

Titre professionnel Développeur web et web mobile

Lucas Cherbuin
Ecoride



Présentation

Parcours professionnel

- Apprentissage CFC Logisticien (Entreprise PWP)
 - 2014-2018
 - Travaille en Intérim
 - 2019-2021
 - Centre de tri Coop Aclens
 - 2021-maintenant
- En parallèle
 - Study
 - Développeur web/web mobile

Réorientation

- Volonté de travailler dans un milieu lié à l'informatique.
- Profiter de changer de voie dans une entreprise ayant une grande diversité de carrières professionnels.

Index

1. Présentation du projet

- Présentation de l'entreprise
- Cahier des charges et dictionnaires

2. Conception

- Wireframe et Maquettage
- Technologies

3. Développement

- Configuration de l'environnement de travail
- Conception Back end
- Erreur liés aux middlewares
- Conception Front end
- Explication du template « blade » et adapter la page selon le rôle
- Sécurité

4. Déploiement

- Intégration de la base de donnée
- Dépendance, sécurité et déploiement

5. Conclusion

- Ce que le projet m'a permis

Présentation du projet



Présentation de l'entreprise

- Entreprise Ecoride
- But : réaliser des covoiturage pour réduire l'empreinte carbone favorisant les énergies propres
- Volonté : Avoir une application sécurisée avec des fonctionnalité pour les consommateurs ainsi que les acteurs principaux de l'entreprise.

Cahier des charges

■ Utilisation

Consommateur	
Conducteur	Réaliser des annonces et pouvoir annoncer les départs.
Passager	Confirmer des covoitages puis laisser un avis
Intervenant de l'entreprise	
Employés	Valider les avis et gérer les litiges
Administrateur	Opérations Crud des utilisateurs et consulter des statiques relatif aux covoitages et rentabilité

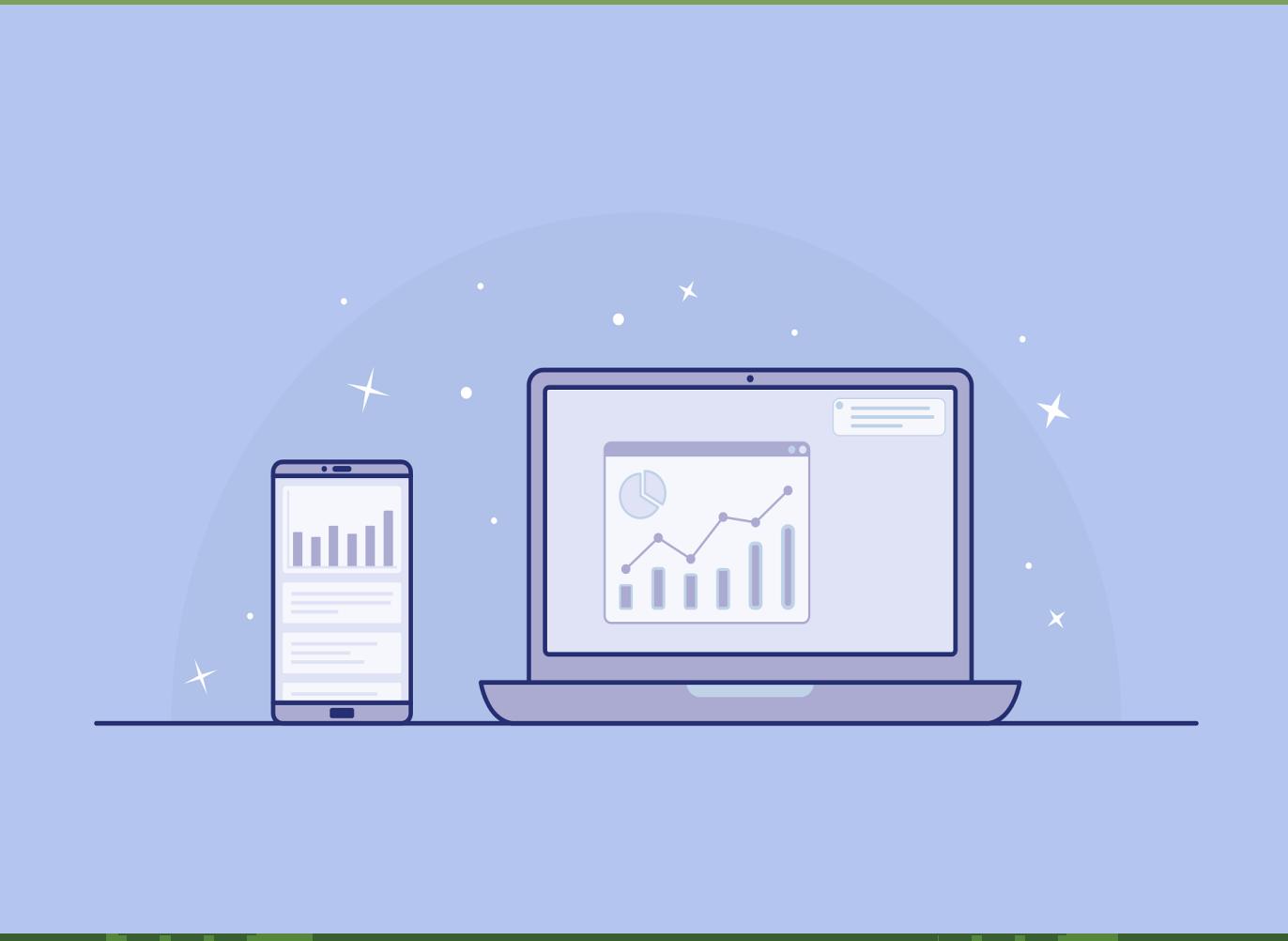
■ Design

- Responsive : pouvoir être utiliser autant sur mobile qu'un ordinateur
- Thème : avoir un thème et ambiance rappelant la nature

Technologies

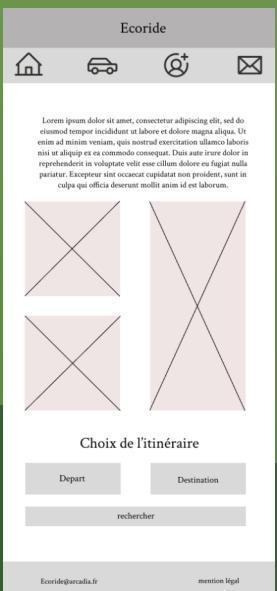
- Back end
 - Laravel (PHP)
 - MySQL
 - MongoDB (NoSql)
 - Mailjet
- Front end
 - Blade (HTML, CSS, PHP)
 - Bootstrap
 - JavaScript
- Déploiement
 - Heroku
- Autre Outils
 - Visual Studio Code
 - Git/Github
 - Docker
 - PhpMyAdmin
 - Xampp

Conception



Wireframe et Maquettage

Wireframes



Maquettes

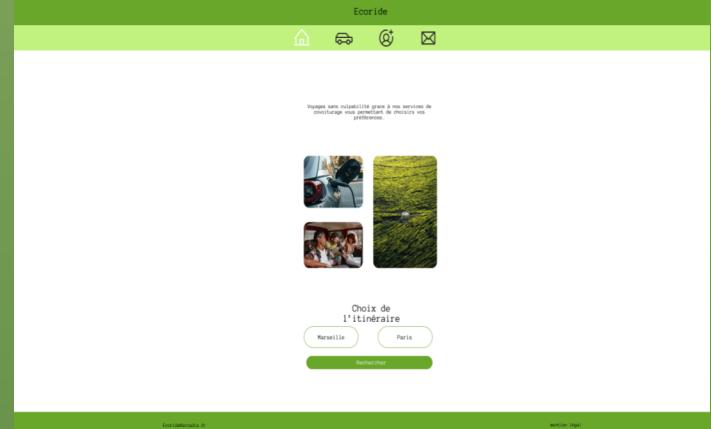
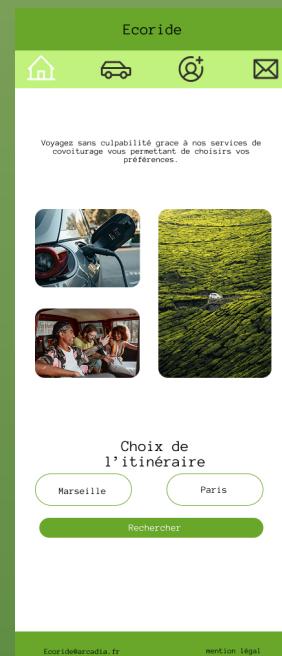
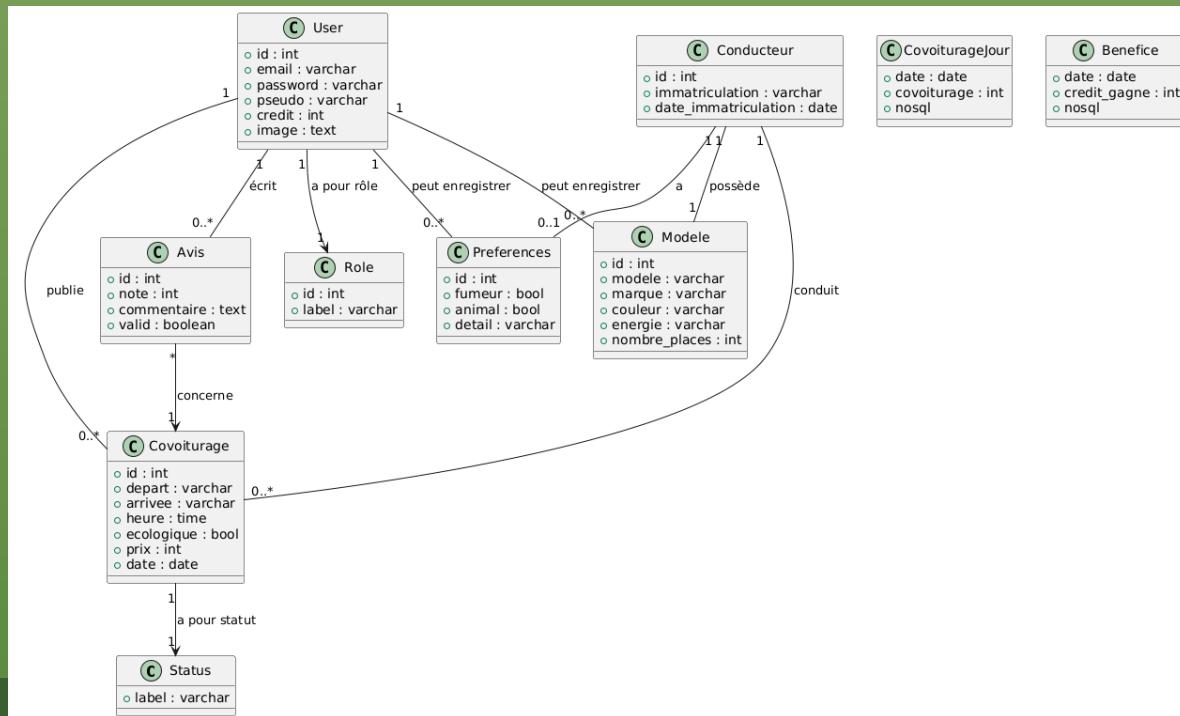


Diagramme de classes et Dictionnaire

Diagramme base de données relationnels



Dictionnaire

Nom de la donnée	But	Type	Contrainte
Avis_id	Code numérique d'un avis	Int	Primary key
Note	Evaluation d'un trajet	Int	Not null
Commentaire	Détail sur un trajet	Text	Not null
Valid	Confirmation si l'avis est affiché sur une annonce	Boolean	Not null
User_id	Relation avec la table user	Int	Foreign key
Covoiturage_id	Relation avec la table covoiturage	Int	Foreign key
Conducteur_id	Code numérique d'un conducteur	Int	Primary key
Immatriculation	Numéro en hexadecimal de la plaque d'immatriculation	Text	Not null
Date_immatriculation	Date d'immatriculation d'un véhicule	Date	Not null
Modele_id	Relation avec la table model	Int	Foreign key
Preference_id	Relation avec la table preference	Int	Foreign key
Covoiturage_id	Relation avec la table covoiturage	int	Foreign Key

Développement

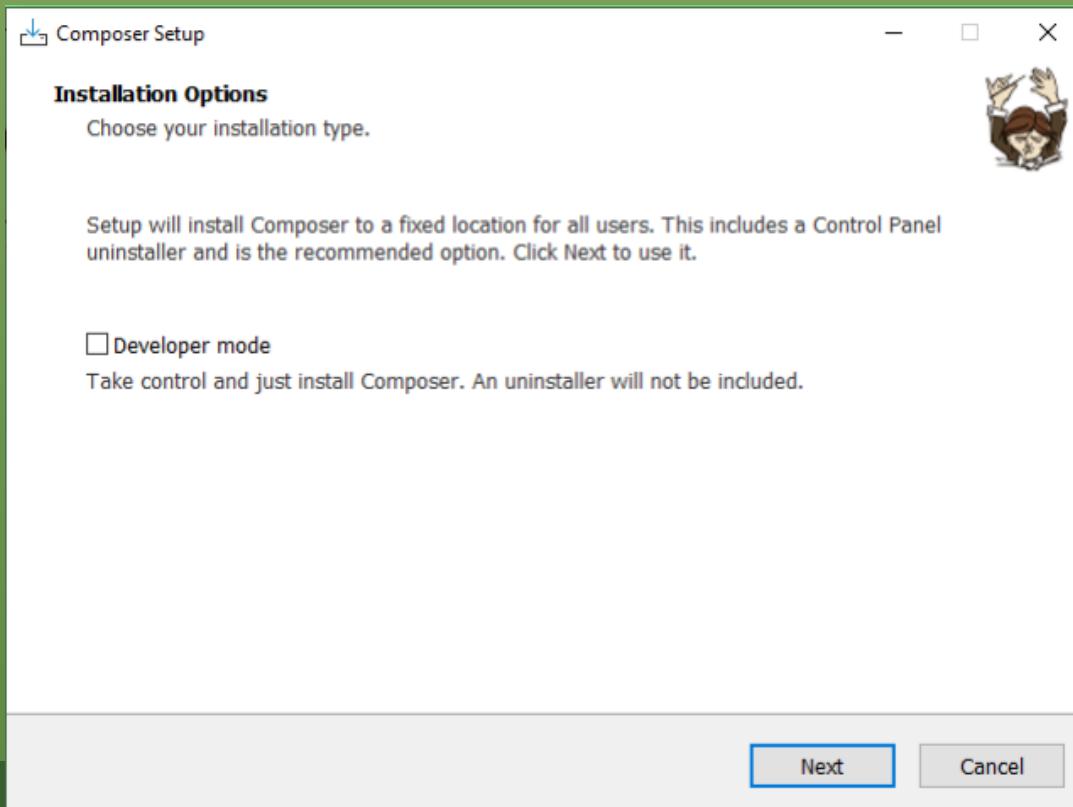
```
<li><a href="index.html">Home</a></li>
<li><a href="home-events.html">Home Events</a></li>
<li class="multi-col-menu.html">Multiple Column Menu on Larger Viewports
    <ul>
        <li><a href="#" class="current">Header Options</a>
        <li><a href="tall-button-header.html">Tall Button Header</a></li>
        <li><a href="image-logo.html">Image Logo</a></li>
        <li class="active"><a href="tall-logo.html">Tall Logo Images</a>
    </ul>
</li>
<li class="has-children"> <a href="#">Carousels</a>
    <ul>
        <li><a href="variable-width-slider.html">Variable Image Width Slider</a>
        <li><a href="testimonial-slider.html">Testimonial Slider</a>
        <li><a href="featured-work-slider.html">Featured Work Slider</a>
        <li><a href="equal-column-slider.html">Equal Column Slider</a>
        <li><a href="video-slider.html">Video Slider</a></li>
        <li><a href="mini-bootstrap-carousel.html">Mini Carousel</a>
    </ul>
</li>
```

Mise en place de l'environnement



Initiation de Laravel

Composer



Laravel

```
PS G:\projet code\Ecoride> composer create-project laravel/laravel Ecoride
```

Structure

The screenshot shows a file explorer interface displaying the directory structure of a Laravel project. The root directory contains several subfolders: '.anima', 'app', 'bootstrap', 'config', 'database', 'public', 'resources', and 'routes'. To the right of these files, there are several small colored dots, likely representing additional files or subdirectories that are too numerous to list individually.

- > .anima
- > app
- > bootstrap
- > config
- > database
- > public
- > resources
- > routes

Configuration variable d'environnement

Paramètre de base

```
APP_NAME=Laravel  
APP_ENV=local  
APP_KEY=base64:wHaHqq081ICnkDmccQXcYt93gn9P4J60Y981XbfyK9w=  
APP_DEBUG=true  
APP_TIMEZONE=UTC  
APP_URL=http://localhost
```

Base de données SQL

```
#Base de données mysql  
DB_CONNECTION=mysql  
DB_HOST=127.0.0.1  
DB_PORT=3306  
DB_DATABASE=laravel
```

Base de données NoSql

```
DB_CONNECTION=mongodb  
DB_HOST=127.0.0.1  
DB_PORT=27017  
DB_DATABASE=EcorideNoSQL
```

Mailer

```
MAIL_MAILER=smtp  
MAIL_HOST=in-v3.mailjet.com  
MAIL_PORT=587  
MAIL_USERNAME  
MAIL_PASSWORD  
MAIL_ENCRYPTION:  
MAIL_FROM_ADDRESS=ecoride@example.com  
MAIL_FROM_NAME="Ecoride"
```

Docker

Docker File

```
You, 2 months ago | 1 author (You)
# Utiliser une image officielle PHP avec Apache
FROM php:8.2-apache

# Installer les extensions PHP nécessaires
RUN apt-get update && apt-get install -y \
    libicu-dev libpq-dev libzip-dev unzip git \
    && docker-php-ext-install intl pdo pdo_mysql zip

# Activer mod_rewrite pour Laravel
RUN a2enmod rewrite

# Installer Composer
COPY --from=composer:latest /usr/bin/composer /usr/bin/composer

# Définir le répertoire de travail
WORKDIR /var/www/html

# Copier les fichiers du projet
COPY . .

# Installer les dépendances PHP
RUN composer install --no-dev --no-interaction --optimize-autoloader

# Définir les permissions pour Laravel
RUN chown -R www-data:www-data /var/www/html/storage /var/www/html/bootstrap/cache

# Exposer le port 80
EXPOSE 80

CMD ["apache2-foreground"]
```

Docker-compose.yml

```
services:
  app:
    > Run Service
    build: .
    container_name: laravel_app
    restart: always
    volumes:
      - ./var/www/html
    ports:
      - "8000:80"
    depends_on:
      - mysql
      - mongodb
    environment:
      APP_ENV: local
      DB_CONNECTION: mysql
      DB_HOST: mysql
      DB_PORT: 3306
      DB_DATABASE: laravel_db
      DB_USERNAME: root
      DB_PASSWORD: root
      MONGO_URL: mongodb://mongodb:27017/laravel_mongo

  mysql:
    > Run Service
    image: mysql:8
    container_name: laravel_mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: root
      MYSQL_DATABASE: laravel_db
    ports:
      - "3307:3306"
    volumes:
      - mysql_data:/var/lib/mysql

  mongodb:
    > Run Service
    image: mongo:6
    container_name: laravel_mongo
    restart: always
    ports:
      - "27018:27017"
    volumes:
      - mongo_data:/data/db

volumes:
  mysql_data:
  mongo_data:
```

Git/GitHub

Dépôt Github

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * Repository name *

 LucasCherbuin /

Great repository names are short and memorable. Need inspiration? How about [psychic-pancake](#) ?

Description (optional)

 Public
Anyone on the internet can see this repository. You choose who can commit.

 Private
You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

 You are creating a public repository in your personal account.

Configuration Git

```
# Initialiser un dépôt Git dans ton projet
git init

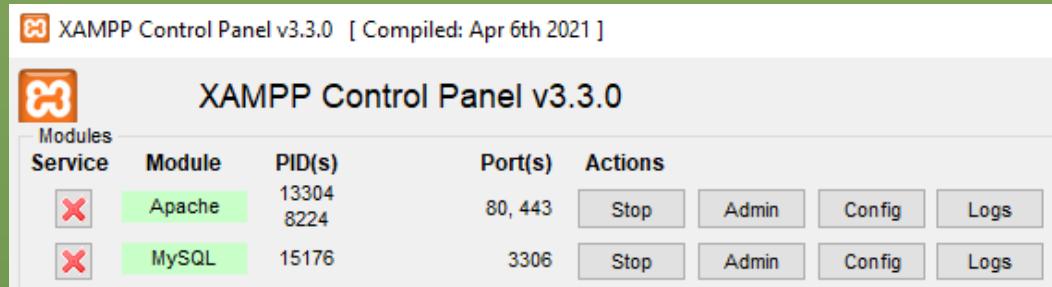
# Ajouter tous les fichiers du projet
git add .

# Faire un premier commit
git commit -m "Premier commit : initialisation du projet"

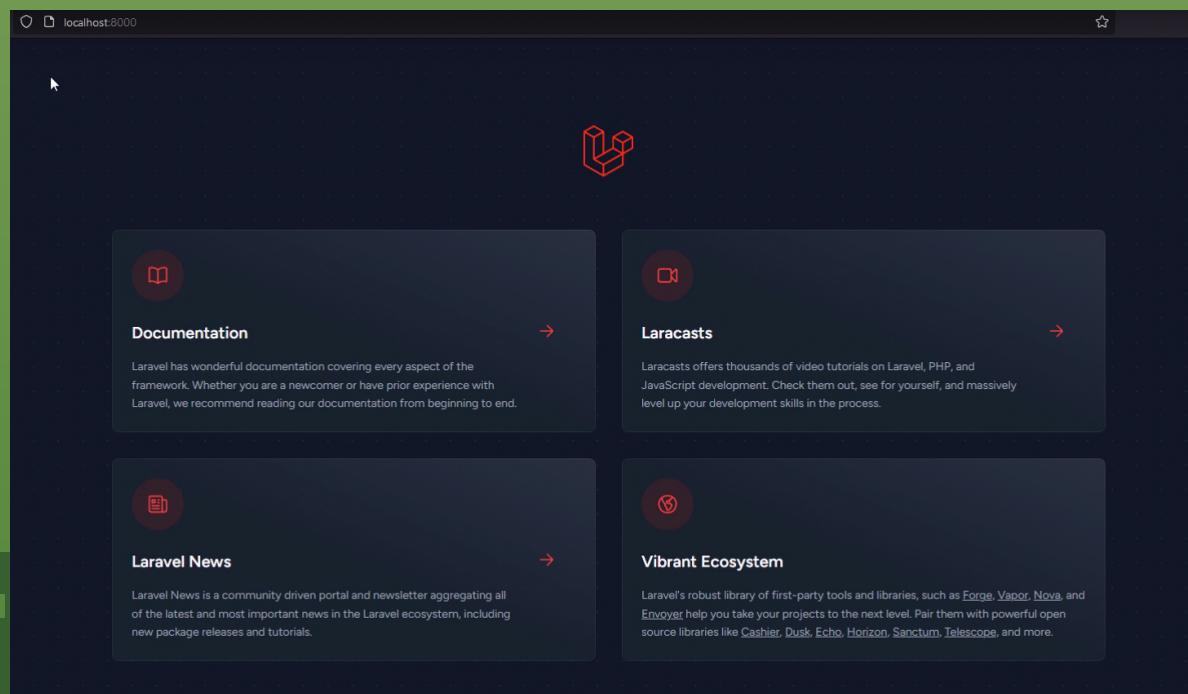
# Ajouter ton dépôt distant GitHub
git remote add origin https://github.com/LucasCherbuin/Ecoride.git

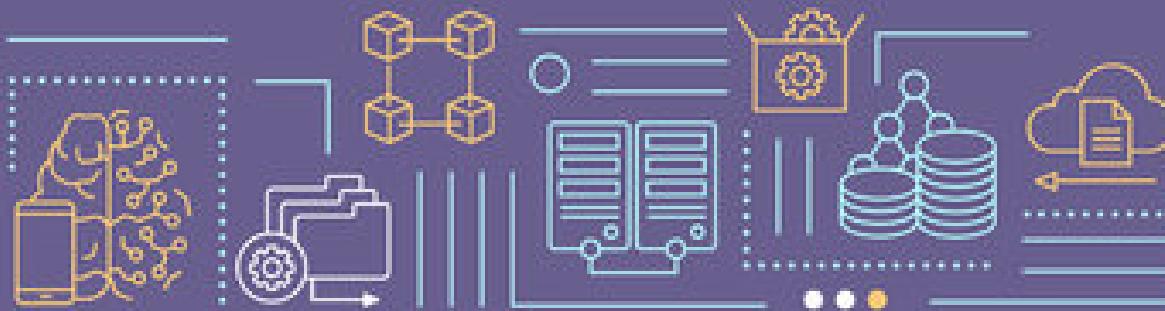
# Pousser le projet vers GitHub sur la branche principale
git push -u origin master
```

Démarrage de l'application en local

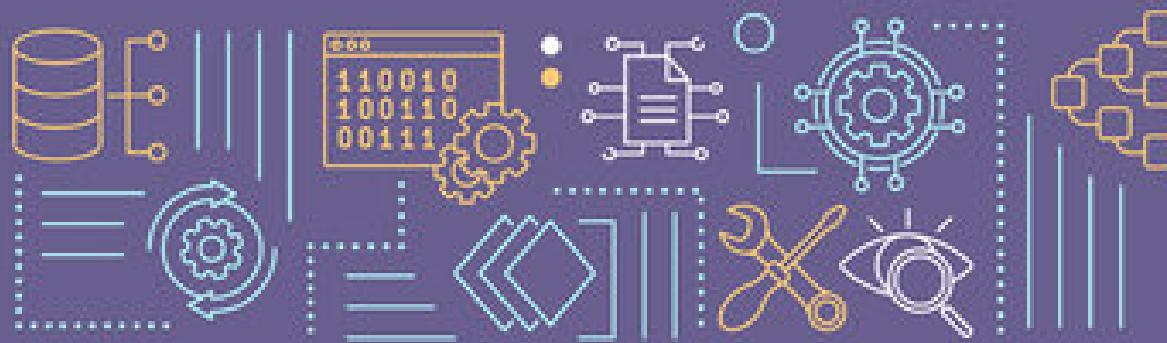


```
PS G:\projet\code\Ecoride> php artisan serve
```





BACK-END



Générations modèles

PS G:\projet code\Ecoride> php artisan make:model covoiturage

```
30 references | 0 implementations | You, 3 weeks ago | 1 author (You)
class Covoiturage extends Model
{
    use HasFactory;

    protected $fillable = [
        'id',
        'date',
        'depart',
        'arrive',
        'prix',
        'heure_depart',
        'heure_arrive',
        'energie_verte',
        'creation'];
}

You, 2 months ago * initiation du projet ecoride sous larav
0 references | 0 overrides
public function users(): HasMany
{
    return $this->hasMany(related: User::class);
}

0 references | 0 overrides
public function conducteur(): HasOne
{
    return $this->hasOne(related: Conducteur::class);
}

0 references | 0 overrides
public function status(): HasOne
{
    return $this->hasOne(related: Status::class);
}

0 references | 0 overrides
public function avis(): HasMany
{
    return $this->hasMany(related: Avis::class);
}
```

```
<?php

namespace App\Models;

// use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use App\Models\Role;

You, 46 minutes ago | 1 author (You) | 67 references | 0 implementations
class User extends Authenticatable
{
    /** @use HasFactory<\Database\Factories\UserFactory> */
    use HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var list<string>
     */
    0 references
    protected $table = 'user';

    0 references
    protected $fillable = [
        'id',
        'image',
        'pseudo',
        'email',
        'role',
        'password',
        'Credit'
    ];
}
```

```
/*
 * The attributes that should be hidden for serialization.
 *
 * @var list<string>
 */
0 references
protected $hidden = [
    'password',
    'remember_token',
];

```

Migrations

```
G:\projet\code\Ecoride>php artisan make:migration ecoride_bdd
```

```
Schema::create('covoiturage', function (Blueprint $table): void {
    $table->id();
    $table->string(column: 'depart');
    $table->string(column: 'arrivee');
    $table->date(column: 'date');
    $table->time(column: 'heure_depart');
    $table->time(column: 'heure_arrive');
    $table->integer(column: 'prix');
    $table->boolean(column: 'ecologique')->default(value: false);
    $table->foreignId(column: 'user_id')->constrained(table: 'user')->onDelete(action: 'cascade');
    $table->foreignId(column: 'conducteur_id')->constrained(table: 'conducteur')->onDelete(action: 'cascade');
    $table->foreignId(column: 'status_id')->constrained(table: 'status')->onDelete(action: 'cascade');
});
```

Table	Action	Lignes	Type	Interclassement	Taille	Perte
avis	★ Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_unicode_ci	48,0 kio	-
conducteur	★ Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_unicode_ci	80,0 kio	-
covoiturage	★ Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_unicode_ci	80,0 kio	-
migrations	★ Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_unicode_ci	16,0 kio	-
modele	★ Parcourir Structure Rechercher Insérer Vider Supprimer	1	InnoDB	utf8mb4_unicode_ci	16,0 kio	-
preference	★ Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_unicode_ci	32,0 kio	-
role	★ Parcourir Structure Rechercher Insérer Vider Supprimer	6	InnoDB	utf8mb4_unicode_ci	32,0 kio	-
sessions	★ Parcourir Structure Rechercher Insérer Vider Supprimer	4	InnoDB	utf8mb4_unicode_ci	48,0 kio	-
status	★ Parcourir Structure Rechercher Insérer Vider Supprimer	4	InnoDB	utf8mb4_unicode_ci	32,0 kio	-
user	★ Parcourir Structure Rechercher Insérer Vider Supprimer	4	InnoDB	utf8mb4_unicode_ci	16,0 kio	-
10 tables	Somme	19	InnoDB	utf8mb4_general_ci	400,0 kio	0 o

▷ Run | □ Select
CREATE TABLE `covoiturage` (
`id` bigint(20) UNSIGNED NOT NULL,
`depart` varchar(255) NOT NULL,
`arrivee` varchar(255) NOT NULL,
`date` date NOT NULL,
`heure` time NOT NULL,
`prix` int(11) NOT NULL,
`ecologique` Boolean(1) NOT NULL DEFAULT 0,
`user_id` bigint(20) UNSIGNED NOT NULL,
`conducteur_id` bigint(20) UNSIGNED NOT NULL,
`status_id` bigint(20) UNSIGNED NOT NULL,
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

Controllers

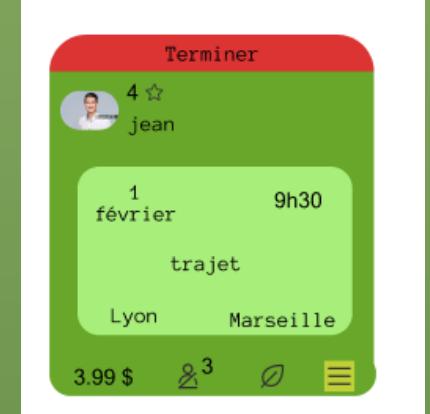
```
3 references | 0 implementations | View code
class StartRideController extends Controller
{
    // Méthode pour faire évoluer le statut du covoiturage
    0 references | 0 overrides
    public function begin($id): JsonResponse|mixed
    {
        // On récupère le covoiturage par son ID avec sa relation "status"
        $covoiturage = Covoiturage::with(relations: 'status')->findOrFail(id: $id);

        try {
            if ($covoiturage->status->label === 'en prévision') {
                $nouveauStatus = Status::where(column: 'label', operator: 'en cours')->first();
            } elseif ($covoiturage->status->label === 'en cours') {
                $nouveauStatus = Status::where(column: 'label', operator: 'terminé')->first();
                $this->sendAvisMail(covoiturage: $covoiturage);
            } else {
                return response()->json(data: ['message' => 'Statut non modifiable.'], status: 400);
            }

            if ($nouveauStatus) {
                $covoiturage->status_id = $nouveauStatus->id;
                $covoiturage->save();
            }

            return response()->json(data: ['message' => 'Statut mis à jour.']);
        }

        catch (\Exception $e) {
            return response()->json(data: [
                'message' => 'Une erreur est survenue.',
                'error' => $e->getMessage()
            ], status: 500);
        }
    }
}
```



```
@foreach($covoiturages as $covoiturage)
    <div class="button" style="background-color: {{ $covoiturage->status->nom == 'en prévision' ? 'lightgreen' : 'red' }}">
        <div class="card-header">
            <h3>{{ $covoiturage->status->nom == 'en prévision' ? 'Démarrer' : 'Terminer' }}</h3>
        </div>
        <div class="card-body">
            @include('components.annonce')

            <button onclick="changerStatut({{ $covoiturage->id }})" class="btn btn-primary mt-2">
                {{ $covoiturage->status->nom == 'en prévision' ? 'Démarrer le covoiturage' : 'Terminer le covoiturage' }}
            </button>
        </div>
    @endforeach

    <script>
        function changerStatut(id) {
            fetch(`/covoiturage/${id}/changer-statut`, {
                method: 'PUT',
                headers: {
                    'X-CSRF-TOKEN': '{{ csrf_token() }}',
                    'Content-Type': 'application/json'
                }
            }).then(response => response.json())
            .then(data => location.reload())
            .catch(error => console.error('Erreur:', error));
        }
    </script>
```

Middleware

RoleMiddleware

```
You, 4 hours ago | 1 author (You)
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Symfony\Component\HttpFoundation\Response;

You, 4 hours ago | 1 author (You) | 4 references | 0 implementations
class RoleMiddleware
{
    0 references | 0 overrides
    public function handle($request, Closure $next, ...$role): mixed
    {
        // Récupère l'utilisateur actuellement authentifié
        $user = Auth::user();

        // Si l'utilisateur est connecté et qu'il a des rôles à vérifier
        if ($user) {
            // Récupère les labels des rôles de l'utilisateur (on suppose une relation 'role')
            $userRole = $user->role->pluck('label')->toArray();

            // Vérifie si l'utilisateur a un des rôles demandés
            if (!array_intersect(array: $role, arrays: $userRole)) {
                // Si non, on renvoie une erreur 403
                abort(code: 403, message: 'Accès interdit');
            }
        } else {
            // Si l'utilisateur n'est pas connecté, on renvoie une erreur 403
            abort(code: 403, message: 'Accès interdit');
        }

        // Continue la requête si l'utilisateur a le rôle adéquat
        return $next($request);
    }
}
```

ImageUploadMiddleware

```
You, 3 weeks ago | 1 author (You)
<?php

namespace App\Http\Middleware;

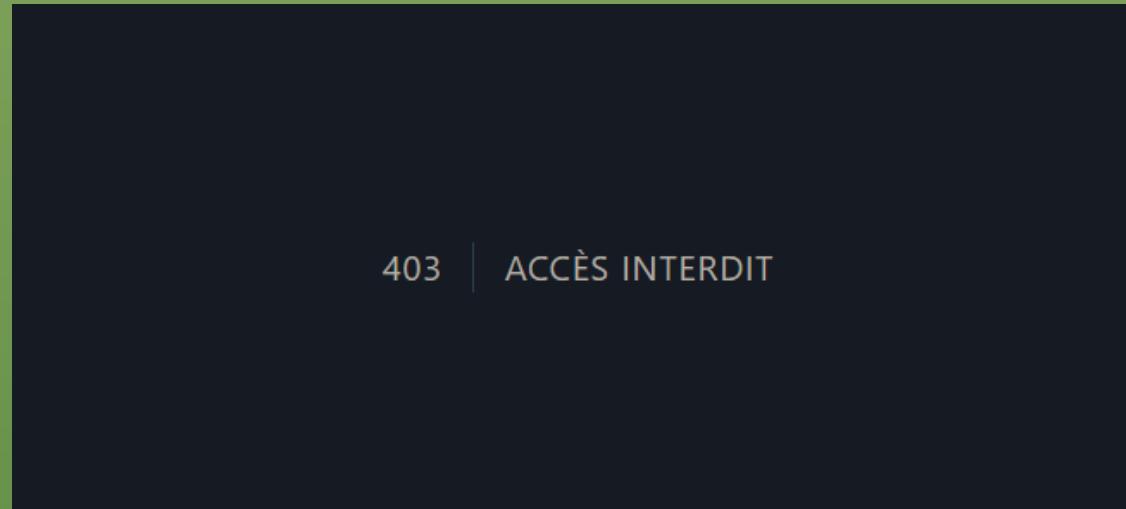
use Closure;
use Illuminate\Http\Request;
use Symfony\Component\HttpFoundation\Response;

You, 3 weeks ago | 1 author (You) | 4 references | 0 implementations
class ImageUploadMiddleware
{
    0 references | 0 overrides
    public function handle(Request $request, Closure $next): Response
    {
        // Vérifie si une image est bien envoyée
        if ($request->hasFile(key: 'image')) {
            $file = $request->file(key: 'image');
            $extension = strtolower(string: $file->getClientOriginalExtension());

            // Vérifie si l'extension est valide
            if (!in_array(needle: $extension, haystack: ['jpg', 'jpeg', 'png'])) {
                return response()->json(data: ['error' => 'Image non supportée'], status: 400);
            }
        }

        return $next($request);
    }
}
```

Erreur lié au Middleware



```
dd(vars: 'RoleMiddleware exécuté', $request->method(), $request->url(), $role);
// Récupère l'utilisateur actuellement authentifié
```

"RoleMiddleware exécuté" // app\Http\Middleware\RoleMiddleware.php:14

Routing

Page Consommateurs

```
Route::middleware(['auth', 'role:ROLE_USER'])->group(callback: function (): void {
    Route::post(uri: '/choisir-role', action: [RoleChoiceController::class, 'choisirRole'])->name(name: 'choisir.role');
});

Route::middleware(['auth', 'role:ROLE_CONDUCTEUR|ROLE_CHAUFFEURPASSAGER'])->group(callback: function () {
    Route::get(uri: '/covoiturage/create', action: [CovoiturageController::class, 'create'])->name(name: 'covoiturage.create');
    Route::post(uri: '/covoiturage/store', action: [CovoiturageController::class, 'store'])->name(name: 'covoiturage.store');
    Route::get(uri: '/covoiturageDemmarage', action: [StartRideController::class, 'begin']);
    Route::put(uri: '/covoiturageDemmarage/{id}/changer-statut', action: [StartRideController::class, 'begin']);
});

Route::middleware(['auth', 'role:ROLE_CONDUCTEUR|ROLE_CHAUFFEURPASSAGER|ROLE_PASSAGER'])->group(callback: function () {
    Route::patch(uri: '/historique', action: [StatusController::class, 'updateStatus']);
    Route::patch(uri: '/historique/{id}/status', action: [StatusController::class, 'updateStatus']);
});

Route::middleware(['auth', 'role:ROLE_CHAUFFEURPASSAGER|ROLE_PASSAGER'])->group(callback: function () {
    Route::patch(uri: '/avis', action: [AvisController::class, 'updateStatus']);
    Route::patch(uri: '/avis/{id}/status', action: [AvisController::class, 'updateStatus']);
});
```

Web

```
require base_path(path: 'routes/auth.php');
require base_path(path: 'routes/pageCustomer.php');
require base_path(path: 'routes/pageEmployee.php');
require base_path(path: 'routes/pagesAdmin.php');

Route::get(uri: '/', action: function () {
    return view(view: 'welcome');
});

Route::get(uri: '/contact', action: [ContactController::class, 'create'])->name(name: 'contact.form');
Route::post(uri: '/contact', action: [ContactController::class, 'send'])->name(name: 'contact.send');

Route::get(uri: '/base', action: function () {
    return view(view: 'base');
});

Route::get(uri: '/covoiturage', action: [RechercheCovoiturageController::class, 'itineraire'])->name(name: 'covoiturage.search');
Route::get(uri: '/annonce/{id}', action: [CovoiturageController::class, 'show'])->name(name: 'covoiturage.show');
Route::post(uri: '/reservation-choice', action: [RechercheCovoiturageController::class, 'handleChoice'])->name(name: 'reservation.choice');
Route::post(uri: '/annonce/reservation/{nombresPlaces}', action: [RechercheCovoiturageController::class, 'reserver']);
```



Gestionnaire de Template

Base.blade.php

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Ecoride</title>

    <!-- Icônes -->
    <script src="https://unpkg.com/@phosphor-icons/web@2.1.1"></script>

    <!-- Styles -->
    <link href="{{ asset(path: 'assets/css/app.css') }}" rel="stylesheet">
</head>
<body>
    <header>...
    </header>

    <main>
        @yield(section: 'content')
    </main>

    <footer>
        <p class="footer-text">Ecoride@arcadia.fr</p>
        <p class="footer-text"><a href="{{ url(path: '/mentions-legales') }}>Mentions légales</a></p>
    </footer>

    You, 1 second ago • Uncommitted changes
</body>
</html>
```

Welcome.blade.php

```
@extends(view: 'base')
@section(section: 'content')

    <p>Voyagez sans culpabilité grâce à nos services de <br> covoiturage vous permettant de choisir vos <br> préférences.</p>

    <div class="accueil">
        
        
        
    </div>
```

Utilisation de blade pour les rôles

Header

```
<header>
  <div class="title">
    <h1>Ecoride</h1>
  </div>

  <nav class="navbar">
    <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
      <div class="navbar-nav">
        <a class="nav-item" href="{{ url(path: '/') }}>
          <i class="ph ph-house-line"></i>
        </a>
        <a class="nav-item" href="{{ url(path: '/covoiturage') }}>
          <i class="ph ph-car-profile"></i>
        </a>
      </div>
    </div>
    @auth
    @php
      $role = auth()->user()->role;
    @endphp

    @if ($role === 'ROLE_ADMIN')
      <a class="nav-item" href="{{ url(path: '/admin/menuAdmin') }}>
        <i class="ph ph-user-circle"></i>
      </a>
    @elseif ($role === 'ROLE_EMPLOYEE')
      <a class="nav-item" href="{{ url(path: '/employee/menuEmployee') }}>
        <i class="ph ph-user-circle"></i>
      </a>
    @elseif (in_array($role, ['ROLE_USER', 'ROLE_CONDUCTEUR', 'ROLE_PASSAGER']))
      <a class="nav-item" href="{{ url(path: '/user/menuCustomer') }}>
        <i class="ph ph-user-circle"></i>
      </a>
    @endif
    @endauth

    @guest
      <a class="nav-item" href="{{ url(path: '/login') }}>
        <i class="ph ph-user-circle-plus"></i>
      </a>
    @endguest
  </nav>
</header>
```

Menu Customer

```
@php

if (Auth::user()->hasRole('ROLE_USER')) {
  $buttonConfig[] = [
    'link' => route(name: 'CustomerUser.RoleAttribution'),
    'icon' => 'ph ph-user-check',
    'description' => 'Validation'
  ];
}

if (Auth::user()->hasRole('ROLE_CONDUCTEUR') || Auth::user()->hasRole('ROLE_CHAUFFEURPASSAGER')) {
  $buttonConfig[] = [
    'link' => route(name: 'CustomerUser.annonceCreation'),
    'icon' => 'ph ph-signpost',
    'description' => 'Création annonce'
  ];
  $buttonConfig[] = [
    'link' => route(name: 'CustomerUser.gestionTrajet'),
    'icon' => 'ph ph-car-profile',
    'description' => 'Gestion des trajets'
  ];
  $buttonConfig[] = [
    'link' => route(name: 'CustomerUser.trajet'),
    'icon' => 'ph ph-flag-banner',
    'description' => 'Trajet en cours (conducteur)'
  ];
}

if (Auth::user()->hasRole('ROLE_PASSAGER') || Auth::user()->hasRole('ROLE_CHAUFFEURPASSAGER')) {
  $buttonConfig[] = [
    'link' => route(name: 'CustomerUser.gestionTrajet'),
    'icon' => 'ph ph-signpost',
    'description' => 'Trajet en cours (passager)'
  ];
  $buttonConfig[] = [
    'link' => route(name: 'CustomerUser.avis'),
    'icon' => 'ph ph-star',
    'description' => 'Avis'
  ];
}
```

Responsive

```
/* Responsive */
@media screen and (max-width: 600px) {
  /* Adapter la grille en une seule colonne */
  .accueil {
    display: flex;
    flex-direction: column;
    width: auto;
    height: auto;
    align-items: center;
  }

  /* Toutes les images en pleine largeur */
  .image-box {
    width: 90%;
    height: auto;
  }

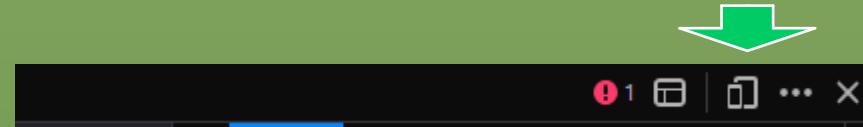
  /* Formulaire ajusté */
  .recherche {
    width: 90%;
  }

  .form-input {
    width: 100%;
  }
}

@media (min-width: 768px) {
  .container {
    font-size: 18px;
    max-width: 720px;
    margin: 0 auto;
  }
}

/* Desktop (min-width: 1024px) */
@media (min-width: 1024px) {
  .container {
    font-size: 20px;
    max-width: 960px;
  }
}
```

You, 2 weeks ago • Uncommitted changes



Version ordinateur

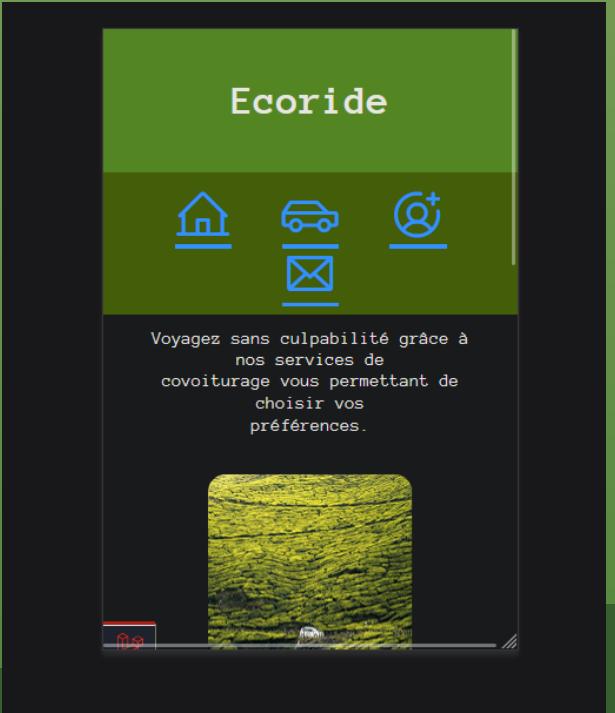
Ecoride

Voyagez sans culpabilité grâce à nos services de covoiturage vous permettant de choisir vos préférences.

Départ
Entrez votre départ
Arrivée
Entrez votre arrivée
Rechercher

Ecoride@arcadia.fr

Mentions légales



Version mobile

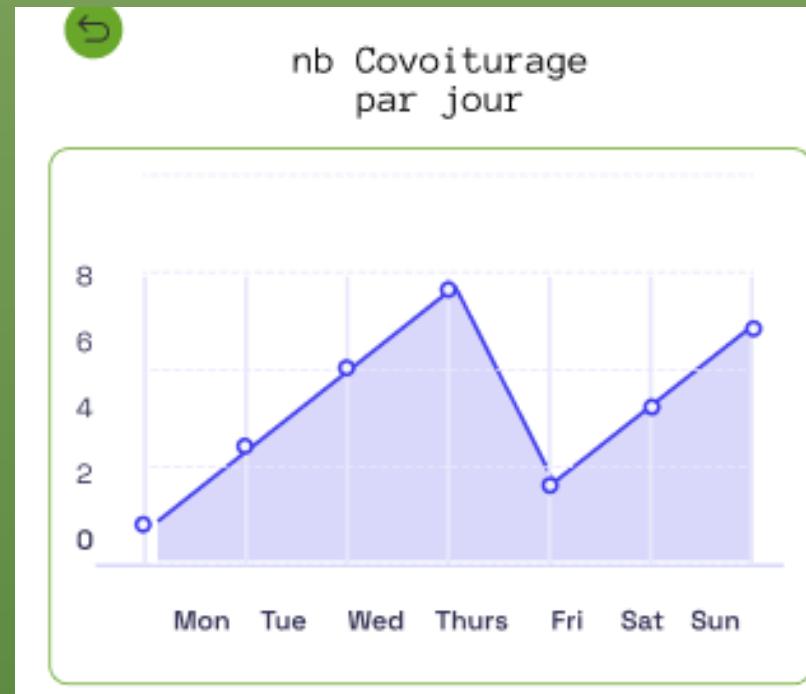
Dashboard dynamique

Dashboard covoiturage

```
<script>
document.addEventListener("DOMContentLoaded", function() {
    // Envoie une requête GET pour récupérer les données JSON depuis ton backend
    fetch('/statistiques/covoiturages')
        .then(response => response.json())
        .then(data => {
            // Extrait les labels (ex: dates) et les valeurs (ex: nombre de trajets)
            const labels = data.map(item => item.date);
            const values = data.map(item => item.nb_covoiturages);

            // Sélectionne le canvas du graphique
            const ctx = document.getElementById('CovoiturageChart').getContext('2d');

            // Crée le graphique avec Chart.js
            new Chart(ctx, {
                type: 'bar',
                data: {
                    labels: labels,
                    datasets: [{
                        label: 'Nombre de covoiturages',
                        data: values,
                        backgroundColor: 'rgba(255, 255, 255, 0.6)',
                        borderColor: 'rgba(163, 230, 53, 1)',
                        borderWidth: 1
                    }]
                },
                options: {
                    responsive: true,
                    scales: {
                        y: {
                            beginAtZero: true,
                            ticks: {
                                stepSize: 1
                            }
                        }
                    }
                }
            });
        })
        .catch(error => {
            console.error("Erreur lors de la récupération des données :", error);
        });
});
</script>
```

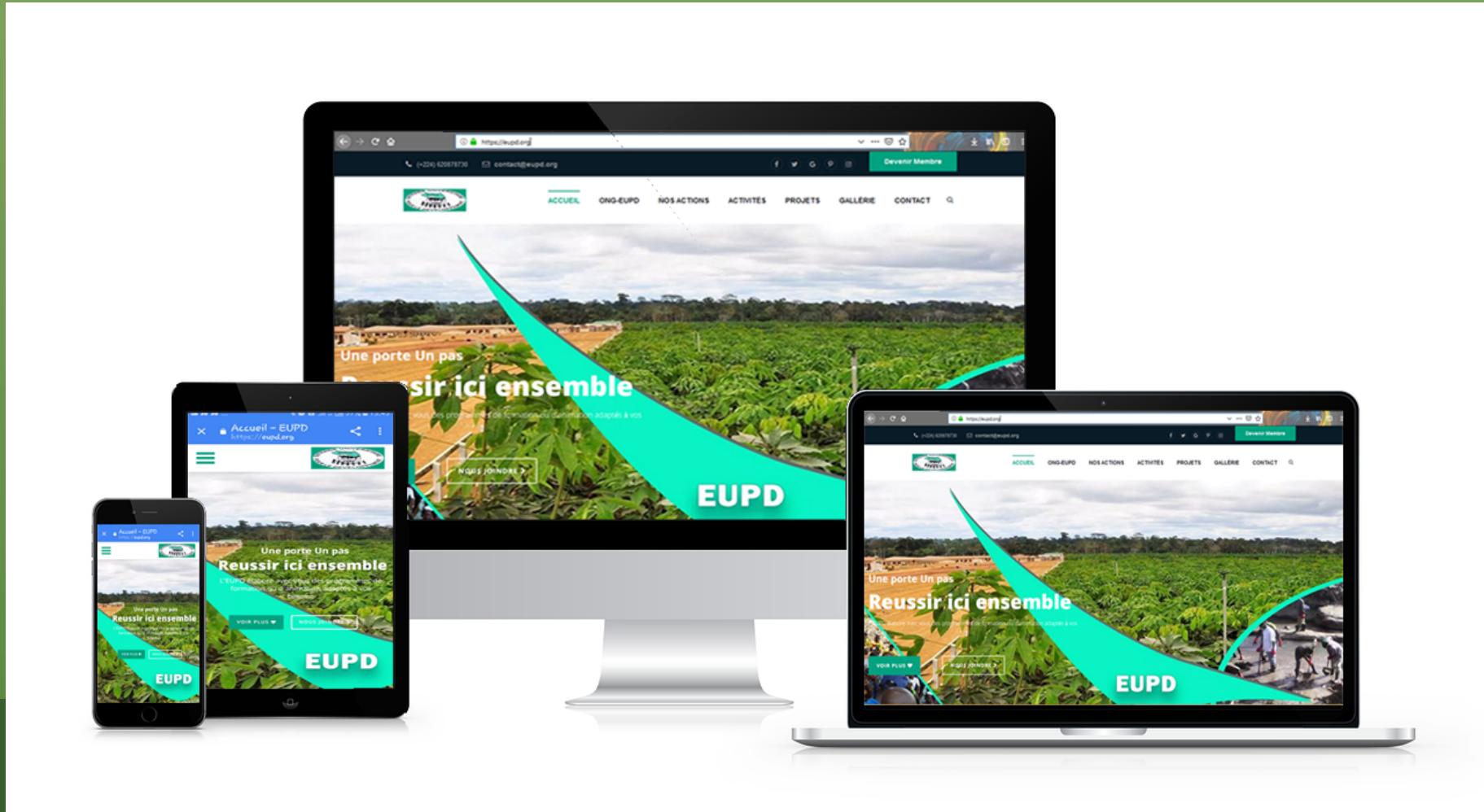


Sécurité



Problème	Solution	Image
CSRF	Intégration de token <code>@csrf</code> dans les formulaires	<pre>--> Formulaire --> <form action="{{ route(name: 'covoiturage.store') }}" method="POST"> @csrf {{-- Protection CSRF obligatoire --}}</pre>
Injection SQL	Utiliser les outils Query Builder ou Eloquant au lieu de requêtes sql	<pre>DB::table(table: 'status')->insert(values: [['label' => 'en prevision'], ['label' => 'en cours'], ['label' => 'annule'], ['label' => 'termine'],]);</pre>
Attaque XSS	Afficher des variables avec des accolades sous Laravel	<pre><tbody id="userTableBody"> @foreach (\$users as \$user) <tr data-role="{{ \$user->role_id->label ?? 'Aucun rôle' }}> <td>{{ \$user->pseudo }}</td> <td>{{ \$user->email }}</td> <td></pre>
Données sensibles (mot de passe)	Utilisation de l'algorithme <i>Bcrypt</i> pour hasher des mots de passes	<pre>// Créer un nouvel utilisateur \$user = User::create(attributes: ['name' => \$request->pseudo, 'email' => \$request->email, 'password' => Hash::make(value: \$request->password),]);</pre> You, 2 months ago * ajout d'un crud utilisateur
Autorisations et authentications	Utiliser le package <i>Laravel breeze</i> , implémenter des rôles, configurer des middlewares relatifs aux utilisateurs	<pre>PS G:\projet code\Ecoride> composer require laravel/breeze --dev >> >> php artisan breeze:install * History restored</pre>

Déploiement



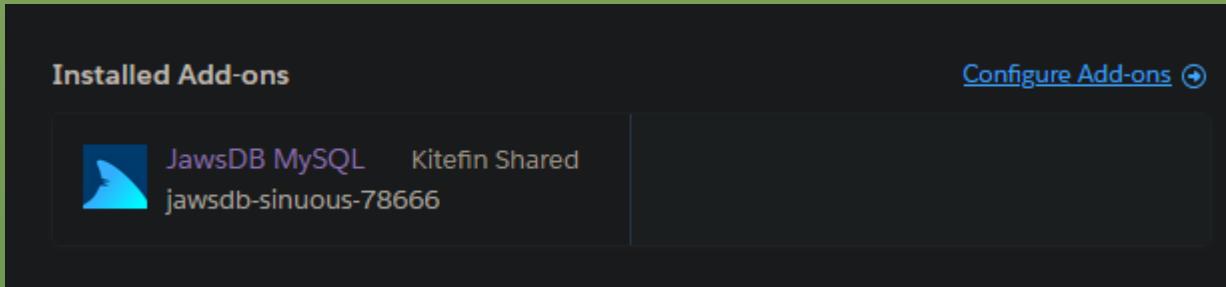
Intégration de la base de donnée



Fichier SQL



EcorideBdd

A screenshot of the Heroku web interface showing the 'Installed Add-ons' section. It lists a single add-on: 'JawsDB MySQL' with the identifier 'jawsdb-sinuous-78666'. The JawsDB icon is a blue shark. To the right is a 'Configure Add-ons' button with a plus sign.

Contenu

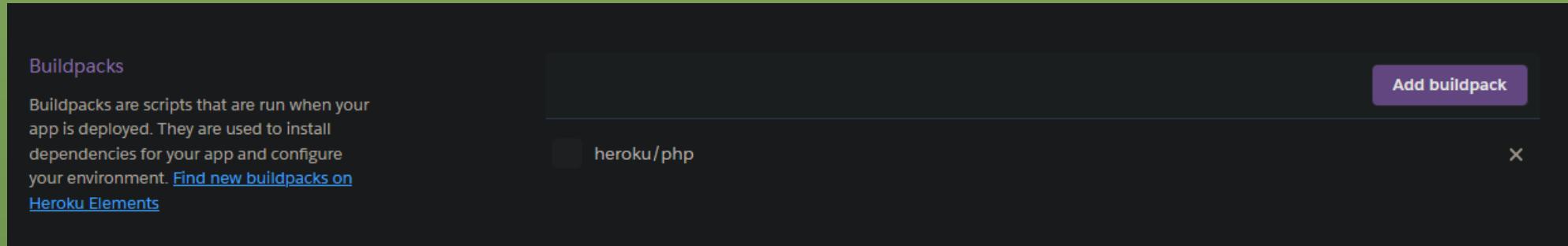
```
▷ Run | ▷ Select
CREATE TABLE `role` (
  `id` bigint(20) UNSIGNED NOT NULL,
  `label` varchar(255) NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- 
-- Déchargement des données de la table `role`
-- 

▷ Run | ▷ Select
INSERT INTO `role` (`id`, `label`, `created_at`, `updated_at`) VALUES
(1, 'ROLE_ADMIN', NULL, NULL),
(2, 'ROLE_EMPLOYEE', NULL, NULL),
(3, 'ROLE_USER', NULL, NULL),
(4, 'ROLE_CONDUCTEUR', NULL, NULL),
(5, 'ROLE_PASSAGER', NULL, NULL),
(6, 'ROLE_CHAUFFEURPASSAGER', NULL, NULL);
```

Dépendances

Builds pack

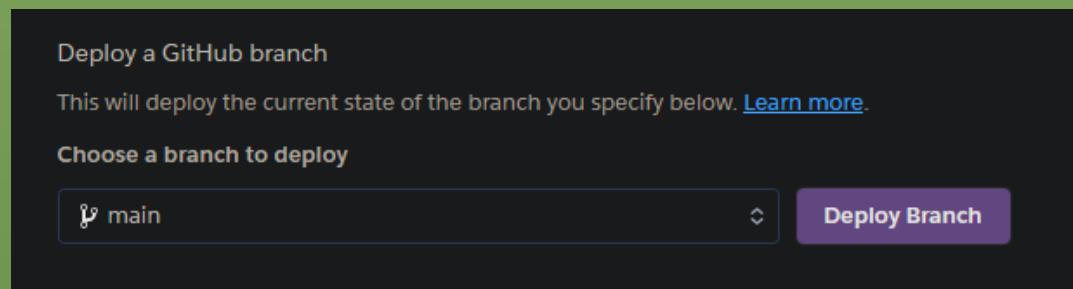
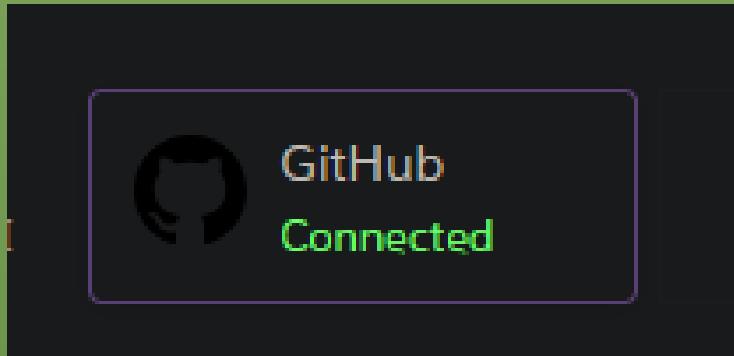


Environnement de variables



Déploiement

déploiement



```
----> Preparing platform package installation...
----> Installing platform packages...
- php (8.4.6)
- apache (2.4.63)
- composer (2.8.8)
- nginx (1.26.3)
- ext-mongodb (1.21.0)
NOTICE: detected userland polyfill packages for PHP extensions
NOTICE: now attempting to install native extension packages
Installing extensions provided by symfony/polyfill-mbstring:
- ext-mbstring...
Installing extensions provided by symfony/polyfill-ctype:
- ext-ctype...
----> Installing dependencies...
Composer version 2.8.8 2025-04-04 16:56:46
Installing dependencies from lock file
Verifying lock file contents can be installed on current platform.
Warning: The lock file is not up to date with the latest changes in composer.json. You may be getting outdated dependencies. It is recommended that you run `composer
```

```
55 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
----> Preparing runtime environment...
NOTICE: No Procfile, using 'web: heroku-php-apache2'.
----> Checking for additional extensions to install...
----> Discovering process types
      Procfile declares types -> web
----> Compressing...
      Done: 26.7M
----> Launching...
      Released v4
      https://ecoridegecko-02c5182522d3.herokuapp.com/ deployed to Heroku
```



Conclusion

- ✓ D'avoir plusieurs casquettes sur 1 projet d'application
- ✓ Utiliser un autre Framework pour appliquer les acquis et corriger les lacunes des autres projets
- ✗ D'avancer sur le projet et la documentation seul tout en ayant un travail à coté
 - ✗ Impression d'avoir un Front end avec peu de JavaScript
 - Travailler sur un projet en équipe
 - Réaliser un projet orienté Front end (React, angular)

