| **NOM** | PLOUZEAU |
| --- | --- |
| **Prénom** | Alexandre |
| **Date de naissance** | 17/09/1987 |



**Copie à rendre**

**TP Développeur Web et Web Mobile**

**Documents à compléter et à rendre**

**Dépot github :** [**https://github.com/Eyola/ECF-Garage.git**](https://github.com/Eyola/ECF-Garage.git)

**Ensemble des fichiers du site**

**Fichier readme pour l’installation en environnement local**

**Dossier “Annexes” pour les fichiers de documentations technique**



**Activité – Type 1 : Développer la partie front-end d’une application web ou web mobile en intégrant les recommandations de sécurité**

Application de type site web, avec pour objectif de faire connaître le garage de M. Parrot.

Le site doit présenter le garage, ses activités, les valeurs de la direction, et permettre un accès aux membres de l’entreprise pour quelques fonctionnalités de mise à jour. Le site doit également permettre de poster des commentaires d’utilisateurs.

De nos jours, un site se doit d’être “responsive”, c'est-à -dire être visitable autant sur ordinateur que sur smartphone ou tablette. Le développement a donc débuté sur une version mobile du site.

Tout d’abord, les pages importantes :

* Une page d’accueil :
  + Divers lien vers les différents services
  + Mise en avant de la vente de voiture d’occasion (activité croissante du client)
  + Mise en avant des avis des clients postés sur le site
  + Un lien vers une page de connexion pour les employés
  + Un pied de page avec les différentes informations utiles comme les horaires, un lien vers un plan d'accès, les réseaux sociaux etc…
* Une page vers les voitures en vente actuellement :
  + La liste complète des véhicules
  + Des filtres afin que les utilisateurs puissent trouver plus facilement le véhicule qui leur correspond suivant plusieurs critères (marques, modèle, prix, kilométrage, etc…)
  + Des tuiles représentant un aperçu de chaque véhicule avec le prix, une photo, l’année de mise en circulation, le kilométrage (demande précise du client)
* Une page listant les différents services :
  + Réparation
  + Entretien
  + Vente de voiture
* Une page présentant les avis des clients :
  + Lire les commentaires
  + Poster des commentaires
* Un formulaire de contact
* Une page présentant les valeurs de l’entreprise
* Un formulaire de connexion

Ensuite, une partie accessible uniquement accessible aux membres de l’entreprise :

* Gestion des véhicules :
  + Ajouter un nouveau véhicule
  + Modifier un véhicule (information erronée lors de l’enregistrement, modification de prix…)
  + Supprimer un véhicule (un véhicule vendu a peu d'intérêt à être toujours affiché)
* Gestion des commentaires :
  + Les commentaires ont tous un statut “en attente” lors de leur envoi. Un membre de l’entreprise doit le valider pour qu’il apparaisse sur le site. Le but étant d’éviter les messages haineux, ou les publicités frauduleuses.
  + Les membres de l’entreprise peuvent également poster directement des commentaires de leur clients
* Gestion du personnel :
  + accessible uniquement à M Parrot
  + Ajout de nouveaux utilisateurs
  + Affichage des utilisateurs enregistré
  + Suppression d’utilisateurs

Les premières maquettes sont présentées en annexe du dossier, ainsi que la charte graphique.

Une fois le code HTML/CSS, passage sur le site <https://validator.w3.org/> afin de valider l’écriture du code HTML.



**Activité – Type 2 : Développer la partie back-end d’une application web ou web mobile en intégrant les recommandations de sécurité**

La partie BACK-END fut développée sous PHP.

Une base de données locale a été créée via requêtes SQL vers un SGBD MariaDB.

Le logiciel XAMPP fut utilisé pour développer en local, via un virtualHost.

Modèle d’architecture MVC pour organiser les fichiers.

Tout d’abord, il a fallu définir les différentes class à créer ainsi que leurs attributs.

Le diagramme est disponible en annexe.

Une fois ces class définies, création des différentes tables, colonnes, et création de l’admin via requêtes SQL.

Définition des class dans le model PHP, avec attributs privés, définitions de méthodes getters/setters pour pouvoir les utiliser.

Chaque class devant interagir avec la base donnée, une class supplémentaire “PDOServer” fut créée et étendue à toutes les class afin de limiter la duplication de code.

Création de formulaires afin d'interagir avec la base de données via une interface graphique. Tous les formulaires sont créés via la méthode “POST” afin de ne pas afficher d'information sensible dans les URL. Les champs html de mots de passe et mail sont respectivement de type “password” et “mail” afin d’effectuer une première vérification de sécurité.

Il convient de réaliser des contrôles supplémentaires dans le controler du MVC pour accroître la sécurité :

* longueur de mot de passe, utilisation de caractères spéciaux ainsi que de chiffres…
* Méthode d’envoi de type “POST”
* Echappement de caractères spéciaux pour empêcher l’injection de code PHP
* Vérification que l’utilisateur s’est authentifié pour accéder à certaines page via un cookie de connexion

Création des méthodes permettant de récupérer les informations de formulaire et les envoyer à la base de données via requêtes SQL générées par PHP en passant par PDO.

Création des méthodes permettant de récupérer les données de la base pour les afficher (voitures et utilisateurs disposant d’un compte).

Création de filtres permettant d’interroger la base de données pour afficher les voitures suivant différents critères. Les filtres eux-mêmes sont générés via PHP :

* Interrogation de la base de donnée pour récupérer les valeurs de chaque colonne
* Génération d’un code HTML pour chaque valeur de filtre disponible (Marque, Modèle, Année de mise en circulation, Kilométrage, Prix)

Génération d’un tableau afin d’afficher la liste des utilisateurs existant (en tant qu’admin), et la possibilité de les supprimer (personne ayant quitté l’entreprise).

Récupération des commentaires postés en attente d’approbation, afin de changer leur statut et pouvoir les afficher dans la section prévue à cet effet, via la création de requêtes SQL, les commentaires étant stockés en base de données.

**Partie 3 :**  **Intitulé de la partie**