CleMoPi Mobility Platform

Developer Technical Specification and System Architecture

Prepared by:
Oualid Boukhris

Company / Project:
Clemopi Mobility Platform

Frontend Applications :

Web App (Admin Dashboard)

Mobile App (User Application)

Backend Technology:
Node.js / Express / Firebase

Version :
1.0.0

Date :
November 2025

Web/Mobile App

Description:

Ensure the management & real-time monitoring of electric scooter usage. The mobile application will allow end users to perform the following operations:

- Geolocate and view charging stations on a map, along with their availability.
- Reserve a station while on the way to the selected location.
- Scan the QR code on the unit to start charging a scooter.
- Pay for the charging service and validate payments.
- View their usage history
- Remote control of electric scooters
- Ensures secure mobile payment transactions.

E-Scooter Rental System — Technical Documentation

1. Overview

The **E-Scooter Rental System** enables users to locate, unlock, and rent electric scooters via a mobile application, while the web dashboard allows administrators to manage customers, scooters, and analytics data.

The system integrates **Node.js/Express** for backend services and **Firebase** for authentication, real-time updates, and cloud storage.

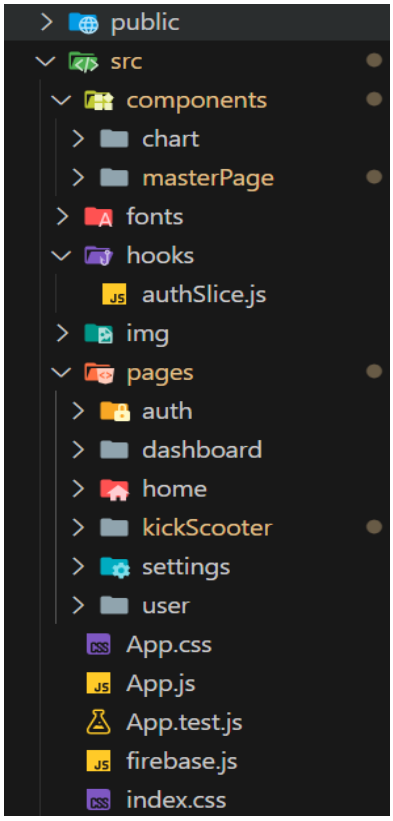
2. System Architecture:

Components

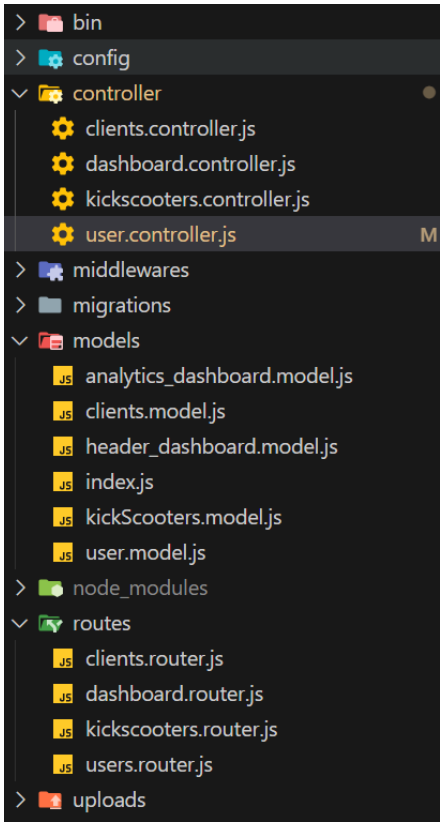
Layer	Technology	Description
Frontend (Web)	React.js	Admin dashboard for managing scooters, clients, and analytics
Frontend (Mobile)	Flutter	End-user app to scan scooters, view balance, start rides
Backend	Node.js + Express	REST API for all CRUD operations and communication between apps and database
Database	Firebase Firestore + MySQL	Stores users, clients, scooters, and ride transactions
Authentication	Firebase Auth,JWT	Handles login, registration, and identity verification
Cloud Storage	Firebase Storage	Stores scooter images, profile photos, etc.

3. Folder Structure (Conceptuel)

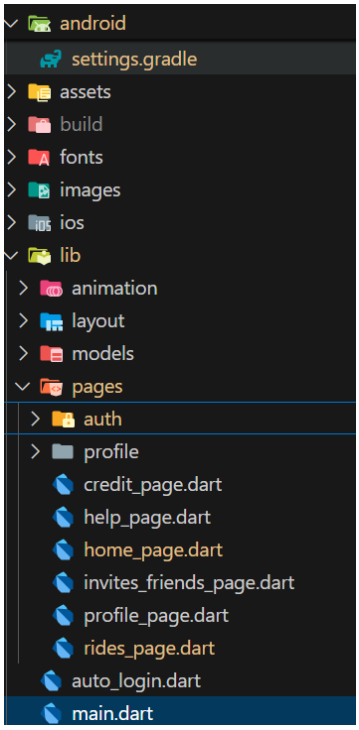
Web App (Front End)



Web App (Back End)



Mobile App



4. Main Features

Web Dashboard (Admin)

- View, add, edit, and delete clients
- Manage scooter fleet: QR codes, battery status, lock status, GPS coordinates
- View analytics dashboard (rides, payments, meters traveled)
- Monitor total orders, total meters, and real-time status
- Secure admin login via Firebase Auth and JWT

Mobile App (Client)

- Sign up / login via Firebase (email, phone, or Google)
- Scan scooter QR code to start riding
- View battery level, ride time, and total distance
- Wallet system: recharge balance via payment gateway
- View history and payment receipts
- Live map (Google Maps API) showing nearby scooters

Backend (Node.js/Express)

- REST endpoints for CRUD operations on clients, kick scooters, orders, analytics
- Middleware for authentication and data validation
- Firebase SDK integration for real-time sync
- Scheduled tasks (e.g. scooter status updates, data sync)

5. Data Flow

✓ User Authentication

- Mobile app → Firebase Auth → returns user token
- Token verified by Express middleware for protected routes

✓ Scooter Unlocking

- Mobile app scans QR → sends scooter ID to backend
- Backend validates scooter status → sends unlock command to scooter firmware (via IoT gateway)

✓ Ride Tracking

- Scooter telemetry (speed, battery, GPS) → backend → updates Firebase
- Web dashboard visualizes real-time data

✓ Payments

- Mobile app deducts ride cost from balance → updates backend → logs analytics

6. Security

- JWT tokens for backend route protection
- Firebase Auth session tokens for user identity
- HTTPS enforced for API requests
- Data validation middleware on every endpoint

7. Future Improvements

- Integrate **real IoT communication** with scooters
- Support **subscription or package plans**
- Integrate **PayPal or Naps payments**
- Enable **push notifications** (Firebase Cloud Messaging)
- Create **reporting & export** features in admin dashboard