Dharmsinh Desai University, Nadiad
Faculty of Technology

Computer Engineering Department
B.tech – Semester - VI

Subject: Web Services Development

Project Title: Student Admin Portal

Guided by: Prof. Prashant M. Jadav

Nai Om Rajeshkumar (CE072) [20CEUBS141]

Dhramsinh Desai University, Nadiad
Department of Computer Engineering
Faculty of Technology

CERTIFICATE

This is to certify that the project work carried out
in the subject of
Web Services Development
is the bonafide work of

Om Nai (CE072) (20CEUBS141)

of B.Tech semester **VI** Computer Engineering during the academic year **2022 - 2023**.

Prof. Prashant M. Jadav
Associate Professor,
Computer Engg. Department,
Faculty of Technology
Dharmsinh Desai University, Nadiad

Dr. C. K. Bhensdadia,
Professor and HoD,
Computer Engg. Department,
Faculty of Technology
Dharmsinh Desai University, Nadiad

Contents

Abstract

- Student admin portal is web application which provides the seamless services to manage the student from the admin side.
- For any school, college, education sites, university etc. managing the student is the toughest challenge as if various paradigms related to any person (here : student) like name, course enrolled, date of birth etc. are associated with that person (here : student).
- This portal serves for the personal and the professional services to manage the student.
- Also, its minimalistic UI provides the admin legitimate good experience and serves the purpose.
- It takes a great leap, as if it developed to serve on any platform and on any browser.

Introduction

About Project :-

Provided student admin portal provides the efficient, flawless and robust student managing functionalities. With its gentle UI it is also user friendly.

Basic set of functionalities related to add new student, delete the particular student, edit the selected student, sorting the student (as per names), update to the existing record the selected student is provided.

Technology Used :-

- API Creation and Backend:
    - C# programming language
    - .Net Core (version : 5.0)
    - .Net runtime (version : 5.0)
    - RESTful API with MVC design pattern
    - Repository Method
    - Domain Model
    - Data Transfer Object (DTO)

- Database and Server Service:
    - Microsoft SQL Server
    - Microsoft entity framework (Core)
    - Microsoft entity framework (SQL Server)
    - Auto Mapper

- Frontend UI:
    - Google Angular
    - Angular Material UI
    - Typescript
    - Fetch API services
- API testing:
    - Swagger
    - Postman
- Version Control:
    - Git
    - GitHub
- Development Environment (Tools Used):
    - Microsoft Visual Studio Community Edition 2019 (For backend and web API development)
    - Microsoft Visual Studio Code (For frontend development)
    - Microsoft SQL Server Management Studio (For UI frontend of database)
    - Microsoft SQL Server Management Configurations

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

Student Admin Portal

Functional Requirements

R 1: Admin Side

Description: The main interactive and minimalistic landing page of the web app which displays the all the students enrolled in various courses. Also, provides the basic set of function to add, update, delete, sort the students.

R 1.1: Add Student

Input: Add the details of the new student.
Process: Verifying the data.
Output: Information will be successfully stored in the database provide it pass the required validators.

R 1.2: View Student

To see the students enrolled.
Output: On user click of home button.

R 1.3: Update Student

Input: Select the student for whom details to be updated.
Output: The updated details will be displayed on successful change of details

R 1.4: Delete Student

Input: On user(admin) click of delete button.
Output: The selected student will be deleted from the system.

R 1.5: Set Pagination

Various pagination options are available.
Input: User selection.
Output: Display the data as per selected pagination format.

R 1.6: Sort Data

Input: User click.

Output: Data will be sorted based on the names of the student.

Non-Functional Requirements:

1. Reliability

The ability of the system to behave consistently in a user-acceptable manner when operating within the environment for which the system was intended.

2. Maintainability

Easy to maintain as if minimalistic UI.

3. Security

Secure access of confidential data (User Information).

4. User friendly

System should be easily used by the user with minimalistic UI.

5. Performance

Performance should be fast.

6. Efficient

System should be efficient for the managing notes.

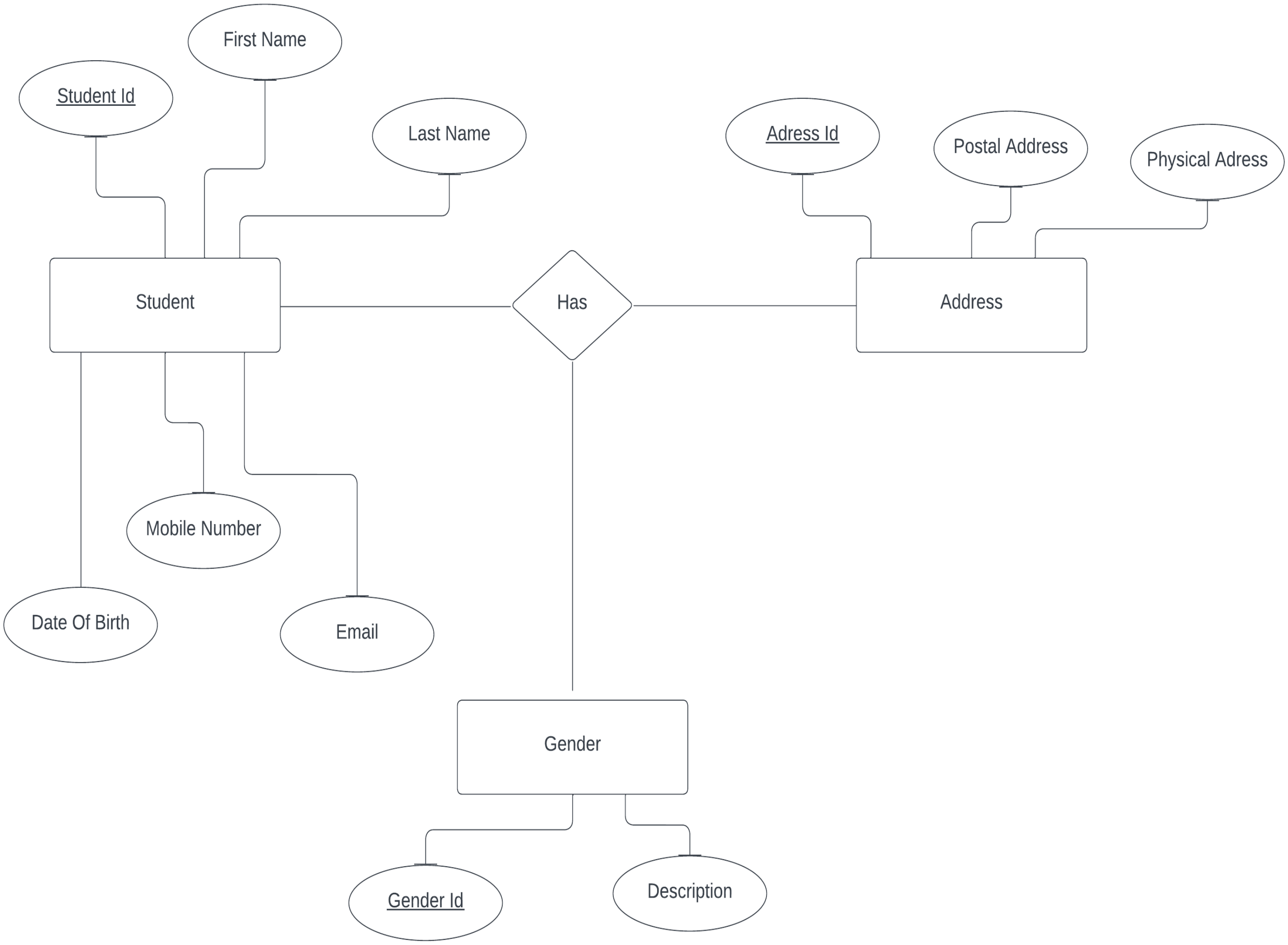
7. Privacy

Personal data of the system should not disclose to anyone.

End Of SRS

DESIGN

ER Diagram



(Fig -1)

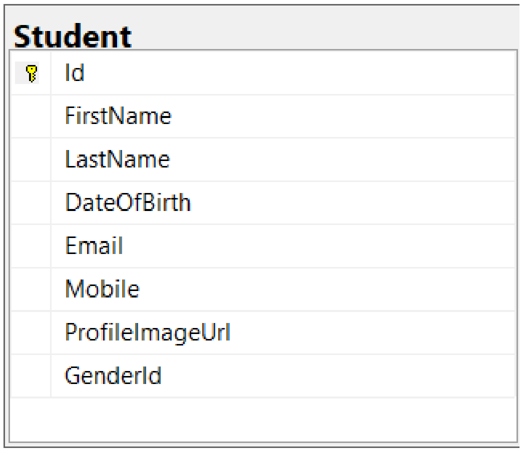
Database Design

Schema Design :-

- **Student Schema :**

Name	Type
Student ID	Guid (Primary Key)
First Name	String
Last Name	String
Date Of Birth	String
Email	Date and Time
Mobile	String
Gender ID	Guid
Gender	Gender (Navigation Property)
Address	Address (Navigation Property)

Screenshot :



Student

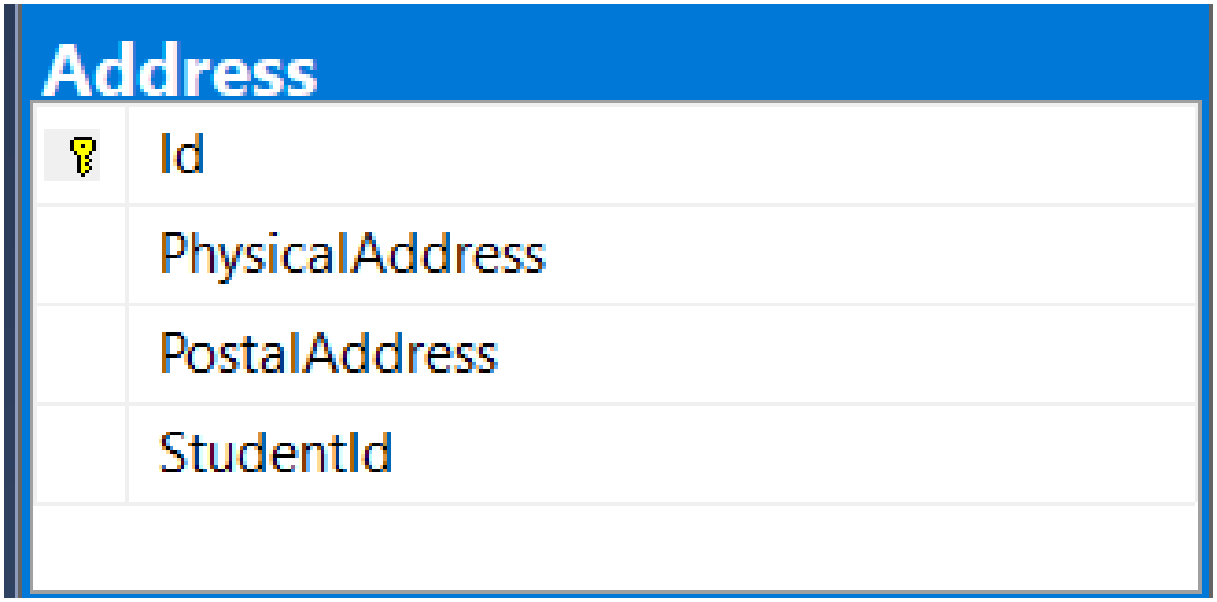
Id
FirstName
LastName
DateOfBirth
Email
Mobile
ProfileImageUrl
GenderId

(Fig -2)

- **Address Schema :**

Name	Type
Id	Guid (Primary Key)
Physical Address	String
Postal Address	String
Student ID	Guid (Navigation Property)

Screenshot :



Address

Id

PhysicalAddress

PostalAddress

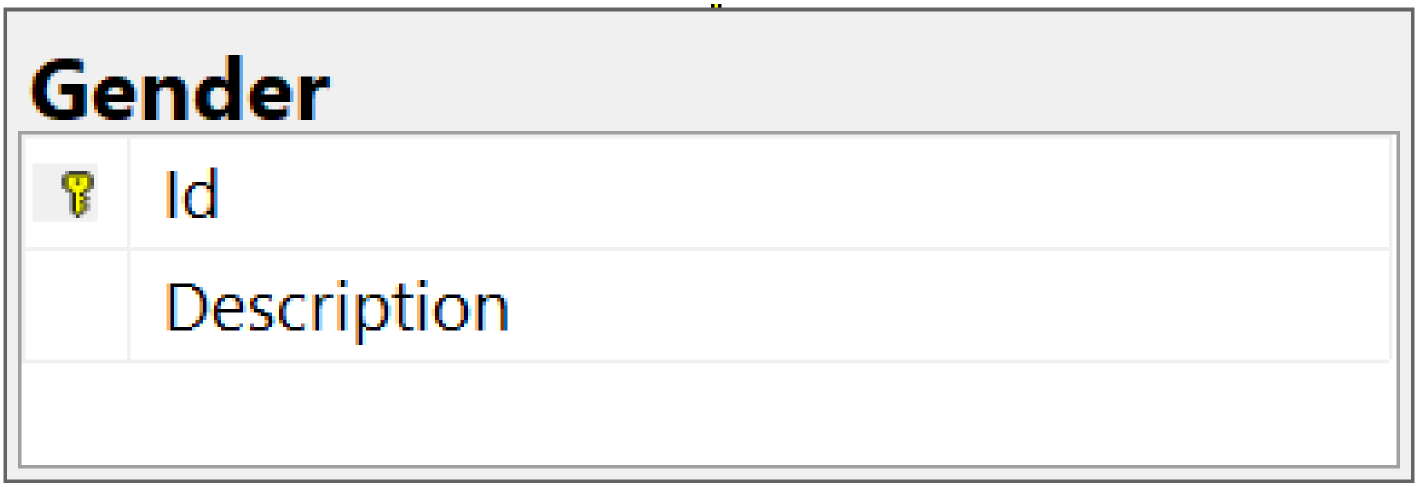
StudentId

(Fig -3)

- **Gender Schema :**

Name	Type
Id	Guid (Primary Key)
Description	String

Screenshot :



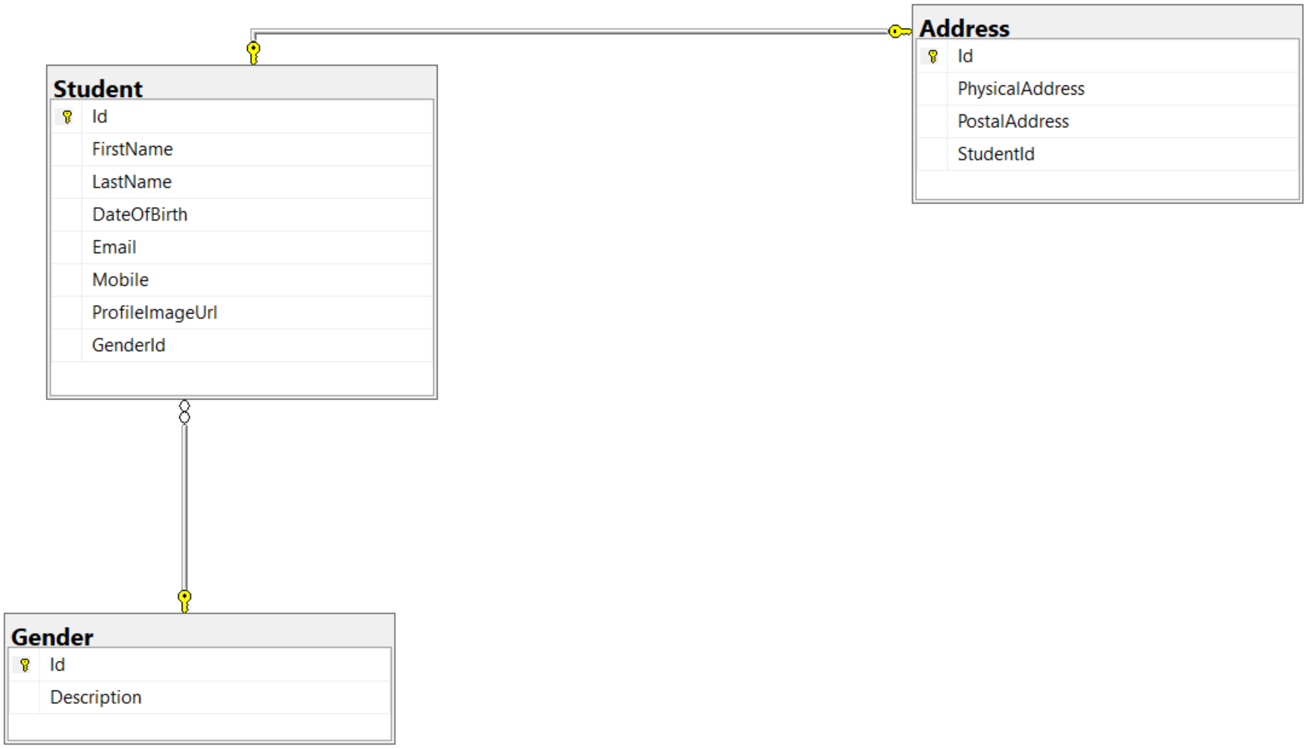
Gender

Id

Description

(Fig -4)

- **Generated Schema Diagram (SQL Server)**



Address

- Id
- PhysicalAddress
- PostalAddress
- StudentId

Student

- Id
- FirstName
- LastName
- DateOfBirth
- Email
- Mobile
- ProfileImageUrl
- GenderId

Gender

- Id
- Description

(Fig -5)

Implementation Details

Modules created and brief description of each module :-

This project consists of **5** *major set of basic functionalities*.

1) Adding a new student

- Admin can create a new student and store into database by adding the name, date of birth, address, mobile number, email, address, and selecting the gender.
- A unique ID is provided on creation of each new student. The property uised is GUID which is 128 bit (16 bytes) value.
- A success message is also shown upon creation of new student data.

2) Updating the selected student

- Admin can update the details of the selected student.
- For that click on the ✏ icon.
- Edit the details which admin like to change.
- A success message is displayed on successful update of data.

3) Deleting the selected student

- Selected the student required to be deleted from the system.
- A success message is displayed on successful deletion of student.

4) Sorting the student

- Click on the button to sort the student based on their names.
- The list is displayed in the sorted order.

5) Pagination

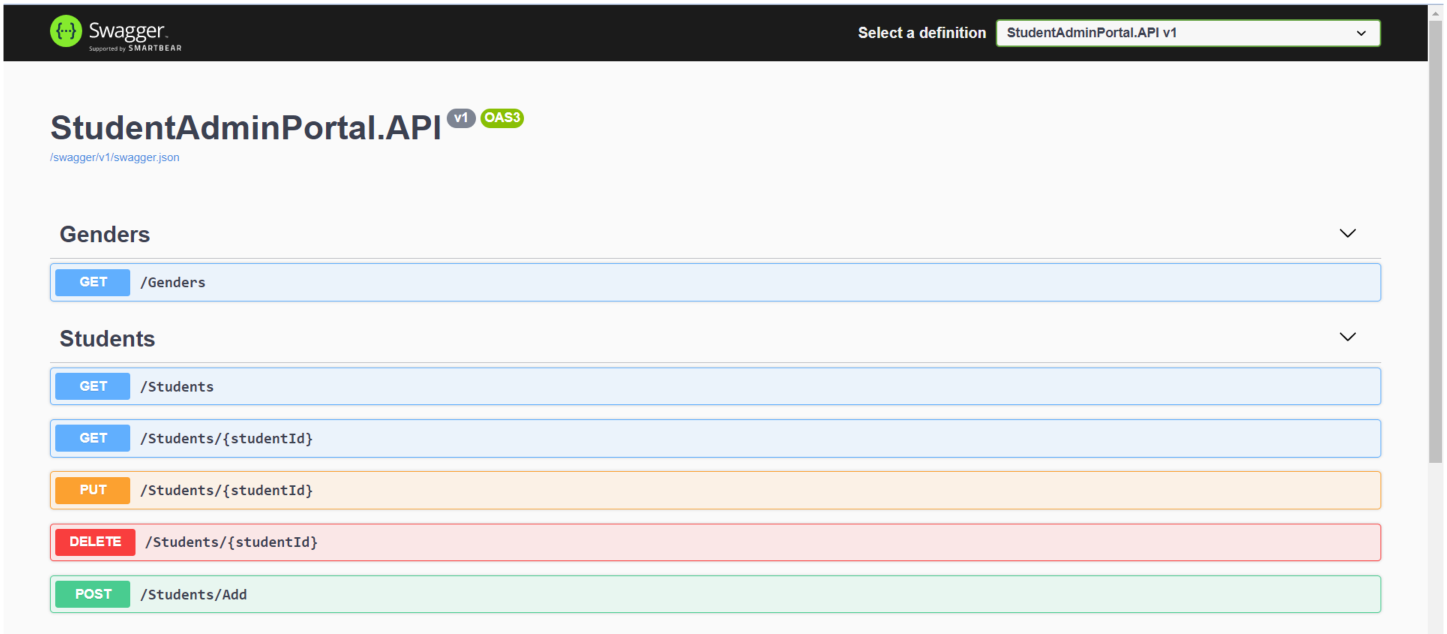
- The pagination of required student at a time can also be displayed.

API endpoints :-

***Web API** based service is used in this project*

- **API Endpoints details**
  - **Here REST (Representational State Transfer) API based method is used which provides the interactive backend support to user.**
  - **Each endpoint is targeted based on the controller calls.**
  - **Request to an API can be only done if the valid access token is there.**
  - **Message passing is done in the JSON format.**

Swagger Screenshot :-



StudentAdminPortal.API v1

StudentAdminPortal.API v1 OAS3

/swagger/v1/swagger.json

Genders

GET /Genders

Students

GET /Students

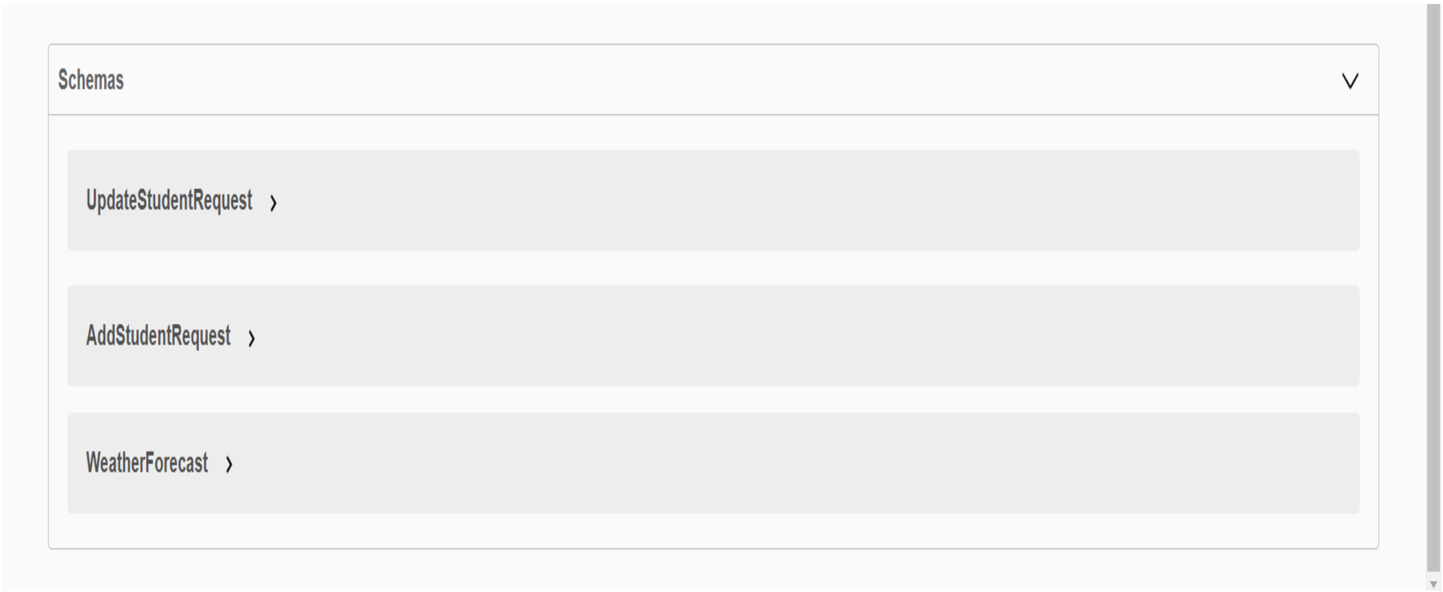
GET /Students/{studentId}

PUT /Students/{studentId}

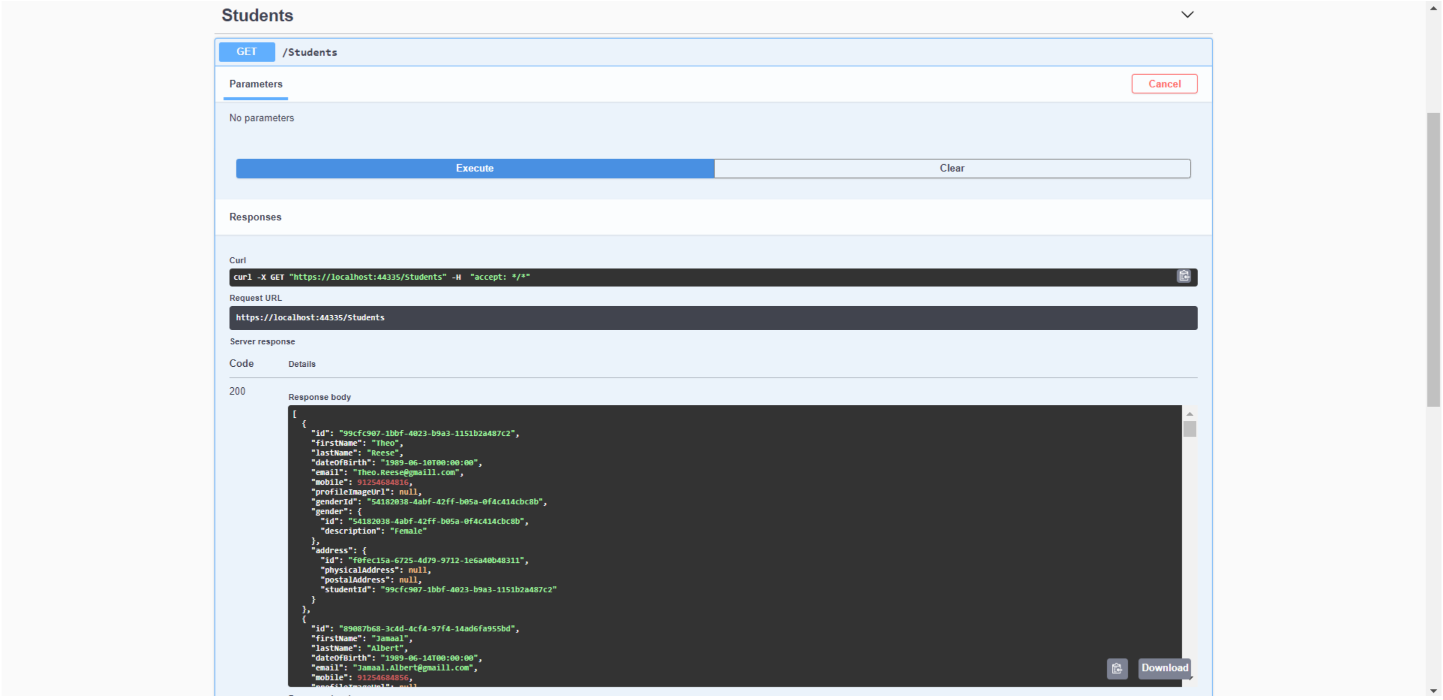
DELETE /Students/{studentId}

POST /Students/Add

(Fig -6)



(Fig -7)



```
curl -X GET "https://localhost:44335/Students" -H  "accept: */*"
```

```
https://localhost:44335/Students
```

200

```
[
  {
    "id": "99cfc907-1bbf-4023-b9a3-1151b2a487c2",
    "firstName": "Theo",
    "lastName": "Reese",
    "dateOfBirth": "1989-06-10T00:00:00",
    "email": "Theo.Reese@gmaill.com",
    "mobile": 91254684816,
    "profileImageUrl": null,
    "genderId": "54182038-4abf-42ff-b05a-0f4c414cbc8b",
    "gender": {
      "id": "54182038-4abf-42ff-b05a-0f4c414cbc8b",
      "description": "Female"
    },
    "address": {
      "id": "f0fec15a-6725-4d79-9712-1e6a40b48311",
      "physicalAddress": null,
      "postalAddress": null,
      "studentId": "99cfc907-1bbf-4023-b9a3-1151b2a487c2"
    }
  },
  {
    "id": "89087b68-3c4d-4cf4-97f4-14ad6fa955bd",
    "firstName": "Jamaal",
    "lastName": "Albert",
    "dateOfBirth": "1989-06-14T00:00:00",
    "email": "Jamaal.Albert@gmaill.com",
    "mobile": 91254684856,
    "profileImageUrl": null
```

(Fig -8)

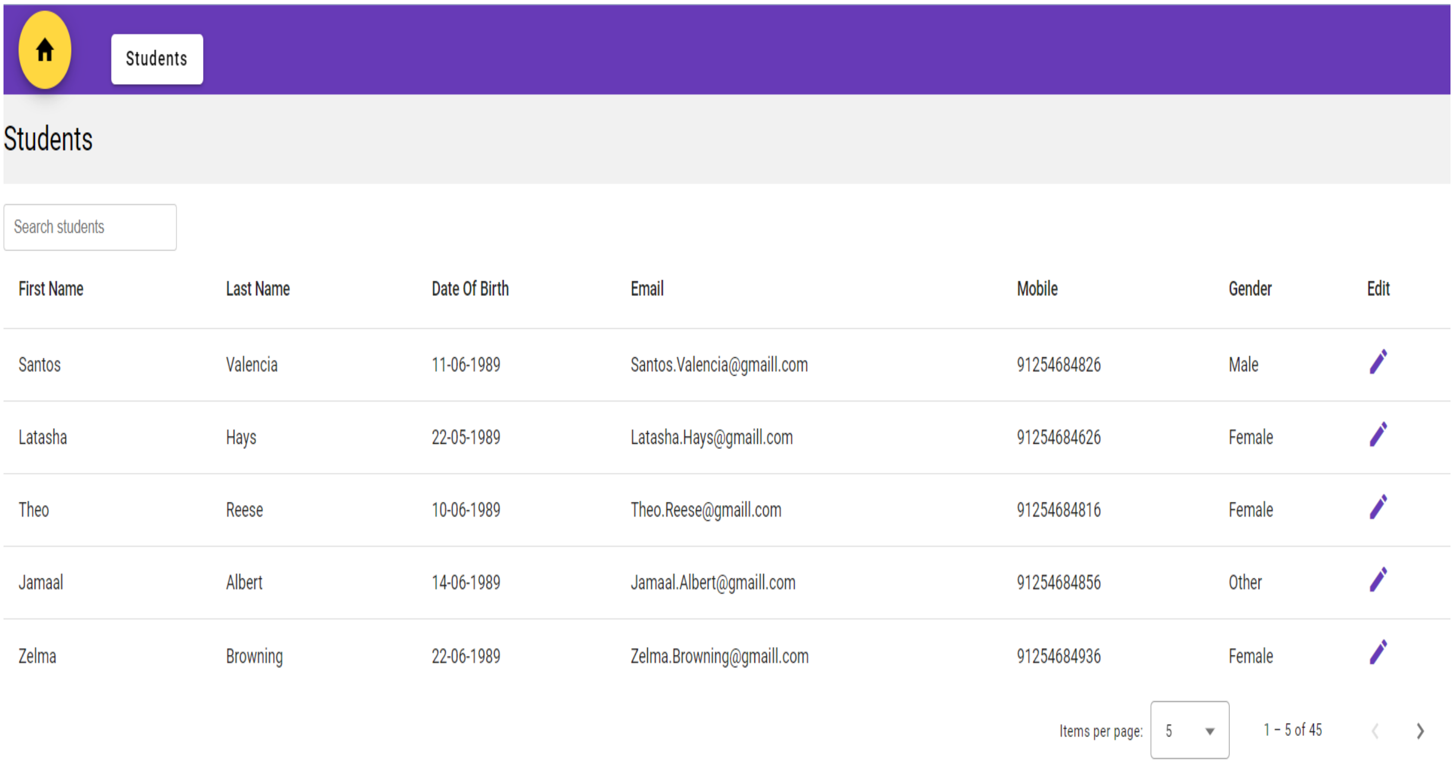
Testing

Test Case Id	Test Case Objective	Input Data	Expected Output	Actual Output	Status
TC_01	Add new student.	Add the required details.	New student added to system and success message.	Validation are successfully passed and new student is created.	**Pass**
TC_02	Update student.	Select the student to be updated.	Changes needs to be reflected successfully.	Success message is displayed and changes are stored.	**Pass**
TC_03	Deletion of the student.	Admin selection.	The selected student needs to be successfully deleted.	The selected student is deleted successfully.	**Pass**
TC_04	View student.	Admin click and default landing page.	Retrieve the records of the student in the system.	Data retrieval.	**Pass**
TC_05	Sorting student.	Admin selection.	The sorted list of students.	Student names are displayed in the sorted list format.	**Pass**
TC_06	Pagination.	Admin selection of number of records to be displayed.	The student record in selected format.	The student record in selected format.	**Pass**
TC_07	API testing	Unit Testing	NONE	Required endpoints are triggered.	**Pass**

(Table-1)

Screenshots

❖ Launch Screen



Students

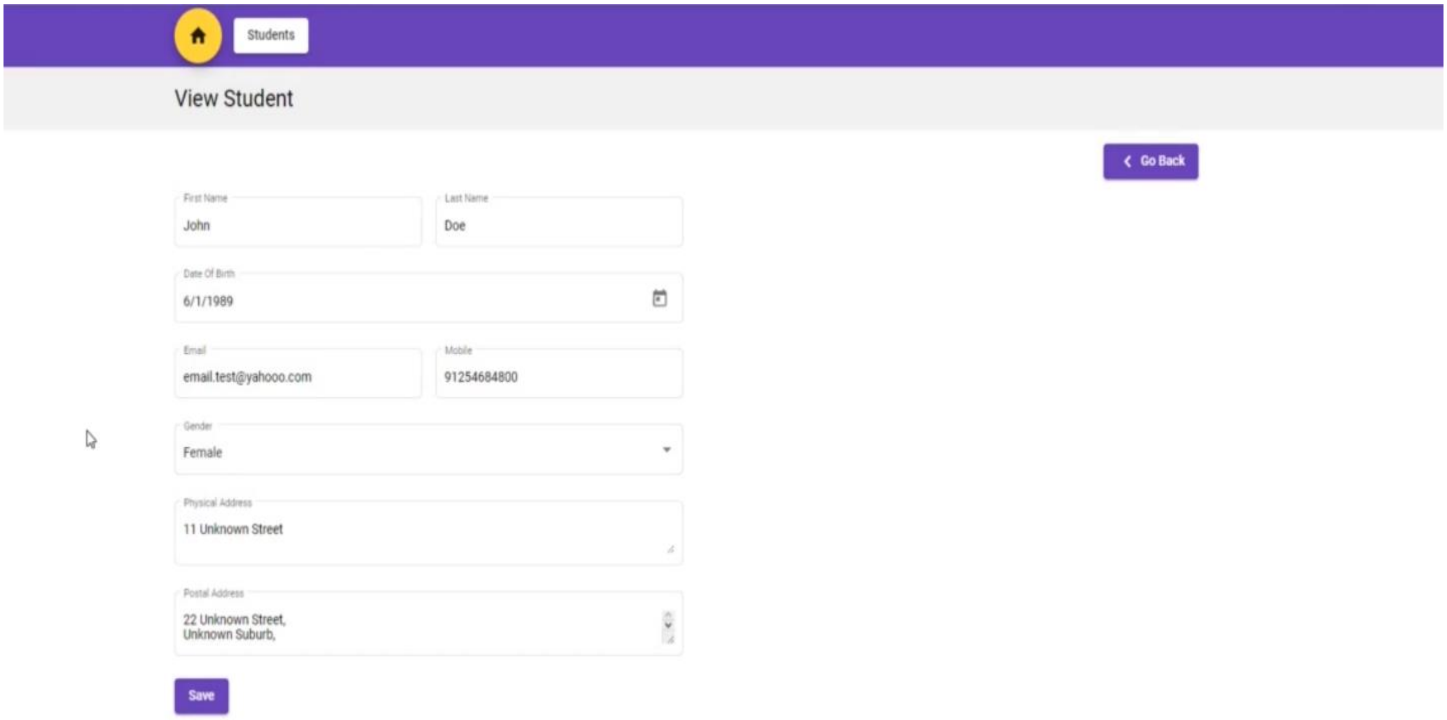
First Name	Last Name	Date Of Birth	Email	Mobile	Gender	Edit
Santos	Valencia	11-06-1989	Santos.Valencia@gmaill.com	91254684826	Male	✎
Latasha	Hays	22-05-1989	Latasha.Hays@gmaill.com	91254684626	Female	✎
Theo	Reese	10-06-1989	Theo.Reese@gmaill.com	91254684816	Female	✎
Jamaal	Albert	14-06-1989	Jamaal.Albert@gmaill.com	91254684856	Other	✎
Zelma	Browning	22-06-1989	Zelma.Browning@gmaill.com	91254684936	Female	✎

Items per page: 5 ▾ 1 – 5 of 45 ‹ ›

(Fig -9)

(Main Page)

❖ Update details page



View Student

First Name: John

Last Name: Doe

Date Of Birth: 6/1/1989

Email: email.test@yahooo.com

Mobile: 91254684800

Gender: Female

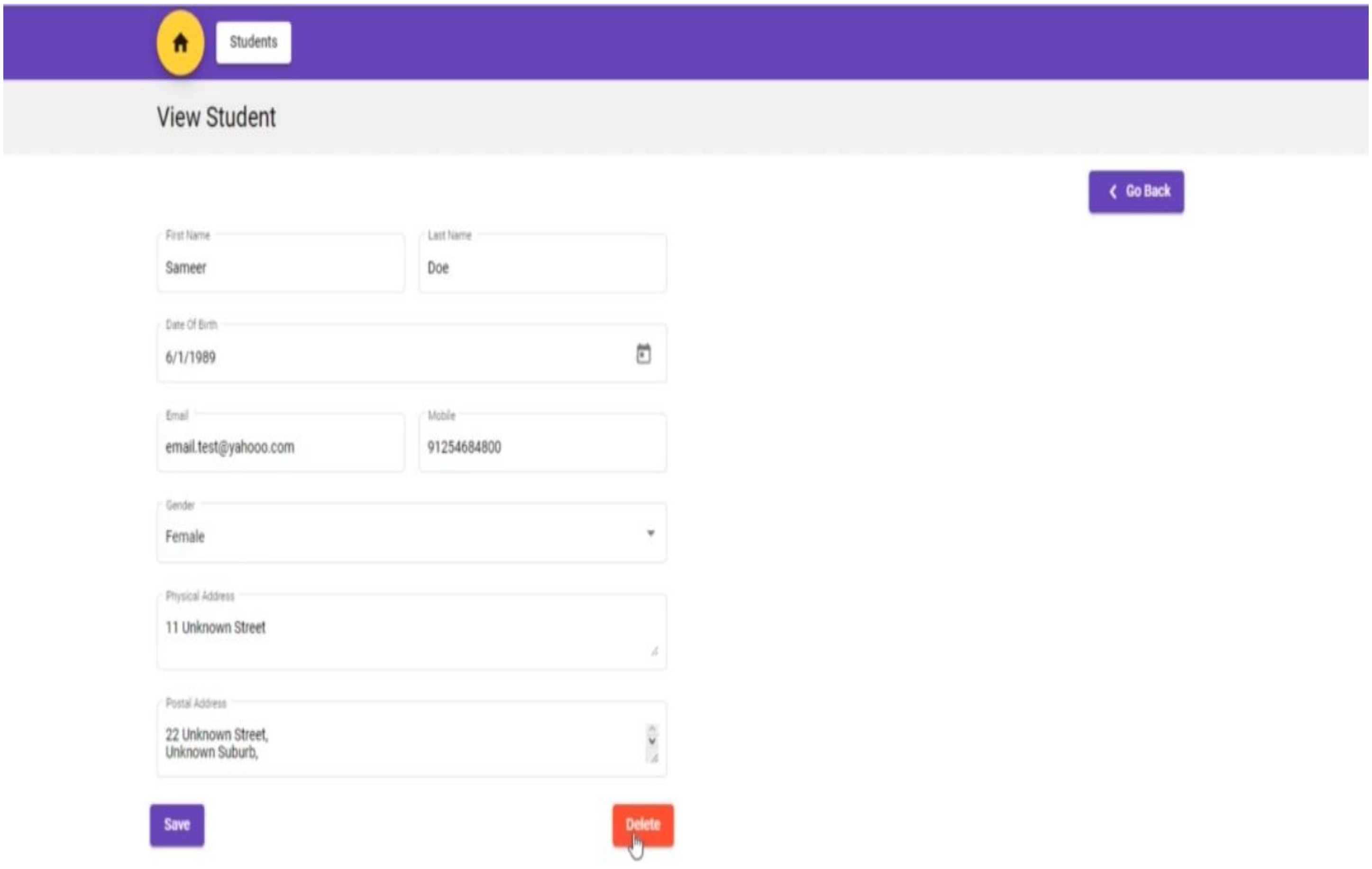
Physical Address: 11 Unknown Street

Postal Address: 22 Unknown Street,
Unknown Suburb,

(Fig -10)

(Update student details)

❖ Delete student



View Student

First Name: Sameer

Last Name: Doe

Date Of Birth: 6/1/1989

Email: email.test@yahooo.com

Mobile: 91254684800

Gender: Female

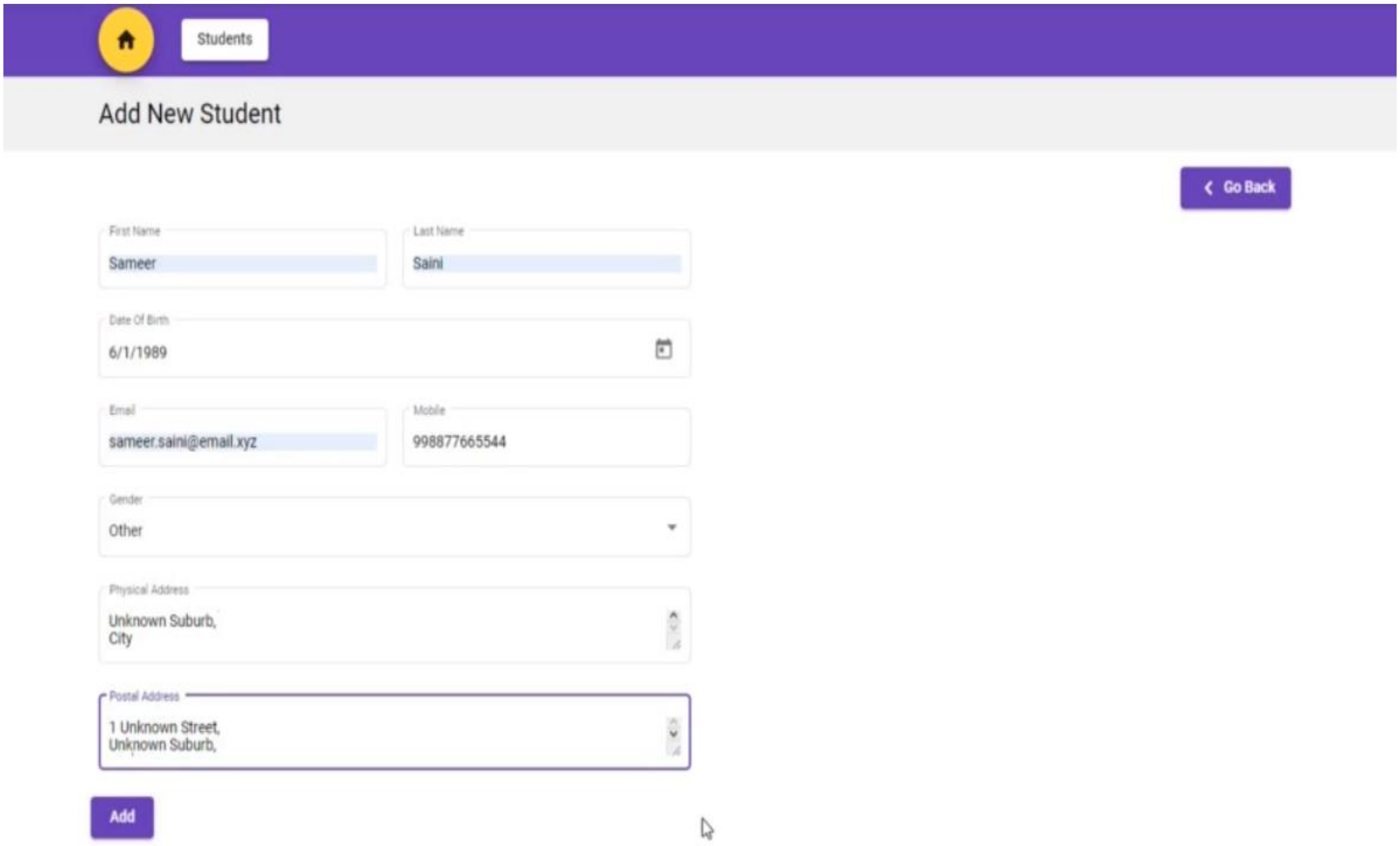
Physical Address: 11 Unknown Street

Postal Address: 22 Unknown Street,
Unknown Suburb,

(Fig -11)

(Delete the selected student record)

❖ Add student



Add New Student

First Name: Sameer

Last Name: Saini

Date Of Birth: 6/1/1989

Email: sameer.saini@email.xyz

Mobile: 998877665544

Gender: Other

Physical Address: Unknown Suburb,
City

Postal Address: 1 Unknown Street,
Unknown Suburb,

(Fig -12)

(Add student)

Conclusion

Student Admin Portal is handy tool for users for managing student used for personal and professional level. Provided with the given set of functionalities.

- Add new student
- Update given student
- Deleting the given student
- View student list
- Sort the student
- Select required pagination.
- Interactive UI
- Cloud based storage of notes

Reference / Bibliography

Following links and websites were referred during the development of this project:

- ✓ Microsoft .net framework
- ✓ Stackoverflow
- ✓ Angular JS
- ✓ Angular JS material UI

THANK YOU