**Car Rental Portal**

**Date: -18-09-2025**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1. Introduction**

**1.1 Purpose**

This document details the essential requirements for the Car Rental Portal, which will provide a platform for users to search, book, and manage Car Rental. It will include features for real-time fare estimation, and secure payment processing.

**The system will provide three main dashboards:**

•**Super Admin Dashboard** → Allows the Super Admin to see all the available cars and the owner details.

•**Business Owner Dashboard** → Enables business owners to manage their cars and car attributes, manage driver availability, reservation management, location management, blog management, role base access control and user support. Designed to enhance convenience, the platform will also streamline communication between owner and user, ensuring a smooth and efficient car rental experience.

•**User Dashboard** → show number of book rides, view booking history, and access support , account settings and Wishlist .

**1.2 Scope**

The Car Rental Portal will provide users with a platform to search, book, and manage car rides based on their location and preferences. The system will offer multiple rental types, self-pickup and delivery of car. Business Owner will have access to a secure backend to manage drivers, car, reservation, and user feedback. The portal will be mobile-friendly and accessible via all major browsers, ensuring ease of use across various devices. Booking history, and customer support features will also be included to streamline the ride experience.

**1.3 Definitions, Acronyms, and Abbreviations**

● **UI** – User Interface

● **API** – Application Programming Interface

● **SEO** – Search Engine Optimization

● **Super Admin** → The highest-level administrator who manages business owners, Car Approval and packages.

● **Business Owner**– A registered entity that manages cars, drivers, and reservations through the owner dashboard.

● **User (Customer)** → A customer using the portal to search, book, and manage car rentals.

**1.5 Overview**

This SRS document outlines the requirements for developing a robust, efficient, and scalable Car Rental Portal. The platform is designed to serve three key user groups: customer seeking car for rent, business owner providing car for rent, and super admin to manage the overall system. The system will focus on seamless car booking and effective communication between users and business owner. It aims to provide an intuitive experience while ensuring smooth platform management and reliable service delivery.

**2.** **Overall Description**

**2.1** **Product Perspective**

The Car Rental Portal is a web-based and mobile-friendly platform that acts as a bridge between customers seeking cars for rent and business owners providing cars. The platform Built using modern technologies like HTML5, CSS3, Bootstrap, React.js, and a Node.js the platform will focus on performance, scalability, and security. The design will be fully responsive and optimized for various screen sizes, ensuring smooth user experiences across smartphones, tablets, and desktops.

**2.2 Product Features**

●**Car Search & Booking:** Users can search cars by location, pickup/delivery option, date, and time.

●**Fare Estimation:** Real-time fare calculation based on booking type and time.

●**Business Owner Tools:** Manage cars, drivers, blogs, role-based access, and reservations.

●**Super Admin Tools:** Oversee system-wide operations, approve owners/cars, manage disputes, and monitor performance.

●**Booking Types:** Daily, Weekly, Monthly, yearly.

●**Content Management System (CMS):** Enables super admin to update, FAQs, testimonial, without technical expertise.

●**Responsive Design:** Optimized for performance across mobile, tablet, and desktop devices.

●**Help Center and FAQs**: Provides categorized, searchable FAQs and support contact options for users seeking assistance.

**2.3 User Classes and Characteristics**

●**Users (Customer):** Can search, book, cancel, and track history

●**Business owner:** Receive and manage bookings, car pricing, update availability, manage driver and customers .

●**Super admin:** Manage users, and platform content

●**Guest Users:** Can search rides and fare but must sign up to book a car.\

**2.4 Operating Environment**

●**Supported Browsers:** Chrome, Firefox, Safari, Edge

●**Mobile:** Android and iOS (responsive design or hybrid app)

●**Backend:** Node.js with MongoDB

●**Hosting:** Secure, scalable

**2.5 Design and Implementation Constraints**

**●Accessibility Compliance:** The website must meet WCAG 2.1 standards to ensure it is accessible to all users, including those with disabilities.

**●Data Security:** The platform must implement strong security measures including SSL encryption, secure login/authentication, and protection against vulnerabilities like SQL injection and cross-site scripting (XSS).

**●Reliable Hosting Environment:** The website will be hosted on reliable, high-availability servers with enhanced security measures to ensure continuous accessibility and protection of sensitive user data.

**●Cross-Browser Compatibility:** The platform must function consistently across all major desktop and mobile browsers including Google Chrome, Firefox, Safari, and Edge.

**●Content Management System (CMS):** A user-friendly CMS must be integrated to allow super admin and business owner to easily manage and update content, services, without needing technical skills.

**●Scalability:** The architecture should support future growth in users, rides, and content, maintaining performance as demand increases.

**●Legal Compliance:** The platform must include all required legal pages such as Terms of Service, Privacy Policy, and Cookie Policy, which should be easily accessible and regularly updated to comply with regulations

**2.6 Assumptions and Dependencies**

**●Internet Connectivity**: The system assumes users have access to a reliable internet connection for smooth ride booking, real-time tracking, and payment processing.

**●Third-Party Integrations:** The platform will rely on external services such as SMS/email providers for notifications.

●**Updated User Devices and Browsers:** Users are expected to access the portal using modern, up-to-date browsers and devices to ensure compatibility and security.

**●Hosting Provider Reliability:** The website’s uptime and performance depend on the hosting provider’s infrastructure, including security, backups, and server availability.

●**Regulatory Compliance:** The platform depends on adherence to local transportation laws, data privacy regulations (like GDPR), and payment processing standards.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**3. System Features**

**3.1 Feature 1: User Features**

**Search & Booking**

•Users can search cars by location, date, time, and preferences (car type, brand, fuel, transmission).

•Option for self-pickup or delivery of the car.

•Display car based on city .

**Fare Estimation**

•System provides instant fare calculation-based time, and rental type.

**Reservation Management**

•Users can book, cancel, or reschedule reservations (based on the policy).

•View booking history with status (on process, approved, completed, cancelled).

Secure Payments

•Multiple payment options: Credit/Debit cards, UPI, Net Banking, Wallets.

•Support for refunds on cancellations.

**User Profile Management**

• Users can register/login using email , otp based forgot password.

• Profile contains personal info, Wishlist, booking history, recent booking.

**Customer Support**

• chat with business owner with their quarry.

**3.2 Business Owner Features**

**Car Management**

* Add/update/delete car details (car type, car brand, car model, fuel, transmission, rental price, color, steering, sets, features, extra features).
* Upload car documents (insurance, RC)
* Car location main and other linking location
* Upload car images and videos
* Car faq
* Car Extra features
* Car availability management (in rental, available, inactive)

**Location management**

* Add / delete / edit – city, state and country according to your preference.

**Customer management**

Add customer and see all the customers whose book your car.

And book reservation for that customer with their documents and license details

**Driver Management**

* Add/manage drivers, their documents, and availability.

**Enquires**

* Show all the enquiries for car rental

**Reservation Handling**

* + Accept or reject reservations.
  + Track active reservations.
  + Create reservation

**Blog & Content Management**

* + Post blogs with category, delete blog, show all the comments of that blog.

**Account setting**

* Secure Login/register as business owner
* change password and profile setting

**Role-Based Access Control (RBAC)**

* + Assign roles (Manager, Staff, Support).
  + Limit permissions accordingly.
  + Analytics & Reports
  + Dashboard showing total cars, active reservations, earnings, and customer feedback, recently added car, total number of cars .

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**3.3 Super Admin Features**

* + User & Owner Management
  + Approve/reject business owner registrations.
  + Suspend/block fraudulent users/owners.
  + Car & Package Approval
  + Review cars submitted by owners before making them live.
  + Manage owner packages.
  + Platform Monitoring
  + Track bookings, payments, car, blogs .
  + Generate system-wide reports.
  + CMS (Content Management System)
  + Update FAQs, Testimonial.
  + See all the people who want to connect for support with the contact form

**3.4 Feature 4: Help Center and FAQs**

● **Description:** Provides users with self-help resources and easy access to support.

**Functional Requirements:**

* Categorized FAQ section.
* Step-by-step guides for booking rides, and account management.
* Contact support options for unresolved queries.

**4. External Interface Requirements**

**4.1 User Interface**

The Car Rental Portal must have a clean, intuitive, and responsive interface designed using React.js, HTML5, Bootstrap, ensuring seamless experience across desktop and mobile devices. The UI should remain consistent for all dashboards (User, Business Owner, and Super Admin) while tailoring functionalities to each role.

**Key Pages & Components:**

* **Home Page:** Car search by location, availability, date, and time.
* Business Owner login and dashboard.
* **User Dashboard:** view all bookings conformed, rejected, view car Wishlist,
* **Super Admin Dashboard**: Manage owners, approve/reject cars, monitor bookings, and disputes.
* **Help center and FAQs**
* **Testimonial**
* **About us page**
* **Contact us page**
* **User register page**
* **Business owner register page**
* **Login page**
* **Business Owner Dashboard:** Car and driver management, reservation management, blog/content management, customer management, location management, role management, account setting.

**4.2 Hardware Interfaces**

The system must be compatible with various hardware devices including:

●Desktop computers and laptops

●Smartphones and tablets running Android and iOS

**4.3 Software Interfaces**

●CRM System: For managing customer and driver interactions, support tickets, and feedback.

●Analytics Tools: Integration with Google Analytics or similar platforms for tracking user behavior and website performance.

●SMS and Email Service Providers: For sending booking confirmations, notifications, and alerts.

**4.4 Communications Interfaces**

●All communications between users and the portal will occur over secure HTTPS protocols to ensure data privacy and integrity.

●The system will send automated email notifications to users when ride confirmed.

**5. System Attributes**

**5.1 Performance Requirements**

●The website must load in under 3 seconds on a standard internet connection to ensure a fast and responsive user experience.

●The system must be capable of handling high volumes of simultaneous users, especially during peak times (e.g., weekends, holidays).

5.2 Safety Requirements

●The system must comply with GDPR (General Data Protection Regulation) for users within the EU and other applicable data protection laws, ensuring personal information is handled securely.

●All personal and financial data, such as customer profiles and payment details, must be encrypted both at rest and in transit using SSL/TLS encryption.

●The system must provide secure authentication methods, including multi-factor authentication (MFA) for sensitive actions (e.g., password changes, payment details).

5.3 Security Requirements

●User Data Protection: All customer, owner, and admin data (including booking history, payment details, and personal information) must be securely stored, with access restricted to authorized roles.

●Security Audits: The platform must undergo regular security audits (penetration testing, vulnerability scanning) to identify and fix weaknesses.

●API Security: All third-party integrations (e.g., Google Maps, payment gateways) should be secured using OAuth or similar authorization methods to prevent unauthorized access to data.

●Role-Based Access Control (RBAC): Super Admin, Business Owners, and Users must have clearly separated permissions to prevent unauthorized access.

**5.4 Software Quality Attributes**

● **Reliability:** The portal must maintain 99.9% uptime, ensuring continuous availability for customers , business owner and super admin.

● **Usability:** The interface must be simple and intuitive, supporting customers of all technical backgrounds in browsing cars, making reservations, and processing payments.

● **Scalability:** The platform should be able to scale horizontally to accommodate increasing numbers of users, ride requests, and transactions. It must handle an increasing volume of traffic and bookings during peak Session without performance degradation.

● **Maintainability:** The codebase should be modular and well-documented to support easy maintenance and updates. It should be designed to allow for quick bug fixes and the addition of new features without disrupting the system's operations.

**5.5 Availability and Redundancy**

●The system must support high availability through load balancing and redundant infrastructure to ensure 24/7 access for users across different time zones.

●Data Backup: Regular backups must be performed to ensure no loss of user data in case of system failure.

5.6 Legal and Compliance Requirements

●The system must comply with all local and international regulations regarding transportation services, including applicable safety and data privacy laws.

●Terms of Service and Privacy Policy should be readily available and easily accessible to users before they use the platform. These documents should be regularly reviewed and updated to ensure compliance with evolving regulations.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**6. Appendix**

**6.1 Glossary**

●**API (Application Programming Interface):** A set of protocols for building and interacting with software applications.

● **SEO (Search Engine Optimization):** The practice of optimizing a website to rank higher in search engine results.

**● HTTPS (Hypertext Transfer Protocol Secure):** A secure protocol for transferring encrypted data between a browser and a website.

**● CRM (Customer Relationship Management):** A system that helps manage customer interactions, support tickets, and communication history.

**● GDPR (General Data Protection Regulation):** EU regulation protecting personal data and privacy.

**● UX (User Experience):** The overall satisfaction and ease of use experienced by users while interacting with the system.

● **Scalability**: The ability of a system to handle increased load and users without affecting performance.

**● MFA (Multi-Factor Authentication):** A login method requiring two or more verification factors (e.g., password + OTP).

● **OAuth**: An open standard for secure, token-based authentication and authorization.

● **RBAC** (Role-Based Access Control): A method for assigning permissions based on user roles (e.g., Super Admin, Business Owner, User).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_