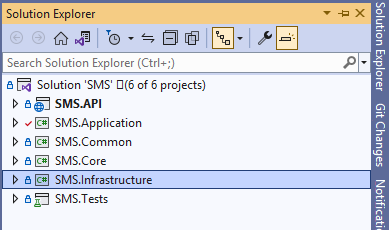
**Student Management System – User Guide**

1. Backend : Dot Net Core – Web API – Code First Approach – Clear Architecture
2. Frontend : React.js with Redux

Backend - <https://drive.google.com/file/d/1xIPk5KMkH0owxVLS7SJDmWA9jR6Vqg7U/view?usp=sharing>

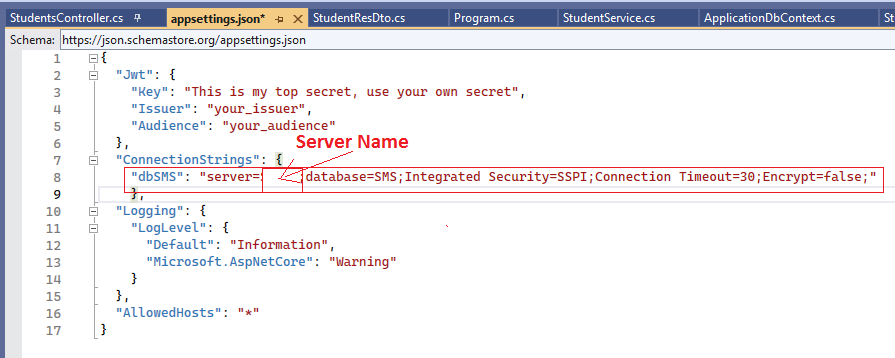
React - <https://drive.google.com/file/d/1aZTPWkjyHF-_BMO9J9OSSZxdKrG_Runi/view?usp=sharing>

**Backend : Dot Net Core – Web API – Code First Approach – Clear Architecture – Site Url**



http://localhost:5071/swagger/index.html

**Set up database details in the appsettings.json**



**Run the following command in Package Manager Console so that it will automatically create database and its tables**

PM> Add-Migration InitialCreate

Build started...

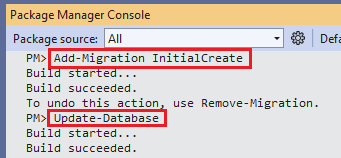
Build succeeded.

To undo this action, use Remove-Migration.

PM> Update-Database

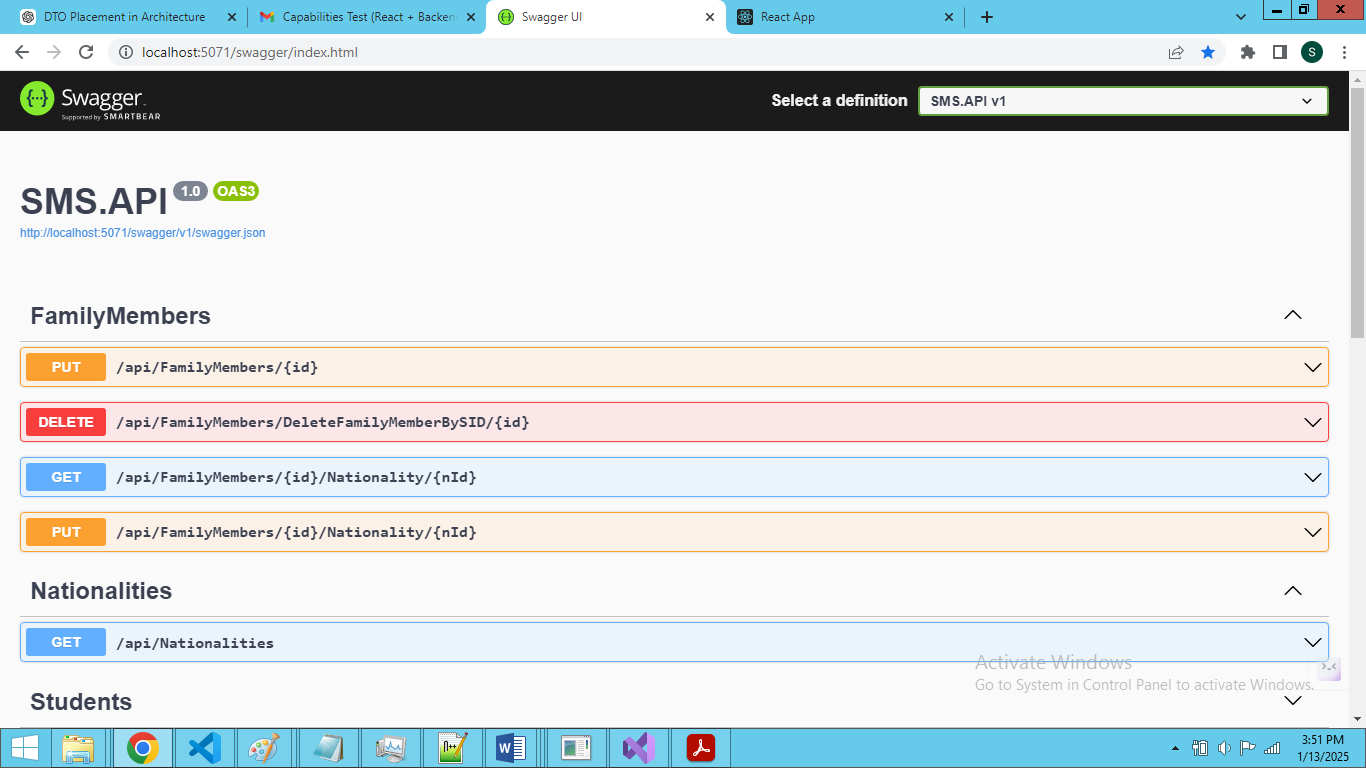
Build started...

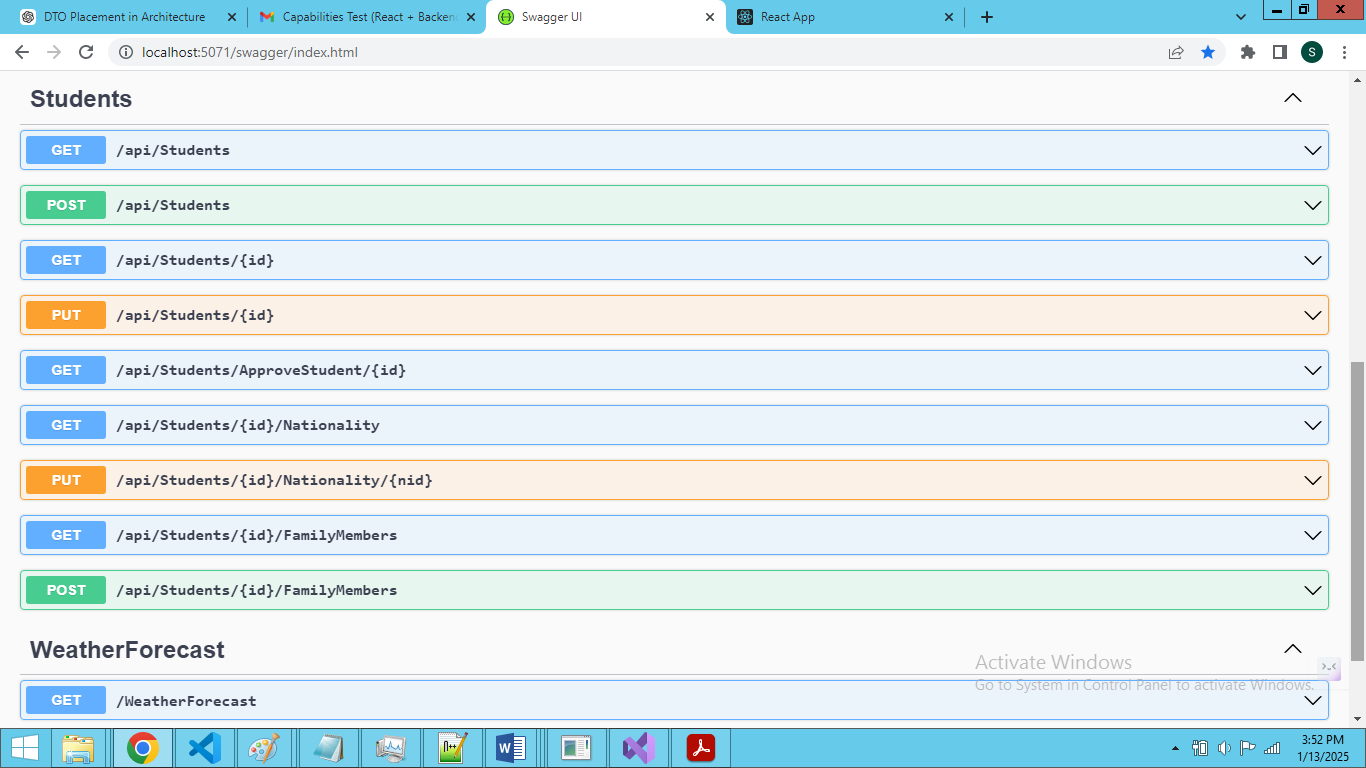
Build succeeded.



<http://localhost:5071/swagger/index.html>

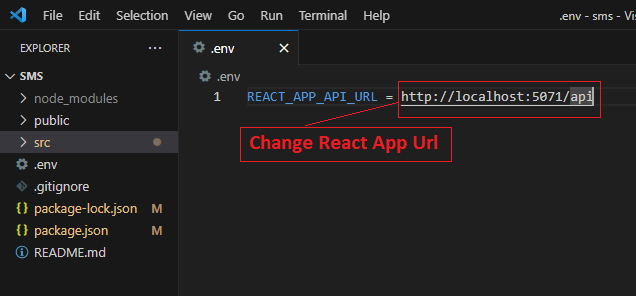
Web Api with FamilyMembers, Nationalities and Students Model



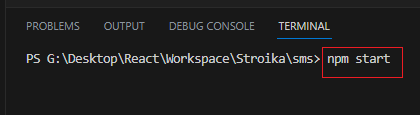


**Frontend – React.js with Redux**

**Change React Application Site Url in the .env file**

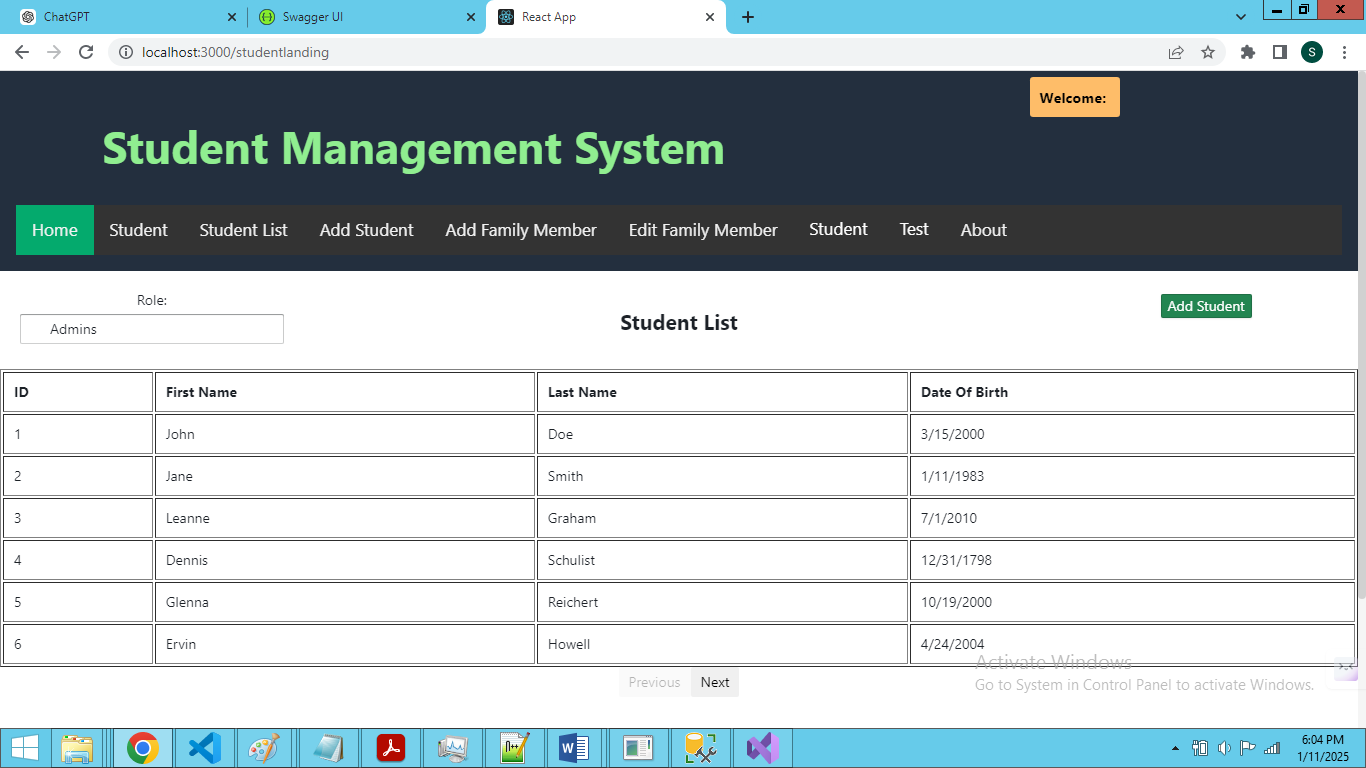
****

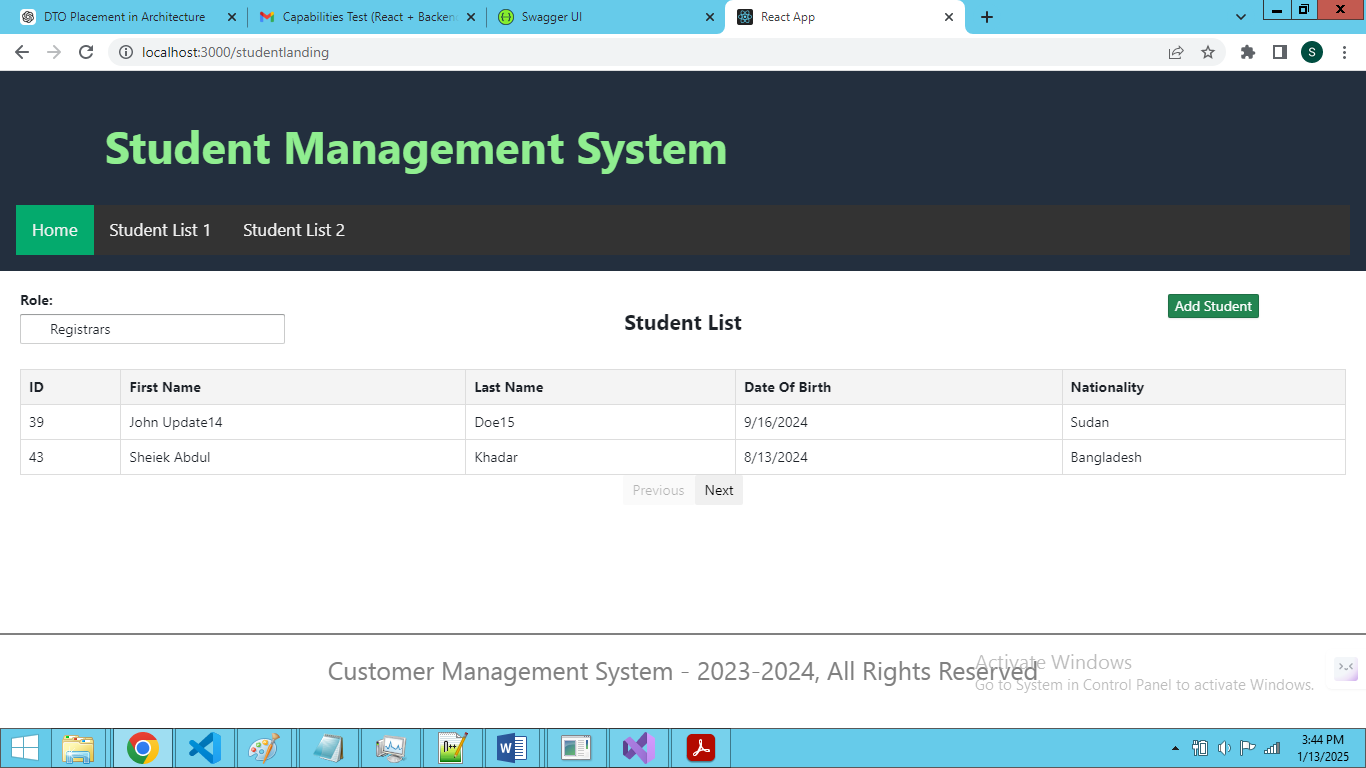
**Run ‘npm start’ command in the Terminal. It will automatically create node\_module files in the react application**

****

**Landing Page:**

* Landing page is displaying students list. Once click student item row it will open Model is Detail Page
* It has Role selection dropdown and Add Student button
* Once Admin enter the Student data. It will display in Landing page.

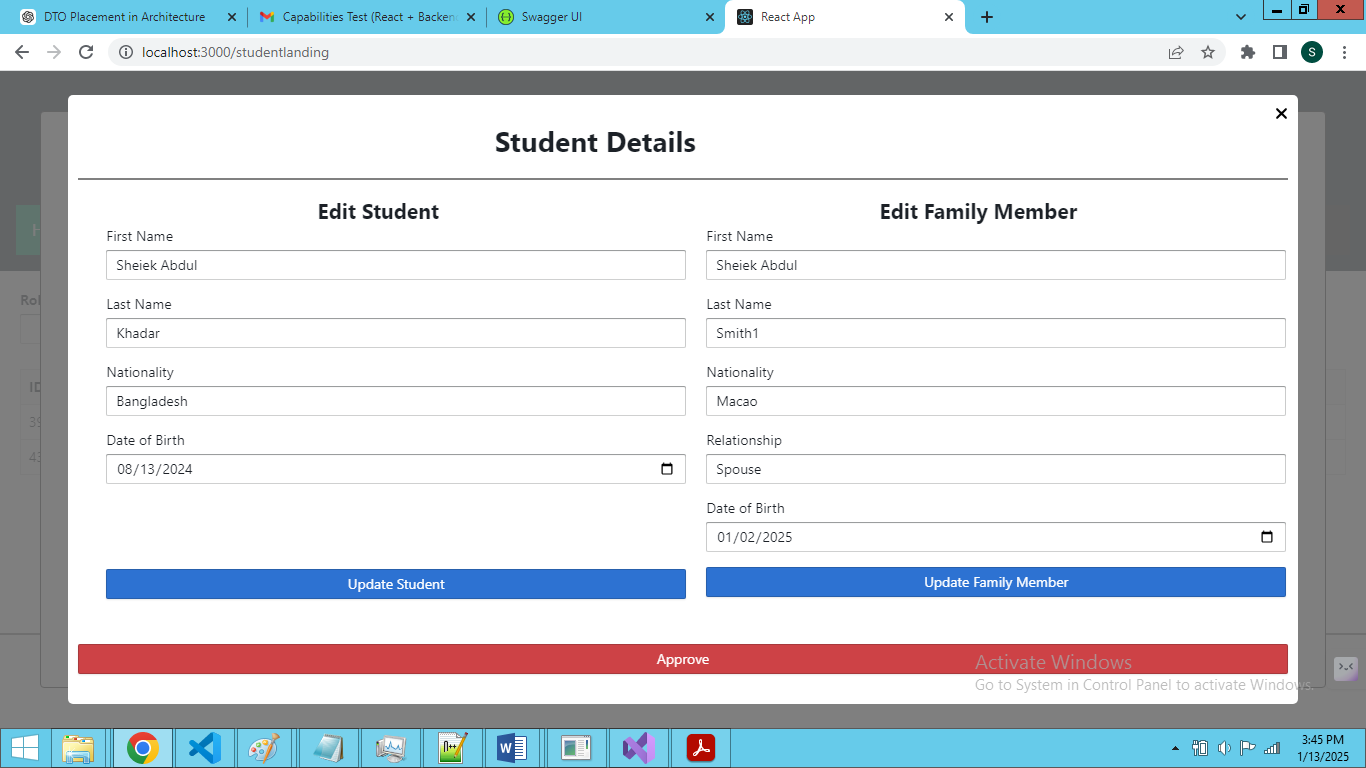




Detail Page:

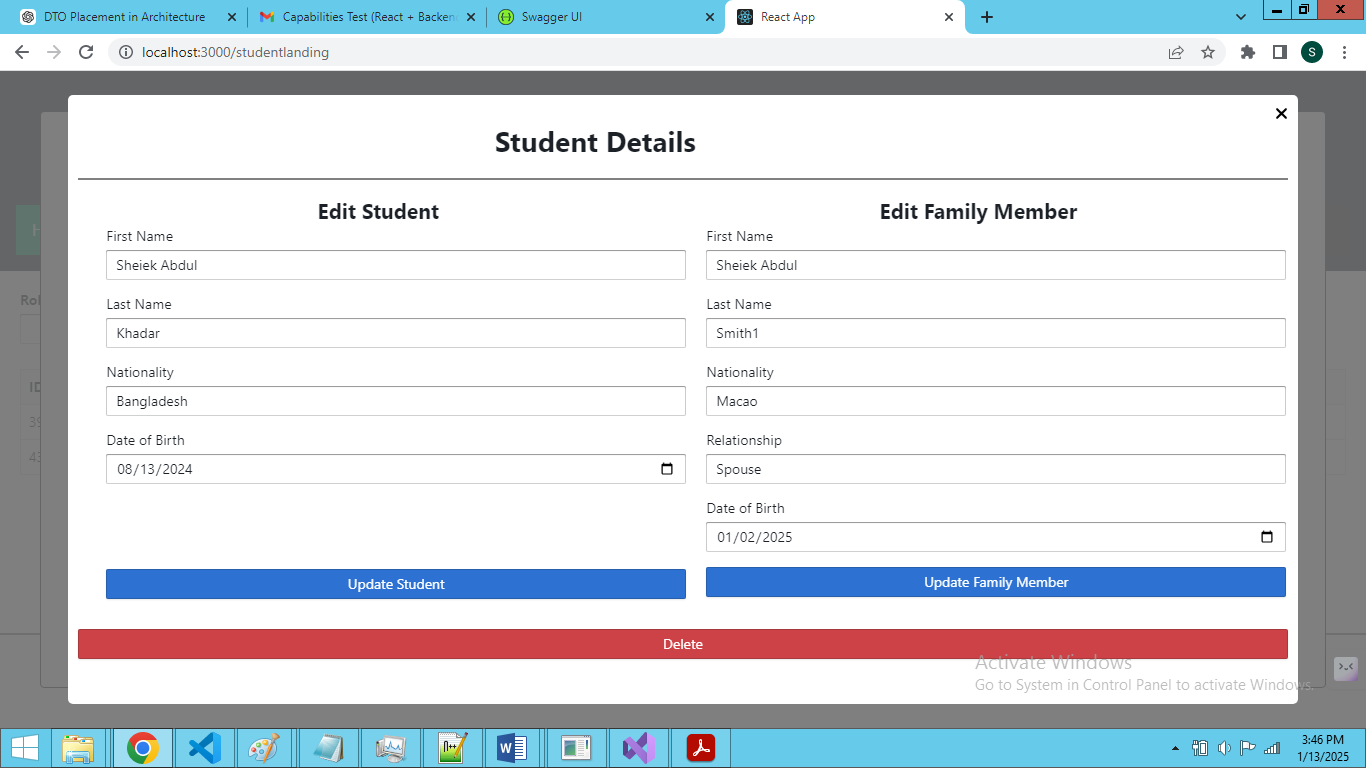
Stage 1:

* User Registrars can edit Student and Family Member datas
* They can Approve Student Details



Stage 2:

Once approve the student details. We can edit or delete the student and family member



Stage 3:

* Once done Approve or Edit or Delete. Page will redirect to Landing page
* Landing page will display updated student lists

