

Car/Bike Rental Platform (AuoMates)

Project Overview:

The Car/Bike Rental Platform(Automates) is a scalable, enterprise-grade application that provides both a React web SPA and a React Native mobile app for seamless bike/car rental management. It supports two primary roles: **Leasers** (users who rent vehicles) and **Owners** (users who list vehicles), along with an **admin portal** built using Spring Boot for centralized control and reporting. The project follows a clear **MVC architecture**, cleanly separating the frontend clients from the RESTful backend API, all designed to be delivered within a 2-week development sprint.

Functional Requirements:

For Leasers (Web & Mobile):

- **User Login & Signup**
- **Dashboard:**
 - View popular and nearby vehicles
 - See ongoing/past rentals
 - View announcements
- **Vehicle Search & Booking:**
 - Search by location, date, type (bike/car)
 - View vehicle details and book with date selection
- **Booking Management:**
 - View current/past bookings
 - Cancel upcoming bookings
- **Profile Management:**
 - Update personal info like name, contact, and ID verification

For Owners (Web & Mobile):

- **Owner Login & Signup**
- **Dashboard:**
 - View stats (bookings, income)
 - Track vehicle availability
- **Vehicle Management:**
 - Add, update, and delete vehicles
 - Set availability, location, and rental rates

- **Booking Oversight:**
 - View who booked what and when
 - Option to approve/reject bookings (if manual mode)
- **Profile & Documents:**
 - Update personal and business info

For Admin (Web Only):

- **Admin Login:**
 - Protected login
- **User & Owner Management:**
 - View and block/unblock users/owners
- **Vehicle Oversight:**
 - View all vehicles, report abuse
- **Announcements:**
 - Create and broadcast important updates or offers
- **Analytics:**
 - View reports: total bookings, peak locations, revenue charts

Technical Stack:

Front End:

- **Web SPA:** React + Vite + React Router
- **Mobile App:** React Native + Expo + React Navigation
- **Styling:** Tailwind CSS / Bootstrap
- **Auth:** JWT, Local Storage
- **API Communication:** Axios / Fetch

Back End (MVC + REST)

- **Model:**
 - **Entities:** User, Vehicle, Booking, Announcement
 - **DAO classes for MySQL access:** UserDAO, VehicleDAO, BookingDAO, AnnouncementDAO
- **View:**
 - **React components for web SPA**
 - **React Native screens for mobile app**
 - **JSP pages for admin portal**

- **Controller:**
 - **Servlets: AuthServicelet, LeaserServlet, OwnerServlet, AdminServlet, VehicleServlet, BookingServlet, AnnouncementServlet**

Database:

- MySQL (schema auto-created by Hibernate or via schema.sql)
- Relational schema covering Users, Vehicles, Bookings, etc.

Project Workflow:

AutoMates Project Workflow

1. Leaser Journey:

- **Login/Dashboard:**
 - Leaser logs in via React/React Native, stores token locally, then lands on the Dashboard.
 - Dashboard to show current and past bookings.
- **Search Vehicles:**
 - Leaser can search available vehicles by location, type, and availability on the Dashboard.
 - Vehicle data is fetched
- **Book Vehicle:**
 - Leaser selects a vehicle and clicks to "Book Now."
 - Navigates to the booking page, where they can review vehicle details and complete the booking form.
 - Booking form validation is done on the client-side.
- **Track Booking Status:**
 - Leaser can view a list of all current bookings and their statuses (pending, confirmed, completed).
 - Clicking on any booking will display the full status and details.
- **Profile Management:**
 - Leaser navigates to "My Profile," where they can view and edit personal information.
 - Profile information is fetched using GET /api/users/{id}, and updates are submitted with PUT /api/users/{id}.

2. Owner Journey:

- **Login/Dashboard:**
 - Owner logs in via React/React Native form (POST /api/auth/login), stores session, and lands on the Owner Dashboard.

- Dashboard calls GET /api/vehicles?ownerId={id} to display the owner's vehicles and GET /api/bookings?ownerId={id} to show current bookings.
- **Manage Vehicles:**
 - Owner can add, update, or delete vehicles from their fleet.
 - To add a new vehicle, the owner submits the form via POST /api/vehicles.
 - To update or delete, the owner can use PUT /api/vehicles/{id} or DELETE /api/vehicles/{id}.
- **Manage Bookings:**
 - Owner can view, accept, or reject booking requests.
 - Booking actions are performed using PUT /api/bookings/{id}/status.
- **Profile Management:**
 - Owner can view and edit personal and business information on "My Profile" page.
 - Profile information is fetched.

3. Admin Journey:

- **Admin Login & Dashboard:**
 - Admin logs in via a web form (POST /api/auth/login), stores session, and lands on the Admin Dashboard.
 - Dashboard displays quick overview statistics (e.g., total users, active bookings)
- **Manage Users:**
 - Admin can manage both Leaser and Owner accounts.
 - Admin can update or delete users.
- **Manage Vehicles:**
 - Admin can view all vehicles listed on the platform, with the ability to approve or suspend vehicles.
- **Manage Announcements:**
 - Admin can create, update, or delete announcements.
- **Reporting:**
 - Admin can view simple statistical reports such as:
 - Total bookings, active bookings, and completed bookings
 - Average booking duration and booking frequency

Project Deliverables:

1. Source Code:

- **Web Client (React):**
 - /automates-web/: Contains React components, service modules, routing, and styling.
- **Mobile Client (React Native):**
 - /automates-mobile/: Contains React Native screens, navigation setup, and API integration.
- **Back End (Servlets):**
 - /automates-backend/: Contains Java Servlets for all controllers
 - Includes MySQL configuration, DAO classes, and servlets to handle all API requests.

2. Documentation:

- **README.md** with setup instructions (npm start / expo start for mobile, mvn servlet:run for backend), MySQL schema details, and a list of API endpoints.
- **UML diagrams** illustrating MVC flow and synchronization between web/mobile clients and backend.

3. Presentation/Demonstration:

- **Live Demo:**
 - **Leaser:** Log in, view vehicles, search and book vehicles, view and track bookings, update profile (both web and mobile).
 - **Owner:** Log in, add/update/delete vehicles, accept/reject bookings, manage profile.
 - **Admin:** Log in, manage users, vehicles, and announcements, view reports.
- **Project Overview:** Summarize architecture, technology stack, key features, and the development timeline (2 weeks).