Documentation Technique

Application EcoRide - Plateforme de Covoiturage

---

1. **Réflexions initiales technologiques**

1.1 Analyse du cahier des charges

Lors de l'analyse initiale du projet, plusieurs contraintes techniques ont orienté les choix technologiques :

Contraintes imposées :

- Utilisation obligatoire d'une base de données relationnelle
- Application web responsive
- Déploiement en production obligatoire

Exigences fonctionnelles identifiées :

- Gestion de 4 types d'utilisateurs (Visiteur, Utilisateur, Employé, Administrateur)
- Système de crédits avec transactions
- Gestion de 13 User Stories complexes
- Sécurité renforcée pour les données utilisateur

1.2 Choix de la stack technique

Backend - PHP 8+ natif : La décision d'utiliser PHP sans Framework s'est basée sur plusieurs critères :

- Maîtrise approfondie du langage nécessaire pour l'évaluation
- Contrôle total de l'architecture et des performances
- Simplicité de déploiement sur plateformes cloud
- Compatibilité native avec MySQL via PDO

Base de données relationnelle - MySQL : MySQL a été choisi pour sa robustesse et son intégration native avec PHP :

- Optimisation des requêtes
- Facilité de déploiement avec JawsDB sur Heroku

- Documentation extensive et communauté active

Frontend - HTML5/CSS3/JavaScript natif : L'approche sans framework frontend permet :

- Chargement rapide des pages
- Contrôle précis du responsive design
- Compatibilité maximale navigateurs
- Maintenance simplifiée

Déploiement - Heroku : Heroku a été sélectionné pour sa simplicité de mise en production :

- Intégration Git native
- Add-on JawsDB pour MySQL
- Configuration automatique PHP
- Monitoring et logs intégrés

1.3 Architecture logicielle retenue

Pattern MVC simplifié :

- **Modèle** : Classes métier (Connexion, Covoiturage, MonCompte)
- **Vue** : Templates PHP avec includes réutilisables
- **Contrôleur** : Scripts PHP gérant les actions utilisateur

Séparation des responsabilités :

- /classes/ : Logique métier et accès données
- /includes/ : Composants d'interface réutilisables
- /fonction_php/ : Fonctions utilitaires
- /assets/ : Ressources statiques (CSS, images)

2. Configuration de l'environnement de travail

2.1 Environnement de développement local

XAMPP - Configuration :

Apache 2.4.58 - PHP 8.2.12 - MySQL 8.0.34 - phpMyAdmin 5.2.1

Structure des dossiers XAMPP :

C:\xampp\htdocs\Projet_EcoRide\

src/ Code source principal

docs/ Documentation et maquettes

sql/ Scripts de base de données

README.md

composer.json # Configuration Heroku

Procfile # Configuration serveur web

2.2 Configuration base de données locale

Création de la base :

CREATE DATABASE `bdd_eco-ride`

Utilisateur de développement :

-- Utilisation de l'utilisateur root par défaut XAMPP

-- Pas de mot de passe en local (environnement de développement)

2.3 Outils de développement

IDE et extensions :

- Visual Studio Code

Versioning Git :

git init

git remote add origin https://github.com/LeChach/Projet_EcoRide.git

git branch -M main

Workflow de branches :

- main : Production stable
- covoiturage : module pour créer, confirmer, ajouter, annuler des covoiturages
- preference : module de gestion des préférence, affichage et modification de celles-ci
- voiture : module d'ajout et de suppression de voiture
- inscription : module de gestion pour inscription d'un utilisateur

3. Architecture des classes métier

3.1 Classe Connexion

Responsabilités :

- Inscription des nouveaux utilisateurs
- Authentification et gestion des sessions
- Validation des données d'entrée
- Gestion des mots de passe sécurisés

Méthodes principales :

- public static function inscriptionMonCompte (PDO $pdo, array $data): array
- public static function connexionMonCompte (PDO $pdo, array $data): array

3.2 Classe Covoiturage

Responsabilités :

- CRUD des covoiturages
- Recherche avec filtres avancés
- Gestion des réservations et crédits
- Système d'avis et notation

Méthodes principales :

- creationCovoiturage
- Participation (permet à un passager de réserver un covoit)
- VoirCovoiturage (donne les informations d’un covoiturage précis)
- detailCovoiturage
- rechercheCovoiturage et rechercheFiltrerCovoiturage (pour la page de recherche)
- supprimerCovoiturage (pour un conducteur)
- annulerCovoiturage (pour un passager)
- démarrerCovoiturage (pour un conducteur)
- terminerCovoiturage (pour un conducteur)

- donnerAvis
- confirmerCovoiturage (pour un passager)

3.3 Classe MonCompte

Responsabilités :

- Gestion des profils utilisateur
- Administration des véhicules
- Configuration des préférences
- Espaces employé et administrateur

Méthodes principales :

- recupDonnee
- changerTypeUtilisateur
- changerPréférence
- ajouterVoiture
- supprimerVoiture
- voirAvis
- chargerAvis (pour un employé)
- changerAvis
- chargerUtilisateurAdmin (charger les données pour le board admin
- suspendreUtilisateur
- nouvelEmploye

---

4. Documentation du déploiement

4.1 Préparation du déploiement

Configuration Heroku :

```
// composer.json
{
  "require": {
    "php": "^8.0"
  },
  "extra": {
    "platform": {
      "php": "8.2"
    }
  }
}
// Procfile
web: vendor/bin/heroku-php-apache2 src/
```

Variables d'environnement :

- DATABASE_URL : Configurée automatiquement par JawsDB
- HEROKU_PHP_DOCUMENT_ROOT : src/

4.2 Étapes de déploiement

1. Préparation du repository :

```
# Vérification de la structure
git add .
git commit -m "Préparation déploiement Heroku"
git push origin main
```

2. Création de l'application Heroku :

```
heroku create eco-ride-2025
```

heroku config:set HEROKU_PHP_DOCUMENT_ROOT=src

3. Configuration de la base de données :

heroku addons:create jawsdb:kitefin

heroku config:get DATABASE_URL

4. Déploiement initial :

git push heroku main

heroku logs --tail

5. Import de la base de données :

Export depuis XAMPP

mysqldump -u root bdd_eco-ride > ecoride_export.sql

Import vers JawsDB via phpMyAdmin ou ligne de commande

mysql -h [JAWSDB_HOST] -u [JAWSDB_USER] -p[JAWSDB_PASSWORD] [JAWSDB_NAME] < ecoride_export.sql

4.3 Résolution des problèmes de déploiement

Problème 1 - Path vers src/ :

Symptôme : Error 404 sur toutes les pages

Cause : Apache pointe vers la racine au lieu de src/

Solution : Configuration Procfile avec heroku-php-apache2 src/

Problème 2 - Variables d'environnement :

Symptôme : Erreur de connexion base de données

Cause : parse_url($_ENV['DATABASE_URL']) non fonctionnel

Solution : Vérification isset() avant parsing

Problème 3 - Timezone :

Symptôme : Problèmes d'horaires pour démarrage covoiturages

Cause : Serveur Heroku en UTC

Solution : date_default_timezone_set('Europe/Paris')

---

8. Sécurité et performance

8.1 Mesures de sécurité implémentées

Authentification :

- Hachage bcrypt avec password_hash()
- Validation stricte des mots de passe
- Sessions sécurisées avec vérification utilisateur

Protection des données :

- Requêtes préparées PDO sur toutes les interactions
- Échappement HTML avec htmlspecialchars()
- Validation côté serveur de tous les formulaires