.NET Car Dealership API Coding Challenge

Overview

Create a car dealership management system using .NET 9 **API**. The system should demonstrate CRUD operations, role-based access, and security best practices.

Core Requirements

1. User Management

- **Register**: Create new user accounts (Customer/Admin roles)
- **Login**: Authenticate existing users with role-based access

2. Admin Use Cases

- **Add Vehicle**: Add new cars to inventory
- **Update Vehicle**: Modify existing vehicle details
- **View All Customers**: List registered customers
- **Process Sale**: Complete vehicle purchase transactions

3. Customer Use Cases

- **Browse Vehicles**: View available cars with filtering
- **View Vehicle Details**: Get detailed information about specific cars
- **Purchase Request**: Request to buy a vehicle
- **View Purchase History**: See their transaction history

Technical Requirements

- Use .NET 9 ASP.NET Core Web API
- Implement role-based authorization (Admin/Customer)
- The following actions must be protected with One-Time Password (OTP) Login, Register, Purchase Request, and Update Vehicle. Implement the complete OTP flow including generation, validation, expiration handling, and storage
- For OTP delivery, you can simulate it (console output) - no actual SMS/email integration required
- Pre-populate the system with at least 10 sample vehicles and at least one admin user
- Implement proper error handling
- Include basic input validation and API documentation

Deliverables

1. Complete source code with proper project structure
2. Brief README explaining:
   - How to run the API
   - Available endpoints and their usage
   - Any assumptions or design decisions made

Evaluation Criteria

- API design and RESTful principles
- Code organization and structure
- Role-based access control implementation
- OTP implementation
- Error handling and edge cases
- Decision-making rationale
- Code readability and documentation

Bonus Points

- Swagger/OpenAPI documentation
- Configuration management
- Logging implementation
- Input sanitization
- Docker containerization

Note: AI usage is encouraged, feel free to use AI assistance for coding, debugging, or research.