Name : Mane Siddharth Dattatray

Task : Both (Frontend and Backend)

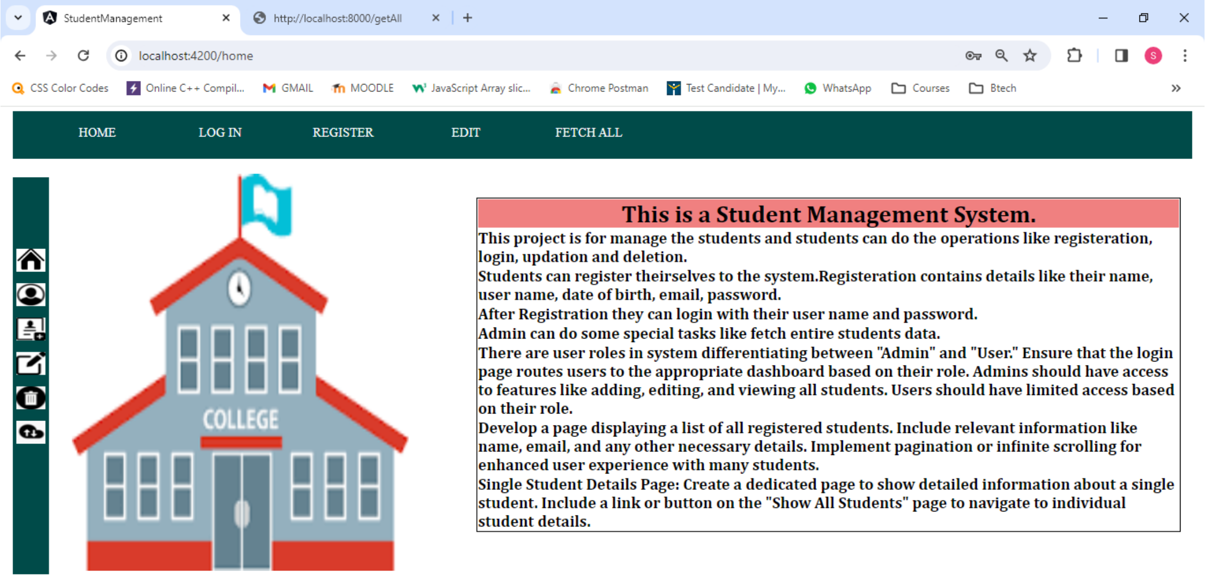
Choosed Task : First(Front End Task)

Technologies used:

Frontend: Angular, HTML, CSS, TypeScript, Bootstrap.

Backend: NodeJs, MySQL

Home Page :



This is a Student Management System.

This project is for manage the students and students can do the operations like registeration, login, updation and deletion.

Students can register theirselves to the system.Registeration contains details like their name, user name, date of birth, email, password.

After Registration they can login with their user name and password.

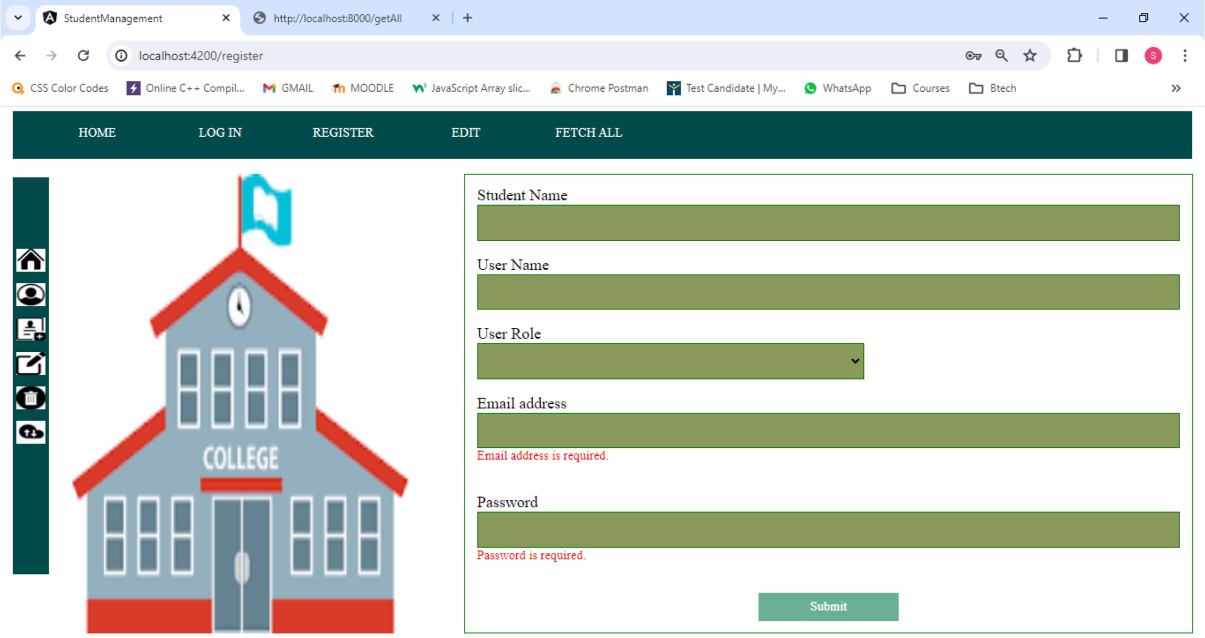
Admin can do some special tasks like fetch entire students data.

There are user roles in system differentiating between "Admin" and "User." Ensure that the login page routes users to the appropriate dashboard based on their role. Admins should have access to features like adding, editing, and viewing all students. Users should have limited access based on their role.

Develop a page displaying a list of all registered students. Include relevant information like name, email, and any other necessary details. Implement pagination or infinite scrolling for enhanced user experience with many students.

Single Student Details Page: Create a dedicated page to show detailed information about a single student. Include a link or button on the "Show All Students" page to navigate to individual student details.

Registration Page :



Student Name

User Name

User Role

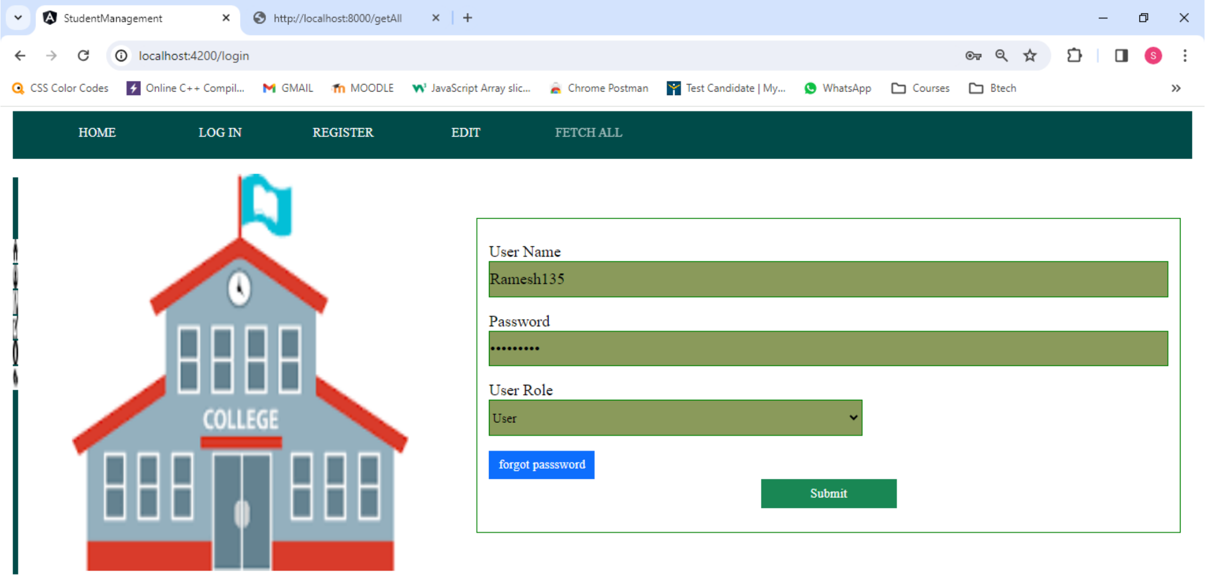
Email address

Email address is required.

Password

Password is required.

Login Page :

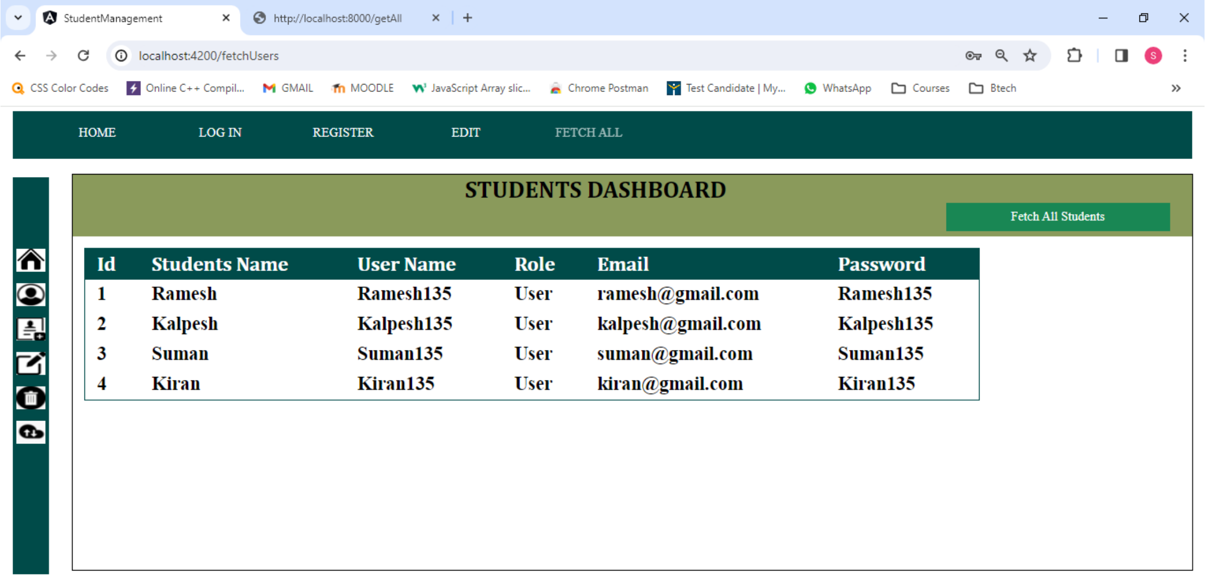


User Name: Ramesh135

Password: •••••••••

User Role: User

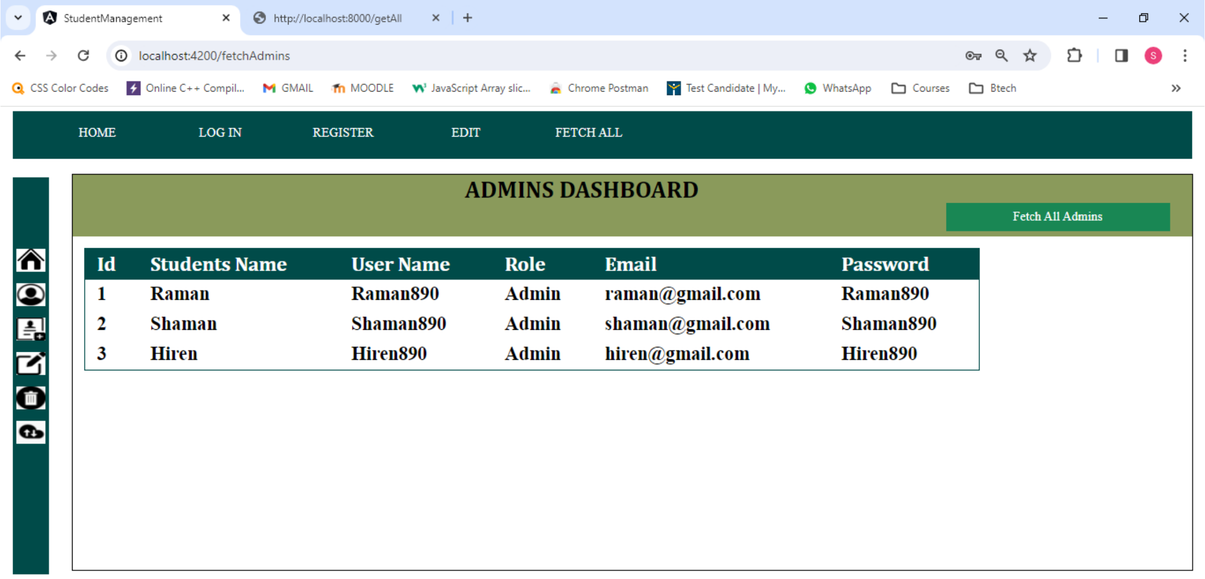
User Dashboard:



STUDENTS DASHBOARD

Id	Students Name	User Name	Role	Email	Password
1	Ramesh	Ramesh135	User	ramesh@gmail.com	Ramesh135
2	Kalpesh	Kalpesh135	User	kalpesh@gmail.com	Kalpesh135
3	Suman	Suman135	User	suman@gmail.com	Suman135
4	Kiran	Kiran135	User	kiran@gmail.com	Kiran135

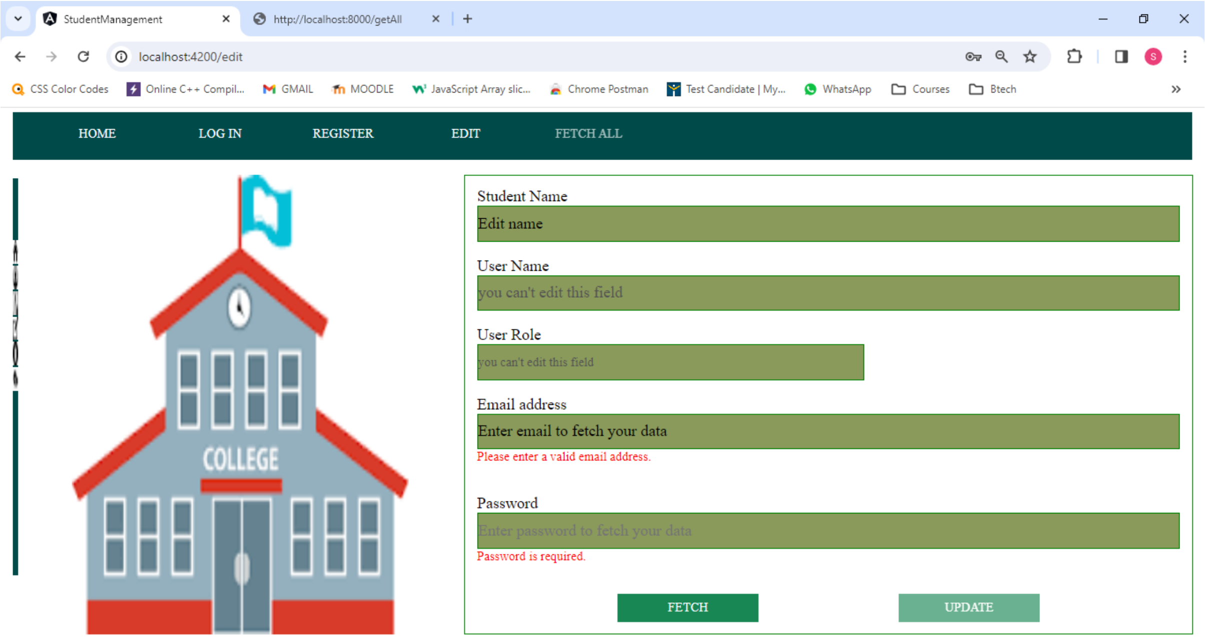
Admin Dashboard:



ADMINS DASHBOARD

Id	Students Name	User Name	Role	Email	Password
1	Raman	Raman890	Admin	raman@gmail.com	Raman890
2	Shaman	Shaman890	Admin	shaman@gmail.com	Shaman890
3	Hiren	Hiren890	Admin	hiren@gmail.com	Hiren890

Edit Page :



Student Name

Edit name

User Name

you can't edit this field

User Role

you can't edit this field

Email address

Enter email to fetch your data

Please enter a valid email address.

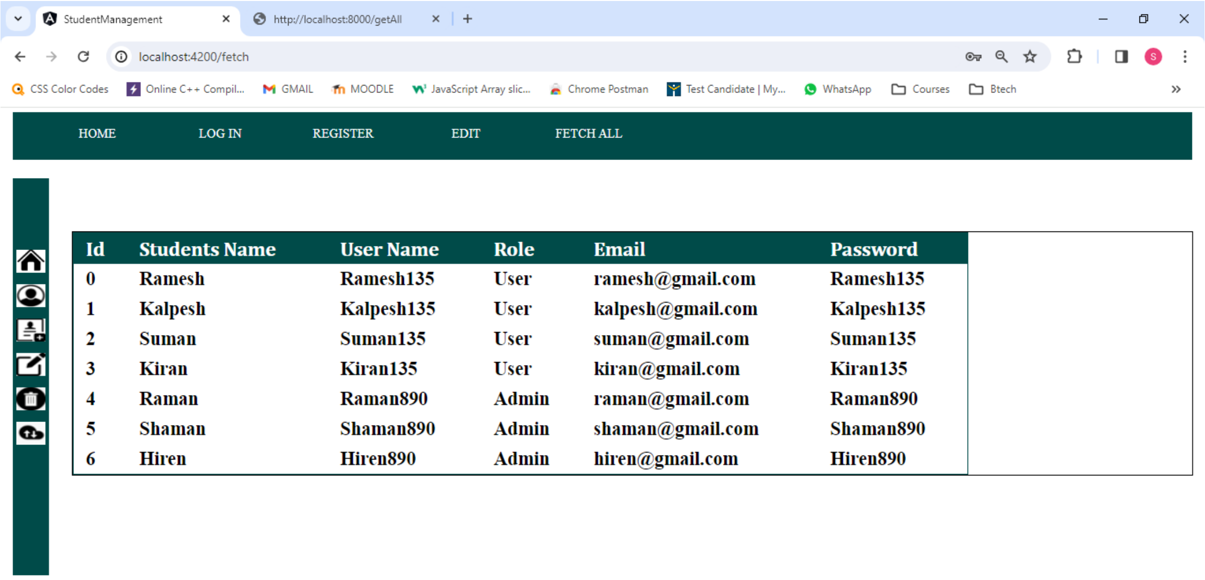
Password

Enter password to fetch your data

Password is required.

FETCH UPDATE

Fetch ALL Page :



Id	Students Name	User Name	Role	Email	Password
0	Ramesh	Ramesh135	User	ramesh@gmail.com	Ramesh135
1	Kalpesh	Kalpesh135	User	kalpesh@gmail.com	Kalpesh135
2	Suman	Suman135	User	suman@gmail.com	Suman135
3	Kiran	Kiran135	User	kiran@gmail.com	Kiran135
4	Raman	Raman890	Admin	raman@gmail.com	Raman890
5	Shaman	Shaman890	Admin	shaman@gmail.com	Shaman890
6	Hiren	Hiren890	Admin	hiren@gmail.com	Hiren890

```
});
```

Server Programming in NodeJS :

```
const express=require('express');
const mysql=require('mysql2');
const bodyparser=require('body-parser');
const cors=require('cors');

const app=express();

app.use(cors());
app.use(bodyparser.json());
app.use(bodyparser.urlencoded({extended:true}));

//configuration for connecting database
const pool=mysql.createPool({
   host:"localhost",
   port:5000,
   user:"root",
   password:"root",
   database:"goapi"
});
```

```
//fetch all users
app.get('/getUsers',(req,res)=>{
    pool.getConnection((err,connection)=>{
        if(err){
            console.log('connection error');
            res.json('{"message":"connection error"}');
        }
        sql="select * from students where role='User'";
        console.log("connected ..... ");
        connection.query(sql,(err,result)=>{
            res.json(result);
        });
        connection.end();
    });
    console.log("disconnected ..... ");
})
```

```
//fetch all admins
app.get('/getAdmins',(req,res)=>{
  pool.getConnection((err,connection)=>{
    if(err){
      console.log('connection error');
      res.json('{"message":"connection error"}');
    }
    sql="select * from students where role='Admin'";
    console.log("connected ..... ");
    connection.query(sql,(err,result)=>{
      res.json(result);
    });
    connection.end();
  });
  console.log("disconnected ..... ");
})
```

```
// fetch all users and admins
app.get('/getAll',(req,res)=>{
    pool.getConnection((err,connection)=>{
        if(err){
            console.log('connection error');
            res.json('{"message":"connection error"}');
        }
        sql="select * from students";
        console.log("connected ..... ");
        connection.query(sql,(err,result)=>{
            res.json(result);
        });
        connection.end();
    });
    console.log("disconnected ..... ");
})
```

```
// running the server
app.listen(8000,()=>{
    console.log("server is running ......");
});
```