

EcoRide

DOCUMENTATION
TECHNIQUE

Table des matières

1. Introduction

- 1.1 Présentation d'EcoRide
- 1.2 Objectifs de l'application

2. Charte graphique

- 2.1 Palette de couleurs
- 2.2 Typographie

3. Maquettes (Wireframes et Mockups)

- 3.1 Maquettes version bureau
 - 3.1.1 Maquette de la page d'accueil
 - 3.1.2 Maquette de la vue des covoiturages
 - 3.1.3 Maquette des détails d'un covoiturage
- 3.2 Maquettes version mobile
 - 3.2.1 Maquette de la page d'accueil
 - 3.2.2 Maquette de la vue des covoiturages
 - 3.2.3 Maquette de l'espace utilisateur

4. Description technique de l'application

- 4.1 Choix techniques et technologies utilisées
- 4.2 Configuration de l'environnement de travail
- 4.3 Modèle conceptuel des données (ou diagramme de classe)
- 4.4 Schémas de la base de données (MCD relationnelle)

5. Fonctionnalités de l'application

- 5.1 Front-end (HTML5, CSS, JS)
 - 5.1.1 Développement des interfaces utilisateur statiques
 - 5.1.2 Développement des interfaces utilisateur dynamiques
- 5.2 Back-end (PHP)
 - 5.2.1 Développement des composants métiers côté serveur
 - 5.2.2 Développement des composants d'accès aux données (SQL)
- 5.3 Base de données
 - 5.3.1 Mise en place de la base de données relationnelle (MariaDB)

6. Sécurité

- 6.1 Mesures de sécurité de l'application
- 6.2 Justifications des choix techniques pour la sécurité

7. Gestion de projet

- 7.1 Organisation de la gestion de projet

- 7.2 Utilisation de Git et branches
 - 7.2.1 Organisation des branches (principale, développement, fonctionnalités)
 - 7.2.2 README.md et élaboration d'instructions de déploiement local
- 7.3 Tableau Kanban (Trello)
 - 7.3.1 Explication du tableau Kanban et répartition des tâches

8. Documentation du déploiement

- 8.1 Démarches pour déployer l'application
- 8.2 Outils de déploiement utilisés (Fly.io, Heroku, Azure, Vercel)
- 8.3 Instructions pour déployer localement

9. Remerciements

Introduction

Présentation d'EcoRide

EcoRide est un projet français de plateforme de covoiturage axé sur l'écologie permettant de mettre en relation des usagers souhaitant voyager ensemble afin de faire des économies, de réduire leur impact environnemental mais aussi de rencontrer d'autres voyageurs.

L'application permet aux visiteurs de consulter les voyages disponibles et d'en consulter les détails (date, trajet, conducteur...). Lorsque ces derniers sont connectés à l'application, ils ont la possibilité de choisir entre plusieurs rôles : celui de **conducteur**, de **passager** ou bien **les deux**.

Le rôle conducteur leur donne la possibilité de créer un voyage tandis que le rôle passager, en plus de pouvoir consulter la liste des voyages, permet de réserver une place au sein de ces derniers.

Quatre profils d'utilisateurs s'inscrivent dans le cadre d'usage de cette application web. Nous retrouvons le **visiteur** qui est un utilisateur ne disposant pas de compte utilisateur sur l'application. Il a la possibilité de consulter la liste de voyages mais ne peut réserver de place au sein de ceux-ci.

Quand le visiteur dispose d'un compte, il devient **utilisateur** et peut désormais sélectionner un ou plusieurs rôles (chauffeur et/ou passager).

Le profil **employé** donne accès à l'utilisateur à une liste exhaustive de fonctionnalités lui permettant d'effectuer des tâches de supervision / modération. Pour finir, le profil **administrateur** dispose des plus hauts privilèges d'administration lui permettant de gérer dans sa globalité la plateforme (avec par exemple la fonction d'ajout / suppression de compte employé).

Objectifs de l'application

EcoRide s'inscrit dans la mouvance des startups made in France dynamiques, innovantes et engagées dont l'objectif principal est de conjuguer progrès technologique et préservation de la planète. En mettant à disposition une solution simple, conviviale et respectueuse de l'environnement, la plateforme EcoRide se veut innovante. Pour parvenir à un service eco-friendly digne de ce nom, voici la liste de ses objectifs :

1. Encourager le covoiturage écoresponsable

L'application doit permettre aux utilisateurs de faire du covoiturage en priorisant l'utilisation de véhicules électriques et peu polluants.

2. Offrir une expérience fluide et intuitive à l'utilisateur

En proposant une interface épurée et intuitive, l'utilisateur doit pouvoir rechercher, filtrer et réserver un voyage en quelques clics seulement.

3. Faciliter la mise en relation des utilisateurs (conducteurs et passagers)

L'application doit permettre aux conducteurs d'enregistrer leurs trajets et aux passagers de réserver une place dans un véhicule. Cette interaction doit être rapide et sécurisée, avec des mécanismes de validation des transactions et de gestion des crédits.

4. Assurer la sécurité et la fiabilité des trajets

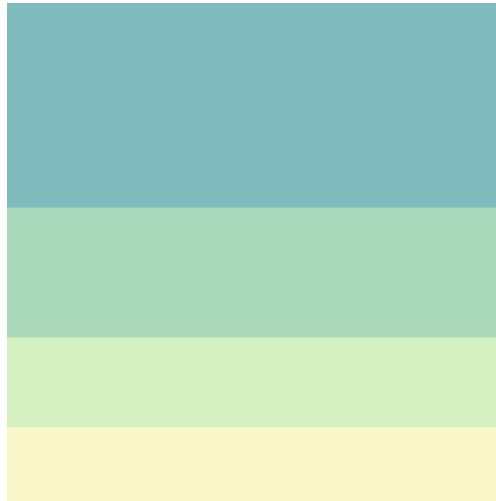
L'application doit inclure des fonctionnalités garantissant la sécurité des utilisateurs, comme la vérification des comptes (conducteurs et passagers), la notation des trajets, et un suivi des avis laissés après les covoiturages.

5. Offrir une solution évolutive et adaptable

Dans une perspective d'évolution dans le temps, d'une maintenabilité efficace mais surtout dans une soucis d'adaptation continu, l'application doit être conçue en dans le respect des normes et des conventions de développement les plus récentes.

Charte graphique

Palette de couleurs



Typographie

Les polices d'écritures utilisées sont : Segoe UI, "Noto Sans", "Source Sans 3", Arial, sans-serif;

Maquettes

Les maquettes ci-dessous illustrent l'organisation des pages principales d'EcoRide. Elles permettent de valider l'ergonomie et les fonctionnalités avant le développement. L'ensemble des maquettes (wireframes et mockups) a été réalisé avec Figma. Conscient de l'importance de maîtriser les standards professionnels actuels, j'ai choisi de développer en priorité la version mobile (**mobile first**), afin de garantir une expérience optimale sur les petits écrans avant d'adapter les interfaces au format desktop.

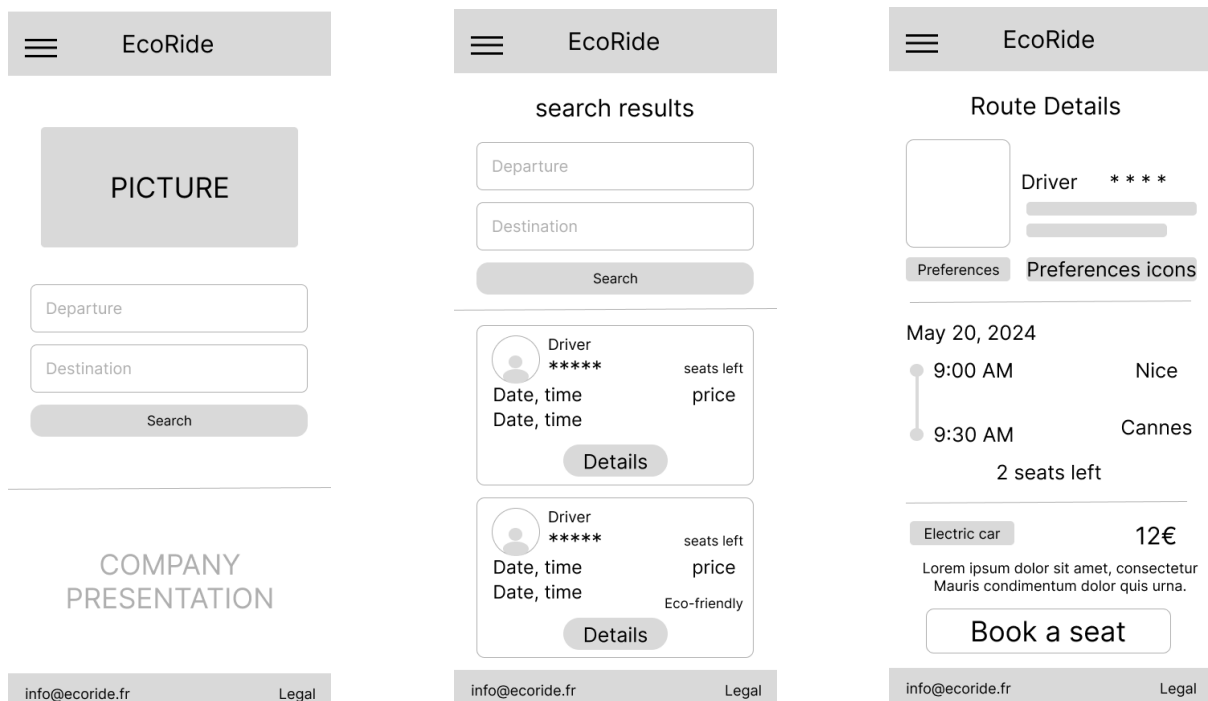
Lien Figma : <https://www.figma.com/design/H0Pn6DjPInGQKtK6YO6pSG/Ecoride?node-id=0-1&t=oyRYEYLzqG729awX-1>

Wireframes

Ces maquettes sont des wireframes basse fidélité, servant à valider l'organisation des éléments avant de passer à la création des maquettes graphiques finales.

Version mobile

Respectivement les pages : d'accueil, d'affichage des résultats de recherche et de détails de trajet



Version desktop non disponible

Description technique de l'application

Choix techniques et technologies utilisées

Le développement de l'application EcoRide s'est appuyé sur les technologies suivantes :

- **PHP** (version 8.x) : langage principal pour le développement back-end.
- **HTML5 / CSS3** : pour le rendu et la structuration des interfaces utilisateurs.
- **JavaScript Vanilla** : pour ajouter de l'interactivité côté client.
- **MariaDB** : gestionnaire de base de données relationnelle.
- **Git / GitHub** : pour la gestion de versions du projet.
- **Trello** : pour la gestion de projet (tableau Kanban).
- **PhpMyAdmin** : pour l'administration de la base relationnelle

Configuration de l'environnement de travail

L'environnement de développement a été configuré de la manière suivante :

- **IDE** : PhpStorm
- **Version PHP** : 8.3
- **Client Git** : Git Bash
- **Navigateur de test** : Google Chrome (mode responsive + devtools).
- **Hébergement final** : LWS.fr (MariaDB)

Modèle conceptuel des données (ou diagramme de classe)

Modèle conceptuel des données (ou diagramme de classe)

Le MCD modélise les entités suivantes :

- **Utilisateur** (utilisateur_id, prénom, nom, email, mot de passe, rôle, etc.)
- **Covoiturage** (covoiturage_id, conducteur_id, lieu départ, lieu arrivée, etc.)
- **Voiture** (voiture_id, modèle, énergie, couleur, etc.)
- **Avis** (passager_id, conducteur_id, note, commentaire, statut)
- **Réservation** (passager_id, covoiturage_id, nb_places, statut)

Les relations entre ces entités sont de type 1-N ou N-N (avec des tables de jointure).

Schémas de la base de données (MCD relationnelle)

Un script SQL complet est fourni dans le dossier /db/ecoride.sql. Il permet la création de toutes les tables, relations et contraintes (clés primaires/étrangères). Le modèle relationnel respecte les normes de conception et garantit la cohérence des données.

Fonctionnalités de l'application

Front-end (HTML5, CSS, JS)

Développement des interfaces utilisateur statiques

Des pages statiques telles que l'accueil, les mentions légales, ou les formulaires (connexion/inscription) ont été développées en HTML5 avec une mise en forme CSS responsive (mobile-first). Les composants sont organisés et réutilisables.

Développement des interfaces utilisateur dynamiques

Certaines pages (liste de trajets, page de réservation, recherche de covoiturages) intègrent JavaScript pour gérer l'interactivité et améliorer l'expérience utilisateur.

Back-end (PHP)

Développement des composants métiers côté serveur

Le projet utilise une architecture MVC (Model-View-Controller) simplifiée. Le contrôleur RideController gère les trajets, UserController les utilisateurs, etc. La logique métier est encapsulée dans les modèles (validation, calculs de durée, traitement des rôles...).

Développement des composants d'accès aux données (SQL)

Les requêtes SQL sont préparées avec PDO pour éviter les injections.

Base de données

Mise en place de la base de données relationnelle (MariaDB)

La base ecoride a été structurée avec soin, en respectant la normalisation des données, les contraintes d'intégrité, et les types adaptés. L'accès s'effectue via PDO et des requêtes sécurisées.

5.3.2 Gestion des données avec MongoDB (NoSQL)

MongoDB permet de stocker certains événements de l'application (logs, historiques de connexions). L'objectif est d'offrir une meilleure évolutivité pour certains types de données non structurées.

Sécurité

Mesures de sécurité de l'application

- **CSRF Token** : généré pour tous les formulaires critiques.
- **Vérification des rôles** : middleware checkConnect() et checkAdmin() utilisé.
- **Sanitization / validation** : des données côté serveur avec filter_var, htmlspecialchars, etc.
- **Hachage des mots de passe** : avec password_hash() et password_verify().

6.2 Justifications des choix techniques pour la sécurité

Le choix de sécuriser les formulaires avec des tokens CSRF protège des attaques cross-site.

L'utilisation de sessions pour stocker les données utilisateur est contrôlée et les identifiants ne sont jamais en clair. Le hachage garantit que les mots de passe restent confidentiels.

Gestion de projet

Organisation de la gestion de projet

Le projet a été découpé en **sprints** avec des objectifs clairs : base de données, modèles, vues, contrôleurs, tests, déploiement. Chaque sprint est visualisé via Trello.

Utilisation de Git et branches

Organisation des branches

- main : branche stable et déployée.
- develop : branche de développement.
- feature/ : pour les nouvelles fonctionnalités (feature/booking, feature/login...).

README.md et instructions de déploiement

Un fichier README.md complet est fourni (voir dépôt). Il contient :

- instructions d'installation locale ;
- détails techniques ;
- structure du projet ;
- lien de démo.

Tableau Kanban (Trello)

- <https://trello.com/b/OOcfuvCw/ecoride-kanban>

Explication du tableau Kanban et répartition des tâches

Le tableau Trello comprend des colonnes : À faire / En cours / À tester / Fait. Chaque fonctionnalité est découpée en tâches (ex : "connexion utilisateur", "système de réservation", "formulaire dynamique").

Documentation du déploiement

Démarches pour déployer l'application

Le projet a été déployé via FTP (FileZilla) sur un serveur web LWS.fr. La base a été importée via phpMyAdmin.

Outils de déploiement utilisés

- **LWS** : hébergement mutualisé (Apache + MariaDB).
- **FileZilla** : transfert FTP.
- **phpMyAdmin** : pour les exports/imports de base.

Instructions pour déployer localement

1. Cloner le dépôt Git.
2. Placer le projet dans htdocs/ (XAMPP).
3. Importer la base depuis /db/ecoride.sql.
4. Lancer localhost/ecoride depuis un navigateur.

Remerciements

Je tiens à remercier :

- ChatGPT mon formateur particulier pour ses conseils et son accompagnement.
- Les personnes ayant testé l'application pendant le développement.
- L'équipe de Studi pour la richesse de la formation.