SRI KRISHNA COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

Approved by AICTE | Affiliated to Anna University Chennai|
Accredited by NBA - AICTE| Accredited by NAAC with ‘A’ Grade
KOVAIPUDUR, COIMBATORE 641042

STUDENT E-LEARNING PORTAL

A PROJECT REPORT

Submitted by

SANGAMITHRA M	**727822TUCS201**
SIVA NITHIYA SREE C	**727822TUCS223**
SOUNDARIYA R	**727822TUCS227**
YASHASHWINI N	**727822TUCS258**

In partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

JULY 2024

BONAFIDE CERTIFICATE

Certified that this project report **STUDENT E-LEARNING PORTAL** is the bonafide work **SANGAMITHRA M 727822TUCS201, SIVA NITHIYA SREE C 727822TUCS223, SOUNDARIYA R 727822TUCS227, YASHASHWINI N 727822TUCS258** who carried out the project work under my supervision.

SIGNATURE

Ms. A. GOMATHY

SUPERVISOR
Assistant Professor,

Department of Computer Science and Engineering,

Sri Krishna College of Technology,Coimbatore- 641042

SIGNATURE

Dr. M. UDHAYAMOORTHI

HEAD OF THE DEPARTMENT

Department of Computer Science and Engineering,
Sri Krishna College of Technology,
Coimbatore-641042

Certified that the candidate was examined by me in the Project Work Viva Voce examination held on ______________at Sri Krishna College of Technology, Coimbatore-641042.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

First and foremost we thank the **Almighty** for being our light and for showering his gracious blessings throughout the course of this project.

We express our gratitude to our beloved Principal, **Dr. M.G. Sumithra,** for providing all facilities.

We are grateful to our beloved Head, Computing Sciences **Dr.T. Senthilnathan,** for her tireless and relentless support.

With the grateful heart, our sincere thanks to our Head of the Department **Dr. M. Udhayamoorthi**, Department of Computer Science and Engineering for the motivation and all support to complete the project work.

We thank **Ms. A. Gomathy**, Assistant Professor, Department of Computer Science and Engineering, for his motivation and support.

We are thankful to all the **Teaching and Non-Teaching Staff** of Department of Computer Science and Engineering and to all those who have directly and indirectly extended their help to us in completing this project work successfully.

We extend our sincere thanks to our **family members** and our beloved **friends**, who had been strongly supporting us in all our endeavour.

ABSTRACT

The Student E-Learning Portal is an innovative software solution aimed at revolutionizing the online learning experience for students. This comprehensive platform empowers students by providing seamless access to a wide range of educational resources and tools. Key features of the portal include course listings and enrollment management, which allow students to easily discover and register for courses. Once enrolled, students can access learning materials, submit assignments, and receive feedback through a user-friendly interface powered by React on the frontend and Spring Boot Microservices on the backend, with a robust MySQL database ensuring data integrity and performance. The portal also fosters an interactive learning environment through discussion forums, enabling students to engage in academic discussions, collaborate on projects, and seek help from peers and instructors. User authentication and profile management ensure a personalized experience, while progress tracking and reporting tools help students monitor their academic achievements and identify areas for improvement. Additionally, the portal includes a review and rating system, allowing students to share feedback on courses and instructors, further enhancing the quality of education. Security is a top priority, with advanced measures implemented to protect user data and ensure a safe learning environment. The Student E-Learning Portal aims to promote academic success, facilitate collaboration, and support the overall educational journey of students, making it an indispensable tool in the modern educational landscape.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Acronym
HTML	HYPERTEXT MARKUP LANGUAGE
CSS	CASCADING STYLESHEET
JS	JAVASCRIPT
SDLC	SOFTWARE DEVELOPMENT LIFE CYCLE
SQL	STRUCTURED QUERY LANGUAGE

CHAPTER 1

INTRODUCTION

This project aims to deliver a comprehensive and intuitive solution for managing online learning and course administration through a Student E-Learning

1.1 PROBLEM STATEMENT

How can we develop a Student E-Learning Portal that enables students to efficiently enroll in courses, access learning materials, submit assignments, and track their academic progress, while providing a user-friendly interface, fostering engagement, and ensuring data security?

1.2 OVERVIEW

In the contemporary educational landscape, students and educators often encounter challenges related to course management, access to learning resources, and tracking academic performance. Traditional learning environments and existing online platforms may suffer from fragmented interfaces, limited interaction features, and inadequate tools for progress monitoring. To address these challenges, we propose the development of a Student E-Learning Portal. This platform will utilize modern web technologies to create a cohesive and engaging online learning environment. Key features will include streamlined course enrollment, easy access to learning materials, interactive discussion forums, and robust progress tracking. By integrating these features into a single platform, the portal aims to enhance the learning experience, improve student engagement, and provide a secure and user-friendly environment for academic success.

1.3 OBJECTIVE

The primary objective of this project is to create a Student E-Learning Portal that simplifies online education by enabling easy course enrollment, providing quick access to learning materials, streamlining assignment submissions, fostering student engagement, tracking academic progress, and ensuring data security.

CHAPTER 2

SYSTEM SPECIFICATION

In this chapter, we are going to see the software that we have used to build the website. This chapter gives you a small description about the software used in the project.

2.1 VS CODE

Visual Studio Code is a source code editor developed by Microsoft for Windows, Linux, and macOS. It includes support for debugging, embedded Git control, syntax highlighting, intelligent code completion, snippets, and code refactoring. It is also customizable, so users can change the editor's theme, keyboard shortcuts, and preferences.

VS Code is an excellent code editor for React projects. It is lightweight, customizable, and has a wide range of features that make it ideal for React development. It has built-in support for JavaScript, JSX, and TypeScript, and enables developers to quickly move between files and view detailed type definitions. It also has a built-in terminal for running tasks. Additionally, VS Code has an extensive library of extensions that allow developers to quickly add features like code snippets, debugging tools, and linting support to their projects.

2.2 LOCAL STORAGE

Local storage is a type of web storage for storing data on the client side of a web browser. It allows websites to store data on a user’s computer, which can then be accessed by the website again when the user returns. Local storage is a more secure alternative to cookies because it allows websites to store data without having to send it back and forth with each request. Local storage is a key-value pair storage mechanism, meaning it stores data in the form of a key and corresponding value. It is similar to a database table in that it stores data in columns and rows, except that local storage stores the data in the browser rather than in a database. Local storage is often used to store user information such as preferences and settings, or to store data that is not meant to be shared with other websites. It is also used to cache data to improve the performance of a website. Local storage is supported by all modern web browsers, including Chrome,

Firefox, Safari, and Edge. It is accessible through the browser's JavaScript API. Local storage is a powerful tool for websites to store data on the client side. It is secure, efficient, and can be used to store data that does not need to be shared with other websites.

Local Storage is a great way to improve the performance of a website by caching data. Local storage in web browsers allows website data to be stored locally on the user's computer. It is a way of persistently storing data on the client side, which is not sent to the server with each request. This allows users to store data such as preferences, login information, and form data without needing to send it to a server. It is typically stored in a browser's cookie file, but it can also be stored in other locations such as HTML5 Local Storage and Indexed DB. The data stored in local storage is persistent and can be accessed by the website even if the user closes the browser or navigates to another page. It is a great way for websites to store user-specific data, as it is secure, reliable, and fast. It is also a great way for developers to store data that does not need to be sent to the server with each request.

One of the key benefits of using local storage is its reliability. Unlike server-side storage, which can be affected by network outages or other server issues, local storage is stored locally on the user's machine, and so is not affected by these issues. Another advantage of local storage is its speed. Because the data is stored locally, it is accessed quickly, as there is no need to send requests to a server. This makes it ideal for storing data that needs to be accessed quickly, such as user preferences or session data. Local storage is also secure, as the data is stored on the user's machine and not on a server. This means that the data is not accessible by anyone other than the user, making it a good choice for storing sensitive information.

CHAPTER 3

PROPOSED SYSTEM

This chapter gives a small description about the proposed idea behind the development of our website

3.1 PROPOSED SYSTEM

The Student E-Learning Portal is an advanced digital solution designed to revolutionize the online learning experience for students and streamline educational administration. This comprehensive platform offers a range of features aimed at enhancing both the efficiency and effectiveness of online education.

At its core, the Student E-Learning Portal simplifies the course enrollment process. Students can browse available courses, review course details, and enroll with just a few clicks. Once enrolled, they gain immediate access to a centralized repository of learning materials, including lecture notes, multimedia content, and supplementary resources

The portal also streamlines assignment management. Students can submit their assignments directly through the platform, receive timely feedback from instructors, and track their submission status. Interactive features such as discussion forums and collaborative tools are integrated into the portal to foster a dynamic learning environment. Students can engage in academic discussions, participate in group projects, and collaborate with peers and instructors in real-time. This enhances the overall learning experience and encourages active participation. To support students' academic progress, the portal includes robust tracking and reporting tools. Students can view their grades, monitor their progress over time, and access personalized recommendations for improvement.

Overall, the Student E-Learning Portal is designed to provide an integrated and user-friendly platform that addresses the needs of modern learners. By automating routine tasks, centralizing access to resources, and fostering engagement, the portal enhances the efficiency and effectiveness of online education, contributing to a more productive and satisfying learning experience for students.

`

3.2 ADVANTAGES

- **Efficiency:** The Student E-Learning Portal streamlines several key processes, including course enrollment, material access, and assignment submission. By integrating these functions into a single platform, it reduces the need for manual interventions and simplifies administrative tasks.

- **Accessibility:** With the portal accessible from any location with internet access, students and educators can engage with the system from anywhere. This accessibility supports remote learning and ensures that students can manage their educational activities without being constrained by physical location or time limitations.

Flexibility: The portal offers significant flexibility by allowing students to manage their schedules, access learning materials, and submit assignments at their convenience. This adaptability accommodates diverse learning preferences and schedules, helping to meet the needs of a wide range of students.

- **Engagement:** By incorporating interactive features such as discussion forums and collaborative tools, the portal enhances student engagement. These features facilitate communication between students and instructors, support collaborative learning, and create a more interactive and engaging educational experience.

Progress Tracking: The portal provides comprehensive tools for tracking academic progress. Students can view their grades, monitor their performance over time, and receive personalized feedback and recommendations. This functionality helps students stay informed about their academic standing and make informed decisions about their learning strategies.

Security: Ensuring the security of user data is a fundamental aspect of the portal. Advanced security measures are implemented to protect personal and academic information from unauthorized access and breaches. By prioritizing data protection, the portal maintains a secure and trustworthy learning environment for all users.

CHAPTER 4

METHODOLOGIE S

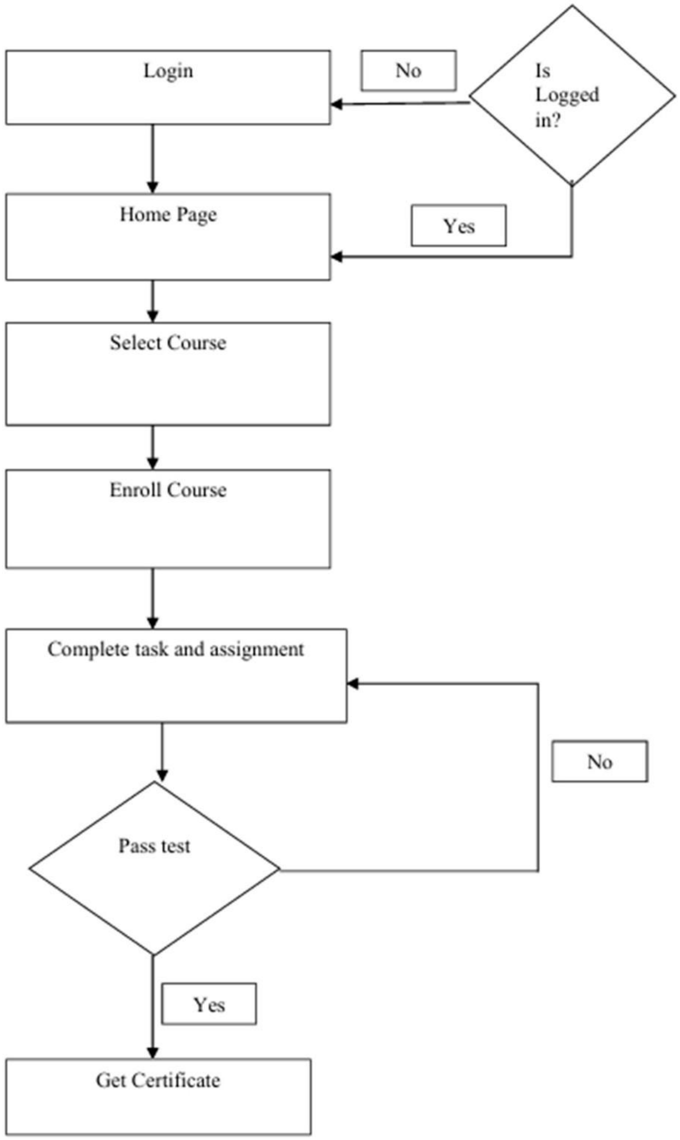


Fig 4.1.Process flow diagram

User Registration:

Staff members begin by registering on the Student E-Learning Portal, providing essential details such as email address, and password. Optional information, including phone number and department, can be added to enhance communication and personalization.

Role-Based Access:

Upon registration or login, users are assigned specific roles such as student, instructor, or administrator. These roles define their access permissions and functionalities within the portal, ensuring that each user interacts with the system according to their responsibilities.

CourseExploration:

Users can explore available courses and learning materials by browsing through categories or using the search functionality. This feature allows students to easily find and enroll in courses that match their interests and academic goals.

Assignment Submission:

Students can submit their assignments through the portal. They can upload documents, track submission status, and receive feedback from instructors. The system manages these processes to ensure timely and accurate handling of academic work.

Progress Tracking:

The portal provides tools for tracking academic progress. Students can view their grades, monitor performance over time, and access detailed reports. This functionality helps students stay informed about their academic achievements and areas for improvement.

Notification:

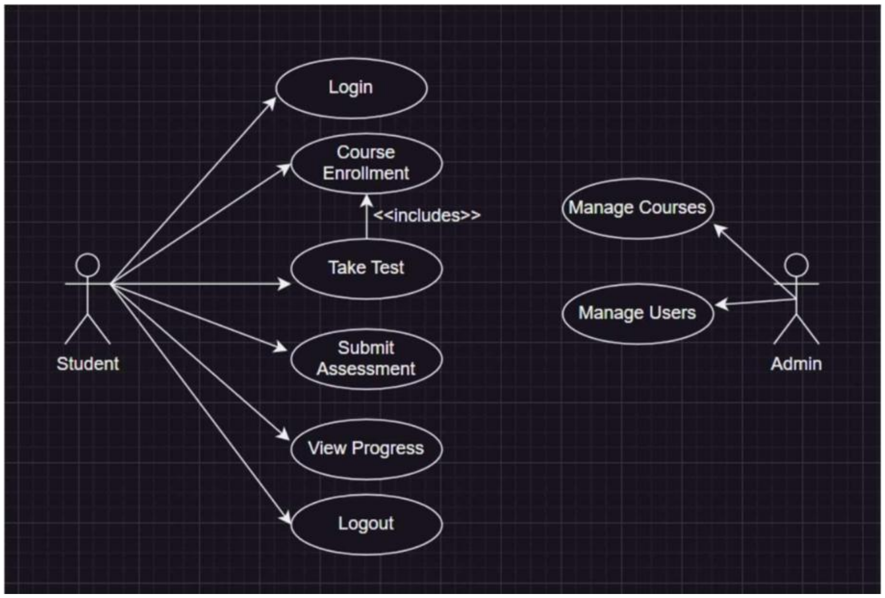
The system generates notifications to keep users updated on important events such as new course enrollments, assignment deadlines, and grade updates. Automated notifications ensure that students and instructors are aware of relevant changes and deadlines.

Feedback and Adjustments:

After interacting with the portal, users can provide feedback on their experience. This feedback helps in refining the system and making necessary adjustments to improve usability and effectiveness.

Use Case Diagram:

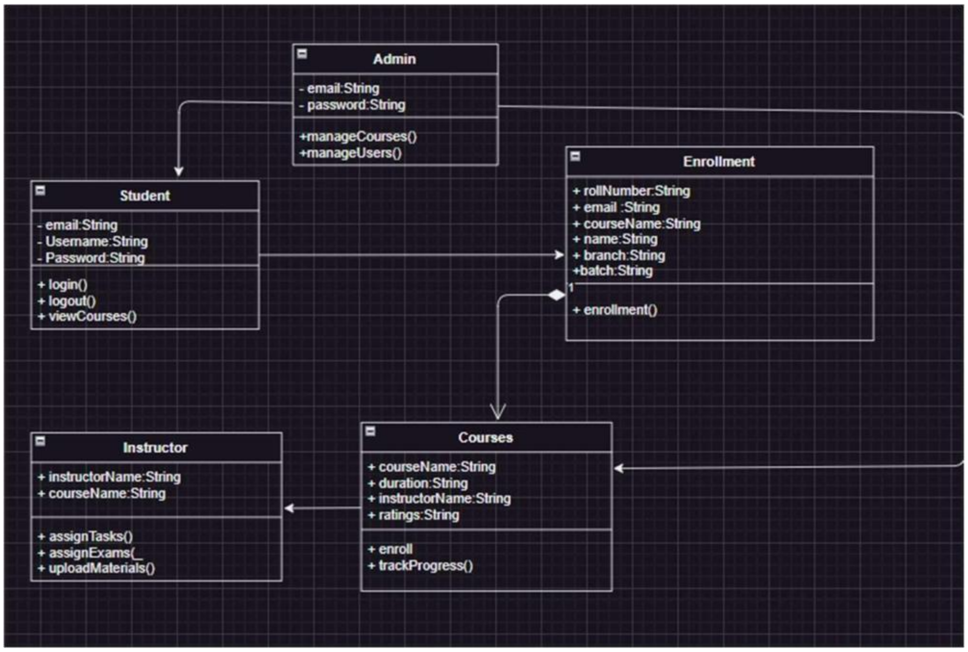
The use case diagram illustrates the interactions within the Student E-Learning Portal, depicting various user roles including Students, Instructors, Administrators, and IT Support. Each role has specific functionalities such as enrolling in courses, submitting assignments, managing courses, and handling technical support. The diagram highlights interaction points where roles perform tasks like course exploration, assignment submission, and progress tracking. It effectively showcases the workflow and demonstrates how different users interact with the system to manage their educational activities and support a seamless learning experience.



4.2 Use Case Diagram

Class Diagram:

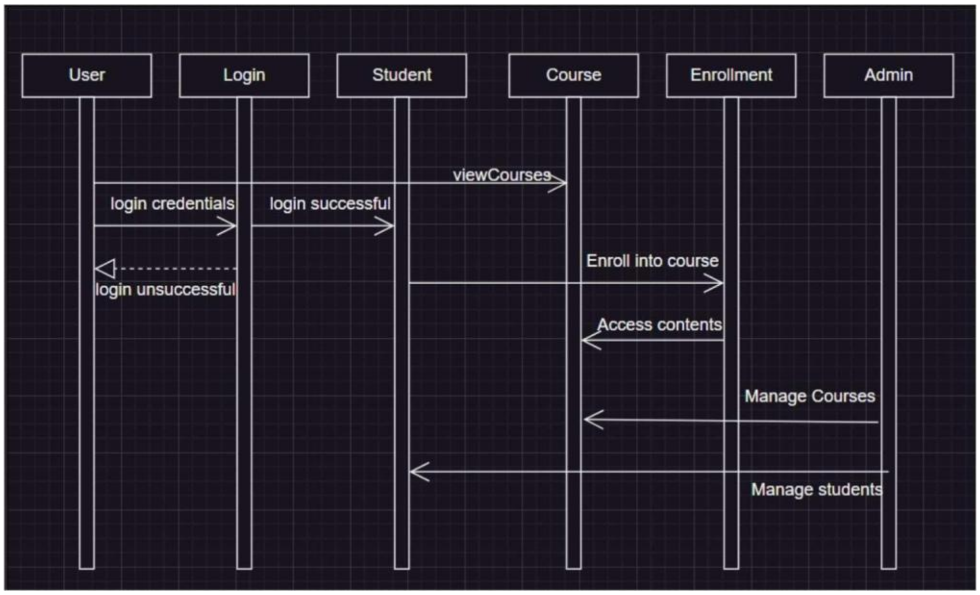
The class diagram outlines the architecture of the Student E-Learning Portal, detailing the roles and their specific functionalities within the system. Key classes include Student, Instructor, Administrator, IT Support, and Guest, each interacting with shared entities such as **User, Course, Assignment, Material,** and **Notification**. This diagram highlights methods for managing user accounts, course enrollments, assignment submissions, and notifications, illustrating how each role contributes to the portal's operations and how they interact with the core components to facilitate a seamless learning experience..



4.3 Class Diagram

Sequence Diagram:

The Room Booking Sequence Diagram illustrates the process for an academic staff member to request a room booking. It shows the interaction between the User, Academic Staff, System, and Resource components to check room availability and confirm the booking if available.



4.4 Sequence Diagram

CHAPTER 5

IMPLEMENTATION AND RESULT

This chapter gives a description about the output that we produced by developing the website of our idea.

5.1 LOGIN

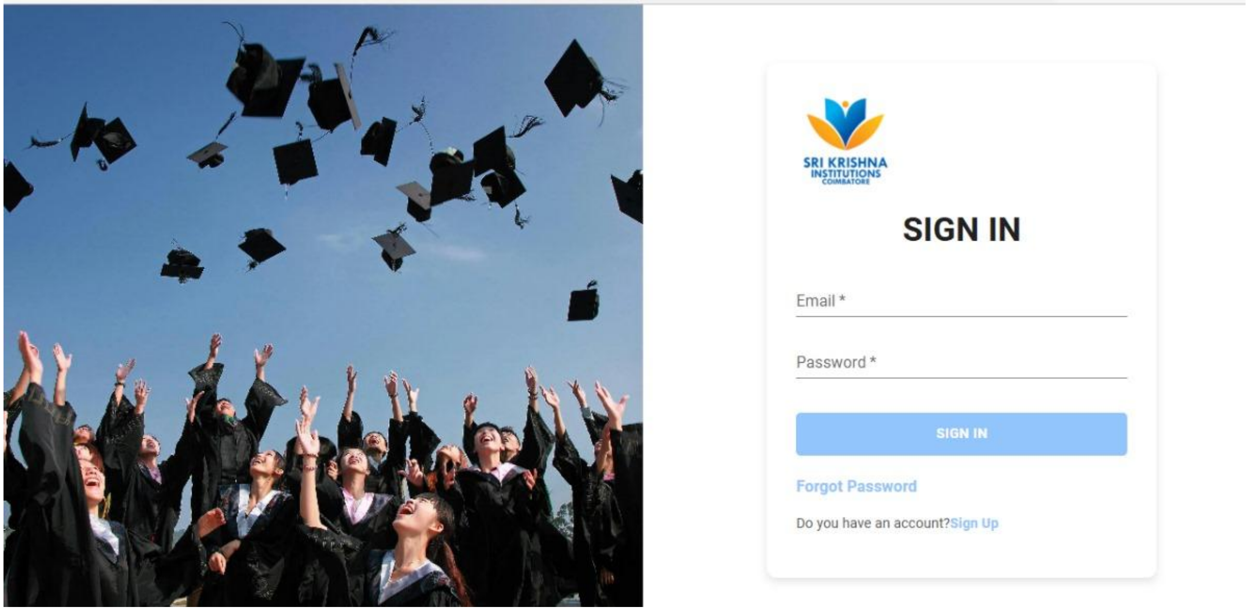


Fig 5.1 LOGIN PAGE

5.1 USER REGISTER

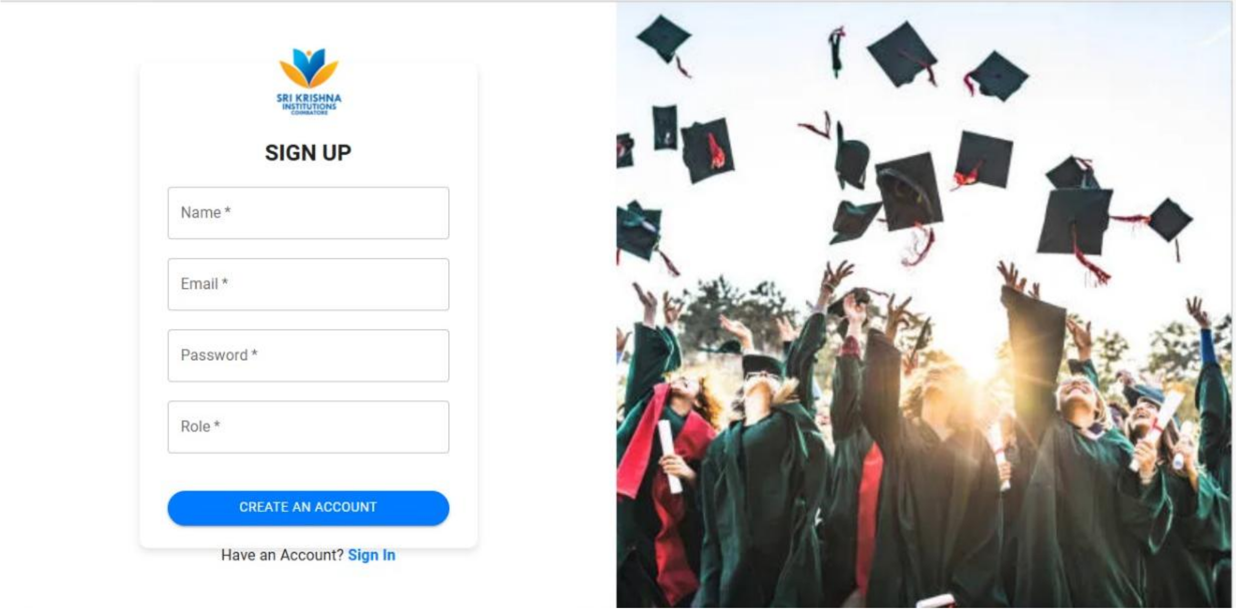
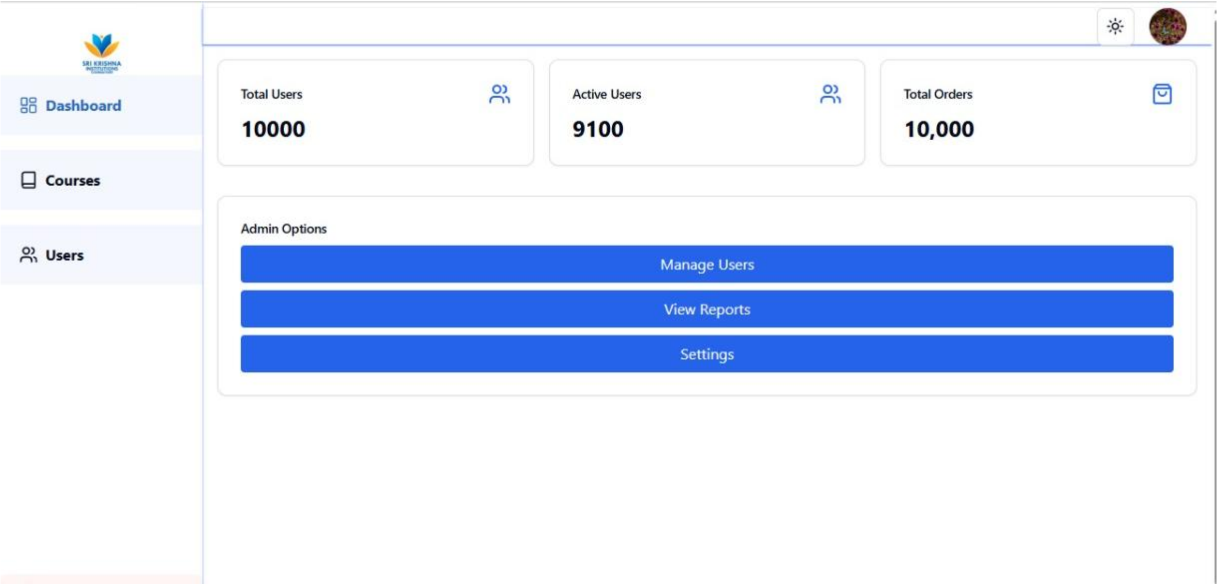


Fig 5.2 SIGN UP PAGE

5.2 DASHBOARD

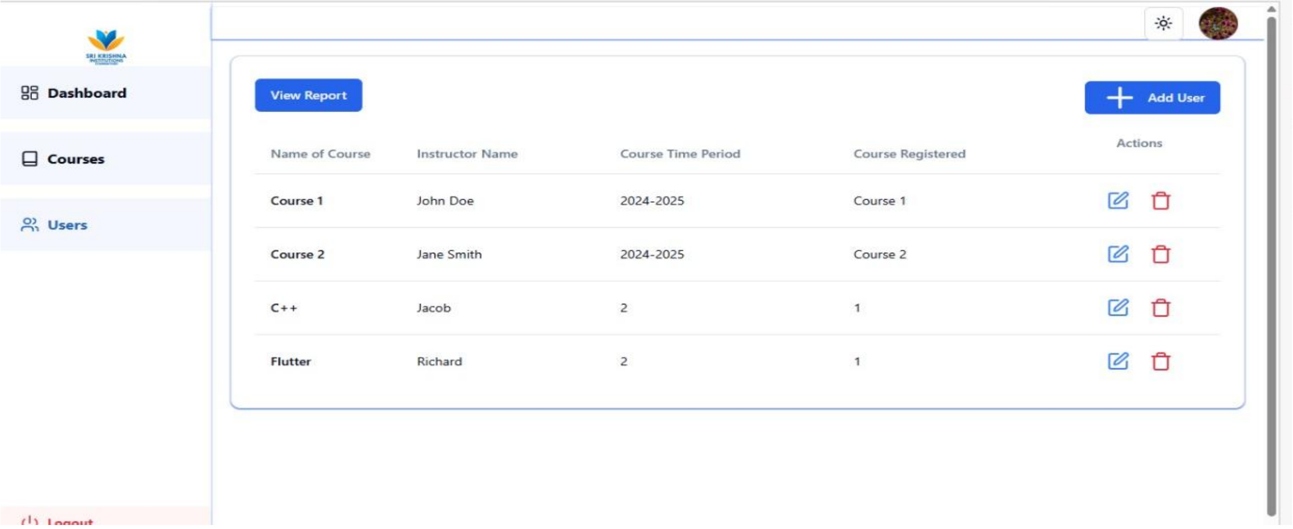


Total Users 10000

Active Users 9100

Total Orders 10,000

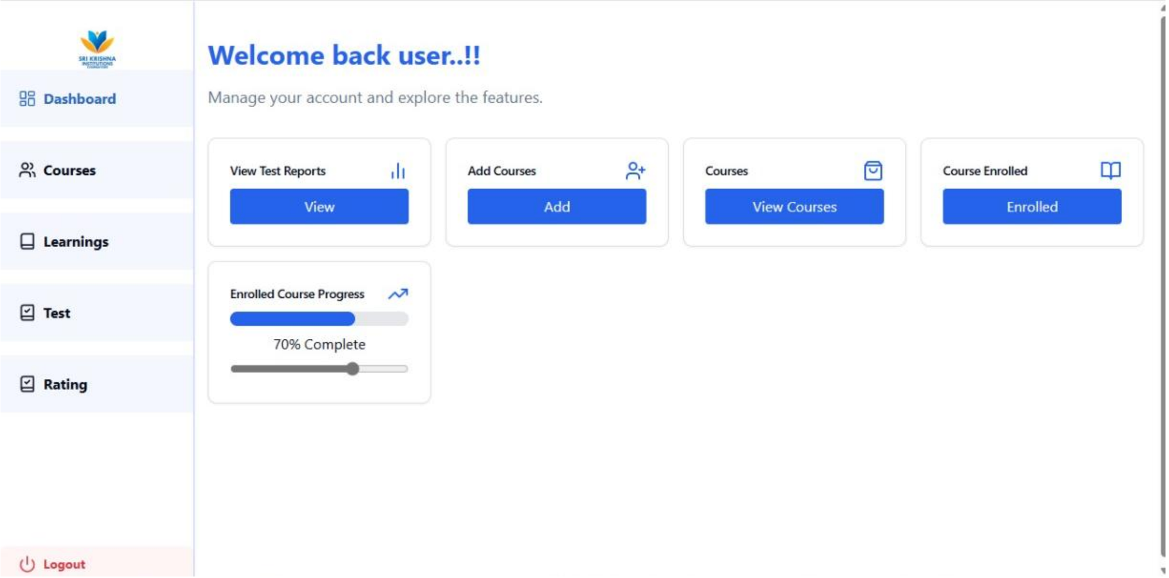
Fig 5.3 Admin Dashboard



Name of Course	Instructor Name	Course Time Period	Course Registered	Actions
Course 1	John Doe	2024-2025	Course 1	
Course 2	Jane Smith	2024-2025	Course 2	
C++	Jacob	2	1	
Flutter	Richard	2	1	

Fig 5.4 Admin Dashboard User Details

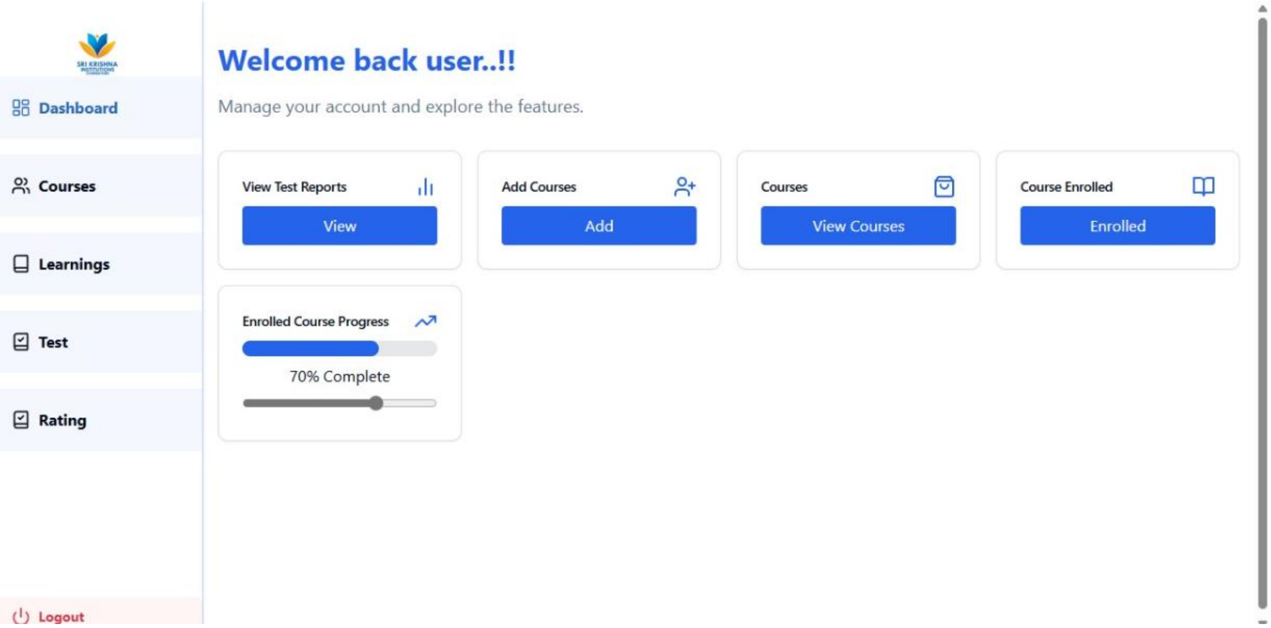
5.3 USER DASHBOARD



Welcome back user..!!

Manage your account and explore the features.

Fig 5.5 Tutor Dashboard

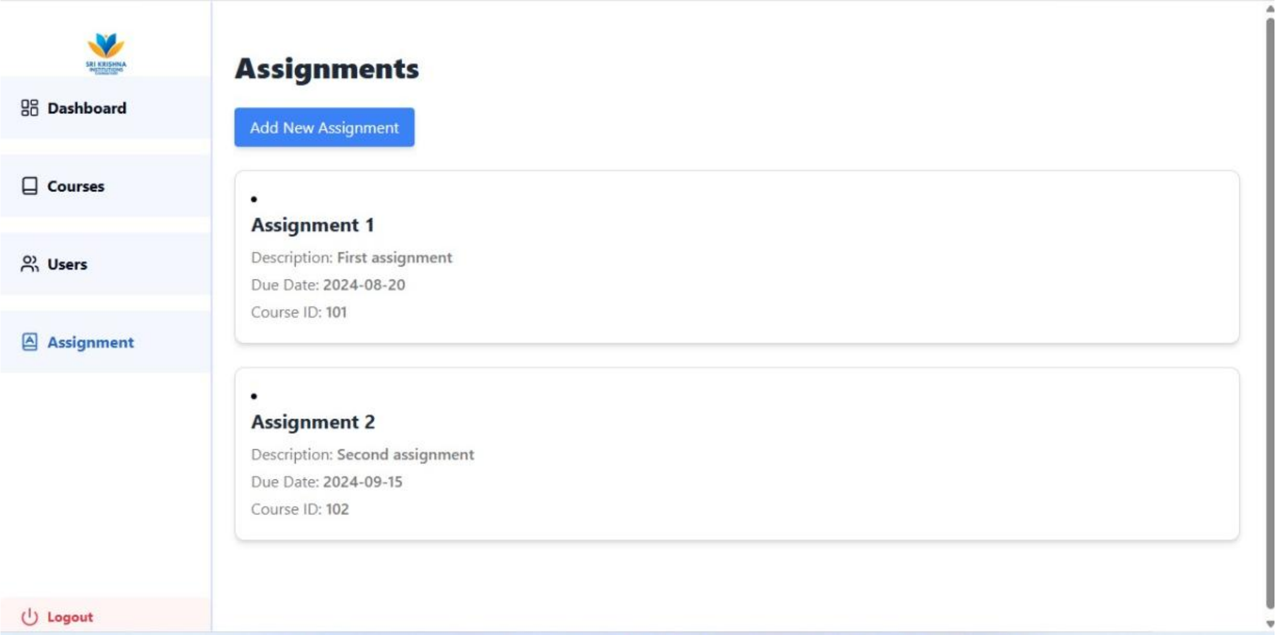


Welcome back user..!!

Manage your account and explore the features.

Fig 5.6 Student Dashboard

5.4 ASSIGNMENT:



Assignments

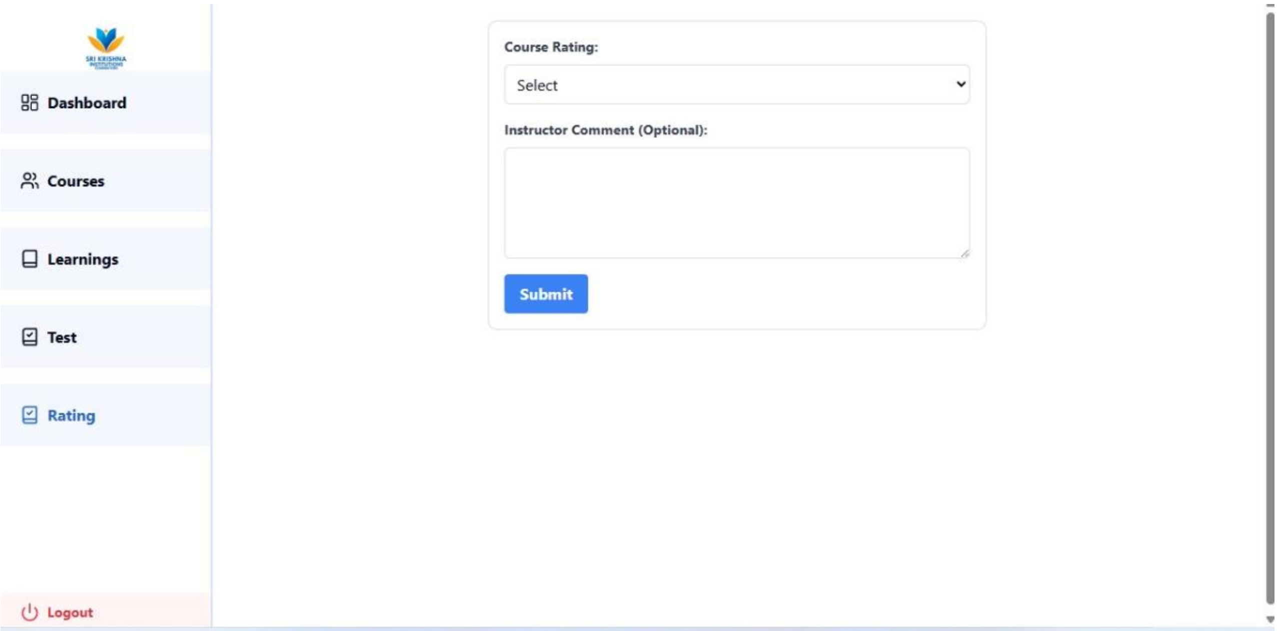
Add New Assignment

- **Assignment 1**
  Description: First assignment
  Due Date: 2024-08-20
  Course ID: 101

- **Assignment 2**
  Description: Second assignment
  Due Date: 2024-09-15
  Course ID: 102

Fig 5.6 Assignment

5.5 RATING:



Course Rating: Select

Instructor Comment (Optional):

Submit

CODING:

FRONTEND CODING:

App.js:

```
     const App = () => {
      return (
    <BrowserRouter>
      <Routes>
        <Route element={<HomeLayout />}>
          <Route path='/' element={<Home />} />
          <Route path='/signup' element={<Register />} />
          <Route path='/signin' element={<Login />} />
          <Route path="/courses" element={<CourseList />} />
          <Route path="/course/:id" element={<CoursesDetails />} />
        </Route>
        <Route element={<UserLayout />}>
          <Route path='/dashboard' element={<UserDashboard />} />
        </Route>
        <Route element={<AdminLayout />}>
          <Route path='/admin/dashboard' element={<AdminDashboard />} />
        </Route>
        <Route element={<TutorLayout />}>
          <Route path='/tutor/dashboard' element={<TutorDashboard />} />
        </Route>

      </Routes>
    </BrowserRouter>
  );
};

export default App;
```

Home.js

```
            const Home = () => {
 return (
  <div>
   <Typography variant="h4" gutterBottom align="center" style={{ marginBottom:
'30px' }} className='text-black dark:text-white'>
    Courses
```

```
      </Typography>
      <div className='card-wrap grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 gap-2
ml-11'>
        <div className='card'>
          <div className='set-image'>
            <img src={img9} alt="No image" />
          </div>
          <div className='set-text mt-6'>
            HTML, or HyperText Markup Language, is the standard language used to
create and design web pages. It structures content on the web by using various tags
and elements to describe headings, paragraphs, links, images, and more. HTML
forms the backbone of web content and is essential for web development.
          </div>
          <div className='ribbon-wrapper-1'>
            <div className='ribbon-1'>Ribbon</div>
          </div>
          <Link to="/courses">
            <button type="submit" className='bg-blue-300 h-[5vh] w-[7vw] border-r-2
rounded-lg mt-5 border-none dark:bg-teal-800 dark:text-white text-black'>
              Go To Course
            </button>
          </Link>
        </div>
        <div className='card'>
          <div className='set-image'>
            <img src={img2} alt="No image" />
          </div>
          <div className='set-text mt-6'>
            HTML, or HyperText Markup Language, is the standard language used to
create and design web pages. It structures content on the web by using various tags
and elements to describe headings, paragraphs, links, images, and more. HTML
forms the backbone of web content and is essential for web development.
          </div>
          <div className='ribbon-wrapper-2'>
            <div className='ribbon-2'>Ribbon</div>
          </div>
          <Link to="/courses">
            <button type="submit" className='bg-blue-300 h-[5vh] w-[7vw] border-r-2
rounded-lg mt-5 border-none dark:bg-teal-800 dark:text-white text-black'>
            Go To Course
            </button>
          </Link>
```

```
</div>
<div className='card'>
  <div className='set-image'>
    <img src={img1} alt="No image" />
  </div>
  <div className='set-text mt-6'>
    HTML, or HyperText Markup Language, is the standard language used to create and design web pages. It structures content on the web by using various tags and elements to describe headings, paragraphs, links, images, and more. HTML forms the backbone of web content and is essential for web development.
  </div>
  <div className='ribbon-wrapper-3'>
    <div className='ribbon-3'>Ribbon</div>
  </div>
  <Link to="/courses">
    <button type="submit" className='bg-blue-300 h-[5vh] w-[7vw] border-r-2 rounded-lg mt-5 border-none dark:bg-teal-800 dark:text-white text-black'>
    Go To Course
    </button>
  </Link>
</div>
<div className='card'>
  <div className='set-image'>
    <img src={img3} alt="No image" />
  </div>
  <div className='set-text'>
    HTML, or HyperText Markup Language, is the standard language used to create and design web pages. It structures content on the web by using various tags and elements to describe headings, paragraphs, links, images, and more. HTML forms the backbone of web content and is essential for web development.
  </div>
  <div className='ribbon-wrapper-4'>
    <div className='ribbon-4'>Ribbon</div>
  </div>
  <Link to="/courses">
    <button type="submit" className='bg-blue-300 h-[5vh] w-[7vw] border-r-2 rounded-lg mt-5 border-none dark:bg-teal-800 dark:text-white text-black'>
    Go To Course
    </button>
  </Link>
</div>
<div className='card'>
```

```
          <div className='set-image'>
            <img src={img4} alt="No image" />
          </div>
          <div className='set-text'>
            HTML, or HyperText Markup Language, is the standard language used to
create and design web pages. It structures content on the web by using various tags
and elements to describe headings, paragraphs, links, images, and more. HTML
forms the backbone of web content and is essential for web development.
          </div>
          <div className='ribbon-wrapper-5'>
            <div className='ribbon-5'>Ribbon</div>
          </div>
          <Link to="/courses">
            <button type="submit" className='bg-blue-300 h-[5vh] w-[7vw] border-r-2
rounded-lg mt-5 border-none dark:bg-teal-800 dark:text-white text-black'>
            Go To Course
            </button>
          </Link>
        </div>
        <div className='card'>
          <div className='set-image'>
            <img src={img5} alt="No image" />
          </div>
          <div className='set-text'>
            HTML, or HyperText Markup Language, is the standard language used to
create and design web pages. It structures content on the web by using various tags
and elements to describe headings, paragraphs, links, images, and more. HTML
forms the backbone of web content and is essential for web development.
          </div>
          <div className='ribbon-wrapper-6'>
            <div className='ribbon-6'>Ribbon</div>
          </div>
          <div>
            <button type="submit" className='bg-blue-300 h-[5vh] w-[7vw] border-r-2
rounded-lg text-black mt-5 border-none dark:bg-teal-800 dark:text-white'>
            Go To Course
            </button>
          </div>
        </div>
        <div className='card'>
          <div className='set-image'>
            <img src={img6} alt="No image" />
```

```
      </div>
      <div className='set-text'>
        HTML, or HyperText Markup Language, is the standard language used to
create and design web pages. It structures content on the web by using various tags
and elements to describe headings, paragraphs, links, images, and more. HTML
forms the backbone of web content and is essential for web development.
      </div>
      <div className='ribbon-wrapper-7'>
        <div className='ribbon-7'>Ribbon</div>
      </div>
      <Link to="/courses">
        <button type="submit" className='bg-blue-300 h-[5vh] w-[7vw] border-r-2
rounded-lg mt-5 border-none dark:bg-teal-800 dark:text-white text-black'>
        Go To Course
        </button>
      </Link>
    </div>
    <div className='card'>
      <div className='set-image'>
        <img src={img7} alt="No image" />
      </div>
      <div className='set-text'>
        HTML, or HyperText Markup Language, is the standard language used to
create and design web pages. It structures content on the web by using various tags
and elements to describe headings, paragraphs, links, images, and more. HTML
forms the backbone of web content and is essential for web development.
      </div>
      <div className='ribbon-wrapper-8'>
        <div className='ribbon-8'>Ribbon</div>
      </div>
      <Link to="/courses">
        <button type="submit" className='bg-blue-300 h-[5vh] w-[7vw] border-r-2
rounded-lg mt-5 border-none dark:bg-teal-800 dark:text-white text-black'>
        Go To Course
        </button>
      </Link>
    </div>
    <div className='card'>
      <div className='set-image'>
        <img src={img8} alt="No image" />
      </div>
      <div className='set-text'>
```

HTML, or HyperText Markup Language, is the standard language used to create and design web pages. It structures content on the web by using various tags and elements to describe headings, paragraphs, links, images, and more. HTML forms the backbone of web content and is essential for web development.

```
          </div>
          <div className='ribbon-wrapper-9'>
            <div className='ribbon-9'>Ribbon</div>
          </div>
          <Link to="/courses">
            <button type="submit" className='bg-blue-300 h-[5vh] w-[7vw] border-r-2 rounded-lg mt-5 border-none dark:bg-teal-800 dark:text-white text-black'>
            Go To Course
            </button>
          </Link>
        </div>
      </div>
    </div>
  );
};

export default Home;
```

Login.js

```
function Login() {
  const navigate = useNavigate();
  const emailRef = useRef(null);
  const passwordRef = useRef(null);

  const handleLogin = async (e) => {
    e.preventDefault();
    try {
      const res = await authService.SignIn(emailRef.current.value, passwordRef.current.value);
      console.log(res.data);

      if (res.status === 200) {
        authService.setToken(res.data.accessToken);
        const userRole = res.data.role;
        const userId = res.data.id;
        localStorage.setItem('userId',userId);
        console.log(User role: ${userRole});
        console.log(User ID: ${userId});

        // Store userId and role in local storage
        localStorage.setItem('userId', userId);
        localStorage.setItem('userRole', userRole);

        toast.success('Welcome');

        if (userRole === 'INSTRUCTOR') {
          navigate('/tutor/dashboard');
        } else if (userRole === 'STUDENT') {
          navigate('/dashboard');
        } else if (userRole === 'ADMIN') {
          navigate('/admin/dashboard');
        } else {
          toast.error('Invalid user role');
        }
      } else {
        toast.error('Login failed');
      }
    } catch (error) {
      console.error('Login error:', error);
      toast.error('Login failed. Please check your credentials and try again.');
    }
  };

  const paperStyle = {
    padding: 25,
    height: '70vh',
    width: 400,
    margin: '100px auto',
  };

  const avatarStyle = {
    backgroundColor: '#93C5FD',
  };

  const buttonStyle = {
    marginTop: 20,
    backgroundColor: '#93C5FD',
    color: '#fff',
    margin: '30px 0',
  };

  const typo = {
    margin: '10px 0',
  };

  return (
    <>
    <Grid container justifyContent="center" alignItems="center" style={{ height:
'100vh' }}>
      <Paper elevation={10} style={paperStyle}>
        <Grid align='center'>
          <Avatar style={avatarStyle}>
            <LockOutlinedIcon />
          </Avatar>
          <h2 style={{ marginBottom: 35 }}>SIGN IN</h2>
        </Grid>
        <form onSubmit={handleLogin}>
          <TextField
            id="email"
            label="Email"
            type="email"
            variant="standard"
            fullWidth
            required
            style={{ marginBottom: 20 }}
            inputRef={emailRef}
          />
          <TextField
            id="password"
            label="Password"
            type="password"
            variant="standard"
            fullWidth
            required
            style={{ marginBottom: 20 }}
            inputRef={passwordRef}
          />
          <Button type="submit" fullWidth style={buttonStyle}>
            SIGN IN
          </Button>
    </form>
    <Typography>
     <Link to="#" style={{ textDecoration: 'none' }}>
      Forgot Password
     </Link>
    </Typography>
    <Typography style={typo}>
     Do you have an account?
     <Link to="/signup" style={{ textDecoration: 'none' }}>
      Sign Up
     </Link>
    </Typography>
   </Paper>
  </Grid>
   <ToastContainer/>
  </>
 );
}

export default Login;
```

Register.js

```
function Register() {
 const [registerData, setRegisterData] = useState({
  name: '',
  email: '',
  password: '',
  role: '',
 });
 const navigate = useNavigate();

 const paperStyle = {
  padding: 20,
  height: '80vh',
  width: 400,
  margin: '72px auto',
 };

 const avatarStyle = {
  backgroundColor: '#93C5FD',
 };
```

```
  const buttonStyle = {
    marginTop: 20,
    backgroundColor: '#93C5FD',
    color: '#fff',
    margin: '15px 0',
  };

  const typo = {
    margin: '10px 0',
  };

  const handleChange = (e) => {
    setRegisterData({ ...registerData, [e.target.id]: e.target.value });
  };

  const handleSubmit = async (e) => {
    e.preventDefault();
    const res = await SignUp(registerData.name, registerData.email,
registerData.password, registerData.role);

    if (res.message === 'An error occurred during registration.') {
      toast.error("Signup failed");
    } else {
      toast.success(res.message || "Registration Success");
      setTimeout(() => {
        navigate('/signin');
      }, 1000);
    }
  };

  return (
    <>
      <Grid>
        <Paper elevation={10} style={paperStyle}>
          <Grid align='center'>
            <Avatar style={avatarStyle}>
              <LockOutlinedIcon />
            </Avatar>
            <h2 style={{ marginBottom: 35 }}>SIGN UP</h2>
            <p>Fill this form to create an Account</p>
          </Grid>
```

```
<form onSubmit={handleSubmit}>
  <TextField
    id="name"
    label="Name"
    type="text"
    variant="standard"
    fullWidth
    required
    style={{ marginBottom: 20 }}
    onChange={handleChange}
  />
  <TextField
    id="email"
    label="Email"
    type="email"
    variant="standard"
    fullWidth
    required
    style={{ marginBottom: 20 }}
    onChange={handleChange}
  />
  <TextField
    id="password"
    label="Password"
    type="password"
    variant="standard"
    fullWidth
    required
    style={{ marginBottom: 20 }}
    onChange={handleChange}
  />
  <TextField
    id="role"
    label="Role"
    type="text"
    variant="standard"
    fullWidth
    required
    style={{ marginBottom: 20 }}
    onChange={handleChange}
  />
  <Button type="submit" fullWidth style={buttonStyle}>
```

```
       Create an Account
      </Button>
      <Typography style={typo}>
       Have an Account..
       <Link to="/signin" style={{ textDecoration: 'none' }}>
        Sign In
       </Link>
      </Typography>
     </form>
    </Paper>
   </Grid>
   <ToastContainer />
  </>
 );
}

export default Register;
```

Navbar.js

```
const Navbar = () => {
 const NavLinks = [
  {
   title: "Home",
   path: "/"
  },
  {
   title: "Sign Up",
   path: "/signup"
  },
  {
   title: "Sign In",
   path: "/signin"
  },
  {
   title: "Courses",
   path: "/courses"
  }
 ];

 const linkStyle = {
  textDecoration: 'none',
    color: 'black',
  };

  const listItemStyle = {
    margin: 0,
    padding: 0,
    listStyleType: 'none',
  };

  return (
    <div className="fixed top-0 left-0 w-full h-[8vh] flex flex-row justify-center items-center shadow-sm bg-blue-300 dark:bg-teal-800 z-50">
      <div className="w-1/4 h-full text-black font-bold flex justify-start items-center text-lg dark:text-white">
        <LibraryBig className='h-6 w-6 mr-2 text-black dark:text-white' />
        LEARNERS
      </div>
      <div className='w-2/4 h-full font-bold flex flex-row justify-end items-center gap-9'>
        {
          NavLinks.map((links, index) => (
            <li key={index} style={listItemStyle}>
              <NavLink
                to={links.path}
                style={linkStyle}
                className={({ isActive }) =>
                  text-black dark:text-white ${isActive ? 'font-bold' : "}
                }
              >
                {links.title}
              </NavLink>
            </li>
          ))
        }
        <div>
           <ModeToggle />
        </div>
      </div>
    </div>
  );
};
```

```
export default Navbar;
```

Footer.js

```
const Footer = () => {
  return (
    <div>
      {/* Contact Header */}
      <header className="text-center my-5">
        <h4>Contact Us</h4>
      </header>

      {/* Main Contact Content */}
      <main>
        <div className="container py-5">
          <div className="row g-5">
            {/* Contact Information Block */}
            <div className="col-xl-6">
              <div className="row g-4">
                <div className="col-md-6">
                  <div className="bg-light p-3 rounded shadow-sm">
                    <div className="d-flex align-items-center mb-2">
                      <EmailIcon className="h3 pe-2" />
                      <span className="h5">Email</span>
                    </div>
                    <span>example@domain.com</span>
                  </div>
                </div>
                <div className="col-md-6">
                  <div className="bg-light p-3 rounded shadow-sm">
                    <div className="d-flex align-items-center mb-2">
                      <PhoneIcon className="h3 pe-2" />
                      <span className="h5">Phone</span>
                    </div>
                    <span>+0123456789, +9876543210</span>
                  </div>
                </div>
                <div className="col-md-12 mt-4">
                  <div className="bg-light p-3 rounded shadow-sm">
                    <div className="d-flex align-items-center mb-2">
                      <LocationOnIcon className="h3 pe-2" />
                      <span className="h5">Office Location</span>
                    </div>
                    <span>#007, Street name, Bigtown BG23 4YZ, England</span>
                  </div>
                </div>
                <div className="col-md-12 mt-4">
                  <iframe
                    className="w-100 rounded"
                    width="100%"
```

```
          height="345"

          frameBorder="0"

          scrolling="no"

          marginHeight="0"

          marginWidth="0"

src="https://maps.google.com/maps?width=100%25&amp;height=300&amp;hl=en&amp;q=1%20Grafton%20Street,%20Dublin,%20Ireland+()&amp;t=&amp;z=14&amp;ie=UTF8&amp;iwloc=B&amp;output=embed"

          title="Google Map"

         >

          <a href="https://www.maps.ie/distance-area-calculator.html">measure
acres/hectares on map</a>

         </iframe>

        </div>

       </div>

      </div>



      {/* Contact Form Block */}

      <div className="col-xl-6">

       <h2 className="pb-4">Leave a Message</h2>

       <form>

        <div className="row g-4">

         <div className="col-12 col-md-6 mb-3">

          <label htmlFor="fname" className="form-label">First Name</label>

          <input type="text" className="form-control" id="fname"
```

```
placeholder="John"/>
          </div>
          <div className="col-12 col-md-6 mb-3">
            <label htmlFor="lname" className="form-label">Last Name</label>
            <input type="text" className="form-control" id="lname"
placeholder="Doe"/>
          </div>
        </div>
        <div className="mb-3">
          <label htmlFor="email" className="form-label">Email</label>
          <input type="email" className="form-control" id="email"
placeholder="name@example.com"/>
        </div>
        <div className="mb-3">
          <label htmlFor="phone" className="form-label">Phone</label>
          <input type="tel" className="form-control" id="phone"
placeholder="+1234567890"/>
        </div>
        <div className="mb-3">
          <label htmlFor="country" className="form-label">Country</label>
          <select className="form-select" id="country">
            <option value="1">USA</option>
            <option value="2">Non USA</option>
          </select>
        </div>
        <div className="mb-3">
         <label htmlFor="message" className="form-label">Message</label>
         <textarea className="form-control" id="message"
rows="3"></textarea>
        </div>
        <button type="submit" className="btn btn-dark">Send
Message</button>
       </form>
      </div>
     </div>
    </div>
   </main>

   {/* Footer Block */}
   {/* <footer id="site-footer"> */}
    <div className="bg-blue-300 dark:bg-teal-800 dark:text-white py-5">
     <div className="container">
      <div className="row">
       <div className="col-md-6 col-xl-3 col-sm-12 mb-4">
        <h5 className="pb-3"><i className="fa-solid fa-link pe-1"></i>
Important Links</h5>
        <ul className="list-unstyled">
         <li><a href="#" className="text-decoration-none
link-secondary">About Us</a></li>
         <li><a href="#" className="text-decoration-none
link-secondary">Privacy Policy</a></li>
```

```
          <li><a href="#" className="text-decoration-none link-secondary">Terms of Services</a></li>
        </ul>
      </div>
      <div className="col-md-6 col-xl-3 col-sm-12 mb-4">
        <h5 className="pb-3"><LocationOnIcon className="pe-1" /> Our Location</h5>
        <span className="text-secondary">
          Milannagar Bazar<br />
          Tamluk, East Medinipore, West Bengal<br />
          720001, India<br />
        </span>
      </div>
      <div className="col-md-6 col-xl-3 col-sm-12 mb-4">
        <h5 className="pb-3"><i className="fa-solid fa-paper-plane pe-1"></i> Stay Updated</h5>
        <form>
          <input type="email" className="w-100 mb-2 form-control" placeholder="Email ID" />
          <button className="w-100 btn btn-dark">Subscribe Now</button>
        </form>
      </div>
    </div>
  </div>
</div>
<div className="bg-blue-300  dark:bg-teal-800 py-3">
```

```
     <div className="container">
      <div className="row">
       <div className="col-md-6 col-sm-12 mb-3 mb-md-0">
        <ul className="list-inline mb-0">
         <li className="list-inline-item"><a className="btn btn-outline-secondary" href="#"><FacebookIcon /></a></li>
         <li className="list-inline-item"><a className="btn btn-outline-secondary" href="#"><YouTubeIcon /></a></li>
         <li className="list-inline-item"><a className="btn btn-outline-secondary" href="#"><TwitterIcon /></a></li>
         <li className="list-inline-item"><a className="btn btn-outline-secondary" href="#"><LinkedInIcon /></a></li>
         <li className="list-inline-item"><a className="btn btn-outline-secondary" href="#"><GitHubIcon /></a></li>
        </ul>
       </div>
       <div className="col-md-6 col-sm-12 text-md-end text-sm-start">
        <span className="text-black">Copyright &copy; 2024</span>
       </div>
      </div>
     </div>
    </div>
   {/* </footer> */}
  </div>
 );
}
```

```
export default Contactus; <a
href="#" target="_blank"
className="h-8 w-8 rounded-md hover:bg-primary/50 hover:text- background flex
justify-center items-center"
>
<Facebook className="h-6 w-6" />
</a>
<a
href="#" target="_blank"
className="h-8 w-8 rounded-md hover:bg-primary/50 hover:text- background flex
justify-center items-center" >
<Twitter className="h-6 w-6" />
</a>
<a
href="#" target="_blank"
className="h-8 w-8 rounded-md hover:bg-primary/50 hover:text- background flex
justify-center items-center"
>
<Instagram className="h-6 w-6" />
</div>


</>
);};

export default Footer;
```

BACKEND CODING:

Config:

```
@Configuration @RequiredArgsConstructor public class ApplicationConfig {
private final UserRepo userRepo;
@Bean

public UserDetailsService userDetailsService() {

return username -> userRepo.findByEmail(username).orElseThrow(() -> new
UsernameNotFoundException(username));
}

@Bean

public AuthenticationProvider authenticationProvider() { DaoAuthenticationProvider
authProvider = new DaoAuthenticationProvider();
authProvider.setUserDetailsService(userDetailsService());
authProvider.setPasswordEncoder(passwordEncoder());
return authProvider;

}

@Bean

public AuthenticationManager authenticationManager(AuthenticationConfiguration
authenticationConfiguration) throws Exception {
return authenticationConfiguration.getAuthenticationManager();

}

@Bean

public PasswordEncoder passwordEncoder()

{

return new BCryptPasswordEncoder();

}

}
```

JwtAuthenticationFilter:

```
@Component @RequiredArgsConstructor
public class JwtAuthenticationFilter extends OncePerRequestFilter { private final
JwtService jwtService;
private final UserDetailsService userDetailsService;


@Override

protected void doFilterInternal( @NonNull HttpServletRequest request, @NonNull
HttpServletResponse response, @NonNull
FilterChain filterChain) throws ServletException, IOException { final String authHeader
= request.getHeader("Authorization"); final String jwtToken;
final String userEmail;

if(authHeader == null || !authHeader.startsWith("Bearer ")){ filterChain.doFilter(request,
response);
return;
}
```

JwtService:

```
@Service
public class JwtService {
private static final String SECRET_KEY =
"EbeEsh7VhXpHMAkLz7Xb3TYm7a4KLMlYn0Kr1NJEhTIOeU9HJsv3t2bMa5Ojoia
D";
public String extractUserName(String token) { return extractClaim(token,
Claims::getSubject);
}
public <T> T extractClaim (String token, Function<Claims, T> claimsResolver) { final
Claims claims = extractAllClaims(token);
return claimsResolver.apply(claims);
}
public String generateToken(UserDetails userDetails) { return generateToken(new
HashMap<>(), userDetails);
}
public String generateToken (Map<String, Object> extraClaims, UserDetails
userDetails) { return Jwts

.builder()
.setClaims(extraClaims)
.setSubject(userDetails.getUsername())
.setIssuedAt(new Date(System.currentTimeMillis()))
.setExpiration(new Date(System.currentTimeMillis() + (24 * 60 * 60 * 1000)))
.signWith(getSignInKey(), SignatureAlgorithm.HS256)
.compact();
}
public Boolean isTokenValid (String token, UserDetails userDetails) { final String
username = extractUserName(token);
return (username.equals(userDetails.getUsername()) && !isTokenExpired(token));
}
private Boolean isTokenExpired (String token) { return
extractExpiration(token).before(new Date());
}
private Date extractExpiration (String token) {
return extractClaim (token, Claims::getExpiration);
}
private Claims extractAllClaims (String token) { return Jwts
.parser()
.setSigningKey(getSignInKey())
.build()
.parseClaimsJws(token)
.getBody();
}
private Key getSignInKey(){
byte[] keyBytes = Decoders.BASE64.decode(SECRET_KEY); return
Keys.hmacShaKeyFor(keyBytes);
}
}

@Configuration
public class LogoutConfiguration { @Bean
public CustomLogoutHandler logoutHandler(TokenRepo tokenRepo, JwtService
jwtService)
```

```
{

return new CustomLogoutHandler(tokenRepo, jwtService);

}

@Bean

public LogoutSuccessHandler logoutSuccessHandler() { return new
CustomLogoutSuccessHandler();
}

}
```

SecurityConfiguration:

```
@Configuration @EnableWebSecurity @RequiredArgsConstructor
public class SecurityConfiguration {

private final JwtAuthenticationFilter jwtAuthFilter;

private final AuthenticationProvider authenticationProvider; private static final String[]
PublicEndPoints = {
"api/v1/auth/**"

};

@Bean

public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
http
.csrf(AbstractHttpConfigurer::disable)

.authorizeHttpRequests( authorize ->
authorize.requestMatchers(PublicEndPoints).permitAll().anyRequest().authenticated()


)

.sessionManagement(
session -> session.sessionCreationPolicy(SessionCreationPolicy.ALWAYS)

.maximumSessions(1)

)

.authenticationProvider(authenticationProvider)

.addFilterBefore(jwtAuthFilter, UsernamePasswordAuthenticationFilter.class); return
http.build();
}

}
```

Home page:

```
@Entity @AllArgsConstructor @NoArgsConstructor @Getter
@Setter @Builder
public class User implements UserDetails {


@Id

@GeneratedValue(strategy = GenerationType.SEQUENCE) private Long id;
private String name; private String email; private String password;
```

```
@Id

@GeneratedValue(strategy = GenerationType.SEQUENCE) private Long id;
private String name; private String email; private String password;
```

```
@JsonBackReference

@OneToMany(mappedBy = "user", cascade = CascadeType.ALL, fetch =
FetchType.LAZY)
private List<Token> token;
```

```
@JsonBackReference

@OneToMany(mappedBy = "user", cascade = CascadeType.ALL, fetch =
FetchType.LAZY)
private List<Token> token;
```

```
@Enumerated(EnumType.STRING) private Role role;
@Override

public Collection<? extends GrantedAuthority> getAuthorities() { return List.of(new
SimpleGrantedAuthority(role.name()));
}

@Override

public String getUsername() { return email;
}

@Override

public boolean isAccountNonExpired() { return true;
}


@Override

public boolean isAccountNonLocked() { return true;
}


@Override

public boolean isCredentialsNonExpired() { return true;
}


@Override

public boolean isEnabled() { return true;
}

}
```

```
@Getter @Setter @Builder
@NoArgsConstructor @AllArgsConstructor @Entity
public class Course { @Id
@GeneratedValue(strategy = GenerationType.IDENTITY) private int id;


private String title; private String category;
private String difficultyLevel; private String syllabus; private String schedule; private
String prerequisites; @ManyToOne
@JoinColumn(name = "user_id") private User user; @OneToMany(mappedBy =
"course") @JsonIgnore
private List<LearningMaterial> learningMaterials; @OneToMany(mappedBy =
"course") @JsonIgnore
private List<Assignment> assignments

}@JsonIgnore

   private List<LearningMaterial> learningMaterials;

   @OneToMany(mappedBy = "course")

   @JsonIgnore

   private List<Assignment> assignments;
   @OneToMany(mappedBy = "course")

   @JsonIgnore

   private List<SubModule> subModules;

   @OneToMany(mappedBy = "course")

   @JsonIgnore

   private List<CourseRating> ratings;

   public Course() {

   }

   public Course(Long id, String title, String category, String difficultyLevel, String syllabus, String
schedule,

         String prerequisites, boolean isApproved, float rating, User user, List<Enrollment> enrollments,


         List<LarningMaterial> learningMaterials, List<Assignment> assignments, List<SubModule>
subModules,
 String prerequisites, boolean isApproved, float rating, User user, List<Enrollment> enrollments,

      List<LarningMaterial> learningMaterials, List<Assignment> assignments, List<SubModule>
subModules,

String prerequisites, boolean isApproved, float rating, User user, List<Enrollment> enrollments,

      List<LarningMaterial> learningMaterials, List<Assignment> assignments, List<SubModule>
subModules,
  List<CourseRating> ratings) {
    this.id = id;
    this.title = title;
    this.category = category;

    this.difficultyLevel = difficultyLevel;

    this.syllabus = syllabus;

    this.schedule = schedule;

    this.prerequisites = prerequisites;

    this.isApproved = isApproved;

    this.rating = rating;

    this.user = user;

    this.enrollments = enrollments;

    this.learningMaterials = learningMaterials;
    this.assignments = assignments;
    this.subModules = subModules;
    this.ratings = ratings;
}

public Long getId() {
    return id;
}

public void setId(Long id) {
    this.id = id;
}
public String getTitle() {
    return title; }
public void setTitle(String title) {
    this.title = title;
}
public String getCategory() {
    return category;
```

```
}
public void setCategory(String category) {
   this.category = category;
}
public String getDifficultyLevel() {
   return difficultyLevel
}
public void setDifficultyLevel(String difficultyLevel) {

   this.difficultyLevel = difficultyLevel;
}
public String getSyllabus() {
   return syllabus;
}
public void setSyllabus(String syllabus) {
   this.syllabus = syllabus; }public String getSchedule() {
   return schedule;
}
public void setSchedule(String schedule) {
   this.schedule = schedule;
}
public String getPrerequisites() {
   return prerequisites;
}
public void setPrerequisites(String prerequisites) {
   this.prerequisites = prerequisites;
}
public boolean isApproved() {
   return isApproved;
}
public void setApproved(boolean isApproved) {
    this.isApproved = isApproved;
}
public float getRating() {
    return rating;
}
public void setRating(float rating) {
    this.rating = rating;

}
public User getUser() {
     return user;
}
public void setUser(User user) {
    this.user = user;
}
public List<Enrollment> getEnrollments() {
    return enrollments;
}
public void setEnrollments(List<Enrollment> enrollments) {
    this.enrollments = enrollments;
}
public List<LearningMaterial> getLearningMaterials() {
   return learningMaterials;

}
public void setLearningMaterials(List<LearningMaterial> learningMaterials) {
    this.learningMaterials = learningMaterials;
}
    this.isApproved = isApproved;
}
public float getRating() {
    return rating;
}
public void setRating(float rating) {
    this.rating = rating;

}
public User getUser() {
     return user;
}
public void setUser(User user) {
    this.user = user;
}
public List<Enrollment> getEnrollments() {
    return enrollments;
}
public void setEnrollments(List<Enrollment> enrollments) {
    this.enrollments = enrollments;
}
public List<LearningMaterial> getLearningMaterials() {
   return learningMaterials;

}
public void setLearningMaterials(List<LearningMaterial> learningMaterials) {
    this.learningMaterials = learningMaterials;
}



public List<Assignment> getAssignments() {
    return assignments;
}
public void setAssignments(List<Assignment> assignments) {
    this.assignments = assignments;
}
    public void setAssignments(List<Assignment> assignments) {
    this.assignments = assignments;
}
}

public void setAssignments(List<Assignment> assignments) {
    this.assignments = assignments;
}
}
public void setAssignments(List<Assignment> assignments) {
    this.assignments = assignments;
}   }
public void setAssignments(List<Assignment> assignments) {
    this.assignments = assignments;
}   }
public void setAssignments(List<Assignment> assignments) {
    this.assignments = assignments;
}
```

CHAPTER 6

CONCLUSION

This chapter tells about the conclusion that anyone can drive from the project and the learning we learnt by taking over this project.

6.1 CONCLUSION

In conclusion, the proposed Student E-Learning Portal is designed to enhance the online learning experience by streamlining course management, simplifying access to learning materials, and improving student engagement. This system is scalable, catering to institutions of varying sizes and educational needs. It facilitates efficient course enrollment, assignment submissions, and progress tracking while ensuring the security of user data. With features such as real-time notifications and interactive tools, the portal supports a dynamic learning environment and fosters better communication between students, instructors, and administrators. By integrating various functionalities into a single platform, the portal enhances operational efficiency, supports academic success, and contributes to a more engaging and effective learning experience. .

6.2 FUTURE SCOPE

Integration of AI and Machine Learning:

Implementing AI algorithms to provide personalized learning recommendations, predict course demand, and optimize student engagement based on individual performance and learning patterns.

Enhanced Security Measures:

Continuously improving security protocols protect user data and ensure compliance with data protection regulations, sensitive academic and personal information.

Mobile Application Development:

Developing a mobile app version of the portal to offer greater accessibility and convenience for students to manage their learning activities and stay updated on-the-go

Analytics and Reporting:

Creating advanced analytics tools to provide insights into student performance, course effectiveness, and engagement trends. This data will help institutions refine their educational strategies and offerings.

Integration with Learning Management Systems (LMS):

Integrating with existing LMS platforms to sync course registrations, materials, and progress, creating a cohesive educational experience across different systems.

Adaptive Exam Models:

Implementing adaptive learning technologies that adjust content difficulty and learning paths based on student performance, providing a more personalized and effective educational experience.

Collaborations with Exam Boards:

Partnering with content providers to offer a wider range of learning materials and resources through the portal, enhancing the breadth and depth of educational content available to students.

Feedback Mechanisms:

Incorporating robust feedback mechanisms for students and instructors to share their experiences and suggestions, facilitating continuous improvement and responsiveness to user needs.

CHAPTER 7

REFERENCES

1. https://www.geeksforgeeks.org/reactjs-introduction/?ref=lbp
2. https://github.com/MaXNeO/SDP_UseCase_LMS_RelationShip_DEMO
3. https://docs.spring.io/spring-boot/reference/index.html
4. https://react.dev/learn/referencing-values-with-refs
5. https://github.com/MaX-NeO/Max-Quiz
6. https://www.w3schools.com/react/
7. https://www.geeksforgeeks.org/spring-security-tutorial/