

# Module 16

## What to learn

### Dependency Injection (DI)

What is Dependency Injection?

Built-in DI in .NET Core

Configuring services

Repository Pattern and its implementation

Hands-on Generic Repository Pattern, register and consume repository

Hands-on: Creating service which inherit from generic repository and injecting the controller.

## Practice Exercise

### Practice 1

do the hands on the following topic from the videos

### Practice 2

## Understanding Dependency Injection

Explain what Dependency Injection (DI) is and how it is used in .NET Core applications. Provide a simple code example demonstrating the concept of DI in a .NET Core Web API project.

### Practice 3

## Implementing Built-in DI in .NET Core

Using the built-in Dependency Injection framework in .NET Core, create a service that fetches book details from a static list. Implement this service in a controller and demonstrate how to inject it using DI.

### Practice 4

## Configuring Services in Startup.cs

In a new .NET Core Web API project, configure a custom service in the Startup.cs file. Explain the purpose of the ConfigureServices method and how to register a service for DI.

### Practice 5

## Authentication and Authorization Basics

Define the concepts of Authentication and Authorization in the context of a Web API. Provide a code snippet that demonstrates how to set up basic authentication in a .NET Core Web API application.

### Practice 6

## Using JSON Web Token (JWT)

Explain what JSON Web Tokens (JWT) are and how they are used for securing APIs. Provide an example of generating a JWT token upon user login and how to validate it in a .NET Core Web API.

### Practice 7

## Adding a Service Layer to the 'Books' API

Enhance an existing 'Books' API by adding a service layer. Create a service that handles business logic related to book management, such as adding, updating, and retrieving book information.

### Practice 8

## Hands-on Authentication and Authorization

Implement a simple authentication and authorization mechanism in a .NET Core Web API. Create endpoints that require user roles and demonstrate how to restrict access based on user roles.

## Assignment Exercise

### Assignment 1

Create A Database design and insert record in the database. Create views In a hospital there are different departments. Patients are treated in these departments by the doctors assigned to patients. Usually each patient is treated by a single doctor, but in rare cases they will have two or three. Healthcare assistants will also attend to patients; every department has many healthcare assistants. Each patient is required to take a variety of drugs during different parts of the day such as morning, afternoon and night. Login Module API After login only below module should be accessible. If he is admin can update these details and others using only view the output. After Creating a Database design. Create ORM with database first Approach. Insert a Doctor Update a Doctor Delete a Doctor Find a report of patient assigned to a particular doctor Find a report of medicine list for a particular patient Display summary report of Doctor and patient (use Include method) After successfully login only these API should be accessible.

Assignment 2

Building a Comprehensive 'Books' API with DI, Authentication, and Authorization

Develop a complete 'Books' API that incorporates Dependency Injection, Authentication, and Authorization. The API should allow users to perform CRUD operations on books, but only authenticated users should be able to add or modify books. Use JWT for securing the API endpoints. The functional flow should include:

- User registration endpoint that stores user credentials securely.
- User login endpoint that generates a JWT token upon successful authentication.
- Endpoints for managing books (GET, POST, PUT, DELETE) that check for user authentication and roles.
- Ensure that the API returns appropriate HTTP status codes and messages based on the success or failure of operations.

Business Logic:

- Only users with the 'Admin' role can add or delete books.
- All authenticated users can view the list of books.
- Implement error handling for unauthorized access attempts.

Ensure that all outputs are displayed on the user interface (UI) and not in the console. Provide a detailed README file explaining how to set up and test the API.

Assignment 3 (68257f9be12dd826bd9eb7f5)

API Endpoints for Bill Generation and Payment System

1. User Registration (With Role Assignment)

/api/v1/users/register

Method Payload

POST	{ "email": "user@example.com", "password": "securePass123", "username": "user123", "role": "customer" } or { "email": "admin@example.com", "password": "securePass123", "username": "adminUser", "role": "admin" }
------	--

Property Name Validation Error Message			Status Code
email	Required	Email is required	400
	Format	Email is invalid	400
	Exists	User with this email already exists	400
username	Required	Username is required	400
	Length	Username must be at least 3 characters long	400
	Exists	User with this username already exists	400
password	Required	Password is required	400
role	Required	Role is required	400
	--	Role must be either "customer" or "admin"	400

Responses	Status Code	Body
Success	201	{ "data": { "message": "User registered successfully", "userId": "unique_id_here" } }
Validation Error	400	{ "errors": [ { "field": "email", "message": "Email is invalid" }, { "field": "role", "message": "Role is required" } ] }
Conflict	400	{ "message": "User with this email or username already exists" }

2. User Login

/api/v1/users/login

Method Payload

POST	{ "username": "user123", "password": "securePass123" }
------	--

Property Name	Validation	Error Message	Status Code
username	Required	Username is required	400
password	Required	Password is required	400

Responses	Status Code	Body
Success	200	{ "data": { "userId": "unique_id_here", "accessToken": "Bearer Token", "role": "customer" } }
Invalid Credentials	401	{ "message": "Invalid user credentials" }

### 3. Generate Bill (Admin Only)

#### /api/v1/bills/generate

##### Method Payload

POST { "customerId": "customer\_id\_here", "amount": 100, "dueDate": "YYYY-MM-DD" }

Property Name	Validation	Error Message	Status Code
customerId	Required	Customer ID is required	400
amount	Required	Amount is required	400
	Positive	Amount must be a positive number	400
dueDate	Required	Due date is required	400
	Date Format	Due date must be in YYYY-MM-DD format	400

Responses	Status Code	Body
Success	201	{ "data": { "message": "Bill generated successfully", "billId": "unique_bill_id" } }
Validation Error	400	{ "errors": [ { "field": "amount", "message": "Amount must be a positive number" } ] }
Unauthorized	403	{ "message": "Unauthorized access. Admin privileges are required." }

### 4. Admin View All Bills

#### /api/v1/bills

##### Method Payload

GET -

Responses	Status Code	Body
Success	200	{ "data": [ { "billId": "unique_bill_id", "customerId": "customer_id_here", "amount": 100, "status": "Unpaid", "dueDate": "YYYY-MM-DD" }, ... ] }
No Bills	404	{ "message": "No bills found" }

### 5. Get Customer Bills (View For customers)

#### /api/v1/bills/customer

##### Method Payload

GET -

Responses	Status Code	Body
Success	200	{ "data": [ { "billId": "unique_bill_id", "amount": 100, "status": "Unpaid", "dueDate": "YYYY-MM-DD" }, ... ] }
No Bills	404	{ "message": "No bills found for this customer" }

### 6. Pay Bill

#### /api/v1/bills/pay/{billId}

##### Method Payload

PATCH { "status": "Paid" }

Property Name	Validation	Error Message	Status Code
status	Required	Bill status is required	400
	--	Status must be "Paid"	400

Responses	Status Code	Body
Success	200	{ "data": { "message": "Bill payment successful", "billId": "unique_bill_id", "newStatus": "Paid" } }

Responses	Status Code	Body
Bill Not Found	404	{ "message": "Bill not found" }
Already Paid	400	{ "message": "Bill has already been paid" }
Unauthorized	403	{ "message": "Unauthorized access. Only the customer can pay the bill." }

## Task Objective

Build a RESTful API for a Bill Generation and Payment System using JavaScript with any backend framework. The API should allow:

- User registration with a specified role ("customer" or "admin") for users.
- User login functionality for both customers and admin users.
- Admin users should be able to generate bills for specific customers and view all bills along with their status (default is "Unpaid").
- Customers should be able to view their bills and make payments for their respective bills by providing the bill ID and changing the status to "Paid".

Ensure proper validation, meaningful error messages, and include scenarios that result in specific HTTP status codes to uphold business logic.

Assignment 4 (6826e2b924c5af26cafd7eec)

## API Endpoints for Enhanced Country Information Application

### Objective

Build a RESTful API that enables a mobile or web application to display a list of countries, view detailed information, manage user preferences for favorite countries, and implement additional business logic pieces to enhance the functionality of the application. The API should be thoroughly designed, adhere to best practices, and incorporate user-related functionalities.

## Note on Implementation

Always delete or reset the database before running test cases, as required by the question.

## Sample Data for Country Details

### Detail of Countries:

```
{  "countries": {    "FR": {      "countryCode": "FR",      "name": "France",      "capital": "Paris",      "population": 65273511,      "region": "Europe",      "languages": ["French"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 551695,      "demonym": "French",      "timezone": "UTC+1",      "rating": 4.5    },    "DE": {      "countryCode": "DE",      "name": "Germany",      "capital": "Berlin",      "population": 83783942,      "region": "Europe",      "languages": ["German"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 357022,      "demonym": "German",      "timezone": "UTC+1",      "rating": 4.0    },    "JP": {      "countryCode": "JP",      "name": "Japan",      "capital": "Tokyo",      "population": 126476461,      "region": "Asia",      "languages": ["Japanese"],      "currency": { "code": "JPY", "name": "Yen" },      "area": 377975,      "demonym": "Japanese",      "timezone": "UTC+9",      "rating": 4.8    },    "BR": {      "countryCode": "BR",      "name": "Brazil",      "capital": "Brasília",      "population": 212559417,      "region": "South America",      "languages": ["Portuguese"],      "currency": { "code": "BRL", "name": "Real" },      "area": 8515767,      "demonym": "Brazilian",      "timezone": "UTC-3",      "rating": 4.6    },    "AU": {      "countryCode": "AU",      "name": "Australia",      "capital": "Canberra",      "population": 25499884,      "region": "Oceania",      "languages": ["English"],      "currency": { "code": "AUD", "name": "Australian Dollar" },      "area": 7692024,      "demonym": "Australian",      "timezone": "UTC+10",      "rating": 4.3    },    "CA": {      "countryCode": "CA",      "name": "Canada",      "capital": "Ottawa",      "population": 37742154,      "region": "North America",      "languages": ["English", "French"],      "currency": { "code": "CAD", "name": "Canadian Dollar" },      "area": 9984670,      "demonym": "Canadian",      "timezone": "UTC-5",      "rating": 4.2    },    "IN": {      "countryCode": "IN",      "name": "India",      "capital": "New Delhi",      "population": 1380004385,      "region": "Asia",      "languages": ["Hindi", "English"],      "currency": { "code": "INR", "name": "Indian Rupee" },      "area": 3287263,      "demonym": "Indian",      "timezone": "UTC+5:30",      "rating": 4.1    },    "IT": {      "countryCode": "IT",      "name": "Italy",      "capital": "Rome",      "population": 60244639,      "region": "Europe",      "languages": ["Italian"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 301340,      "demonym": "Italian",      "timezone": "UTC+1",      "rating": 4.5    },    "ES": {      "countryCode": "ES",      "name": "Spain",      "capital": "Madrid",      "population": 46754778,      "region": "Europe",      "languages": ["Spanish"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 505992,      "demonym": "Spanish",      "timezone": "UTC+1",      "rating": 4.4    },    "US": {      "countryCode": "US",      "name": "United States",      "capital": "Washington, D.C.",      "population": 331002651,      "region": "North America",      "languages": ["English"],      "currency": { "code": "USD", "name": "United States Dollar" },      "area": 9372610,      "demonym": "American",      "timezone": "UTC-5",      "rating": 4.6    },    "RU": {      "countryCode": "RU",      "name": "Russia",      "capital": "Moscow",      "population": 145912025,      "region": "Europe/Asia",      "languages": ["Russian"],      "currency": { "code": "RUB", "name": "Russian Ruble" },      "area": 17098242,      "demonym": "Russian",      "timezone": "UTC+3",      "rating": 3.9    },    "ZA": {      "countryCode": "ZA",      "name": "South Africa",      "capital": "Pretoria",      "population": 59308690,      "region": "Africa",      "languages": ["Afrikaans", "English", "Zulu"],      "currency": { "code": "ZAR", "name": "South African Rand" },      "area": 1219090,      "demonym": "South African",      "timezone": "UTC+2",      "rating": 4.1    },    "MX": {      "countryCode": "MX",      "name": "Mexico",      "capital": "Mexico City",      "population": 128635261,      "region": "North America",      "languages": ["Spanish"],      "currency": { "code": "MXN", "name": "Mexican Peso" },      "area": 1964375,      "demonym": "Mexican",      "timezone": "UTC-6",      "rating": 4.0    },    "NG": {      "countryCode": "NG",      "name": "Nigeria",      "capital": "Abuja",      "population": 206139300,      "region": "Africa",      "languages": ["English"],      "currency": { "code": "NGN", "name": "Nigerian Naira" },      "area": 923768,      "demonym": "Nigerian",      "timezone": "UTC+1",      "rating": 3.8    },    "KE": {      "countryCode": "KE",      "name": "Kenya",      "capital": "Nairobi",      "population": 54985600,      "region": "Africa",      "languages": ["English", "Swahili"],      "currency": { "code": "KES", "name": "Kenyan Shilling" },      "area": 224961,      "demonym": "Kenyan",      "timezone": "UTC+3",      "rating": 3.7    },    "EG": {      "countryCode": "EG",      "name": "Egypt",      "capital": "Cairo",      "population": 102334404,      "region": "Africa",      "languages": ["Arabic"],      "currency": { "code": "EGP", "name": "Egyptian Pound" },      "area": 1001450,      "demonym": "Egyptian",      "timezone": "UTC+2",      "rating": 3.6    },    "SA": {      "countryCode": "SA",      "name": "Saudi Arabia",      "capital": "Riyadh",      "population": 34813871,      "region": "Asia",      "languages": ["Arabic"],      "currency": { "code": "SAR", "name": "Saudi Riyal" },      "area": 2149690,      "demonym": "Saudi Arabian",      "timezone": "UTC+3",      "rating": 3.5    },    "AE": {      "countryCode": "AE",      "name": "United Arab Emirates",      "capital": "Abu Dhabi",      "population": 9892000,      "region": "Asia",      "languages": ["Arabic", "English"],      "currency": { "code": "AED", "name": "UAE Dirham" },      "area": 83612,      "demonym": "Emirati",      "timezone": "UTC+4",      "rating": 4.0    },    "OM": {      "countryCode": "OM",      "name": "Oman",      "capital": "Muscat",      "population": 2970000,      "region": "Asia",      "languages": ["Arabic"],      "currency": { "code": "OMR", "name": "Omani Rial" },      "area": 114858,      "demonym": "Omani",      "timezone": "UTC+4",      "rating": 3.9    },    "KW": {      "countryCode": "KW",      "name": "Kuwait",      "capital": "Skuhayb",      "population": 3135000,      "region": "Asia",      "languages": ["Arabic"],      "currency": { "code": "KWD", "name": "Kuwaiti Dinar" },      "area": 17818,      "demonym": "Kuwaiti",      "timezone": "UTC+3",      "rating": 3.8    },    "BH": {      "countryCode": "BH",      "name": "Bahrain",      "capital": "Manama",      "population": 1506000,      "region": "Asia",      "languages": ["Arabic"],      "currency": { "code": "BHD", "name": "Bahraini Dinar" },      "area": 460,      "demonym": "Bahraini",      "timezone": "UTC+3",      "rating": 3.7    },    "QA": {      "countryCode": "QA",      "name": "Qatar",      "capital": "Doha",      "population": 2610000,      "region": "Asia",      "languages": ["Arabic"],      "currency": { "code": "QAR", "name": "Qatari Riyal" },      "area": 11586,      "demonym": "Qatari",      "timezone": "UTC+3",      "rating": 3.6    },    "LB": {      "countryCode": "LB",      "name": "Lebanon",      "capital": "Beirut",      "population": 6010000,      "region": "Asia",      "languages": ["Arabic", "French"],      "currency": { "code": "LBP", "name": "Lebanese Pound" },      "area": 10452,      "demonym": "Lebanese",      "timezone": "UTC+2",      "rating": 3.5    },    "SY": {      "countryCode": "SY",      "name": "Syria",      "capital": "Damascus",      "population": 22500000,      "region": "Asia",      "languages": ["Arabic"],      "currency": { "code": "SYP", "name": "Syrian Pound" },      "area": 185180,      "demonym": "Syrian",      "timezone": "UTC+2",      "rating": 3.4    },    "IR": {      "countryCode": "IR",      "name": "Iran",      "capital": "Tehran",      "population": 83000000,      "region": "Asia",      "languages": ["Persian"],      "currency": { "code": "IRR", "name": "Iranian Rial" },      "area": 1648195,      "demonym": "Iranian",      "timezone": "UTC+3:30",      "rating": 3.3    },    "PK": {      "countryCode": "PK",      "name": "Pakistan",      "capital": "Islamabad",      "population": 235000000,      "region": "Asia",      "languages": ["Urdu", "Punjabi", "Sindhi", "Pashto"],      "currency": { "code": "PKR", "name": "Pakistani Rupee" },      "area": 796095,      "demonym": "Pakistani",      "timezone": "UTC+5",      "rating": 3.2    },    "BD": {      "countryCode": "BD",      "name": "Bangladesh",      "capital": "Dhaka",      "population": 164000000,      "region": "Asia",      "languages": ["Bengali"],      "currency": { "code": "BDT", "name": "Bangladeshi Taka" },      "area": 147570,      "demonym": "Bangladeshi",      "timezone": "UTC+5:30",      "rating": 3.1    },    "MM": {      "countryCode": "MM",      "name": "Myanmar",      "capital": "Nay Pyi Taw",      "population": 55000000,      "region": "Asia",      "languages": ["Burmese"],      "currency": { "code": "MMK", "name": "Myanmar Kyat" },      "area": 676582,      "demonym": "Burmese",      "timezone": "UTC+6:30",      "rating": 3.0    },    "TH": {      "countryCode": "TH",      "name": "Thailand",      "capital": "Bangkok",      "population": 66000000,      "region": "Asia",      "languages": ["Thai"],      "currency": { "code": "THB", "name": "Thai Baht" },      "area": 513120,      "demonym": "Thai",      "timezone": "UTC+7",      "rating": 3.9    },    "VN": {      "countryCode": "VN",      "name": "Vietnam",      "capital": "Hanoi",      "population": 96000000,      "region": "Asia",      "languages": ["Vietnamese"],      "currency": { "code": "VND", "name": "Vietnamese Dong" },      "area": 331212,      "demonym": "Vietnamese",      "timezone": "UTC+7",      "rating": 3.8    },    "LA": {      "countryCode": "LA",      "name": "Laos",      "capital": "Vientiane",      "population": 7200000,      "region": "Asia",      "languages": ["Lao"],      "currency": { "code": "LAK", "name": "Lao Kip" },      "area": 236800,      "demonym": "Lao",      "timezone": "UTC+7",      "rating": 3.7    },    "KH": {      "countryCode": "KH",      "name": "Cambodia",      "capital": "Phnom Penh",      "population": 16000000,      "region": "Asia",      "languages": ["Khmer"],      "currency": { "code": "KHR", "name": "Cambodian Riel" },      "area": 181035,      "demonym": "Cambodian",      "timezone": "UTC+7",      "rating": 3.6    },    "MY": {      "countryCode": "MY",      "name": "Malaysia",      "capital": "Kuala Lumpur",      "population": 32000000,      "region": "Asia",      "languages": ["Malay", "English"],      "currency": { "code": "MYR", "name": "Malaysian Ringgit" },      "area": 330846,      "demonym": "Malaysian",      "timezone": "UTC+8",      "rating": 3.8    },    "SG": {      "countryCode": "SG",      "name": "Singapore",      "capital": "Singapore",      "population": 5600000,      "region": "Asia",      "languages": ["English", "Malay", "Tamil", "Chinese"],      "currency": { "code": "SGD", "name": "Singapore Dollar" },      "area": 710,      "demonym": "Singaporean",      "timezone": "UTC+8",      "rating": 4.0    },    "TW": {      "countryCode": "TW",      "name": "Taiwan",      "capital": "Taipei",      "population": 23000000,      "region": "Asia",      "languages": ["Mandarin"],      "currency": { "code": "TWD", "name": "New Taiwan Dollar" },      "area": 36192,      "demonym": "Taiwanese",      "timezone": "UTC+8",      "rating": 3.9    },    "HK": {      "countryCode": "HK",      "name": "Hong Kong",      "capital": "Hong Kong",      "population": 7400000,      "region": "Asia",      "languages": ["Cantonese", "English"],      "currency": { "code": "HKD", "name": "Hong Kong Dollar" },      "area": 2755,      "demonym": "Hong Konger",      "timezone": "UTC+8",      "rating": 3.8    },    "MO": {      "countryCode": "MO",      "name": "Macau",      "capital": "Macau",      "population": 650000,      "region": "Asia",      "languages": ["Portuguese", "Cantonese", "English"],      "currency": { "code": "MOP", "name": "Macau Pataca" },      "area": 30,      "demonym": "Macanese",      "timezone": "UTC+8",      "rating": 3.7    },    "CN": {      "countryCode": "CN",      "name": "China",      "capital": "Beijing",      "population": 1400000000,      "region": "Asia",      "languages": ["Mandarin"],      "currency": { "code": "CNY", "name": "Chinese Yuan" },      "area": 9600000,      "demonym": "Chinese",      "timezone": "UTC+8",      "rating": 3.5    },    "AU": {      "countryCode": "AU",      "name": "Australia",      "capital": "Canberra",      "population": 25499884,      "region": "Oceania",      "languages": ["English"],      "currency": { "code": "AUD", "name": "Australian Dollar" },      "area": 7692024,      "demonym": "Australian",      "timezone": "UTC+10",      "rating": 4.3    },    "NZ": {      "countryCode": "NZ",      "name": "New Zealand",      "capital": "Wellington",      "population": 4800000,      "region": "Oceania",      "languages": ["English", "Maori"],      "currency": { "code": "NZD", "name": "New Zealand Dollar" },      "area": 268826,      "demonym": "New Zealander",      "timezone": "UTC+12",      "rating": 3.9    },    "FI": {      "countryCode": "FI",      "name": "Finland",      "capital": "Helsinki",      "population": 5500000,      "region": "Europe",      "languages": ["Finnish", "Swedish"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 153300,      "demonym": "Finnish",      "timezone": "UTC+2",      "rating": 4.0    },    "SE": {      "countryCode": "SE",      "name": "Sweden",      "capital": "Stockholm",      "population": 10000000,      "region": "Europe",      "languages": ["Swedish"],      "currency": { "code": "SEK", "name": "Swedish Krona" },      "area": 450295,      "demonym": "Swedish",      "timezone": "UTC+1",      "rating": 3.9    },    "NO": {      "countryCode": "NO",      "name": "Norway",      "capital": "Oslo",      "population": 5400000,      "region": "Europe",      "languages": ["Norwegian", "Sami"],      "currency": { "code": "NOK", "name": "Norwegian Krone" },      "area": 385203,      "demonym": "Norwegian",      "timezone": "UTC+1",      "rating": 3.8    },    "DK": {      "countryCode": "DK",      "name": "Denmark",      "capital": "Copenhagen",      "population": 5600000,      "region": "Europe",      "languages": ["Danish"],      "currency": { "code": "DKK", "name": "Danish Krone" },      "area": 4309,      "demonym": "Danish",      "timezone": "UTC+1",      "rating": 3.7    },    "IS": {      "countryCode": "IS",      "name": "Iceland",      "capital": "Reykjavik",      "population": 340000,      "region": "Europe",      "languages": ["Icelandic"],      "currency": { "code": "ISK", "name": "Icelandic Krona" },      "area": 101824,      "demonym": "Icelandic",      "timezone": "UTC+0",      "rating": 3.6    },    "PT": {      "countryCode": "PT",      "name": "Portugal",      "capital": "Lisbon",      "population": 10300000,      "region": "Europe",      "languages": ["Portuguese"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 92090,      "demonym": "Portuguese",      "timezone": "UTC+0",      "rating": 3.5    },    "GR": {      "countryCode": "GR",      "name": "Greece",      "capital": "Athens",      "population": 11500000,      "region": "Europe",      "languages": ["Greek"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 131957,      "demonym": "Greek",      "timezone": "UTC+2",      "rating": 3.4    },    "CY": {      "countryCode": "CY",      "name": "Cyprus",      "capital": "Nicosia",      "population": 850000,      "region": "Europe",      "languages": ["Greek", "Turkish", "Arabic"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 9251,      "demonym": "Cypriot",      "timezone": "UTC+2",      "rating": 3.3    },    "LU": {      "countryCode": "LU",      "name": "Luxembourg",      "capital": "Luxembourg",      "population": 620000,      "region": "Europe",      "languages": ["French", "German", "Luxembourgish"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 1286,      "demonym": "Luxembourger",      "timezone": "UTC+1",      "rating": 3.2    },    "BE": {      "countryCode": "BE",      "name": "Belgium",      "capital": "Brussels",      "population": 11500000,      "region": "Europe",      "languages": ["Dutch", "French", "German"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 30528,      "demonym": "Belgian",      "timezone": "UTC+1",      "rating": 3.1    },    "NL": {      "countryCode": "NL",      "name": "Netherlands",      "capital": "Amsterdam",      "population": 17000000,      "region": "Europe",      "languages": ["Dutch"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 41850,      "demonym": "Dutch",      "timezone": "UTC+1",      "rating": 3.0    },    "DE": {      "countryCode": "DE",      "name": "Germany",      "capital": "Berlin",      "population": 83783942,      "region": "Europe",      "languages": ["German"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 357022,      "demonym": "German",      "timezone": "UTC+1",      "rating": 4.0    },    "FR": {      "countryCode": "FR",      "name": "France",      "capital": "Paris",      "population": 65273511,      "region": "Europe",      "languages": ["French"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 551695,      "demonym": "French",      "timezone": "UTC+1",      "rating": 4.5    },    "IT": {      "countryCode": "IT",      "name": "Italy",      "capital": "Rome",      "population": 60244639,      "region": "Europe",      "languages": ["Italian"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 301340,      "demonym": "Italian",      "timezone": "UTC+1",      "rating": 4.5    },    "ES": {      "countryCode": "ES",      "name": "Spain",      "capital": "Madrid",      "population": 46754778,      "region": "Europe",      "languages": ["Spanish"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 505992,      "demonym": "Spanish",      "timezone": "UTC+1",      "rating": 4.4    },    "PT": {      "countryCode": "PT",      "name": "Portugal",      "capital": "Lisbon",      "population": 10300000,      "region": "Europe",      "languages": ["Portuguese"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 92090,      "demonym": "Portuguese",      "timezone": "UTC+0",      "rating": 3.5    },    "GR": {      "countryCode": "GR",      "name": "Greece",      "capital": "Athens",      "population": 11500000,      "region": "Europe",      "languages": ["Greek"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 131957,      "demonym": "Greek",      "timezone": "UTC+2",      "rating": 3.4    },    "CY": {      "countryCode": "CY",      "name": "Cyprus",      "capital": "Nicosia",      "population": 850000,      "region": "Europe",      "languages": ["Greek", "Turkish", "Arabic"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 9251,      "demonym": "Cypriot",      "timezone": "UTC+2",      "rating": 3.3    },    "LU": {      "countryCode": "LU",      "name": "Luxembourg",      "capital": "Luxembourg",      "population": 620000,      "region": "Europe",      "languages": ["French", "German", "Luxembourgish"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 1286,      "demonym": "Luxembourger",      "timezone": "UTC+1",      "rating": 3.2    },    "BE": {      "countryCode": "BE",      "name": "Belgium",      "capital": "Brussels",      "population": 11500000,      "region": "Europe",      "languages": ["Dutch", "French", "German"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 30528,      "demonym": "Belgian",      "timezone": "UTC+1",      "rating": 3.1    },    "NL": {      "countryCode": "NL",      "name": "Netherlands",      "capital": "Amsterdam",      "population": 17000000,      "region": "Europe",      "languages": ["Dutch"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 41850,      "demonym": "Dutch",      "timezone": "UTC+1",      "rating": 3.0    },    "DE": {      "countryCode": "DE",      "name": "Germany",      "capital": "Berlin",      "population": 83783942,      "region": "Europe",      "languages": ["German"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 357022,      "demonym": "German",      "timezone": "UTC+1",      "rating": 4.0    },    "FR": {      "countryCode": "FR",      "name": "France",      "capital": "Paris",      "population": 65273511,      "region": "Europe",      "languages": ["French"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 551695,      "demonym": "French",      "timezone": "UTC+1",      "rating": 4.5    },    "IT": {      "countryCode": "IT",      "name": "Italy",      "capital": "Rome",      "population": 60244639,      "region": "Europe",      "languages": ["Italian"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 301340,      "demonym": "Italian",      "timezone": "UTC+1",      "rating": 4.5    },    "ES": {      "countryCode": "ES",      "name": "Spain",      "capital": "Madrid",      "population": 46754778,      "region": "Europe",      "languages": ["Spanish"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 505992,      "demonym": "Spanish",      "timezone": "UTC+1",      "rating": 4.4    },    "PT": {      "countryCode": "PT",      "name": "Portugal",      "capital": "Lisbon",      "population": 10300000,      "region": "Europe",      "languages": ["Portuguese"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 92090,      "demonym": "Portuguese",      "timezone": "UTC+0",      "rating": 3.5    },    "GR": {      "countryCode": "GR",      "name": "Greece",      "capital": "Athens",      "population": 11500000,      "region": "Europe",      "languages": ["Greek"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 131957,      "demonym": "Greek",      "timezone": "UTC+2",      "rating": 3.4    },    "CY": {      "countryCode": "CY",      "name": "Cyprus",      "capital": "Nicosia",      "population": 850000,      "region": "Europe",      "languages": ["Greek", "Turkish", "Arabic"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 9251,      "demonym": "Cypriot",      "timezone": "UTC+2",      "rating": 3.3    },    "LU": {      "countryCode": "LU",      "name": "Luxembourg",      "capital": "Luxembourg",      "population": 620000,      "region": "Europe",      "languages": ["French", "German", "Luxembourgish"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 1286,      "demonym": "Luxembourger",      "timezone": "UTC+1",      "rating": 3.2    },    "BE": {      "countryCode": "BE",      "name": "Belgium",      "capital": "Brussels",      "population": 11500000,      "region": "Europe",      "languages": ["Dutch", "French", "German"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 30528,      "demonym": "Belgian",      "timezone": "UTC+1",      "rating": 3.1    },    "NL": {      "countryCode": "NL",      "name": "Netherlands",      "capital": "Amsterdam",      "population": 17000000,      "region": "Europe",      "languages": ["Dutch"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 41850,      "demonym": "Dutch",      "timezone": "UTC+1",      "rating": 3.0    },    "DE": {      "countryCode": "DE",      "name": "Germany",      "capital": "Berlin",      "population": 83783942,      "region": "Europe",      "languages": ["German"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 357022,      "demonym": "German",      "timezone": "UTC+1",      "rating": 4.0    },    "FR": {      "countryCode": "FR",      "name": "France",      "capital": "Paris",      "population": 65273511,      "region": "Europe",      "languages": ["French"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 551695,      "demonym": "French",      "timezone": "UTC+1",      "rating": 4.5    },    "IT": {      "countryCode": "IT",      "name": "Italy",      "capital": "Rome",      "population": 60244639,      "region": "Europe",      "languages": ["Italian"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 301340,      "demonym": "Italian",      "timezone": "UTC+1",      "rating": 4.5    },    "ES": {      "countryCode": "ES",      "name": "Spain",      "capital": "Madrid",      "population": 46754778,      "region": "Europe",      "languages": ["Spanish"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 505992,      "demonym": "Spanish",      "timezone": "UTC+1",      "rating": 4.4    },    "PT": {      "countryCode": "PT",      "name": "Portugal",      "capital": "Lisbon",      "population": 10300000,      "region": "Europe",      "languages": ["Portuguese"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 92090,      "demonym": "Portuguese",      "timezone": "UTC+0",      "rating": 3.5    },    "GR": {      "countryCode": "GR",      "name": "Greece",      "capital": "Athens",      "population": 11500000,      "region": "Europe",      "languages": ["Greek"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 131957,      "demonym": "Greek",      "timezone": "UTC+2",      "rating": 3.4    },    "CY": {      "countryCode": "CY",      "name": "Cyprus",      "capital": "Nicosia",      "population": 850000,      "region": "Europe",      "languages": ["Greek", "Turkish", "Arabic"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 9251,      "demonym": "Cypriot",      "timezone": "UTC+2",      "rating": 3.3    },    "LU": {      "countryCode": "LU",      "name": "Luxembourg",      "capital": "Luxembourg",      "population": 620000,      "region": "Europe",      "languages": ["French", "German", "Luxembourgish"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 1286,      "demonym": "Luxembourger",      "timezone": "UTC+1",      "rating": 3.2    },    "BE": {      "countryCode": "BE",      "name": "Belgium",      "capital": "Brussels",      "population": 11500000,      "region": "Europe",      "languages": ["Dutch", "French", "German"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 30528,      "demonym": "Belgian",      "timezone": "UTC+1",      "rating": 3.1    },    "NL": {      "countryCode": "NL",      "name": "Netherlands",      "capital": "Amsterdam",      "population": 17000000,      "region": "Europe",      "languages": ["Dutch"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 41850,      "demonym": "Dutch",      "timezone": "UTC+1",      "rating": 3.0    },    "DE": {      "countryCode": "DE",      "name": "Germany",      "capital": "Berlin",      "population": 83783942,      "region": "Europe",      "languages": ["German"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 357022,      "demonym": "German",      "timezone": "UTC+1",      "rating": 4.0    },    "FR": {      "countryCode": "FR",      "name": "France",      "capital": "Paris",      "population": 65273511,      "region": "Europe",      "languages": ["French"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 551695,      "demonym": "French",      "timezone": "UTC+1",      "rating": 4.5    },    "IT": {      "countryCode": "IT",      "name": "Italy",      "capital": "Rome",      "population": 60244639,      "region": "Europe",      "languages": ["Italian"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 301340,      "demonym": "Italian",      "timezone": "UTC+1",      "rating": 4.5    },    "ES": {      "countryCode": "ES",      "name": "Spain",      "capital": "Madrid",      "population": 46754778,      "region": "Europe",      "languages": ["Spanish"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 505992,      "demonym": "Spanish",      "timezone": "UTC+1",      "rating": 4.4    },    "PT": {      "countryCode": "PT",      "name": "Portugal",      "capital": "Lisbon",      "population": 10300000,      "region": "Europe",      "languages": ["Portuguese"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 92090,      "demonym": "Portuguese",      "timezone": "UTC+0",      "rating": 3.5    },    "GR": {      "countryCode": "GR",      "name": "Greece",      "capital": "Athens",      "population": 11500000,      "region": "Europe",      "languages": ["Greek"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 131957,      "demonym": "Greek",      "timezone": "UTC+2",      "rating": 3.4    },    "CY": {      "countryCode": "CY",      "name": "Cyprus",      "capital": "Nicosia",      "population": 850000,      "region": "Europe",      "languages": ["Greek", "Turkish", "Arabic"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 9251,      "demonym": "Cypriot",      "timezone": "UTC+2",      "rating": 3.3    },    "LU": {      "countryCode": "LU",      "name": "Luxembourg",      "capital": "Luxembourg",      "population": 620000,      "region": "Europe",      "languages": ["French", "German", "Luxembourgish"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 1286,      "demonym": "Luxembourger",      "timezone": "UTC+1",      "rating": 3.2    },    "BE": {      "countryCode": "BE",      "name": "Belgium",      "capital": "Brussels",      "population": 11500000,      "region": "Europe",      "languages": ["Dutch", "French", "German"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 30528,      "demonym": "Belgian",      "timezone": "UTC+1",      "rating": 3.1    },    "NL": {      "countryCode": "NL",      "name": "Netherlands",      "capital": "Amsterdam",      "population": 17000000,      "region": "Europe",      "languages": ["Dutch"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 41850,      "demonym": "Dutch",      "timezone": "UTC+1",      "rating": 3.0    },    "DE": {      "countryCode": "DE",      "name": "Germany",      "capital": "Berlin",      "population": 83783942,      "region": "Europe",      "languages": ["German"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 357022,      "demonym": "German",      "timezone": "UTC+1",      "rating": 4.0    },    "FR": {      "countryCode": "FR",      "name": "France",      "capital": "Paris",      "population": 65273511,      "region": "Europe",      "languages": ["French"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 551695,      "demonym": "French",      "timezone": "UTC+1",      "rating": 4.5    },    "IT": {      "countryCode": "IT",      "name": "Italy",      "capital": "Rome",      "population": 60244639,      "region": "Europe",      "languages": ["Italian"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 301340,      "demonym": "Italian",      "timezone": "UTC+1",      "rating": 4.5    },    "ES": {      "countryCode": "ES",      "name": "Spain",      "capital": "Madrid",      "population": 46754778,      "region": "Europe",      "languages": ["Spanish"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 505992,      "demonym": "Spanish",      "timezone": "UTC+1",      "rating": 4.4    },    "PT": {      "countryCode": "PT",      "name": "Portugal",      "capital": "Lisbon",      "population": 10300000,      "region": "Europe",      "languages": ["Portuguese"],      "currency": { "code": "EUR", "name": "Euro" },      "area": 92090,      "demonym": "Portuguese",      "timezone": "UTC+0",      "rating": 3.5    },    "GR": {      "countryCode": "GR",      "name": "Greece",      "capital": "Athens",      "population": 11500000,      "region": "Europe",      "languages": ["Greek"],      "currency": { "code":
```

"Mexico", "capital": "Mexico City", "population": 128932753, "region": "North America", "languages": ["Spanish"], "currency": { "code": "MXN", "name": "Mexican Peso" }, "area": 1964375, "demonym": "Mexican", "timezone": "UTC-6", "rating": 4.2 }, "GB": { "countryCode": "GB", "name": "United Kingdom", "capital": "London", "population": 67886011, "region": "Europe", "languages": ["English"], "currency": { "code": "GBP", "name": "British Pound" }, "area": 243610, "demonym": "British", "timezone": "UTC+0", "rating": 4.3 }, "MY": { "countryCode": "MY", "name": "Malaysia", "capital": "Kuala Lumpur", "population": 32365999, "region": "Asia", "languages": ["Malay", "English"], "currency": { "code": "MYR", "name": "Malaysian Ringgit" }, "area": 330803, "demonym": "Malaysian", "timezone": "UTC+8", "rating": 4.0 }, "ID": { "countryCode": "ID", "name": "Indonesia", "capital": "Jakarta", "population": 273523615, "region": "Asia", "languages": ["Indonesian"], "currency": { "code": "IDR", "name": "Indonesian Rupiah" }, "area": 1904569, "demonym": "Indonesian", "timezone": "UTC+7", "rating": 4.1 } }

## Enhanced Business Logic Requirements

- Each user should be able to save multiple preferred countries.
- Users should be able to rate their preferred countries (1 to 5 stars).
- The API should allow users to get a list of countries based on their ratings.
- Implement search functionality to find countries by name or capital.

### 1. Get List of Countries

#### /api/v1/countries

**Method Payload**

GET	N/A
-----	-----

**Responses Status Code Body**

Success	200	{ "data": [...], "message": "Countries fetched successfully", "success": true }
---------	-----	---

### 2. Search Countries

#### /api/v1/countries/search

**Method Payload**

GET	{ "query": "France" }
-----	-----------------------

Property Name	Validation	Error Message	Status Code
query	Required	Search query is required	400

Responses	Status Code	Body
Success	200	{ "data": <Country-specific data: France>, "message": "Countries matching the query fetched successfully." }
Not Found	404	{ "message": "No countries found matching your query." }

### 3. Get Details of a Selected Country

#### /api/v1/countries/{countryCode}

**Method Payload**

GET	N/A
-----	-----

Responses	Status Code	Body
Success	200	{ "data": <Country-specific data: countryCode>, "message": "Country details fetched successfully", "success": true }
Not Found	404	{ "message": "Country not found" }

### 4. Save Preferred Country with Rating

#### /api/v1/preferences/countries

**Method Payload**

POST	{ "countryCode": "FR", "rating": 5 }
------	--------------------------------------

Property Name	Validation	Error Message	Status Code
countryCode	Required	Country code is required	400

Property Name	Validation Error Message	Status Code
rating	Required Rating is required	400
	Range Rating must be between 1 and 5	400
	Exists Country must exist in the database	404

Responses	Status Code	Body
Success	201	{ "message": "Country saved as preferred with rating successfully" }
Validation Error	400	{ "message": "Rating must be between 1 and 5" }
Not Found	404	{ "message": "Country not found" }

## 5. Remove Preferred Country

/api/v1/preferences/countries/{countryCode}

Method Payload

DELETE N/A

Property Name	Validation Error Message	Status Code
countryCode	Required Country code is required	400
	Exists Country must exist in preferences	404

Responses	Status Code	Body
Success	200	{ "message": "Country removed from preferred successfully" }
Validation Error	400	{ "message": "Country code is required" }
Not Found	404	{ "message": "Country must exist in preferences" }

## 6. Get Preferred Countries with Ratings

/api/v1/preferences/countries

Method Payload

GET N/A

Responses	Status Code	Body
Success	200	{ "data": [ <countries data>... ], "message": "Preferred countries fetched successfully" }
Not Found	404	{ "message": "No preferred countries found" }

Assignment 5 (68283007e12dd826bd9eb83d)

## API Endpoints for Job Application Portal with Admin User Implementation

### Task Objective

Build an API for a Job Application Portal that includes user registration with a role assignment (either 'user' or 'admin'), login functionality, and manages job postings by admin users. Admin users will have special privileges to manage job applications, while normal users will only be able to apply for jobs and view their applications. Ensure proper validation and meaningful error messages are provided in the API responses, including scenarios that result in 409 Conflict status codes to handle specific business logic conflicts.

### 1. User Registration (With Role Assignment)

/api/v1/users/register

Method Payload

POST { "email": "user@example.com", "password": "securePass123", "username": "applicantUser", "role": "user" } or { "email": "admin@example.com", "password": "securePass123", "username": "adminUser", "role": "admin" }

Property Name	Validation Error Message	Status Code
email	Required Email is required	400
	Format Email is invalid	400
	Exists User with this email already exists	409

Property Name	Validation Error Message	Status Code
username	Required	Username is required 400
	Length	Username must be at least 3 characters long 400
	Exists	User with this username already exists 409
password	Required	Password is required 400
role	Required	Role is required 400
	--	Role must be one of the following: ["user", "admin"] 400

Responses	Status Code	Body
Success	201	{ "data": { "message": "User registered successfully", "userId": "unique_id_here" } }
Validation Error	400	{ "errors": [ { "field": "email", "message": "Email is invalid" }, { "field": "role", "message": "Role must be either 'user' or 'admin'" } ] }
Conflict	409	{ "message": "User with this email or username already exists" }

## 2. User Login

### /api/v1/users/login

#### Method Payload

POST { "username": "applicantUser", "password": "securePass123" }

Property Name	Validation Error Message	Status Code
username	Required	Username is required 400
password	Required	Password is required 400

Responses	Status Code	Body
Success	200	{ "data": { "userId": "unique_id_here", "accessToken": "Bearer Token", "role": "user" } }
Invalid Credentials	401	{ "message": "Invalid user credentials" }

## 3. Job Posting (Admin Only)

### /api/v1/jobs

#### Method Payload

POST { "title": "Software Engineer", "description": "Job description goes here", "position": "Software Developer" }

Property Name	Validation Error Message	Status Code
title	Required	Job title is required 400
description	Required	Job description is required 400
position	Required	Position is required 400

Responses	Status Code	Body
Success	201	{ "data": { "message": "Job posted successfully", "jobId": "unique_id_here" } }
Validation Error	400	{ "errors": [ { "field": "title", "message": "Job title is required" } ] }
Unauthorized	401	{ "message": "Unauthorized access. Admin privileges are required." }

## 4. User Job Application

### /api/v1/job-applications

#### Method Payload

POST { "email": "applicant@gmail.com", "name": "John Doe", "positionId": "jobId\_here" }

Property Name	Validation Error Message	Status Code
email	Required	Email is required 400
	Format	Email is invalid 400
	Exists	An application with this email already exists 409
name	Required	Name is required 400
positionId	Required	Position ID is required 400
	Exists	Provided job ID does not exist 404

Responses	Status Code	Body
Success	201	{ "data": { "message": "Job application submitted successfully", "applicationId": "unique_id_here" } }
Validation Error	400	{ "errors": [ { "field": "email", "message": "Email is invalid" } ] }
Conflict	409	{ "message": "An application with this email already exists" }
Not Found	404	{ "message": "Provided job ID does not exist" }

## 5. Admin: View All Job Applications

/api/v1/jobs/{jobId}/applications

### Method Payload

GET -

Responses	Status Code	Body
Success	200	{ "data": [ { "applicationId": "id_here", "email": "applicant@gmail.com", "name": "John Doe", "status": "Under Review" }, ... ] }
Not Found	404	{ "message": "No applications found for this job" }
Unauthorized	403	{ "message": "Unauthorized access. Admin privileges are required." }

## 6. View User's Own Applications

/api/v1/my-applications

### Method Payload

GET -

Responses	Status Code	Body
Success	200	{ "data": [ { "applicationId": "id_here", "email": "applicant@gmail.com", "name": "John Doe", "status": "Under Review" }, ... ] }
Not Found	404	{ "message": "No applications found for this user" }

## 7. Admin: Update Job Application Status

/api/v1/job-applications/{applicationId}/status

### Method Payload

PATCH { "status": "Interview Scheduled" }

Property Name	Validation Error Message	Status Code
status	Required Status is required	400
--	Status must be one of the following: ["Under Review", "Interview Scheduled", "Rejected", "Hired"]	400
Exists	Application status cannot be updated if already marked as 'Rejected' or 'Hired'	409

Responses	Status Code	Body
Success	200	{ "data": { "message": "Application status updated successfully", "applicationId": "unique_id_here", "newStatus": "Interview Scheduled" } }
Validation Error	400	{ "errors": [ { "field": "status", "message": "Status must be one of the following: ['Under Review', 'Interview Scheduled', 'Rejected', 'Hired']" } ] }
Conflict	409	{ "message": "Application status cannot be updated if already marked as 'Rejected' or 'Hired'" }
Not Found	404	{ "message": "Application not found" }
Unauthorized	403	{ "message": "Unauthorized access. Admin privileges are required." }

Assignment 6 (682da8195611ba4b46206687)

API Endpoints for Vehicle Service Booking Management System (VSBMS)



You are required to build a RESTful API for a **Vehicle Service Booking Management System (VSBMS)** in .NET Core Web API. The system includes user management, vehicle CRUD operations, and booking services. It also incorporates features like JWT-based authentication and role-based authorization, custom validation, middleware handling, file uploads, soft deletes, and automatic audit tracking.

## Special Instructions

### Entity Metadata

All entities must include the following fields:

CreatedById  
ModifiedById  
CreatedDate  
ModifiedDate


### Soft Delete

All delete operations should implement **soft delete** using an IsDeleted flag. Soft-deleted records must not appear in API results.

### Image Upload (Vehicle)

Image should be uploaded with the vehicle POST request using multipart/form-data.  
Save image to the Uploads folder.  
Register Uploads as a static file folder.  
Store the relative URL (e.g., /Uploads/car.png) in the database.

### Date-Sensitive Testing Note

 The current date is 19 May 2025. Ensure booking date validation handles future dates dynamically so automated tests behave correctly.

## Authentication & Authorization

Use JWT Bearer Tokens  
Roles: admin, customer  
Use role-based authorization on endpoints

403 Forbidden Response Example:

```
{ "message": "You are not authorized to perform this action." }
```

## Business Rules (Service Charges)

Booking charges must be automatically applied based on the selected serviceType:

### Service Type Charge (INR)

Basic	500
Premium	1000
Full	1500

 The charge is not passed by the client. It is calculated server-side and stored with the booking.

## 1. User Registration

/api/v1/auth/register

### Method Payload

POST { "name": "John Doe", "email": "john@example.com", "phone": "9999999999", "password": "Pass@123", "role": "customer" }

Property	Name	Validation	Error Message	Status Code
name	Required		"Name is required."	400
email	Required, Email, Unique		"A valid and unique email is required."	400
phone	Required, 10-digit		"Phone number must be 10 digits."	400
password	Required		"Password is required."	400

Property Name	Validation	Error Message	Status Code
role	Must be "admin" or "customer"	Role must be 'admin' or 'customer'.	400

Responses	Status Code	Body
Success	201	{ "message": "User registered successfully." }
Validation Error	400	{ "errors": [ { "field": "email", "message": "A valid and unique email is required." } ] }
Conflict	409	{ "message": "User with this email already exists." }

## 2. User Login

### /api/v1/auth/login

#### Method Payload

POST { "email": "john@example.com", "password": "Pass@123" }

Property Name	Validation	Error Message	Status Code
email	Required	"Email is required."	400
password	Required	"Password is required."	400

Responses	Status Code	Body
Success	200	{ "accessToken": "jwt", "role": "customer" }
Invalid Credentials	401	{ "message": "Invalid credentials." }

## 3. Add Vehicle

### /api/v1/vehicles

#### Method Payload

POST multipart/form-data with fields: { "make": "Toyota", "model": "Corolla", "registrationNumber": "ABC1234" } and file:imageFile

Property Name	Validation	Error Message	Status Code
imageFile	Required, jpg/png, max 2MB	"Image must be in .jpg or .png format and less than 2MB."	400
registrationNumber	Required, Unique	"Unique registration number per user is required."	400

Responses	Status Code	Body
Success	201	{ "message": "Vehicle added successfully.", "imageUrl": "/Uploads/car.png", "vehicleId": "<unique_vehicleId>" }
Validation Error	400	{ "errors": [ { "field": "imageFile", "message": "Image must be in .jpg or .png format." } ] }
Forbidden	403	{ "message": "You are not authorized to perform this action." }
Conflict	409	{ "message": "Duplicate registration number." }

## 4. Get Vehicles (Own)

### /api/v1/vehicles

#### Method Payload

GET -

Responses	Status Code	Body
Success	200	{ "vehicles": [ { "id": "...", "make": "...", "model": "...", "registrationNumber": "...", "imageUrl": "..." }, ... ] }
Forbidden	403	{ "message": "You are not authorized to perform this action." }

## 5. Get Vehicle by ID

### /api/v1/vehicles/{id}

#### Method Payload

GET -

Responses	Status Code	Body
Success	200	{ "vehicle": { "id": "...", "make": "...", "model": "...", "registrationNumber": "...", "imageUrl": "..." } }
Forbidden	403	{ "message": "Unauthorized access." }
Not Found	404	{ "message": "Entity not found." }

## 6. Update Vehicle

/api/v1/vehicles/{id}

### Method Payload

PUT multipart/form-data with fields: { "make": "Honda", "model": "Civic", "registrationNumber": "XYZ123" } and optional file:imageFile

Property Name	Validation	Error Message	Status Code
imageFile	jpg/png, max 2MB	"Image must be in .jpg or .png format and less than 2MB."	400

Responses	Status Code	Body
Success	204	(empty body)
Validation Error	400	{ "errors": [ { "field": "imageFile", "message": "Image must be in .jpg or .png format." } ] }
Forbidden	403	{ "message": "Unauthorized access." }
Not Found	404	{ "message": "Entity not found." }

## 7. Delete Vehicle

/api/v1/vehicles/{id}

### Method Payload

DELETE -

Responses	Status Code	Body
Success	204	(empty body)
Forbidden	403	{ "message": "Unauthorized access." }
Not Found	404	{ "message": "Entity not found." }

## 8. Book Service

/api/v1/bookings

### Method Payload

POST { "vehicleId": "guid", "bookingDate": "2025-05-20", "serviceType": "Premium" }

Property Name	Validation	Error Message	Status Code
bookingDate	Future date, not Sunday	"Booking date must be a future date and not Sunday."	422
serviceType	Must be Basic/Premium/Full	"Invalid service type."	422

Responses	Status Code	Body
Success	201	{ "message": "Service booked successfully with charges applied.", "bookingId": "..." }
Validation Error	422	{ "errors": [ { "field": "bookingDate", "message": "Booking date must be a future date and not Sunday." } ] }
Forbidden	403	{ "message": "You are not authorized to perform this action." }

## 9. Get Own Bookings

/api/v1/bookings/customer

### Method Payload

GET -

Responses	Status Code	Body
Success	200	{ "bookings": [ { "id": "...", "vehicleId": "...", "serviceType": "...", "bookingDate": "...", ... } ] }
Forbidden	403	{ "message": "You are not authorized to perform this action." }

## 10. Get All Bookings

/api/v1/bookings

### Method Payload

GET -

#### Responses Status Code Body

Success	200	{ "bookings": [ { "id": "...", "vehicleId": "...", "serviceType": "...", "bookingDate": "...", ... } ] }
Forbidden	403	{ "message": "You are not authorized to perform this action." }

## 11. Get Booking by ID

/api/v1/bookings/{id}

#### Method Payload

GET -

#### Responses Status Code Body

Success	200	{ "booking": { "id": "...", "vehicleId": "...", "serviceType": "...", "bookingDate": "...", ... } }
Forbidden	403	{ "message": "Unauthorized access." }
Not Found	404	{ "message": "Entity not found." }

## 12. Cancel Booking (Soft Delete)

/api/v1/bookings/{id}

#### Method Payload

DELETE -

#### Responses Status Code Body

Success	204	(empty body)
Forbidden	403	{ "message": "Unauthorized access." }
Not Found	404	{ "message": "Entity not found." }

#### Online Reference

<https://docs.microsoft.com/en-us/aspnet/core/security/authentication/identity-api-authorization?view=aspnetcore-5.0>

### .NET Core Web API

#### WEB API (old)

#### Authentication And Authorization (WEBAPI)(old)

#### FullStackDevelopment\_With\_Dotnet\_AND\_Angular