

DOSSIER DU PROJET

TOUCHE PAS AU KLAXON

BRUNO ROJO

SOMMAIRE

SOMMAIRE.....	2
INTRODUCTION.....	3
I. Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité.....	4
II. Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité.....	5
RÉSUMÉ DU PROJET.....	6
CAHIER DES CHARGES.....	7
I. Besoin et objectifs de l'application.....	7
II. Users stories.....	8
III. Arborescence.....	9
IV. MVP.....	10
V. Fonctionnalité détaillées des pages.....	11
VI. Evolution potentielles.....	13
VII. Design Ui & UX.....	15
• La charte graphique.....	15
• Les vues fournies.....	15
• La maquette.....	19
SPÉCIFICATION TECHNIQUE.....	24
Technologies.....	24
Création de la base de données.....	25
Méthode Merise.....	25
Routes front et back.....	27
PRÉSENTATION DU JEUX D'ESSAI.....	29
SÉCURITÉ, VEILLE et test.....	31
• Sécurité.....	31
• Test.....	32
CONCLUSION.....	35

INTRODUCTION

Depuis tout jeune, je suis passionné par les technologies et en particulier la programmation. Bien qu'ayant suivi un parcours m'éloignant de l'univers de l'informatique, j'ai continué pendant longtemps à m'auto former sur mon temps libre sur les sujets liés à la programmation. Grâce à cet apprentissage individuel, j'ai pu continuer la pratique du code et à suivre régulièrement les actualités du secteur.

Après un certain problème de santé dans mon précédent domaine d'activité, j'ai dû choisir de trouver une nouvelle voie qui me permettrait de valider l'acquisition d'une certaine expérience. Après quelques recherches sur les moyens d'y parvenir, j'ai décidé de m'engager auprès du centre européen de formation. J'ai effectué un test de positionnement pour accéder directement à la formation.

Au-delà de l'acquisition de nouvelles compétences et la validation d'une expérience, je souhaite également apprendre à travailler en groupe sur un projet. Ayant toujours réalisé des petits projets de façon autonome, je souhaite acquérir des bonnes pratiques pour la réalisation d'un travail de groupe, et comprendre l'organisation et le fonctionnement d'une équipe pour un projet.

I. Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

A. Maquetter une application

En amont de la réalisation à proprement parler de cette application, j'ai choisi dans un premier temps de travailler sur la réalisation des documents de conception. Dans ce but, j'ai donc réalisé en premier lieu des users stories pour définir très clairement ce que pourront faire les utilisateurs sur notre application. Puis je me suis servi des vues du projet, aux formats desktop pour imaginer la structuration des pages, puis enfin j'ai utilisé la charte graphique pour les maquettes afin de voir concrètement à quoi allait ressembler cette application.

B. Réaliser une interface utilisateur web statique et adaptable

Ce projet étant conçu comme un outil quotidien à la proposition de trajet de covoiturage entre agence au sein de l'entreprise. Au vu de la diversité probable des employés, notamment au niveau de l'âge, il était nécessaire d'avoir une interface simple et facile d'utilisation, peu importe le niveau de compétence en informatique. De plus, cette application est utile "sur le terrain" pour les employés.

C. Développer une interface utilisateur web

Au vu des users stories réalisées en amont de la réalisation du projet, il n'était impératif que notre application dispose d'une interface dynamique, dès lors je me suis très rapidement tourné vers l'interface desktop.

II. Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité

A. Créer une base de données

Tout comme la réalisation du maquettage de l'application, j'ai voulu dès le début organiser la façon dont allaient être structurées nos données. J'ai donc commencé par réfléchir aux données que je souhaitais intégrer à l'application, puis j'ai créé un MCD, en MLD. Pour gérer la base de données de notre application, j'ai choisi d'utiliser MySql.

B. Développer les composants d'accès aux données

Pour réaliser la couche d'accès aux données de cette application, j'ai utilisé l'objet PDO.

C. Développer la partie back-end d'une application web ou web mobile

Pour réaliser le back-end de cette application, j'ai utilisé PHP 8. J'ai organisé cette application avec différentes routes, liées à des controllers. J'ai également créé plusieurs services, notamment pour l'authentification ou la validation des données.

RÉSUMÉ DU PROJET

Dans une grande entreprise qui possède des implantations géographiques multiples. De nombreux trajets sont réalisés entre les différents sites (mobilité inter-sites). Parfois, plusieurs véhicules font le même trajet le même jour avec un taux d'occupation très faible (uniquement le conducteur dans de nombreux cas).

Pour remédier à cette situation, réaliser une application permettant de diffuser au sein de l'entreprise les trajets prévus afin de favoriser le covoiturage.

Ainsi, cette application a été pensée pour permettre à l'ensemble des employés présents de se positionner en toute autonomie sur les trajets à effectuer et d'en gérer leur état. En d'autres termes, ils pourront prendre connaissance des trajets existants et y postuler en envoyant un mail au créateur du trajet, ou créer le leur et le gérer en fonction des autres employés intéressés à y participer.

Cette application sera disponible via un site web déployé sur l'intranet de l'entreprise. Dans cette première version, le besoin fonctionnel est relativement simple.

En définitive, cette solution favorise la cohérence des trajets, un gain de temps organisationnel, et un gain financier sur la gestion du parc de véhicules.

CAHIER DES CHARGES

I. Besoin et objectifs de l'application

A.Besoins

Pour mieux comprendre les besoins de l'entreprise, il a fallu effectuer des recherches en amont pour connaître l'utilité du projet. Il faut commencer par connaître l'organisation interne de cette dernière.

Les employés sont amenés à se rendre pour leur travail dans des agences d'autre villes et par conséquent prendre un véhicule afin de s'y rendre.

Les employés gèrent leur propre déplacement il ne sont donc pas au fait du planning des autres.

B.Objectifs

Nous comprenons que pour optimiser l'utilisation des véhicules les informations à prendre en compte sont :

- La liste des trajets disponibles entre les agences.
- L'heure de départ et d'arrivée pour chaque trajet.
- Le nombre de places disponibles.
- Pouvoir contacter le créateur d'un trajet.
- Les employés organisent leur trajet entre eux.

II. Users stories

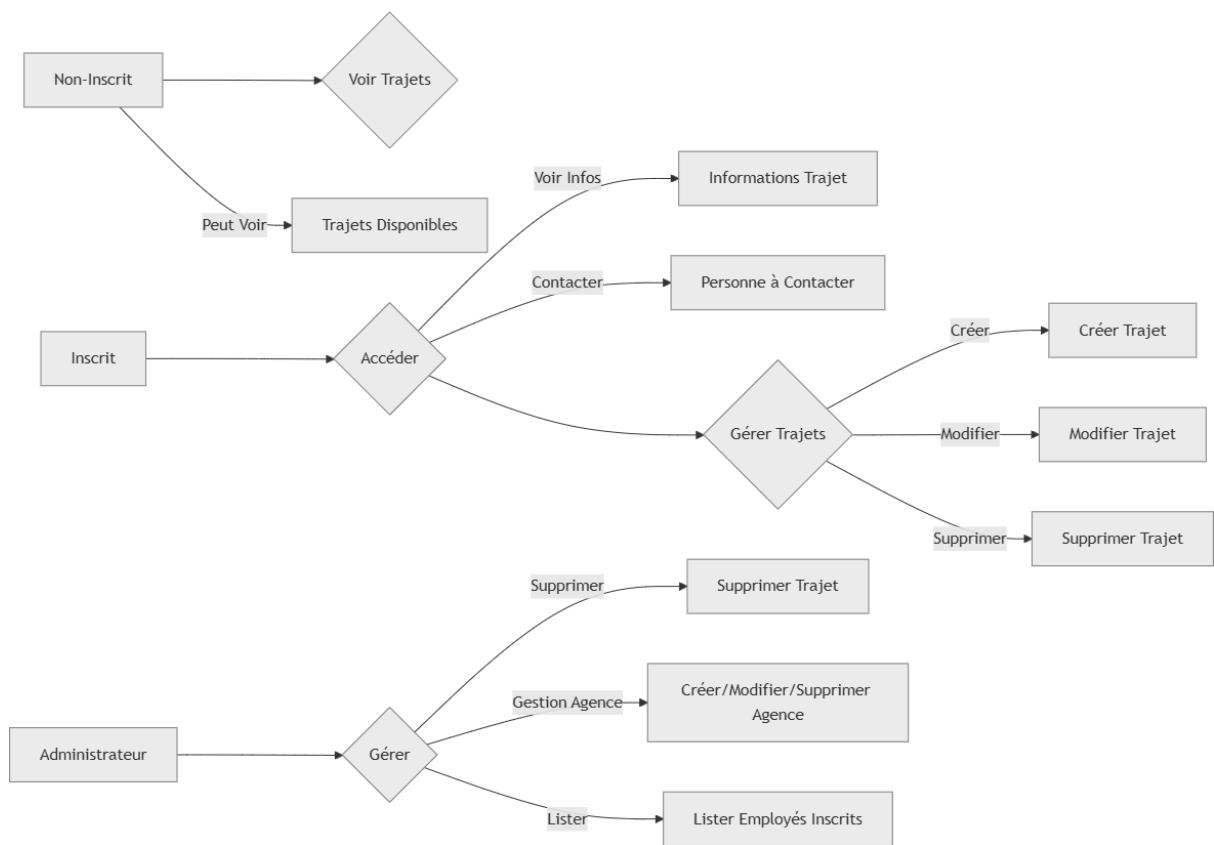
Légende:

MVP

Version à venir

En tant que	Je souhaite	Afin de
visiteur	consulter la liste des trajets à venir	voir si un trajet correspond à mes attentes
visiteur	me connecter à l'appli	accéder au différent service
visiteur	me déconnecter de l'appli	pour laisser la place à d'autre utilisateur
utilisateur	voir les informations d'un trajet	pouvoir contacter son propriétaire
utilisateur	créer un trajet	pouvoir faire profiter de mon déplacement
utilisateur	avoir une liste des villes	pouvoir les créer sans fautes et rapidement
utilisateur	avoir le temps de trajet entre les agences	définir plus facilement l'heure d'arrivée
utilisateur	modifier un trajet	en cas d'erreur ou souscription
utilisateur	supprimer un trajet	si le déplacement n'est plus d'actualité
administrateur	lister les utilisateurs du service	afin d'en connaître leur nombres
administrateur	lister les agences	pour vérifier si elles sont toutes répertoriées
administrateur	créer une agence	de répertorier les nouvelles agences
administrateur	avoir une liste des villes	pouvoir les créer sans fautes et rapidement
administrateur	modifier une agence	pouvoir intervenir en cas de changement
administrateur	supprimer une agence	pouvoir intervenir en cas de fermeture
administrateur	lister les trajets	pouvoir les répertorier et les traiter
administrateur	supprimer les trajets	pouvoir alléger la base de données

III. Arborescence



IV. MVP

Mon produit Minimum viable s'est naturellement construit autour de la gestion des trajets, ils sont au cœur de l'application. Pour cela une authentification sécurisée est primordiale, ainsi que les fonctionnalités lui à l'administrateur qui permette que les informations donne au employé soit disponible et mise à jour.

Voici donc les fonctionnalité présenté dans mon MVP:

- L'utilisateur doit pouvoir consulter la liste des trajets disponible
- L'utilisateur (employé ou administrateur) doit pouvoir s'authentifier avec un email et un mot de passe.

A. Employé

- Pour afficher les informations d'un trajet, il doit cliquer sur le bouton du trajet concerné.
- Sur la fiche du trajet, il doit pouvoir consulter les informations (information sur le créateur, place disponible) et valider la fermeture par un bouton.
- Pour modifier un trajet, il doit en être le créateur.
- Pour supprimer un trajet, il doit en être le créateur et confirmer la demande de suppression sur le bouton concerné.

dans le menu:

- Il doit cliquer sur le bouton concerné pour créer un trajet.
- Il doit cliquer sur le bouton concerné pour se déconnecter

B. Administrateur

- Arrive sur la page des trajets il doit pouvoir:
- Avoir un menu spécialisé pour:
 - Consulter la liste des employé enregistré
 - Consulter la liste totale des trajets
 - Supprimer les trajets.
 - Accéder au menu des agences
 - créer une agence
 - Modifier une agence
 - Supprimer une agence

V. Fonctionnalité détaillées des pages

Les pages/routes sont sécurisées et donc accessibles seulement à partir du moment où l'utilisateur est connecté. Seules les pages d'accueil et de connexion sont accessibles par tout le monde.

> Page de connexion :

Pour le moment l'utilisateur n'a pas l'accès au menu. Il doit compléter les champs :

- email
- mot de passe

Il valide son authentification grâce à un bouton. Il accède à la page d'accueil, le menu spécifique s'active.

> Page accueil :

Cette page est donc propre à tout le monde. Elle liste les trajets à venir ou il reste des places de disponible. Dans la partie navigation, il peuvent maintenant accéder à leur menu spécifique et la liste des trajet apporte l'option de visualisation du trajet ouvrant une fenêtre modale ainsi que la modification/suppression si ils sont créateurs de trajet en cours.

➤ Page Modale :

Un popup laissant apparaître le nom prénom email numéro de téléphone du créateur du trajet ainsi que le nombre de place restaurant pour le trajet.

➤ Formulaire de trajet :

Le formulaire de trajet comporte les champs suivants :

- nom
- prénom
- email
- numéro de téléphone
- agence de départ
- date et heure de départ
- agence d'arrivée
- date et heure d'arrivée
- nombre de place totale
- nombre de place disponible

Le formulaire est modulaire il se précharge des informations personnelles de l'utilisateur et il s'adapte en fonction des conditions:

- Création d'un trajet si aucun argument n'est fourni
- Modification/suppression si les conditions sont remplies (le trajet existe et l'utilisateur est le créateur du trajet les champs sont préremplis)

➤ Page de suppression de trajet :

Cette page accessible uniquement après avoir demandé la suppression d'un trajet comporte un bouton permettant de valider définitivement sa suppression.

➤ Page liste des employés enregistré (administrateur) :

TOUCHE PAS AU KLAXON



Cette page offre une liste des employés enregistrés sur l'application avec leur information personnelle.

➤ Liste des trajets (administrateur) :

Cette page liste tous les trajets de la base de données, il revient à l'administrateur de supprimer les trajets dépassés grâce au bouton à cliquer en face de chaque trajet.

➤ Pages des agences (administrateur) :

Cette page comporte deux formulaires.

- Le formulaire de création d'agence contient un champ de type texte :
 - Nom de la villeet un bouton de validation.
- Le formulaire de modification/suppression contient un select avec la liste des ville de la base de donnée; un champ de type texte :
 - Nouveau nom de la villeet les boutons modifier et supprimer.

VI. Evolution potentielles

Une intégration interactive en JavaScript pourrait faciliter l'usage des formulaires.

A. Utilisateur

Une interaction avec une API externe pourrait faciliter la création du trajet en apportant le temps de trajet entre les 2 villes sélectionnées et par la même remplir automatiquement le champ de l'heure d'arrivée. L'API de **nominatim openstreetmap** (<https://nominatim.openstreetmap.org>) qui transforme le nom des villes en donnée de géolocalisation, couplé à l'API du **projet osrm** (<https://project-osrm.org>) qui permet de calculer la distance et le temps de trajet entre deux localisations. En revanche pour éviter de nombreux appel à l'api une nouvelle table pourrait être créé

TOUCHE PAS AU KLAXON



enregistrant les trajet déjà demandé afin de pouvoir avoir un résultat local

B. Administrateur

Grace a l'API GEO.API.GOUV.FR

(<https://geo.api.gouv.fr/decoupage-administratif/communes>)

il seraient possible lors de la création/modification d'une agence de récupérer les villes correspondante au caractère remplis par l'administrateur et donc d'éviter les fautes d'orthographes et erreur de saisie.

VII. Design Ui & UX

Les vues et la charte graphique ayant été fournies dès le départ du projet j'ai pu passer directement à la conception des maquettes.

- La charte graphique

#f1f8fc	#0074c7	#00497c	#384050	#cd2c2e	#82b864

- Les vues fournies.

Visiteur

Touche pas au klaxon
Connexion

Pour obtenir plus d'informations sur un trajet, veuillez vous connecter

Départ	Date	Heure	Destination	Date	Heure	Places
xxxxxxxxxx	XX/XX/XX	XX:XX	xxxxxxxxxx	XX/XX/XX	XX:XX	X
xxxxxxxxxx	XX/XX/XX	XX:XX	xxxxxxxxxx	XX/XX/XX	XX:XX	X
xxxxxxxxxx	XX/XX/XX	XX:XX	xxxxxxxxxx	XX/XX/XX	XX:XX	X
xxxxxxxxxx	XX/XX/XX	XX:XX	xxxxxxxxxx	XX/XX/XX	XX:XX	X
xxxxxxxxxx	XX/XX/XX	XX:XX	xxxxxxxxxx	XX/XX/XX	XX:XX	X
xxxxxxxxxx	XX/XX/XX	XX:XX	xxxxxxxxxx	XX/XX/XX	XX:XX	X
xxxxxxxxxx	XX/XX/XX	XX:XX	xxxxxxxxxx	XX/XX/XX	XX:XX	X

© 2024 - CENEF - MVC PHP

TOUCHE PAS AU KLAXON



Centre Européen
de Formation
Assurez votre succès

Utilisateur

Touche pas au klaxon

Créer un trajet

Bonjour Xxxxxxx Xxxxxxx

Déconnexion

Trajets proposés

Départ	Date	Heure	Destination	Date	Heure	Places	
XXXXXXXXXXXX	XX/XX/XX	XX:XX	XXXXXXXXXXXX	XX/XX/XX	XX:XX	X	⊕ ✖️ 🗑
XXXXXXXXXXXX	XX/XX/XX	XX:XX	XXXXXXXXXXXX	XX/XX/XX	XX:XX	X	⊕
XXXXXXXXXXXX	XX/XX/XX	XX:XX	XXXXXXXXXXXX	XX/XX/XX	XX:XX	X	⊕ ✖️ 🗑
XXXXXXXXXXXX	XX/XX/XX	XX:XX	XXXXXXXXXXXX	XX/XX/XX	XX:XX	X	⊕
XXXXXXXXXXXX	XX/XX/XX	XX:XX	XXXXXXXXXXXX	XX/XX/XX	XX:XX	X	⊕

© 2024 - CENEF - MVC PHP

Modale

The screenshot shows a web browser window with a modal dialog open. The modal has a title bar "Modale". The main content area displays a table with travel information:

Départ	Date	Heure	Destination	Date	Heure	Places	
XXXXXXXXXX	XX/XX/XXXX				X	X	(edit)
XXXXXXXXXX	XX/XX/XXXX				X	X	(view)
XXXXXXXXXX	XX/XX/XXXX				X	X	(edit)
XXXXXXXXXX	XX/XX/XXXX				X	X	(view)
XXXXXXXXXX	XX/XX/XXXX				X	X	(view)

Below the table, there is a message: "Pour obtenir plus d'informations sur un trajet, veuillez vous connecter". A tooltip or callout box is overlaid on the table, containing the following user details:

- Auteur : XXXXXX XXXXXX
- Téléphone : XX XX XX XX XX
- Email : XXXXXXXXX@XXXXXX.XX
- Nombre total de places : X

At the bottom right of the modal is a "Fermer" button.

At the bottom of the page, there is a footer: "© 2024 - CENEF - MVC PHP"

TOUCHE PAS AU KLAXON



Message Flash

Touche pas au klaxon

Créer un trajet

Bonjour Xxxxxxx Xxxxxxx

Déconnexion

Le trajet a été modifié

Trajets proposés

Départ	Date	Heure	Destination	Date	Heure	Places	
xxxxxxxxxx	XX/XX/XX	XX:XX	xxxxxxxxxx	XX/XX/XX	XX:XX	X	⊕ ✎ 🗑
xxxxxxxxxx	XX/XX/XX	XX:XX	xxxxxxxxxx	XX/XX/XX	XX:XX	X	⊕
xxxxxxxxxx	XX/XX/XX	XX:XX	xxxxxxxxxx	XX/XX/XX	XX:XX	X	⊕ ✎ 🗑
xxxxxxxxxx	XX/XX/XX	XX:XX	xxxxxxxxxx	XX/XX/XX	XX:XX	X	⊕
xxxxxxxxxx	XX/XX/XX	XX:XX	xxxxxxxxxx	XX/XX/XX	XX:XX	X	⊕

© 2024 - CENEF - MVC PHP

Navigation administrateur

Touche pas au klaxon

Utilisateurs

Agences

Trajets

Bonjour Xxxxxxx Xxxxxxx

Déconnexion

TOUCHE PAS AU KLAXON



- La maquette

Visiteur

Touche pas au klaxon

[Connexion](#)

Pour obtenir plus d'information sur un trajet, veuillez vous connecter

Départ	Date	Heure	Destination	Date	Heure	Places

Touche pas au Klaxon © Darckside 2026

Connexion

Touche pas au klaxon

[Connexion](#)

Formulaire de connexion

Email :

Mot de passe :

[Connexion](#)

Touche pas au Klaxon © Darckside 2026

TOUCHE PAS AU KLAXON



Navigation utilisateur

Touche pas au klaxon

[Créer un trajet](#)

Bonjour..

[Déconnexion](#)

Utilisateur authentifié avec succès.

Liste des trajets

Départ	Date	Heure	Destination	Date	Heure	Places	
							⊕ ✎ 🗑
							⊕
							⊕ ✎ 🗑
							⊕ ✎ 🗑

Touche pas au Klaxon © Darckside 2026

Navigation administrateur

Touche pas au klaxon

[Utilisateurs](#)[Agences](#)[Trajets](#)

Bonjour

[Déconnexion](#)

Utilisateur authentifié avec succès.

Liste des trajets

Départ	Date	Heure	Destination	Date	Heure	Places	
							⊕
							⊕ ✎ 🗑
							⊕
							⊕

Touche pas au Klaxon © Darckside 2026

TOUCHE PAS AU KLAXON



Création de trajet

Touche pas au klaxon

Utilisateurs

Agences

Trajets

Bonjour

Déconnexion

Créer un trajet

Email :

alexandre.martin@email.fr

Nom :

Martin

Prenom :

Alexandre

Telephone :

0612345678

Ville de Départ : Bordeaux

Ville d'arrivée : Bordeaux

Horaire départ

Horaire arrivée

2026-01-27T16:21

2026-01-27T17:21

Places totales

4

Places dispos

2

Créer le trajet

Touche pas au Klaxon © Darckside 2026

TOUCHE PAS AU KLAXON



Modification / suppression de trajet

Touche pas au klaxon

Utilisateurs

Agences

Trajets

Bonjour

Déconnexion

Modifier le trajet : 5

Email :

alexandre.martin@email.fr

Nom :

Martin

Prenom :

Alexandre

Telephone :

0612345678

Ville de Départ : Bordeaux

Horaire départ

2026-01-29T10:00

Ville d'arrivée : Toulouse

Horaire arrivée

2026-01-29T13:00

Places totales

6

Places dispos

3

Modifier le trajet

Supprimer le trajet

Touche pas au Klaxon © Darckside 2026

Liste des utilisateurs

Touche pas au klaxon

Utilisateurs

Agences

Trajets

Bonjour

Déconnexion

Liste des utilisateurs

Nom	Prénom	Téléphone	Email

Touche pas au Klaxon © Darckside 2026

TOUCHE PAS AU KLAXON



Gestion des agences

Touche pas au klaxon

Utilisateurs

Agences

Trajets

Bonjour

Déconnexion

Formulaires de gestion des agences

Création d'une agence

Nom de la ville :

Créer une agence

Modification/Suppression d'une agence

Sélectionner une agence! Bordeaux

Supprimer une agence

Nouveau nom de la ville :

Modifier une agence

Touche pas au Klaxon © Darckside 2026

Gestion des trajets

Touche pas au klaxon

Utilisateurs

Agences

Trajets

Bonjour

Déconnexion

Administration des trajets

Crée un trajet

Liste des trajets

Créateur du trajet	Ville de départ	Date de départ	Heure de départ	Ville d'arrivée	Date d'arrivée	Heure d'arrivée	Place Totale	Place disponible	

Touche pas au Klaxon © Darckside 2026

SPÉCIFICATION TECHNIQUE

Technologies

- **Developer experience & gestionnaire de dépendance**
 - Live Sass Compiler
 - Composer
 - Vuclas/phpdotenv
 - PHPUnit
 - PhpStan
 - PhpDocumentor
- **Front-end**
 - HTML
 - SASS
 - Bootstraps
- **Back-end**
 - PHP 8.5
 - MySQL

Création de la base de données

Méthode Merise

Merise permet de comprendre rapidement comment construire et structurer une base de données, mais aussi de communiquer de la même façon entre développeurs ou encore de se poser les bonnes questions au moment de la création du projet.

Toute base de données va donner lieu à une double représentation :

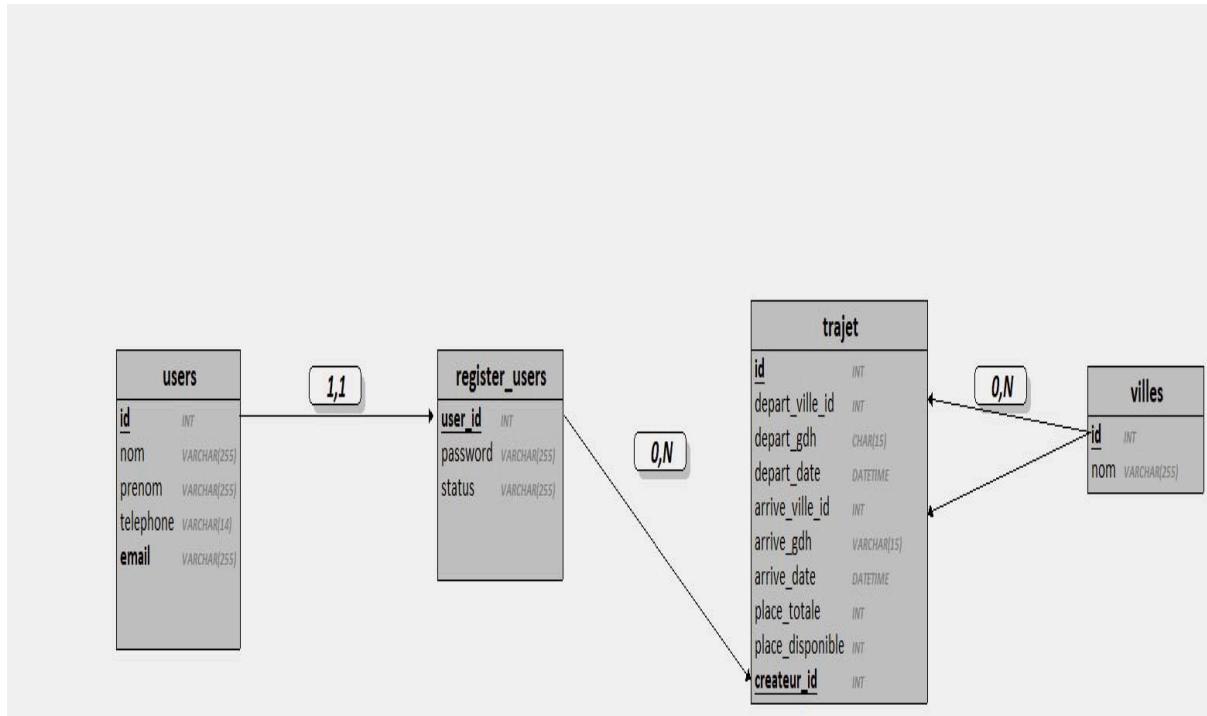
- Le Modèle Conceptuel de Données (MCD) : La représentation la plus abstraite, mais qui contient toutes les informations indispensables pour la construction de la base de données. C'est dans ce modèle que sont établies les diverses relations entre les tables .
- Le Modèle Logique des Données (MLD) : La représentation la plus proche de ce que sera la base effective, telle qu'elle sera réalisée sur la machine (présence des clefs étrangères).

Le MLD se déduit strictement du MCD d'après des règles formelles. Autrement dit, une fois le MCD réalisé, il n'y a plus besoin de réfléchir pour produire le MLD : tout se fait automatiquement. Il existe des logiciels qui proposent de réaliser le MLD d'un clic de souris, d'après le MCD .

Pour réaliser le MCD, j'ai utilisé le logiciel " JMerise "
[\(http://www.jfreesoft.com/JMerise/\)](http://www.jfreesoft.com/JMerise/) pour modéliser la base de données .

MCD et MLD

A.MCD



B. MLD

users = (id INT, nom VARCHAR(255), prenom VARCHAR(255), telephone VARCHAR(14), email VARCHAR(255));

villes = (id INT, nom VARCHAR(255));

trajet = (id INT, createur_id INT, depart_ville_id INT, depart_gdh CHAR(15),
depart_date DATETIME, arrive_ville_id INT, arrive_gdh VARCHAR(15), arrive_date
DATETIME, place_totale INT, place_disponible INT);

register_users = (user_id INT, password VARCHAR(255), status VARCHAR(255));

Routes front et back

A.Back

	ENDPOINT	OUTPUT	CONTROLLER	AUTH
Connexion				
GET	/FormConnet	Page de connexion	FormConnet.php	utilisateur
GET	/Deconnexion	Service de déconnexion	Deconnexion.php	utilisateur
POST	/ValidFormConnect	Permet de se connecter	ValidFormConnect.php	utilisateur
Trajet				
POST	/ValidFormTrajet	Création d'un trajet	ValidFormTrajet.php	utilisateur
POST	/ValidFormTrajet/{id}	Modification d'un trajet	ValidFormTrajet.php	utilisateur
GET	/ValidDeleteTrajet{id}	Suppression d'un trajet	ValidDeleteTrajet.php	utilisateur
Agence				
POST	/ValidFormAgence	Vérification du formulaire	ValidFormAgence.php	administrateur

B.Front

ROUTE	PAGE	USER
/	Accueil	utilisateur
/FormConnet	Page de connexion	utilisateur
/Modale	Fenêtre modale	utilisateur
/FormTrajet	Formulaire des trajet création	utilisateur
/FormTrajet/{id}	Formulaire des trajet modif/suppr	utilisateur
/DeleteTrajet{id}	Confirmation de suppression	utilisateur
/ListeUtilisateur	Liste des utilisateur enregistré	administrateur
/FormAgence	Formulaire de création d'agence	administrateur
/ListeTrajet	Liste de tous les trajets	administrateur
/Success	Succès de la déconnexion	utilisateur
/PageInconnue	En cas de page non trouvé	utilisateur

PRÉSENTATION DU JEUX D'ESSAI

Mon jeu d'essai comprend deux fichiers:

l'un pour simuler les employés de l'entreprise avec de fausses données personnelles et l'autre pour la liste des villes. Ils étaient préformés, les informations étaient séparées de virgule comme dans un fichier CSV.

(copie des fichier texte)

Informations personnelles	Agences
nom,prenom,telephone,email	nom
Martin,Alexandre,0612345678,alexandre.martin@email.fr	Paris
Dubois,Sophie,0698765432,sophie.dubois@email.fr	Lyon
Bernard,Julien,0622446688,julien.bernard@email.fr	Marseille
Moreau,Camille,0611223344,camille.moreau@email.fr	Toulouse
Lefèvre,Lucie,0777889900,lucie.lefeuvre@email.fr	Nice
Leroy,Thomas,0655443322,thomas.leroy@email.fr	Nantes
Roux,Chloé,0633221199,chloe.roux@email.fr	Strasbourg
Petit,Maxime,0766778899,maxime.petit@email.fr	Montpellier
Garnier,Laura,0688776655,laura.garnier@email.fr	Bordeaux
Dupuis,Antoine,0744556677,antoine.dupuis@email.fr	Lille
Lefebvre,Emma,0699887766,emma.lefebvre@email.fr	Rennes
Fontaine,Louis,0655667788,louis.fontaine@email.fr	Reims
Chevalier,Clara,0788990011,clara.chevalier@email.fr	
Robin,Nicolas,0644332211,nicolas.robin@email.fr	
Gauthier,Marine,0677889922,marine.gauthier@email.fr	
Fournier,Pierre,0722334455,pierre.fournier@email.fr	
Girard,Sarah,0688665544,sarah.girard@email.fr	
Lambert,Hugo,0611223366,hugo.lambert@email.fr	
Masson,Julie,0733445566,julie.masson@email.fr	
Henry,Arthur,0666554433,arthur.henry@email.fr	

Il a donc été assez simple de les inclure directement et sans trop de modifications dans mon script SQL d'alimentation.

TOUCHE PAS AU KLAXON



```
USE `covoiturable_interne`;

-- Insertion des villes (agences) dans la table ville
INSERT INTO ville (nom) VALUES
('Paris'),
('Lyon'),
('Marseille'),
('Toulouse'),
('Nice'),
('Nantes'),
('Strasbourg'),
('Montpellier'),
('Bordeaux'),
('Lille'),
('Rennes'),
('Reims');

-- Insertion des utilisateurs dans la table utilisateur
INSERT INTO utilisateur (nom, prenom, telephone, email) VALUES
('Martin', 'Alexandre', '0612345678', 'alexandre.martin@email.fr'),
('Dubois', 'Sophie', '0698765432', 'sophie.dubois@email.fr'),
('Bernard', 'Julien', '0622446688', 'julien.bernard@email.fr'),
('Moreau', 'Camille', '0611223344', 'camille.moreau@email.fr'),
('Lefèvre', 'Lucie', '0777889900', 'lucie.lefeuvre@email.fr'),
('Leroy', 'Thomas', '0655443322', 'thomas.leroy@email.fr'),
('Roux', 'Chloé', '0633221199', 'chloe.roux@email.fr'),
('Petit', 'Maxime', '0766778899', 'maxime.petit@email.fr'),
('Garnier', 'Laura', '0688776655', 'laura.garnier@email.fr'),
('Dupuis', 'Antoine', '0744556677', 'antoine.dupuis@email.fr'),
('Lefebvre', 'Emma', '0699887766', 'emma.lefebvre@email.fr'),
('Fontaine', 'Louis', '0655667788', 'louis.fontaine@email.fr'),
('Chevalier', 'Clara', '0788990011', 'clara.chevalier@email.fr'),
('Robin', 'Nicolas', '0644332211', 'nicolas.robin@email.fr'),
('Gauthier', 'Marine', '0677889922', 'marine.gauthier@email.fr'),
('Fournier', 'Pierre', '0722334455', 'pierre.fournier@email.fr'),
('Girard', 'Sarah', '0688665544', 'sarah.girard@email.fr'),
('Lambert', 'Hugo', '0611223366', 'hugo.lambert@email.fr'),
('Masson', 'Julie', '0733445566', 'julie.masson@email.fr'),
('Henry', 'Arthur', '0666554433', 'arthur.henry@email.fr');
```

SÉCURITÉ, VEILLE ET TEST

● Sécurité

1. Protection contre les Injections SQL

J'utilise l'objet PDO avec ces méthodes `prepare()` et `execute()`, ce qui est la règle d'or. Cela sépare les instructions SQL des données envoyées par l'utilisateur.

Le principe : On ne concatène jamais une variable utilisateur directement dans la chaîne SQL. On utilise des "placeholders" (`?` ou `:nom`).

La validation du typage :

Dans mes méthodes d'insertion ou d'update, je force le typage avec `(int)`, `(string)` ... ce qui ajoute une barrière supplémentaire.

exemple:

```
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]);
}
```

Pourquoi ce code est sécurisé ?

L'utilisation de `prepare()` et `execute()` sépare la **structure** de la requête des **données**.

1. **Le "Mur" de sécurité :** Quand j'envoie `sql à prepare()`, la base de données compile le plan d'exécution.
2. **Neutralisation :** Si un utilisateur malveillant essaie de saisir une ville nommée `'Paris');` `DROP TABLE ville;--`, le moteur PDO traitera toute cette chaîne comme un simple texte (le nom de la ville) et non comme une commande SQL.
3. **Typage :** En déclarant `string $ville` dans tes arguments, tu forces PHP à valider le type de donnée avant même que la base ne soit sollicitée.

2. Le Principe du Moindre Privilège

J'ai créé un système de "types de connexion" (0, 1, 2) pour ma stratégie de sécurité.

L'idée :

Si un pirate arrive à compromettre le compte appUser (type 1), il ne pourra pas supprimer des tables ou modifier la structure de la base, car ce compte n'a que des droits SELECT, INSERT, UPDATE sur des tables spécifiques. Aucun utilisateur de l'appli n'a les droits CREATE, DROP ou ALTER sur la base.

● Test

PHPUnit est le framework de tests unitaires de référence pour le langage PHP. Son rôle est de permettre aux développeurs de vérifier automatiquement que chaque petite unité de code (comme une fonction ou une méthode) se comporte exactement comme prévu. En créant des tests automatisés, on s'assure que l'ajout d'une nouvelle fonctionnalité ou la modification d'une portion de code n'entraîne pas de régressions (bugs) sur le reste de l'application. Dans le cadre de ce projet, l'utilisation de PHPUnit garantit la fiabilité des calculs et de la logique métier, tout en facilitant la maintenance à long terme grâce à un code testable et robuste.

exemple de test effectuer sur l'insertion de donnée:

```
<?php
namespace App\Tests;

use PHPUnit\Framework\TestCase;
use App\Db\DbAddService;
use PDO;

class DbAddServiceTest extends TestCase {
```

TOUCHE PAS AU KLAXON



```
private DbAddService $service;
private PDO $dbVerify;

protected function setUp(): void {
    $this->service = new DbAddService();
    // Connexion directe pour vérifier les résultats en base
    $this->dbVerify = new
PDO("mysql:host=localhost;dbname=test_covoitnage_interne", "a_app",
"a_app_mdp");
}

public function testAddVilleSuccess(): void {
    $nomVille = "Lyon-Sud-" . uniqid(); // Nom unique pour éviter
les conflits d'index
    $this->service->addVille($nomVille);

    $stmt = $this->dbVerify->prepare("SELECT * FROM ville WHERE nom
= ?");
    $stmt->execute([$nomVille]);
    $this->assertNotEmpty($stmt->fetch(), "La ville devrait être
présente en base.");
}

public function testAddTrajetSuccess(): void {
    $data = [
        'depart_ville' => 1,
        'depart_gdh' => '211000Z JAN 26',
        'depart_date' => '2026-01-21 10:00:00',
        'arrive_ville' => 2,
        'arrive_gdh' => '211200Z JAN 26',
        'arrive_date' => '2026-01-21 12:00:00',
        'place_totale' => 4,
        'place_disponible' => 4,
        'createur_id' => 1
    ];

    $this->service->addTrajet($data);

    $stmt = $this->dbVerify->query("SELECT * FROM trajet ORDER BY
id DESC LIMIT 1");
    $trajet = $stmt->fetch();
```

TOUCHE PAS AU KLAXON



```
$this->assertEquals('211000Z JAN 26', $trajet['depart_gdh']);
$this->assertEquals(1, $trajet['depart_ville_id']);
}

public function testUserCannotAddVille(): void {
    $this->expectException(\PDOException::class);
    // On force l'utilisation du type 1 (u_app) pour une action
admin
    $pdo = $this->service->connexion(1);
    $pdo->exec("INSERT INTO ville (nom) VALUES ('Ville
Interdite')");
}

public function testSecurityUserCannotDeleteVille(): void {
    // On s'attend à ce qu'une exception PDO soit lancée
    $this->expectException(\PDOException::class);

    // On récupère une connexion de type 1 (u_app)
    $pdo = $this->service->connexion(1);

    // Cette commande doit échouer car u_app n'a que SELECT sur la
table ville
    $pdo->exec("DELETE FROM ville WHERE id = 1");
}
}
```

CONCLUSION

En conclusion, ce projet a été l'occasion de consolider mes compétences en développement full-stack et de valider mon apprentissage auprès du centre européen de formation. De la conception du MCD à la mise en œuvre de la logique back-end en PHP, chaque étape m'a permis d'appliquer des bonnes pratiques de développement et de sécurité.

Au-delà de l'aspect technique, la réflexion menée sur l'UX et les besoins des utilisateurs m'a permis d'appréhender le développement web non plus seulement comme une écriture de code, mais comme la création d'un service utile et fonctionnel.

'Touche pas au klaxon' répond efficacement aux problématiques de mobilité inter-sites de l'entreprise en offrant une solution de covoiturage simple et robuste. L'utilisation de PHP 8 couplée à une architecture sécurisée (requêtes préparées, gestion stricte des priviléges SQL) garantit l'intégrité des données des collaborateurs. Cette première version (MVP) pose des bases techniques saines.

Bien que l'application soit pleinement fonctionnelle dans son état actuel, elle a été pensée pour être évolutive. Les prochaines étapes de développement pourraient inclure l'automatisation des calculs de temps de trajet via l'API Nominatim et une interface plus dynamique grâce à JavaScript.

En optimisant les coûts de transport et en favorisant les échanges entre employés, cette solution s'inscrit durablement dans la politique de transition écologique et de gain d'efficacité de l'entreprise.