AALIM MUHAMMED SALEGH COLLEGE OF ENGINEERING
Approved by All India Council for Technical Education - New Delhi, Affiliated to Anna University, Chennai
NAAC Accredited Institution
"Nizara Educational Campus", Muthapudupet, Avadi - IAF, Chennai - 600 055.
ANNA UNIVERSITY COUNSELLING CODE : 1101
NBA ACCREDITED COURSES (Mech Engg, ECE, CSE & IT)

ONLINE LEARNING PLATFORM USING MERN

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NAAN MUDHALVAN PROJECT TITLE:

ONLINE LEARNING PLATFORM USING MERN

PROJECT TEAM MEMBERS:

1. **NIVEDHA P M –** (110121104072)
2. **POORNIMA S –** (110121104074)
3. **RANISHA S –** (110121104077)
4. **YASMIN PARVEEN –** (110121104105)

AALIM MUHAMMED SALEGH COLLEGE OF ENGINEERING
Approved by All India Council for Technical Education - New Delhi, Affiliated to Anna University, Chennai
NAAC Accredited Institution
"Nizara Educational Campus", Muthapudupet, Avadi - IAF, Chennai - 600 055.
ANNA UNIVERSITY COUNSELLING CODE : 1101
NBA ACCREDITED COURSES (Mech Engg, ECE, CSE & IT)

BONAFIDE CERTIFICATE

Certified that this project report on "**ONLINE LEARNING PLATFORM USING MERN**" is the Bonafide record of work done by Nivedha P M (110121104072), Poornima S (110121104074), Ranisha S (110121104077), Yasmin Parveen (110121104105). From the Department of Computer Science and Engineering at Anna University, Chennai.

Internal Guide

Head of the Department

Internal Examiner

External Examiner

ABSTRACT

This project presents the development of a dynamic online learning platform website using the MERN stack (MongoDB, Express.js, React.js, and Node.js). Designed to support diverse user roles, the platform enables students to register, explore and enroll in a range of courses, and earn completion certificates. Teachers can create and manage course content, while administrators maintain system oversight to ensure a secure and efficient environment. The platform leverages MongoDB for reliable data storage, Node.js and Express.js for backend functionality, and React.js for a responsive user interface. This MERN-based website offers a flexible and scalable educational experience, supporting interactive and self-paced learning in a structured online environment.

INTRODUCTION

The Online Learning Platform (OLP) project is a web-based educational solution built using the MERN (MongoDB, Express.js, React.js, Node.js) stack. This platform is designed to make learning more accessible by offering a structured environment where students can explore a range of courses, interact with educators, and achieve certifications. With dedicated user roles—students, teachers, and administrators—OLP provides tailored experiences for each user type. The system allows students to track their progress, enroll in paid or free courses, and engage in discussions. Teachers can create, manage, and update course content, while administrators oversee platform activity, ensuring a smooth and secure operation for all users. By leveraging the MERN stack, this project delivers a scalable, efficient, and user-friendly solution to support flexible, self-paced learning.

SCENARIO-BASED CASE STUDY:

Scenario: Learning a New Skill

User Registration: Sarah, a student interested in learning web development, visits the Online Learning Platform and creates an account. She provides her email and chooses a password.

Browsing Courses: Upon logging in, Sarah is greeted with a user-friendly interface displaying various courses categorized by topic, difficulty level, and popularity.

She navigates through the course catalog, filtering courses by name and category until she finds a "Web Development Fundamentals" course that interests her.

Enrolling in a Course: Sarah clicks on the course and reads the course description, instructor details, and syllabus. Impressed, she decides to enroll in the course.

After enrolling, Sarah can access the course materials, including video lectures, reading materials, and assignments.

Learning Progress: Sarah starts the course and proceeds through the modules at her own pace. The platform remembers her progress, allowing her to pick up where she left off if she needs to take a break.

Interaction and Support: Throughout the course, Sarah engages with interactive elements such as discussion forums and live webinars where she can ask questions and interact with the instructor and other learners.

Course Completion and Certification: After completing all the modules and assignments, Sarah takes the final exam. Upon passing, she receives a digital certificate of completion, which she can download and add to her portfolio.

Paid Courses: Sarah discovers an advanced web development course that requires payment. She purchases the course using the platform's payment system and gains access to premium content.

Teacher's Role: Meanwhile, John, an experienced web developer, serves as a teacher on the platform. He creates and uploads new courses on advanced web development topics, adds sections to existing courses, and monitors course enrollments.

Admin Oversight: The admin oversees the entire platform, monitoring user activity, managing course listings, and ensuring smooth operation. They keep track of enrolled students, handle any issues that arise, and maintain the integrity of the platform.

TECHNICAL ARCHITECTURE

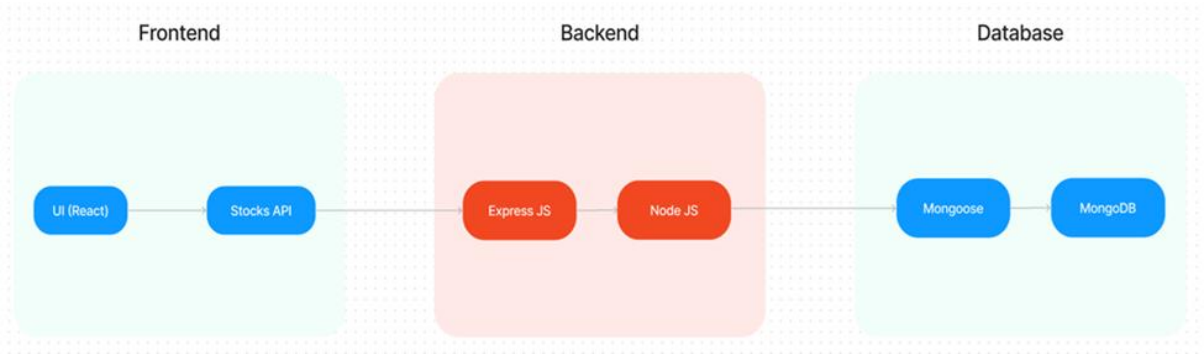
The technical architecture of OLP app follows a client-server model, where the frontend serves as the client and the backend acts as the server. The frontend encompasses not only the user interface and presentation but also incorporates the Axios library to connect with backend easily by using RESTful Apis.

The frontend utilizes the bootstrap and material UI library to establish a real-time and better UI experience for any user.

On the backend side, we employ Express.js frameworks to handle the server-side logic and communication.

Our backend relies on MongoDB for data storage and retrieval. MongoDB allows for efficient and scalable storage of user data and necessary information about the place.

Together, the frontend and backend components, along with Express.js, and MongoDB, form a comprehensive technical architecture for our OLP app. This architecture enables real-time communication, efficient data exchange, and seamless integration, ensuring a smooth and immersive blogging experience for all users.

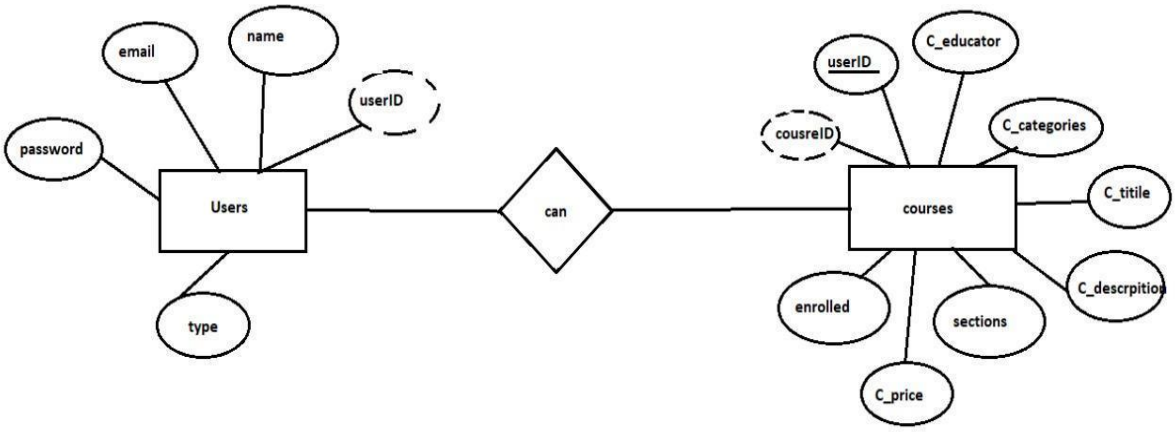


ER-DIAGRAM

Here there is 2 collections namely users, courses which have their own fields in

Users:

1. _id: (MongoDB creates by unique default)
2. name
3. email
4. password
5. type



Courses:

1. userID: (can act as a foreign key)
2. _id: (MongoDB creates by unique default)
3. C_educator
4. C_categories
5. C_title
6. C_description
7. sections
8. C_price
9. enrolled

PRE-REQUISITES

Here are the key prerequisites for developing a full-stack application using Node.js, Express.js, MongoDB, React.js:

?**Vite:**

Vite is a new frontend build tool that aims to improve the developer experience for development with the local machine, and for the build of optimized assets for production (go live). Vite (or ViteJS) includes: a development server with ES _native_ support and Hot Module Replacement; a build command based on rollup.

npm create vite@latest

?**Node.js and npm:**

Node.js is a powerful JavaScript runtime environment that allows you to run JavaScript code on the server-side. It provides a scalable and efficient platform for building network applications.

Install Node.js and npm on your development machine, as they are required to run JavaScript on the server-side.

Download: https://nodejs.org/en/download/

Installation instructions: https://nodejs.org/en/download/package-manager/

npm init

?**Express.js:**

Express.js is a fast and minimalist web application framework for Node.js. It simplifies the process of creating robust APIs and web applications, offering features like routing, middleware support, and modular architecture.

Install Express.js, a web application framework for Node.js, which handles server-side routing, middleware, and API development.

Installation: Open your command prompt or terminal and run the following command:

npm install express

?**MongoDB:**

MongoDB is a flexible and scalable NoSQL database that stores data in a JSON-like format. It provides high performance, horizontal scalability, and seamless integration with Node.js, making it ideal for handling large amounts of structured and unstructured data.

Set up a MongoDB database to store your application's data.

Download: https://www.mongodb.com/try/download/community

Installation instructions: https://docs.mongodb.com/manual/installation/

?**React.js:**

React.js is a popular JavaScript library for building user interfaces. It enables developers to create interactive and reusable UI components, making it easier to build dynamic and responsive web applications.

Install React.js, a JavaScript library for building user interfaces.

Follow the installation guide: https://reactjs.org/docs/create-a-new-react-app.html

?**HTML, CSS, and JavaScript**: Basic knowledge of HTML for creating the structure of your app, CSS for styling, and JavaScript for client-side interactivity is essential.

?**Database Connectivity**: Use a MongoDB driver or an Object-Document Mapping (ODM) library like Mongoose to connect your Node.js server with the MongoDB database and perform CRUD (Create, Read, Update, Delete) operations. To Connect the Database with Node JS go through the below provided link:

https://www.section.io/engineering-education/nodejs- mongoosejs-mongodb/

?**Front-end Framework**: Utilize Reactjs to build the user-facing part of the application, including entering booking room, status of the booking, and user interfaces for the admin dashboard.

For making better UI we have also used some libraries like material UI and boostrap.

Install Dependencies:

• Navigate into the cloned repository directory:

cd containment-zone

• Install the required dependencies by running the following commands:

cd frontend

npm install

cd ../backend

npm install

Start the Development Server:

• To start the development server, execute the following command:

npm start

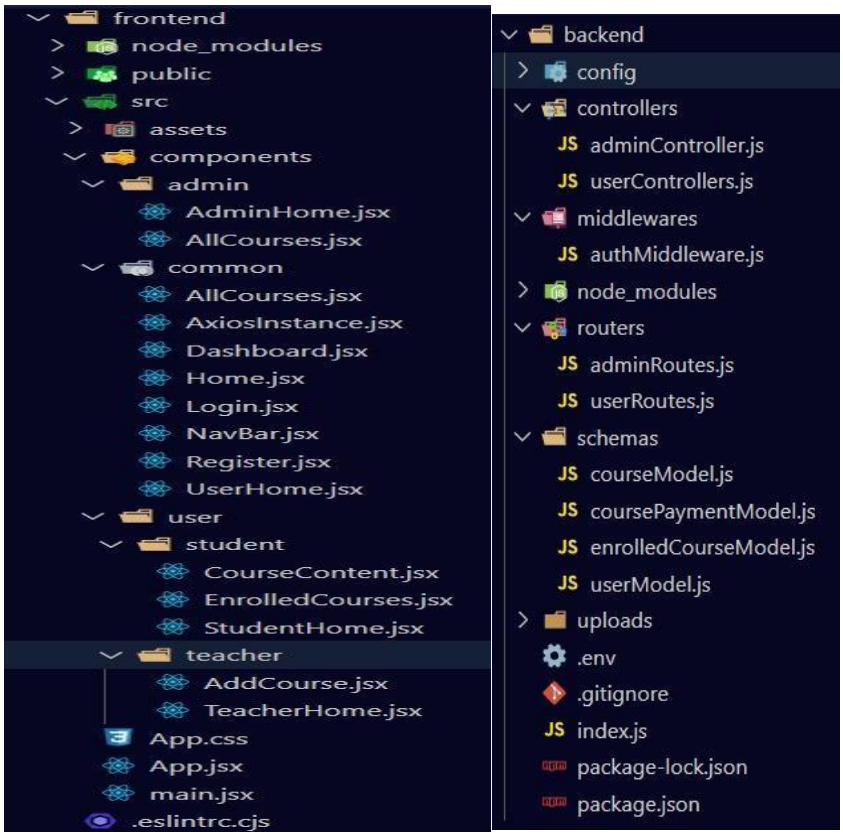
• The OLP app will be accessible at http://localhost:5172

You have successfully installed and set up the Online leraning app on your local machine. You can now proceed with further customization, development, and testing as needed.

PROJECT STRUCTURE

The first image is of frontend part which is showing all the files and folders that have been used in UI development

The second image is of Backend part which is showing all the files and folders that have been used in backend development



APPLICATION FLOW

The project has a user called– teacher and student and other will be Admin which takes care of all the user. The roles and responsibilities of these users can be inferred from the API endpoints defined in the code. Here is a summary:

Teacher:

1. Can add courses for the student.
2. Also delete the course if no student enrolled in it or for any other reasons.
3. Also add sections to courses.

Student:

1. Can enroll in an individual or multiple course.
2. Can start the course where it has stopped.
3. Once the course is completed, they can download their certificate of completion of the course.
4. For a paid course, they need to purchase it and then they can start the course.
5. They can filter out the course by searching by name, category, etc

Admin:

1. They can alter all the courses that are present in the app.
2. Watch out for all kinds of users in the app.
3. Record all the enrolled students that are enrolled in the course.

PROJECT FLOW

Let's start with the project development with the help of the given activities.

Project Setup And Configuration

Folder setup:

- Create frontend and
- Backend folders

Open the backend folder to install necessary tools

For backend, we use:

- cors
- bcryptjs
- express
- dotenv
- mongoose
- Multer
- Nodemon
- jsonwebtoken

BACKEND DEVELOPMENT

Setup express server

1. Create index.js file in the server (backend folder).
2. define port number, mongodb connection string and JWT key in env file to access it.
3. Configure the server by adding cors, body-parser.

Add authentication:

You need to make middleware folder and in that make authMiddleware.js file for the authentication of the projects and can use in.

DATABASE DEVELOPMENT

Configure MongoDB

1. Import mongoose.
2. Add database connection from config.js file present in config folder.
3. Create a model folder to store all the DB schemas.

FRONTEND DEVELOPMENT

Installation of required tools:

For frontend, we use:

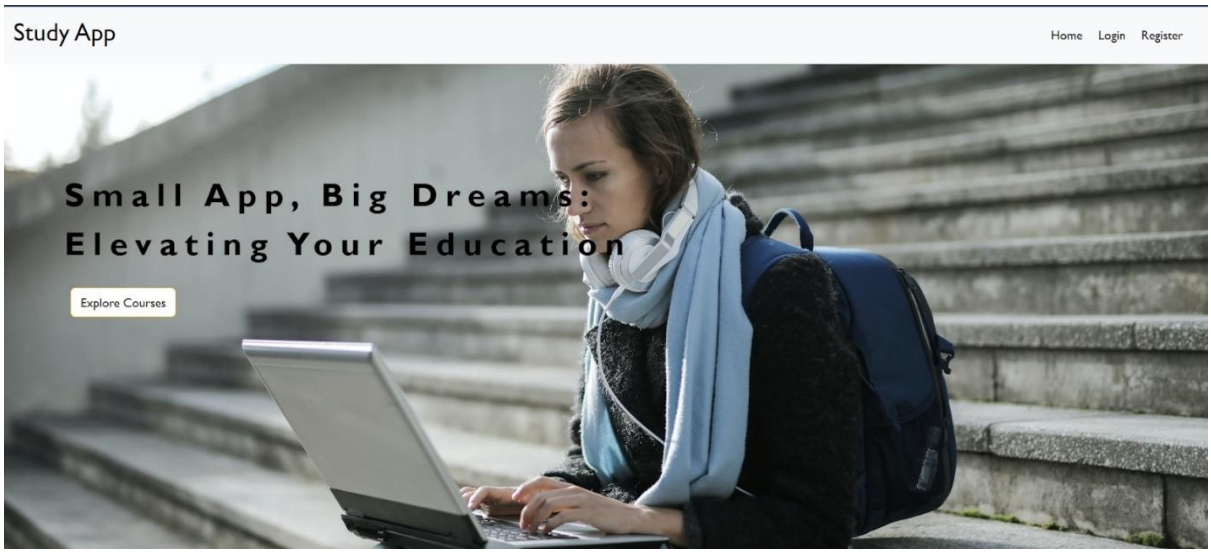
1. React
2. Bootstrap
3. Material UI
4. Axios
5. Antd
6. mdb-react-ui-kit
7. react-bootstrap

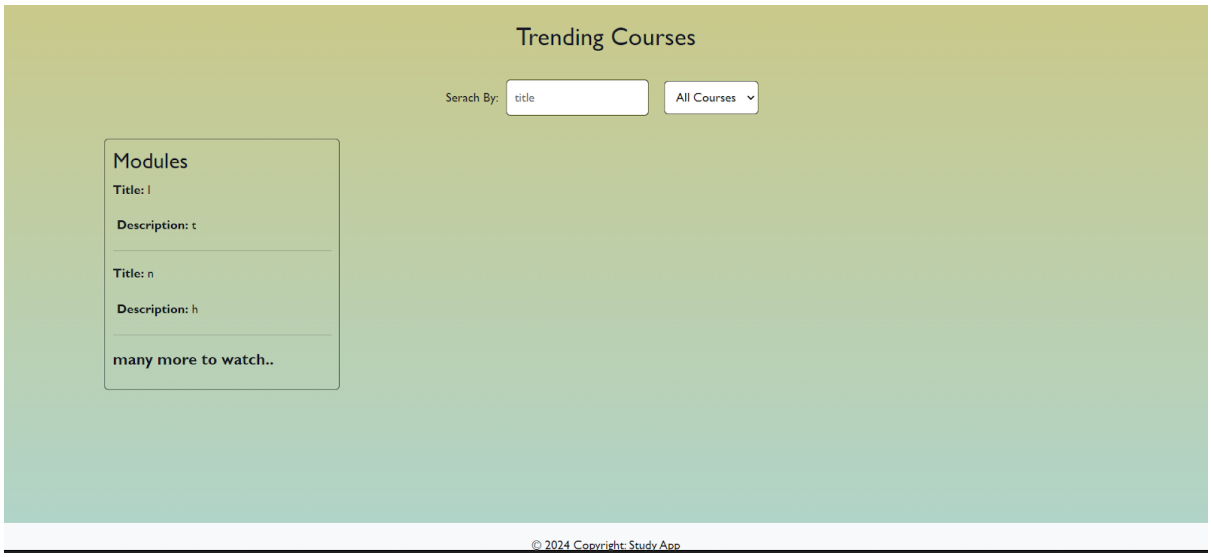
```
{
  "name": "frontend",
  "private": true,
  "version": "0.0.0",
  "type": "module",
  ▷ Debug
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "lint": "eslint . --ext js,jsx --report-unused-disable-directives --max-warnings 0",
    "preview": "vite preview"
  },
  "dependencies": {
    "@emotion/react": "^11.11.1",
    "@emotion/styled": "^11.11.0",
    "@mui/icons-material": "^5.14.9",
    "@mui/material": "^5.14.9",
    "axios": "^1.5.0",
    "bootstrap": "^5.3.2",
    "html2canvas": "^1.4.1",
    "jspdf": "^2.5.1",
    "mdb-react-ui-kit": "^6.1.0",
    "mdb-ui-kit": "^6.4.0",
    "react": "^18.2.0",
    "react-bootstrap": "^2.8.0",
    "react-dom": "^18.2.0",
    "react-player": "^2.13.0",
    "react-router-dom": "^6.16.0"
  },
  "devDependencies": {
    "@types/react": "^18.2.15",
    "@types/react-dom": "^18.2.7",
    "@vitejs/plugin-react": "^4.0.3",
    "eslint": "^8.45.0",
    "eslint-plugin-react": "^7.32.2",
    "eslint-plugin-react-hooks": "^4.6.0",
    "eslint-plugin-react-refresh": "^0.4.3",
    "vite": "^4.4.5"
  }
}
```

PROJECT IMPLEMENTATION & EXECUTION

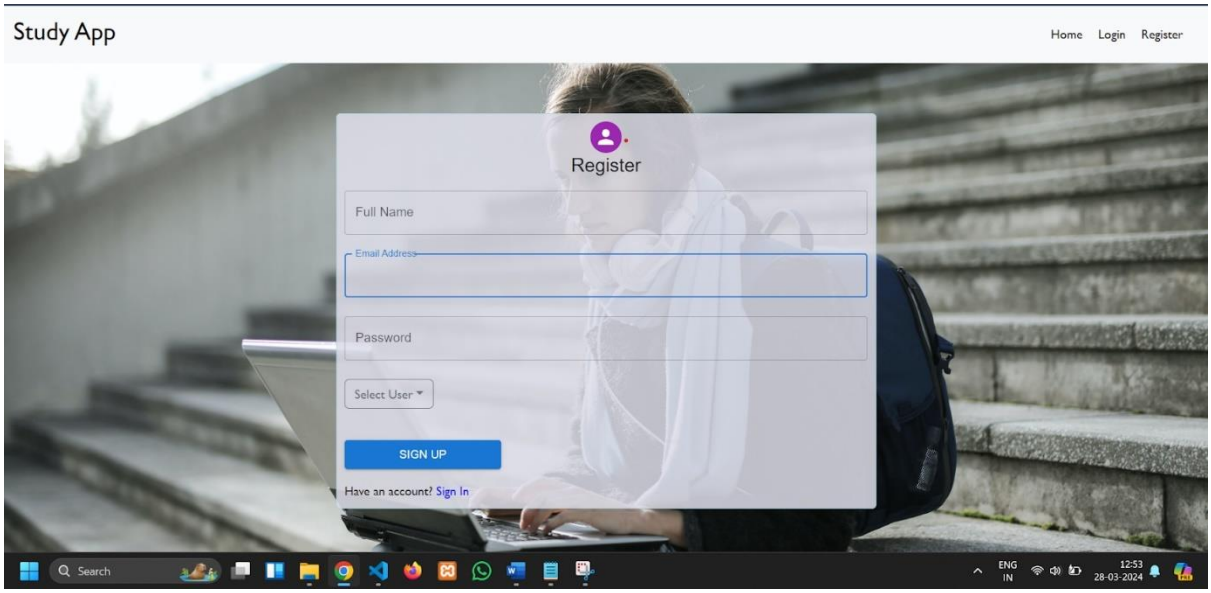
On completing the development part, we then run the application one last time to verify all the functionalities and look for any bugs in it. The user interface of the application looks a bit like the one's provided below.

Landing page

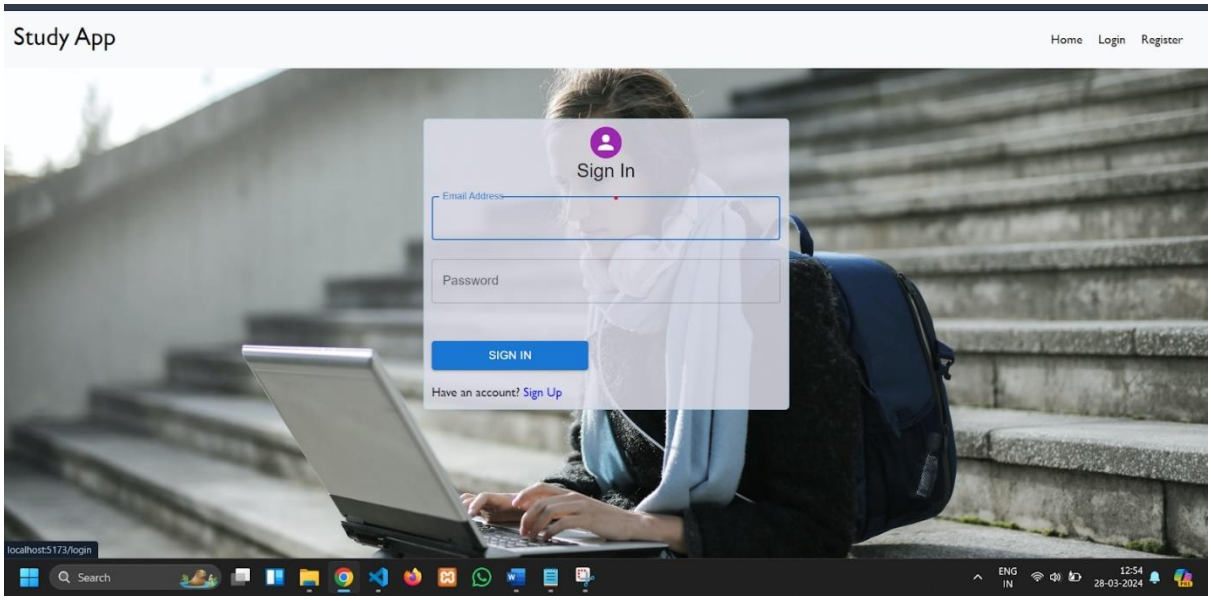




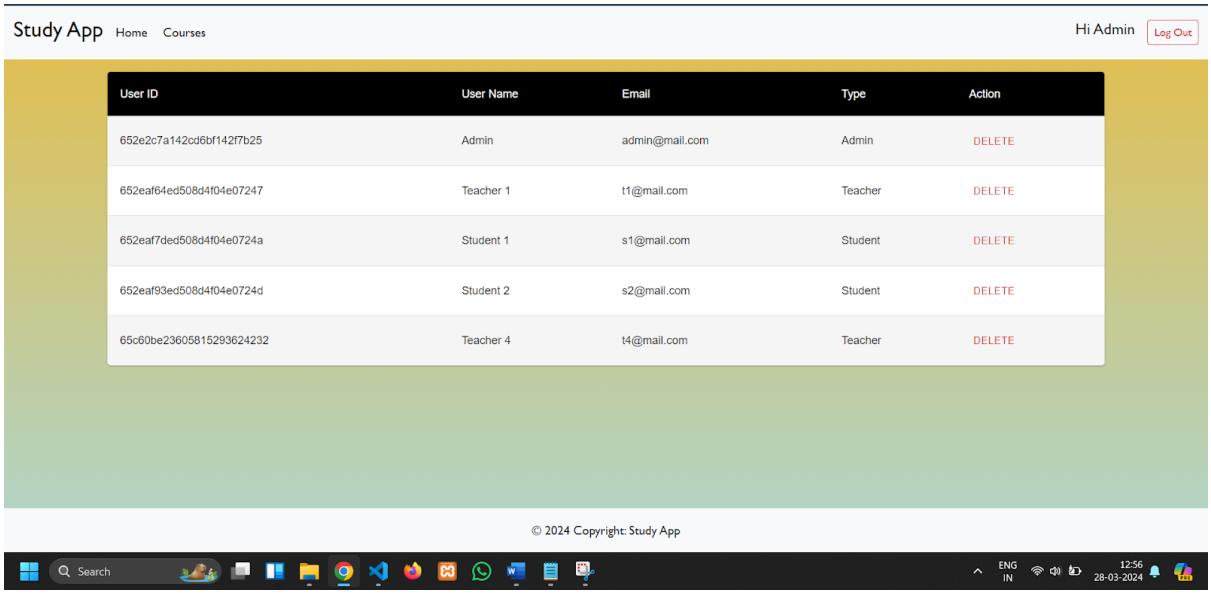
Register page:



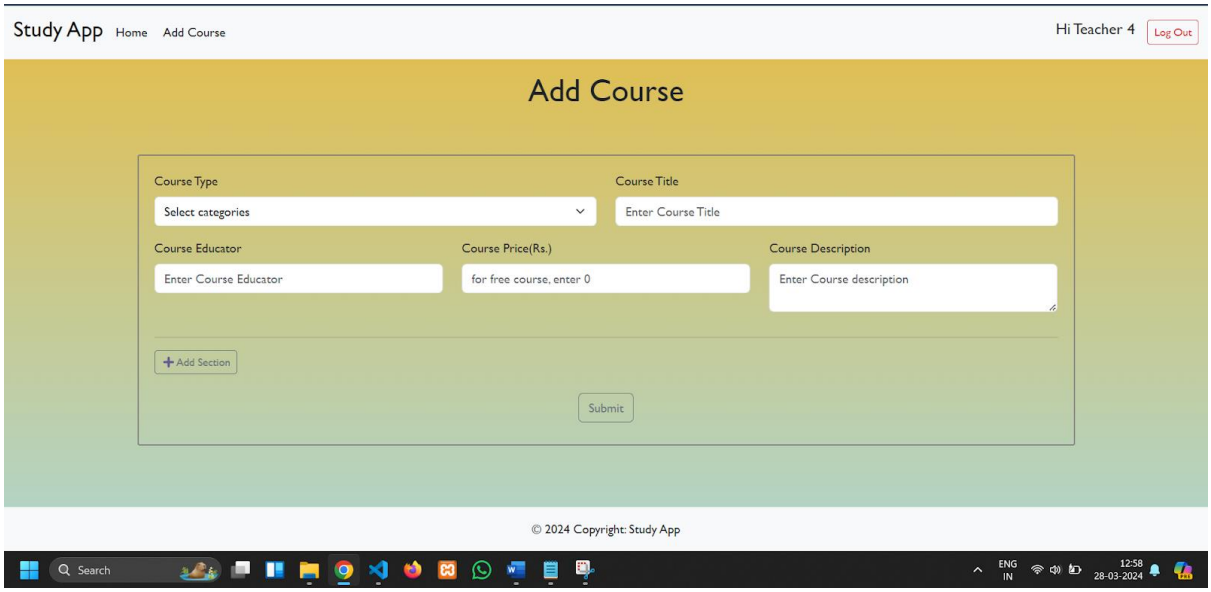
Login page:



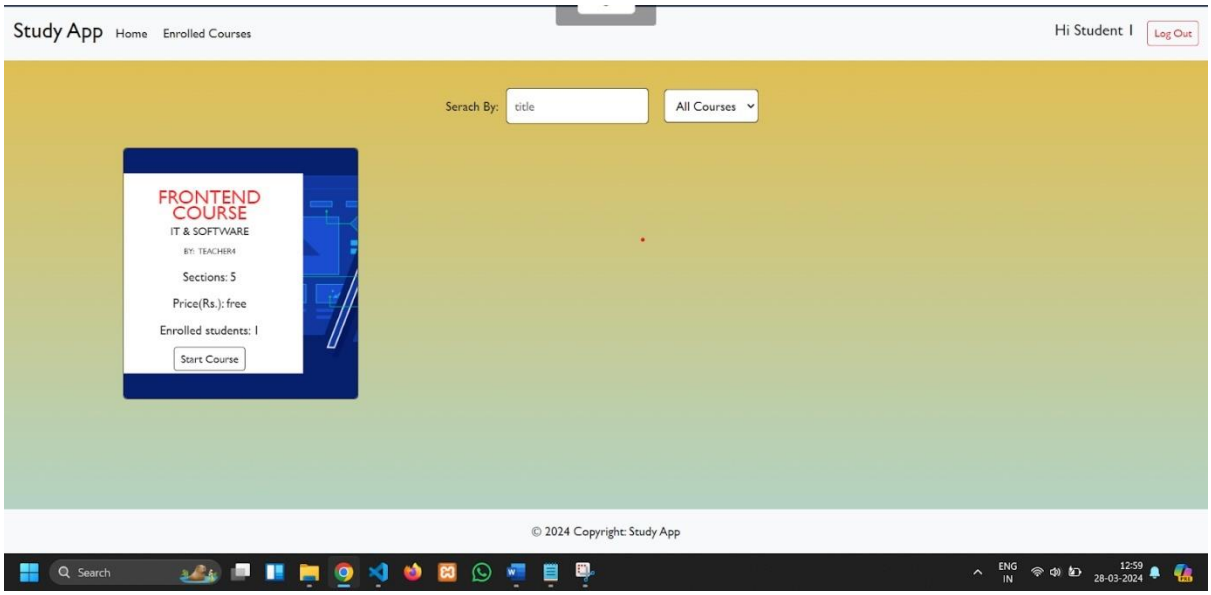
Admin Dashboard:

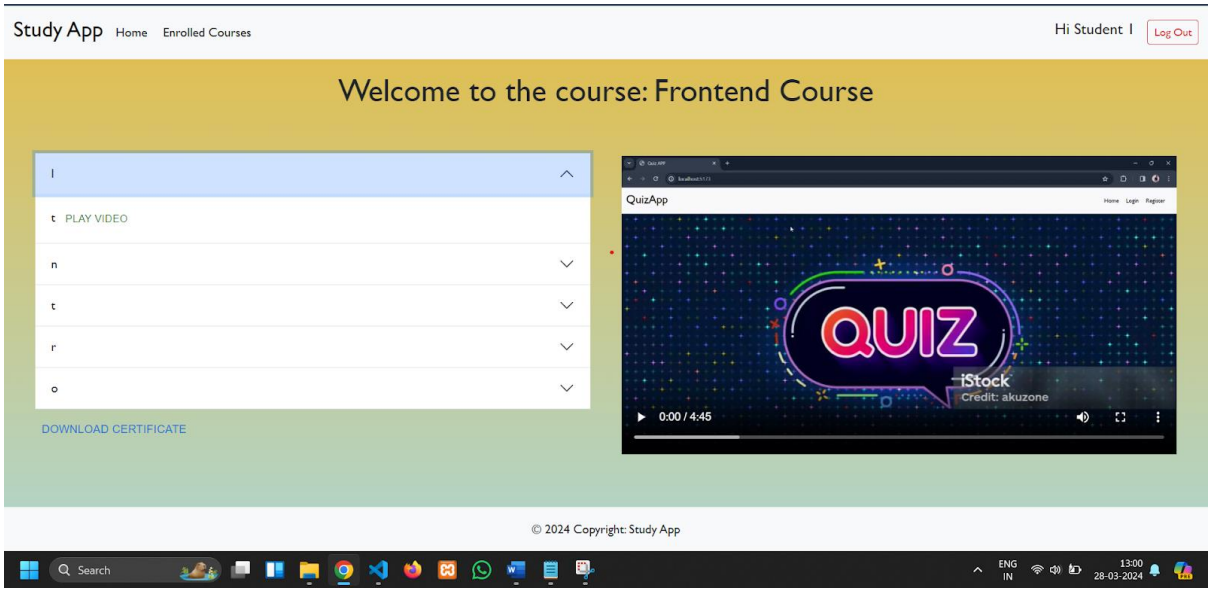


Teacher Dashboard:



Student Dashboard:




Study App
Home
Enrolled Courses
Hi Student I
Log Out
Welcome to the course: Frontend Course
I
t PLAY VIDEO
n
t
r
o
DOWNLOAD CERTIFICATE
QuizApp
Home
Login
Register
QUIZ
iStock
Credit: akuzone
0:00 / 4:45
© 2024 Copyright: Study App
Search
ENG
IN
13:00
28-03-2024