NOM	RECHES
Prénom	Loïc
Date de naissance	27 Novembre 1996

Copie à rendre

TP – Développeur Web et Web Mobile

Documents à compléter et à rendre

Lien du git : https://github.com/Horkii/ECF-projet-coRide-STUDI
Lien de l'outil de gestion de projet : https://www.notion.so/1f371d04fff2805eb858db47b0dce376?v=1f371d04fff281a4ba69000c2c75c9af&pvs=4
Lien du déploiement :
Login et mot de passe administrateur :

SANS CES ELEMENTS, VOTRE COPIE SERA REJETEE

Partie 1 : Analyse des besoins

1. **Effectuez un résumé du projet en français d'une longueur d'environ 20 lignes soit 200 à 250 mots**

J'ai été contacté par José pour réaliser le projet de sa startup « Ecoride » qui a pour objectif de réduire l'impact environnemental des déplacements en encourageant le covoiturage.
Ainsi, à travers les pages de l'application, José souhaite différentes interactivités.
Une page d'accueil qui transmet les idées et le message de la startup par rapport à l'écologie et le covoiturage, une navigation fluide et facile d'utilisation pour les internautes, une page de recherche de trajet dynamique avec la possibilité de filtrer et de classer ses recherches et des détails pour chaque trajet.
José aimerait que chaque utilisateur de son application web puisse créer un compte afin d'avoir un espace personnalisé avec ses propres données (en adéquation avec les lois RGPD). L'utilisateur doit pouvoir choisir s'il veut être passager ou chauffeur, pour mettre en place les trajets de son choix avec ses caractéristiques.
Il souhaite également avoir un compte administrateur pour avoir les données relatives à l'utilisation de son application (trajets effectués, crédits gagnés par la plateforme...) et pouvoir créer les comptes employés qui vont pouvoir modérer les commentaires et les notes des utilisateurs.
José espère susciter l'envie de covoiturer afin de réduire l'emprunte carbone de chacun et sauver la planète.

2. **Exprimez le cahier des charges, l'expression du besoin ou les spécifications fonctionnelles du projet**

=> US 1 Page d'accueil
=> US 2 Menu nav barre
=> US 3 Recherche de trajets
=> US 4 Filtres
=> US 5 Détails trajet
=> US 6 Participer au trajet
=> US 7 Création de compte
=> US 8 Espace User
=> US 9 Ajout de trajet
=> US 10 Historique de trajet
=> US 11 Bouton Démarrer/Arrêter trajet
=> US 12 Espace employé
=> US 13 Espace Admin

Partie 2 : Spécifications technique

1. **Spécifiez les technologies que vous avez utilisé en justifiant les conditions d'utilisation et pourquoi le choix de ses éléments**

J'ai choisi d'utiliser Symfony car il me semblait plus facile à utiliser pour un projet d'une telle envergure, de part sa gestion du projet et sa face technique simplifiée, notamment grâce à Fetch intégré pour Javascript mais aussi d'un point de vue sécurité avec Doctrine, l'ORM de Symfony, à travers les requêtes DQL et le QueryBuilder.
Avec Composer, j'ai pu rajouter des extensions à PHP pour lier les BDD de l'application et des extensions pour les e-mails d'inscriptions. J'ai justement choisi MongoDB et PostgreSQL pour leur adéquation avec Symfony et les extensions existantes pour PHP. J'ai installé Docker pour le déploiement avec Fly.io qui est gratuit contrairement à Heroku.

2. **Comment avez-vous mis en place votre environnement de travail ? Justifiez vos choix. (README.md)**

   J'ai d'abord installé PHP avec sa dernière version et les extensions nécessaires au bon fonctionnement des BDD. Ensuite, j'ai installé Symfony et Symfony CLI pour avoir un serveur en local. Puis j'ai installé MongoDB et PostgreSQL avec leur desktop (respectivement MongoDB Compass et pgAdmin4). Pour ce qui est de Git, je l'avais déjà téléchargé pour le cours sur Git.

3. **Énumérez les mécanismes de sécurité que vous avez mis en place, aussi bien sur vos formulaires que sur les composants front-end ainsi que back-end.**

   Lors de la création de mon projet, j'ai pour l'instant choisi mettre le fichier .env en gitignore pour protéger ces informations dans mon dépôt git publique.
   Si j'avais fini mon application, j'aurais mis en place la sécurité des champs avec un nombre de caractères limités ainsi que l'interdiction de caractères spéciaux pour éviter les injection SQL, même si Symfony à été choisi en partie pour cela (Doctrine ORM).
   J'aurais également limité le nombre de véhicules par compte à cinq afin d'éviter le DDoS et rajouter du temps au formulaire de connexion et d'inscription (2s d'attente avant le lancement de la requête pour éviter le brut force).
   J'ai également essayé de mettre en place le hash des mots de passe en BDD.

4. **Décrivez une veille technologique que vous avez effectuée, sur les vulnérabilités de sécurité.**

   Pour ma veille technologique, je me suis inscrit à Medium qui me permet d'avoir tous les jours un « Medium Daily digest » et ainsi de surveiller tous les sujets qui sont importants pour moi, grâce à des tags. Dans l'un d'eux, j'ai reçu un article sur « PaswordHasher » de Symfony. Cet article, rédiger par une personne de la communauté (ici, une personne se décrivant comme un expert informatique, passionné de technologie et père de quatre enfants), explique comment fonctionne « Pasword Hashing » et pourquoi l'utiliser. Cela permet de comprendre qu'il faut « hasher » les mots de passe lors de leur stockage en base de données afin de sécuriser les comptes et respecter le RGPD.

Partie 3 : Recherche

1. **Décrivez une situation de travail ayant nécessité une recherche durant le projet à partir de site anglophone. N'oubliez pas de citer la source**

   J'ai dû, lors de la mise en place de mon environnement, chercher une solution par rapport à Composer sur Symfony qui semblait avoir un problème de certificat SSL. Après différentes solutions essayées (self-update, téléchargement d'un nouveau certificat, etc) je suis allé sur le site proposé par Composer lui-même, https://getcomposer.org/doc/articles/troubleshooting.md#ssl-certificate-problem-unable-to-get-local-issuer-certificate, une fois traduit, j'ai pu comprendre qu'il s'agissait d'Avast qui bloque Composer.

2. **Mentionnez l'extrait du site anglophone qui vous a aidé dans la question précédente en effectuant une traduction en français.**

   Voici l'extrait :

   « SSL certificate problem: unable to get local issuer certificate#

1. *Check that your root certificate store / CA bundle is up to date. Run composer diagnose -vvv and look for Checked CA file ... or Checked directory ... lines in the first lines of output. This will show you where Composer is looking for a CA bundle. You can get a new cacert.pem from cURL and store it there.*
2. *If this did not help despite Composer finding a valid CA bundle, try disabling your antivirus and firewall software to see if that helps. We have seen issues where Avast on Windows for example would prevent Composer from functioning correctly. To disable the HTTPS scanning in Avast you can go in "Protection > Core Shields > Web Shield > uncheck Enable HTTPS scanning". If this helps you should report it to the software vendor so they can hopefully improve things. »*

   Et voici la traduction :

   « Problème de certificate SSL : incapable d'obtenir le certificat de l'émetteur local.

   1. *Vérifier si votre dossier de certificats racines ou le bundle CA est à jour. Lancer « composer diagnose -vvv » et chercher les lignes « Checked CA file… » ou « Checked directory… » au début du résultat. Cela nous dira où Composer cherche le bundle de certificats.*
      Vous pouvez télécharger un nouveau fichier « cacert.pem » depuis cURL et le mettre à cet endroit.
   2. *Si cela ne fonctionne toujours pas, même si Composer trouve un bon bundle CA, essayer de désactiver votre antivirus et pare-feu pour voir si cela change quelque*

chose. Nous avons pu constater que des programmes comme Avast sur Windows peuvent empêcher Composer de fonctionner correctement. Pour désactiver le scan HTTPS dans Avast, aller dans « Protection > Agents de sécurité > Agent web », puis décocher cet agent.
Si cela règle votre souci, pensez à le dire à l'éditeur du logiciel, ainsi ils pourront peut-être corriger le problème. »

Partie 4 : Informations complémentaire

1. **Autres ressources**

2. **Informations complémentaires**