

Full Stack Development with MERN

Project Documentation

1. Introduction

- **Project Title:** LearnHub: Your Center for Skill Enhancement
- **Team Members:** List of Team Members and their Roles

Team Members	Roles
Kurra Madhu Priya	Team Leader & Backend Lead
Kondareddy Kiran Kumar	Admin Panel & Agent Dashboard Developer
Kistaparapu Sai Durga	Front End Developer
Kodipilla Poojitha	Testing, Documentation & Deployment Engineer

2. Project Overview

- **Purpose:**

Learnhub aims to be a comprehensive and user-friendly platform for online learning and skill development. It empowers students by providing access to structured, high-quality courses anytime and from anywhere, while enabling instructors to easily create, manage, and deliver engaging educational content. The platform supports personalized learning experiences through progress tracking, interactive lessons, and certification upon course completion. With secure authentication, cloud-based storage, and scalable infrastructure, LearnHub ensures a reliable and safe environment for both learners and educators. Its goal is to make education accessible, interactive, and effective for a wide audience.

- **Features:**

1. **User Roles:** Separate dashboards for Admin, Instructor, and Student
2. **Course Management:** Create, edit, and organize courses with videos, PDFs, and quizzes
3. **Progress Tracking:** Monitor student learning progress and completion status
4. **Certification:** Auto-generate certificates upon course completion
5. **Secure Authentication:** JWT-based login with role-based access control

6. **Cloud Storage:** Upload and stream content securely via cloud storage
7. **Interactive Learning:** Include comments, quizzes, and course discussions
8. **Payment Integration (*optional*):** Allow paid courses with Razorpay/Stripe
9. **Analytics Dashboard:** Track user activity, enrollment stats, and feedback
10. **Responsive Design:** Accessible on mobile, tablet, and desktop

3. Architecture

Frontend (React.js)

- Built with React.js for a dynamic and responsive UI
- Provides role-based interfaces for students, instructors, and admins
- Allows users to browse courses, track progress, and view certificates
- Connects to backend APIs to fetch and display data
- Ensures a smooth user experience across devices (responsive design)

Backend (Node.js + Express.js)

- Developed using Node.js with Express.js framework
- Handles business logic and all API routing
- Manages user registration, login, authentication, and role access
- Controls course creation, progress tracking, and certificate generation
- Integrates with external services like payment gateways and cloud storage

Database (MongoDB + Mongoose)

- Uses MongoDB for flexible and scalable data storage
- Mongoose handles data modeling and schema validation

- Stores users, roles, course content, enrollments, and progress
- Supports efficient queries and updates for real-time data
- Scales easily with increasing users and course content

4. Setup Instructions

- **PRE-REQUISITES:**

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

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/>

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/>

Version Control:

- Use Git for version control, enabling collaboration and tracking changes throughout the development process. Platforms like GitHub or Bitbucket can be used to host your repository.
- **Git Installation:** Download and installation instructions can be found at:
<https://git-scm.com/downloads>
- **Vs Code:** Download from <https://code.visualstudio.com/download>

Install Dependencies:

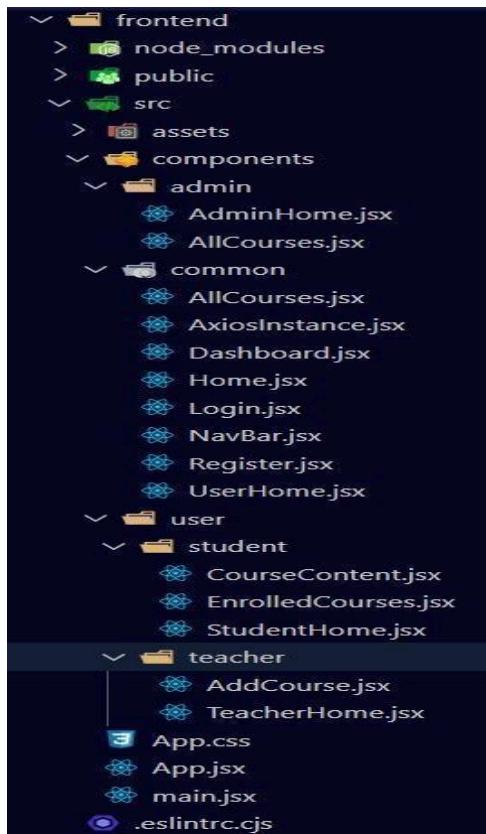
- Navigate into the cloned repository directory:
cd containment-zone
- Install the required dependencies by running the following commands:
cd frontend
npm install
cd ../backend
npm install

Start the Development Server:

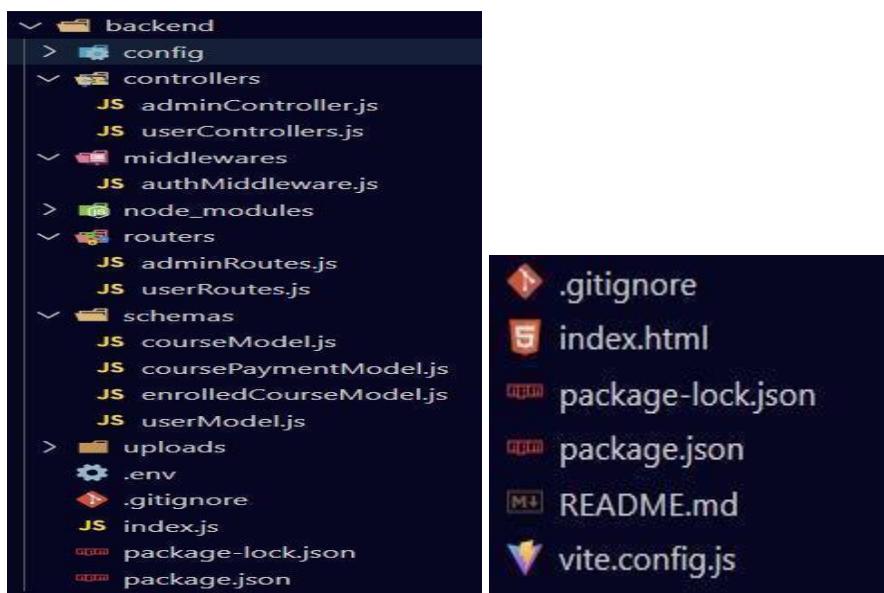
- To start the development server, execute the following command:
npm start
- The OLP app will be accessible at <http://localhost:5172>

5.Folder Structure

- Client: Describe the Structure of the React frontend.



- Server: Explain the organization of the Node.js backend.



6. Running the Application

Provide commands to start frontend and backend servers locally.

- **Frontend:** npm run dev in client directory.
- **Backend:** npm start in server directory.

7. API Documentation

- The backend exposes RESTful API endpoints to manage complaints, users, and authentication. Below are key endpoints:

- **User Routes**

POST /api/users/register - Register a new user

Body: { name, email, password, phone, userType }

POST /api/users/login - Authenticate a user

Body: { email, password }

- **Example response for POST/api/users/register**

```
{  
  "message": "User registered successfully",  
  "user": {  
    "_id": "64b8c2f5e25f9e001a2e3a88",  
    "name": "Ayesha Khan",  
    "email": "ayesha@example.com",  
    "phone": "9876543210",  
    "userType": "student",  
    "createdAt": "2025-07-20T09:00:00.000Z"  
  },  
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."  
}
```

- **Example response for POST/api/users/login**

```
{  
  "message": "Login successful",  
  "user": {  
    "_id": "64b8c2f5e25f9e001a2e3a88",  
    "name": "Ayesha Khan",  
    "email": "ayesha@example.com",  
    "userType": "student"  
  },  
  "token": "eyJhbGciOiJIUzI1NilsInR5cCI6IkpXVCJ9..."  
}
```

8.Authentication

- We use JWT(JSON Web Token) based Authentication.
- After login,a JWT is issued and stored in localStorage.
- Protected routes require the token in the Authorization header.
- Middleware on the backend validates the token to allow access to resources.
- **Roles Supported:**

User: Can register/login and create/view their complaints.

Admin: Can view/update/delete all complaints

9.User Interface

- The user interface is built using [React.js](#) and styled with React(UI) & Stocks API.
- Key UI Components:

Register/Login Forms

Course Registration Form

Course Progress Dashboard

Admin Dashboard for Course Creation

- Home Page



- Landing Page

The image shows the landing page of the study app. At the top center is the heading "Trending Courses". Below it is a search bar with the placeholder "Search By: title" and a dropdown menu labeled "All Courses". On the left, there is a sidebar titled "Modules" containing two entries: "Title: l" and "Description: t" on top, and "Title: n" and "Description: h" on the bottom. At the very bottom of the page, there is a footer bar with the copyright notice "© 2024 Copyright: Study App".

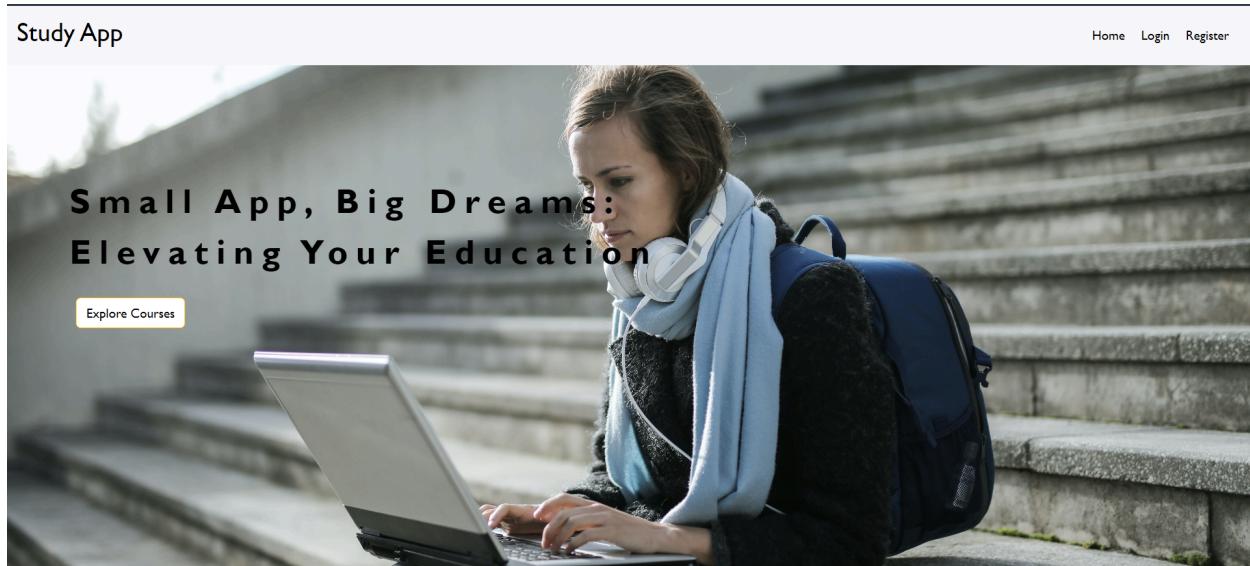
10. Testing:

- We followed manual and component-level testing for both frontend and backend.
- Backend Testing: Used Postman to test REST APIs for user and complaint modules.

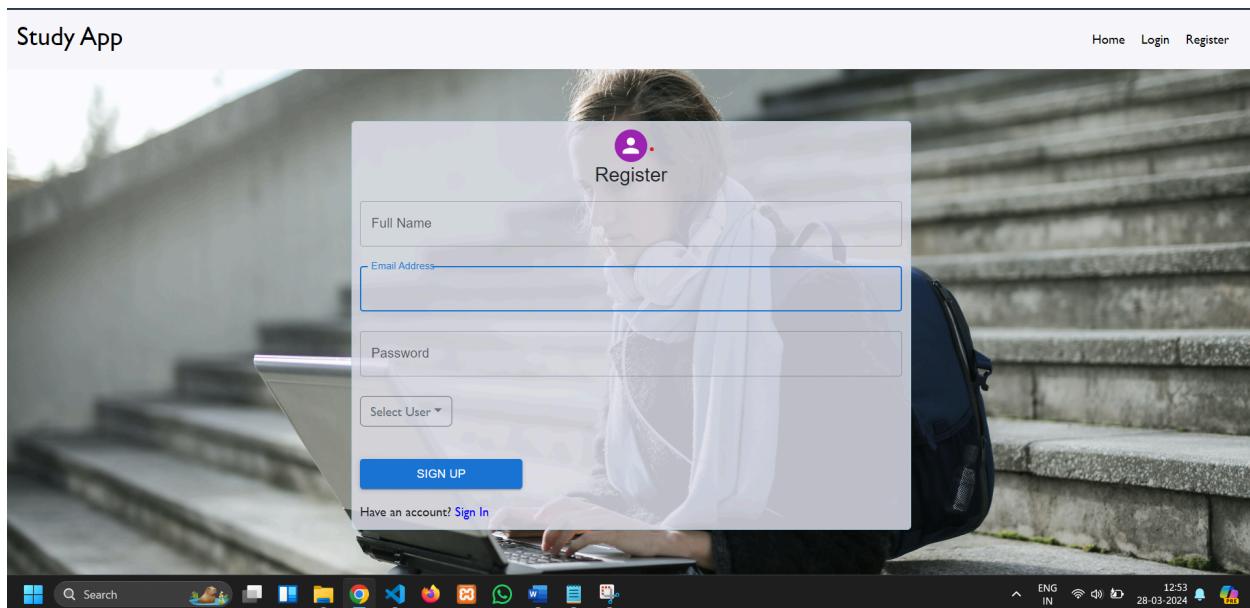
- Frontend Testing: React components React components tested with sample data and checked for state changes for UI response.

11. Screenshots or Demo

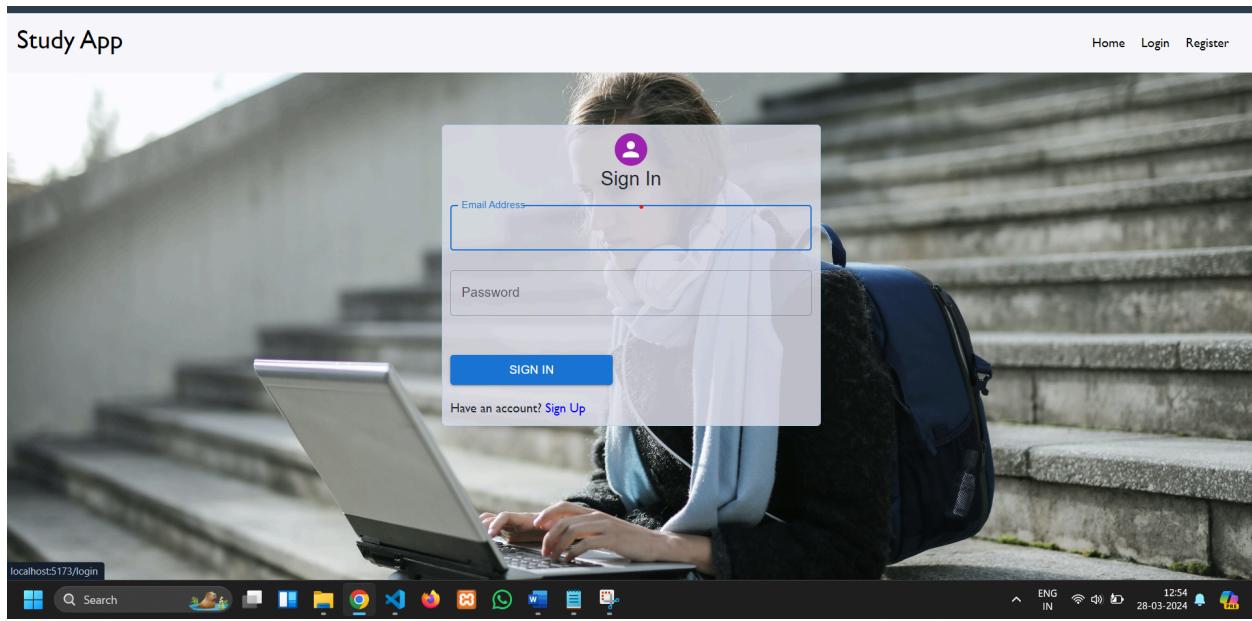
- Home Page



- Register Page



- Login Page



- Admin Dashboard

A screenshot of the 'Study App' Admin Dashboard. The dashboard has a dark header with the text 'Study App' and navigation links for 'Home' and 'Courses'. On the right side of the header is a 'Hi Admin' greeting and a 'Log Out' button. The main content area is a table titled 'User ID' with columns for 'User ID', 'User Name', 'Email', 'Type', and 'Action'. The table lists five users: Admin (admin@mail.com), Teacher 1 (t1@mail.com), Student 1 (s1@mail.com), Student 2 (s2@mail.com), and Teacher 4 (t4@mail.com). Each row contains a 'DELETE' link under the 'Action' column. The footer of the page includes the copyright notice '© 2024 Copyright: Study App' and a taskbar with various application icons and system status indicators.

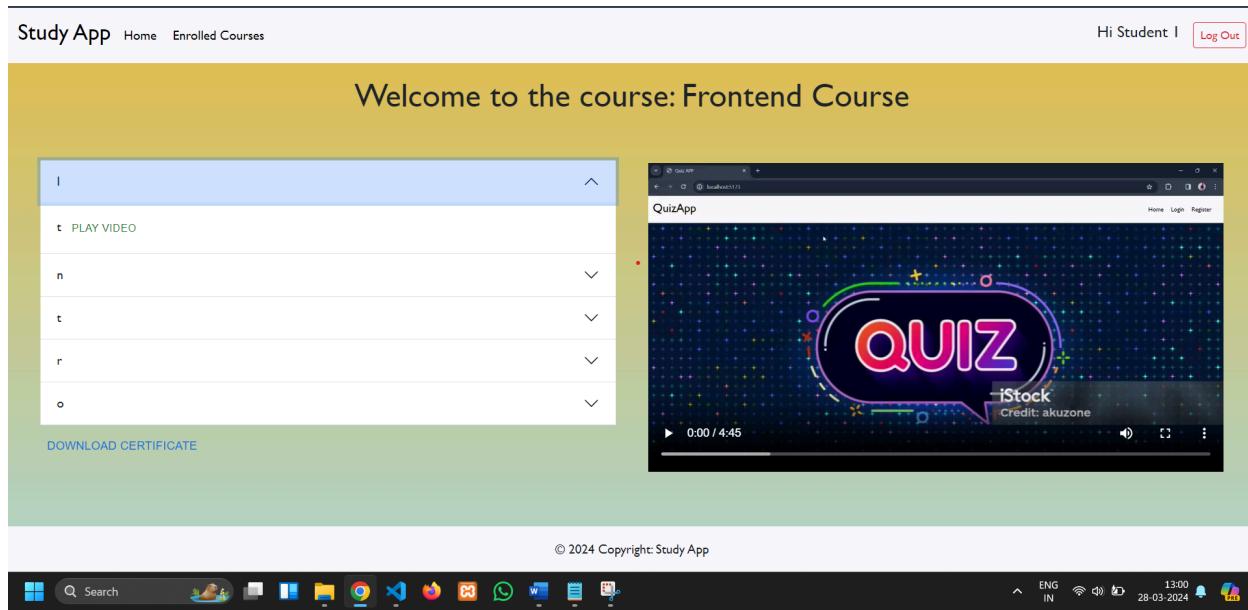
User ID	User Name	Email	Type	Action
652e2c7a142cd6bf142f7b25	Admin	admin@mail.com	Admin	DELETE
652eaf64ed508d4f04e07247	Teacher 1	t1@mail.com	Teacher	DELETE
652eaf7ded508d4f04e0724a	Student 1	s1@mail.com	Student	DELETE
652eaf93ed508d4f04e0724d	Student 2	s2@mail.com	Student	DELETE
65c60be23605815293624232	Teacher 4	t4@mail.com	Teacher	DELETE

- Teacher Dashboard

The screenshot shows the 'Add Course' page of the Study App. At the top, there's a header with 'Study App' and 'Home Add Course' links, and a 'Hi Teacher 4' greeting with a 'Log Out' button. The main area has a yellow-to-green gradient background. It contains fields for 'Course Type' (dropdown menu 'Select categories'), 'Course Title' (text input 'Enter Course Title'), 'Course Educator' (text input 'Enter Course Educator'), 'Course Price(Rs.)' (text input 'for free course, enter 0'), and 'Course Description' (text input 'Enter Course description'). Below these is a button '+ Add Section' and a 'Submit' button. At the bottom, it says '© 2024 Copyright: Study App' and shows a taskbar with various icons and system status.

- Student Dashboard:

The screenshot shows the Student Dashboard. At the top, there's a header with 'Study App' and 'Home Enrolled Courses' links, and a 'Hi Student 1' greeting with a 'Log Out' button. The main area displays a course card for 'FRONTEND COURSE IT & SOFTWARE'. The card includes details: 'BY: TEACHER4', 'Sections: 5', 'Price(Rs.): free', and 'Enrolled students: 1'. A 'Start Course' button is at the bottom. Above the card, there are search filters 'Search By: title' and 'All Courses'. At the bottom, it says '© 2024 Copyright: Study App' and shows a taskbar with various icons and system status.



12. Known Issues

- JWT token expiry may cause unexpected logout if not refreshed.
- Weak role-based checks can lead to unauthorized data access.
- Missing input validation can result in data inconsistency or security risks.
- Large media uploads may fail without proper size and timeout handling.

13. Future Enhancements

- Add real-time chat for student-instructor interaction.
- Integrate AI-based course recommendations and personalized learning paths.
- Enable live classes and webinar support using video conferencing APIs.
- Implement progress-based gamification with badges and leaderboards.