DV200 Open Brief Proposal

Rouxan Potgieter 231013
Open Window, Creative Technologies
DV 200
Tsungai Katsuro
12 September 2024

Project Proposal:
Car-Sharing Platform Development Using MERN Stack

Problem Statement

In South Africa, many licensed drivers face the challenge of being unable to afford personal vehicles. The high cost of purchasing and maintaining a car, coupled with unreliable public transportation options in certain regions, hinders mobility, work opportunities, and personal convenience. This issue disproportionately affects individuals in urban and suburban areas, especially those who commute for work or daily tasks.

The proposed solution is a car-sharing platform that allows people to rent cars on-demand, either for short or long trips, addressing mobility issues by offering a cost-effective, flexible alternative to car ownership. This platform will not only help individuals who can't afford cars but also promote a more sustainable transportation model by reducing the number of vehicles on the road.

Target Audience
The primary users of the application include:

Drivers with Valid Licenses: Individuals who have valid driving licenses but lack the financial means to purchase or maintain a car.
Car Owners: People who own cars but want to earn money by renting out their vehicles when they are not in use.
Urban and Suburban Commuters: People living in urban and suburban areas where public transportation is unreliable or unavailable, and owning a car is either impractical or too expensive.

Benefits:
Drivers: Access to affordable car rentals for daily needs or special trips without the long-term financial commitment of car ownership.
Car Owners: Ability to generate passive income by renting out their vehicles when idle.
General Population: A more sustainable transport solution that reduces the number of personal vehicles needed on the road, lowering congestion and environmental impact.

Technology Stack
I will use the MERN stack (MongoDB, Express, React, Node.js) for this project.

Justification:
Real-Time Features: The application will require real-time updates for car availability and live location tracking, which the MERN stack is well-suited for.
Scalability and Flexibility: The MERN stack offers scalability, making it ideal for a platform that could grow in terms of users, cars, and transactions.
Full-Stack JavaScript: MERN allows for using JavaScript on both the front-end and back-end, which improves development efficiency and enables seamless communication between components.

Overview of Each Component:
MongoDB: A NoSQL database will store user data, car details, and rental information. Its geospatial capabilities will help track and query nearby cars.
Express: This Node.js framework will handle API routing, user authentication, and business logic for CRUD operations.
React: The front-end will provide an interactive user interface (UI) for booking cars, displaying nearby options, and managing user accounts.
Node.js: The back-end will manage the server-side logic, including handling requests, performing CRUD operations, and managing real-time features.

Application Features
Core Features:
User Authentication: CRUD: Create accounts, read profiles, update profile info, and delete accounts.

Car Listings: CRUD: Car owners can list cars (create), view their cars (read), edit details (update), or remove cars from the listing (delete).

Car Booking: CRUD: Drivers can search and book cars (create), view their bookings (read), modify rental details (update), and cancel bookings (delete).

Live Car Tracking: Real-time tracking of cars and displaying the nearest available cars on a map.

Database Design
Key Collections:
Users Collection:
userID, name, email, password, licenseNumber, userType (Driver or Owner)
Cars Collection:
carID, ownerID, make, model, location (GeoJSON), availability
Bookings Collection:
bookingID, carID, driverID, startTime, endTime, status

Data Considerations:
Geospatial Data: Storing car locations as GeoJSON for efficient querying of nearby cars using MongoDB's geospatial indexing.

Normalisation: The database will be normalized to reduce redundancy and maintain data integrity, with clear relationships between users, cars, and bookings.

User Interface and Experience

The UI will be intuitive and user-friendly, catering to users with varying technical skills. The design will focus on:

Simple Car Booking Process: A streamlined process from car search to booking confirmation with minimal steps.

Interactive Map: A map view showing nearby cars and real-time availability for users based on their manually entered location.

Responsive Design: The app will be fully responsive, ensuring usability on both mobile and desktop devices.

Security Considerations

Potential Security Risks:

User Data Breach: Personal details such as driver's licenses and payment information are at risk.

Unauthorized Access: Malicious users attempting to access or manipulate car listings or bookings.

Mitigation Strategies:

JWT Authentication: Secure token-based authentication to manage user sessions.
Input Validation: Strict input validation to prevent SQL injection, cross-site scripting (XSS), and other vulnerabilities.

Encryption: Encrypt sensitive data such as passwords and payment information.
Authorization Levels: Role-based access control, ensuring users can only perform actions relevant to their role (e.g., drivers cannot modify car listings).

Project Timeline

Planning & Research	1 Week	Finalize problem statement, stack selection, and project structure.
Database Design	1 Week	Design and create the database schema, collections, and relationships.
Backend Development	2 Weeks	Set up Node.js server, create API endpoints, implement CRUD operations.
Frontend Development	2 Weeks	Develop React UI, integrate Google Maps API, implement user and car interfaces.
Testing & Debugging	1 Week	Test for bugs, optimize UI, backend, and security features.
Deployment	1 Week	Deploy the application on a cloud platform (e.g., Heroku, AWS).

Challenges and Risks

Potential Challenges:

Real-Time Features: Implementing real-time car availability updates could be complex and time-consuming.

Risk Mitigation:
I will start with a **minimal viable product (MVP)** to ensure core functionality before adding advanced features.

Conclusion
This car-sharing platform addresses the problem of limited access to affordable transportation in South Africa by providing a solution for people with valid licenses but without cars. The use of the MERN stack ensures a scalable, real-time solution that can grow with user demand. By addressing mobility issues, the platform will contribute to greater access to jobs, services, and social opportunities, making a significant impact on the lives of its users.