NOTICE

ECF Graduate développeur front-end / Garage Parrot.

Sommaire

1) Informations sur le déroulement de l'examen.

Madame, Monsieur,

La conception et le développement de cette application à été réalisée suivant le protocole demandé sur l'énoncé qui m'a été fourni.

A ce jour (c'est à dire en date butoir du 21 septembre 2023), le développement de l'application est inachevé en raison d'une mauvaise gestion de mon temps.

Cette dernière peut être visitée car la partie front-end est en grande partie fonctionnelle, mais la partie back-end est en cours de montage.

De ce fait, certains boutons, liens ou fonctionnalités sont inopérantes. Il en va de même pour les «annonces automobile» dont la vitrine n'est pas importée de la base de données , mais un simple code HTML qui n'est pas modifiable.

Toutefois, j'explique la méthode de développement de l'application dans les paragraphes ci-dessous.

2) Spécifications techniques.

Serveur utilisé:

- MariaDB (version 10.4.28)
- Extension PHP: PDO

Langage Front-end :

- HTML 5
- CSS
- Bootstrap
- JavaScript

Langage Back-end :

- PHP
- MySQL

Nota : L'application fonctionne avec une base de données MySQL via phpMyAdmin, cependant le langage SQL utilisé n'est pas généré par celui-ci. Les scripts SQL sont visibles sur le fichier **ECF/Data/database/database.sql**

3) Création d'un administrateur. (en partie fonctionnelle)

La création d'un administrateur se fait via la page **CONNECTION** de l'application.
En partant du principe qu'il n'y a qu'un seul administrateur, l'identifiant et le mot de passe sont fournis dés la livraison de l'application, l'administrateur pourra ensuite les modifier s'il le désire.

Identifiant : Admin
Mot de passe : admin01

Lors de sa connexion validée, l'application indique un message de bienvenue. L'administrateur pourra alors créer un nouvel utilisateur (employé) qui pourra se connecter à son tour avec un identifiant et un mot de passe.

Table **Admin** (voir diagramme de classe).

L'administrateur verra lors de sa connexion, apparaître des boutons spécifiques sur chaque page, lui permettant de modifier les paramètres de son site, a savoir :

- Modifier les horaires d'ouverture de chaque service ainsi que l'adresse.
- Modifier les types de prestations proposées par le service atelier.
- Créer / supprimer un utilisateur

- Créer / supprimer un avis utilisateur
- Créer / supprimer une annonce

Lorsque l'administrateur ou un utilisateur employé est connecté, l'onglet **CONNECTION** dans la barre de navigation devient **DECONNECTION** quelle que soit la page sur laquelle on se trouve pour pouvoir se déconnecter à tout moment.

4) Création d'un utilisateur employé (non-fonctionnel)

La création d'un utilisateur employé ne peut se faire que par l'administrateur lui-même. Une fois connecté, ce dernier doit cliquer sur le lien **Ajouter un utilisateur** , qui lui permettra à travers une fenêtre modale de définir un nom et prénom, un identifiant ainsi qu'un mot de passe. Une notification sera envoyée par mail à l'utilisateur si la checkbox correspondante à été cochée.

Table **Employee** (voir diagramme de classe).

L'utilisateur employé verra lui aussi lors de sa connexion, apparaître des boutons spécifiques sur chaque page, lui permettant de modifier les paramètres du site, a savoir :

- Créer / supprimer une annonce
- Créer / supprimer un avis utilisateur
- Consulter / supprimer les demandes de rendez-vous commerciaux
- Consulter / supprimer les demandes de rendez-vous atelier

De la même manière, l'utilisateur employé ou l'administrateur pourra supprimer le compte souhaité une fois connecté dessus avec le lien **Supprimer un utilisateur**.

5) Fonctionnalités (en partie fonctionnels)

a) L'utilisateur-employé peut :

Créer une annonce sur la page **OCCASIONS** en cliquant sur le bouton **Créer une annonce**. Cette action redirige la demande sur une page PHP contenant un formulaire et un aperçu de l'annonce souhaitée. Si celle-ci convient, elle pourra être mise en ligne avec le bouton **Publier**.

Supprimer une annonce en cliquant sur le lien Supprimer de l'annonce intéressée. Ou bien cliquer sur le bouton **Supprimer une annonce** qui ouvrira une fenêtre modale lui demandant l'Id de l'annonce à supprimer.

Créer un avis sur la page **ACCUEIL** au même titre qu'un internaute lambda en cliquant sur le bouton **Donner un avis**. Cette action ouvre une fenêtre modale sur laquelle on peut remplir un formulaire demandant un nom, un commentaire et une note, puis **Envoyer**.
Les données sont ensuite envoyé sur une table spécifique en base de données **UserReview** (voir diagramme de classe) et une notification est envoyée par mail aux utilisateurs employés avec un lien pour consulter l'avis et le valider (ou le supprimer) si nécessaire.

Supprimer un avis en cliquant sur le lien Supprimer de l'avis intéressé. Ou bien cliquer sur le bouton **Supprimer un avis** qui ouvrira une fenêtre modale lui demandant l'Id de l'avis à supprimer.

Consulter les demandes de rendez-vous commerciaux et atelier, par la simple réception d'une **notification par mail** donnant un lien pour voir le contenu de la table **WorkShop** (voir diagramme de classe).

b) L'administrateur peut en plus :

Modifier les horaires d'ouverture de chaque service ainsi que l'adresse sur n'importe quelle page du site en cliquant au niveau du bas de page sur le bouton **Modifier**.
Cette action permet d'ouvrir une page PHP permettant de communiquer avec la base de données et de modifier le contenu de la table **ServicesSchedules** (voir diagramme de classe) ainsi que **SalesSchedules** (voir diagramme de classe). De ce fait les horaires sont remis à jour.

Modifier les types de prestations proposées par le service atelier sur la page **SERVICES** en cliquant sur le bouton **Modifier** au niveau des prestations atelier.
Cette action permet d'ouvrir une page PHP permettant de communiquer avec la base de données et de modifier le contenu de la table **WorkshopServices** (voir diagramme de classe), de ce fait les prestations sont remises à jour.

c) L'utilisateur internaute peut :

Naviguer sur le site et découvrir les différents services de l'établissement en cliquant sur les différents onglets de la barre de navigation : **ACCUEIL**, **SERVICES** ou **OCCASIONS**.

Rechercher un véhicule d'occasion en le filtrant sur la page **OCCASIONS** au niveau du haut de page dans l'encart **Filtre**. L'utilisateur doit sélectionner par des boutons "range" une tranche de prix, une tranche d'année de mise en circulation, et une tranche kilométrique. L'utilisateur doit ensuite cliquer sur **Rechercher** pour trier sa recherche et faire apparaître à l'écran les véhicules correspondant a son filtrage (s'il y en a...).
Une requête est alors envoyée à la base de données à la table **UsedCars** (voir diagramme

de classe) qui, en outre est utilisée pour afficher l'ensemble des annonces, pour filtrer les véhicules correspondant et n'afficher que ces derniers.

Demander un rendez-vous avec un commercial pour avoir plus d'informations sur un véhicule, en cliquant sur Contacter sur l'annonce d'un véhicule, ou sur le bouton **Contacter un conseiller** situé au bas des annonces.
Cette action ouvre une fenêtre modale contenant un formulaire demandant un nom, un prénom, un numéro de téléphone, un e-mail et un message.
Les informations saisies sont transmises à un employé par mail et sauvegardées en base de donnée sur la table **User** (voir diagramme de classe).

Créer un avis sur la page **ACCUEIL** en cliquant sur le bouton **Donner un avis**.
Cette action ouvre une fenêtre modale sur laquelle on peut remplir un formulaire demandant un nom, un commentaire et une note, puis **Envoyer**.
Les données sont ensuite envoyé sur une table spécifique en base de données **UserReview** (voir diagramme de classe) et une notification est envoyée par mail aux utilisateurs employés avec un lien pour consulter l'avis et le valider (ou le supprimer) si nécessaire.

Rappel : *Par contrainte de temps, je n'ai pas terminé le développement de toutes les fonctionnalités, par conséquent certains boutons sont inactifs et certaines demandes n'aboutissent pas.*

6) Conception du projet

a) Réflexion initiale.

A la lecture de l'énoncé, ma réflexion initiale s'est portée sur le temps et l'organisation nécessaire pour mener à bien le projet tout en assurant les contraintes familiales quotidiennes.
J'ai donc opté pour une gestion du travail sur Trello.
De cette manière, j'ai pu créer un tableau kanban me permettant de visualiser chaque tache, et de les dispatcher sur l'ensemble du temps que j'ai alloué au projet.

Voici le lien pour accéder au tableau des taches :
https://trello.com/b/kevkzNQo/ecf-garage-parrot

b) configuration de l'environnement de travail.

Je travaille sur PC portable équipé de Windows 11.
Mon éditeur de code favori est Visual Studio Code que j'utiliserai durant toute la phase de

développement. Il possède les extensions Prettier, Live-Server, PHP Intelephense et Twig.

Afin de mettre en place un serveur web local qui me permettra de visualiser l'interface utilisateur en php, j'ai installé le logiciel XAMPP qui me permet en outre d'acceder à une base de données MySQL.

Pour la sauvegarde et la publication du projet, j'ai crée sur GitHub un repository que j'ai ensuite clôné grâce à "**`git clone [url]`**" sur le repertoire **c:/xampp/htdocs** de mon disque dur.

Pour l'étude et la conception des différents diagrammes de cas d'utilisation, de séquences, et de classes, j'ai opté pour le l'outil UML Visual Paradigm en version online.
(la version d'essai de 30 jours est suffisante pour les travaux que j'ai besoin d'effectuer)

L'UX Design de l'application sera effectué sur le logiciel Adobe XD. En passant par l'étude (zoning), la mise en place (wireframes) et le maquettage (Mockups).
Les differentes phases seront appliquées en format desktop et mobile.

La partie rédigée que vous lisez actuellement sera saisie sur Apache OpenOffice puis convertie en format PDF pour les besoins des correcteurs.

c) Les diagrammes

Le diagramme de cas d'utilisation est visible en format PDF sur :
ECF/Tech/diagrams/Garage-Parrot – Cas d'utilisation.pdf

Les diagrammes de séquence décris dans l'énoncé sont visible en format PDF sur :
ECF/Tech/diagrams/Garage-Parrot – Séquence-(1-4).pdf
(il y a 4 diagrammes, soit 3 alternatives d'erreurs informatiques)

Le diagramme de classe est visible en format PDF sur :
ECF/Tech/diagrams/Garage-Parrot – Diagramme de classe.pdf

*(Chacun des diagrammes est également proposé en format **".vp"** si vou souhaitez directement les ouvrir avec VisualParadigm.)*

d) La charte graphique

La charte graphique se compose de deux parties :
-La police et la palette couleur (Fonts and Path)
-Les images et les icônes utilisées (Pictures and icons)

La police et la palette couleur sont visibles en format PDF sur :
ECF/Tech/graphicChart/Fonts_and_path.pdf

Les images et les icônes utilisées sont visibles en format PDF sur :
ECF/Tech/graphicChart/Pictures_and_icons.pdf

(Ces images et icônes présentes dans la charte graphique se retrouveront dans la banque d'image qui serviront au développement du front-end au niveau du dossier : ***ECF/UI/img****)*

e) Les Wireframes

Les Wireframes sont décomposés en plusieurs fichiers PDF mais sont aussi visibles sur un fichier **"Wireframes.xd"** si l'on souhaite les visualiser sur un seul et même fichier sur le logiciel Adobe XD.

On trouve ces fichiers sur :
ECF/Tech/wireframes/[Tous les fichiers PDF]

Ou :
ECF/Tech/wireframes/Wireframes.xd *(ou* **Wireframes.pdf** *si l'on souhaite voir tous les fichiers ensemble en PDF)*

7) Les scripts SQL

Les différentes requètes SQL qui permettront de générer la base de donnée locale sont inscrites dans le répertoire **ECF/Data/database.sql**

Ces scripts à proprement parler n'ont aucune utilité directe dans ce repertoire, ils seront utilisés et interprêtés dans les differentes fonctionnalités de l'application.

Le module permettant de faire la liaison entre la base de données SQL et l'application utilise l'objet PDO, il se trouve dans le repertoire : **ECF/Data/connexion.php**

contenu :

```php
<?php

try {
  $db = new PDO('mysql:host=localhost;dbname=garage_parrot', 'root', "");
} catch (PDOException $e) {
  die('Erreur connexion : ' . $e->getMessage());
}
```

8) Dépot GitHub

Le dépôt Git de l'application est accessible sur l'URL suivante :
https://github.com/Stephane-SERRA/ECF.git

L'application sera mise en ligne grâce à : **Netlify**

9) Conclusion

En conclusion, l'Evaluation en Cours de Formation qui m'a été soumise m'a pris autour de 100h de travail, dont 60h pour la conception (reflexion, diagrammes, maquettes, charte graphique), 10h pour la mise au points des documents de présentation (mise au point du planning Trello, rédaction de la notice) et environ 30h pour en arriver au niveau de développement de l'application, tel qu'il est au moment de la livraison du projet.

Mes difficultés au niveau du développement se sont ressenties au niveau de la maitrise de JavaScript qui m'on amené a effectuer beaucoup de recherches sur des sites tels que MDN ou differents forums de developpeurs qui ne m'amenaient pas toujours les reponses souhaitées. J'ai du faire face à beaucoup d'échecs de fonctionnement dont une partie seulement ont pu être résolus.

Le retard causé par le temps passé sur la résolution des problèmes en amont se sont répercutés sur la suite du code, par conséquent, je n'ai pas encore pu mettre au point toutes les fonctionnalités liés avec la communication en base de données utilisant le langage PHP.

Sur un plan professionnel, si je devais estimer le temps restant pour que l'application puisse être pleinement opérationnelle, je donnerais un délai d'environ deux a trois semaines de travail.

Je laisse à ce jour à votre appréciation la notation du travail effectué et vous remercie par avance.

Bien cordialement.

Stéphane SERRA
Apprenant Studi