Name: Achuth Chandra Moonnamkuttiyil
UMD Email ID: achuth@umd.edu
UMD Directory ID: Achuth
UID: 120311482

Instructions To Run The Project Locally

Step 1: Install Required Software
Before downloading the project, make sure you have the following software installed:

1. **Git**: A distributed version control system for tracking changes in source code.
   - Download Git
2. **.NET SDK**: Includes everything needed to build and run .NET projects.
   - Download .NET SDK
3. **Microsoft SQL Server**: A relational database management system required for your project.
   - Download SQL Server (Express or Developer edition will work).
4. **SQL Server Management Studio (SSMS)**: A tool to manage and connect to SQL Server databases.
   - Download SSMS
5. **Visual Studio Code** or **Visual Studio**: A code editor or IDE to work on the project.
   - Download Visual Studio Code
   - Download Visual Studio

Step 2: Clone the GitLab Repository

1. **Open Git Bash or Command Prompt** on your machine.
2. **Navigate to the directory** where you want to clone the repository. You can use the following command to change directories:
   cd path/to/your/folder
3. **Clone the Repository** from GitLab using the provided HTTPS URL:
   git clone --branch Phase3
   https://code.umd.edu/achuth/ENPM680Fall2024Project-achuth.git
4. **Navigate to the project directory** after cloning:
   cd ENPM680Fall2024Project-achuth

Step 3: Install Dependencies
Once the repository is cloned, you'll need to restore the dependencies:

1. **Open a terminal or command prompt** inside the project folder.
2. Run the following command to restore the NuGet packages:
   dotnet restore
   This will install all the necessary dependencies for the project.

Step 4: Configure SQL Server and Connection String

4.1: Install and Set Up SQL Server
If you haven't already installed **Microsoft SQL Server** and **SSMS**:

1. **Install SQL Server**: During installation, choose the edition (Express or Developer) and set up your SQL Server instance.
2. **Install SSMS**: After SQL Server is installed, use **SSMS** to manage your database.
    - SSMS Setup Guide

4.2: Create the Database

1. **Open SSMS** and connect to your local SQL Server instance.
2. **Create a new database**:
    - Right-click on **Databases** and choose **New Database...**
    - Name the database something like Edusync.
3. **Note your SQL Server connection details**, including:
    - Server name
    - Authentication method (SQL Server Authentication or Windows Authentication)
    - Database name (e.g., Edusync)

4.3: Modify Connection String in appsettings.json
You will need to update the **connection string** in the appsettings.json file to reflect your local SQL Server configuration.

1. Open the appsettings.json file in your project.
2. Find the ConnectionStrings section and modify it as needed. Here's an example connection string:

```
"ConnectionStrings": {
  "DefaultConnection": "Server=YOUR_SERVER_NAME;Database=Edusync;User Id=YOUR_USERNAME;Password=YOUR_PASSWORD;Trusted_Connection=False;MultipleActiveResultSets=true;"
}
```

Replace the placeholders:

- YOUR_SERVER_NAME: The name of your SQL Server instance.
- YOUR_USERNAME: Your SQL Server username (if using SQL Server Authentication).
- YOUR_PASSWORD: Your SQL Server password (if using SQL Server Authentication).

If you are using **Windows Authentication**, you can use this connection string instead:

```
"ConnectionStrings": {
  "DefaultConnection": "Server=YOUR_SERVER_NAME;Database=Edusync;Trusted_Connection=True;MultipleActiveResultSets=true;"
}
```

Step 5: Update Database Schema (Apply Migrations)
Since project uses **Microsoft SQL Server** and includes **Entity Framework** for database management, you need to apply the migrations to create the necessary tables in the SQL Server database.

1. **Open a terminal** inside the project folder.
2. Run the following command to update the database with the latest schema:
   dotnet ef database update
   This will apply any pending migrations and create/update the database tables in SQL Server.

   Note: If you face any issues related to the migration process, ensure that the Entity Framework Core tools are installed globally with:
   dotnet tool install --global dotnet-ef

Step 6: Build and Run the Project
Once the connection string is set up, and the database is ready, you can build and run the project.

1. **Build the Project**:
   dotnet build
2. **Run the Project**:
   dotnet run
   For https:
   dotnet run --launch-profile https

This will start the application, and will display the link for browsing.

Step 7: Running Unit Tests
If you want to run the unit tests, follow these steps:

1. Navigate to the test project folder inside the solution:
   cd ENPM680Fall2024Project-achuth/Edusync.Tests
2. Run the unit tests using the following command:
   dotnet test
   This will execute all the unit tests that are part of the project.