VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science Engineering and Information Systems (SCORE)

MTech Integrated Software Engineering

SWE3099- Industrial Internship

Project tittle: - School Management System

MAY 2023 – JUNE 2023

Submitted
By

GUGGILLA GANGABALAJI-20MIS0033

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACRONYM	EXPANSION
MERN	(MongoDB, Express.js, React, Node.js)
HTML	Hyper Text markup Language
CSS	Cascading Style Sheets
JS	Java Script

CHAPTER 1

1.INTRODUCTION

A school management system project is a comprehensive software solution designed to streamline and automate various administrative and academic processes within an educational institution. This system aims to enhance the efficiency and effectiveness of school management by integrating and centralizing all key functions, from student enrolment and attendance tracking to grade management and resource allocation. It serves as a digital backbone for educational institutions, such as schools, colleges, and universities, helping them manage their day-to-day operations seamlessly.

1.1 Problem Statement

Help educational institutions manage and streamline various administrative and academic tasks. Solving the problem of managing and streamlining administrative and academic tasks in educational institutions is important because it enhances efficiency, productivity, collaboration, data-driven decision making, student satisfaction, and compliance. By addressing these challenges, educational institutions can focus more on providing highquality education to their students.

1.2 Motivation

The school management system project is a powerful tool that promises to revolutionize the way educational institutions operate. It's driven by the desire to enhance efficiency, transparency, and accessibility in managing academic processes, from student enrolment to grading and administrative tasks. This project embodies the vision of creating a seamless, digital ecosystem where teachers, students, and parents can easily access and exchange information, ultimately improving the overall quality of education. The motivation behind this project lies in the belief that by implementing an innovative and user-friendly school

management system, we can empower educators and students, streamline administrative tasks, and foster a more engaging and productive learning environment for all stakeholders.

1.3 OBJECTIVE

Creating a School Management System using the MERN (MongoDB, Express.js, React, Node.js) stack for the Ethnus organization can help streamline various administrative and academic processes. Here are the objectives and key features that such a system could aim to achieve:

1. **Student Information Management**:
    - Maintain a centralized database of student information.
    - Manage student profiles, including personal details, contact information, and academic records.
    - Track attendance and generate reports.
2. **Academic Records Management**:
    - Store and manage academic records for each student.
    - Generate report cards, transcripts, and other academic documents.
    - Calculate and display grade points and GPA.
3. **Timetable and Scheduling**:
    - Create and manage class schedules and timetables.
    - Allow for dynamic changes and adjustments.
    - Ensure efficient utilization of resources and classrooms.
4. **Staff Management**:
    - Maintain staff profiles, including teachers and administrative personnel.
    - Manage staff attendance, leaves, and payroll.
    - Assign and track responsibilities and tasks for the staff.
5. **Resource Management**:
    - Manage school resources such as classrooms, laboratories, and equipment.
    - Schedule and track the availability of resources for classes and events.
    - Automate resource booking and allocation.
6. **Fee Management**:
    - Manage student fees, including tuition, books, and other charges.
    - Generate fee invoices and receipts.
    - Keep track of fee payments and send reminders to parents and guardians.
7. **Communication and Notifications**:
    - Facilitate communication between teachers, students, and parents.
    - Send notifications about important events, assignments, and meetings.
    - Provide a secure and efficient messaging system.
8. **Attendance Tracking**:
    - Record and manage student and staff attendance.

- Automatically generate attendance reports and notify parents of absent students.

9. **Report Generation and Analytics**:
    - Generate various reports and analytics for administrators, teachers, and parents.
    - Provide insights into student performance, resource utilization, and overall school operations.
10. **Security and Access Control**:
    - Ensure data security and access control to protect sensitive student and staff information.
    - Implement user roles and permissions to control access to system features.
11. **User-Friendly Interface**:
    - Develop an intuitive and user-friendly interface for easy navigation.
    - Ensure the system is responsive and accessible from various devices.
12. **Scalability and Maintenance**:
    - Design the system to be scalable to accommodate future growth.
    - Regularly maintain and update the system to ensure it meets evolving needs.
13. **User Training and Support**:
    - Provide training and support for administrators, teachers, and other users to ensure effective system utilization.

1.3.1 Proposed System

The proposed school management system is a comprehensive software solution designed to streamline and enhance the administrative, academic, and communication aspects of educational institutions. this system will offer a user-friendly interface for teachers, students, and administrators to efficiently manage tasks such as student enrollment, attendance tracking, grade recording, resource allocation, and communication. with features like online gradebooks, parent-teacher communication portals, and real-time attendance monitoring, the system aims to improve the overall efficiency, transparency, and effectiveness of school management, ultimately enhancing the learning experience for students and reducing the administrative burden on educators and staff.

1.3.2 Advantages Of Proposed System

The proposed school management system offers several significant advantages over traditional manual systems. it streamlines administrative tasks, such as student enrolment, attendance tracking, and grade management, making the entire process more efficient and error-free. additionally, it enhances communication between teachers, students, and parents through a user-friendly online portal, providing real-time access to academic information and updates. the system also promotes data security and confidentiality, ensuring that sensitive

student information is safeguarded. moreover, it offers data analytics capabilities, enabling educators to gain insights into student performance and make informed decisions to improve educational outcomes. overall, this proposed system represents a modern and comprehensive solution that enhances the overall efficiency and effectiveness of school management.

2. TECHNOLOGIES LEARNT

2.1 HTML

Html Is The Backbone Of Web Development, And It Played A Fundamental Role In Our "School management system" Project. With Html, We Learned To Structure The Content Of Our Website. It Allowed Us To Create Web Pages, Define Headings, Paragraphs, Lists, Images, And Links. In Our Project, Html Was Used To Build The Foundational Structure Of The Site, Ensuring That Our Content Was Organized And Semantically Meaningful. It Also Enabled The Integration Of Multimedia Elements, Such As Images And Videos, And The Creation Of Links That Facilitated Navigation Within The Website. By Understanding Html, We Could Craft The Visual And Structural Framework Upon Which Our School management system Platform Was Built.

2.2 CSS

Css Is A Vital Component Of Web Development That Allowed Us To Control The Visual Presentation And Layout Of Our Website. We Utilized Css To Define The Colour Schemes, Fonts, Spacing, And Overall Aesthetics Of Our Platform. It Enabled Us To Create A Visually Appealing And Consistent User Interface, Ensuring A Seamless And Engaging Experience For Our Visitors. With Css, We Could Customize The Look And Feel Of Our Website, Making It Visually Captivating And In Line With The Artistic Theme Of Our Gallery. It Played A Crucial Role In Turning Our Html Structure Into An Attractive And User-Friendly Platform.

2.3 JS

We Harnessed Javascript To Add Interactivity And Functionality To Our Platform. This Technology Allowed Us To Create Dynamic Features Such As Real-Time Comments And Feedback Systems, Image Galleries, And User Authentication. By Mastering Javascript, We Could Make Our Website Come To Life, Providing Artists And Art Enthusiasts With A Responsive And Engaging User Experience. Javascript Was Used To Validate User Inputs, Handle Asynchronous Requests, And Enable Smooth Transitions And Animations. This Technology Was Instrumental In Ensuring The Website's User Interaction And Functionality, Contributing To The Overall Success Of Our Project.

2.4 REACT JS

We Harnessed Javascript To Add Interactivity And Functionality To Our Platform. This

Technology Allowed Us To Create Dynamic Features Such As Real-Time Comments And Feedback Systems, Image Galleries, And User Authentication. By Mastering Javascript, We Could Make Our Website Come To Life, Providing Artists And Art Enthusiasts With A Responsive And Engaging User Experience. Javascript Was Used To Validate User Inputs, Handle Asynchronous Requests, And Enable Smooth Transitions And Animations. This Technology Was Instrumental In Ensuring The Website's User Interaction And Functionality, Contributing To The Overall Success Of Our Project.

2.5 EXPRESS JS

We Adopted Express To Build The Server-Side Components Of Our Application. This Allowed Us To Handle Http Requests, Define Routes, And Manage The Backend Functionalities Required For User Authentication, Data Retrieval, And Api Endpoints. With Express, We Could Efficiently Communicate Between The Client-Side (Front-End) And The Server-Side (Back-End) Of Our Platform, Ensuring Seamless Data Transmission And Interaction. This Technology Was Instrumental In Creating A Secure And Performant Web Application, Supporting User Registration, Login, And Data Retrieval For Artists And Art Enthusiasts, Ultimately Enhancing The Overall Functionality And Responsiveness Of Our School management system.

2.6 NODE JS

We Integrated Node.Js To Power The Server-Side Components Of Our Application. It Enabled Us To Handle Data Processing, Manage Server Requests, And Establish A Robust BackEnd Infrastructure. With Node.Js, We Could Build Scalable, High-Performance ServerSide Applications, Ensuring That Our Website Could Efficiently Handle A Large Volume Of Users And Data. This Technology Allowed Us To Create Real-Time Features, User Authentication, And Data Storage, Facilitating The Seamless Operation Of Our School management system. Node.Js Was Instrumental In Making Our Project Responsive And Capable Of Handling Diverse User Interactions, Ultimately Enhancing The Overall Functionality And User Experience.

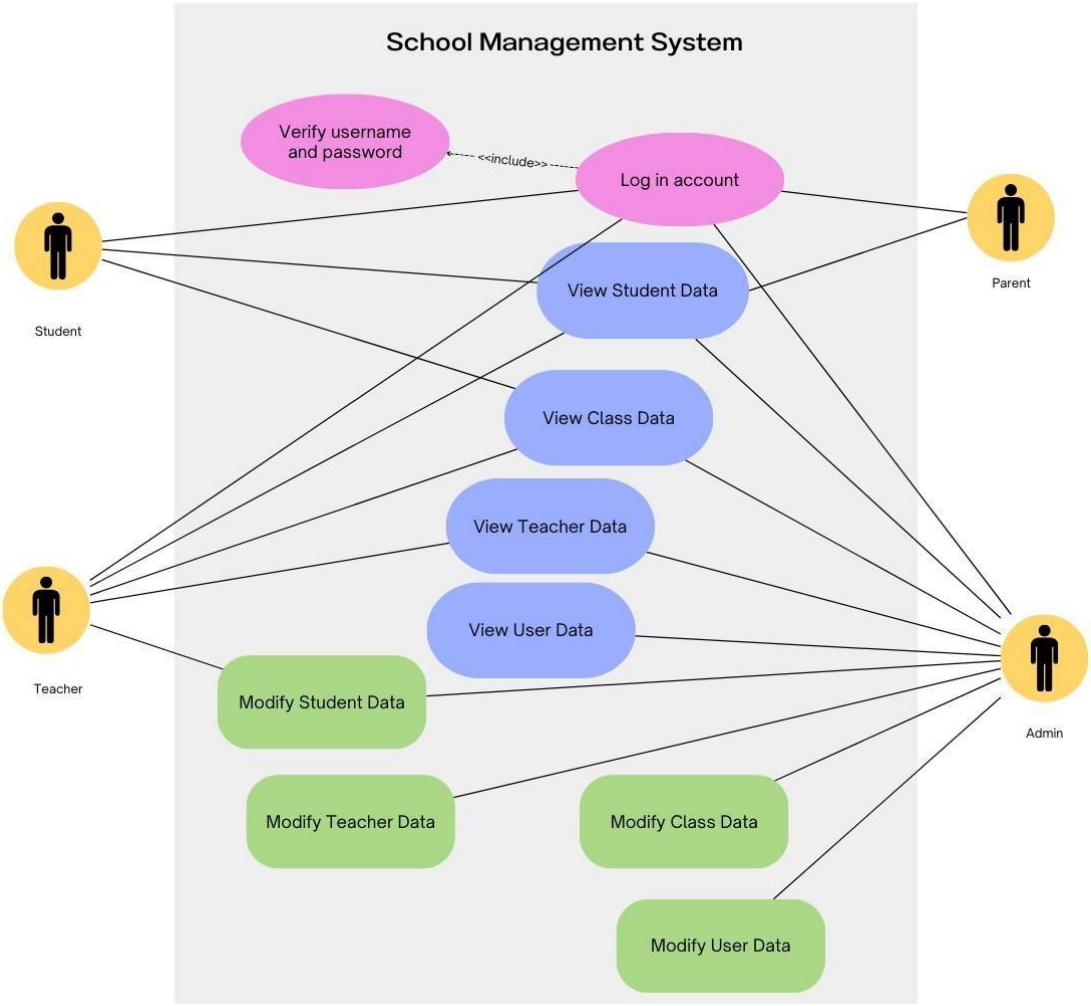
2.7 MONGO DB

Mongodb, A Nosql Database, Was Pivotal In Our "School management system Website" Project, Serving As Our Data Repository. Its Flexible Document-Based Structure Allowed Us To Efficiently Store, Manage, And Retrieve A Wide Range Of Data Related To Artists, Artworks, User Profiles, And Comments. Mongodb's Scalability Ensured Our Platform Could Handle Growing Data Volumes Generated By Artists And Art Enthusiasts, Enabling Real-Time Data

Updates And Retrieval For A Dynamic User Experience. Additionally, Its Geospatial Data Support Enhanced The Platform By Enabling Location-Based Search Features, Contributing Significantly To The Overall Functionality Of Our School management system Website.

3. SYSTEM DEIGN

3.1 System Architecture



(1)

3.2 Module Description

1. Student Module:

- Profile Management: Students Can Create And Manage Their Profiles, Including Personal Information, Contact Details, And A Profile Picture.
- Course Enrolment: Students Can View Available Courses, Select Their Courses, And Check Their Schedules.
- Attendance: Students Can View Their Attendance Records And Receive Notifications About Absenteeism.
- Grades And Results: Access To Exam Results, Progress Reports, And Gpa Calculations. - Library Access: Check Available Books, Borrow/Return Books, And Access Digital Resources.
- Communication: Send And Receive Messages To/From Teachers And Administrators.

- Events And Calendar: View School Events, Class Schedules, And Academic Calendars. - Fees And Payments: Check Fee Statements, Make Online Payments, And View Payment History.

2. Parent Module:

- Student Profiles: Access To Their Children's Profiles, Including Personal Information And Academic Records.
- Attendance Monitoring: View Their Child's Attendance And Receive Alerts For Absenteeism. - Grades And Progress Reports: Access To Their Child's Grades, Exam Results, And Progress Reports.
- Communication: Send And Receive Messages To/From Teachers And School Administrators.
- Events And Calendar: View School Events, Class Schedules, And Academic Calendars. - Fees And Payments: Access To Fee Statements, Make Online Payments, And View Payment History.
- Parent-Teacher Conferences: Schedule And Attend Parent-Teacher Conferences.

3. Teacher Module:

- Profile Management: Teachers Can Create And Manage Their Profiles, Including Qualifications And Contact Details.
- Class Management: Assign Courses, Manage Student Lists, And Mark Attendance. - Grading And Assessments: Enter Grades, Create And Grade Assignments, And Manage Exams.
- Communication: Send And Receive Messages To/From Students, Parents, And Administrators.
- Lesson Plans: Create And Manage Lesson Plans, Share Resources, And Upload Assignments.
- Library Access: Manage Library Resources, Checkouts, And Digital Materials.
- Attendance Reports: View And Generate Attendance Reports For Students.
- Leave Requests: Request And Manage Leave Requests.

4. Admin Module:

- User Management: Create, Edit, And Delete User Accounts (Students, Parents, Teachers), And Assign Roles And Permissions.
- Course And Curriculum Management: Add, Edit, And Delete Courses And Manage The Curriculum.
- School Calendar: Manage And Update School Events, Holidays, And Academic Calendars. - Finance And Fees: Generate Fee Statements, Manage Payment Records, And Handle Financial Transactions.
- Employee Management: Manage Teacher And Staff Profiles, Attendance, And Leave Requests.
- Reports And Analytics: Generate Various Reports, Such As Attendance Reports, Financial Reports, And Academic Performance Reports.
- Security And System Settings: Ensure The Security And Configuration Of The System.

3.3 System Specification 3.3.1 Software Requirements:

Creating a School Management System using the MERN (MongoDB, Express.js, React, Node.js) stack for Ethnus organization would require a variety of tools and technologies for both the frontend and backend development. Here's a list of tools and technologies commonly used for building such a system: **Frontend:**

1. **React**: React is a JavaScript library for building user interfaces. It's a key part of the MERN stack and is ideal for creating interactive, responsive user interfaces.
2. **Redux**: Redux is often used for state management in React applications. It helps manage the state of the application and ensure data consistency.
3. **React Router**: To manage the routing and navigation within the application, you can use React Router. It's essential for creating different pages and views.
4. **CSS**: Basic technologies for structuring and styling web pages.
5. **Bootstrap or Material-UI**: These are popular CSS frameworks that provide predesigned components for creating a modern and responsive UI.
6. **Axios or Fetch**: For making HTTP requests to the backend API.
7. **JWT (JSON Web Tokens)**: For implementing user authentication and authorization. **Backend:**

1. **Node.js**: Node.js is used for server-side scripting. It allows you to run JavaScript on the server, which makes it a great fit for the MERN stack.
2. **Express.js**: A minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.
3. **MongoDB**: A NoSQL database that stores data in JSON-like documents. It's ideal for handling structured and unstructured data, which is common in school management systems.
4. **Mongoose**: A MongoDB object modelling tool that provides a structured way to interact with MongoDB.
5. **Passport.js**: A popular authentication middleware for Node.js. It can be used to implement various authentication strategies, including JWT.
6. **JWT (JSON Web Tokens)**: As mentioned earlier, JWTs are used for user authentication and authorization in the backend as well.

This is a general set of tools and technologies that you can use to build a School Management System using the MERN stack. The specific requirements and tools may vary depending on the complexity and features you want to implement in the system.

3.3.2 Hardware Requirements:

Server: We need a capable web server to host our website. The hardware specifications of the server will depend on factors such as the expected traffic, the complexity of our website,

and our performance requirements. Common server options include dedicated servers, virtual private servers, or cloud-based hosting solutions.

Processor (CPU): The CPU power of our server should be adequate to handle the expected load. For our small to medium-sized website, multi-core processors with sufficient clock speed should be suitable. However, if we anticipate high traffic, we may need more powerful CPUs.

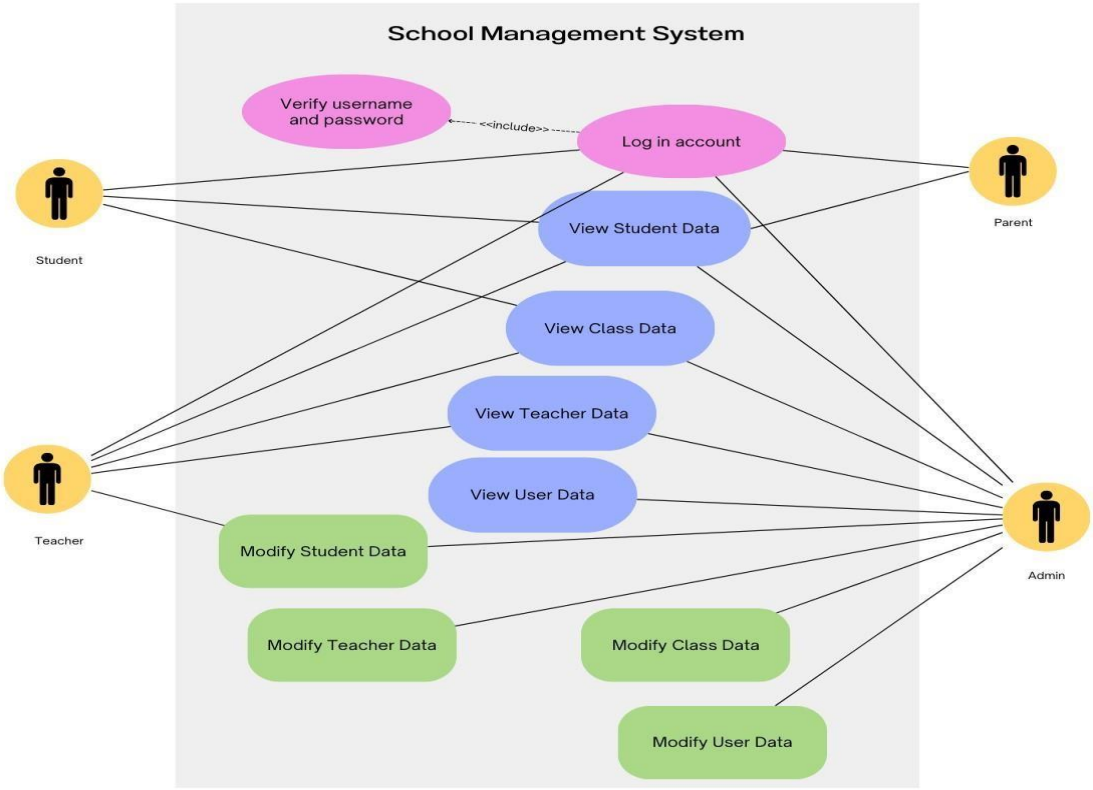
Memory (RAM): The amount of RAM is crucial for ensuring smooth website performance. The more RAM our server has, the better it can handle concurrent user requests and database operations.

Storage: Our server will need storage for our website's code, media files, and database. We should consider using Solid State Drives (SSDs) for faster data access and reduced loading times.

Network Interface: A robust network interface card (NIC) is essential for fast and reliable network connectivity. We should opt for high-speed connections to ensure efficient data transfer.

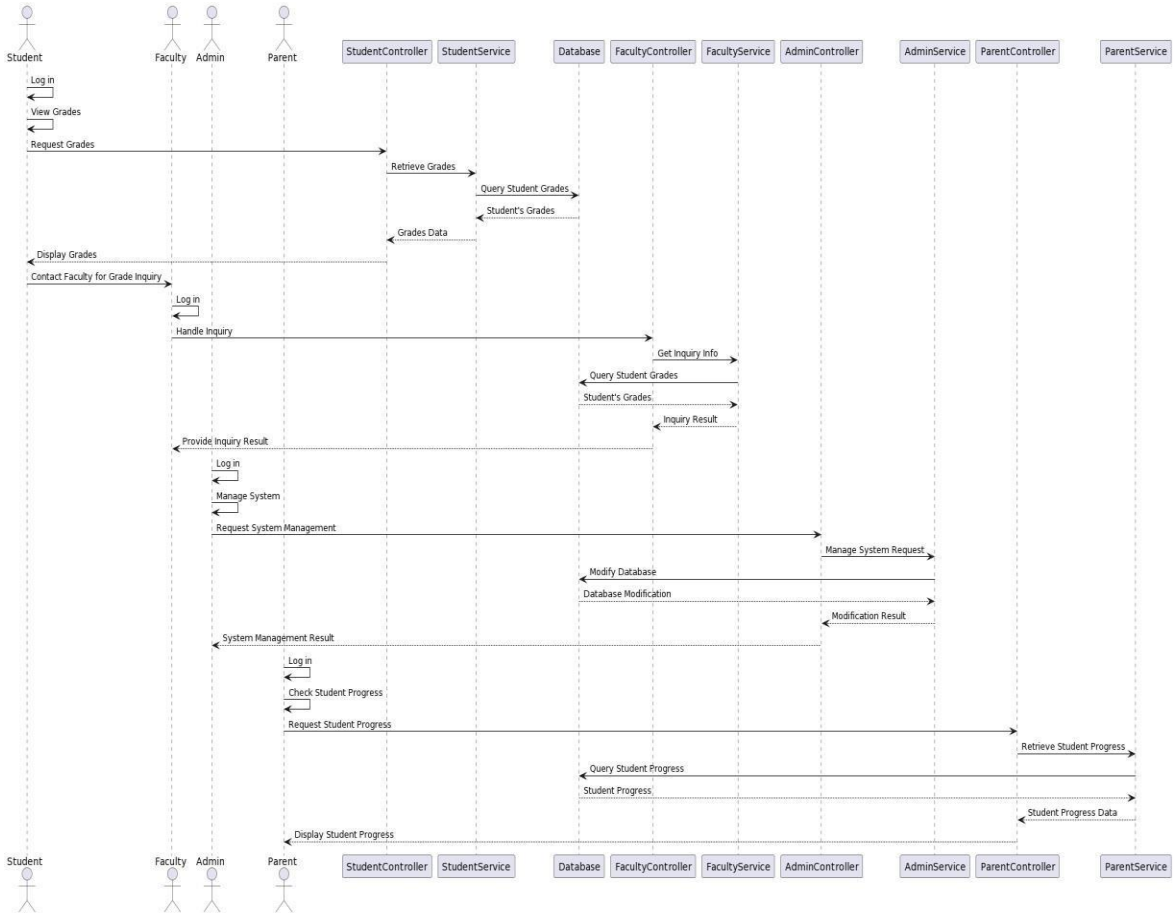
3.4 DETAILED DESIGN

3.4.1 Use case Diagram



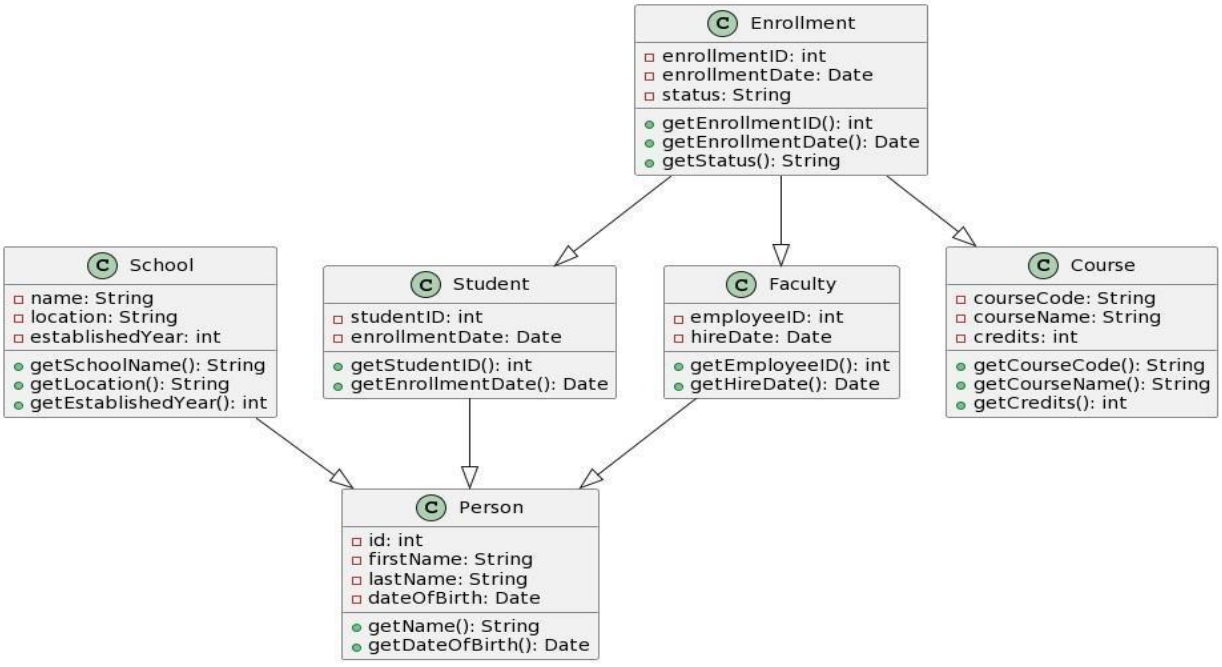
(2)

3.4.2 Sequence Diagram



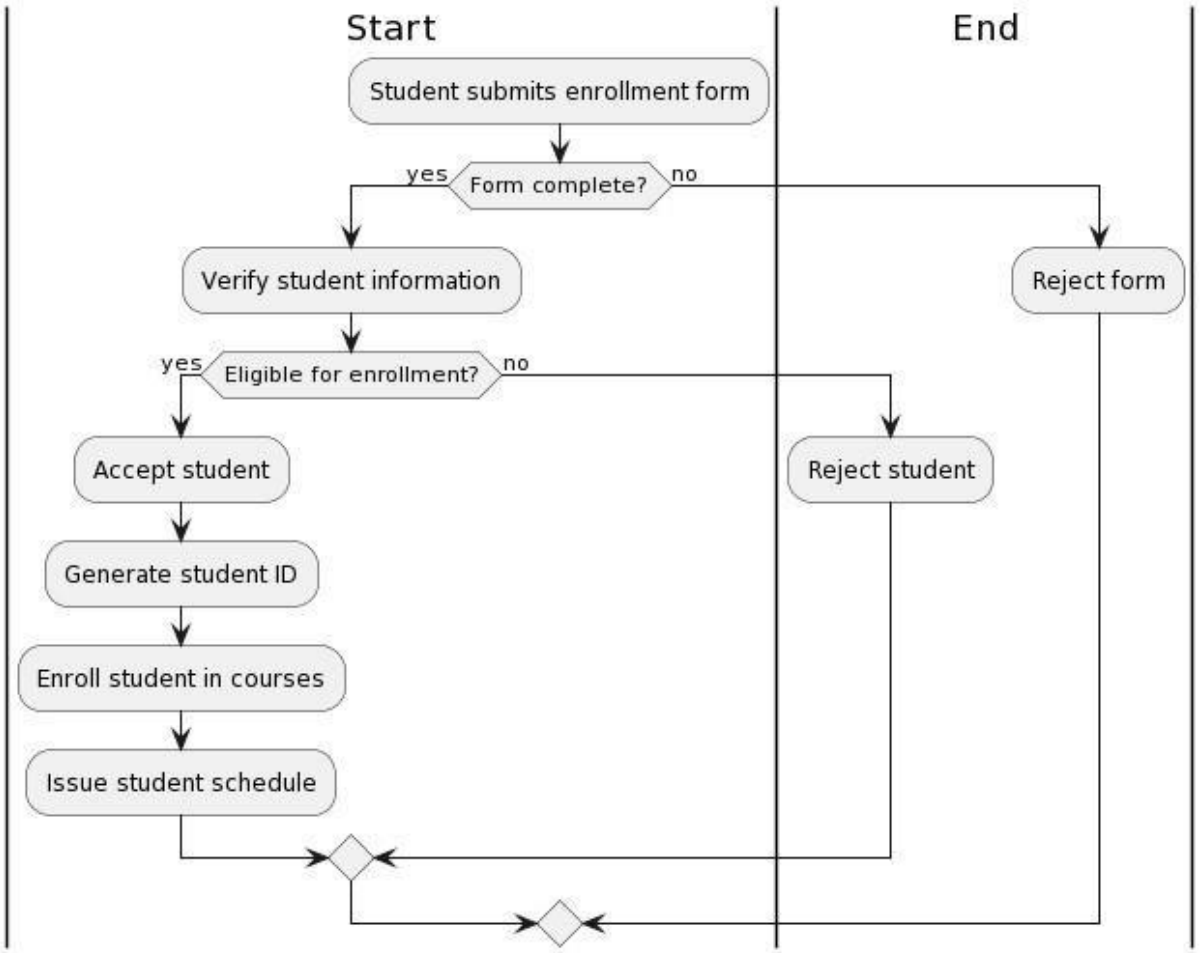
(3)

3.4.3 Class Diagram



(4)

3.4.4 Activity diagram



(5)

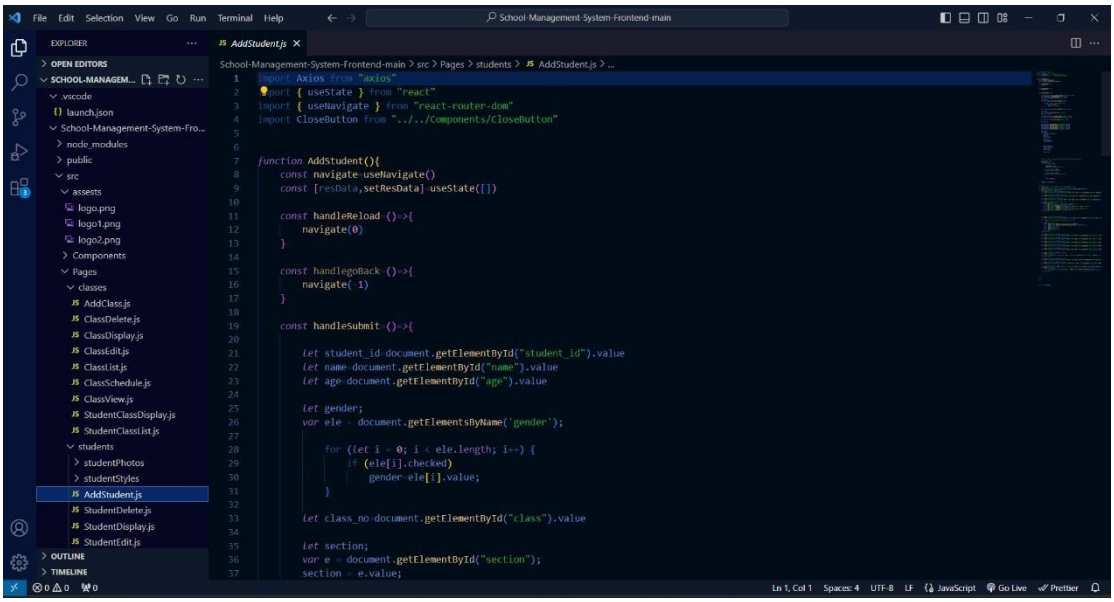
4.IMPLEMENTATION

4.1 Implementation details

The implementation of our "School management system" project was executed with precision and dedication, making use of Visual Studio Code (VSCode) as our primary integrated development environment. VSCode's versatility and powerful extensions facilitated a smooth development process, enabling our team to write and debug code efficiently. Throughout the implementation phase, we meticulously built the front-end using React, creating a dynamic and visually captivating user interface. The back-end, based on Express.js and Node.js, was carefully crafted to manage server-side functionality, including user authentication, real-time interactions, and data processing. MongoDB served as our robust NoSQL database, providing a scalable and flexible solution for data storage and retrieval. The project's development in VSCode offered a collaborative and streamlined environment, allowing for code version control and rapid testing, ultimately resulting in a well-structured.

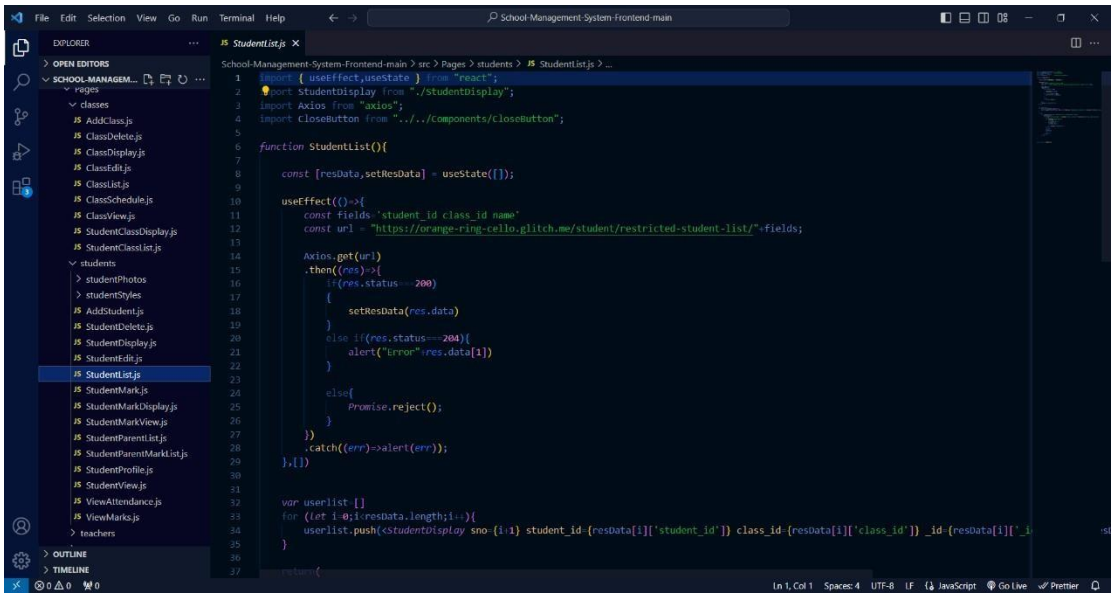
Codes:

About us page:



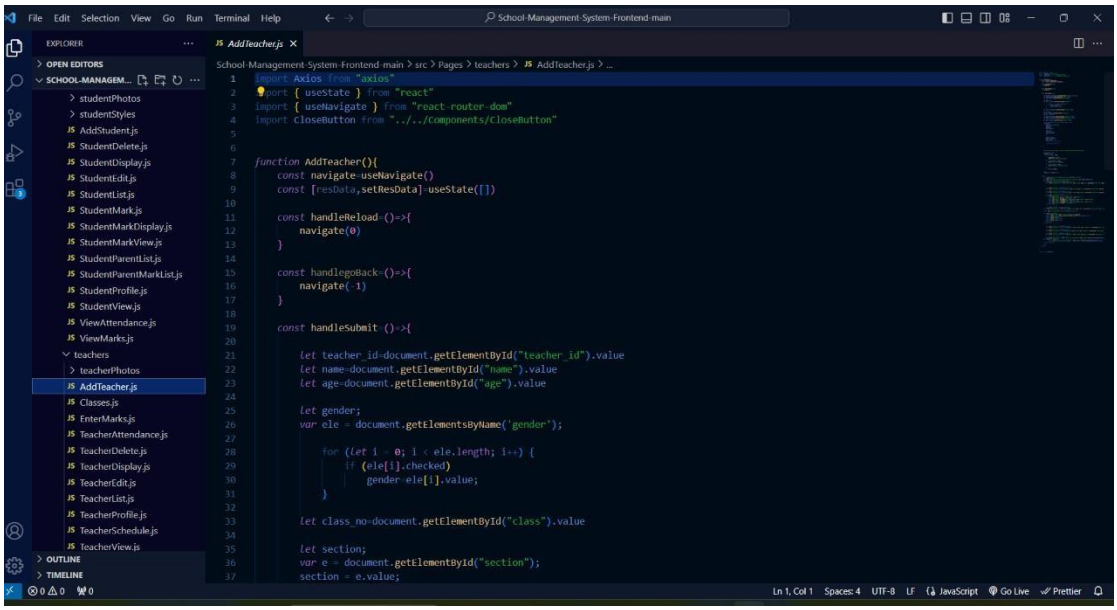
```
import Axios from "axios"
import { useState } from "react"
import { useNavigate } from "react-router-dom"
import CloseButton from "../../Components/CloseButton"


function AddStudent(){
    const navigate=useNavigate()
    const [resData,setResData]=useState([])

    const handleReload=()=>{
        navigate(0)
    }

    const handlegoBack=()=>{
        navigate(-1)
    }

    const handleSubmit=()=>{

        let student_id=document.getElementById("student_id").value
        let name=document.getElementById("name").value
        let age=document.getElementById("age").value

        let gender;
        var ele = document.getElementsByName('gender');

            for (let i = 0; i < ele.length; i++) {
                if (ele[i].checked)
                    gender=ele[i].value;
            }

        let class_no=document.getElementById("class").value

        let section;
        var e = document.getElementById("section");
        section = e.value;
```

STUDENT LIST PAGE:



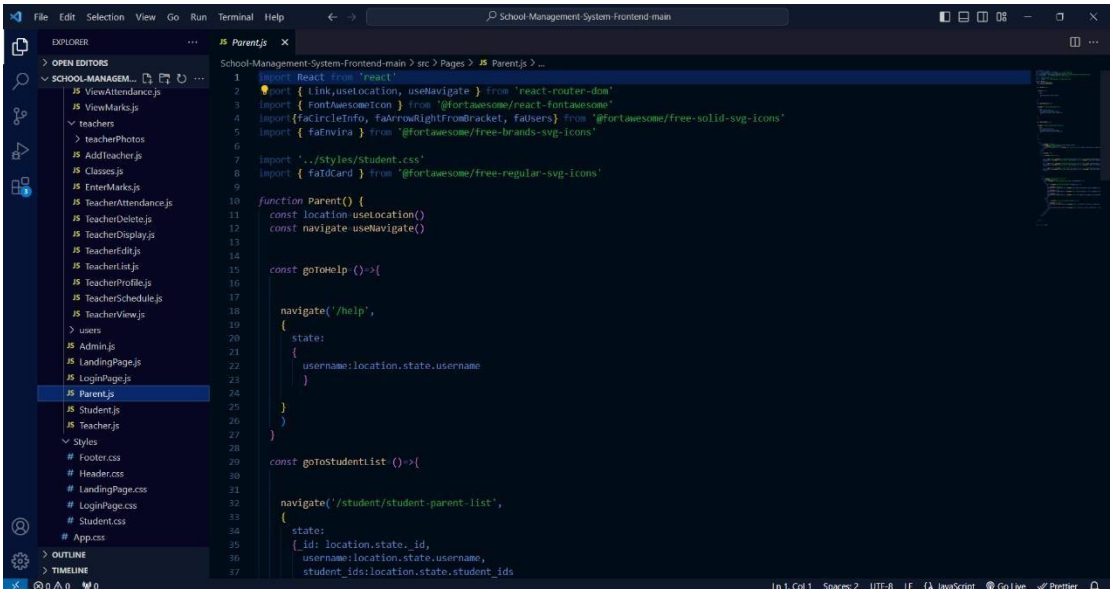
```
import { useEffect,useState } from "react";
import StudentDisplay from "./StudentDisplay";
import Axios from "axios";
import CloseButton from "../../Components/CloseButton";

function StudentList(){

    const [resData,setResData] = useState([]);

    useEffect(()=>{
        const fields='student_id class_id name'
        const url = "https://orange-ring-cello.glitch.me/student/restricted-student-list/"+fields;

        Axios.get(url)
        .then((res)=>{
            if(res.status===200)
            {
                setResData(res.data)
            }
            else if(res.status===204){
                alert("Error"+res.data[1])
            }

            else{
                Promise.reject();
            }
        })
        .catch((err)=>alert(err));
    },[])


    var userlist=[]
    for (let i=0;i<resData.length;i++){
        userlist.push(<StudentDisplay sno={i+1} student_id={resData[i]['student_id']} class_id={resData[i]['class_id']} _id={resData[i]['_i
    }

    return(
```

TEACHER LOGIN PAGE:



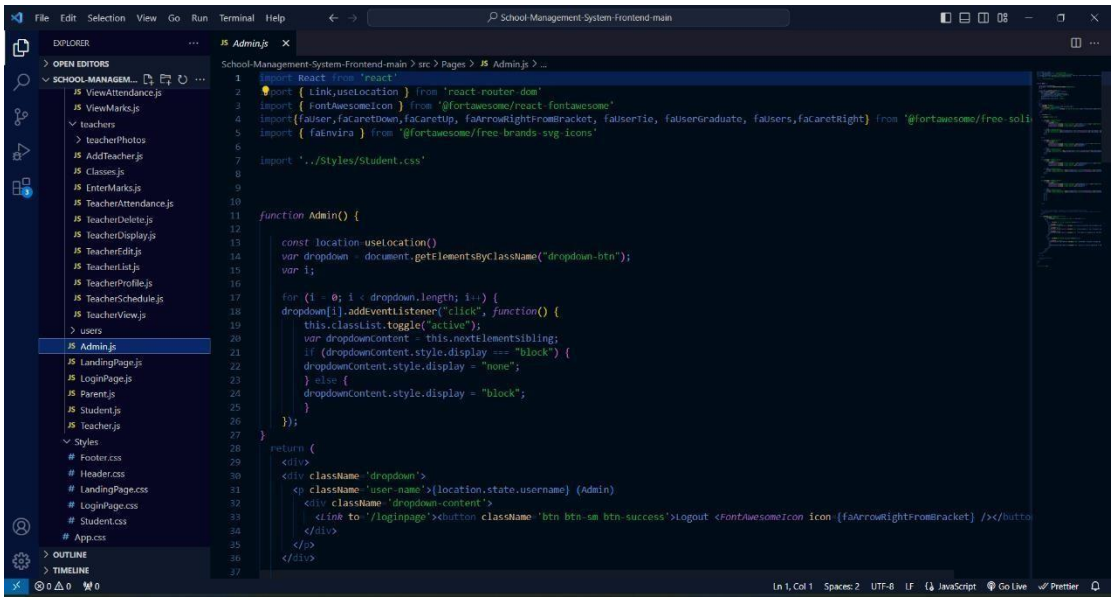
```
import Axios from "axios"
import { useState } from "react"
import { useNavigate } from "react-router-dom"
import CloseButton from "../../Components/CloseButton"


function AddTeacher(){
    const navigate=useNavigate()
    const [resData,setResData]=useState([])

    const handleReload=()=>{
        navigate(0)
    }

    const handlegoBack=()=>{
        navigate(-1)
    }

    const handleSubmit =()=>{

        let teacher_id=document.getElementById("teacher_id").value
        let name=document.getElementById("name").value
        let age=document.getElementById("age").value

        let gender;
        var ele = document.getElementsByName('gender');

            for (let i = 0; i < ele.length; i++) {
                if (ele[i].checked)
                    gender=ele[i].value;
            }

        let class_no=document.getElementById("class").value

        let section;
        var e = document.getElementById("section");
        section = e.value;
```

PARENT LOGIN PAGE:



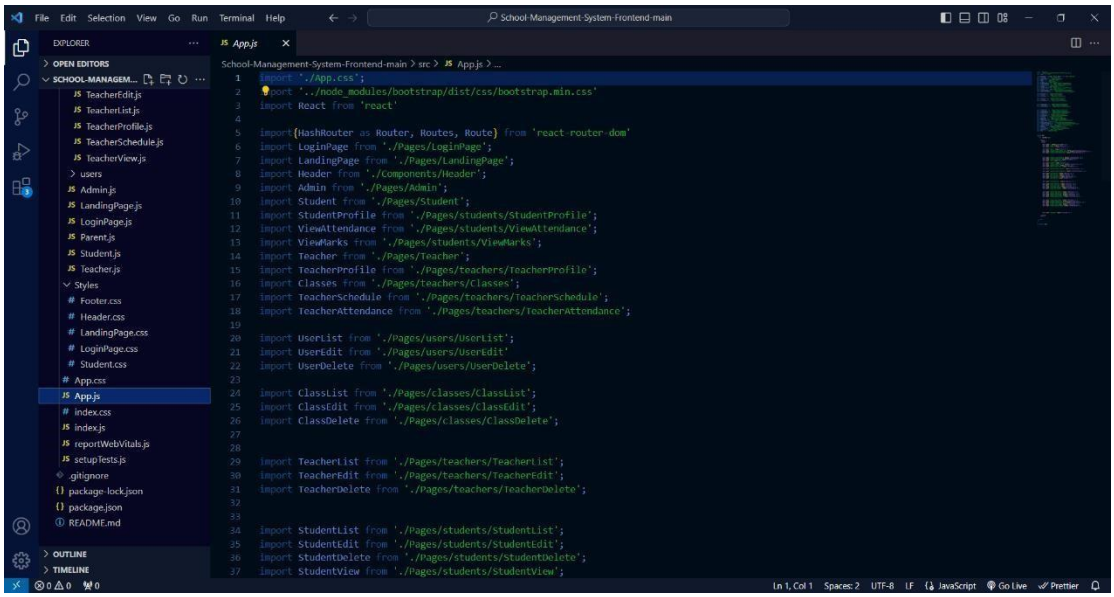
```
import React from 'react'
import { Link,useLocation, useNavigate } from 'react-router-dom'
import { FontAwesomeIcon } from '@fortawesome/react-fontawesome'
import{faCircleInfo, faArrowRightFromBracket, faUsers} from '@fortawesome/free-solid-svg-icons'
import { faEnvira } from '@fortawesome/free-brands-svg-icons'

import '../Styles/Student.css'
import { faIdCard } from '@fortawesome/free-regular-svg-icons'

function Parent() {
  const location=useLocation()
  const navigate=useNavigate()


  const goToHelp =()=>{


    navigate('/help',
    {
      state:
      {
        username:location.state.username
        }

    }
    )
  }

  const goToStudentList =()=>{


    navigate('/student/student-parent-list',
    {
      state:
      {_id: location.state._id,
        username:location.state.username,
        student_ids:location.state.student_ids
```

ADMIN LOGIN PAGE:



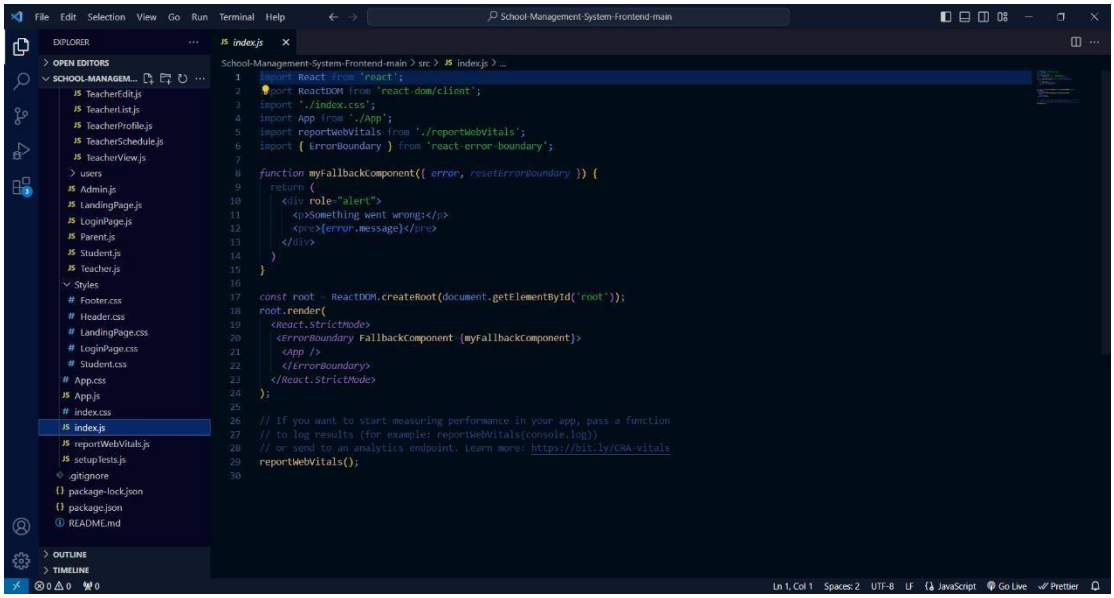
```
import React from 'react'
import { Link,useLocation } from 'react-router-dom'
import { FontAwesomeIcon } from '@fortawesome/react-fontawesome'
import{faUser,faCaretDown,faCaretUp, faArrowRightFromBracket, faUserTie, faUserGraduate, faUsers,faCaretRight} from '@fortawesome/free-soli
import { faEnvira } from '@fortawesome/free-brands-svg-icons'

import '../Styles/Student.css'



function Admin() {

    const location=useLocation()
    var dropdown = document.getElementsByClassName("dropdown-btn");
    var i;

    for (i = 0; i < dropdown.length; i++) {
    dropdown[i].addEventListener("click", function() {
        this.classList.toggle("active");
        var dropdownContent = this.nextElementSibling;
        if (dropdownContent.style.display === "block") {
        dropdownContent.style.display = "none";
        } else {
        dropdownContent.style.display = "block";
        }
    });
}
  return (
    <div>
    <div className 'dropdown'>
      <p className 'user-name'>{location.state.username} (Admin)
        <div className 'dropdown-content'>
          <Link to '/loginpage'><button className 'btn btn-sm btn-success'>Logout <FontAwesomeIcon icon={faArrowRightFromBracket} /></butto
        </div>
      </p>
    </div>
```

APP JS:

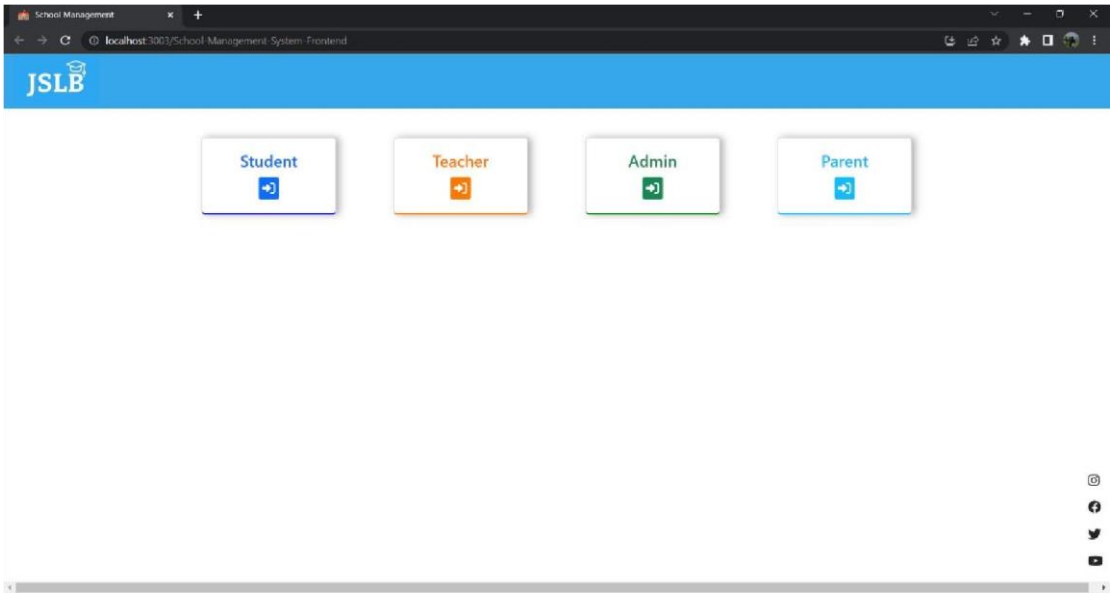


```
import './App.css';
import '../node_modules/bootstrap/dist/css/bootstrap.min.css'
import React from 'react'

import{HashRouter as Router, Routes, Route} from 'react-router-dom'
import LoginPage from './Pages/LoginPage';
import LandingPage from './Pages/LandingPage';
import Header from './Components/Header';
import Admin from './Pages/Admin';
import Student from './Pages/Student';
import StudentProfile from './Pages/students/StudentProfile';
import ViewAttendance from './Pages/students/ViewAttendance';
import ViewMarks from './Pages/students/ViewMarks';
import Teacher from './Pages/Teacher';
import TeacherProfile from './Pages/teachers/TeacherProfile';
import Classes from './Pages/teachers/Classes';
import TeacherSchedule from './Pages/teachers/TeacherSchedule';
import TeacherAttendance from './Pages/teachers/TeacherAttendance';

import UserList from './Pages/users/UserList';
import UserEdit from './Pages/users/UserEdit'
import UserDelete from './Pages/users/UserDelete';

import ClassList from './Pages/classes/ClassList';
import ClassEdit from './Pages/classes/ClassEdit';
import ClassDelete from './Pages/classes/ClassDelete';


import TeacherList from './Pages/teachers/TeacherList';
import TeacherEdit from './Pages/teachers/TeacherEdit';
import TeacherDelete from './Pages/teachers/TeacherDelete';


import StudentList from './Pages/students/StudentList';
import StudentEdit from './Pages/students/StudentEdit';
import StudentDelete from './Pages/students/StudentDelete';
import StudentView from './Pages/students/StudentView';
```

INDEX JS:

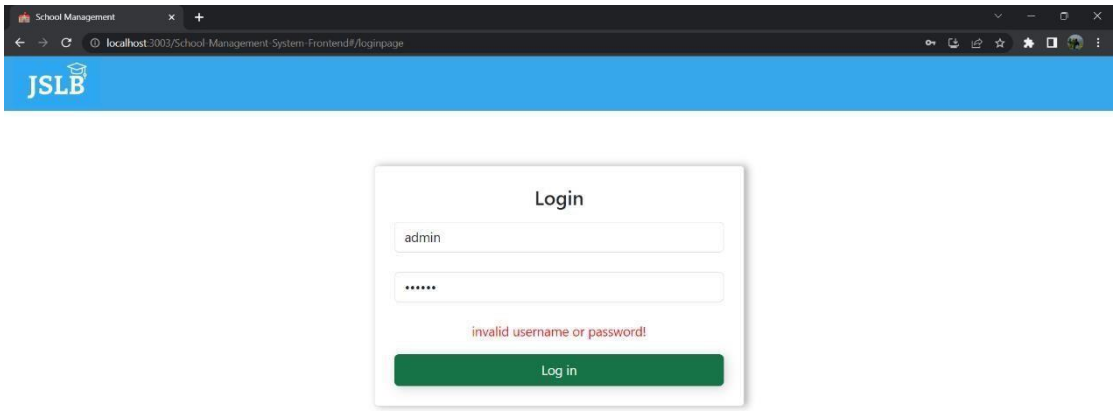


5. TEST RESULTS

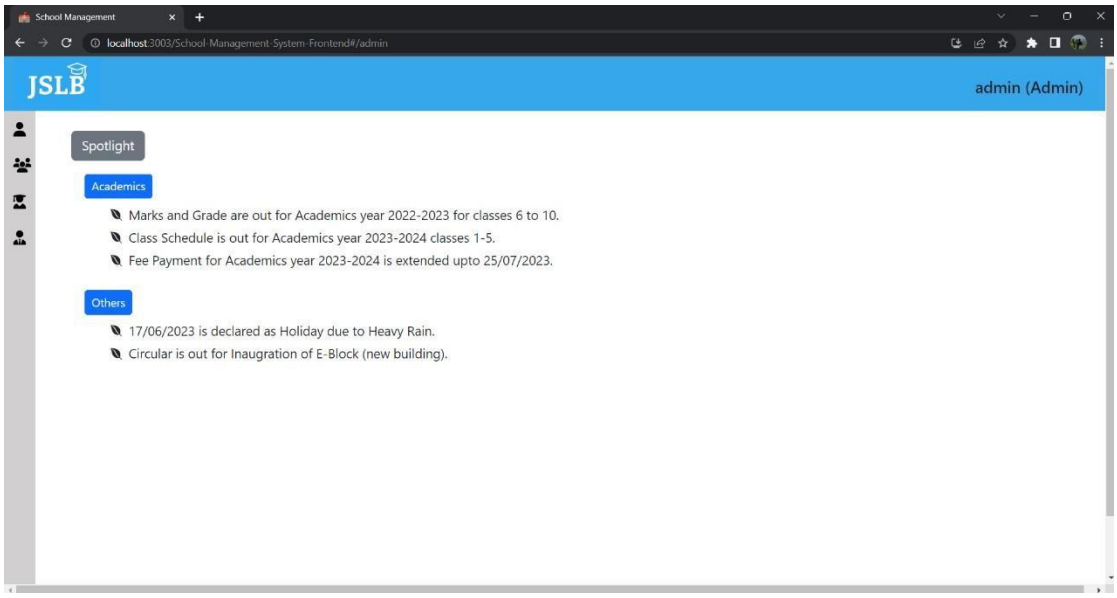


5.1 TEST CASES

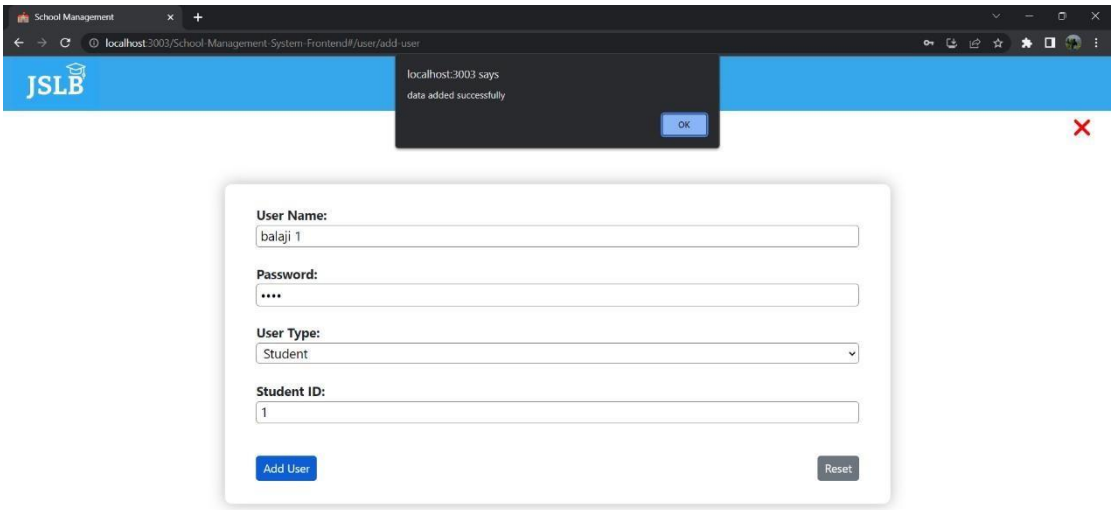
Test case1: Wrong admin username/password → Outcome – Fail → Test case – Passed



Test case2: Correct admin username & password → Outcome – Pass → Test case – Passed

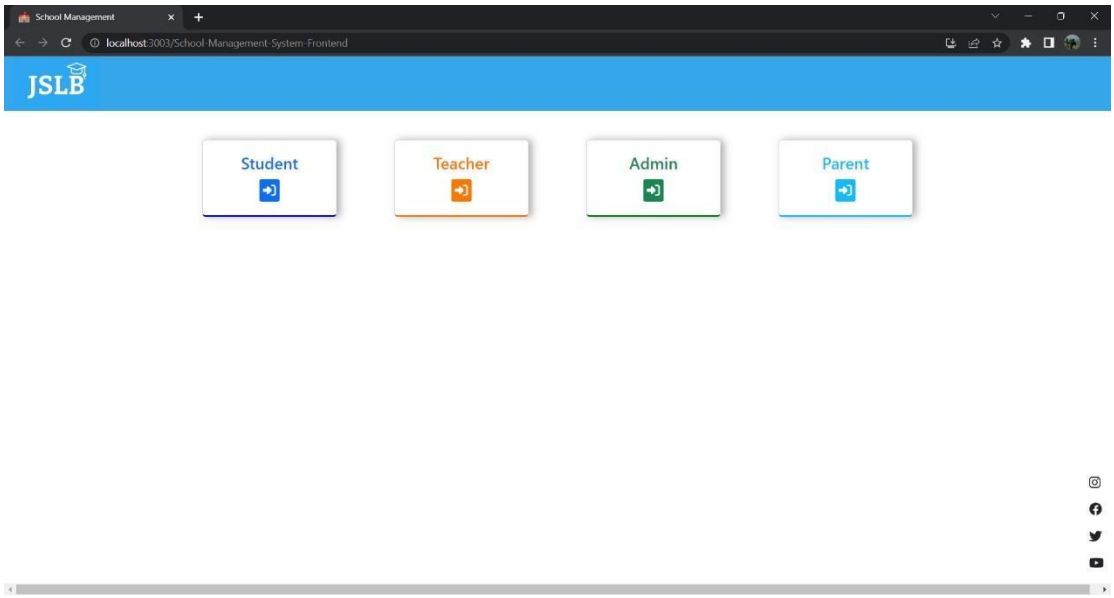


Test case3: Add user→ Outcome – Pass → Test case – Passed

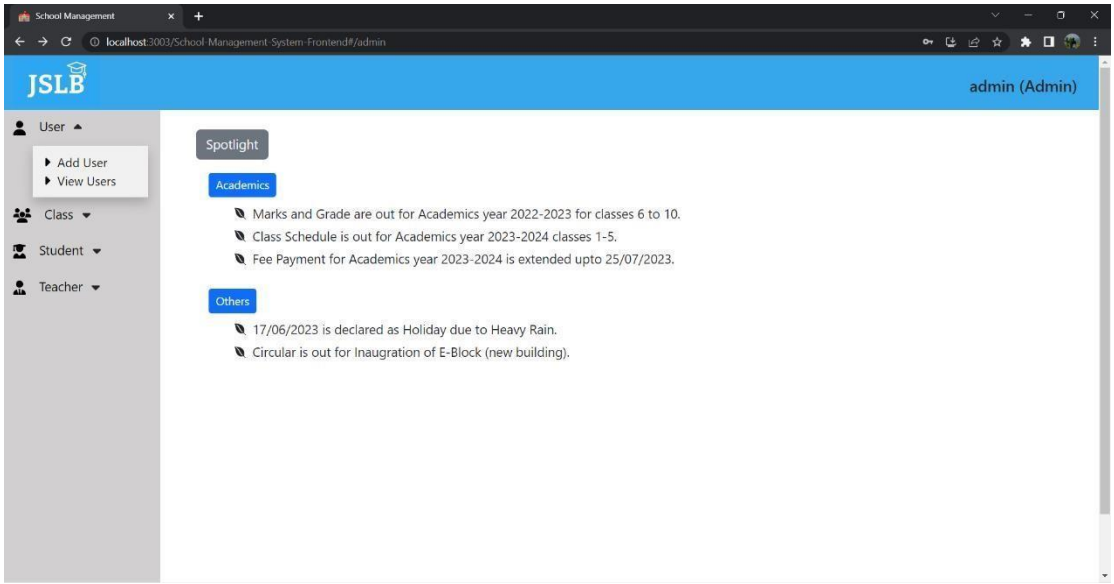


6. RESULTS AND DISCUSSIONS

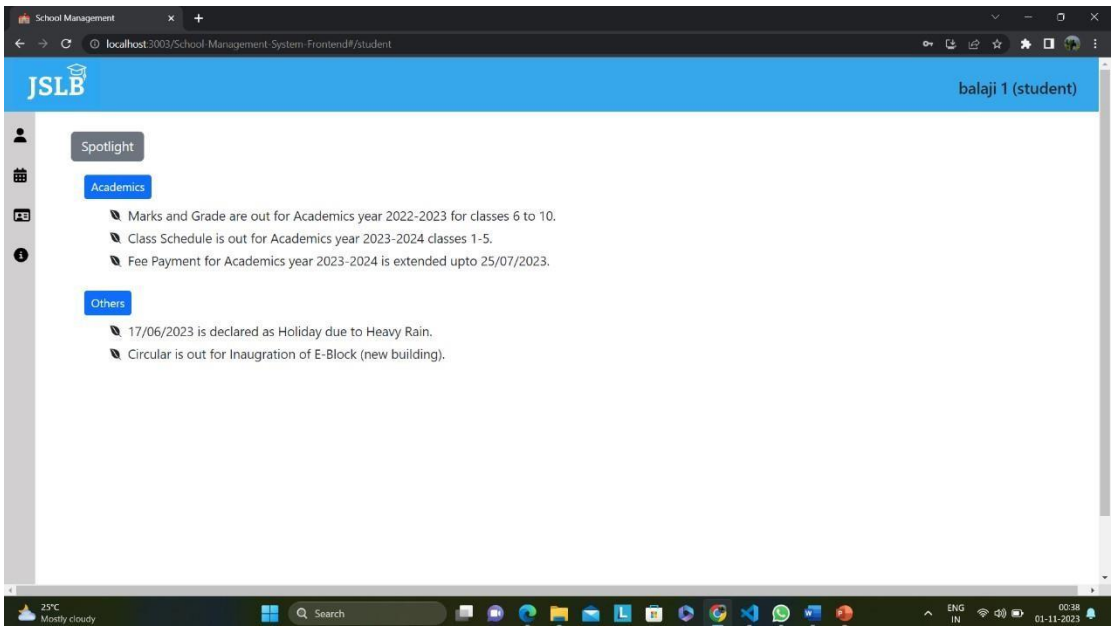
Home page:



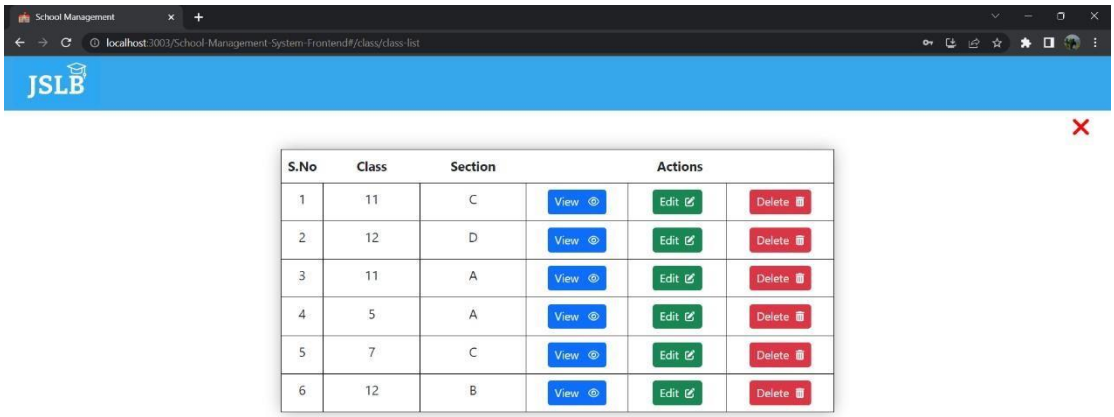
Admin Login page:



Student Login page:

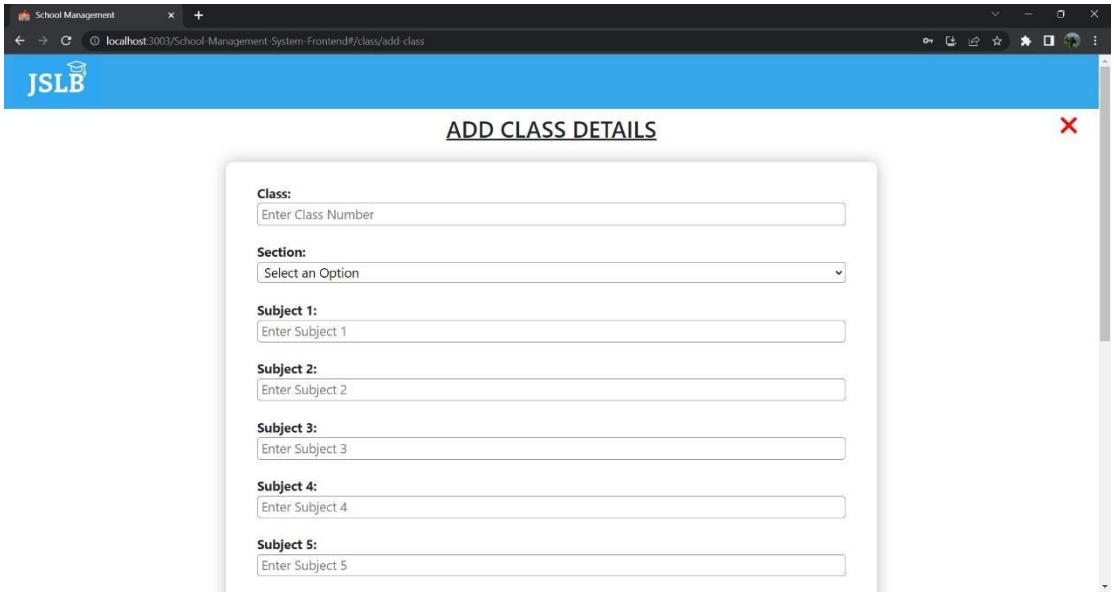


CLASS DETAILS:

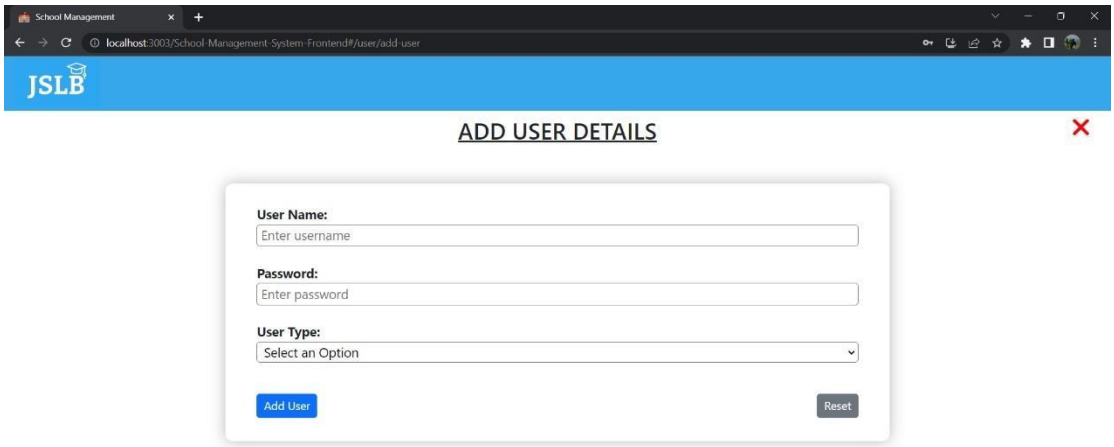


S.No	Class	Section		Actions	
1	11	C	View	Edit	Delete
2	12	D	View	Edit	Delete
3	11	A	View	Edit	Delete
4	5	A	View	Edit	Delete
5	7	C	View	Edit	Delete
6	12	B	View	Edit	Delete

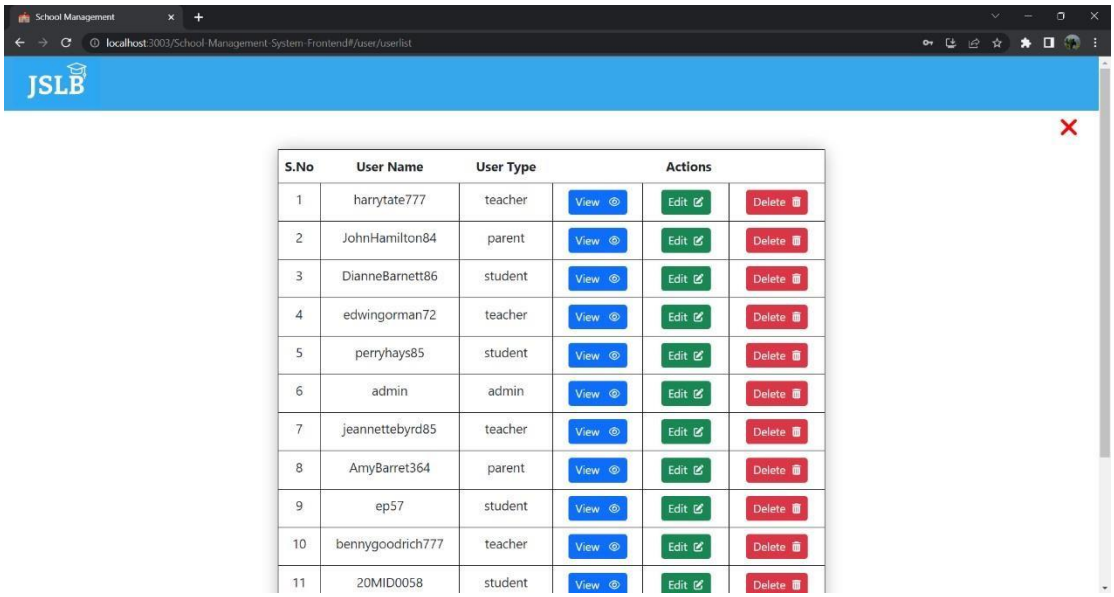
ADD CLASS:



ADD USER DETAILS PAGE:

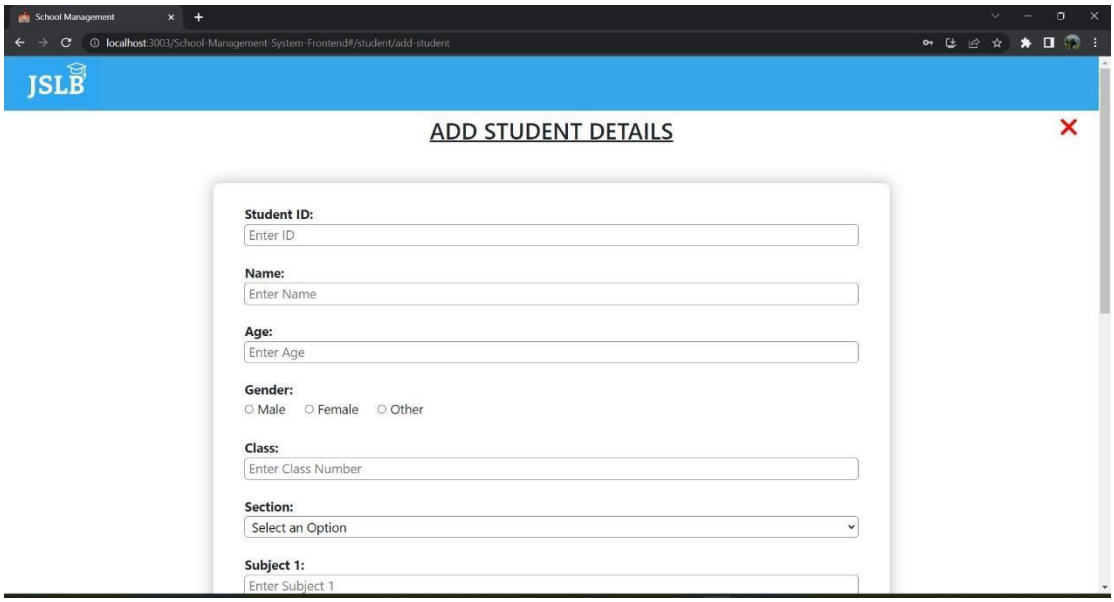


USERS ADDED:

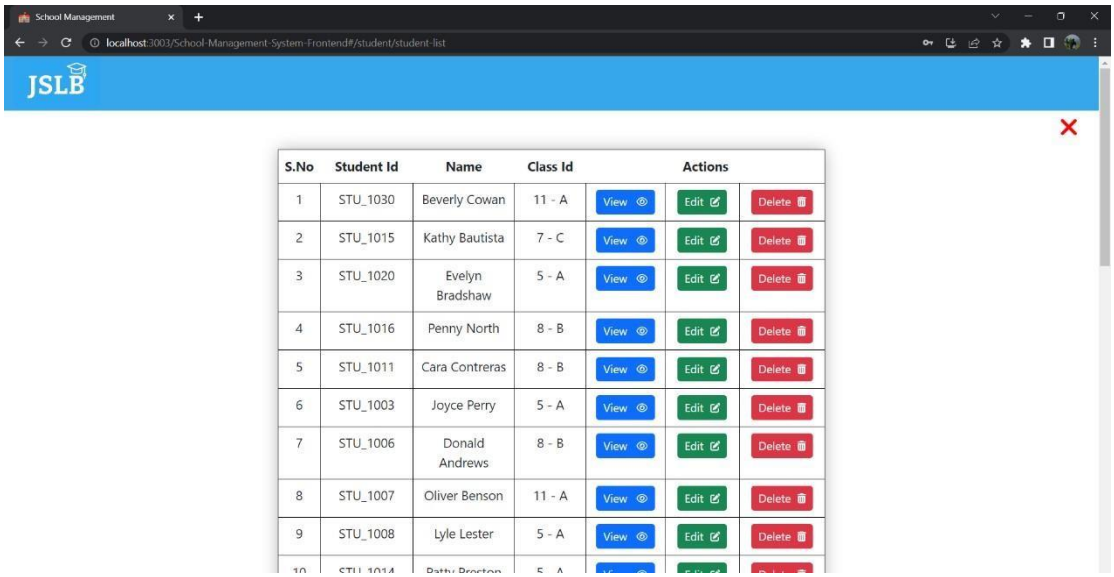


S.No	User Name	User Type		Actions	
1	harrytate777	teacher	View	Edit	Delete
2	JohnHamilton84	parent	View	Edit	Delete
3	DianneBarnett86	student	View	Edit	Delete
4	edwingorman72	teacher	View	Edit	Delete
5	perryhays85	student	View	Edit	Delete
6	admin	admin	View	Edit	Delete
7	jeannettebyrd85	teacher	View	Edit	Delete
8	AmyBarret364	parent	View	Edit	Delete
9	ep57	student	View	Edit	Delete
10	bennygoodrich777	teacher	View	Edit	Delete
11	20MID0058	student	View	Edit	Delete

ADD STUDENT DETAILS:

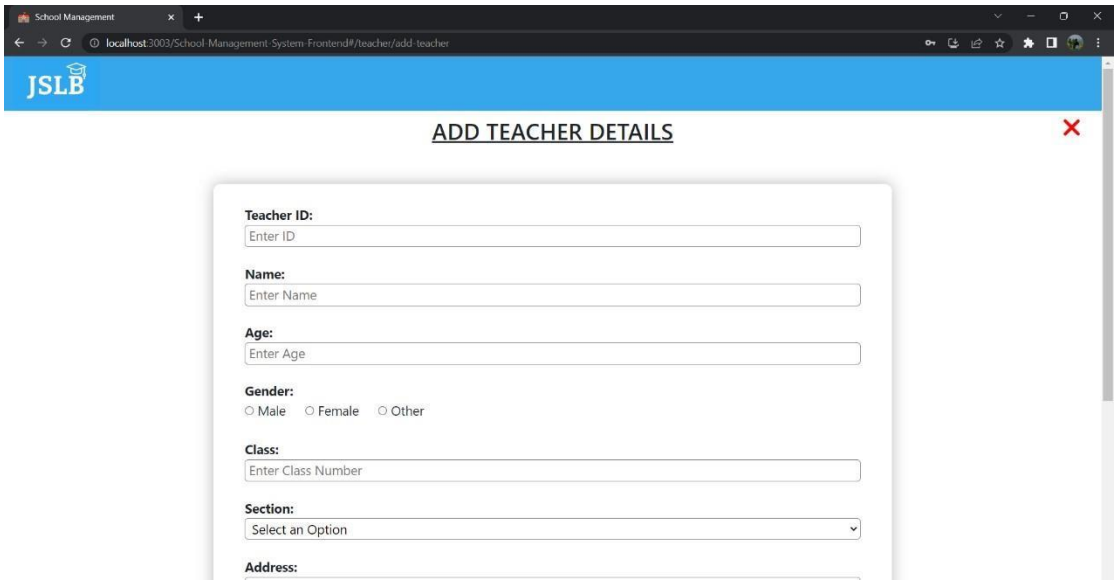


ADDED STUDENTS DETAILS:

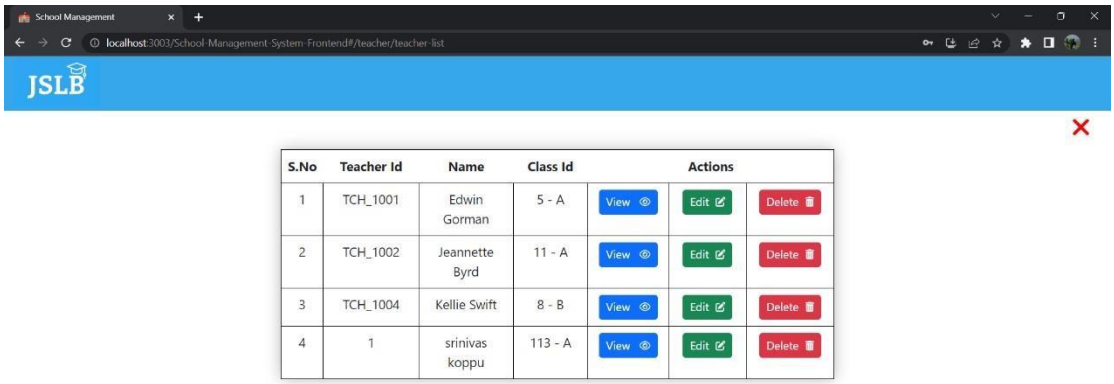


S.No	Student Id	Name	Class Id	Actions		
1	STU_1030	Beverly Cowan	11 - A	View	Edit	Delete
2	STU_1015	Kathy Bautista	7 - C	View	Edit	Delete
3	STU_1020	Evelyn Bradshaw	5 - A	View	Edit	Delete
4	STU_1016	Penny North	8 - B	View	Edit	Delete
5	STU_1011	Cara Contreras	8 - B	View	Edit	Delete
6	STU_1003	Joyce Perry	5 - A	View	Edit	Delete
7	STU_1006	Donald Andrews	8 - B	View	Edit	Delete
8	STU_1007	Oliver Benson	11 - A	View	Edit	Delete
9	STU_1008	Lyle Lester	5 - A	View	Edit	Delete

ADD TEACHER DETAILS:



ADDED TEACHER DETAILS:



A School Management System Has A Significant Impact On The Efficiency And Effectiveness Of Educational Institutions. By Automating Administrative Tasks, Such As Student Enrollment, Attendance Tracking, And Grade Management, It Streamlines The Daily Operations Of Schools, Reducing Paperwork And Minimizing Human Errors. This Leads To Improved Accuracy In Record-Keeping And Frees Up Educators And Staff To Focus More On Teaching And Student Support. Additionally, School Management Systems Facilitate Better Communication Between Teachers, Students, And Parents Through Online Portals, Helping To Enhance The Overall Educational Experience. Furthermore, They Provide Valuable Data And Analytics That Can Inform Decision-Making, Enabling Schools To Identify Areas For Improvement And Make Informed Strategic Choices. In Summary, A Well-Implemented

School Management System Contributes To Greater Efficiency, Transparency, And Accountability In Educational Institutions, Ultimately Benefiting Both Students And The School Community.

7. CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

The School Management System developed using the MERN (MongoDB, Express.js, React, Node.js) stack for Ethnus Organization represents a significant milestone in modernizing and enhancing the educational administration processes. This comprehensive solution has seamlessly integrated various aspects of school management, including student information, attendance tracking, grade management, and communication channels between teachers, students, and parents. Through this system, Ethnus Organization has not only improved its operational efficiency but also provided a more transparent and userfriendly interface for its stakeholders. The MERN stack's flexibility and scalability have ensured that the School Management System can easily adapt to the evolving needs of the organization and the education sector in general.

Furthermore, the successful implementation of this School Management System underscores Ethnus Organization's commitment to leveraging cutting-edge technology to improve educational outcomes. The MERN stack's robust architecture has ensured that the system is highly reliable, secure, and easily maintainable, which is crucial for an organization like Ethnus that caters to a diverse and dynamic student and faculty population. In conclusion, the MERN-based School Management System has not only streamlined administrative tasks but also empowered the Ethnus community with valuable tools to foster communication, data-driven decision-making, and ultimately, the continued pursuit of educational excellence. This project serves as a testament to the power of technology in transforming educational institutions and ensuring a brighter future for students and staff alike.

7.2 FUTURE WORK

The future work for a school management system involves continuous innovation and adaptation to meet the evolving needs of educational institutions. This includes enhancing the system's user interface and experience to ensure user-friendliness for administrators, teachers, parents, and students. Integration with emerging technologies like artificial intelligence and machine learning can enable predictive analytics for student performance and help tailor educational strategies. Improved security measures must be implemented to safeguard sensitive student and staff data. Additionally, the system should support remote learning and collaboration tools, reflecting the increasing importance of online education. Ongoing updates and maintenance are essential to keep the school management system current and efficient in managing the administrative, academic, and communication aspects of educational institutions.

8. REFERENCES

[1]. E. Burke and W. Erben. Practice and Theory of Automated Timetabling, Third InternationalConference, Germany, Springer Private Limited, August 2000

[2]. J. G. Hedberg et. al. (1992). Educational information systems: Problems of the smalleducational organisation. Australian Journal of Educational Technology, 8(2), 132160.http://www.ascilite.org.au/ajet/ajet8/hedberg.html

[3]. M. Marte. Models and Algorithms for School Timetabling, A Constraint-ProgrammingApproach, Ph.D dissertation, an der Fakult¨at f¨ur Mathematik, Informatik und Statistik der Ludwig-Maximilians-Universit ¨at M¨unchen, July, 2002

[4] R.J. Willemen. School Timetable Construction: Algorithms and Complexity, Thesis, Facultyof Mathematics and Computer Science, Technische Universiteit Eindhoven, 2002.

[5]. S. Petrovic and E. Burke. University Timetabling, School of Computer Science and Information Technology, University of Nottingham, 2002, pp. 1-4