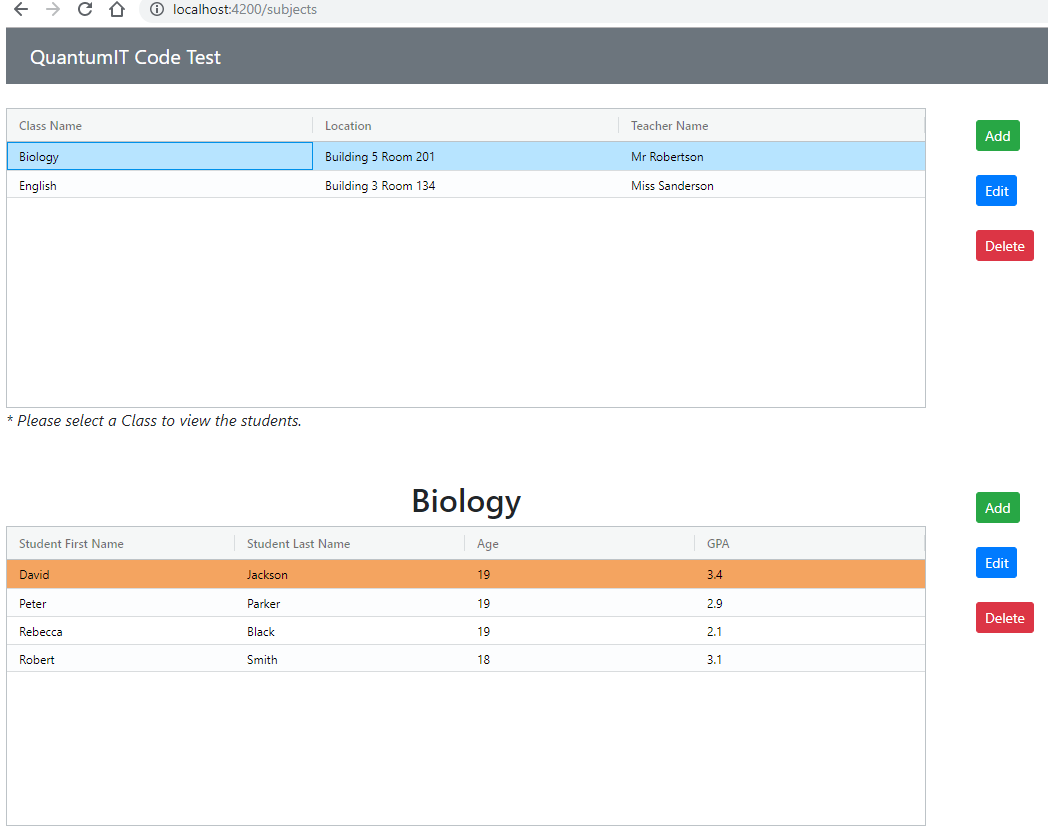
QuantumIT Code Test

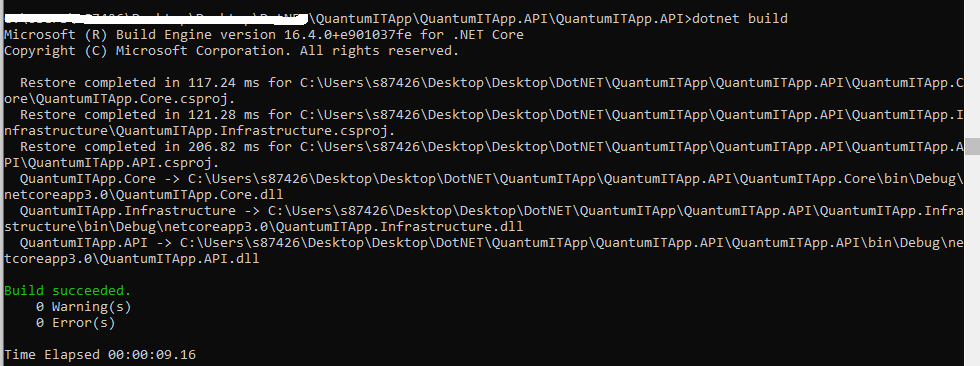


QuantumITApp.API

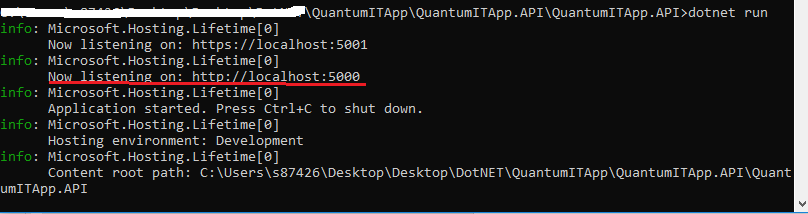
Back-end (ASP.NET Core 3.0 RESTful API)

# Build and run

1. Use command ‘dotnet build’ to build the application.



1. Use command ‘dotnet run’ to run the application.



# Tools used

1. Visual Studio 2019
2. .NET Core 3.0 WebAPI
3. ORM – Entity Framework Core 3.0
4. Persistence – Data is persisted to quantumitapp.db using Microsoft.Data.Sqlite 3.0 library
5. AutoMapper
6. Built-in Dependency Injection is used.

# End-Points

Class

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **URL** | **Payload** | **Description** |
| GET | <http://localhost:5000/api/subjects> |  | Displays all the classes |
| POST | <http://localhost:5000/api/subjects> | {  "name": "Biology",  "location": "Building 5 Room 201",  "teacherName": "Mr Robertson"  } | Add a new class ‘Biology’ |
| PUT | <http://localhost:5000/api/subjects/17> | {  "name": "Anatomy",  "location": "Building 5 Room 201",  "teacherName": "Mr Robertson"  } | Update class details |
| DELETE | <http://localhost:5000/api/subjects/17> |  | Delete a class |

Students

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **URL** | **Payload** | **Description** |
| GET | <http://localhost:5000/api/students> |  | Displays all the students |
| GET | <http://localhost:5000/api/subjects/13/students> |  | Displays only the students associated with a particular class. |
| POST | <http://localhost:5000/api/students> | {  "subjectId": 13,  "firstName": "David",  "surName": "Jackson",  "age": 19,  "gpa": 3.4  } | Add a student to a particular class. |
| PUT | <http://localhost:5000/api/students/31> | {  "subjectId": 13,  "firstName": "David",  "surName": "Bunting",  "age": 19,  "gpa": 3.4  } | Update the details of a particular student (in this case, the student with id 31) who belongs to a particular Class (in this case, the class with id 13). |
| DELETE | <http://localhost:5000/api/students/31> |  | Delete a student. |

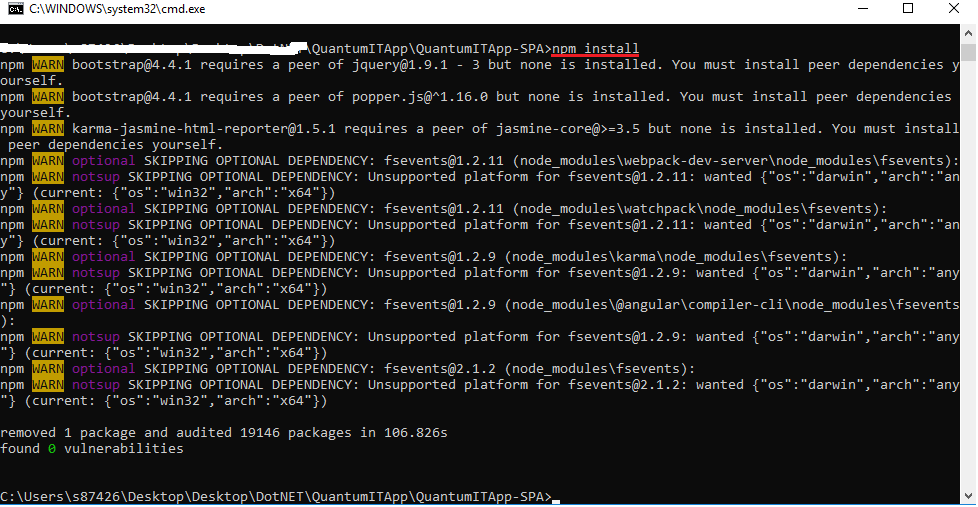
QuantumITApp-SPA

Front-end (Angular 8)

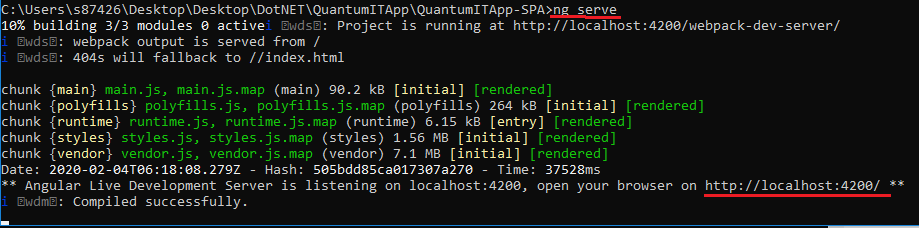
URL - <http://localhost:4200/subjects>

# Build and run

1. Use command ‘npm install’ to install and resolve the dependencies.

****

1. Use command ‘ng serve’ to run the angular application.



# Tools used

1. Angular CLI 8 or above
2. Ag-Grid-Angular for the grid
3. AlertifyJS for notifications

Improvements that could have been made

The application can be improved to add an existing student to the class. As of now, the system only allows to create a brand new user.

Also, the delete functionality can be improved to remove the user only from a particular class and not from the database itself.

Logging and error handling could have been better.