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

Build and Installation Guide

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 Phase5
   https://code.umd.edu/achuth/ENPM680Fall2024Project-achuth.git
4. **Navigate to the project directory** after cloning:
   cd ENPM680Fall2024Project-achuth

Step 3: Set Up the SQL Server Database

1. **Install Microsoft SQL Server** if you don't have it installed. You can download it here.
2. **Install SQL Server Management Studio (SSMS)** for managing the database. You can download it here.
   Setting up Database – Method 1 (Preferred):
3. **Restore the Database Backup:**
   - Obtain the .bak file (backup file) from the project's repository. Its named as "SchoolManagementDb.bak". Copy it to a place where SSMS can access it

without any issues. (Like C:\Program Files\Microsoft SQL Server\MSSQL16.SQLEXPRESS\MSSQL\Backup)
    - Open **SQL Server Management Studio (SSMS)**.
    - Connect to your local SQL Server instance.
    - Right-click on **Databases** in the Object Explorer and select **Restore Database**.
    - In the **Source** section, select **Device, click on ...** and then locate and select the SchoolManagementDb.bak file.
    - In the **Destination** section, make sure the database name is set to SchoolManagementDb.
    - Click **OK** to restore the database.
4. **Setting up Database – Method 2 (If method 1 doesn't work for you):**
    - SchoolManagementDb.sql file is available in the project folder.
    - Open **SQL Server Management Studio (SSMS)** and Connect to your local SQL Server instance.
    - Go to File menu -> Open -> File and then browse and select SchoolManagementDb.sql file to open the sql script in the SSMS
    - Click on Execute. This will create a database called SchoolManagementDb inside your database. Refresh the database to see it.

Step 3: Update the appsettings.json Connection String

1. **Open the Project in Visual Studio Code**:
    - If you don't have VS Code installed, download it from here.
    - In VS Code, open the project folder (the one cloned in Step 1).

2. **Modify the appsettings.json File:**
    - Navigate to the appsettings.json file located in the project's root directory.
    - Update the **connection string** to point to your local SQL Server instance.
        - Example connection string:
        - If you have windows authentication for the SQL database:

```
{
  "ConnectionStrings": {
    "DefaultConnection":
"Server=YOUR_SERVER_NAME;Database=SchoolManagementDb;Trusted_Connection=false;MultipleActiveResultSets=true;Encrypt=false;Integrated Security=True;"
  }
}
```

- Replace YOUR_SERVER_NAME with your local SQL Server instance name. If you're using the default instance, you can use (local) or localhost.

- If you have SQL Server authentication for database, with SQL Server name 'DESKTOP-BHPP4CG\\SQLEXPRESS' (for example) :

```
{
  "ConnectionStrings": {
    "SchoolManagementDbConnection": "Server=DESKTOP-
BHPP4CG\\SQLEXPRESS,
1433;Database=SchoolManagementDb;Trusted_Connection=false;Multiple
ActiveResultSets=true;Encrypt=false;User id=your_user_id;
Password=your_password;"
  }
```

- Replace your_user_id and your_password with your SQL Server authentication username and password.

3. **Save the Changes.**

Step 4: Install .NET 8.0 SDK

1. **Install the .NET 8.0 SDK** (if not already installed):
   - You can download the .NET SDK from here.
2. **Verify installation**:
   - Run the following command in a terminal or command prompt:
     dotnet --version
   - Ensure it shows .NET 8.x.x.

Step 5: Restore Dependencies

1. Open a terminal in **Visual Studio Code** (or any other terminal tool).
2. Run the following command to restore the NuGet packages:
   dotnet restore

(I had to install xUnit package separately, and was triggering errors related to it, in that case run :

dotnet add package xunit
dotnet add package xunit.runner.visualstudio)

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

By default the application is configured to run on https. Even if you click on the http link in the terminal, it will redirect to https version.

If you are using the SchoolManagementDb.bak backup file to set up the SQL database, users are already created in the database for testing the application, use the below credentials for login:

Admin Account:
Username: Admin
Password: Edusync123!

Teacher Account:
Username: Teacher
Password: Edusync123!

Student Account:
Username: Student
Password: Edusync123!

If you are using the SchoolManagementDb.sql script to set up database, then:

- There is code to seed an admin account when the program is run for the first time (Code only added for testing purposes), you can use this admin account for starting out and exploring the application.
    - Username: Admin
    - Password: Edusync123!
- To test out Teacher and Student accounts, use register option in the home page to register new accounts. By default, new accounts will be registered as student. So log in to Admin account again and then go to manage roles and change role of newly created account to teacher role to test out the teacher account.

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.