

Cahier des Charges - FastMoto

1. Présentation du Projet

1.1 Description Générale

FastMoto est une plateforme innovante de location de motos entre particuliers. Elle permet aux propriétaires de motos de rentabiliser leurs véhicules en les proposant à la location, et aux utilisateurs de trouver facilement une moto à louer selon leurs besoins.

1.2 Objectifs du Projet

- Créer une plateforme sécurisée et conviviale pour la location de motos
- Faciliter la mise en relation entre propriétaires et locataires
- Assurer la sécurité des transactions et des réservations
- Offrir une expérience utilisateur optimale
- Gérer efficacement le processus de location de bout en bout

1.3 Public Cible

- Propriétaires de motos souhaitant rentabiliser leur véhicule
- Passionnés de moto cherchant des locations occasionnelles
- Touristes désirant louer une moto pour leurs déplacements
- Utilisateurs réguliers ayant besoin d'une moto temporairement

2. Fonctionnalités Principales

2.1 Gestion des Utilisateurs

- Inscription et authentification sécurisée
- Profils détaillés pour propriétaires et locataires
- Système de vérification d'identité
- Gestion des rôles (utilisateur, propriétaire, administrateur)
- Historique des locations et transactions

2.2 Gestion des Motos

- Publication d'annonces détaillées
- Photos et descriptions complètes
- Calendrier de disponibilité

-
- Tarification personnalisable
 - Localisation des motos sur carte
 - Filtres de recherche avancés

2.3 Système de Réservation

- Réservation en ligne intuitive
- Vérification automatique des disponibilités
- Gestion des demandes spéciales
- Confirmation automatique ou manuelle
- Annulation et modification de réservation
- Système de caution

2.4 Paiements et Transactions

- Paiement sécurisé en ligne
- Intégration Stripe
- Gestion des cautions
- Facturation automatique
- Historique des transactions
- Remboursements automatisés

2.5 Communication

- Messagerie intégrée entre utilisateurs
- Notifications en temps réel
- Système d'évaluation et avis
- Alertes automatiques
- Support client intégré

2.6 Administration

- Interface d'administration complète
- Gestion des utilisateurs et des annonces
- Modération des contenus
- Statistiques et rapports
- Gestion des litiges

3. Aspects Techniques

3.1 Architecture Technique

- Backend Node.js avec Express
- Frontend React avec Material-UI

-
- Base de données MySQL
 - API RESTful sécurisée
 - WebSocket pour le temps réel

3.2 Sécurité

- Authentification JWT
- Validation OTP
- Chiffrement des données sensibles
- Protection contre les attaques courantes
- Gestion des sessions sécurisée

3.3 Performance

- Optimisation des requêtes
- Mise en cache
- Chargement différé des images
- Optimisation mobile
- Temps de réponse rapide

3.4 Intégrations

- Passerelle de paiement Stripe
- Authentification Google
- Services de cartographie
- Envoi d'emails automatisés
- Notifications push

4. Interface Utilisateur

4.1 Design

- Interface moderne et responsive
- Navigation intuitive
- Adaptabilité mobile
- Thème personnalisable
- Accessibilité optimisée

4.2 Expérience Utilisateur

- Parcours utilisateur optimisé
- Formulaires intelligents
- Feedback instantané
- Aide contextuelle

5. Déploiement et Maintenance

5.1 Hébergement

- Configuration serveur
- Gestion des environnements
- Sauvegarde automatique
- Monitoring continu
- Plan de reprise

5.2 Maintenance

- Mises à jour régulières
- Correction des bugs
- Améliorations continues
- Support technique
- Documentation maintenue

6. Évolutions Futures

6.1 Fonctionnalités Planifiées

- Application mobile native
- Système de fidélité
- Intégration de nouveaux moyens de paiement
- Expansion géographique
- Fonctionnalités communautaires

6.2 Améliorations Techniques

- Migration vers TypeScript
- Optimisation des performances
- Amélioration de la scalabilité
- Renforcement de la sécurité

7. Support et Formation

7.1 Documentation

- Guide utilisateur
- Documentation technique
- FAQ détaillée

-
- Tutoriels vidéo
 - Base de connaissances

7.2 Formation

- Formation des administrateurs
- Support aux utilisateurs
- Guides de dépannage
- Ressources d'aide

8. Spécifications de la Base de Données

8.1 Objectif

Cette section définit la structure de la base de données FastMoto qui supporte toutes les fonctionnalités spécifiées dans ce document (réservation, paiement, chat, notifications...).

8.2 Schéma Global

La base de données nommée `motor_db` est constituée des tables suivantes :

- users
- motors
- motor_locations
- bookings
- payments
- notifications
- chat_rooms, chat_messages
- reviews
- insurance_records, maintenance_records
- orders
- page_views, visitor_countries
- user_sessions
- countries

8.3 Détails des Tables

8.3.1 Table users

Colonne	Type	Description
id	INT	Identifiant utilisateur
email	VARCHAR	Email (unique)
password_hash	TEXT	Mot de passe haché
role	ENUM(user, seller,	

	admin)	Rôle de l'utilisateur
is_blocked	BOOLEAN	Statut de blocage
google_id	VARCHAR	Pour login via Google
otp_code	VARCHAR	Code OTP temporaire
created_at	DATETIME	Date d'inscription

Supporte l'authentification par mot de passe, OTP, OAuth.

8.3.2 Table motors

Colonne	Type	Description
id	INT	Identifiant moto
user_id	INT (FK)	Propriétaire
title	VARCHAR	Titre de l'annonce
description	TEXT	Détails complets
price_per_day	DECIMAL	Prix journalier
available	BOOLEAN	Disponibilité
created_at	DATETIME	Date d'ajout

Liée aux tables motor_locations, bookings, insurance_records, maintenance_records.

8.3.3 Table motor_locations

Colonne	Type	Description
id	INT	Identifiant
motor_id	INT (FK)	Moto concernée
latitude	DOUBLE	Latitude GPS
longitude	DOUBLE	Longitude GPS
address	TEXT	Adresse complète

Sert à afficher les motos sur la carte.

8.3.4 Table bookings

Colonne	Type	Description
id	INT	Réservation
user_id	INT	Client
motor_id	INT	Moto louée
status	ENUM	en_attente, confirmée, en_cours, terminée, annulée
start_date	DATE	Début location
end_date	DATE	Fin location
total_price	DECIMAL	Prix total
payment_status	ENUM	payé, en_attente

Suivi du cycle de vie des réservations.

8.3.5 Table payments

Colonne	Type	Description
---------	------	-------------

id	INT	Paielement
booking_id	INT (FK)	Réservation associée
method	ENUM	stripe, virement, à_la_livraison
amount	DECIMAL	Montant
status	ENUM	confirmé, en_attente, remboursé
stripe_txn_id	VARCHAR	ID Stripe

Gestion intégrée Stripe et manuelle.

8.3.6 Table reviews

Colonne	Type	Description
id	INT	Avis
user_id	INT	Utilisateur auteur
motor_id	INT	Moto concernée
rating	INT	Note sur 5
comment	TEXT	Commentaire
reply	TEXT	Réponse du vendeur

Permet aux clients de noter et commenter les motos.

8.3.7 Table notifications

Colonne	Type	Description
id	INT	Notification
user_id	INT	Destinataire
content	TEXT	Message
type	ENUM	réservation, paiement, message, info
is_read	BOOLEAN	Lu ou non
created_at	DATETIME	Date envoi

Notification en temps réel via WebSocket.

8.3.8 Tables chat_rooms & chat_messages

Permet le chat entre utilisateurs/vendeurs/admins.

****chat_rooms****

- id
- user1_id
- user2_id
- last_message_id

****chat_messages****

- id
- room_id
- sender_id
- content
- is_read

-
- timestamp

Supporte état "vu", indicateurs de frappe.

8.3.9 Table insurance_records

Colonne	Type	Description
id	INT	ID
motor_id	INT (FK)	Moto concernée
provider	VARCHAR	Assureur
start_date	DATE	Début couverture
end_date	DATE	Fin couverture

Suivi administratif des assurances.

8.3.10 Table maintenance_records

Colonne	Type	Description
id	INT	ID
motor_id	INT (FK)	Moto concernée
description	TEXT	Type de maintenance
date	DATE	Date réalisation

Historique des entretiens.

8.4 Relations et Contraintes

- users -> motors, bookings, reviews
- motors -> motor_locations, insurance_records, maintenance_records
- bookings -> payments, notifications
- chat_rooms -> chat_messages
- Foreign keys avec ON DELETE CASCADE
- Indexation sur user_id, motor_id, booking_id

8.5 Validation & Sécurité

- Contraintes sur les ENUM pour les statuts
- Validation sur formats email, dates, et montants
- Hachage des mots de passe (backend)
- Historique d'activités et audit log (via user_sessions, page_views)