

# Full Stack Development with MERN

## Project Documentation: Online Learning Platform

### 1. Introduction

Project Title: Online Learning Platform (OLP) using MERN Stack(LearnHub

Team Members:

Uppalapati Karthikeya Varma – Full Stack Developer

Maddumala Meghana – Frontend Developer

Shaik Shamila – Backend Developer

Pammi Likhitha– Database Administrator

### 2. Project Overview

Purpose:

To provide a digital platform for students and teachers to engage in online learning. The platform allows students to enrol in courses, complete lessons, and earn certifications, while teachers can create and manage courses.

Features:

- User Authentication (Student, Teacher, Admin)
- Course Management (CRUD operations for teachers/admin)
- Student Enrollment and Progress Tracking
- Certificate Generation
- Course Filtering and Search
- Admin Dashboard for monitoring all platform activity
- Payment integration for paid courses

### 3. Architecture

Frontend:

Built using React.js with Vite for fast development build, styled with Bootstrap and Material UI. Uses Axios for API calls.

Backend:

Node.js with Express.js for server-side logic and REST API development. Uses JSON Web Tokens (JWT) for authentication and middleware for protected routes.

Database:

MongoDB is a NoSQL database with Mongoose ODM. Collections include:

- Users Collection
- Courses Collection

## 4. Setup Instructions

Prerequisites:

- Node.js
- npm
- MongoDB
- Vite
- Express.js
- React.js

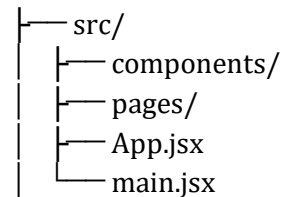
Installation Steps:

1. Clone the repository
2. Backend setup: npm install
3. Frontend setup: npm install
4. Database setup: Ensure MongoDB is running locally or via Atlas

## 5. Folder Structure

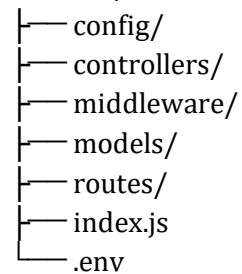
Client (Frontend):

frontend/



Server (Backend):

backend/



## 6. Running the Application

Frontend:

npm run dev

Backend:

npm start

Access URL:

Frontend: <http://localhost:5172>

Backend: <http://localhost:5000>

## 7. API Documentation

Example Endpoints:

- POST /api/users/register
- POST /api/users/login
- GET /api/courses
- POST /api/courses
- PUT /api/courses/:id
- DELETE /api/courses/:id
- POST /api/courses/enroll/:id
- GET /api/users/me

## 8. Authentication

Method Used:

JSON Web Tokens (JWT)

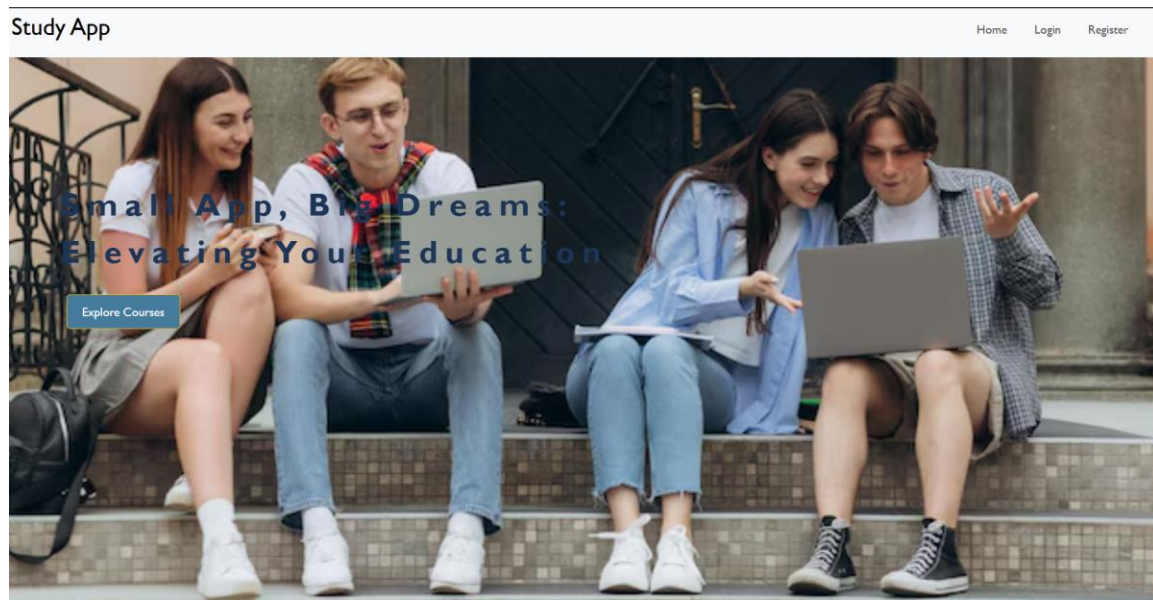
How it Works:

- Users register and log in to get JWT tokens.
- Protected routes use authentication middleware.
- User roles (student, teacher, admin) are checked at the API level.

## 9. User Interface

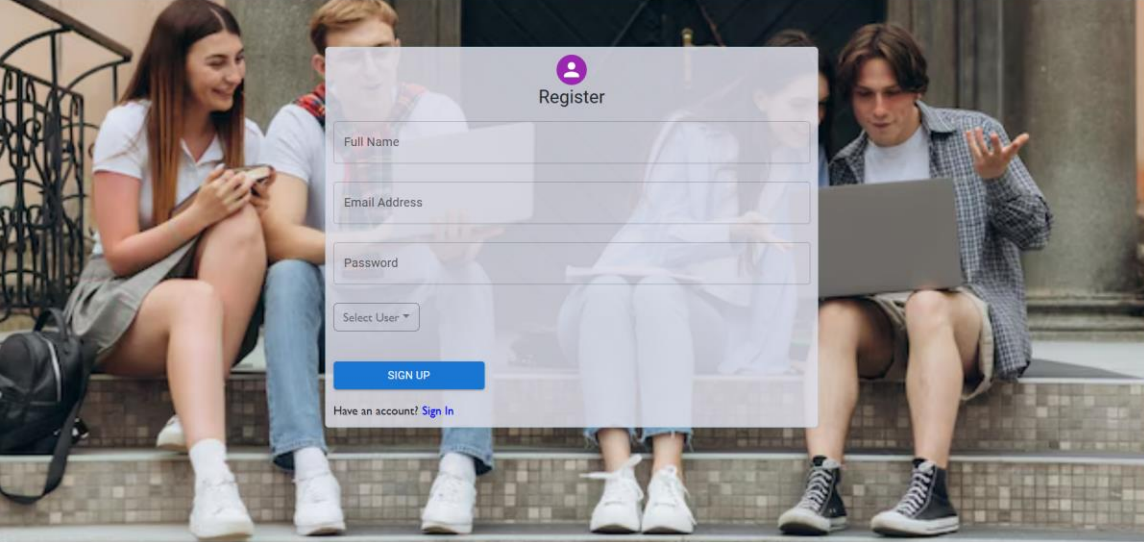
Screens Included:


- Landing Page



## - Register Page

Study App Home Login Register



 Register

Full Name

Email Address

Password

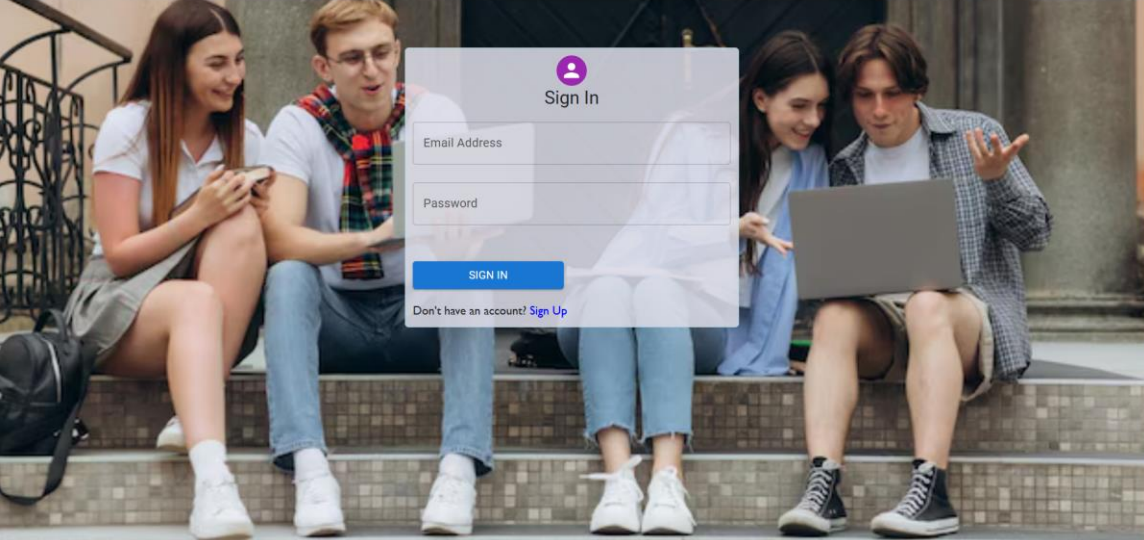
Select User ▾


SIGN UP

Have an account? [Sign In](#)

## - Login Page

Study App Home Login Register



 Sign In

Email Address

Password

SIGN IN

Don't have an account? [Sign Up](#)

- Teacher Dashboard

Study App

HomeAdd Course

Hi Teacher1234Log Out

mongodb

Description: mb

Category: IT & Software

Sections: 1

Enrolled students: 1

Delete

© 2025 Copyright: Study App

- Student Dashboard

Study App

HomeEnrolled Courses

Hi Karthikeya varmaLog Out

Available Courses

Search By:

All Courses

Modulee

1234

IT & SOFTWARE

BY: 6787

Sections: 1

Price(Rs): 7556

Enrolled students: 3

Start Course

Modules

Title:

Description:

many more to watch..

© 2025 Copyright: Study App

-Admin Dashboard

User ID	User Name	Email	Type	Action
685d962a75cdfb203bc88984	Karthikeya varma	karthikeyavarma125@gmail.com	Student	DELETE
685d967a75cdfb203bc8898b	Teacher123	teacher@gmail.com	Teacher	DELETE
685e31c682b190c05025a8b4	Karthikeya varma uppalapati	228x1a05g2@khitguntur.ac.in	Student	DELETE
685e364a82b190c05025a8c5	Teacher1234	teacher33@gmail.com	Teacher	DELETE
685e376f82b190c05025a8cf	Karthikeya varma	student@gmail.com	Student	DELETE
685ea5fb684efb9a3d484ac	Teacher1238	teacher23@gmail.com	Teacher	DELETE

## 10. Testing

Testing Tools:

- Postman for API testing
- Manual frontend testing
- Basic unit tests on backend routes (optional)

## 11. Demo

Demo Video: Link to [project-implementation.mp4](#)

## 12. Known Issues

- Limited payment gateway features
- Minimal frontend form validation
- Error handling for API responses can be improved

## 13. Future Enhancements

- Add role-based dashboards
- Implement full-featured payment gateway integration
- Improve UI/UX styling
- Add live chat and notifications
- Unit testing and test automation