



# Présentation

Bruno Rojo



# **Titre Professionnel**

## Développeur web et web mobile

### Niveau 5

## Projet “Touche pas au klaxon”

Développé sous PHP 8

Bruno Rojo Janvier 2026



# Descriptions

## Touche pas au klaxon

Vise à résoudre une problématique de transport entre les agences française de la société.

Souvent un seul employé par véhicule.

Et plusieurs véhicule pour le même trajet.



# Description

Concevoir une application interne complète  
permettant au employé de covoiturer  
entre les agences

# Sommaire

```
    <ul>
      <li><a href="index.html">Home</a>
      <li><a href="home-events.html">Events</a>
      <li><a href="multi-col-menu.html">Multi Col</a>
      <li class="has-children"> <a href="#> More
        <ul>
          <li><a href="#">Tall Buttons</a>
          <li><a href="#">Image Logos</a>
          <li class="active"><a href="#">Variables</a>
        </ul>
      </li>
      <li class="has-children"> <a href="#> More
        <ul>
          <li><a href="#">Variable Data</a>
        </ul>
      </li>
    </ul>
```

1. Cahier des charges
  2. Outil organisationnel
  3. Maquettage
  4. Modélisation
  5. Développement

## Conclusion



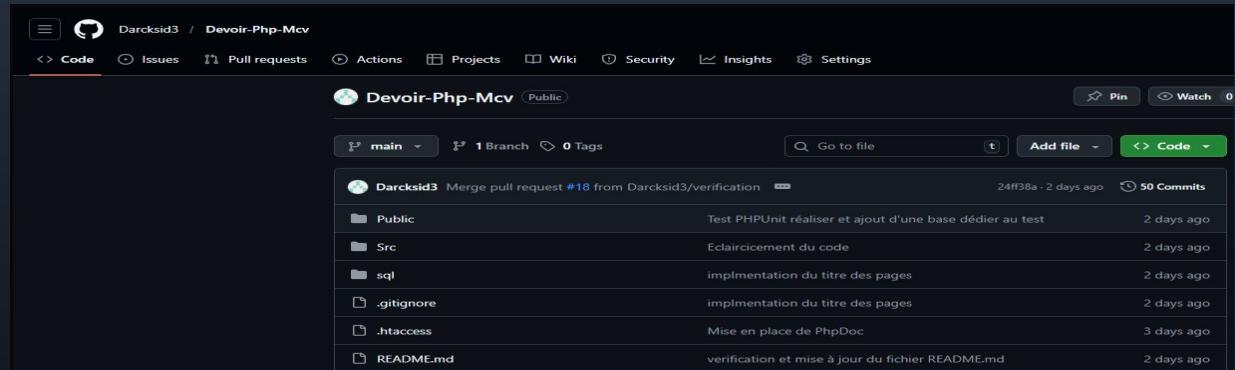
## **fonctionnement**

L'employé se connecte et cherche un trajet  
lui correspondant ou créer un trajet afin d'en  
faire bénéficier les autres employés

## **Finalité**

Le rôle du site est de permettre aux employés de se mettre en relation pour améliorer la gestion des véhicules de l'entreprise

# Outil organisationnel

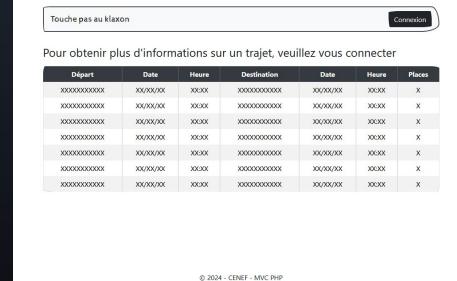
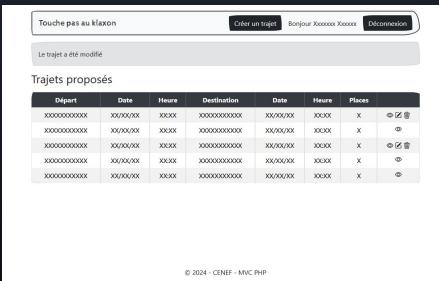


## Versioning du projet sur GitHub

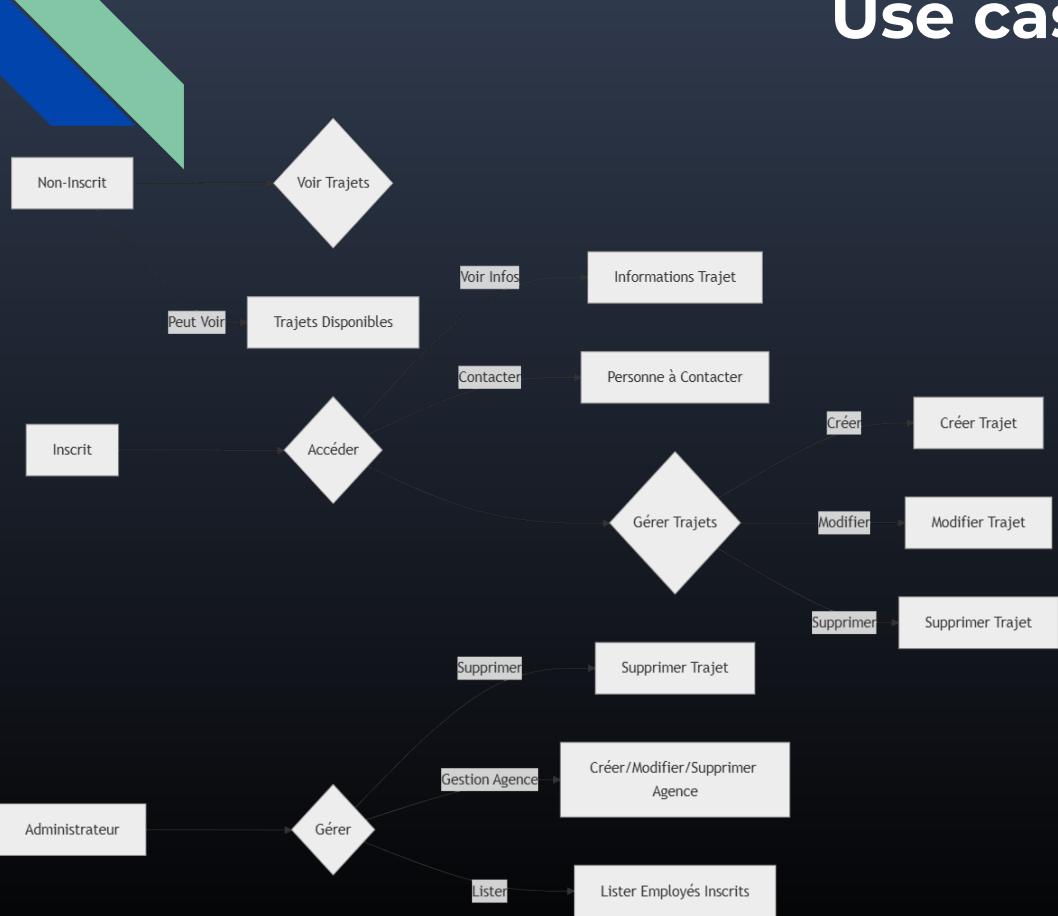
# Maquettage

L'apparence n'a pas été au centre du projet .

Les vue de rendu ont été fournis ainsi qu'un code couleur.



# Use case



- Les non inscrit
  - peuvent voir les trajets disponibles
- Les inscrit
  - voir les information d'un trajet et la personne à contacter pour en profiter
  - peuvent créer/modifier/supprimer leur propre trajet
- L'administrateur pourrait
  - supprimer un trajet
  - créer/modifier/supprimer une agence
  - lister les employé inscrit



# Modélisation

Séparation des utilisateurs de la base de donnée.



Système  
d'enregistrement



Application



Administrateur



# Modélisation

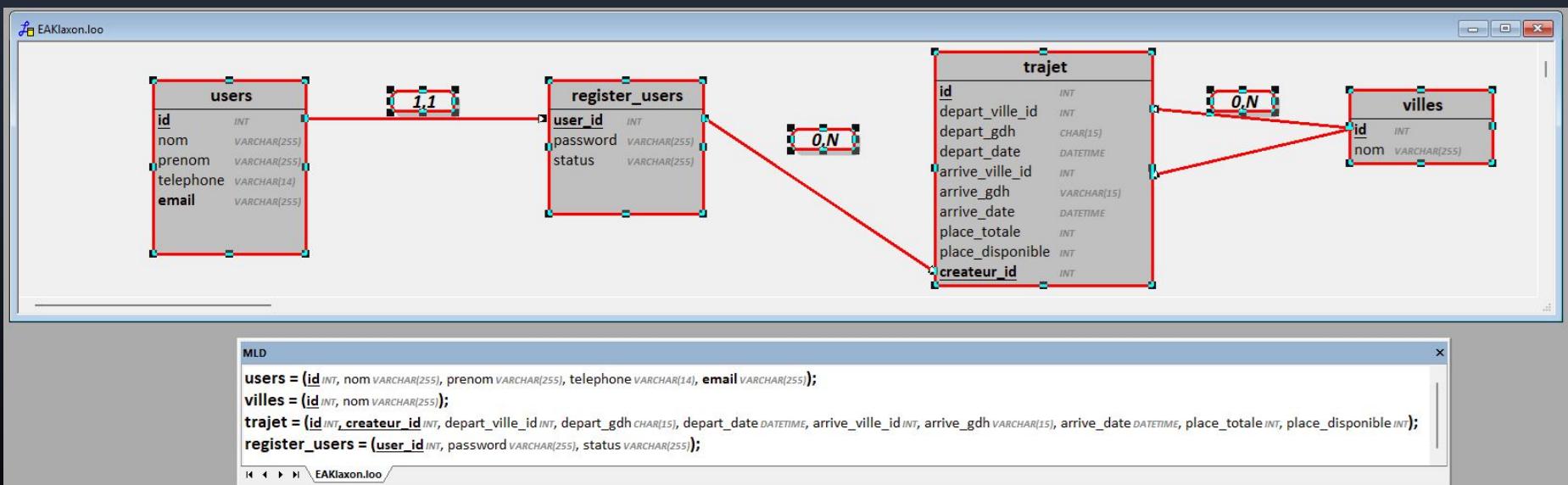
la base de donnée contient 4 tables avec des relations entre elles.

La modélisation des futures tables était donc nécessaire, avant de les créer.

Pour les modéliser, j'ai utilisé la méthode "**Merise**" et le logiciel **Looping**.

Il permet de créer un diagramme de tables (le MCD et MLD).

# MCD / MLD





# Développement

## LOCAL

Sous window

en utilisant WAMP

Sous linux/manjaro

En utilisant apache,php,mariadb

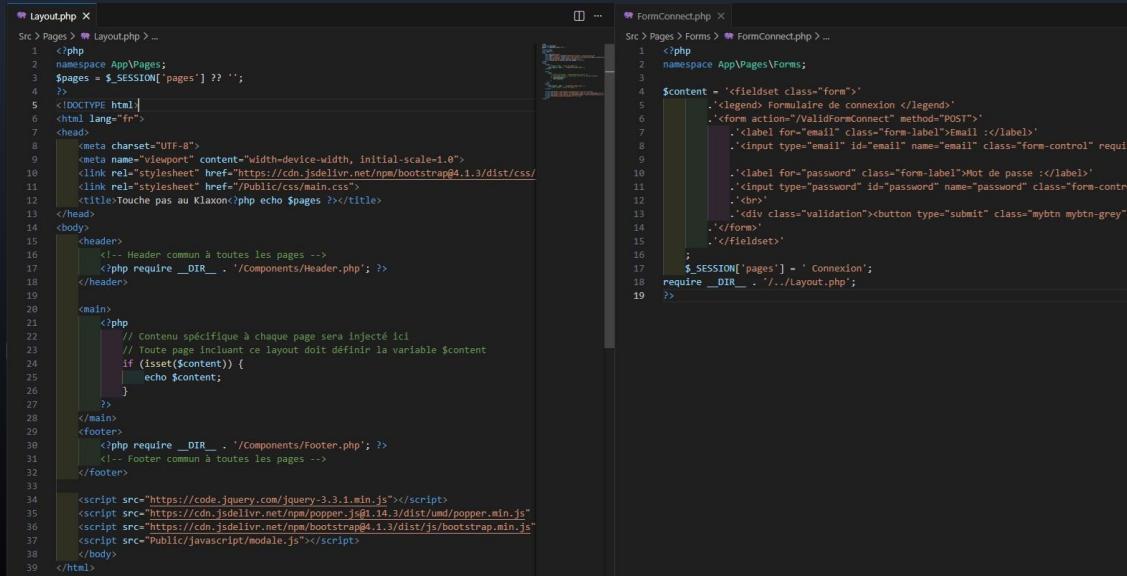
Utilisation de composer :

Autoloader Gestion de dépendance (Dot Env)

# Développement

## Le layout

Création d'un layout qui recevras les vue par injection d'une variable



The image shows a code editor with two tabs open. On the left is `Layout.php` and on the right is `FormConnect.php`. Both files are written in PHP.

```
Src > Pages > Layout.php
1 <?php
2 namespace App\Pages;
3 $pages = $_SESSION['pages'] ?? '';
4 ?>
5 <!DOCTYPE html>
6 <html lang="fr">
7 <head>
8     <meta charset="UTF-8">
9     <meta name="viewport" content="width=device-width, initial-scale=1.0">
10    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/css/bootstrap.min.css" />
11    <link rel="stylesheet" href="Public/css/main.css" />
12    <title>Touche pas au Klaxon</title>
13 </head>
14 <body>
15     <header>
16         <!-- Header commun à toutes les pages -->
17         <?php require __DIR__ . '/Components/Header.php'; ?>
18     </header>
19
20     <main>
21         <?php
22             // Contenu spécifique à chaque page sera injecté ici
23             // Toute page incluant ce layout doit définir la variable $content
24             if (isset($content)) {
25                 echo $content;
26             }
27         ?>
28     </main>
29     <footer>
30         <?php require __DIR__ . '/Components/Footer.php'; ?>
31         <!-- Footer commun à toutes les pages -->
32     </footer>
33
34     <script src="https://code.jquery.com/jquery-3.3.1.min.js"></script>
35     <script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.3/dist/umd/popper.min.js" />
36     <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/js/bootstrap.min.js" />
37     <script src="Public/javascript/module.js"></script>
38 </body>
39 </html>
```

```
Src > Pages > Forms > FormConnect.php
1 <?php
2 namespace App\Pages\Forms;
3
4 $content = <fieldset class="form">
5     <legend> Formulaire de connexion </legend>
6     <form action="/ValidFormConnect" method="POST">
7         <label for="email" class="form-label">Email </label>
8         <input type="email" id="email" name="email" class="form-control" required>
9
10        <label for="password" class="form-label">Mot de passe </label>
11        <input type="password" id="password" name="password" class="form-control" />
12        <br>
13        <div class="validation"><button type="submit" class="mybtn mybtn-grey">Co
14    </div>
15 </form>
16 </fieldset>
17 ;
18 $_SESSION['pages'] = ' Connexion';
19 ?>
```

# Développement

## Le router

l'utilisation d'un router été demander

j'avais le choix d'utiliser une librairie existante

mais une de mes récente lecture m'est revenue php8 et match()

c'est la solution que j'ai choisi mené par la hype et mon envie de tester  
création du routeur qui donnerait accès au vue et au controllers

```
function dynamicPath($path,$finalpath, &$matches ) {
    $patch = preg_replace('#([a-zA-Z]+\#)', '(\^/)+?', $path);
    return preg_match("#^($patch)$#", $finalpath, $matches) >= 1 ? $finalpath : '///';
}

/* Mise en place du routeur
$callable = match($finalpath) {
    'GET:' /> function() {
        require __DIR__ . '/Src/Pages/Home.php';
    },
    /* FORMULAIRES
    // Connexion
    'GET:/FormConnect' /> function() {
        require __DIR__ . '/Src/Pages/Forms/FormConnect.php';
    },
    'POST:/ValidFormConnect' /> function() {
        require __DIR__ . '/Src/Controllers/ValidFormConnect.php';
    },
    // Trajets
    'GET:/FormTrajet' /> function() {
        require __DIR__ . '/Src/Pages/Forms/FormTrajet.php';
    },
    dynamicPath('POST:/ValidFormTrajet/{id}', $finalpath, $matches) /> function () use ($matches) {
        require __DIR__ . '/Src/Controllers/ValidFormTrajet.php';
    },
}
```



# Développement

## Environnement

### Mise en place des variable d'environnement

C'est la base de tout notre site. Sans celle-ci, rien n'est possible.

Il faut d'abord créer le fichier ".env".

```
DB_HOST="localhost"
DB_NAME="covoiturage_interne"
DB_USER_RS="r_service"
DB_PASSWORD_RS=
```



# Développement

## Base de donnée

Grâce au travail de modélisation qui a été fait en amont (Méthode Merise), j'ai connaissance du nombre de tables dont j'ai besoin, ainsi que le nom de chaque champ et les réalisations entre les tables.

j'ai donc pu créer le script sql pour la création de la base et de ces table en ayant déjà toute les informations.

# Rendu du script de création

```
36  CREATE TABLE IF NOT EXISTS `trajet` (
37      `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
38
39      -- Clé étrangère vers la table `ville` pour la ville de départ
40      `depart_ville_id` INT UNSIGNED NOT NULL,
41
42      `depart_gdh` VARCHAR(15) NOT NULL,
43      `depart_date` DATETIME NOT NULL,
44
45      -- Clé étrangère vers la table `ville` pour la ville d'arrivée
46      `arrive_ville_id` INT UNSIGNED NOT NULL,
47
48      `arrive_gdh` VARCHAR(15) NOT NULL,
49      `arrive_date` DATETIME NOT NULL,
50
51      `place_totale` INT NOT NULL,
52      `place_disponible` INT NOT NULL,
53
54      -- Clé étrangère vers la table `utilisateurs` (le créateur du trajet)
55      `createur_id` INT UNSIGNED NOT NULL,
56
57      PRIMARY KEY (`id`),
58
59      -- Contrainte pour la ville de départ
60      FOREIGN KEY (`depart_ville_id`)
61          REFERENCES `ville` (`id`)
62          ON UPDATE CASCADE ON DELETE RESTRICT, -- RESTRICT empêche la suppression
63
64      -- Contrainte pour la ville d'arrivée
65      FOREIGN KEY (`arrive_ville_id`)
66          REFERENCES `ville` (`id`)
67          ON UPDATE CASCADE ON DELETE RESTRICT,
68
69      -- Contrainte pour le créateur du trajet
70      FOREIGN KEY (`createur_id`)
71          REFERENCES `utilisateur` (`id`)
72          ON UPDATE CASCADE ON DELETE RESTRICT -- RESTRICT empêche la suppression
73  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

# Rendu PHPmyAdmin

PHPMyAdmin interface showing two database tables:

**Table: trajet**

ID	depart_ville_id	depart_gdh	depart_date	arrive_ville_id	arrive_gdh	arrive_date	place_totale	place_disponible	createur_id
2	9	161426Z JAN 26	2026-01-16 14:26:00	4	161626Z JAN 26	2026-01-16 16:26:00	4	0	1
3	10	211300Z JAN 26	2026-01-21 13:00:00	1	211600Z JAN 26	2026-01-21 16:00:00	3	2	2
4	9	161731Z JAN 26	2026-01-16 17:31:00	3	161831Z JAN 26	2026-01-16 18:31:00	4	4	1
5	9	291000Z JAN 26	2026-01-29 10:00:00	4	291300Z JAN 26	2026-01-29 13:00:00	6	4	1
6	8	031000Z FEB 26	2026-02-03 10:00:00	5	031300Z FEB 26	2026-02-03 13:00:00	3	2	2
7	3	171341Z JAN 26	2026-01-17 13:41:00	8	171441Z JAN 26	2026-01-17 14:41:00	4	2	3
8	4	280800Z JAN 26	2026-01-28 08:00:00	3	281500Z JAN 26	2026-01-28 15:00:00	4	2	2
9	2	040800Z FEB 26	2026-02-04 08:00:00	9	041700Z FEB 26	2026-02-04 17:00:00	2	2	2

**Table: utilisateur**

Action	Rows	Type	Collation	Size	Overhead
Browse Structure Search Insert Empty Drop	8	InnoDB	utf8mb4_ucs2_ci	64.0 Kib	-
Browse Structure Search Insert Empty Drop	20	InnoDB	utf8mb4_ucs2_ci	32.0 Kib	-
Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_ucs2_ci	16.0 Kib	-
Browse Structure Search Insert Empty Drop	12	InnoDB	utf8mb4_ucs2_ci	32.0 Kib	-
Sum	43	InnoDB	utf8mb4_ucs2_ci	144.0 Kib	0 B



# Développement

## La classe de connexion

Création d'une class abstract pour la connexion a la base.

Utilisation de l'objet PDO.

Elle sera la classe parente de mes opération CRUD.

Et de gestion des utilisateurs de la base .



# Développement

## Les classes d'interaction

Utilisation du “**CRUD**” pour manipuler les données des tables.

Grâce au formulaire et au controller j'appelle les classe spécifique .

à chaque opération en base de donnée.

Chaque classe est spécifique à une opération .

Elles hérite de la classe de connexion.



# Développement

## Utilisation et importance

Les classes utilise PDO

Sa mise en forme avec préparation des mes requête en amont

Réduction des risques d'injection SQL



# Développement

## Affichage

ayant eu des vue associer au projet la structure d'affichage était la suivante

image de la vue

note : une table simple faisant apparaître les information des trajet des option en fonction du statut de l'utilisateur



# Développement

## Le formulaire

Alimentation par formulaire, création et modification . la vue s'adapte

Inclusion d'informations personnelles du créateur du formulaire

Villes alimenter en amont par un select de la table ville

Les champs datetime augmenter

+1 H sur l'heure de départ et +2H sur l'heure d'arrivé



# Développement

## Traitement des données

Toutes les information de la table ne sont pas insérable via le formulaire

Les horaires GDH de départ et d'arrivé à calculer après que les horaire ne soit validé



# Développement

## Les controllers

Le controllers

récupération des information

vérification du format

fonction annexe

il sert a aussi à vérifier si c'est une création une modification ou une suppression qui est demander par l'utilisateur

# Développement

## Suppression Sécurisation

La suppression est redirigé vers une page de confirmation

Touche pas au klaxon

Utilisateurs

Agences

Trajets

Bonjour Martin Alexandre

Déconnexion

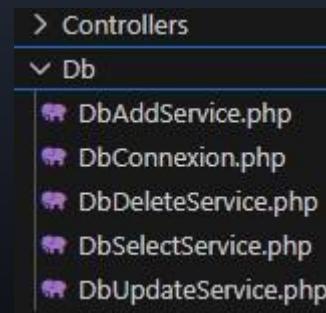
Page de confirmation de suppression de trajet

Confirmer la suppression

Touche pas au Klaxon © Darckside 2026

# Développement

Utilisation des classes d'accès à la base de donnée





# Développement

Les jointures pour la sélection.

exemple:

```
$sql = "select t.*,
v1.nom as depart_ville_nom,
v2.nom as arrive_ville_nom
from trajet t
inner join ville v1 on t.depart_ville_id = v1.id
inner join ville v2 on t.arrive_ville_id = v2.id
where t.depart_date >= NOW() and t.place_disponible > 0 order by depart_date ASC";
```

# Développement

## Sécurisation des données grâce à l'objet PDO

```
/** * Ajout d'une nouvelle agence * @param string $ville * @throw Exception erreur de connection à la base de donnée * @return void */ public function addVille(string $ville): void {     $connexion = $this->connexion(2);     if (!$connexion instanceof \PDO) {         throw new \Exception("La connexion à la base de données a échoué.");     }     $sql = "insert into ville (nom) values (:ville)";     $query = $connexion->prepare((string) $sql);     $query->execute(['ville' => $ville]); }
```

# Développement

## Interaction avec l'utilisateur

les erreurs affichage du message flash

affichage de la liste

**Touche pas au klaxon**

Utilisateurs

Agences

Trajets

Bonjour Martin Alexandre

Déconnexion

Trajet Modifier avec succès



# Conclusion

Recherche constante d'information .

Un univers en constante évolution .

Autodidactisme pour constamment évoluer.

**En route vers le Concepteur développeur d'application.**