

# Rapport du Mini\_Projet

## Location\_Voitures

### □ Description du projet :

Le projet a pour objectif de faciliter la **location de voitures** de manière rapide, sécurisée et accessible à tous. Elle s'adresse aussi bien aux **particuliers** qu'aux **professionnels** qui souhaitent réserver un véhicule selon leurs besoins. L'utilisateur peut **consulter les voitures disponibles, réserver en ligne, gérer sa réservation**, et même **suivre l'état du véhicule loué**. L'application vise à rendre la location plus fluide, transparente et adaptée aux usages modernes grâce à une interface simple et intuitive.

### □ Architecture technique :

#### Stack technologique :

- Backend :

```
spring-boot-starter-parent
```

```
<version>3.5.7</version>
```

```
spring-boot-starter-data-jpa
```

- Frontend :

```
spring-boot-starter-thymeleaf
```

HTML/CSS/Bootstrap.

- Base de données :

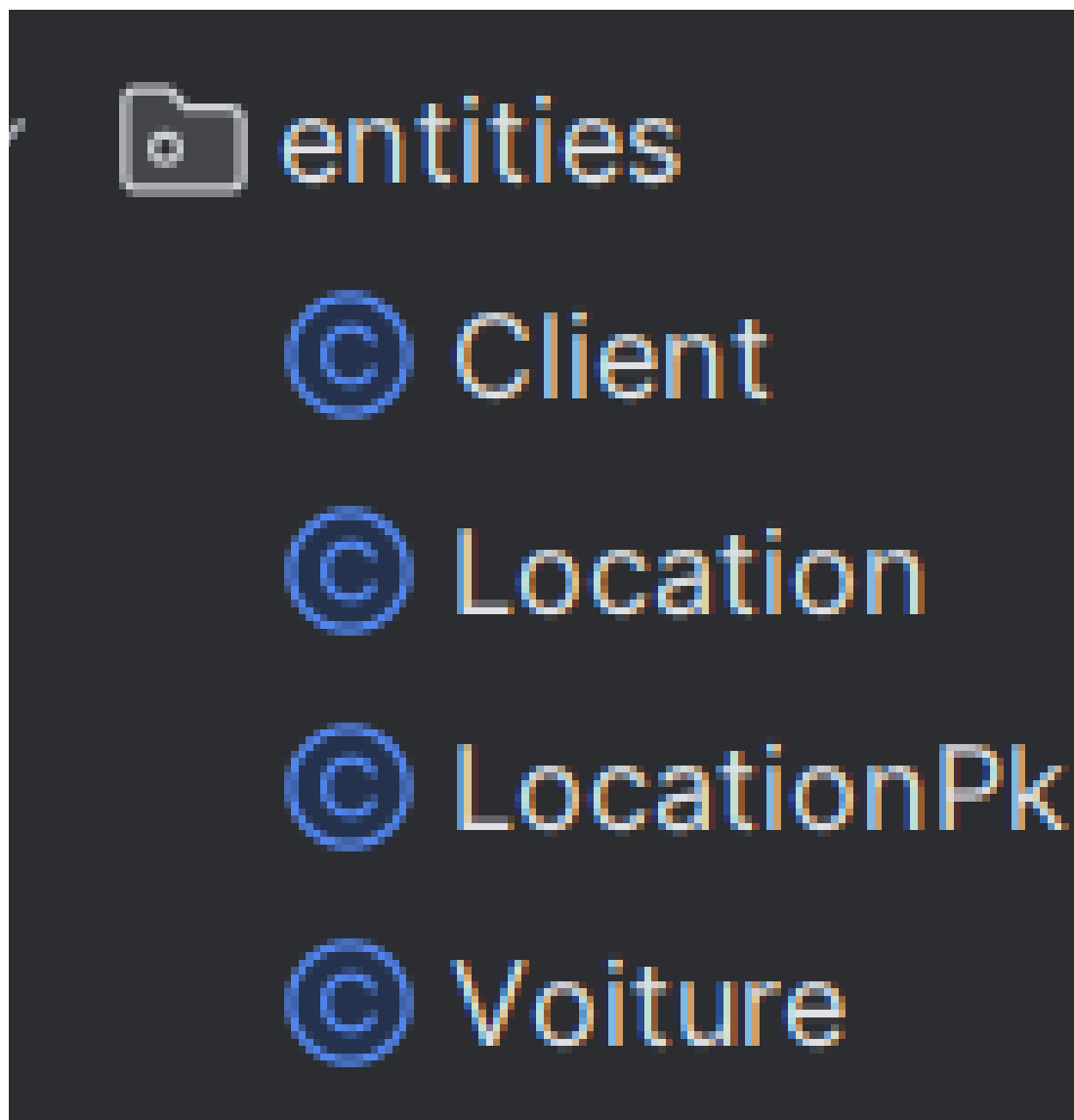
```
mysql
```

- Build :

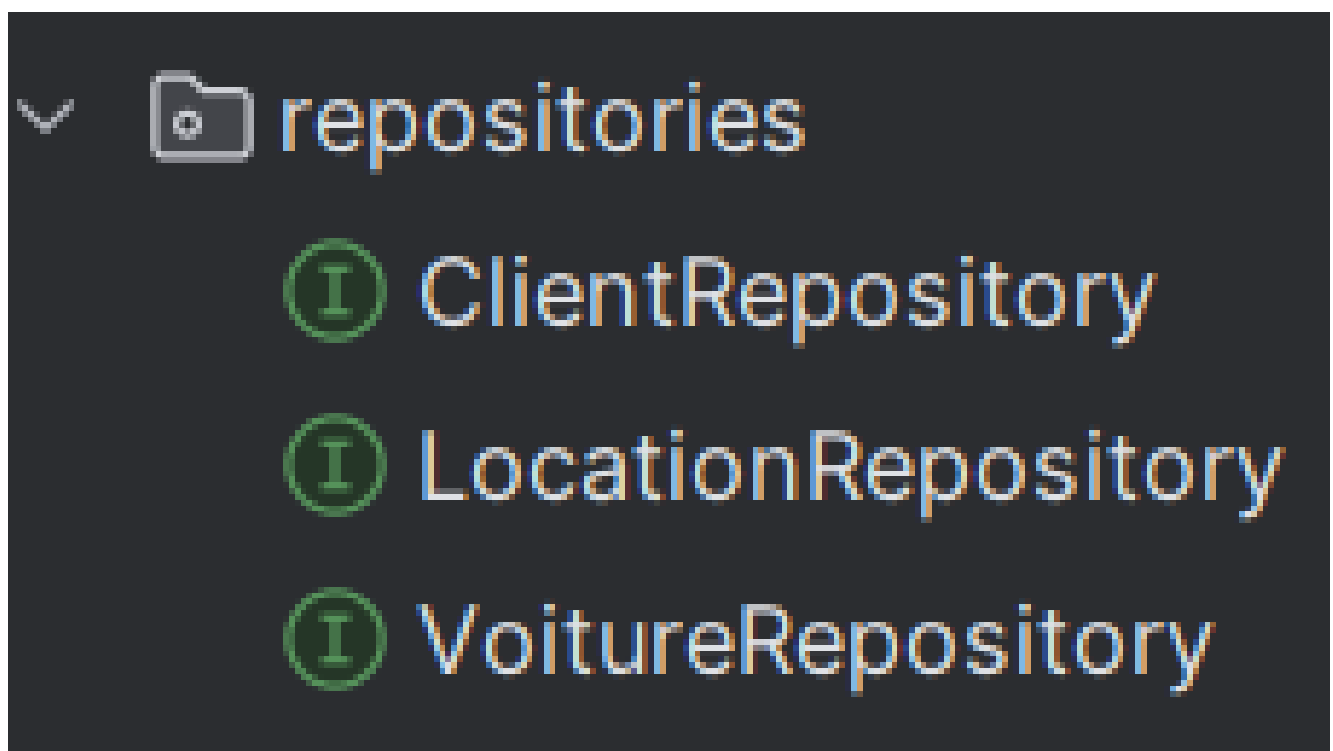
```
spring-boot-maven-plugin
```

#### Structure du code :

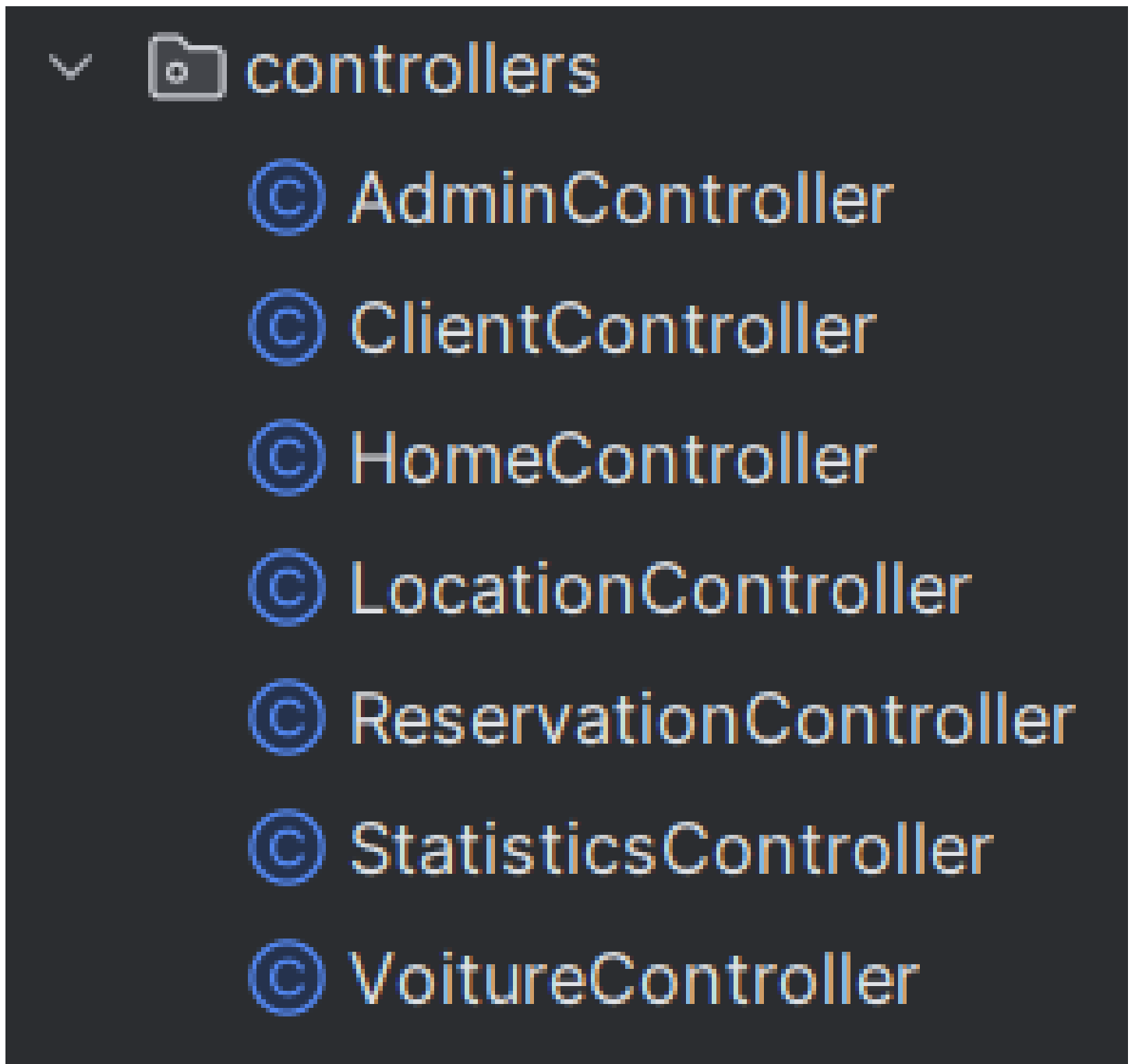
- entity :



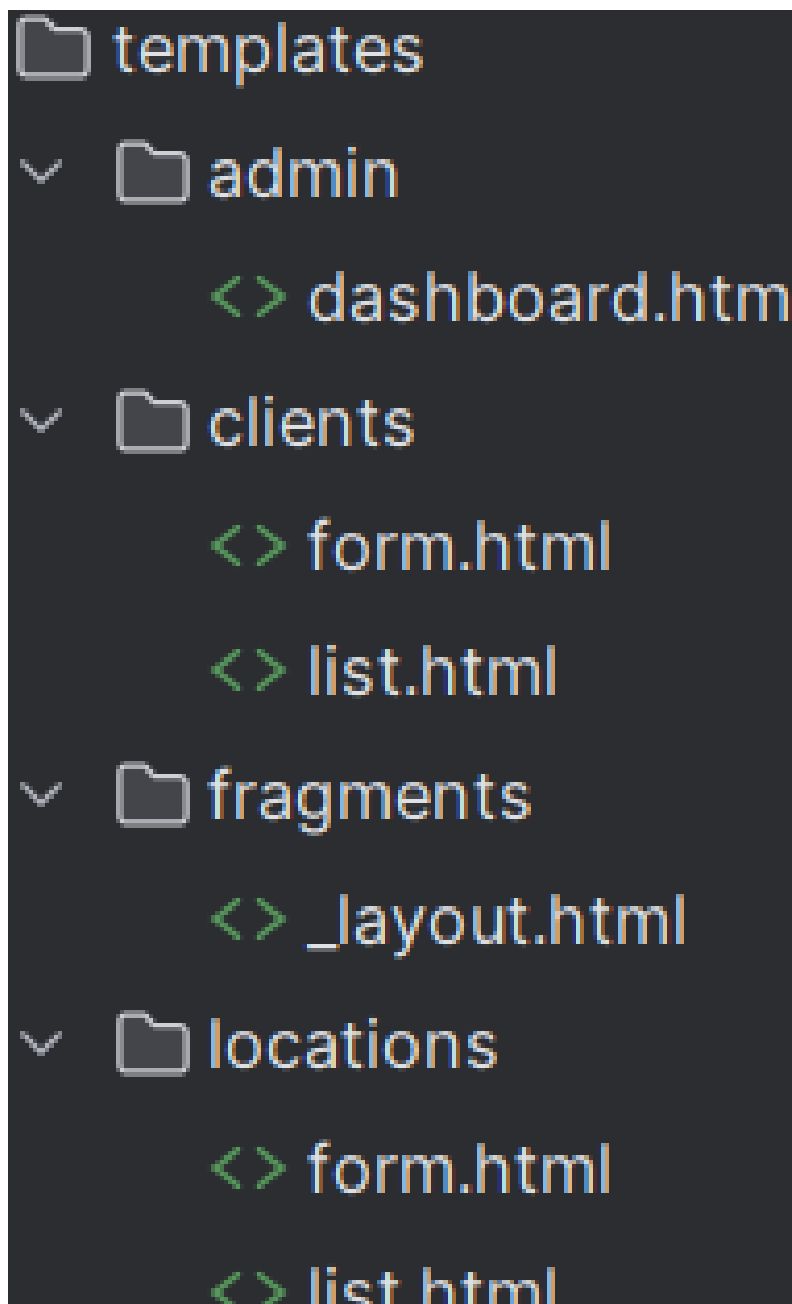
- repository :



- controller :



- templates/static :



## Diagramme d'architecture :

- Flux :

Dans mon application de location de voitures, le fonctionnement suit une architecture **MVC (Modèle – Vue – Contrôleur)** basée sur **Spring Boot**. L'utilisateur interagit d'abord avec l'**interface du navigateur**, qui envoie ses actions (comme une recherche ou une réservation) au **contrôleur Spring**. Ce contrôleur transmet ensuite la demande au **service**, qui contient la logique métier, comme la vérification de la disponibilité d'une voiture. Le service fait appel au **repository**, chargé d'accéder à la **base de données** pour récupérer ou enregistrer les informations. Enfin, les données traitées sont renvoyées vers la **vue Thymeleaf**, qui affiche le résultat à l'utilisateur (par exemple la liste des voitures disponibles ou la confirmation de réservation).

## ❑ Fonctionnalités principales :

**CRUD sur les entités principales** : l'application permet d'**ajouter, modifier, supprimer et consulter** les informations des voitures, des clients et des réservations.

**Recherche / filtrage** : l'utilisateur peut **rechercher et filtrer** les voitures par **marque, modèle, disponibilité ou catégorie** afin de trouver rapidement le véhicule souhaité.

**Tableau de bord / statistiques** : une page de **tableau de bord** affiche des **indicateurs clés** comme le nombre total de voitures, de réservations, de clients, et le taux d'occupation des véhicules.

**Gestion des statuts** : le système gère les **états des réservations** (en attente, confirmée, annulée) et des **voitures** (disponible, louée, en maintenance) pour assurer un suivi clair et précis.

## ❑ Modèle de données :

### Entités :

- **Voiture** : contient les attributs principaux comme id, marque, modèle, immatriculation, prixParJour, et statut (disponible ou louée).
- **Client** : comprend id, nom, prénom, email, téléphone, et éventuellement adresse.
- **Location** : dateDébut, dateFin, montantTotal, et le statut (confirmée, annulée, en attente).

### Relations :

- Une **voiture** peut avoir **plusieurs réservations** → relation @OneToMany.
- Une **réservation** est faite par **un seul client**, mais un client peut faire plusieurs réservations → @ManyToOne.

### Configuration base de données

- **URL de connexion** : jdbc:mysql://localhost:8080/location\_voitures
- **Identifiants de test** : username = root, password = "" .
- **Stratégie de génération** : spring.jpa.hibernate.ddl-auto = update → les tables sont créées ou mises à jour automatiquement selon les entités.

## ❑ Lancer le projet :

### Prérequis :

Avant de lancer l'application, il faut s'assurer que ton environnement est correctement configuré :

- **Java 25** est requis pour exécuter le projet Spring Boot.
- **Maven** doit être installé pour gérer les dépendances et compiler le projet.

Ces outils garantissent que le projet puisse être compilé, exécuté et testé sans erreurs.

## Installation :

### Cloner le dépôt

Télécharge ou clone le projet depuis ton dépôt Git avec la commande :

```
git clone https://github.com/ton-utilisateur/ton-projet-location-voitures.git
```

puis ouvre le projet dans ton IDE (IntelliJ, Eclipse, VS Code...).

### Configurer la base de données

Ouvre le fichier `src/main/resources/application.properties` et configure la connexion à MySQL

```
spring.application.name=Location_Voitures
# DataSource MySQL
spring.datasource.url=jdbc:mysql://localhost:3306/location_voitures?useSSL=false&serverTimezone=UTC
spring.datasource.username=root
spring.datasource.password=
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

# JPA / Hibernate
spring.jpa.hibernate.ddl-auto=create
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.format_sql=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

# Port (optionnel)
server.port=8080
```

### Lancer l'application :

#### Depuis l'IDE :

Ouvre la classe principale `LocationVoituresApplication.java`

→ clique sur **Run** .

### Accès

Une fois le projet démarré avec succès :

- **Page d'accueil :**

Page d'accueil - Location de Voitures

→ C'est la page principale de l'application où l'utilisateur peut consulter ou rechercher des voitures disponibles.

- Tableau de bord / statistiques :

Espace Admin - Location Voitures

Statistiques - Location Voitures

→ Cette page affiche des indicateurs comme le nombre de voitures, de clients, de réservations, et les taux d'occupation.



☐ Jeu de données initial (optionnel) :

Location Voitures

AccueilVoituresGalerie

Gestion des Voitures

Ajouter

Image	ID	Immatriculation	Marque	Segment	Disponible	Prix/jour	Actions
	2	S475	Dacia	Compact	Oui	200.00 DH	<div>ModifierSupprimer</div>
	3	GFYT76	Toyota	Luxe	Non	200.00 DH	<div>ModifierSupprimer</div>

Retour à l'accueil

Location Voitures © 2025 - Tous droits réservés

Location Voitures

AdminVoituresClientsLocationsStatistiques

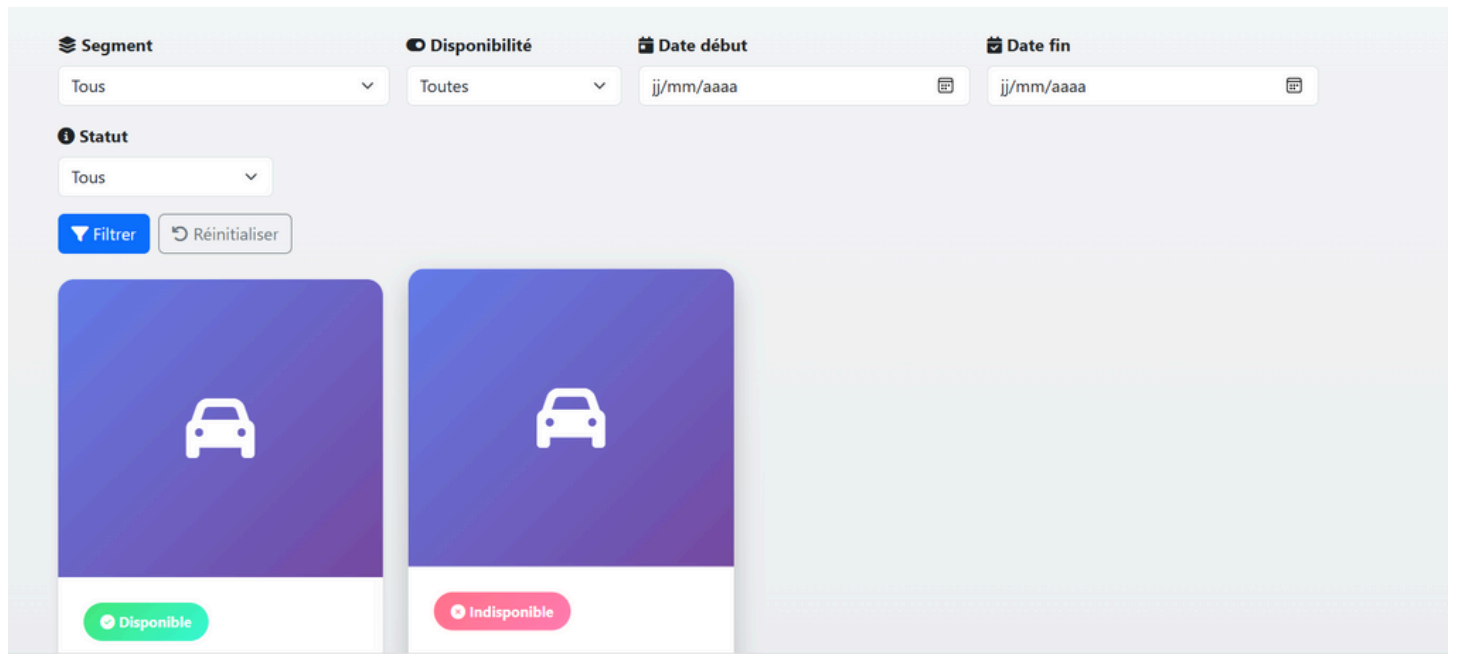
Gestion des Clients

Ajouter un Client

# ID	Nom	CIN	Téléphone	Actions
3	omayma el allali	RD	87787432	<div>ModifierSupprimer</div>

Retour au Dashboard

© 2025 Location Voitures - Tous droits réservés



## ❑ Démonstration (Vidéo) :

<https://drive.google.com/drive/folders/1AhTakR9BnM-jgUWkpZ9TMiMyOvKgd5cr?usp=sharing>

Réalisée par **AITLBIZ Kaoutar**

Encadrer par **M.LCHGAR Mohamed**

Module : Techniques de Programmatons Avancées

Ecole Normal Supérieure 2025/2026



