

NOM	MIMOUNI
Prénom	Taieb
Date de naissance	24/01/1990

Copie à rendre

TP – Développeur Web et Web Mobile

Documents à compléter et à rendre

Lien du git : https://github.com/REDEVILS24/ECF_ECORIDE

Lien de l'outil de gestion de projet : <https://trello.com/b/nBO2xQIU/ecoride>

Lien du déploiement :

<https://ecf-ecoride-mimounitaieb-87e15deda74d.herokuapp.com/templates/home/index.php>

Login et mot de passe administrateur : admin@ecoride.com / Admin123!

**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.

Nous rencontrons José, le directeur technique de la société "Ecoride", cette dernière a pour vocation de rendre les déplacements écoresponsables en réduisant l'empreinte carbone des covoiturages. La création de l'application web qui nous a été mandatée par José se doit d'être une plateforme à la hauteur des ambitions d'Ecoride. La clientèle visée apporte une importance capitale aux déplacements en voitures et est soucieuse de l'environnement sans oublier une solution économique.

C'est à partir de là que José m'a choisi pour développer l'application web.

Il doit y avoir trois types d'utilisateurs : les visiteurs, les utilisateurs (conducteur, passager) et les employés (admin).

Le site doit comporter un accueil avec redirection vers cette dernière, un accès aux covoiturages, connexion et contact.

Pour les visiteurs, le covoiturage ne doit pas être visible par défaut, ils doivent remplir un formulaire pour afficher les covoiturages si disponibles, alors les informations du covoiturage et du conducteur doivent apparaître.

Des filtres doivent être mis en avant si besoin, en particulier pour l'énergie.

Une fois le covoiturage possible, si certaines conditions sont remplies, le visiteur pourra s'inscrire pour choisir son covoiturage.

La création se fera à l'aide d'un formulaire : pseudo, mail et mot de passe sécurisé seront demandés pour l'inscription. L'utilisateur bénéficiera de 20 crédits à l'inscription.

L'utilisateur depuis son profil pourra choisir s'il est chauffeur ou passager ou les deux. En devenant chauffeur, il doit obligatoirement fournir des informations sur son véhicule. Le système de réservation permet aux passagers de réserver des trajets en échange de crédits, tandis que les conducteurs en gagnent. L'application a été développée avec PHP 8, MySQL, MongoDB et Docker pour assurer une architecture robuste et moderne.

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

Objectifs :

Développer une plateforme de covoiturage éco-responsable favorisant les véhicules électriques, offrant une solution économique et écologique aux utilisateurs.

Acteurs :

Trois types d'utilisateurs définis

- Visiteurs (consultation via formulaire)
- Utilisateurs inscrits (conducteurs/passagers)
- Administrateurs (gestion globale).

Fonctionnalités principales :

Système d'authentification sécurisé avec attribution de 20 crédits à l'inscription

Gestion des trajets : création, modification, recherche avec filtres (énergie, prix, date)

Système de réservation avec déduction automatique des crédits

Gestion des véhicules avec obligation de déclaration pour conducteurs

Règles métier :

Système de crédits (gain pour conducteurs, paiement pour passagers), priorité aux véhicules électriques, vérification des places disponibles.

Exigences techniques : Interface responsive, authentification sécurisée, protection des données (RGPD), validation des entrées, architecture modulaire avec APIs RESTful.

UX/UI :

Design éco-responsable, navigation adaptative selon statut de connexion, temps de chargement < 3 secondes, compatibilité navigateurs modernes.

Contraintes :

Support 100+ utilisateurs simultanés, sécurité renforcée contre attaques, conformité légale des données personnelles.

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

HTML5 / CSS3 JavaScript Vanilla (interactivité et appels API) :

J'ai utilisé ces technologies car HTML5/CSS3 sont l'image d'un site web et constituent les technologies de base à utiliser pour créer une application web.

Quant au JavaScript, il permet la dynamisation du front avec une performance maximale et un contrôle total du code sans dépendance externe. J'ai choisi de le faire en Vanilla, donc de ne pas utiliser Bootstrap ni Tailwind comme frameworks, car je voulais avoir de bonnes bases et bien comprendre ce que je faisais sans vouloir gagner du temps pour mes débuts en formation. Je comprends mieux la manipulation du DOM de cette manière.

BACKEND :

PHP 8 (logique serveur et APIs) / PDO (accès sécurisé aux bases de données) :

J'ai opté pour PHP 8 car c'est un langage spécialement conçu pour le développement web avec une syntaxe accessible et un écosystème mature. Il offre une excellente performance pour les applications web et une facilité d'hébergement sur la plupart des serveurs. PHP 8 apporte également des améliorations significatives en termes de vitesse et de nouvelles fonctionnalités.

Concernant PDO, j'ai fait ce choix pour sécuriser mes accès aux bases de données. Cette extension me permet d'utiliser des requêtes préparées qui protègent efficacement contre les injections SQL. De plus, PDO offre une abstraction de la base de données, ce qui me permet de changer de SGBD sans modifier profondément mon code. En tant que débutant, cela me rassure d'avoir cette couche de sécurité supplémentaire pour gérer les données sensibles des utilisateurs.

BASES DE DONNÉES :

MySQL 8.0 (données relationnelles : utilisateurs, trajets, réservations) / MongoDB 5.0 (données NoSQL : voitures, avis, préférences) :

J'ai choisi une approche hybride avec deux types de bases de données pour répondre aux différents besoins du projet. MySQL 8.0 pour gérer les données relationnelles car il offre une structure stable et fiable pour les informations critiques comme les utilisateurs, trajets et réservations qui nécessitent des relations complexes et des contraintes d'intégrité.

Pour MongoDB 5.0, j'ai opté pour cette solution NoSQL afin de gérer des données plus flexibles comme les voitures, avis et préférences utilisateurs. Cette approche me permet d'adapter facilement la structure des données sans migration complexe et d'apprendre les deux paradigmes de bases de données, ce qui enrichit ma formation.

SERVEUR ET INFRASTRUCTURE :

Apache (serveur web) / Docker (conteneurisation) / Docker Compose (orchestration des services) :

Apache reste un choix sûr et éprouvé pour servir des applications PHP avec une configuration simple et une documentation abondante. En tant qu'étudiant, j'ai privilégié la stabilité et la facilité de mise en œuvre.

Docker et Docker Compose m'ont permis de créer un environnement de développement reproductible et isolé. Cette approche me garantit que mon application fonctionne de manière identique sur tous les environnements, ce qui évite les problèmes de "ça marche sur ma machine". C'est également une compétence très recherchée en entreprise que je voulais acquérir.

OUTILS DE DÉVELOPPEMENT :

Composer (gestionnaire de dépendances PHP) / Git (gestion de versions) / GitHub (hébergement du code) :

Composer est devenu incontournable dans l'écosystème PHP pour gérer les dépendances de manière automatisée et professionnelle. Cela m'évite de télécharger et gérer manuellement les bibliothèques.

Git et GitHub constituent des outils essentiels dans le métier de développeur. J'ai voulu dès le début adopter les bonnes pratiques de versioning et de sauvegarde de code. GitHub me permet également de montrer mon travail et mes compétences aux futurs employeurs avec un portfolio visible.

INTERFACES D'ADMINISTRATION :

PhpMyAdmin (interface MySQL) / MongoDB Express (interface MongoDB) :

Ces interfaces graphiques me facilitent grandement le travail de développement et de débogage. Plutôt que d'utiliser uniquement les lignes de commande, ces outils me permettent

de visualiser mes données, tester mes requêtes et comprendre la structure de mes bases de données de manière intuitive, ce qui accélère mon apprentissage.

GESTION DE PROJET :

Trello (organisation des tâches) :

J'ai utilisé Trello pour organiser mon travail selon une méthodologie agile avec des colonnes "À faire", "En cours" et "Terminé". Cette approche m'aide à structurer mon développement, suivre ma progression et respecter les délais. C'est aussi une compétence organisationnelle valorisée en entreprise.

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

J'ai mis en place mon environnement de travail en utilisant Docker et Docker Compose pour créer un environnement de développement isolé et reproductible. Cette approche me permet d'avoir exactement les mêmes versions de PHP, MySQL et MongoDB que mes collaborateurs ou les serveurs de production.

De plus, j'ai créé une architecture qui me permet de modifier n'importe quelle partie de mon code ou le déboguer sans impacter toute mon application.

Mon README.md détaille précisément les étapes d'installation car je voulais que n'importe qui puisse lancer mon projet facilement. J'ai inclus les prérequis, les commandes Docker et les accès aux différents services. Cette documentation rigoureuse fait partie des bonnes pratiques que j'ai voulu adopter dès ma formation. Cela me force également à bien comprendre chaque étape de mon setup et évite les problèmes lors du déploiement.

2. 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.

Sur le front-end, j'ai mis en place une validation des formulaires côté client pour améliorer l'expérience utilisateur et effectuer une première vérification des données. J'utilise également `textContent` au lieu de `innerHTML` pour afficher les données dynamiques afin d'éviter les attaques XSS.

Côté back-end, j'ai implémenté plusieurs couches de sécurité : utilisation de requêtes préparées avec PDO pour prévenir les injections SQL, hachage sécurisé des mots de passe avec `password_hash()` et `password_verify()`, gestion des sessions PHP pour l'authentification, et validation/nettoyage de toutes les données entrantes. J'ai également mis en place des vérifications d'autorisation pour m'assurer qu'un utilisateur ne peut accéder qu'à ses propres données.

Ces mesures me permettent de protéger les données sensibles des utilisateurs tout en apprenant les fondamentaux de la sécurité web.

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

Durant mon projet, j'ai effectué une veille sur les vulnérabilités liées aux applications web PHP, notamment en consultant les ressources OWASP (Open Web Application Security Project). J'ai particulièrement étudié le Top 10 OWASP 2021 qui liste les principales vulnérabilités des applications web.

Cette recherche m'a sensibilisé aux risques d'injection SQL, aux failles XSS (Cross-Site Scripting) et aux problèmes d'authentification défaillante. J'ai également consulté la documentation officielle de PHP sur les bonnes pratiques de sécurité et suivi des articles sur des sites spécialisés comme ANSSI pour comprendre l'évolution des menaces.

Cette veille m'a permis d'adapter mon code en conséquence et d'intégrer dès le développement les bonnes pratiques de sécurité plutôt que de les ajouter après coup.

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

Durant le développement du système de réservation, j'ai rencontré une difficulté avec la gestion simultanée des crédits utilisateurs et des places disponibles. Je devais m'assurer qu'aucune condition de course (race condition) ne puisse permettre à deux utilisateurs de réserver la dernière place disponible en même temps.

J'ai effectué des recherches sur les bonnes pratiques de gestion des transactions en PHP et MySQL. J'ai consulté la documentation officielle MySQL en anglais, notamment la section sur les "Transactions and Locking" qui explique comment utiliser les verrous pour éviter ces problèmes de concurrence.

Source : MySQL Official Documentation - <https://dev.mysql.com/doc/refman/8.0/en/innodb-locking-transaction-model.html>

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

Extrait anglais :

"InnoDB uses row-level locking. Only the rows that are being updated or deleted need to be locked, not the whole table. This reduces lock contention when multiple transactions are accessing different rows of the same table."

Traduction française :

"InnoDB utilise un verrouillage au niveau des lignes. Seules les lignes qui sont mises à jour ou supprimées doivent être verrouillées, pas toute la table. Cela réduit la contention des verrous lorsque plusieurs transactions accèdent à différentes lignes de la même table."

1. Autres ressources

Documentation officielle PHP 8

Tutoriels Docker pour développeurs

Guide des bonnes pratiques de sécurité web

2. Informations complémentaires

Ce projet m'a permis d'acquérir une vision complète du développement web full-stack et de comprendre l'importance de la sécurité dans les applications manipulant des données utilisateurs.