apprentissage au 8 rue d'Hozier 13002 Marseille.
Le programme vise à dispenser les connaissances et compétences nécessaires pour acquérir l'obtention du titre professionnel : technologies du web, maquettage d'applications, modélisation de bases de données, développement de sites web statiques, dynamiques et responsives, déploiement de CMS, base d'algorithmie, projet professionnel.
Cette année de formation est divisée en 4 Units destinées à apprendre de nouvelles technologies : Unit 1 / HTML-CSS, Unit 2 / PHP MySql, Unit 3 /Javascript, Unit 4 / Projet Pro.

J'ai ainsi appris à coder avec les technologies suivantes : **HTML / CSS, PHP, MySql, Javascript, Jquery, Ajax, Symfony**… J'ai travaillé en groupe sur des projets de sites vitrines, des interfaces d'inscription et connexion, des e-commerce, un réseau social etc.

2. Présentation du projet

2.1 Présentation de l'entreprise

Nom de l'entreprise : L'ornithorynque tattoo
Adresse : 18 rue Jean-François Leca 13002 Marseille
Consultant : Julien Testard, propriétaire du salon de Tatouage.

L'ornithorynque tattoo est un salon de tatouage établi sur Marseille, il y avait auparavant deux salons "L'ornithorynque". Le premier et toujours actif est basé à la Joliette, le deuxième qui a fermé se trouvait sur la corniche. Le salon était au début composé de deux tatoueurs, il est maintenant composé de 5 artistes tatoueurs, ce qui illustre l'agrandissement de la structure durant ces dernières années, le propriétaire de la boutique manifeste dorénavant son désir de disposer d'un **site web**.

"L'ornithorynque tattoo" sera la version bêta d'un **site vitrine** qui permettra à chaque tatoueur d'exposer ses réalisations ainsi que ses "flashs" disponibles. L'ornithorynque tattoo a pour objectifs à long terme de disposer d'une partie e-commerce lui permettant de vendre du merchandising. Le site en version bêta disposera d'une partie administrateur uniquement accessible par les tatoueurs qui permettra :

- d'ajouter du contenu sur différentes pages

- d‘ajouter et supprimer des administrateurs
- ajouter des rendez-vous via un formulaire
- visualiser les rendez-vous de chaque tatoueur

2.2 Liste des compétences couvertes par le projet

Vous trouverez ci-dessous les compétences développées nécessaires à la validation du titre de développeur web et web mobile dont voici la liste :

Activité type 1 : développer la partie front-end d’une application web ou web mobile :

- maquetter une application
- réaliser une interface utilisateur web statique et adaptable

Activité type 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
- Elaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce
- spécifications techniques du projet
- organisation
- présentation d’une fonctionnalité représentative
- veille et sécurité
- recherche à partir d’un site anglophone

2.3 Résumé du projet :

Le projet est né dès que le salon de tatouage m’a fait part de ses problèmes pour centraliser les prises de contacts clients, celles-ci se font par multiples canaux : Facebook, Instagram, Messenger, Mail.

Il me semblait évident de proposer un site capable de **centraliser les demandes** à un seul et même endroit. Le site a pour objectif de faire augmenter le chiffre d'affaires du salon et le nombre de clients potentiels. Le site aura aussi pour objectif de limiter la prise de contact

ailleurs que sur le site internet, ce qui donnera une image bien plus professionnelle à l'ornithorynque tattoo.

Via un **panel admin**, chaque tatoueur pourra ajouter ses réalisations favorites ainsi que ses "flashs", on appelle "flashs" un dessin imprimé ou dessiné sur papier ou carton, et peut être considéré comme une espèce de dessin industriel.

Il est généralement affiché sur les murs des salons de tatouage et dans des classeurs pour donner aux clients sans rendez-vous des idées de tatouages. Le tatoueur pourra ajouter et supprimer des réalisations et pourra en faire de même avec des flashs tout en améliorant le référencement du site.

Une interaction utilisateurs et tatoueur aura lieu via un **formulaire de contact** qui sera rempli par l'utilisateur à l'attention du tatoueur, la requête sera ensuite prise en charge par le tatoueur le plus apte à y répondre (mail).

Le site possède un back-office, un espace administrateur qui permet de gérer l'ajout et la suppression de contenu, d'administrateur et de rendez-vous. Ainsi j'ai tenté de répondre aux attentes du salon de tatouage et de développer les fonctionnalités à son optimisation et sa **sécurité**. Le **back-office** doit être un tableau de bord facile d'accès et à manipuler pour les différents membres du site.

2.4 Cahier des charges du projet

Avant de passer à quelques démarches techniques il est important d'établir un cahier des charges nécessaire au bon déroulement d'un projet d'une telle envergure.

2.4.1 Les objectifs du site

Le site a pour objectif de faire augmenter le chiffre d'affaires du salon et le nombre de clients potentiels. Le site aura aussi pour objectif de limiter la prise de contact sur des plateformes autres que le site internet. Il sera intéressant et nécessaire d'observer le revenu engendré par les rdv effectués sur le site internet.

2.4.2 Les fonctionnalités du site

Le site doit comporter une partie vitrine et une partie **administrateur**.

La partie vitrine contiendra les différentes pages de tatoueurs qui contiendront chacune les réalisations et flashs de chacun.

La partie **administrateur**, elle, contiendra :

- **Formulaire de contact** (contact.php) sécuriser avec des **multiples inputs** qui vont préciser la demande : motif de la prise de contact (renseignement, partenariat, retouche tatouage, prise de rdv) style de tattoo, préférence tatoueur (tchang, poupou, nachos, serge, fanny, je ne sais pas), budget (50-100€, 150-300€, + etc..) et âge avec un **textarea** qui permettra au client de préciser au mieux à sa requête.
  Une fois la demande reçue du côté du tatoueur, il se réserve le droit de continuer de prendre contact avec le futur client.

- Interface d'**ajout de rendez-vous** pour des séances de tatouage via un formulaire.

- **Planning** par semaine avec créneaux disponibles ainsi que les tatoueurs avec visibilité des tatoueurs et créneaux disponibles.

2.4.3 Les cibles

Les **utilisateurs** du site web seront :

- Les personnes majeures qui ont connu la boutique via les réseaux sociaux ou le bouche à oreille.
- Les personnes majeures vivant à Marseille à la recherche d'un tatoueur
- Les tatoueurs du Salon "l'ornithorynque tattoo".
- Les personnes susceptibles de vouloir effectuer des partenariats avec le salon de tatouage.

L'application doit être facile à utiliser aussi bien pour des personnes qui peuvent parfois être peu familières avec les outils internet.

2.4.4 Le périmètre du projet

L'application est développée en français. Elle utilise une base de données relationnelle créée par mes soins. **La partie front-end est développée en HTML/CSS**. Le site est accessible à tous les supports (mobiles, tablettes) grâce au responsive. **La partie back-end est développée en PHP**.

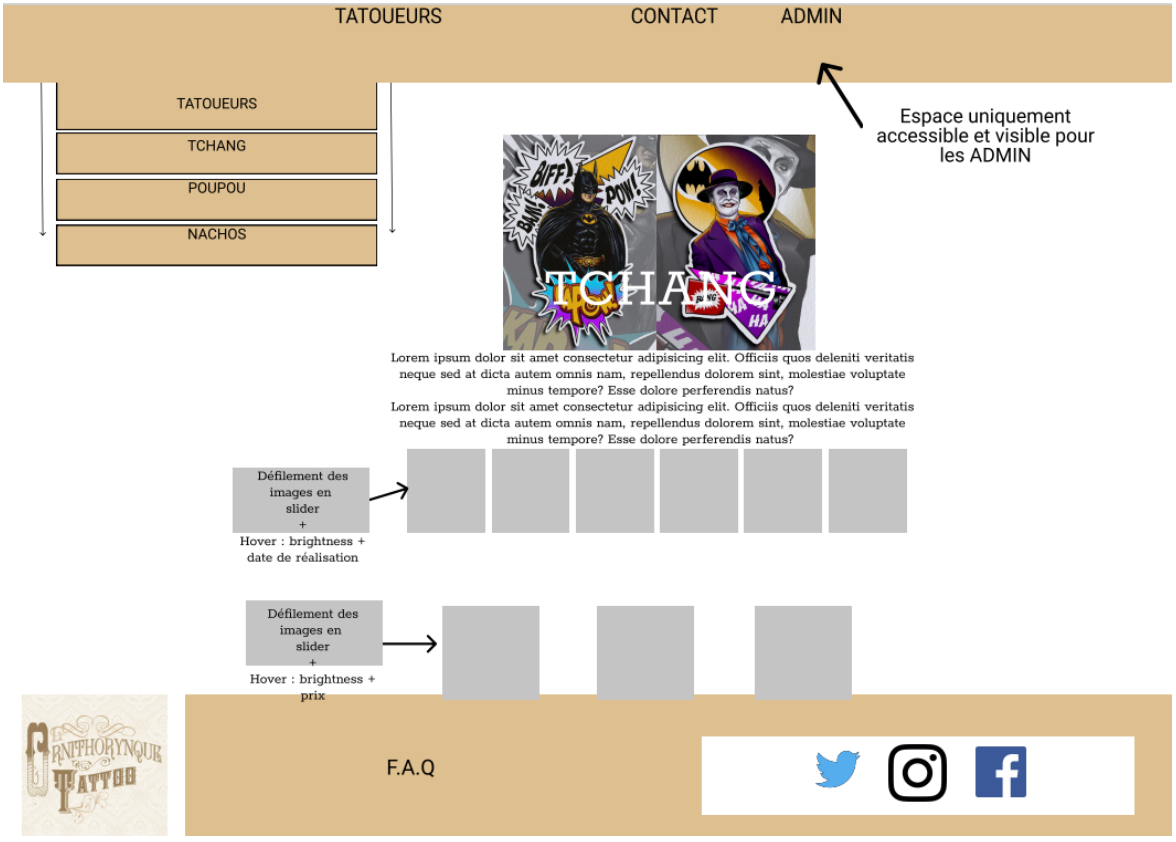
3. Développement de la partie FrontEnd de l'application

Avant de réaliser **la charte graphique** et **la maquette** du site, j'ai effectué la veille de sites web de boutique de tatouages de la ville de Marseille. J'ai effectué une sélection de sites internet au design intéressant.

Références :
https://www.tattoobysote.com/
http://loveguntattoo.com/
http://sailinontattoos.com/
http://www.popink-tattoo.com/
https://www.blacklabtattoo.fr/

3.1 Maquette

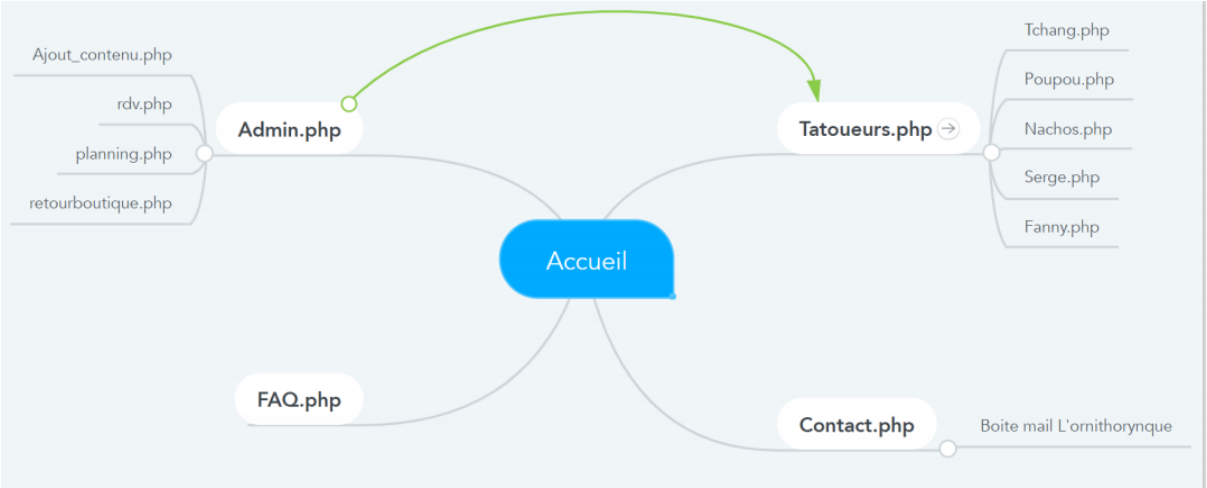
Avant de décider de la charte graphique pour mon projet il était important d'arriver avec un premier prototype de maquette pour le client que voici :



Veuillez noter que **ce code couleur** fut l'une de mes premières propositions au client, suite à plusieurs discussions avec celui-ci et l'utilisation de sites de palettes de couleurs tels que https://colorhunt.co/ , nous avons réussi à dégager une charte graphique qui mêle couleurs actuelles et baroques.

3.1.1 Arborescence

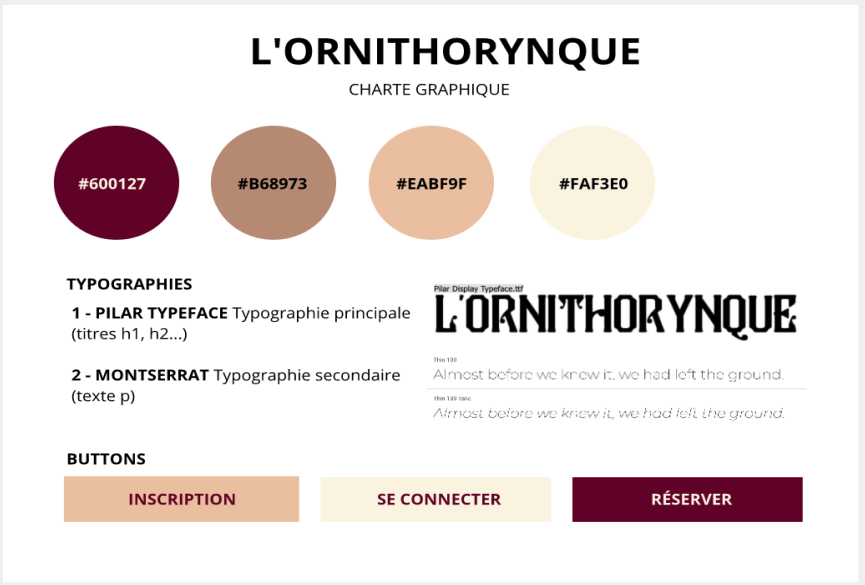
Avant de réaliser le prototypage, j'ai réalisé une arborescence afin de déterminer **le parcours utilisateur** et estimer le **nombre de pages** que le site allait comporter. Cette architecture permet de suivre comment l'utilisateur va **naviguer** d'une page à une autre.



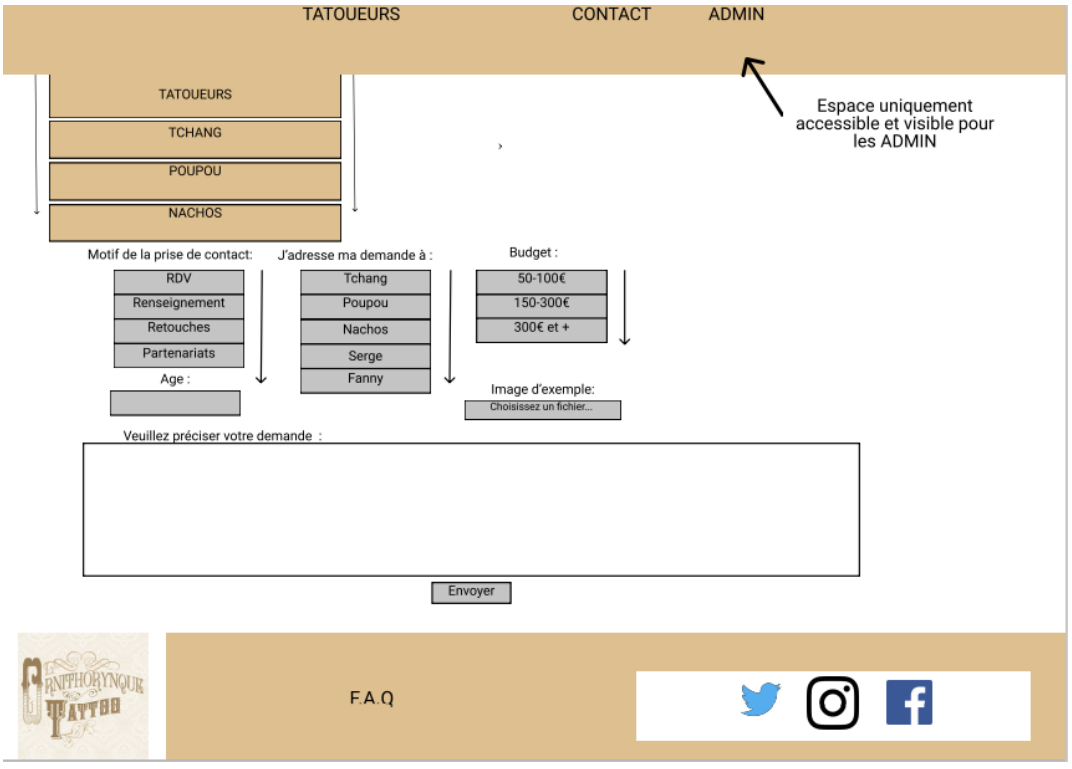
3.1.2 Charte Graphique

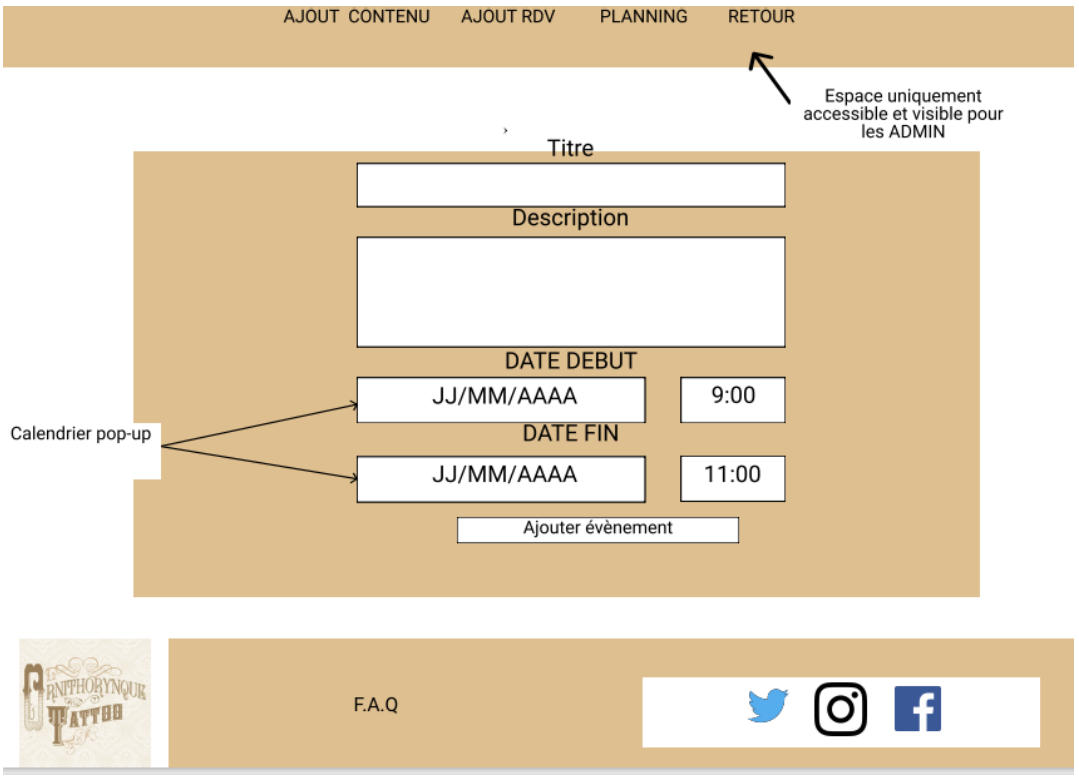
J'ai par la suite élaboré une charte graphique. Celle-ci reprend :

- un logo que j'ai créé pour le site
- une liste des typographies à importer sur le site
- les couleurs principales de l'application
- des formes et couleurs de boutons



3.1.3 Prototype :





3.1.4 Outils

Les outils suivants ont été utilisés pour les différentes étapes du maquettage :

- **Photoshop** pour **la charte graphique**
- **Colorhunt.co** pour la charte graphique et la **sélection des couleurs**
- **Dafont** et **Google fonts** pour la **sélection des typographies**
  https://fonts.google.com/ https://www.dafont.com/fr/
- Chrome et son inspecteur

3.1.5 : Difficultés rencontrées

La difficulté principale lors de la réalisation de la maquette a été de définir un parti pris esthétique et graphique correspondant à l'esprit d'un salon de tatouage. Le design se doit d'être immersif, chaleureux et baroque tout en mettant en valeur les éléments présentés.

3.2 Développer une interface utilisateur web dynamique :

J 'ai utilisé le **HTML/CSS** comme structure des pages à développer et pour l'intégration. Le style des pages est réalisé en **CSS**. Il est utilisé pour adapter les pages au multi-supports,

au responsive en utilisant les **media queries**. Le contenu des pages est généré ou s'adapte à l'utilisateur grâce au langage PHP.

3.2.1 Intégration et contenus dynamiques

Index : c'est une page web **statique et adaptable**. Cette page a été pensée comme une introduction au site. Elle y présentera **une vidéo du salon** de tatouage qui a pour objectif **"l'immersion"** de l'utilisateur dans l'ambiance du salon. Des **liens href** vers les principaux réseaux sociaux sont disposés.

Header : Il est divisé en deux blocs. Une partie en en-tête qui reprend **le logo** de "L'ornithorynque tattoo shop" sur la gauche et sur la droite **une icône contact.** Une icône sera cliquable pour permettre aux administrateurs d'accéder au **formulaire de connexion** d'admin et toutes ses fonctionnalités.

L'icône user est un lien href pour se connecter. Lorsqu'une session admin est ouverte, un **dropdown menu** permet d'accéder aux différents onglets de l'admin.

Les différentes sous-pages de l'admin sont reprises dans **une balise nav**. Les sous-pages sont visibles grâce à un système de **dropdown**. Une méthode de la **classe Admin** permet de sélectionner les noms de 4 pages présentes dans l'admin. Une boucle **foreach** permet d'associer sous l'onglet admin les sous-pages associées. L'onglet et href "contact" permettra à l'utilisateur d'accéder au formulaire de contact permettant à l'utilisateur de faire parvenir tous types de demande.

Footer :Le footer est composé en 3 colonnes. La première partie fera un rappel du logo, la deuxième sera un href vers une page F.A.Q qui contiendra des réponses aux questions les plus fréquemment posées aux tatoueurs mais également un téléchargement de la feuille de soin. La troisième contiendra, à droite, les icônes des réseaux sociaux. Les 4 catégories principales sont reprises afin de pouvoir accéder aux pages sans passer par le header.

Page planning: elle met en avant toutes **les réservations de la semaine en cours** grâce à une méthode qui récupère dans un **select** les rendez-vous ajoutés pendant la semaine en cours. Le titre de **la bannière de la page** est généré dynamiquement. En effet, c'est le la semaine en cours qui est affiché grâce à la **fonction date** de php.

Page tatoueur: elle détaille l'ensemble des tatoueurs. Ses réalisations ainsi que ses flashs sont listés sur cette page. l'onglet tatoueurs ainsi que toutes les sous catégories (tatoueur1,

tatoueur2, tatoueur3, tatoueur4, tatoueur5) seront visible grâce à un système de **dropdown**. **Une méthode de la classe catégorie** permet de sélectionner les noms de tatoueurs. **Une boucle foreac**h permet d'associer le nom de chaque tatoueur à la catégorie tatoueurs.

Page connexion-admin : ce sont des formulaires avec **les inputs** indispensables à l'authentification. Des échanges avec la bdd sont effectués lors du remplissage des inputs permettant d**'afficher des messages d'erreur** en cas d'entrée d'informations erronées.

Page contact: un formulaire de contact qui sera récupéré par le tatoueur qui contient toutes les entrées de l'utilisateur dans ce formulaire. Ces informations sont affichées grâce à des méthodes de la **classe admin**.

Page admin: Une fois connecté l'admin pourra injecter du contenu entre deux pages, index.php et tatoueur.php. Il pourra accéder à son planning des rendez-vous de la semaine et à un formulaire d'ajout de réservation.

3.2.2 Responsive design

Afin de faciliter l'**adaptation de l'application web aux tablettes et mobiles**, j'ai utilisé des **display flex** pour composer les pages et l'ensemble des ses éléments.

J'ai utilisé l'inspecteur de Chrome afin de pouvoir visualiser l'application sur les différents formats d'écran. En fonction de la taille de l'écran j'ai modifié les dimensions et la disposition des différents éléments de chaque page avec les **Media Queries**.

J'ai changé le type de **mise en page en fonction de la taille de l'écran**. Au lieu d'avoir une seule mise en page pour toutes les tailles d'écran, la mise en page est modifiée. Les éléments sont repositionnés, les typographies réduites, les images redimensionnées pour les écrans plus petits.

3.3 Conclusion

Après avoir réalisé la charte graphique et le prototype de l'application web, j'ai procédé à l'intégration de la maquette pour les pages que j'ai développées d'un point de vue back et front. J'ai respecté la charte graphique et les éléments définis sur le prototype.

Aussi, la maquette ayant été développée pour une interface web, l'adaptation de certains éléments à **une interface mobile** s'est avérée laborieuse à plusieurs reprises. Il aurait été judicieux de réaliser une maquette pour mobiles pour prévoir ces difficultés ou de partir sur un principe de mobile first. Mes connaissances en UI UX méritent d'être approfondies afin d'offrir une expérience utilisateur plus pertinente.

4. Développement de la partie back-end de l'application web ou web mobile en intégrant les recommandations de sécurité

4.1 Organisation

Pour réaliser ce projet, j'ai choisi d'utiliser la technologie **PHP** et **P.O.O**.

Je me suis occupé de développer la partie connexion, inscription, gestion des administrateurs. J'ai généré des pages tatoueurs, des plannings qui contiennent les rendez-vous de chaque tatoueur. J'ai réalisé le **formulaire** qui permet d'envoyer les rendez-vous en **base de données** et directement dans le planning respectif du tatoueur.

Je me suis également occupé du transit de données entre la page contact et l'espace administrateur qui va regrouper les prises de contact mais aussi du formulaire de la page ajout_contenu.php qui permettra aux tatoueurs d'ajouter ses dernières réalisations et flashs disponibles.

Pour m'organiser, j'ai défini une liste de fonctionnalités. Après répartition de ces dernières un tableau de bord sur **Trello** m'a permis de suivre l'avancée de chacune.

Les outils suivants ont été utilisés pour les différentes étapes du développement de la partie back-end du projet :

- **Trello pour le suivi des tâches**
- **Diagramme de Gantt pour la répartition et estimation du temps passé**
- **LucidChart pour la conception et le schéma de la base de données**
- **Visual Studio Code pour l'I.D.E.**
- **phpMyadmin pour la création et gestion de la base de données**
- **PHP/MySQL pour le langage back et les échanges avec la base de données**
- **Github pour le versionning**

4.2 Création de la base de données

SCHÉMA DE BASE DE DONNÉES MCD ci dessous :

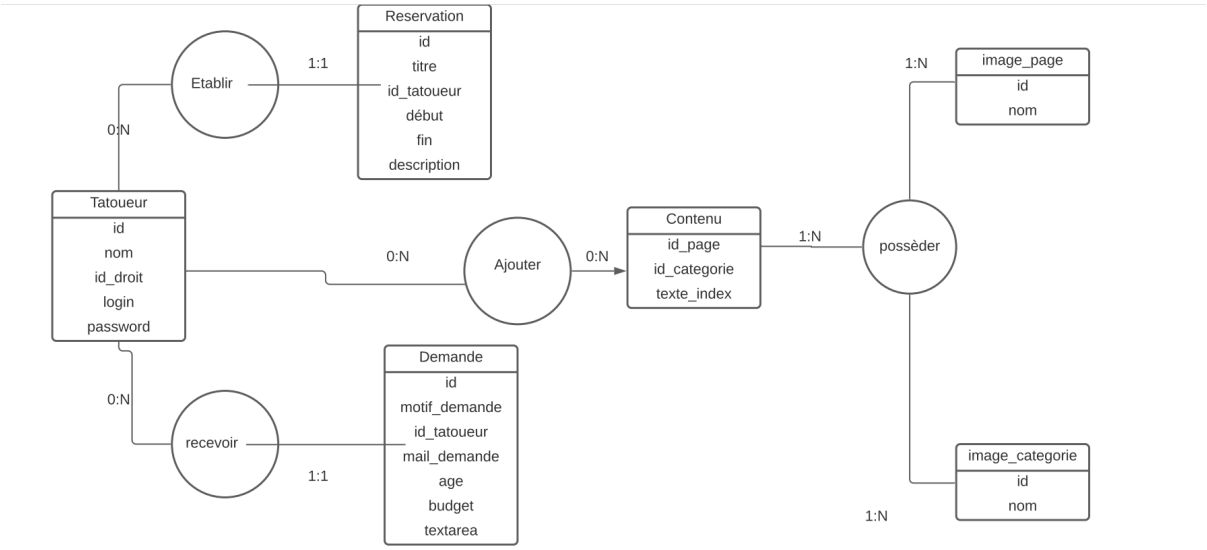
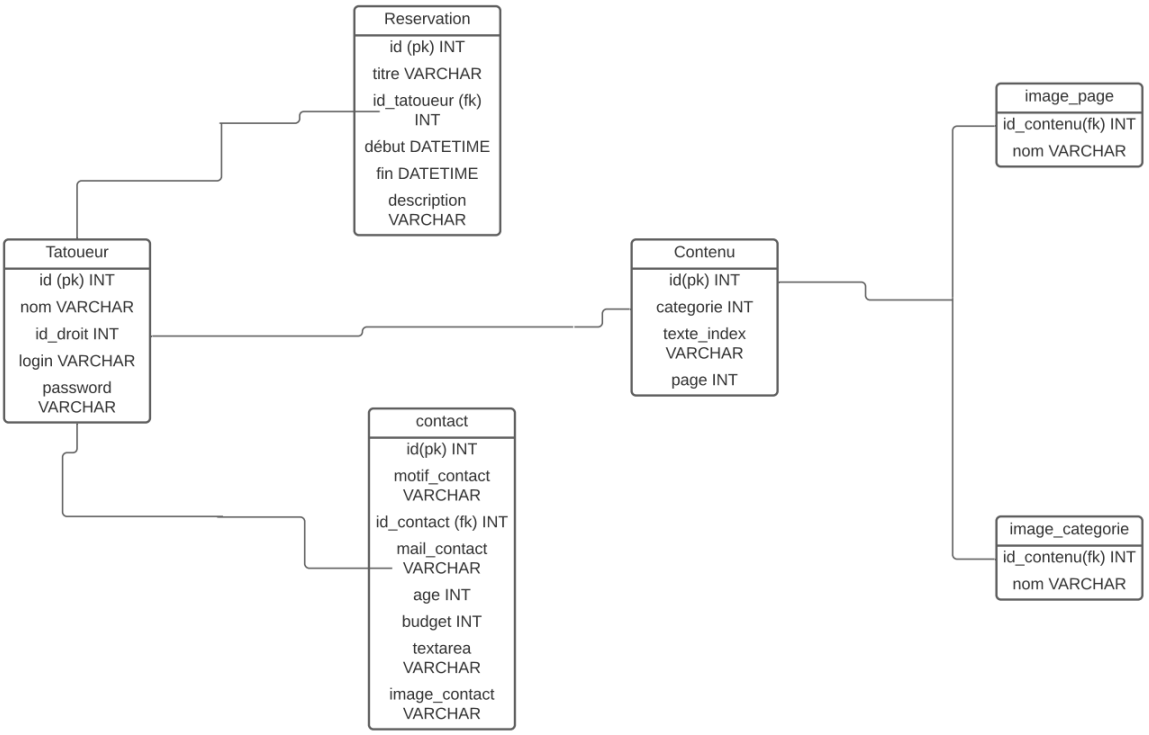


SCHÉMA DE BASE DE DONNÉES UML ci dessous :



J’ai réalisé **la base de données relationnelle** du site en collaboration avec l’équipe pédagogique. Nous avons pour cela listé les informations qui nous paraissent indispensables au bon fonctionnement du site. **La BDD comporte 6 tables**. On distingue **4 grands groupes** : les tables liées à l’ajout de contenu, la table qui fait référence aux réservations, la table liée à la prise de contact des clients et la table liée aux administrateurs.

La table “contenu” va servir de **table de liaison** avec les tables "image_catégorie" et “image_page”. En effet, dans ce cas la table “contenu” possèdera une ou plusieurs image_catégorie / image_page.

Ces tables d’associations sont utilisées dans le cadre d'une **cardinalité plusieurs-à-plusieurs** entre deux objets. Elle est composée d'au moins **2 clés étrangères**, référençant chacune l'un des 2 objets.

Pour accéder à la base de données dans les différentes classes et fonctions réalisées au cours du projet, j’ai développé **une classe Db**.

Classe Db : La fonction connectDb permet de se connecter à la base de données. J’ai utilisé un **« try and catch »** la fonction php pour gérer les erreurs. La gestion d’une erreur via une exception se fait en deux temps

On va utiliser un bloc try dans lequel le code qui peut potentiellement retourner une **erreur** va être exécuté. On crée à l'intérieur une nouvelle connexion grâce à l'objet **new PDO**.

```
$conn = new PDO("mysql:host=$servername;dbname=$dbname", $username,
$password);
```

On va créer un bloc catch dont le but va être d'attraper l'exception si celle-ci a été lancée et de définir la façon dont doit être gérée l'erreur. La fonction est appelée dans les autres classes afin de réaliser différentes requêtes.

4.3 Développement des composants d'accès aux données

Le projet a été réalisé **en P.O.O, programmation orientée objet** dans le langage **PHP**. J'ai codé les classes suivantes pour accéder aux données à travers différents composants :

- **Db** pour se connecter à la base de données
- **Admin** pour gérer le profil administrateur, l'inscription, la connexion, la suppression
- **Réservation** pour générer les réservations dans un planning hebdomadaire propre à chaque tatoueur.
- **Demande** pour gérer les requêtes de chaque client entrées dans le formulaire de contact.
- **Contenu** pour gérer l'ajout de contenu soit sur la page index.php soit sur la page tatoueur.php
- **Image_page et image_categorie** me permettra d'effectuer un INNER JOIN de la table contenu.

4.4 Développement de la partie back-end de l'application

4.4.1 Fonctionnalité significative la page admin

La page admin est l'une des artères principales du site internet, en effet beaucoup d'opérations non visibles par un utilisateur lambda seront visibles par l'administrateur :

Pour **identifier un utilisateur en tant qu'administrateur** il faut faire appel à la base de données comme une connexion classique. Il s'agit d'interroger la base de données pour savoir si le login et le mot de passe entrés dans les champs du formulaires correspondent à ceux inscrits en base de données. S' ils correspondent, une connexion est établie.

Extrait de code illustrant une connexion administrateur:

```php
// CONNEXION DE L'ADMIN
    public function connectAdmin(){
        $login = $_POST['login'];
        $password = $_POST['password'];

        $login = htmlspecialchars(trim($login));
        $password = htmlspecialchars(trim($password));
        if (isset($_POST['connectAdmin']) && !empty($login)&&
!empty($password)) {
            $db = new Database;
            $db->connect();

            $connectAdmin = $db->conn->prepare("SELECT * FROM admin
WHERE login = :login");
            $connectAdmin->bindValue(':login', $login, PDO::PARAM_STR);
            $connectAdmin->execute();
            $admin = $connectAdmin->fetch();
            var_dump($admin);
        }else{
        echo"aeoihfaelbgaelbjg";
        }
        if (!empty($login)) {
            echo $password . "<br>";
            echo $admin['password'];
            if (password_verify($password, $admin['password'])) {
                $this->id = $admin['id'];
                $this->login = $admin['login'];
                $this->password = $admin['password'];
                $this->droit = $admin['id_droit'];

                $_SESSION['admin'] = [
                'id' =>
                    $this->id,
                'login' =>
                    $this->login,
                'password' =>
                    $this->password,
                'droit' =>
                    $this->droits,
                ];
                var_dump($_SESSION['admin']);
                header('location:../pages/panel_admin.php');
```

```
            }else{
                echo "Mot de passe erroné";
            }
        }else{
            echo 'veuillez remplir les champs';
        }
    }
```

Page administrateur :

- La page admin sera par définition uniquement accessible aux administrateurs, si une session administrateur est initiée alors le header changera en conséquence, en effet il contiendra un onglet “ajout contenu”, “ajout rdv”, “planning” ainsi qu’un bouton déconnexion. Elle permet aussi à l’administrateur principal de conférer les droits d'administrateurs aux personnes qu’il désire.

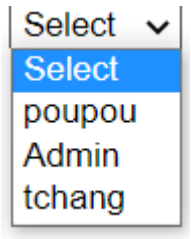
Extrait de code illustrant l’ajout et la suppression de tatoueurs par l’admin:

```
    public function registerAdmin(){
    $login = $_POST['login'];
    $password = $_POST['password'];
    $confirmPW = $_POST['confirmPW'];

    $login = htmlspecialchars(trim($login));
    $password = htmlspecialchars(trim($password));
    $confirmPW = htmlspecialchars(trim($confirmPW));

    if (!empty($login) && !empty($password) && !empty($confirmPW)) {
        $db = new Database;
        $db->connect();
        $register = $db->conn->prepare("SELECT login FROM admin WHERE
login=:login");
        $register->bindValue(":login", $login, PDO::PARAM_STR);
        $register->execute();
```

```
        $fetch = $register->fetch();
            if (!$fetch){
                if ($password == $confirmPW) {
                $cryptedpass = password_hash($password,
PASSWORD_ARGON2I);
                $insert = $db->conn->prepare("INSERT INTO admin (login,
password, id_droit)  VALUES (:login, :cryptedpass, 1337)");
                $insert->bindValue (":login", $login, PDO::PARAM_STR);
                $insert->bindValue (":cryptedpass", $cryptedpass,
PDO::PARAM_STR);
                $insert->execute();
                echo"Nouvel administrateur ajouté";
                }else{
                    echo "Les mots de passe ne correspondent pas";
                }
            }else{
                echo "l'utilisateur n'existe pas";
            }
        }else {
            echo"veuillez remplir tous les champs";
        }
    }
```

Le tatoueur ajouté sera ensuite affiché dynamiquement grâce à une méthode PHP de la classe admin dans la page panel_admin. Le select reprend tous les noms des tatoueurs inscrits en base de données.

```
    public function dropDownDisplay(){
        $db = new Database;
        $db->connect();
        $admin = new Admin;
        $adminTab = $admin->dropDown();

        foreach ($adminTab as $value) {
            echo '<option value="' . $value[0] . '">' . $value[1] .
'</option>';
            var_dump($value[0]);
        }
    }
```



Extrait de code illustrant la suppression de tatoueurs par l'admin:

```
    public function deleteAdmin(){
    $login = $_POST['login'];
    $db = new Database;
    $db->connect();

    $deleteAdmin =  $db->conn->prepare("DELETE FROM admin WHERE id =
:login");
    $deleteAdmin->bindValue(":login", $login, PDO::PARAM_INT);
    $deleteAdmin->execute();
    }
```

- Ajout contenu contiendra un formulaire qui permettra à l'administrateur d'ajouter des images soit dans la partie "flash disponibles" ou "dernières réalisations" de la page tatoueur soit d'ajouter du texte à la page d'accueil (congés annuels, tatoueurs invités, offres spéciales, évènements).

Au moment de l'ajout le contenu sera traité par des classes CSS qui permettra de dimensionner et styliser automatiquement le contenu ajouté.

Cette page est composée de plusieurs inputs:

- un input qui permettra de préciser le nom de l'image ajoutée.
- un input qui permettra de choisir à quel tatoueur correspond l'image et donc l'ajouter sur la page du bon tatoueur.
- un input qui permettra de préciser la taille du tatouage
- un input qui permettra de choisir l'emplacement final de l'image : soit dans les "dernières réalisations" soit dans les "flash dispos".
- un input qui permettra d'uploader l'image en question.
- Un input permettant d'envoyer toutes les informations ci-dessus vers la page voulue.

- Une fois qu'un tatoueur décide de prendre en charge une demande client il se dirige vers la page Ajout Rdv. Elle représente **un formulaire** qui permettra l'ajout d'un RDV directement

sur la page planning. La page contiendra un input qui spécifie le titre, un input qui permettra d'y ajouter une description approfondie, un input qui spécifie le début et la fin de la séance et pour terminer un input permettant d'envoyer le RDV sur le planning.

- **Planning** affichera les différents rendez-vous de la semaine en cours insérés par le tatoueur dans la page Ajout rdv. Chaque tatoueur aura un planning dédié accessible via une connexion en tant qu'administrateur.

-**Page contact** : La page contact est un formulaire qui permettra d'envoyer un formulaire de contact qui sera transmis par mail sur la boite mail du salon de tatouage. Ce formulaires contient plusieurs inputs :

- un qui permet de choisir parmis les différents motifs de prises de contact : Demande de RDV, renseignements, retouches ou partenariats
- un qui permet à l'utilisateur qui remplit le formulaire de spécifier son âge. Ce qui spécifie au tatoueur si le/la client(e) est majeur(e).
- un input permettra de choisir à qui la demande sera adressée
- un input qui permettra au client de spécifier une fourchette de budget (50-100€, 150-300€ , 300€ et plus)
- un input qui permettra au client de télécharger des photos dites "d'exemple" qui seront jointes au mail.
- Pour terminer il y a un textarea qui permettra au client de préciser au mieux sa demande, de manière à ce que la requête soit la plus imagée et claire pour le tatoueur qui s'apprête à prendre la suite du dossier.
- Enfin un dernier input permettra au client d'envoyer sa requête.

4.5 Veille et sécurité

Il est important de travailler sur la veille sur la sécurité pour ce projet. Nous nous sommes appuyées sur la riche documentation de l'OWASP et effectué différentes recherches pour lister les principales failles et trouver les solutions pour les contrer.

4.5.1 référencement des principales failles de sécurité existantes

Oubli de valider les entrées des utilisateurs. Injections dans les formulaires.
La plus basique et sûrement la plus connue, est l'injection SQL. Deux tiers des attaques sur le web portent dessus.

- **Contrôle d'accès inefficace.**

Il s’agit de la possibilité pour un pirate d’utiliser l’identité d’autres personnes sur un site Internet, si des éléments permettant d’authentifier un utilisateur (login / mot de passe) sont mal protégés dans le site web. La deuxième possibilité est de pouvoir deviner ou modifier les éléments d’authentification facilement.

- **Mauvaise gestion des sessions.**

Cette faille exploite des faiblesses dans les mécanismes qui permettent au serveur du site web de se souvenir de qui vous êtes, une fois que vous vous êtes authentifié.

- Cross Site Scripting

Cette faille touche les sites web qui laissent les internautes publier du code HTML susceptible d'être vu par les autres utilisateurs du site (dans un forum, par exemple). Cela permet d'exécuter des contenus dynamiques sur les navigateurs, avec les droits associés au site web.

- Dépassement de mémoire tampon ou buffer overflow

Une faille qui consiste en une corruption de la mémoire, bien souvent la mémoire de la pile des appels. La plupart du temps, le programme va planter, mais ceci ouvre aussi une porte au hacker qui veut contrôler un processus à distance.

- Injection de commandes

L’injection de commande (ou Shell Code Injection) est une attaque qui consiste à exécuter des commandes systèmes non autorisées sur le système d’exploitation d’une victime via une application vulnérable.

- Désérialisation non sécurisée (Insecure Deserialisation)

Une vulnérabilité de type “insecure sérialisation" permet à un utilisateur malveillant d'accéder et de modifier les fonctionnalités de l’application ciblée.

Mauvaise utilisation du chiffrement

Pour stocker des informations sensibles il faut les convertir en une chaîne de caractères illisible, on dit haché, pour qu’elles ne soient plus lisibles de manière irréversible.

- Utiliser un logiciel ou des composants présentant des vulnérabilités

Lorsqu'une faille est découverte, les développeurs de l'application en question proposent généralement un patch qui permet de corriger le problème. Cependant, si la mise à jour n'est pas faite, l'application s'expose à la faille.

- Défaut dans la configuration des paramètres de sécurité

Elle est due à une configuration par défaut non sécurisée, des configurations incomplètes, des messages d'erreurs contenant des informations sensibles.

4.5.2 pratiques suivies pour sécuriser le site

Les normes consultées :

- ISO/IEC 27000
- RGPD règlement général sur la protection des données
- L'Open Web Application Security Project (OWASP) est un organisme impartial, mondial et sans but lucratif. Il évalue les dix principaux risques pour la sécurité des applications web et préconise un développement logiciel sécurisé.

Sécurisation de l'application contre l'injection SQL

- Les requêtes préparées : on peut écrire les requêtes SQL en paramétrant les variables. C'est ce qu'on appelle une requête préparée. Préparer la requête permet de l'exécuter une fois que l'on a stocké les arguments à envoyer en base de données. Une vérification sur le type de données que l'utilisateur a entré est ainsi effectuée en amont.

exemple de requête préparée dans la classe user pour un INSERT en bdd :

```
public function newsletter($email){

     $connexion = $this->db->connectDb();

     $this->newsletter = ($_POST['newsletter']);

     //var_dump($this->newsletter);

         $q = $connexion->prepare(

         "INSERT INTO newsletter(email_utilisateur) VALUES (:email)"
```

```
            );

            $q->bindParam(':email', $email, PDO::PARAM_STR);

            $q->execute();

            header('location:connexion.php');

    }
```

Protection des données stockées sur l'application :

- Les algorithmes de hachage pour crypter les données : Il existe de nombreux algorithmes de hachage : Bcrypt, Scrypt, SHA, MD5, Argon5 et PBKDF2, par exemple. Crypter le mot de passe avec le hachage permet de générer une empreinte unique pour une entrée. Cependant, cela n'empêchera pas le phishing qui reste une méthode très utilisée par les hackeurs pour récupérer les mots de passe des utilisateurs.

Hachage du mot de passe pour la création d'un compte utilisateur pour un insert en bdd dans la classe Admin:

```
            if (!$fetch){
                if ($password == $confirmPW) {
                $cryptedpass =
password_hash($password,PASSWORD_ARGON2I);
                $insert = $db->conn->prepare("INSERT INTO admin (login,
password, id_droit)  VALUES (:login, :cryptedpass, 1337)");
                $insert->bindValue (":login", $login, PDO::PARAM_STR);
                $insert->bindValue (":cryptedpass", $cryptedpass,
PDO::PARAM_STR);
                $insert->execute();
                echo"Nouvel administrateur ajouté";
```

- **import de fichiers :** La faille upload est un risque rencontré lorsqu'on permet à un utilisateur de télécharger des documents sur le site web. Il est notamment aisé de télécharger un document contenant un malware sur le site web ou la base de

données. L'administrateur pouvant importer des photos, quelques tests sur le fichier à télécharger (format, taille etc) ont été effectués. Cependant, il ne faut pas se fier uniquement au nom du fichier, car il est possible de nommer un fichier d'une façon trompeuse, telle que "fichier.exe.jpg" pour passer le filtre avec succès.

Empêchez le piratage de session

- demande d'un mot de passe fort pour les utilisateurs contenant des majuscules, des minuscules, des chiffres et des caractères spéciaux

- accès limité Permettre l'accès à certaines parties du site uniquement aux personnes ayant les droits suffisants. Pour chaque page, des vérifications sont faites : l'utilisateur est-il connecté, est-il un administrateur, etc. Ces vérifications permettent ainsi de limiter le risque que des personnes non habilitées puissent accéder à certaines informations sensibles.

Autres préconisations à effectuer sur les versions suivantes de l'application :

- demande aux utilisateurs qu'ils changent régulièrement leur mot de passe en cas d'attaque de credential stuffing ;

- implémentation d'une authentification forte, c'est-à-dire avec plusieurs facteurs d'authentification, comme la validation par SMS ou par mail, par exemple.

- Le cas des cookies de session :
  - ➔ s'assurer que les cookies sont chiffrés lors de la transmission via HTTPS ;
  - ➔ pas de stockage d'informations d'identification en texte clair dans les cookies
  - ➔ définir une date d'expiration pour les cookies-session.
  - ➔ ne pas mettre pas l'ID de session dans l'URL ;

- PHP possède une bibliothèque appelée SessionManager avec des fonctions qui peuvent être utilisées pour valider les sessions avec des restrictions

- Le certificat SSL pour protéger le mot de passe lorsqu'il est transmis sur le réseau. Obtenir un certificat SSL et l'ajouter au serveur. Ce certificat est nécessaire pour chiffrer les données en cours de transmission.

- Sécurisation avec l’API OWASP

L'organisation OWASP dispose d'une API appelée the OWASP Enterprise Security API (ESAPI). Elle peut être utilisée pour sécuriser vos applications web

- Utilisation d’un pare-feu d'application web ou WAF, pour Web Application Firewall. Ce pare-feu se place entre l'utilisateur et l’application web et permet de vérifier et d'intercepter les données envoyées.

4.5.3 Conclusion

La sécurité est un enjeu majeur dans le développement d’une application web. Il est indispensable de mettre en place toutes les stratégies possibles à la sécurisation des données dans toutes les étapes du code. Cependant, d’autres implémentations, citées précédemment, sont nécessaires pour améliorer la sécurité du site pour sa mise en ligne.

4.6 Recherche Anglophone

4.6.1 Description d’une situation nécessitant une recherche sur un site anglophone

Sur le panel admin j’ai généré **une liste déroulante** à l’aide d’une **boucle foreach**. Cette boucle récupère l’ensemble des tatoueurs inscrits en base de données avec leur nom. Rencontrant des difficultés pour récupérer les noms j’ai effectué une recherche en anglais sur Google : “Dropdown Menu foreach PHP” qui m’a dirigé sur une page stackoverflow.com.

4.6.2 Extrait du site anglophone

This is what is stored in the $names variable.

> Array ([0] => Array ([name] => Web Design) [1] => Array ([name] => Art History) [2] => Array ([name] => Gym) [3] => Array ([name] => English) [4] => Array ([name] => Biology) [5] => Array ([name] => 3D Animation) [6] => Array ([name] => Tech Disc) [7] => Array ([name] => Math) [8] => Array ([name] => Dance) [9] => Array ([name] => Video Production) [10] => Array ([name] => Home Ec) [11] => Array ([name] => Government) [12] => Array ([name] => Physics))

I have this dropdown list created and it work, but it is hand coded for each value in the array. I want to modify this so it 'loops' through all results in the array to create the dropdown.

```
<label for="per1"></label>
<select name="per1" id="per1">
    <option selected="selected">Choose one</option>
    <option value="<?php echo $names[0]['name'];?>"><?php echo $names[0]['name'];?></optior
    <option value="<?php echo $names[1]['name'];?>"><?php echo $names[1]['name'];?></optior
    <option value="<?php echo $names[2]['name'];?>"><?php echo $names[2]['name'];?></optior
    <option value="<?php echo $names[3]['name'];?>"><?php echo $names[3]['name'];?></optior
    <option value="<?php echo $names[4]['name'];?>"><?php echo $names[4]['name'];?></optior
    <option value="<?php echo $names[5]['name'];?>"><?php echo $names[5]['name'];?></optior
    <option value="<?php echo $names[6]['name'];?>"><?php echo $names[6]['name'];?></optior
    <option value="<?php echo $names[7]['name'];?>"><?php echo $names[7]['name'];?></optior
    <option value="<?php echo $names[8]['name'];?>"><?php echo $names[8]['name'];?></optior
    <option value="<?php echo $names[9]['name'];?>"><?php echo $names[9]['name'];?></optior
    <option value="<?php echo $names[10]['name'];?>"><?php echo $names[10]['name'];?></opti
    <option value="<?php echo $names[11]['name'];?>"><?php echo $names[11]['name'];?></opti
    <option value="<?php echo $names[12]['name'];?>"><?php echo $names[12]['name'];?></opti
</select>
```

Can someone please help?

This solution works for current PHP versions. Simple case of using a foreach:

```
<select name="per1" id="per1">
  <option selected="selected">Choose one</option>
  <?php
    foreach($names as $name) { ?>
      <option value="<?= $name['name'] ?>"><?= $name['name'] ?></option>
  <?php
    } ?>
</select>
```

4.6.3 Traduction en Français du site

Voici ce que j'ai stocké dans le nom des variables. J'ai cette liste déroulante de créée et qui marche. Mais c'est écrit en dur pour chaque valeur du tableau. Je voudrais le modifier de manière à ce que ca boucle tous les résultats du tableau pour créer le menu déroulant. Quelqu'un peut m'aider?

5. Conclusion

Le projet Ornithorynque Tattoo Shop a permis de développer une interface web complète en termes de site vitrine. Je la considère comme une version bêta car elle nécessite des améliorations avant de pouvoir la mettre en ligne.

Les mesures de sécurité doivent être adaptées à un usage concret. Une optimisation de la base de données est aussi à envisager pour gérer l'ensemble des données dans l'optique d'une adoption d'un module e-commerce.

La partie formulaire de contact pour gérer les relations avec les futurs clients est à revoir pour envisager une automatisation des demandes clients. Une phase de test unitaire doit être réalisée afin de vérifier l'ensemble des fonctionnalités du site.

J'ai acquis de nouvelles compétences notamment pour la création du planning, et la gestion du profil administrateur avec les vérifications nécessaires à son enregistrement.
J'ai pris beaucoup de plaisir à développer ce site mais j'ai aussi réalisé de l'ampleur des intérêts qui entrent en jeu dans la conception et le développement d'un site web tant d'un point de vue expérience utilisateur que du traitement des données.