LearnHub-Your Center For Skill Enhancement

1. Introduction

Project Title: *LearnHub: Your Center For Skill Enhancement*

Team ID: LTVIP2025TMID58662

Team Size: 3

- **Team Leader**: Mopidi Bandaru Sai Gnana Tejaswini – Project Lead & Model Developer
- **Team Member**: Chinta Vijaya Varshitha – Data Engineer
- **Team Member**: Kuruva Kishore Kumar – Frontend & Backend Developer

2.Project Overview

2.1. Purpose

LearnHub is a full-stack online learning platform built using the MERN stack (MongoDB, Express.js, React.js, Node.js). It serves as a centralized digital hub where learners can explore, enroll in, and complete courses to develop new skills at their own pace. Teachers can create and manage course content, while an admin oversees platform operations.

2.2. Features

- Role-based access: Student, Teacher, Admin
- User authentication and registration ○ Course creation, enrollment, and filtering ○ Payment gateway for premium courses ○ Certificate generation after course completion
- Dashboards for Admin, Teacher, and Student
- Responsive UI with search/filter options

3. Architecture

3.1. Frontend

- Built using **React.js**, styled with **Material UI**, **Bootstrap**, **AntD**, and **MDB UI Kit**
- Utilizes **Axios** for API calls
- Vite is used for optimized builds and fast reloads

3.2. Backend

- Built on Node.js with Express.js
- RESTful APIs handle CRUD operations
- Middleware for authentication (JWT)
- File uploads handled via Multer

Dataset:

- **MongoDB** with **Mongoose ODM**
- Two main collections: Users and Courses
- Schemas support user roles, course details, enrollments

4. Setup Instructions

4.1. Prerequisites

- Node.js
- MongoDB
- npm/yarn
- Vite
- Git

4.2. Installation

```
 # Clone the repository git
clone <your-repo-link> cd
learnhub

# Frontend setup cd
frontend
npm install

# Backend setup cd
../backend
npm install

# Set up .env file in backend #
Example:
PORT=5000
MONGO_URI=mongodb://localhost:27017/learnhub
JWT_SECRET=your_jwt_secret
```

5. Folder Structure

5.1. Client (Flask Frontend)

```
/frontend
  ├── src/
  │   ├── components/
```

```
│   ├── pages/
│   └── App.jsx
├── public/
└── vite.config.js
```

5.2. Server (Flask Backend)

```
/backend
├── controllers/
├── models/
├── routes/
├── middleware/
├── config/
└── index.js
```

6. Running the Application

- **Frontend:**

  cd frontend

  npm start

- **Backend:**

  cd backend

  npm start

7. API Documentation

Method	Endpoint	Description
POST	/api/users/register	Register new user
POST	/api/users/login	User login
GET	/api/courses	Fetch all courses
POST	/api/courses	Add new course (teacher only)

Method	Endpoint	Description
PUT	/api/courses/:id	Update course

DELETE	/api/courses/:id	Delete course
POST	/api/enroll/:id	Enroll in a course
GET	/api/dashboard	Dashboard data by role

8. Authentication

- **Method**: JWT (JSON Web Token)
- **Flow**:
- User logs in → Receives token
- Token attached to headers for protected routes
- Middleware verifies token and grants access based on role

9. User Interface

- **Student Dashboard**: View enrolled courses, resume learning, download certificates
- **Teacher Dashboard**: Manage course uploads, view enrolled students
- **Admin Dashboard**: Control all courses and users

UI Libraries Used:

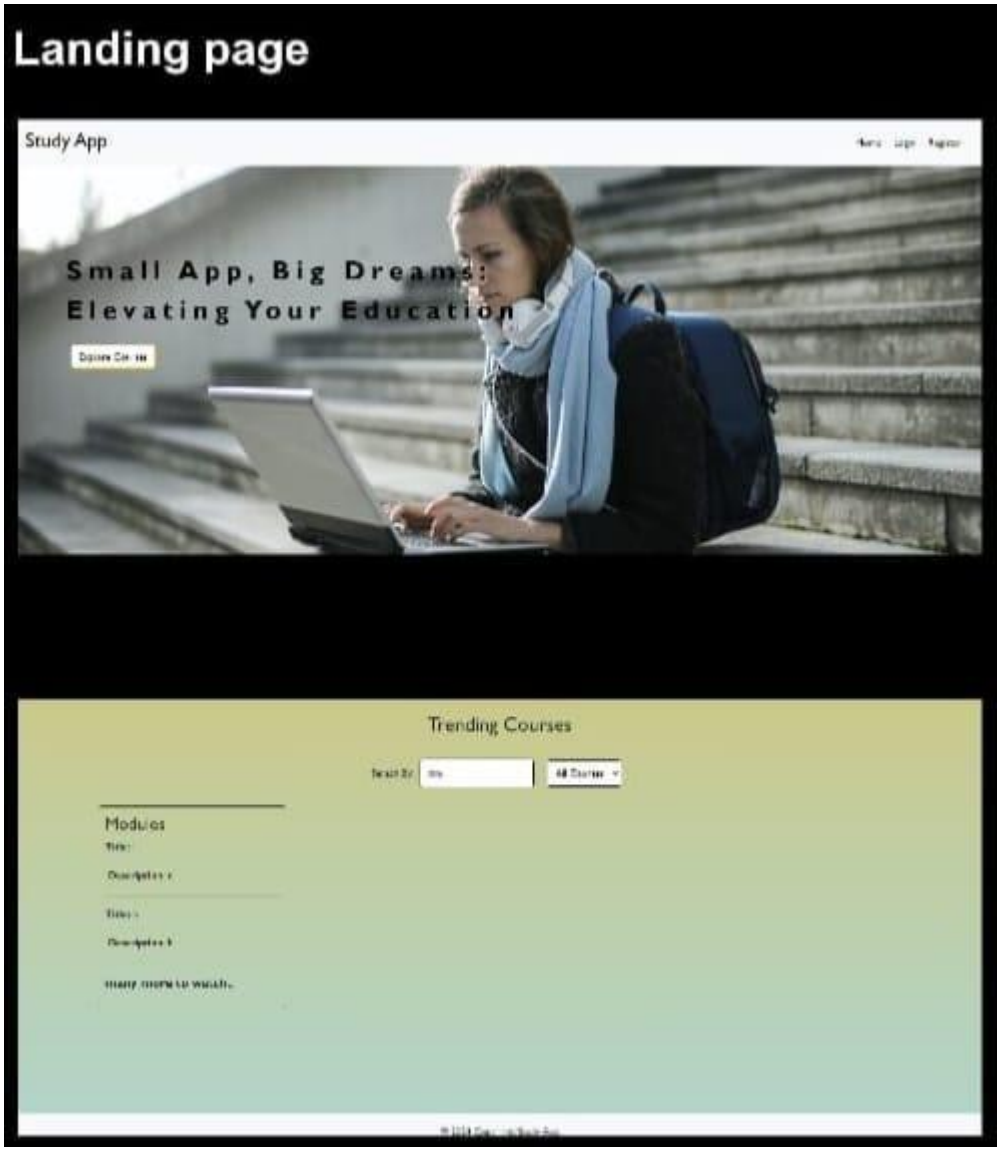
- Material UI
- Bootstrap
- MDB React UI Kit

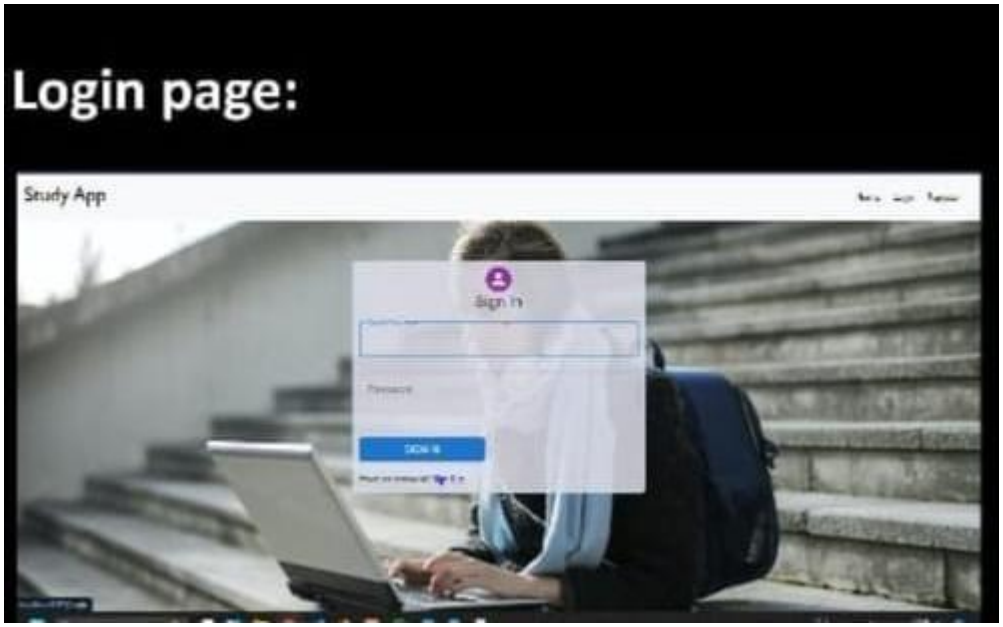
10. Testing

- Manual testing for:
  - Registration/Login
  - Course creation and deletion
  - Enrollment and certificate generation
- Backend APIs tested via Postman
- Frontend tested via browser and console logs

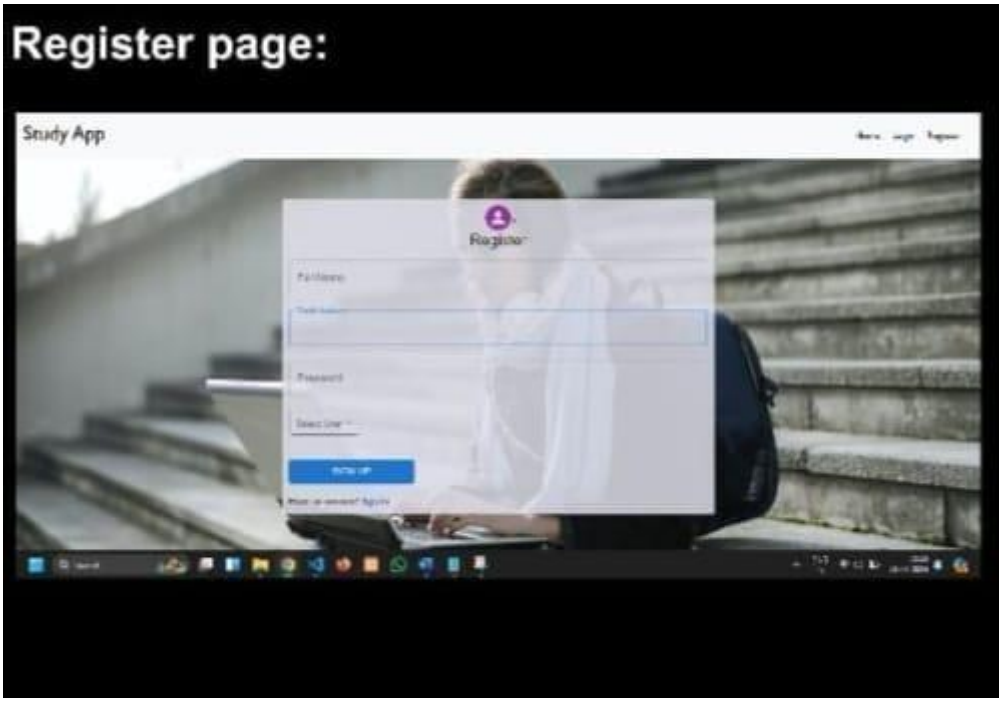
11. Screenshots or Demo

- Landing Page


Landing page
Study App
Small App, Big Dreams:
Elevating Your Education
Trending Courses
Modules

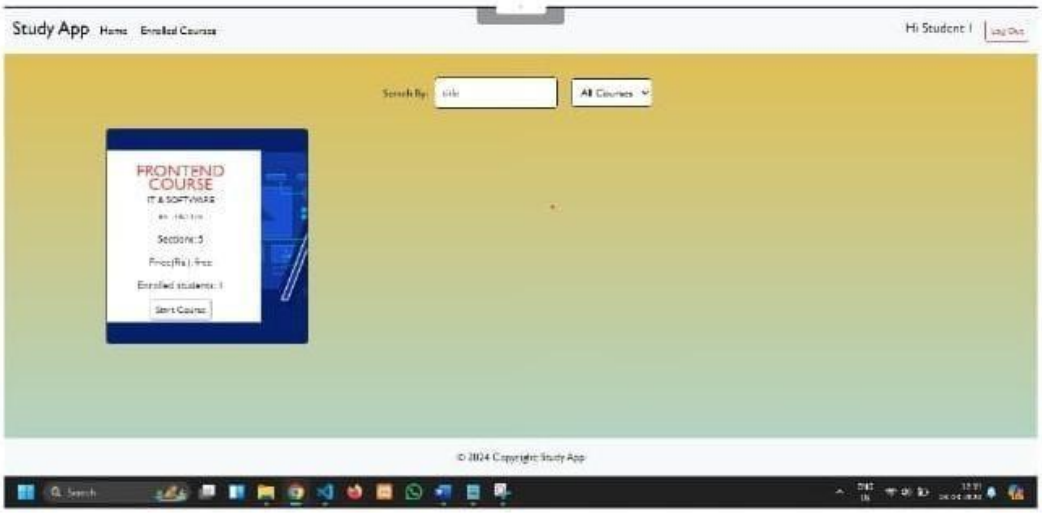
- 
Login page:
Study App
Sign In

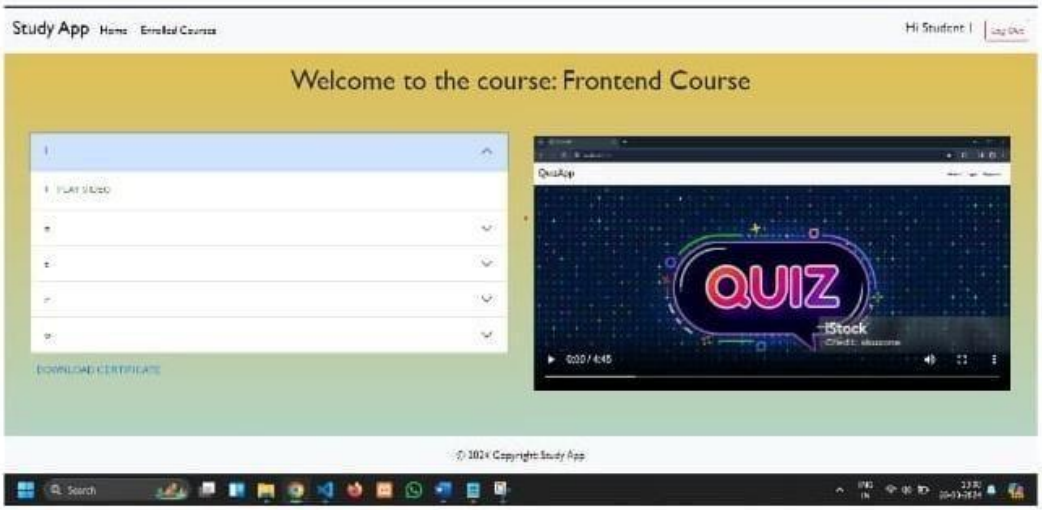
Login page



Register page

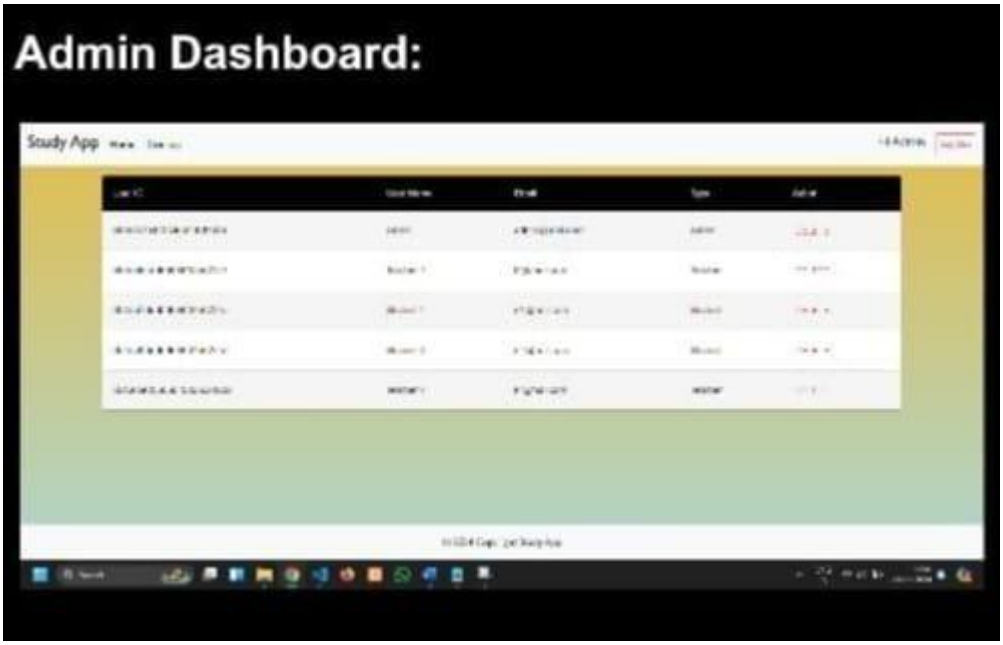
Student Dashboard:





Student page

Teacher Dashboard



Admin Dashboard

Demo Link:

https://www.canva.com/design/DAFzCeEE0QA/VgD1I3SKJUVAab2HS_SN8w/watch?utm_content=DAFzCeEE0QA&utm_campaign=designshare&utm_medium=link&utm_source=editor

12. Known Issues

- No batch enroll feature
- Mobile responsiveness may need improvement on some screens

- Payments are mocked (no real gateway integration)

13. Future Enhancements

- Integrate real payment gateway (e.g., Stripe/Razorpay)
- Add quizzes and assessments to courses
- Enable video streaming for course lectures
- Mobile app version
- AI-based course recommendations