

# MINI – PROJECT

## Car Rental Service

### Project Overview: Car Rental Service Web API

#### Objective

The primary objective of this project is to develop a robust and efficient Web API for a car rental service. This API will enable customers to view their rental history and associated vehicle information. Additionally, it will offer functionality for filtering and counting operations to analyze vehicle popularity and rental frequency.

#### Key Features

##### *Customer Management*

- **Add Customer:** Register a new customer by inserting their details into the system.
- **Delete Customer:** Remove a customer record by their ID.
- **Update Customer:** Modify existing customer information based on their ID.
- **Search Customer:** Find customers by ID, first name, last name, email, or phone number.
- **View All Customers:** Retrieve a list of all customers in the system.
- **Total Count of Customers:** Display the total number of customers.
- **History of Rental:** Display the current customer rental history.
- **Sort Customers:** Arrange customer records in ascending or descending order based on selected fields (e.g., ID, first name, last name).

##### *Rental Management*

- **Add Rental:** Register a new vehicle rental by linking a customer and a vehicle, with validation for availability and rental dates.
- **Delete Rental:** Remove a rental record by its ID.
- **Update Rental:** Modify the details of an existing rental, ensuring the rental ID matches and the vehicle is not already rented.
- **Search Rental:** Find rentals by ID, customer ID, vehicle ID, or other criteria.
- **View All Rentals:** List all rental records in the system.
- **Filter Rentals by Date:** Retrieve rentals within a specified date range.

- **Filter Rentals by Any:** Search rentals using various criteria (e.g., customer name, vehicle details) and sort results.
- **Total Count of Rentals:** Display the total number of rentals.

### *Vehicle Management*

- **Add Vehicle:** Register a new vehicle by inserting its details into the system.
- **Delete Vehicle:** Remove a vehicle record by its ID.
- **Update Vehicle:** Modify existing vehicle information based on its ID.
- **Search Vehicle:** Find vehicles by ID, make, model, year, or color.
- **View All Vehicles:** Retrieve a list of all vehicles in the system.
- **Filter Vehicles by Any:** Search vehicles using various criteria (e.g., make, model) and sort results.
- **Total Count of Vehicles:** Display the total number of vehicles.

### *Token Management*

- **User Login:** Authenticate users (customers and admins) and provide a JWT token for access.
- **Update Last Login:** Record the last login date of the user upon successful authentication.

### **Expected Interaction:**

For a Car Rental Service API, the expected user interactions focus on making HTTP requests to perform CRUD (Create, Read, Update, Delete) operations and querying data. Below is a detailed description of how users are expected to interact with the API endpoints for managing admin, customers, rentals, vehicles, and authentication:

## **1. Customer Management**

### **Add Customer:**

- **Endpoint:** POST /api/customer/register
- **Objective:** Insert new customer details into the system.
- **Request:**

```
{  
  "firstName": "John",  
  "lastName": "Doe",
```

```
"email": sample@example.com,  
"passwordHash": "newpassword",  
"phoneNumber": "1234567890"  
}
```

- **Response:**
  - **Success (201 Created):** Confirmation of the new customer with details.
  - **Failure (400 Bad Request):** Validation errors or missing fields.

#### Delete Customer:

- **Endpoint:** DELETE /api/customer/{id}
- **Objective:** Remove a customer record by ID.
- **Request:** No body required.
- **Response:**
  - **Success (204 No Content):** Confirmation of the deletion.
  - **Failure (404 Not Found):** Customer with specified ID does not exist.

#### Update Customer:

- **Endpoint:** PUT /api/customer/{id}
- **Objective:** Modify existing customer information.
- **Request:**

```
{  
  "firstName": "John",  
  "lastName": "Smith",  
  "email": "sampleUpdate@example.com",  
  "passwordHash": "newpassword",  
  "phoneNumber": "0987654321"  
}
```
- **Response:**
  - **Success (200 OK):** Confirmation of the updated customer details.
  - **Failure (400 Bad Request or 404 Not Found):** Validation errors or customer ID mismatch.

#### Search Customer:

- **Endpoint:** GET /api/customer
- **Objective:** Find customers by ID, first name, or last name.
- **Request Parameters:**
  - id: Optional

- firstName: Optional
- lastName: Optional
- **Response:**
  - **Success (200 OK):** List of matching customers.
  - **Failure (404 Not Found):** No customers match the criteria.

#### **View All Customers:**

- **Endpoint:** GET /api/customer
- **Objective:** List all customers in the system.
- **Response:**
  - **Success (200 OK):** List of all customers.
  - **Failure (500 Internal Server Error):** Unexpected error.

#### **Total Count of Customers:**

- **Endpoint:** GET /api/customer/count
- **Objective:** Display the total number of customers.
- **Response:**
  - **Success (200 OK):** Total count of customers.
  - **Failure (500 Internal Server Error):** Unexpected error.

## **2. Rental Management**

#### **Add Rental:**

- **Endpoint:** POST /api/rental/register
- **Objective:** Create a new rental linking customers and vehicles.
- **Request:**

```
{
  "customerId": "123",
  "vehicleId": "abc",
  "rentalStartDate": "2024-01-01T00:00:00Z",
  "rentalEndDate": "2024-01-07T00:00:00Z"
}
```
- **Response:**
  - **Success (201 Created):** Confirmation of the new rental.

- **Failure (400 Bad Request):** Validation errors or issues with customer/vehicle.

#### Delete Rental:

- **Endpoint:** DELETE /api/rental/{id}
- **Objective:** Remove a rental record by ID.
- **Request:** No body required.
- **Response:**
  - **Success (204 No Content):** Confirmation of the deletion.
  - **Failure (404 Not Found):** Rental with specified ID does not exist.

#### Update Rental:

- **Endpoint:** PUT /api/rental/{id}
- **Objective:** Modify existing rental details.
- **Request:**

```
{  
  
  "customerId": "123",  
  
  "vehicleId": "abc",  
  
  "rentalStartDate": "2024-01-01T00:00:00Z",  
  
  "rentalEndDate": "2024-01-10T00:00:00Z"  
}
```
- **Response:**
  - **Success (200 OK):** Confirmation of the updated rental details.
  - **Failure (400 Bad Request or 404 Not Found):** Validation errors or rental ID mismatch.

#### Search Rental:

- **Endpoint:** GET /api/rental
- **Objective:** Find rentals by ID, customer ID, or vehicle ID.
- **Request Parameters:**
  - id: Optional
  - customerId: Optional
  - vehicleId: Optional
- **Response:**
  - **Success (200 OK):** List of matching rentals.

- **Failure (404 Not Found):** No rentals match the criteria.

#### **View All Rentals:**

- **Endpoint:** GET /api/rental
- **Objective:** List all rentals in the system.
- **Response:**
  - **Success (200 OK):** List of all rentals.
  - **Failure (500 Internal Server Error):** Unexpected error.

#### **Filter Rentals by Date:**

- **Endpoint:** GET /api/rental/filter
- **Objective:** Retrieve rentals within a specified date range.
- **Request Parameters:**
  - startDate: Optional
  - endDate: Optional
- **Response:**
  - **Success (200 OK):** List of rentals within the date range.
  - **Failure (500 Internal Server Error):** Unexpected error.

#### **Filter Rentals by Any:**

- **Endpoint:** GET /api/rental/filter/any
- **Objective:** Retrieve rentals based on various search criteria.
- **Request Parameters:**
  - Various optional filters such as firstName, lastName, vehicleName, etc.
- **Response:**
  - **Success (200 OK):** List of rentals matching the criteria.
  - **Failure (500 Internal Server Error):** Unexpected error.

#### **Total Count of Rentals:**

- **Endpoint:** GET /api/rental/count
- **Objective:** Display the total number of rentals.
- **Response:**
  - **Success (200 OK):** Total count of rentals.
  - **Failure (500 Internal Server Error):** Unexpected error.

### 3. Vehicle Management

#### Add Vehicle:

- **Endpoint:** POST /api/vehicle/register
- **Objective:** Insert new vehicle details into the system.
- **Request:**

```
{  
  "make": "Toyota",  
  "model": "Camry",  
  "year": 2022,  
  "color": "Blue"  
}
```
- **Response:**
  - **Success (201 Created):** Confirmation of the new vehicle.
  - **Failure (400 Bad Request):** Validation errors or duplicate vehicle.

#### Delete Vehicle:

- **Endpoint:** DELETE /api/vehicle/{id}
- **Objective:** Remove a vehicle record by ID.
- **Request:** No body required.
- **Response:**
  - **Success (204 No Content):** Confirmation of the deletion.
  - **Failure (404 Not Found):** Vehicle with specified ID does not exist.

#### Update Vehicle:

- **Endpoint:** PUT /api/vehicle/{id}
- **Objective:** Modify existing vehicle information.
- **Request:**

```
{  
  "make": "Toyota",  
  "model": "Corolla",  
  "year": 2023,  
  "color": "Red"  
}
```

}

- **Response:**
  - **Success (200 OK):** Confirmation of the updated vehicle details.
  - **Failure (400 Bad Request or 404 Not Found):** Validation errors or vehicle ID mismatch.

#### Search Vehicle:

- **Endpoint:** GET /api/vehicle
- **Objective:** Find vehicles by ID, make, model, year, or color.
- **Request Parameters:**
  - id: Optional
  - make: Optional
  - model: Optional
  - year: Optional
  - color: Optional
- **Response:**
  - **Success (200 OK):** List of matching vehicles.
  - **Failure (404 Not Found):** No vehicles match the criteria.

#### View All Vehicles:

- **Endpoint:** GET /api/vehicle
- **Objective:** List all vehicles in the system.
- **Response:**
  - **Success (200 OK):** List of all vehicles.
  - **Failure (500 Internal Server Error):** Unexpected error.

#### Filter Vehicles by Any:

- **Endpoint:** GET /api/vehicle/filter/any
- **Objective:** Retrieve vehicles based on various search criteria.
- **Request Parameters:**
  - Various optional filters such as make, model, year, etc.
- **Response:**
  - **Success (200 OK):** List of vehicles matching the criteria.
  - **Failure (500 Internal Server Error):** Unexpected error.

#### Total Count of Vehicles:

- **Endpoint:** GET /api/vehicle/count
- **Objective:** Display the total number of vehicles.



- **Response:**
  - **Success (200 OK):** Total count of vehicles.
  - **Failure (500 Internal Server Error):** Unexpected error.

## 4. Token Management

### User Login:

- **Endpoint:** POST /api/token/login
- **Objective:** Authenticate users and generate tokens.
- **Request:**

```
{  
    "email": "user@example.com",  
    "password": "password123"  
}
```
- **Response:**
  - **Success (200 OK):** Returns a JWT token for authentication.
  - **Failure (401 Unauthorized):** Invalid credentials.

### Update Last Login:

- **Endpoint:** Managed automatically by the system upon successful login.
- **Objective:** Update the last login timestamp for the user.
- **Interaction:** No explicit endpoint for users; handled internally by the API.

## Admin Management

### 1. Register Admin

- **Endpoint:** POST /api/admin/register
- **Objective:** Insert a new admin into the system.
- **Request:**

```
{  
  "email": "admin@example.com",  
  "passwordHash": "password",  
  "fullName": "Admin Name"  
}
```

- **Response:**
  - **Success (201 Created):** Confirmation of the new admin creation, including the admin details.
  - **Failure (400 Bad Request):** Validation errors (e.g., invalid email format, weak password).
  - **Failure (409 Conflict):** Admin with the same email or username already exists.

## 2. Delete Admin

- **Endpoint:** DELETE /api/admin/{id}
- **Objective:** Remove an admin record by ID.
- **Request:** No body required.
- **Response:**
  - **Success (204 No Content):** Confirmation of the admin deletion.
  - **Failure (404 Not Found):** Admin with the specified ID does not exist.

## 3. Update Admin

- **Endpoint:** PUT /api/admin/{id}
- **Objective:** Modify existing admin information.
- **Request:**

```
{  
  "email": "updatedadmin@example.com",  
  "passwordHash": "newpassword",  
  "fullName": "Updated Admin Name"  
}
```
- **Response:**
  - **Success (200 OK):** Confirmation of the updated admin details.

- **Failure (400 Bad Request):** Validation errors (e.g., ID mismatch, invalid email format).
- **Failure (404 Not Found):** Admin with the specified ID does not exist.

#### 4. Get All Admins

- **Endpoint:** GET /api/admin
- **Objective:** Retrieve a list of all admins.
- **Response:**
  - **Success (200 OK):** List of all admins.
  - **Failure (500 Internal Server Error):** Unexpected error.

#### 5. Get Admin by ID

- **Endpoint:** GET /api/admin/{id}
- **Objective:** Retrieve details of a specific admin by ID.
- **Request:** No body required.
- **Response:**
  - **Success (200 OK):** Details of the specified admin.
  - **Failure (404 Not Found):** Admin with the specified ID does not exist.

#### 6. Filter Admins

- **Endpoint:** GET /api/admin/Filter/Any
- **Objective:** Retrieve admins based on various search criteria.
- **Request Parameters:**
  - adminId: Optional
  - username: Optional
  - email: Optional
  - fullName: Optional
  - sortBy: Optional (default is "Username")
  - sortDescending: Optional (default is false)
- **Response:**
  - **Success (200 OK):** List of admins matching the filter criteria.
  - **Failure (500 Internal Server Error):** Unexpected error.

#### 7. Get Total Admins

- **Endpoint:** GET /api/admin/Count
- **Objective:** Display the total number of admins.
- **Response:**
  - **Success (200 OK):** Total count of admins.

- **Failure (500 Internal Server Error):** Unexpected error.