**Vehicle Rental System**

**1. Introduction**

The Vehicle Rental System is a digital platform designed to simplify the process of renting vehicles such as cars, bikes, and vans. It enables customers to search, book, and pay for rentals seamlessly while providing administrators with tools to manage vehicles, users, reservations, and payments efficiently.

**2. Objectives**

 To provide an easy-to-use platform for customers to rent vehicles online.

 To reduce manual paperwork and streamline rental operations.

 To ensure secure user authentication and safe online payments.

 To provide admins with real-time vehicle availability, booking management, and reports.

**3. Key Features**

**(a)** **Customer Features**

 User Registration & Login (with email/phone verification).

 Search & Browse Vehicles (by location, type, price, availability).

 Online Booking & Payment (secure gateway, wallet, UPI, cards).

 Booking History & Invoices.

 Cancellation & Refund Management.

 Notifications & Alerts (SMS/Email reminders).

**(b)** **Admin Features**

 Dashboard to view active bookings and revenues.

 Vehicle Management (add, update, remove vehicles).

 Customer Management (view user details, feedback).

 Booking Management (approve, cancel, reschedule)

payment & Refund Tracking.

 Reports & Analytics.

**4. System Architecture**

 Frontend: React.js / Angular (for user interface).

 Backend: FastAPI / Node.js (for business logic & APIs).

 Database: MongoDB / MySQL (for storing users, vehicles, bookings).

 Authentication: JWT-based secure login.

 Hosting & Deployment: Docker, AWS/Azure/GCP.

**5. Workflow**

 a. Customer registers or logs in.

 b. Customer searches available vehicles by date & location.

 c. Customer books a vehicle and makes payment.

 d. Admin gets booking details and confirms availability.

 e. Customer receives confirmation via app/email.

 f. After usage, the customer returns the vehicle, and billing is finalized.

 g. System updates availability for the next user.

**6. Benefits**

 For Customers: Quick booking, multiple payment options, transparency.

 For Admins: Reduced manual work, real-time insights, improved efficiency.

 For Business: Scalability, digital transformation, higher customer satisfaction.

**7. Future Enhancements**

 AI-based pricing suggestions.

 GPS tracking of rented vehicles.

 Integration with insurance services.

 Mobile App (Android/iOS) with push notifications.

**8.Pseudocode**

**8.1 User Authentication**

 FUNCTION Login(username, password):

    user = FindUserByUsername(username)

    IF user == NULL:

        RETURN "User Not Found"

    IF VerifyPassword(password, user.hashedPassword) == TRUE:

        token = GenerateJWT([user.id](http://user.id))

        RETURN "Login Successful", token

    ELSE:

        RETURN "Invalid Credentials"

**8.2 Vehicle Search & Availability**

 FUNCTION SearchVehicle(location, startDate, endDate):

    availableVehicles = []

    FOR each vehicle IN VehicleDatabase:

        IF vehicle.location == location AND

           VehicleAvailable([vehicle.id](http://vehicle.id" \o "http://vehicle.id/" \t "_blank), startDate, endDate) == TRUE:

               availableVehicles.ADD(vehicle)

    RETURN availableVehicles

**8.3 Vehicle Booking**

 FUNCTION BookVehicle(userId, vehicleId, startDate, endDate):

    IF VehicleAvailable(vehicleId, startDate, endDate) == TRUE:

        bookingId = CreateBooking(userId, vehicleId, startDate, endDate)

        paymentStatus = ProcessPayment(userId, bookingId)

        IF paymentStatus == SUCCESS:

            ConfirmBooking(bookingId)

            RETURN "Booking Successful with ID: " + bookingId

        ELSE:

            CancelBooking(bookingId)

            RETURN "Payment Failed, Booking Cancelled"

    ELSE:

        RETURN "Vehicle Not Available"

**8.4 Payment Processing**

 FUNCTION ProcessPayment(userId, bookingId):

    userWallet = GetUserWallet(userId)

    bookingCost = GetBookingCost(bookingId)

    IF userWallet.balance >= bookingCost:

        DeductAmount(userWallet, bookingCost)

        RETURN SUCCESS

    ELSE:

        RETURN FAILURE

**8.5 Booking Cancellation & Refund**

 FUNCTION CancelBooking(bookingId):

    booking = GetBooking(bookingId)

    IF booking.status == CONFIRMED:

        booking.status = CANCELLED

        refundAmount = CalculateRefund(booking)

        ProcessRefund(booking.userId, refundAmount)

        RETURN "Booking Cancelled and Refund Processed"

    ELSE:

        RETURN "Booking Cannot Be Cancelled"