Project Assignment: Vehicle Rental Service API

# 1. Project Description

The Vehicle Rental Service API is a production-ready RESTful API that allows managing the full lifecycle of renting vehicles, such as cars, bikes, and trucks.   
The system allows managing vehicle inventory, creating bookings, scheduling maintenance, generating invoices, reporting damages, and providing customer rental history.   
Each endpoint is secured using API key authentication to ensure authorized access only.

# 2. Authentication

All requests made to the API must include a valid API key passed in the request header using the field 'X-API-Key'. If the API key is missing or invalid, the system will respond with HTTP 401 Unauthorized.

Example header: **X-API-Key: abc123XYZ**

# 3. Data Models

## 3.1 Vehicle

- id (Long): Unique identifier  
- make (String): Manufacturer of the vehicle  
- model (String): Model name  
- type (String): Vehicle type (Car, Truck, Bike)  
- available (boolean): Availability status  
- rentalRatePerDay (double): Rental cost per day

## 3.2 Booking

- id (Long): Unique identifier  
- vehicleId (Long): ID of rented vehicle  
- customerId (Long): ID of customer  
- pickupDate (Date): Pickup date  
- dropoffDate (Date): Drop-off date  
- extraServices (List<String>): Extra services requested (GPS, child seat, etc.)

## 3.3 Customer

- id (Long): Unique identifier  
- name (String): Customer name  
- email (String): Email address

## 3.4 MaintenanceLog

- id (Long): Unique identifier  
- vehicleId (Long): ID of maintained vehicle  
- description (String): Maintenance description  
- maintenanceDate (Date): Maintenance date

## 3.5 DamageReport

- id (Long): Unique identifier  
- bookingId (Long): Related booking ID  
- description (String): Damage description  
- estimatedCost (double): Estimated repair cost

## 3.6 Invoice

- id (Long): Unique identifier  
- bookingId (Long): Related booking ID  
- amount (double): Total amount billed  
- invoiceDate (Date): Invoice date

# 4. API Endpoints

|  |  |  |
| --- | --- | --- |
| **Method** | **Endpoint** | **Description** |
| POST | /vehicles | Add a new vehicle with make, model, type, availability and rental rate. |
| GET | /vehicles/{vehicleId} | Retrieve vehicle details by ID. |
| PATCH | /vehicles/{vehicleId}/availability | Update vehicle availability status. |
| GET | /vehicles | List all vehicles in inventory. |
| POST | /bookings | Create a booking for a customer and vehicle. |
| GET | /bookings/{bookingId} | Retrieve booking details by ID. |
| PATCH | /bookings/{bookingId} | Modify booking dates or extra services. |
| DELETE | /bookings/{bookingId} | Cancel a booking. |
| POST | /maintenance | Log a maintenance record for a vehicle. |
| GET | /maintenance/{vehicleId} | Retrieve maintenance logs for a vehicle. |
| POST | /customers | Register a new customer. |
| GET | /customers/{customerId}/history | Retrieve customer rental history. |
| POST | /damage | Submit a damage report for a booking. |
| POST | /invoices | Generate invoice for a booking. |

# 5. Error Handling

The API follows RESTful conventions for HTTP error responses, returning meaningful HTTP status codes and error messages:  
- 400 Bad Request: Invalid or missing parameters.  
- 401 Unauthorized: Invalid API Key.  
- 404 Not Found: Requested resource not found.  
- 500 Internal Server Error: Unexpected server errors.

# 6. Logging & Auditing

The system logs create, update and delete operations including the timestamp, operation details and associated API key for audit purposes.