LearnHub: Your Center for Skill Enhancement

Team ID : LTVIP2025TMID54721
Team Size : 5
Team Leader : Pannem Jyothi Priya
Team member : Pannem Varshini
Team member : Parchuri Sai Varshitha
Team member : Pasupuleti Anusha
Team member : Pathan Mohammad Khan

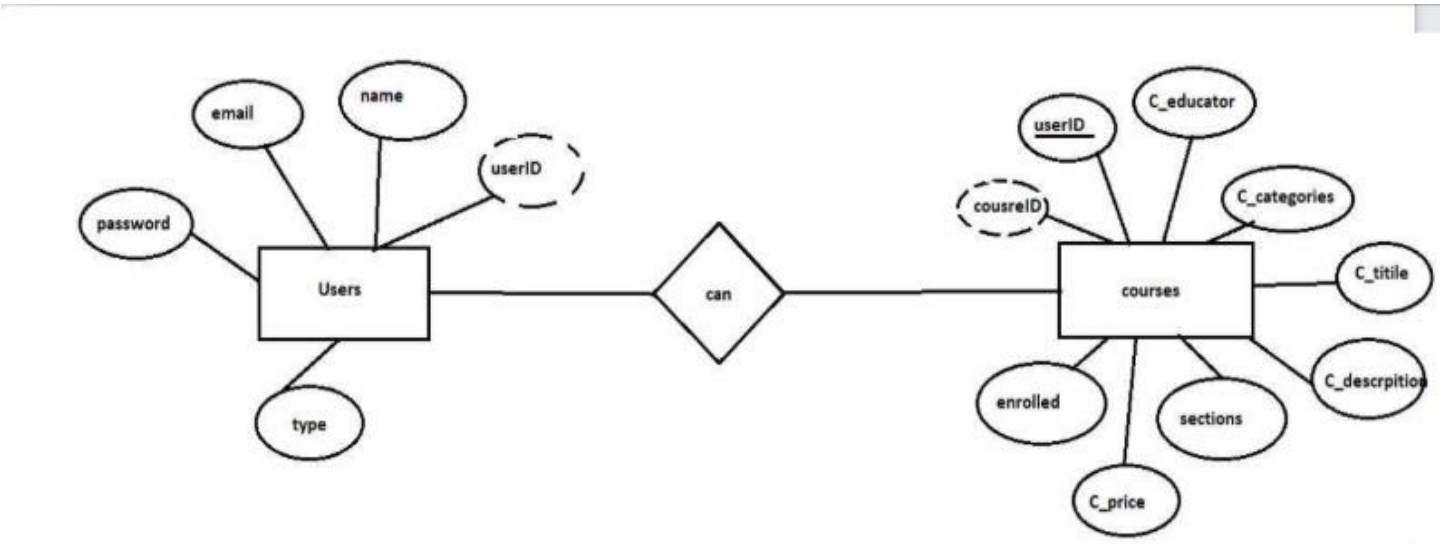
Project Overview

LearnHub is a comprehensive online learning platform (OLP) developed using the MERN stack (MongoDB, Express.js, React.js, Node.js). This project is built as part of a Smart Internship on LearnHub to apply full-stack development skills in a real-world scenario. These platforms have become increasingly popular, especially in recent years, as they offer flexibility and accessibility for learners of all ages and backgrounds.

Objectives

- Design a scalable full-stack web application for digital learning.
- Allow students to browse, enroll, and access video courses.
- Enable instructors to upload and manage course content (videos, notes).
- Provide administrators with control over user and course management.
- Ensure secure login system using JWT authentication and Bcrypt hashing.
- Deploy the project using modern cloud hosting services.

ER Diagram



Users

- **Description:**

LearnHub: Your Center for Skill Enhancement

This entity stores all user information in the system, including students, teachers, and admins.

- Attributes:
- userID *(Primary Key)* — unique identifier for each user (shown with dashed oval).
- name — full name of the user.
- email — user's email address.
- password — user's encrypted password for authentication.
- type — defines the user's role (Student, Teacher, or Admin).

Courses

Description:

This entity stores information about each course available on the platform.

- **Attributes:**
  - courseID *(Primary Key)* — unique identifier for each course (shown with dashed oval).
  - C_title — title of the course.
  - C_description — detailed description of the course content.
  - C_categories — categories/tags the course belongs to.
  - C_educator — the instructor or teacher associated with the course.
  - sections — sections or modules included in the course.
  - C_price — price of the course (zero if free).
  - enrolled — keeps track of how many students are enrolled.
  - userID *(Foreign Key)* — references the Users table to link the course creator.

Relationship

can

Meaning:

- The relationship "can" shows that a User can create, manage, or enroll in a Course.
- Teachers can create and manage courses.
- Students can enroll in courses.
- Admins can monitor and manage all courses and users.

Key Points

- The primary keys (userID and courseID) are shown with dashed ovals.
- The relationship ensures that every course must be linked to an educator (teacher) through userID.
- The ER diagram clearly maps which user owns or interacts with which course, forming the backbone of the platform's database design.

LearnHub: Your Center for Skill Enhancement

This ER diagram serves as a blueprint for designing the MongoDB collections and the Mongoose schemas in the Node.js backend.
It ensures data consistency, clear roles, and efficient queries for user enrollment, course creation, and management.

Technology Stack

Frontend:

- React.js: For building interactive UI components.

- Tailwind CSS: For fast and responsive styling.

Backend:

- Node.js: JavaScript runtime for backend logic.

- Express.js: Web framework to build APIs.

Database:

- MongoDB: NoSQL database to store data like users, courses, enrollments.

- Mongoose: ODM library to structure MongoDB interactions.

Authentication:

- JWT (JSON Web Tokens): For secure token-based authentication

LearnHub: Your Center for Skill Enhancement

Bcrypt: For password hashing and verification.

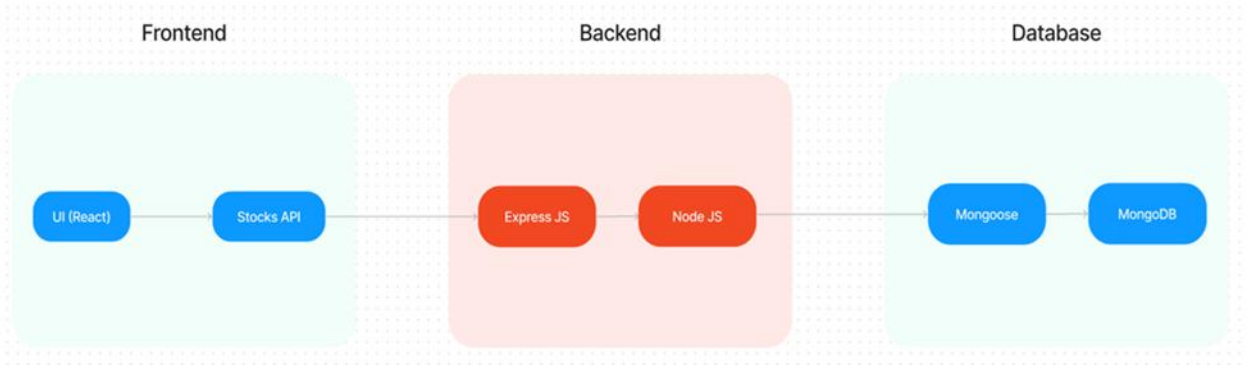
Hosting:

- Vercel/Netlify: Frontend deployment.

- Render/Railway: Backend deployment.

- MongoDB Atlas: Cloud-hosted database service.

System Architecture



The front end 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 For data storage and retrieval, our backend relies on MongoDB. 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

- The application follows a Model-View-Controller (MVC) architecture.

- React.js handles the view and dynamic routing.

- Express.js acts as the controller handling client requests and responses.

- MongoDB models manage data and structure.

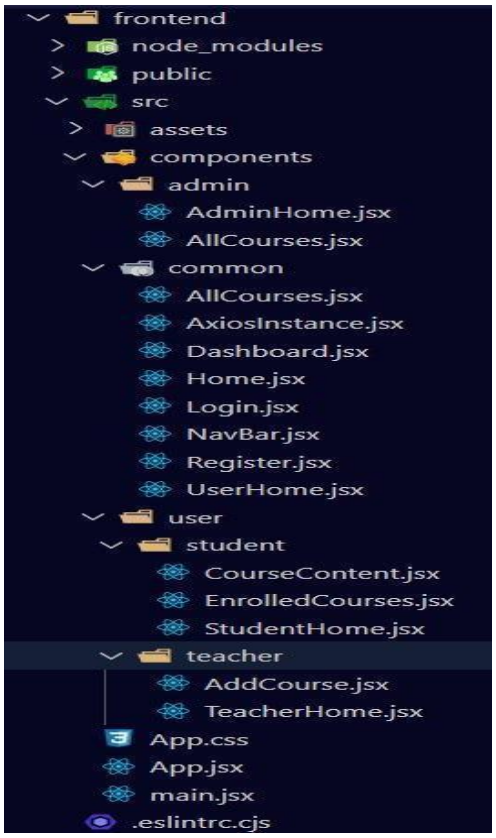
- All components are connected through RESTful APIs.

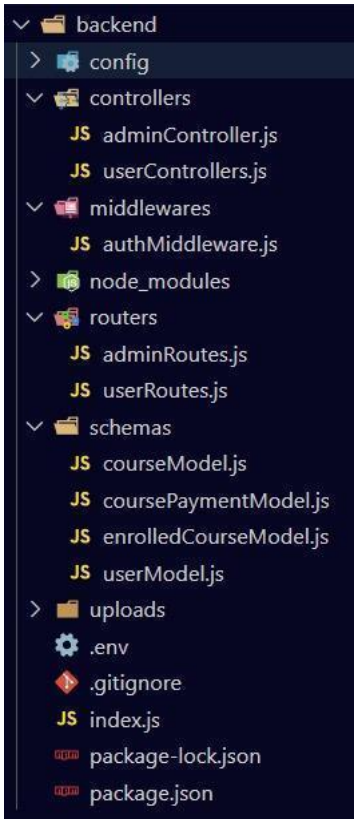
PROJECT STRUCTURE

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

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

LearnHub: Your Center for Skill Enhancement





LearnHub: Your Center for Skill Enhancement

Functional Modules

1. Student Module:

- Registration/Login
- View and enroll in courses
- Watch video lectures and download notes
- Track course progress

2. Instructor Module:

- Instructor dashboard
- Upload new courses (title, description, media)
- Edit or delete existing courses

3. Admin Module:

- Monitor platform usage
- Manage users and courses
- Approve or reject new course submissions

LearnHub: Your Center for Skill Enhancement

Database Schema Design

Users Collection:

- _id, name, email, password, role (student, instructor, admin)

Courses Collection:

- _id, title, description, instructorId, videoURLs, notes

Enrollments Collection:

- _id, courseId, studentId, progressPercentage

Important API Endpoints

- `POST /api/register` Register user
- `POST /api/login` Login and receive JWT token
- `GET /api/courses` Retrieve all published courses
- `POST /api/courses` Instructor uploads a new course
- `POST /api/enroll/:id` Enroll a student into a course
- `GET /api/user/courses` Get list of enrolled courses

Authentication & Security

- All passwords are hashed using Bcrypt before storing in the database.
- JWT tokens are generated on login and used to verify subsequent requests.
- Middleware ensures only authorized users can access protected routes.

Frontend Functionality

- Built with React.js and Tailwind CSS.
- React Router is used for page navigation.
- Dynamic role-based rendering for student/instructor/admin dashboard.
- Integrated video player and download buttons for learning materials.

LearnHub: Your Center for Skill Enhancement

Backend Functionality

- Built using Express.js and Node.js.
- Organized folder structure with routes, models, controllers, and middleware.
- RESTful API implementation to communicate with the frontend.
- Multer used for handling file uploads (PDFs, videos).

Testing and Validation

- Postman used for API testing (GET, POST, PUT, DELETE).
- Validation added on both frontend (form checks) and backend (middleware).
- Role-based route access tested for all user types.

Hosting and Deployment

- Frontend deployed on Vercel/Netlify.
- Backend deployed using Render/Railway.
- MongoDB Atlas for online NoSQL database storage.
- Environment variables stored securely in `.env` files.

Outcomes and Learnings

- Gained hands-on experience in full-stack development.
- Learned real-world use of JWT, REST APIs, and cloud deployment.
- Developed problem-solving skills for authentication and routing challenges.
- Understood modern development workflows and MVC architecture.

Future Enhancements

- Add quiz and auto-certification for completed courses.
- Integrate Razorpay/Stripe for paid courses.
- Implement live sessions via Zoom or WebRTC.
- Add notification system and in-app messaging.

Prerequisites for Developing E-LearnHub

Below are the key prerequisites for building a full-stack Online Learning Platform using the MERN stack with modern tooling.

Vite

Vite is a modern frontend build tool that provides an ultra-fast development server with ES modules and Hot Module Replacement (HMR).

It uses Rollup under the hood to bundle optimized assets for production. To create a new project with Vite:

npm create vite@latest

Node.js & npm

Node.js is a JavaScript runtime environment for running server-side code. npm (Node Package Manager) is required for managing packages.

Install Node.js and npm from nodejs.org. Initialize your project: npm init

Express.js

Express.js is a lightweight web application framework for Node.js. It handles API routes, middleware, and server logic.

Install with: npm install express

MongoDB

MongoDB is a powerful NoSQL database storing data in a JSON-like format. Install the community version: https://www.mongodb.com/try/download/community Follow setup instructions: https://docs.mongodb.com/manual/installation/

React.js

React.js is a popular JavaScript library for building dynamic and reusable UI components.

To set up, follow the official guide: https://reactjs.org/docs/create-a-new-react-app.html

HTML, CSS, JavaScript

A solid understanding of HTML (structure), CSS (styling), and JavaScript (interactivity) is essential.

Database Connectivity

Use Mongoose or the native MongoDB driver to connect Node.js with MongoDB.

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

Frontend Framework & Styling

The UI is built with React.js plus Material UI and Bootstrap for modern, responsive components.

Installing Dependencies

```
# Clone your repository, then:
cd containment-zone

# Install frontend dependencies:
cd frontend
npm install

# Install backend dependencies:
cd ../backend
npm install
```

Start the Development Server

```
npm start

Access the app at: http://localhost:5172
```

Roles & Responsibilities

Teacher

- Add new courses for students.
- Add sections to existing courses.
- Delete courses (if no students enrolled or as needed).

Student

- Enroll in one or multiple courses.
- Resume a course at last progress point.
- Download a certificate of completion once finished.
- Purchase premium (paid) courses.
- Search/filter courses by name, category, etc.

Admin

- Monitor and manage all courses.
- Manage all users in the system.
- Track enrollment records for all students.

To set up, follow the official guide: https://reactjs.org/docs/create-a-new-react-app.html

HTML, CSS, JavaScript

A solid understanding of HTML (structure), CSS (styling), and JavaScript (interactivity) is essential.

Database Connectivity

Use Mongoose or the native MongoDB driver to connect Node.js with MongoDB.

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

Frontend Framework & Styling

The UI is built with React.js plus Material UI and Bootstrap for modern, responsive components.

Installing Dependencies

```
# Clone your repository, then:
cd containment-zone

# Install frontend dependencies:
cd frontend
npm install

# Install backend dependencies:
cd ../backend
npm install
```

Start the Development Server

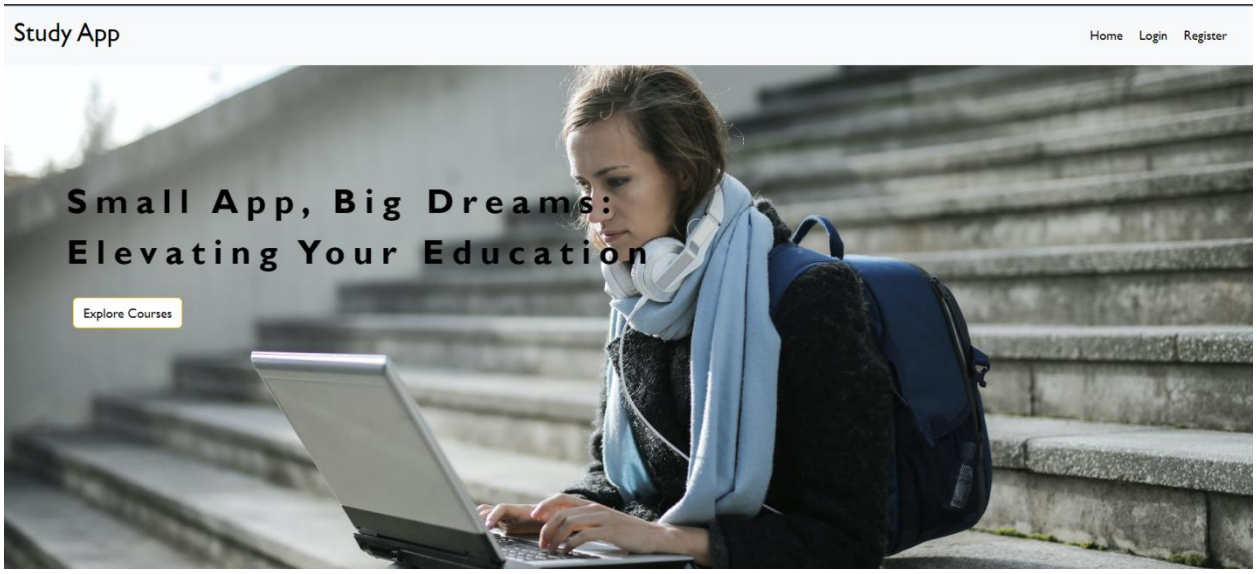
```
npm start

Access the app at: http://localhost:5172
```

Roles & Responsibilities

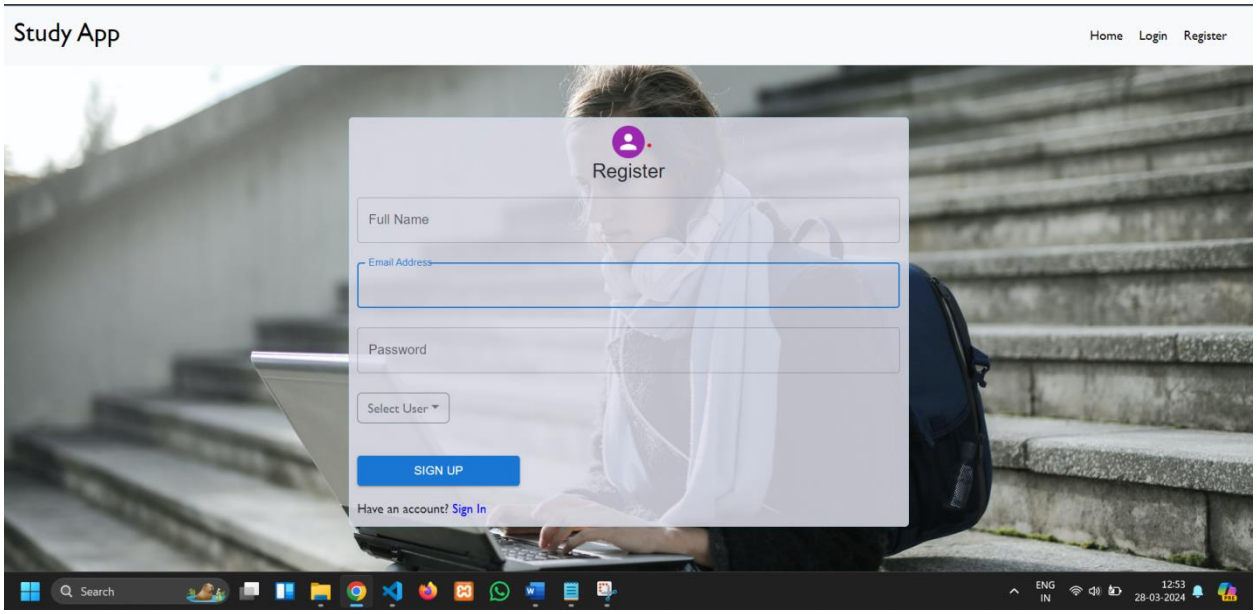
Teacher

- Add new courses for students.
- Add sections to existing courses.
- Delete courses (if no students enrolled or as needed).

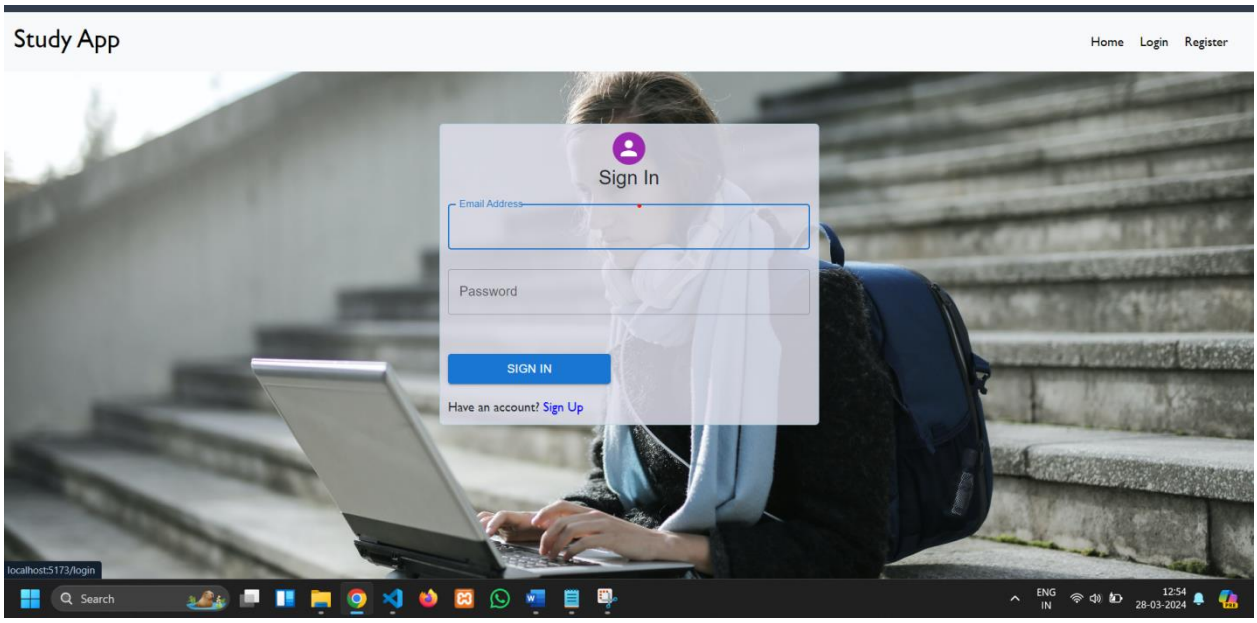
HOME PAGE:



SIGN UP PAGE:



SIGN IN PAGE:



Conclusion

The E-LearnHub Online Learning Platform is designed to deliver a flexible, user-friendly, and scalable solution for modern education needs. By integrating robust backend technologies such as Node.js, Express.js, and MongoDB with a dynamic React.js frontend and modern build tools like Vite, the platform provides a seamless experience for students, teachers, and administrators alike.

With role-based access, students can easily enroll in courses, track progress, and receive certificates of completion, while teachers can create, manage, and update courses effortlessly. The admin module ensures the platform remains secure and organized by managing all courses and monitoring user activity.
Through features like secure authentication (JWT & bcrypt), a clean UI built with Material UI and Bootstrap, and scalable data management with Mongoose, E-LearnHub lays a strong foundation for a real-world online learning system.

Future enhancements such as integrated quizzes, automated certification, payment gateways, live webinars, and real-time notifications will make the platform even more powerful and industry-ready, bridging the gap between traditional learning and modern e-learning demands.
In summary, E-LearnHub demonstrates how modern web development technologies can be combined to build a robust, secure, and feature-rich online education system — empowering learners and educators to connect and grow from anywhere, at any time.