

# **Dossier de Projet**

Création de Site Web

. Site web Peggy Coiffure

**PRÉSENTÉ PAR HUGO TOUMI & STEPHANE MATHIEU**

# *Sommaire*

- 1 Introduction
- 2 Présentation
- 3 Présentation personnelle
- 4 Présentation de l'entreprise Peggy Coiffure
- 5 Description du poste
- 6 Compétences techniques demandées
- 7 Réalisations en développement Front-End
- 8 Développement de la partie Back-End
- 9 Organisation
- 10 Création de base de données
- 11 Veille et sécurité
- 12 Spécifications techniques
- 13 Difficultés rencontrées
- 14 Solutions mises en œuvre
- 15 Conclusion
- 16 Synthèse

## *Introduction*

### Résumé du dossier de projet :

Ce dossier présente une partie du travail effectué en tant que développeur web à La Plateforme.

Durant l'année nous avons réalisé de nombreux projets en PHP , HTML , CSS , JAVASCRIPT et dans ce dossier Je vais vous présenter le dernier projet qui finalise les connaissances acquises durant cette année d'apprentissage.

## *Présentation*

### Présentation personnel :

Je m'appelle Hugo Toumi, j'ai 20 ans, je suis actuellement étudiant en développement web à la Plateforme et j'habite à Marseille.

Après mon bac professionnel en vente et une expérience professionnelle de 8 mois dans une entreprise de communication.

Je décide cette année de reprendre mes études dans un domaine qui me passionne depuis longtemps : le développement web.

## *Présentation de l'entreprise Peggy Coiffure*

### Présentation de l'entreprise :

Peggy Coiffure est une entreprise française située à Marseille qui propose des prestations de coiffure à domicile, avec plus de 30 ans d'expérience dans la coiffure.

Soucieuse de son image sur internet je me suis proposé et présenté en tant que développeur web full stack pour créer son site internet accompagné de mon collègue de travail Stéphane Mathieu.

## *Description du Poste*

### Description du poste :

Nous étions à deux pour créer ce site internet accompagné de mon collègue de travail Stéphane Mathieu nous avons la responsabilité de gérer toutes les étapes de la création de ce site vitrine.

Passant par le maquettage du site avec une maquette basse qualité et une autre haute fidélité pour avoir une vision commune et d'ensemble dans la réalisation graphique de ce dernier.

Ou alors la conception de la base de données pour la gestion des profils créer par l'utilisateur.

## *Compétences technique demandées :*

### Connaissance Informatique :

#### *Des langages HTML & CSS*

Nous avons utilisé le HTML / CSS comme structure des pages à développer et pour l'intégration. Le CSS m'a aidé pour le responsive des pages en utilisant des **media queries**.

#### *Un logiciel de maquettage*

J'ai approfondi mes connaissances en design grâce à la gestion du logiciel de maquettage "**Figma**"

#### *En langage PHP*

Le côté Back-End a entièrement été réalisé en PHP.

#### *Utilisation du framework Bootstrap*

Certains éléments du site ont été réalisés avec Bootstrap pour faciliter le responsive des éléments choisis comme par exemple le header du site.

#### *En langage JS*

Nous avons aussi intégré du **Javascript** dans l'aspect design du site comme par exemple les animations pour apporter plus de dynamisme à ce dernier.

Le **Javascript** a aussi été utilisé dans la création du formulaire d'inscription.

## *Réalisation en développement Front-End:*

- HTML/CSS pour la structure du site
- Bootstrap pour le CSS et le responsive
- Javascript pour l'interaction utilisateur

J'ai gardé en tête pour ce projet que l'utilisateur devrait pouvoir se servir du site sur n'importe quel navigateur et depuis n'importe quel périphérique.

En conséquence, nous avons développé le CSS du site en nous servant des **medias-queries** afin de modifier l'affichage en fonction de la taille de l'écran.

Nous avons également fait en sorte que le comportement de l'application soit adapté à un maximum de périphérique différents.

## Réalisation en développement Back-End:

Avant de commencer le projet nous avons fait plusieurs recherches sur les choix techniques et nous avons mis en place une méthode de travail.

### Versioning et stockage

Nous avons utilisé GIT comme logiciel de **versionning** couplé à **GITHUB** pour stocker notre code versionné sur un serveur distant et pouvoir ainsi travailler facilement.

Nous utilisons principalement **1 branche master**.

Voici notre repo :

<https://github.com/hugo-toumi/peggycoiffure/tree/master>

Nous avons commencé par développer en local sur notre machine respective. Une fois les changements effectués nous effectuons des “comit” et des “push” sur notre repository **GITHUB**

Nous utilisons l'éditeur de code **Visual Studio Code** pour faire les “comités” les “pushs” et les “pulls”

Il nous est aussi arrivé de “push” et de “comit” via le terminal

Voici l'arborescence du site web dans laquelle seront intégrées les différentes fonctionnalités

## Accueil

- Inscription
- Connexion

## Page d'administration

- Afficher le profil utilisateur
- Modifier le profil utilisateur

## Courte description de chaque fonctionnalité

- **Accueil** présentation de la page. Après la visite de la page d'accueil le visiteur doit avoir bien compris ce qu'est **Peggy Coiffure** et ce que cela peut lui apporter.
- **Inscription** : permet à un visiteur anonyme de s'inscrire.
- **Connexion** : permet à un visiteur inscrit de se connecter.
- **Afficher le catalogue** : liste des toutes les prestations disponible
- **Afficher le profil utilisateur** : donne accès au visiteur l'accès aux informations de son profil
- **Modifier le profil utilisateur** : formulaire permettant à l'administrateur de modifier les paramètres de son profil.



## Permissions et fonctionnalités

Le site web se comporte différemment selon le statut du visiteur

**J'ai identifié 2 type de visiteurs :**

Anonyme : le visiteur anonyme

### Utilisateur et administrateur

le visiteur connecté

le visiteur connecté et disposant des droits d'administrations

Ces 2 profils types ont le même point d'entrée sur le site web mais pas par les mêmes possibilités.

## Fonctionnalités en fonction des permissions

### Anonyme

- Accueil
- Inscription
- Connexion
- Afficher les tarifs

### Utilisateur et administrateur

- Accueil
- Connexion
- Afficher les tarifs
- Afficher le profil utilisateur
- Modifier le profil utilisateur
- Se déconnecter du site

## Création de base de données :

Concrètement j'ai commencé par créer une base de données dans **PHPmyAdmin**.

J'ai choisi **PHPmyAdmin** car c'est l'outil le plus utilisé en formation.

Donc la structure a été créée depuis l'interface de **PHPmyAdmin**  
Nous l'avons nommé " web " en encodage UTF8-mb4 (pour la gestion des émoticônes.)

Elle contient la table " User "

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1 id	int(11)			Non	Aucun(e)		AUTO_INCREMENT	 Modifier  Supprimer  Plus
<input type="checkbox"/>	2 email	varchar(255)	utf8mb4_general_ci		Non	Aucun(e)			 Modifier  Supprimer  Plus
<input type="checkbox"/>	3 password	varchar(255)	utf8mb4_general_ci		Non	Aucun(e)			 Modifier  Supprimer  Plus

 ☐ Tout cocher Avec la sélection :  Parcourir  Modifier  Supprimer  Primaire  Unique  Index  Spatial  Texte entier

La première étape, avant de se lancer dans la conception du site, est de créer une connexion directe avec la base de données pour pouvoir stocker des données par la suite. Pour cela, j'ai créé un nouveau dossier, à la racine du dossier du site, que j'ai nommé 'bd'. Dans ce dossier, se trouve un nouveau fichier que j'ai appelé Database.php. Il contient le code permettant d'établir la connexion. J'ai renseigné quelques informations essentielles, telles que :

- Le nom de l'host, (si je suis en local : localhost).
- Le nom de ma base de données.
- Le nom de l'utilisateur ('root').
- Le mot de passe.. Voir extrait de code:

libraries > Models > Database.php > ...

```
1  ?php
2  namespace Models;
3  use PDO;
4
5  class Database {
6      public static function getPdo(): PDO{
7          $pdo = new PDO('mysql:host=localhost;dbname=web;charset=utf8', 'root', '', [
8              PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
9              PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC
10         ]);
11         return $pdo;
12     }
13 }
14
15
16  ?
```

## Les langage de développement back-end

Au niveau du back-end et de la base de données, j'ai choisi PHP afin de pouvoir récupérer les informations contenues dans les requêtes (POST) faites au serveur. J'ai choisi de travailler avec PHP, langage le plus abordé au cours de ma formation.

Concernant la base de données, PHPmyAdmin m'a déjà fourni un système relationnel. Nous avons choisi d'héberger la base de données sur un serveur distant, plus simple qu'en local.

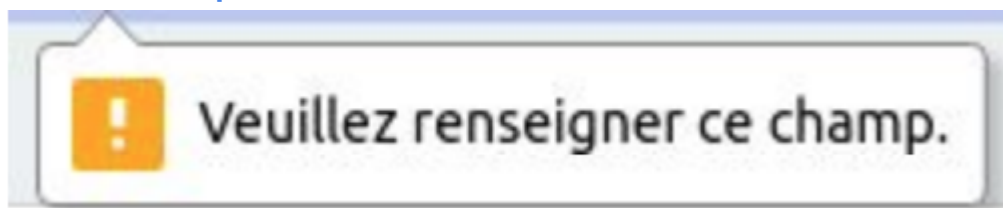
Veille et sécurité

## Exemple d'un composant

Voici, en trois parties, un exemple concret d'un composant que j'ai codé :

➤ D'un point de vue utilisateur ➤ Comment utiliser ce composant ? ➤ Comment fonctionne ce composant ? Le composant est une notification dont le but est d'avertir l'utilisateur d'une information (généralement pour remplir tous les champs des formulaires), sans perturber sa navigation au sein de l'application.

Voici un exemple :



En gris clair l'arrière-plan de l'application, avec la notification arrivant sur chaque champ non rempli. Celle-ci est composée d'un cadre en forme de bulle rectangulaire, d'un message au centre, et à gauche d'un caractère spécial. Celui-ci est couplé à une couleur de fond permettant à l'utilisateur d'identifier en un clin d'œil le type de notification. Dans l'exemple ci-dessus, la notification correspond à un champ non renseigné. Pour la page d'inscription, la notification est une 'info' :

➤ Pour renseigner correctement tous les champs. Au cas où l'email existerait déjà (chaque email est unique dans la base de données). ➤ Pour ne pas oublier de mettre un arobase dans l'email.

Prénom

Ce mail existe déjà

Email

Mot de passe

Confirmez mot de passe

Pour la page de connexion, la notification est une 'info':

- Pour renseigner correctement tous les champs.
- Pour ne pas oublier de mettre un arobase dans l'email (sinon il n'est pas pris en compte).

Peggy Coiffure

Inscription / Connexion

Tarifs

Contact

Mon histoire

Email

Adresse pas Bon format

Mot de passe

Le mot de passe doit contenir au moins 9 caracteres

Connexion

## Comportement

Les notifications sont affichées en dessous de chaque champ ou parfois au-dessus comme texte. Une notification apparaît au clic . Elle reste ensuite sur l'écran :

- 5 secondes si elle est de type 'info'. Puis elle disparaît.

## Comment utiliser ce composant ?

Lorsque j'ai réalisé ces notifications, je voulais qu'elles soient simples à paramétrer et mettre en place, accessibles à quelqu'un qui ne connaîtrait pas le code. Pour cela, j'ai représenté chaque champ du formulaire relié à:

- un id
- un nom
- prénom
- un email
- un mot de passe
- la date de création du compte

Si la variable "\$\_Post" contient les informations, alors je les traite. Je place sur le bon formulaire grâce au "name" de la balise "input". Et je fais les vérifications ensuite.

## Comment fonctionne ce composant

D'une manière générale

Pour avoir accès à toutes les pages du site, et ainsi pour devenir membre administrateur de son profil, un visiteur doit s'inscrire. Sur la page d'inscription, il entre les informations dans les champs du formulaire. Il est ainsi validé et inscrit dans la base de données.

J'ai créé mon formulaire avec le framework Bootstrap. Ce formulaire fonctionne avec la méthode POST. Il présente les différents champs

de saisie et le bouton pour soumettre le formulaire. Le 'name' du bouton se nomme 'inscription', il enregistre toutes les valeurs des champs. Je reçois les données en PHP grâce à une session 'start' qui sert à savoir si une connexion est démarrée, pour

restituer ensuite les valeurs de cette session. Je récupère la valeur du \$POST, si elle est remplie. Puis je vais l'extraire et la placer vers le bon formulaire. Les valeurs des champs sont générées grâce au 'name'. Voici les premiers traitements :

- Les 'htmlentities': pour convertir tous les caractères éligibles en entités HTML.
- Les 'trim' : pour enlever les espaces avant et après.
- Les 'strtolower': pour mettre les mails en minuscules ;

En parallèle, je vérifie :

- Si l'utilisateur a bien entré un nom, un prénom, un mail, un mot de passe... Il faut s'assurer pour le mail que la règle de construction a bien été respectée (preg-match : vérifie '@', '.com').
- Si le mail de l'utilisateur existe. Pour cela, une condition précise qu'un utilisateur ne peut pas se connecter avec la même adresse mail pour deux comptes différents. L'adresse email restera unique.

J'effectue aussi une vérification dans la base de données pour voir si cette adresse existe vraiment. J'utilise à la fois ma connexion à la base de données et la fonction Query. Si j'ai un retour d'infos SELECT, c'est-à-dire un retour du mail, je transmets un message d'erreur pour avertir que ce mail existe déjà.

- Le mot de passe et la confirmation du mot de passe. Si le mot de passe est différent, alors un message d'infos précise que le mot de passe ne correspond pas.

Lorsque toutes les conditions sont remplies - que '\$valid' est égal à 'TRUE' - alors le mot de passe est crypté avec une clé unique (je suis allée chercher l'information dans la documentation de PHP en ligne - fonction crypt.php). J'ai copié et collé une clé que je peux personnaliser. Ensuite, quand un utilisateur s'inscrit, une date de création du compte est générée et s'insère directement dans la base de données. Puis, la fonction INSERT permet d'insérer toutes les données des champs.

Chaque ' ? ' correspond à une variable, dans l'ordre. '\$prenom', '\$nom', '\$mail', '\$mdp' sont stockés dans un tableau. En parallèle, je rafraîchis pour dire sur quelle page je souhaite me positionner, suivi d'un 'exit' pour stopper le traitement (on ne va pas exécuter de nouveau un traitement si on fait un retour en arrière). Le but de cette méthode est d'arriver à la fin d'une inscription en PHP. Ceci permet de choisir une quantité d'informations à récupérer, limiter et sécuriser les données, vérifier si elles sont utiles.

## Exemple

Exemple :

- Enlever les espaces avant et après.
- Vérifier que le mail est bien une adresse email.
- Confirmer le mot de passe et le crypt pour garantir la sécurité de ma base de donnée et de mon site



### *Difficultés rencontrées:*

Dans l'ensemble nous nous sommes bien organisés dans notre gestion du travail et du temps.

Nous avons respecté toutes nos deadlines et les fonctionnalités vital et principal pour ce projet.

La ou la chose s'est compliqué est au niveau de la gestions des imprévus mais nous avons su garder le cap de manière générale

### **Solution mises en oeuvre**

Ayant eu plusieurs projet “ stressant ” durant cette année nous avons pu régler certaines choses comme par exemple la mise en place d'un travaille commun en présentiel pour être plus rapide dans la résolution des problèmes rencontrés

### *Conclusion:*

J'étais enthousiaste à l'idée d'organiser et gérer le développement de ce site. Et comme cité plus haut le gain d'expérience nous a aidé énormément dans la réalisation de ce projet.

## *Synthèse du projet :*

Enfin je garde un plutôt bon souvenir de ce projet , il est l'apogée du fruit d'un travail d'une année dans le développement web.

Mon collègue et moi avons vraiment travaillé de manière collaborative et sincère durant l'élaboration de ce projet , pour nous la satisfaction de la cliente était primordiale dans la finalisation de notre projet.

Ce projet m'a aussi conforté dans mon intérêt pour le développement web, le fait de travailler d'une manière beaucoup plus libre m'a convaincu de rester dans ce domaine.

Curieux d'acquérir de nouvelles compétences dans ce domaine, il me tarde de concrétiser d'autres projets de ce type et bien plus encore.

Merci de m'avoir lu et d'avoir porter un intérêt certains dans ce projet

*Hugo Toumi & Stéphane Mathieu*