

Documentation EcoRide – Gestion de Projet

Vue d'ensemble du Projet

EcoRide est une plateforme de covoiturage écologique développée dans le cadre d'une formation DWWW (Développeur Web et Web Mobile). L'objectif principal est de créer une solution web permettant aux utilisateurs de partager leurs trajets pour réduire l'empreinte carbone liée au transport.

Objectifs du Projet

- Faciliter le covoiturage entre utilisateurs
- Contribuer à la réduction de l'empreinte carbone
- Offrir une interface intuitive pour la gestion des trajets
- Permettre la gestion de profils utilisateurs
- Promouvoir la mobilité durable

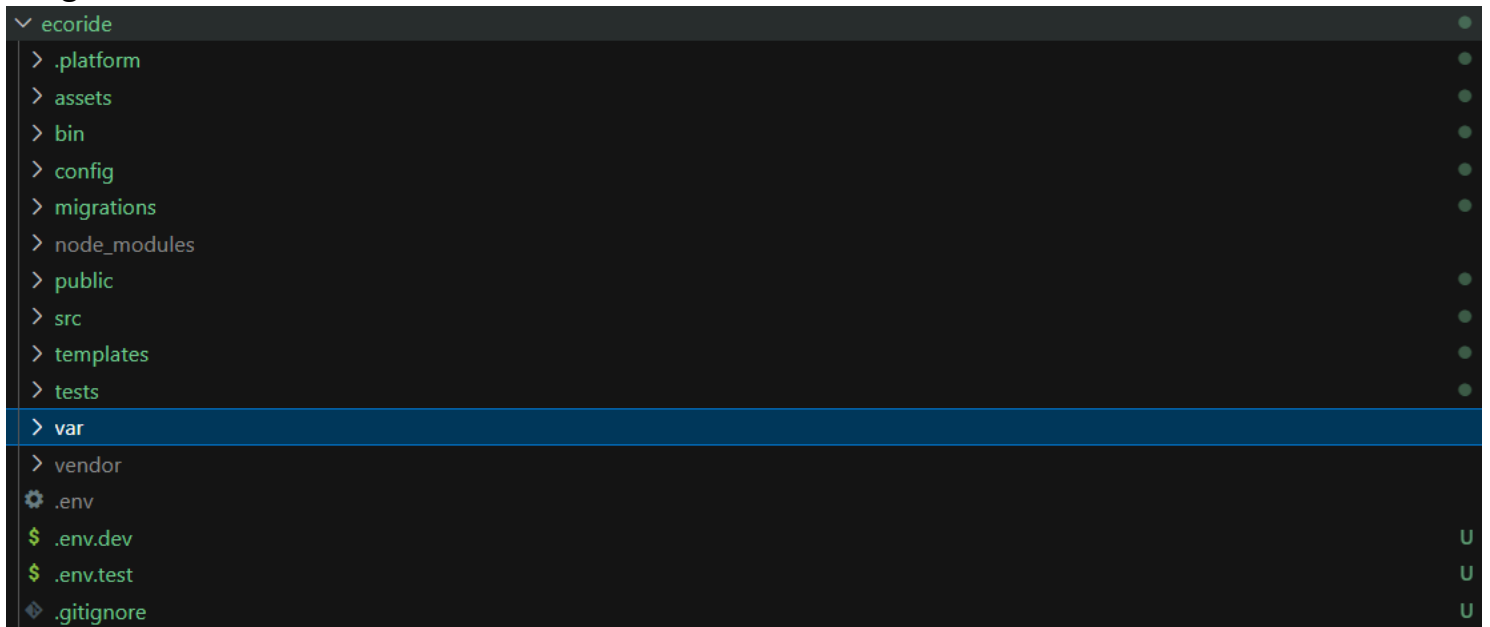
Gestion de Projet

Méthodologie Adoptée

Le projet suit une approche itérative et incrémentale, s'inspirant des principes agiles pour un projet individuel en formation :

1. Phase de Conception Initiale
 - Analyse des besoins utilisateurs
 - Définition des fonctionnalités
 - Choix technologiques argumentés
 - Mise en place de l'architecture
2. Phase de Développement Itératif
 - Développement par fonctionnalités
 - Documentation
3. Phase de Finalisation
 - Tests d'intégration
 - Déploiement et mise en production
 - Documentation finale

Organisation du Code et Structure



Gestion des Versions et Branches

Le projet utilise Git avec une stratégie de branches adaptée au développement individuel :

- main : Branche principale stable
- développement : Branches pour le développement de nouvelles fonctionnalités

Outils de Gestion

1. Versioning : Git & GitHub
2. Dependency Management
 - Composer (PHP)
 - npm/Yarn (JavaScript)
3. Containerisation : Docker avec compose.yaml
4. Build Tools : Webpack Encore pour les assets
5. Testing : PHPUnit pour les tests

Processus de Développement

1. Planification
 - Définition des user stories
 - Priorisation des fonctionnalités
 - Estimation des tâches
2. Développement
 - Création de branches feature
 - Développement avec test

- Merge vers main

3. Déploiement

- Tests d'intégration
- Migration de base de données
- Déploiement

Réflexions Technologiques Initiales

Stack Technologique Choisie

Backend : Symfony (PHP) Justifications

du choix :

1. Maturité et Robustesse

- Framework PHP mature avec une architecture MVC solide
- Ecosystem riche avec de nombreux bundles
- Performance optimisée pour les applications web

2. Productivité

- ORM Doctrine intégré
- Système de routing flexible
- Validation et sérialisation intégrées

3. Sécurité

- Composant Security robuste
- Protection CSRF intégrée
- Gestion des sessions sécurisées
- Chiffrement et hashage des mots de passe

4. Maintenabilité

- Architecture modulaire
- Injection de dépendances
- Tests unitaires et fonctionnels facilités
- Documentation extensive

Base de Données : Approche Hybride

MongoDB + Doctrine ODM :

- Flexibilité pour les données de review
- Scalabilité horizontale

Justification de l'approche hybride :

- MongoDB pour les données flexibles
- Potentiel MySQL/PostgreSQL pour les données relationnelles strictes

Frontend : Twig + SCSS + JavaScript

Twig (Templating) :

- Intégration native avec Symfony
- Syntaxe claire et sécurisée
- Héritage de templates efficace

SCSS (Styling) :

- Organisation modulaire des styles
- Compatibilité avec Bootstrap

Bootstrap + JavaScript :

- Framework CSS responsive
- Composants UI prêts à l'emploi
- JavaScript vanilla pour les interactions

Outils de Build : Webpack Encore

Avantages :

- Intégration native Symfony
- Compilation automatique des assets
- Hot reload en développement
- Optimisation pour la production

Architecture de l'Application

Pattern MVC Étendu

Couches d'Architecture

1. Présentation : Controllers + Templates Twig
2. Application : Services métier
3. Domaine : Entities et logique métier
4. Infrastructure : Repositories et accès données

Considérations de Performance

1. Optimisation Base de Données

- Requêtes optimisées via Doctrine ODM

Sécurité

1. Authentification & Autorisation

- Système de rôles Symfony Security
- Hash sécurisé des mots de passe

2. Audit et Logs

- Logging des actions sensibles
- Monitoring des erreurs
- Traçabilité des modifications

Documentation

- Documentation technique (ce document)
- README.md avec instructions d'installation

Perspectives d'Amélioration

1. API REST
2. Real-time : WebSockets pour notifications en temps réel
3. Microservices : Découpage en services spécialisés
4. CI/CD : Pipeline d'intégration continue
5. Monitoring : Outils de monitoring applicatif

Conclusion

Le projet EcoRide démontre une approche méthodique du développement web moderne, combinant des technologies éprouvées (Symfony) avec des outils contemporains (Webpack, Docker). L'architecture choisie privilégie la maintenabilité et la scalabilité, tout en respectant les contraintes d'un projet de formation.

Les choix technologiques reflètent une volonté d'apprendre les bonnes pratiques du développement web professionnel, avec une attention particulière portée à la sécurité, aux performances et à la qualité du code.