.NET Developer Task

ProgressSoft Corporation, Apollo Team

The goal of this task is to develop a web application for managing business card information, focusing on clean code and modern design principles. The backend APIs must be written in C#, and the database layer can be implemented with either SQL Server or Oracle. The application should also support features like file import, export, and optional filtering of data.

Deadline: This assignment is due two weeks from the date it is assigned.

In addition, the source code must be uploaded to a GitHub repository, following best practices for version control, with a clear commit history and a detailed README file to guide the setup and usage of the application.

Requirements:

1. **Technology Stack**:
   - **Backend**: .NET Core (minimum .net 6) Web API (C#)
   - **Frontend**: Angular for the user interface
   - **Database**: SQL Server, Oracle, or Postgresql.
   - **Photo Encoding**: Base64
2. **Business Card Information**:
   - Fields: Name, Gender, Date of Birth, Email, Phone, Photo (optional, max 1MB), Address.
   - Photo is encoded as a base64 string for import/export.

Tasks Breakdown:

1. **Backend (C# APIs in .NET Core)**:
   - **Entity Definition**: Define a BusinessCard model with all required properties.
   - **API Endpoints**:
     - **Create New Business Card**:
       - Accept input from the UI and through file import (XML, CSV, optional QR code).

- **View Business Cards**:
  - API to list all business cards.
- **Delete Business Card**:
  - API to delete a specific card.
- **Export Business Cards**:
  - Export to XML and CSV.
- **Optional Filtering**: By Name, DOB, Phone, Gender, or Email.

2. **Frontend (Angular)**:
   - **Add New Business Card**:
     - Create a form in Angular for user input.
     - Include drag-and-drop or file upload options for XML, CSV, and QR code import.
     - Display a preview of the business card before submitting.
   - **List Business Cards**:
     - A page that lists all stored business cards.
     - Option to delete or export a business card.
   - **Optional Filtering**: Add filtering options in the UI to refine the displayed results.
3. **File Imports (XML/CSV)**:
   - **File Handling in Backend**: Parse XML/CSV files in C.
   - **Preview before Submitting**: Temporarily store the imported data in the frontend and display it for user review before making the API call to save.
4. **Optional QR Code Import**:
   - **QR Code Parsing**: Use a library like ZXing.Net to extract data from QR codes (excluding the photo).
5. **Database**:
   - Use either Oracle, SQL Server, or Postgresql for persistence. Define the schema to store business card data.

6. **Unit Tests**:
    - Create unit tests for the backend implementation and ensure that critical paths like data import/export are tested.

Deliverables:

1. **Source Code**:
    - Clean and modular code following SOLID principles.
2. **GitHub Repository:**
    - Ensure the project is uploaded to GitHub, with clear commit history and comments.
    - Include a well-structured **README.md** file in the repository for setup instructions, dependencies, and usage guidelines.
3. **ReadMe File**:
    - Include a section describes the implementation, and setup instructions.
4. **Database file**:
    - Provide an SQL or Oracle DB dump for testing purposes.

Wish you the best of luck (Apollo Team)