Table of Contents

Réflexions initiales technologiques *1*

Choix du Framework : Symfony **1**

Composants utilisés : **1**

Gestion des données **1**

Structure du code **1**

Cette stratégie a permis de : **1**

Configuration de l'environnement de travail *2*

Outils utilisés en local **2**

Déploiement en ligne **2**

Sécurité et bonnes pratiques **2**

Quelques exécutions pour setup l'environnement de travail et tester la connexion à la base de données **2**

Modèle conceptuel de données avec dbdiagram.io *6*

Diagramme d'utilisation avec PlantUML *6*

Documentation du déploiement de l'application *9*

Démarche globale 9

Étapes détaillées 9

Particularités 10

Documentation technique EcoRide

Réflexions initiales technologiques

Choix du Framework : Symfony

Symfony a été choisi pour sa robustesse, sa structure MVC claire, sa sécurité native et sa documentation complète. Il permet de gérer facilement les entités métier avec Doctrine ORM.

Composants utilisés :

- Symfony avec composer
- Doctrine pour la gestion des entités et des relations MySQL
- Twig pour les templates HTML côté front
- Bootstrap 5 + CSS personnalisé (ecoride.css) pour une interface sobre, accessible et écoresponsable
- Git / GitHub avec workflow dev > main structuré et pushes vers Heroku

Gestion des données

- Le choix pédagogique a été de ne pas utiliser DataFixtures au départ
- Toutes les données ont été insérées via requêtes SQL manuelles, afin de pratiquer le langage MySQL (CREATE, INSERT, ALTER...) avec JawsDB ou localement

Structure du code

- Routes et contrôleurs Symfony respectent les US pas à pas
- Une méthode progressive de développement a été appliquée, avec validation de chaque étape (CRUDs, sécurités, vues)
- L'utilisateur a validé les entités en respectant son propre MCD logique, partagé dans public/docs/

Cette stratégie a permis de :

- Prendre en main le framework Symfony
- Apprendre le SQL en parallèle
- Comprendre la structure MVC et le routage
- Utiliser Doctrine et composant MakerBundle
- S’exercer à la construction d’un projet complet, réaliste et cohérent

Configuration de l'environnement de travail

Outils utilisés en local

- MacBook Air / macOS
- Visual Studio Code pour le développement
- Terminal zsh pour les commandes Git, Symfony
- MySQL pour les données relationnelles
- Sequel Ace pour gérer la base locale et pour l'accès à JawsDB (production)

Déploiement en ligne

- Hébergement via Heroku, avec push depuis Git
- Base de données en ligne via JawsDB (MySQL)
- Connexion entre Symfony et JawsDB via variable d'environnement DATABASE_URL

Sécurité et bonnes pratiques

- .env.local ignoré par Git
- Séparation claire entre branche dev (développement) et main (prod)
- Historique des merges et tests de fonctionnalités respectés avant mise en prod

Quelques exécutions pour setup l'environnement de travail et tester la connexion à la base de données

1. *Créer un nouveau projet Symfony sur Desktop*

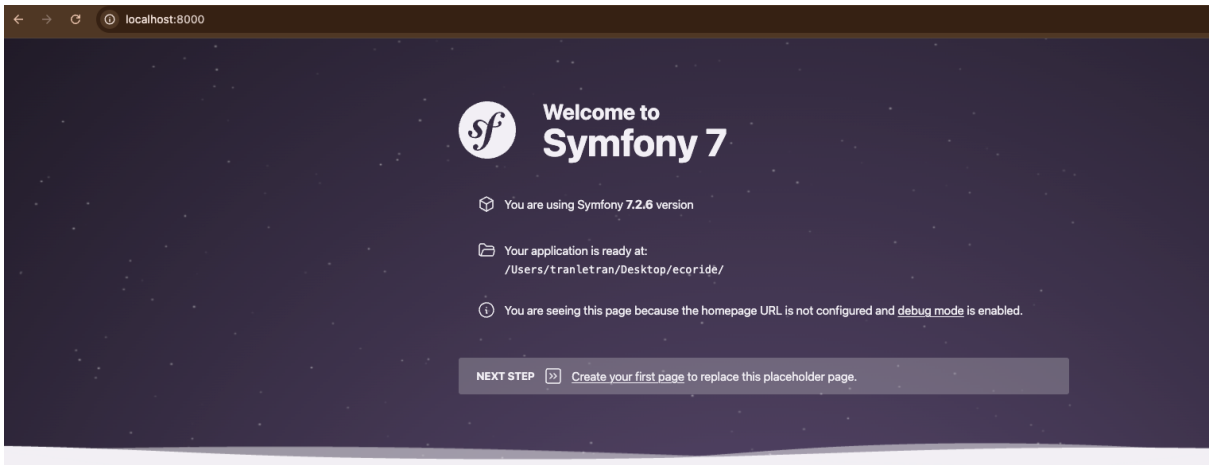
   Symfony new ecoride --webapp

2. *Créer un fichier .env.local avec la bonne connexion MySQL (locale ou JawsDB)*

3. *Démarrer le serveur local Symfony :*

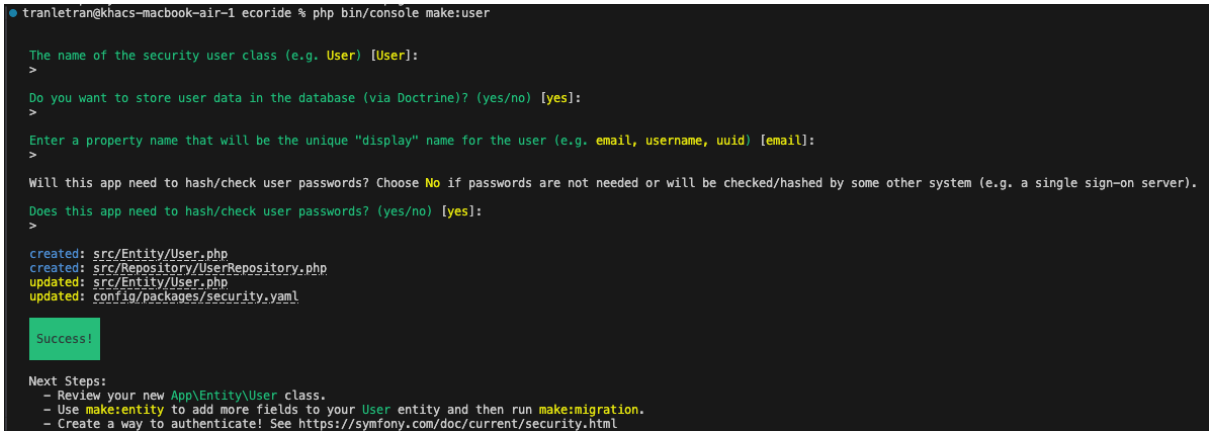
   symfony server:start

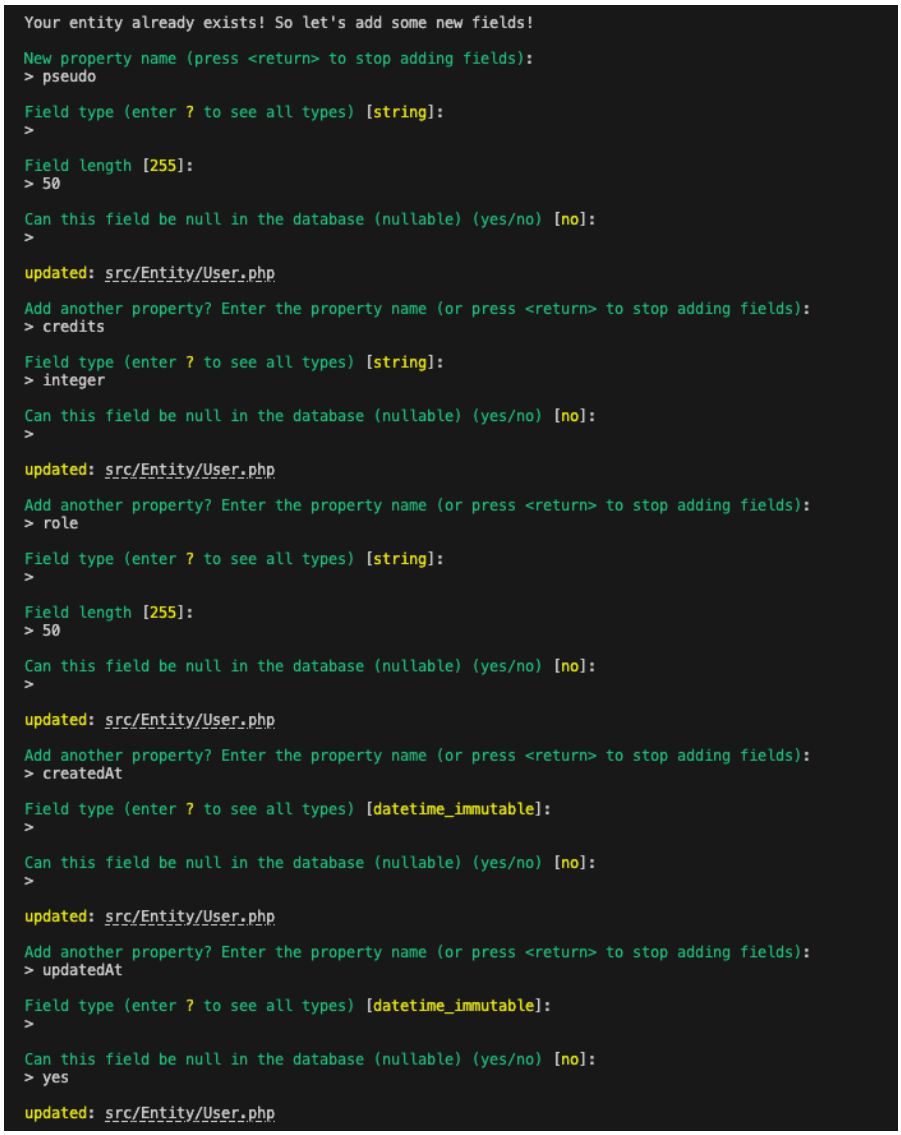
```
[tranletran@khacs-macbook-air-1 Desktop % cd ~/Desktop
[tranletran@khacs-macbook-air-1 Desktop % symfony new ecoride --webapp
* Creating a new Symfony project with Composer
* Setting up the project under Git version control
  (running git init /Users/tranletran/Desktop/ecoride)

 [OK] Your project is now ready in /Users/tranletran/Desktop/ecoride

[tranletran@khacs-macbook-air-1 Desktop % cd ~/Desktop/ecoride
[tranletran@khacs-macbook-air-1 ecoride % symfony server:start

 [WARNING] run "symfony server:ca:install" first if you want to run the web se
 rver with TLS support, or use "--p12" or "--no-tls" to avoid this w
 arning

Following Web Server log file (/Users/tranletran/.symfony5/log/0128a94287c49ea5c0f8039f2b2900d6d133d1ce.lo
g)
Following PHP-FPM log file (/Users/tranletran/.symfony5/log/0128a94287c49ea5c0f8039f2b2900d6d133d1ce/53fb8
ec204547646acb3461995e4da5a54cc7575.log)

 [WARNING] The local web server is optimized for local development and MUST ne
 ver be used in a production setup.


 [WARNING] Please note that the Symfony CLI only listens on 127.0.0.1 by defau
 lt since version 5.10.3.
           You can use the --allow-all-ip or --listen-ip flags to ch
 ange this behavior.

 [OK] Web server listening
      The Web server is using PHP FPM 8.4.7
      http://127.0.0.1:8000
```



4. *Créer une Base de données (exemple)*

4.1 Créer l'entité User, configurer l'enregistrement d'un nouvel utilisateur avec un formulaire d'inscription.





4.2 Migration de BDD :

Php bin/console doctrine :database :create

Php bin/console make :migrate

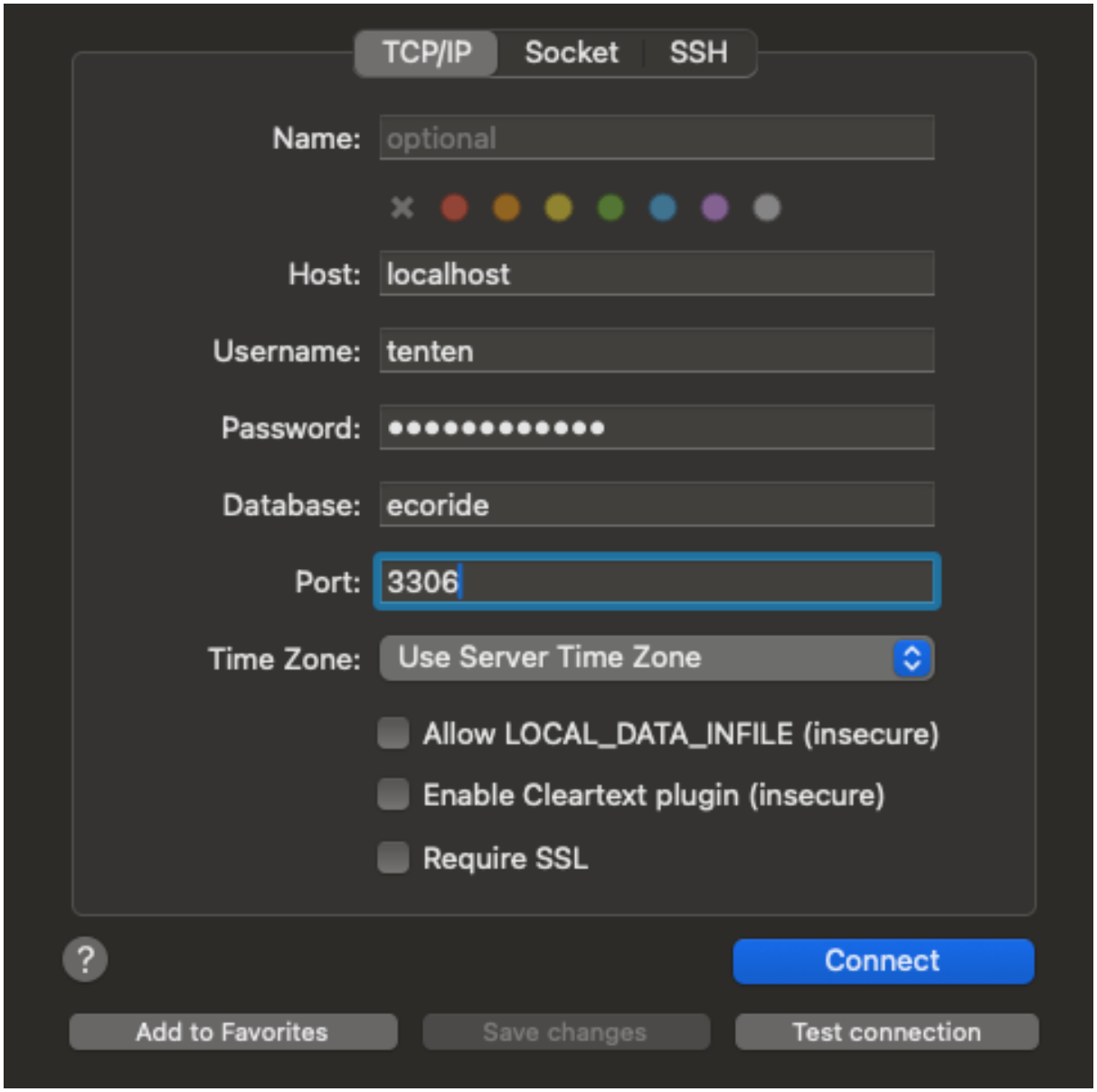
```
tranletran@khacs-macbook-air-1 ecoride % php bin/console d:d:c
Created database `ecoride` for connection named default
tranletran@khacs-macbook-air-1 ecoride % php bin/console make:migration
 created: migrations/Version20250526043313.php

  Success!

 Review the new migration then run it with php bin/console doctrine:migrations:migrate
 See https://symfony.com/doc/current/bundles/DoctrineMigrationsBundle/index.html
```

Php bin/console doctrine :migrations :migrate

```
tranletran@khacs-macbook-air-1 ecoride % php bin/console doctrine:migrations:migrate

 WARNING! You are about to execute a migration in database "ecoride" that could result in schema changes and data loss. Are you sure you wish to continue? (yes/no) [yes]:
 >

[notice] Migrating up to DoctrineMigrations\Version20250526043313
[notice] finished in 43.1ms, used 22M memory, 1 migrations executed, 2 sql queries

 [OK] Successfully migrated to version: DoctrineMigrations\Version20250526043313
```

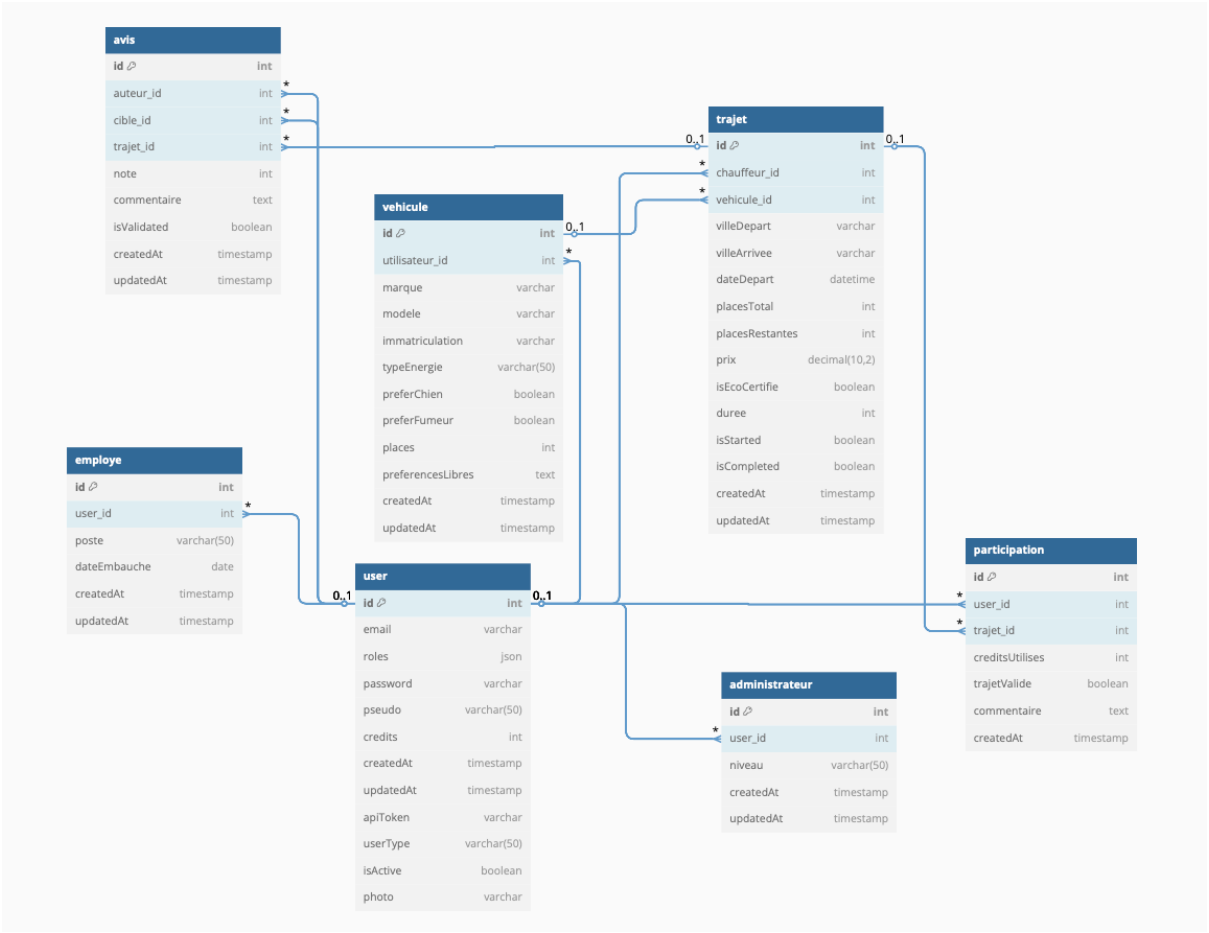
5. *Test la connexion :*

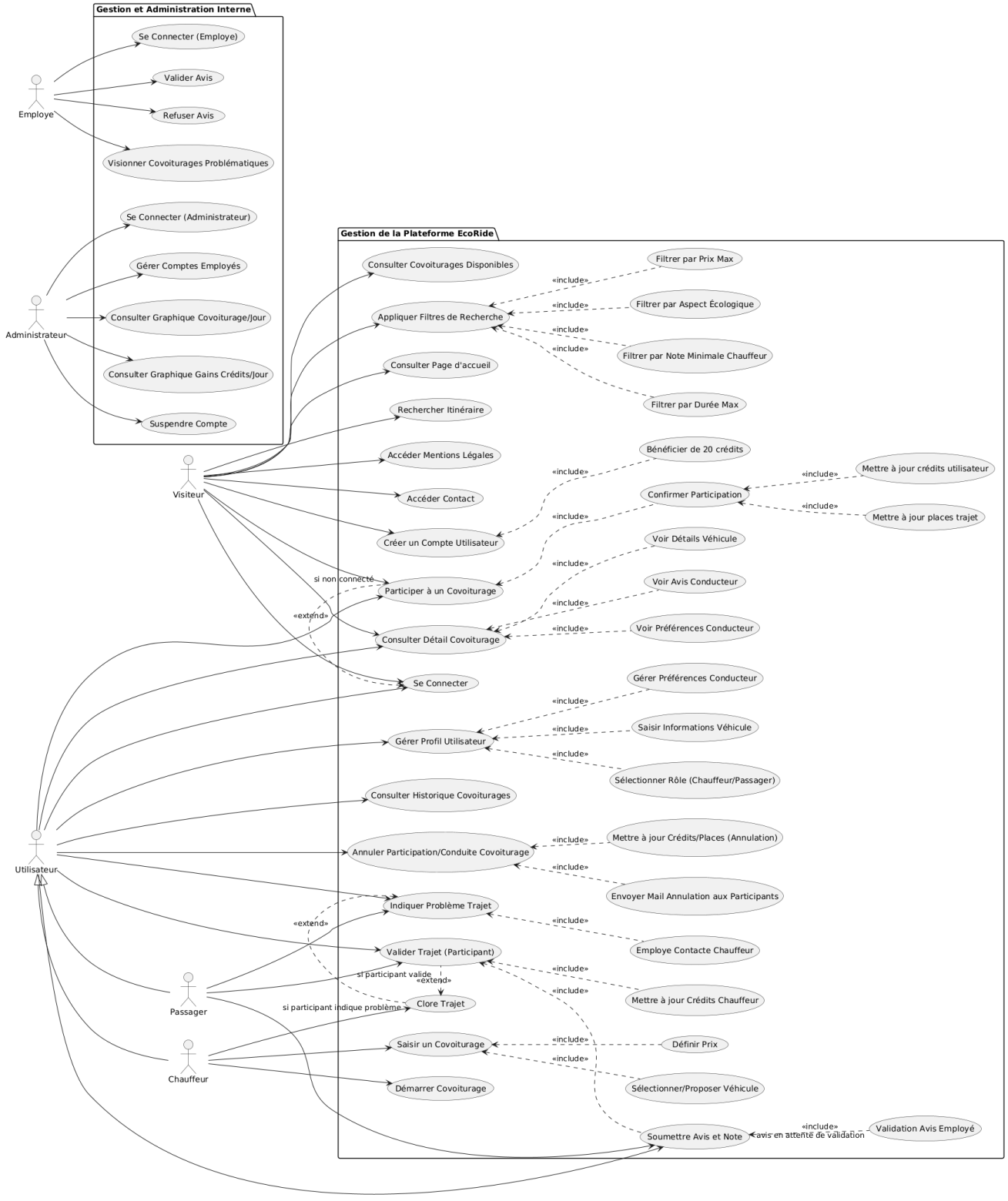
```
tranletran@khacs-macbook-air-1 ecoride % mysql -u tenten -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 138
Server version: 9.3.0 Homebrew

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| data               |
| information_schema |
| laravel            |
| mysql              |
| performance_schema |
| restaurant         |
| sys                |
| tenten_data        |
+--------------------+
8 rows in set (0,038 sec)

mysql> exit;
Bye
```



Modèle conceptuel de données avec dbdiagram.io

Diagramme d'utilisation avec PlantUML


avis
id int
auteur_id int
cible_id int
trajet_id int
note int
commentaire text
isValidated boolean
createdAt timestamp
updatedAt timestamp
trajet
id int
chauffeur_id int
vehicule_id int
villeDepart varchar
villeArrivee varchar
dateDepart datetime
placesTotal int
placesRestantes int
prix decimal(10,2)
isEcoCertifie boolean
duree int
isStarted boolean
isCompleted boolean
createdAt timestamp
updatedAt timestamp
vehicule
id int
utilisateur_id int
marque varchar
modele varchar
immatriculation varchar
typeEnergie varchar(50)
preferChien boolean
preferFumeur boolean
places int
preferencesLibres text
createdAt timestamp
updatedAt timestamp
employe
id int
user_id int
poste varchar(50)
dateEmbauche date
createdAt timestamp
updatedAt timestamp
user
id int
email varchar
roles json
password varchar
pseudo varchar(50)
credits int
createdAt timestamp
updatedAt timestamp
apiToken varchar
userType varchar(50)
isActive boolean
photo varchar
participation
id int
user_id int
trajet_id int
creditsUtilises int
trajetValide boolean
commentaire text
createdAt timestamp
administrateur
id int
user_id int
niveau varchar(50)
createdAt timestamp
updatedAt timestamp


Gestion et Administration Interne
Se Connecter (Employe)
Valider Avis
Refuser Avis
Visionner Covoiturages Problématiques
Employe
Se Connecter (Administrateur)
Gérer Comptes Employés
Consulter Graphique Covoiturage/Jour
Consulter Graphique Gains Crédits/Jour
Suspendre Compte
Administrateur
Gestion de la Plateforme EcoRide
Consulter Covoiturages Disponibles
Filtrer par Prix Max
Appliquer Filtres de Recherche
Filtrer par Aspect Écologique
Filtrer par Note Minimale Chauffeur
Filtrer par Durée Max
Consulter Page d'accueil
Rechercher Itinéraire
Accéder Mentions Légales
Accéder Contact
Créer un Compte Utilisateur
Visiteur
Bénéficier de 20 crédits
Confirmer Participation
Mettre à jour crédits utilisateur
Mettre à jour places trajet
Participer à un Covoiturage
Voir Détails Véhicule
Voir Avis Conducteur
Voir Préférences Conducteur
Consulter Détail Covoiturage
Se Connecter
Gérer Préférences Conducteur
Saisir Informations Véhicule
Gérer Profil Utilisateur
Sélectionner Rôle (Chauffeur/Passager)
Consulter Historique Covoiturages
Mettre à jour Crédits/Places (Annulation)
Annuler Participation/Conduite Covoiturage
Envoyer Mail Annulation aux Participants
Utilisateur
Indiquer Problème Trajet
Employe Contacte Chauffeur
Valider Trajet (Participant)
Mettre à jour Crédits Chauffeur
Passager
Clore Trajet
Saisir un Covoiturage
Définir Prix
Chauffeur
Démarrer Covoiturage
Sélectionner/Proposer Véhicule
Soumettre Avis et Note
Validation Avis Employé
«include»
«extend»
si non connecté
si participant valide
si participant indique problème
avis en attente de validation

Documentation du déploiement de l'application

Démarche globale

Le déploiement de l'application EcoRide s'est fait en suivant une méthode progressive : développement local validé par étapes (chaque US est fait et testé et validé), versionnage propre sur GitHub depuis Vscode, puis mise en ligne via Heroku.

Étapes détaillées

1. Création d'un dépôt GitHub public
2. Utilisation des branches `dev` pour le développement et `main` pour la production
3. Création un compte Heroku et connexion depuis vscode avec `heroku create`
4. Test local du projet Symfony (*`symfony server:start`*)
5. Merge de la branche *`dev`* vers *`main`* une fois stable
6. Push de *`main`* vers GitHub :

   `git push origin main`

7. Push de `main` vers Heroku :

   `git push heroku main`

8. Configuration *`framework.yaml`* et *`.env.local`*

```
TRUSTED_HOSTS=localhost,127.0.0.1
TRUSTED_PROXIES=127.0.0.1,REMOTE_ADDR
```

```
framework:
    secret: '%env(APP_SECRET)%'

    session:
        handler_id: null
        cookie_secure: auto

    trusted_proxies: '%env(TRUSTED_PROXIES)%'
    trusted_hosts: '%env(TRUSTED_HOSTS)%'
```

9. Connexion de l'application Symfony à JawsDB (MySQL)

Name JawsDB EcoRide

Host `sh4ob67ph9l80v61.cbetxkdyhwsb.us-east-1.rds.amazonaws.com`

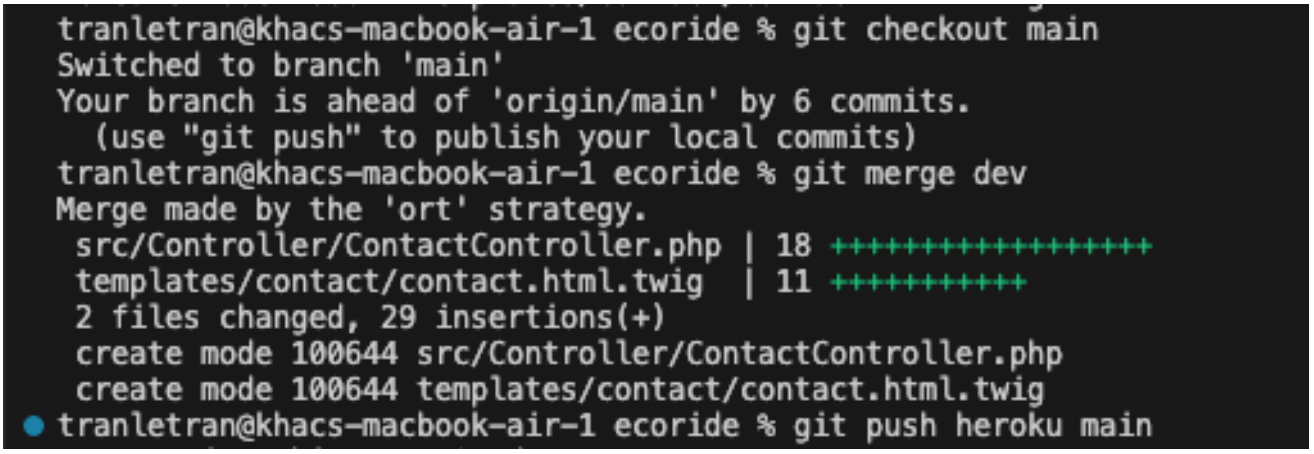
Username `fnh9mzt4bmkesuh3`

Password `nbq6nzhw2c9hs02m`

Database `idyfmn02vbpvokn8`

Port `3306`

10. Mise à jour manuelle de la base JawsDB si besoin



Particularités

- Aucun usage de `doctrine:migrations` : les modifications SQL ont été faites à la main pour compréhension
- Les fichiers `.sql` nécessaires sont stockés dans `public/docs/`
- Tests de fonctionnement réalisés via Heroku live (formulaires, sessions, requêtes)

Cette démarche permet un contrôle total sur chaque étape du déploiement, et garantit une compréhension approfondie du cycle de vie complet d'une application web Symfony.