

[https://www.figma.com/design/jVQsSftaR15Lp49jypO6Nq/Car-Rental-Figma-Template-\(Community\)?node-id=0-1&t=pEB1cJKfUYPEh8Gn-1](https://www.figma.com/design/jVQsSftaR15Lp49jypO6Nq/Car-Rental-Figma-Template-(Community)?node-id=0-1&t=pEB1cJKfUYPEh8Gn-1)

1. Présentation Générale du Projet

Le projet consiste à développer une plateforme de location de voitures en ligne, permettant aux utilisateurs de rechercher, réserver et gérer leurs réservations de véhicules de manière simple et efficace.

1.1 Objectifs du Projet

- **Plateforme de réservation en ligne** : Offrir aux utilisateurs un moyen facile et rapide de rechercher des véhicules disponibles et de réserver en quelques clics.

1.2 Public Cible

1.2.1 Clients (Utilisateurs finaux)

- **Profil** : Particuliers recherchant une solution simple et flexible pour louer un véhicule (week-end, vacances, voyages d'affaires, etc.).
- **Besoins** : Facilité d'utilisation, transparence des prix, disponibilité en temps réel des véhicules.

2. Exigences Fonctionnelles

2.1 Fonctionnalités Principales

1. **Page de liste des véhicules** avec filtres (marque, prix, carburant...).
2. **Système de réservation** avec gestion des dates, prix et disponibilités.
3. **Création de compte et authentification** sécurisée.

4. Affichage des informations de chaque voiture avec galerie photo.

2.2 Règles Métier

2.2.1 Règles liées aux Voitures

1. Création de voiture :

- Marque, modèle, année de mise en circulation, prix par jour, image, etc.

Table `vehicle` (Véhicule)

Nom affiché (FR)	Nom en base de données (EN)	Type	Description	
Identifiant	<code>id</code>	PK, int	Identifiant unique du véhicule	
Marque	<code>brand</code>	string	Marque du véhicule	
Modèle	<code>model</code>	string	Modèle du véhicule	
Année de mise en circulation	<code>year</code>	int	Année de mise en circulation	
Prix par jour	<code>price_per_day</code>	decimal	Prix à la journée pour la location	
Image	<code>image_url</code>	string (URL)	URL de l'image du véhicule	
Nombre de portes	<code>doors</code>	int	Nombre de portes (3 ou 5)	
Type d'énergie	<code>fuel_type</code>	enum	<code>essence</code> , <code>diesel</code> , <code>électrique</code> , <code>hybride</code>	
Air conditionné	<code>air_conditioning</code>	boolean	Véhicule équipé d'air conditionné	

Nombre de places	seats	int	Nombre de places assises (2, 4, 5, 7)	
Boîte de vitesses	transmission	enum	automatique , manuelle	
Type de véhicule	vehicle_type_id	FK, int	Référence à la table vehicle_type	

Table `vehicle_type` (Type de véhicule)

Nom affiché (FR)	Nom en base de données (EN)	Type	Description
Identifiant	id	PK, int	Identifiant unique du type
Nom du type	name	string	Exemple : SUV, Crossover, Cabriolet, PickUp, Minivan, Sedan

2. Photos des véhicules

- **Un véhicule** peut avoir plusieurs photos associées. Par exemple, un SUV peut avoir 5 photos différentes.
- **Affichage** : Lorsqu'un utilisateur consulte un véhicule, il peut voir toutes les photos qui lui sont liées, organisées par l'ordre défini dans `display_order`.

Table `vehicle_photo` (Photos des véhicules)

Nom affiché (FR)	Nom en base de données (EN)	Type	Description
Identifiant	id	PK, int	Identifiant unique de la photo
Véhicule	vehicle_id	FK, int	Clé étrangère vers la table <code>vehicle</code> , référence au véhicule
URL de l'image	image_url	string	URL de l'image liée au véhicule

Ordre d'affichage	<code>display_order</code>	int	Ordre d'affichage des photos (pour organiser les images)
Date d'ajout	<code>created_at</code>	datetime	Date de création de la photo (ou date d'ajout)

2.3 Règles liées aux Réservations

2.3.1 Création d'une Réservation

1. Un utilisateur peut réserver une voiture uniquement si elle est disponible.
2. La réservation nécessite une **date de début** et une **date de fin**.
3. Les dates doivent respecter certaines contraintes (date de début avant date de fin, date de début au moins égale à la date actuelle).

Table `reservation` (Réservation)

Nom affiché (FR)	Nom en base de données (EN)	Type	Description
Identifiant	<code>id</code>	PK, int	Identifiant unique de la réservation
Utilisateur	<code>user_id</code>	FK, int	Clé étrangère vers la table <code>user</code>
Véhicule	<code>vehicle_id</code>	FK, int	Clé étrangère vers la table <code>vehicle</code>
Date de début	<code>start_date</code>	date	Date de début de la réservation
Date de fin	<code>end_date</code>	date	Date de fin de la réservation
Date de création	<code>created_at</code>	datetime	Date de création de la réservation
Prix total	<code>total_price</code>	decimal	Calcul du montant total de la réservation

2.3.2 Prix de Location

- Le prix est calculé par jour en fonction du **prix par jour** du véhicule.

2.3.3 Modification ou Annulation d'une Réservation

- Des règles sont appliquées concernant la modification ou l'annulation (annulation avant confirmation, etc.).
-

3. Contraintes Techniques

- **Technologie** : Le projet sera développé en **PHP Laravel**.
 - **Base de données** : Utilisation d'une base de données **MySQL**.
-

4. Sécurité

1. **Authentification sécurisée** : Utilisation du hachage pour le stockage des mots de passe (par exemple, bcrypt).
 2. **Accès restreint pour les administrateurs** : Les administrateurs auront un accès limité aux informations sensibles (gestion des utilisateurs, des véhicules, des réservations).
-

5. Conception et UI/UX

1. **UI/UX Design** : La conception de l'interface utilisateur doit suivre des lignes directrices précises concernant la palette de couleurs, typographie, iconographie, et l'ergonomie générale.
 - Un guide de style est disponible via ce [lien Figma](#).
-

6. Deadline

Le projet doit être fonctionnel et disponible sur GitHub avant le 18 **Avril 2025 à 18h00**.

Cahier des charges (1)

Arborescence

- Page d'accueil

- Page qui affiche tous les véhicules
- Page détail d'un véhicule
- Page de reservation d'un véhicule
- Inscription / Connexion