Learn Hub – Seamless Appointment Booking for Health

Team Members:

TEAM ID : LTVIP2025TMID59525

MEMBER	RESPONSIBILITY
1. Kalisetti Prem Kumar	Testing & Documentation
2. Guna Gowri Nandini	Database
3. Gonuguntla Purnesh	Frontend , Backend
4. Gurajala Ameer	Resource Gathering

1. Introduction

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

2. Project Overview

Purpose:

Learn Hub is a full-stack online learning platform (OLP) that enables users to explore, enroll in, and track learning modules and skill development courses. The platform bridges the gap between learners and educators by offering a centralized hub for educational content, progress monitoring, and resource sharing.

Features:

- User Registration & Login (Students & Instructors)
- Browse courses by category, difficulty, or instructor
- Enroll in and track course progress
- Video content support with optional file uploads (PDF, PPT, etc.)
- Instructor dashboard for uploading and managing courses
- Admin panel for monitoring and approving instructor activities
- JWT-based authentication with role-based access control
- Notifications for enrollments, deadlines, and new course uploads

3. Architecture

Frontend:

Built with React.js, following a modular, scalable component-based structure:

- Component Structure: Functional components using React Hooks (useState, useEffect)
- Routing: React Router v6 for route handling
- State Management: Context API for authentication and shared state
- UI Library: Bootstrap, Material UI, MDBReact for responsive design
- HTTP Requests: Axios for communication with backend APIs

Backend:

Developed using Node.js and Express.js:

- API Design: RESTful API routes with clear modularity
- Middleware: cors, dotenv, morgan, multer, bcryptjs, jsonwebtoken
- Authentication: JWT-based token system for protected routes
- Structure: Routes, Controllers, Models, and Middleware organized for scalability

Database:

MongoDB with Mongoose ORM is used to store user data, course details, progress tracking, and uploaded content metadata.

4. Setup Instructions

Prerequisites:

- Node.js (v16.x or higher)
- npm or yarn
- MongoDB Atlas or local MongoDB instance
- Git

Installation Steps:

- **Clone the repository**
  git clone https://github.com/Purnesh4215/LEARNING_HUB
  cd LEARNING_HUB

- **Setup environment variables**
  MONGO_DB = mongodb://localhost:27017/
  JWT_KEY = my-secret-key-123-456-$
  PORT = 8000
- **Install Dependencies**
  Backend
  cd backend
  npm install

  Frontend
  cd frontend
  npm install

5.Folder Structure

BACKEND:

```
/backend
├── config/
│   └── db.js
├── controllers/
│   ├── adminController.js
│   └── userControllers.js
├── middlewares/
│   └── authMiddleware.js
├── routers/
│   ├── adminRoutes.js
│   └── userRoutes.js
├── schemas/
│   ├── courseModel.js
│   ├── coursePaymentModel.js
│   ├── enrolledCourseModel.js
│   └── userModel.js
├── uploads/
├── .env
├── .gitignore
├── index.js
├── package.json
└── package-lock.json
```

FRONTEND:

```
/frontend
├── public/
├── src/
│   ├── assets/
│   ├── components/
│   │   ├── admin/
│   │   │   └── AdminHome.jsx
│   │   ├── common/
│   │   │   ├── AllCourses.jsx
│   │   │   ├── AxiosInstance.jsx
│   │   │   ├── Dashboard.jsx
│   │   │   ├── Home.jsx
│   │   │   ├── Login.jsx
│   │   │   ├── NavBar.jsx
│   │   │   ├── Register.jsx
│   │   │   └── UserHome.jsx
│   │   └── user/
│   │       ├── student/
│   │       │   ├── CourseContent.jsx
│   │       │   ├── EnrolledCourses.jsx
│   │       │   └── StudentHome.jsx
│   │       └── teacher/
│   │           ├── AddCourse.jsx
│   │           └── TeacherHome.jsx
│   ├── App.css
│   ├── App.jsx
│   ├── main.jsx
│   └── eslintConfig.js
├── index.html
├── package.json
├── package-lock.json
├── vite.config.js
└── README.md
```

6. Running the Application

Frontend:

cd frontend

npm run dev

Runs at: http://localhost:5173

Backend:

cd backend

npm start

Runs at: http://localhost:8000

7. API Documentation

Base URL:

http://localhost:8000/api

Auth Endpoints:

- **POST** /auth/register – Register a new user (student/teacher)
- **POST** /auth/login – Login and receive JWT token

Course Endpoints:

- **POST** /courses – Create a new course (teacher only)
- **GET** /courses – Fetch all available courses
- **GET** /courses/:id – Fetch course details
- **DELETE** /courses/:id – Delete a course

User Endpoints:

- **GET** /users/me – Fetch logged-in user's profile
- **GET** /users/teachers – Get a list of instructors

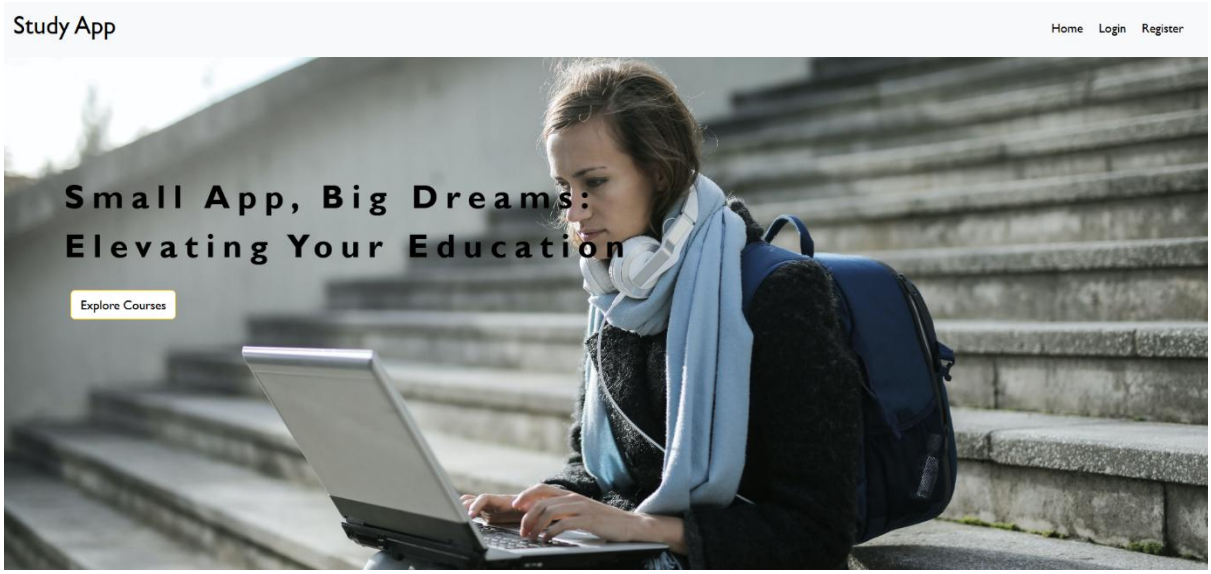
8. Authentication

Authentication uses **JWT (JSON Web Tokens)**.

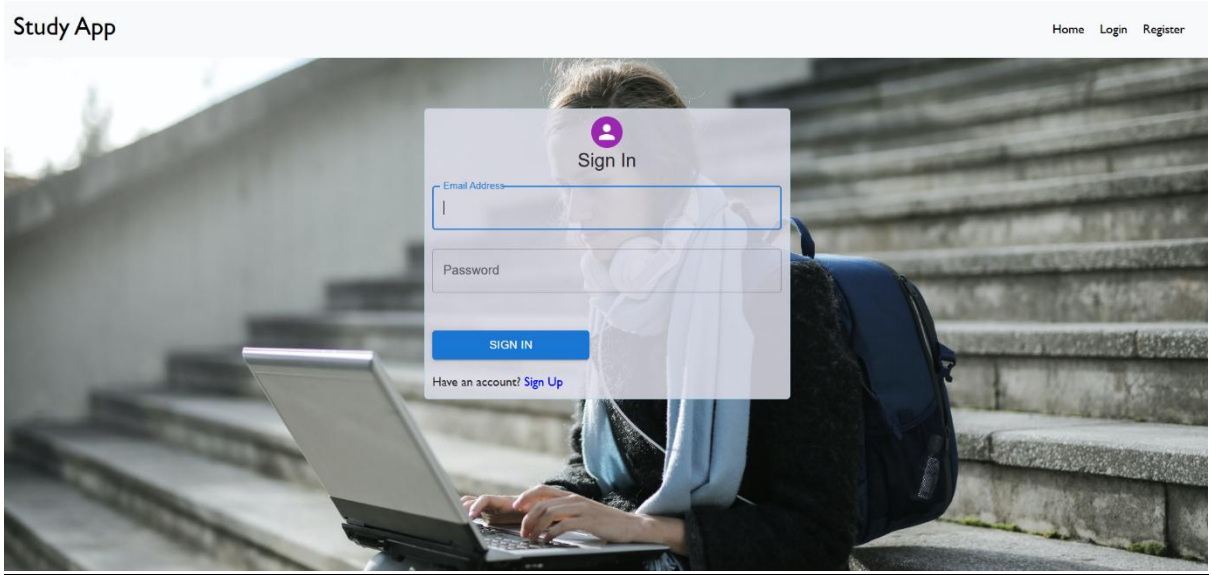
- Token is generated during login and stored in browser localStorage
- Sent in Authorization headers for all protected API requests
- Middleware (authMiddleware.js) verifies token and decodes user info
- Role-based access (student, teacher) supported via logic in backend controllers

9. Screenshots or Demo

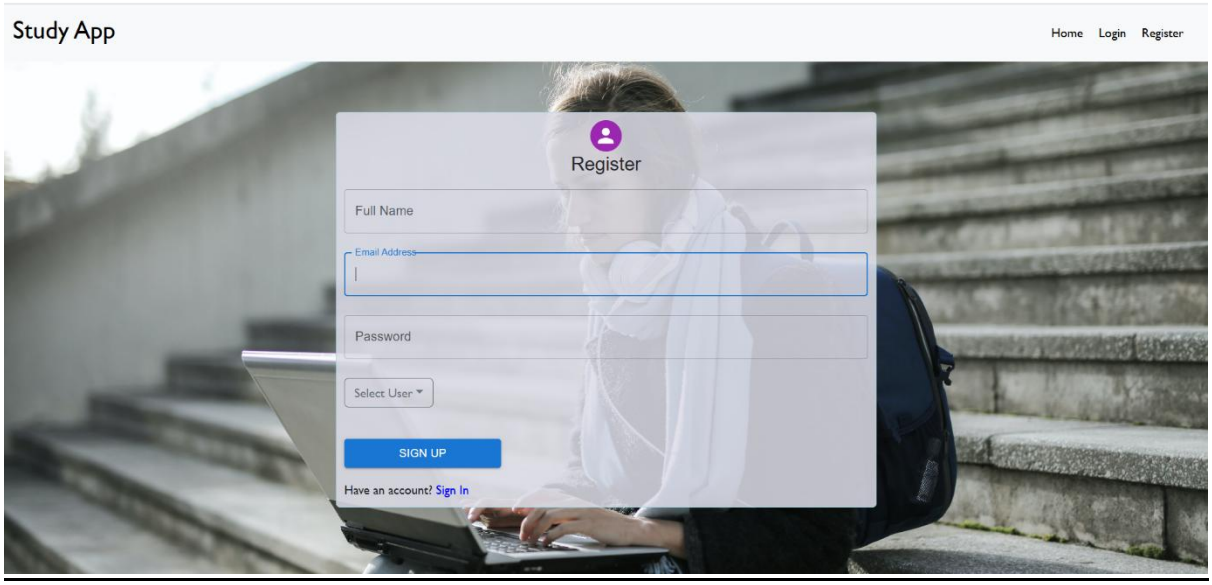
HOME PAGE :



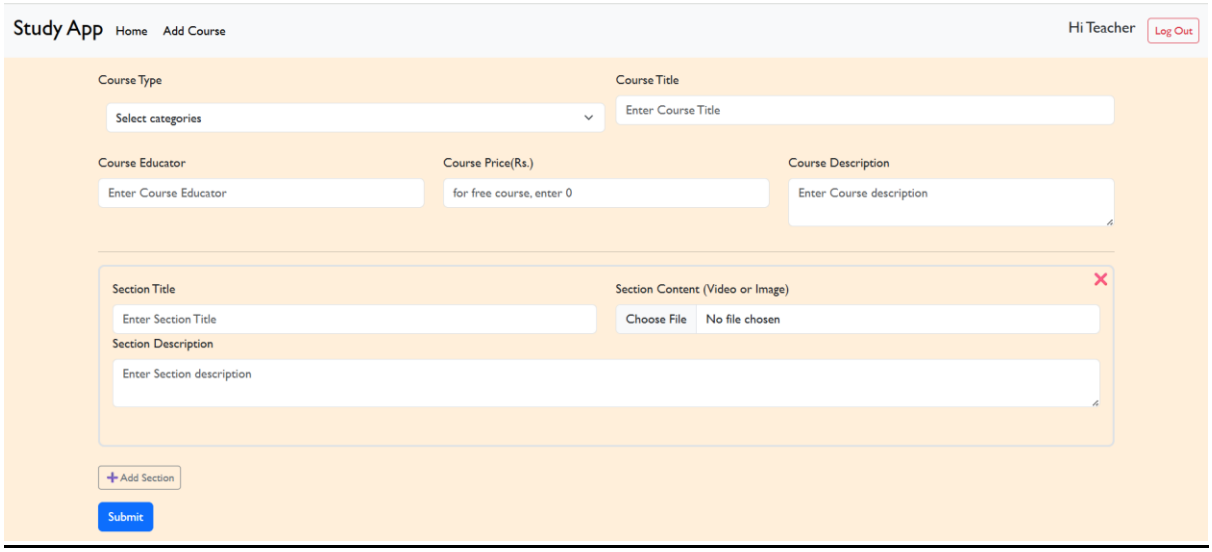
LOGIN PAGE:



REGISTER PAGE :

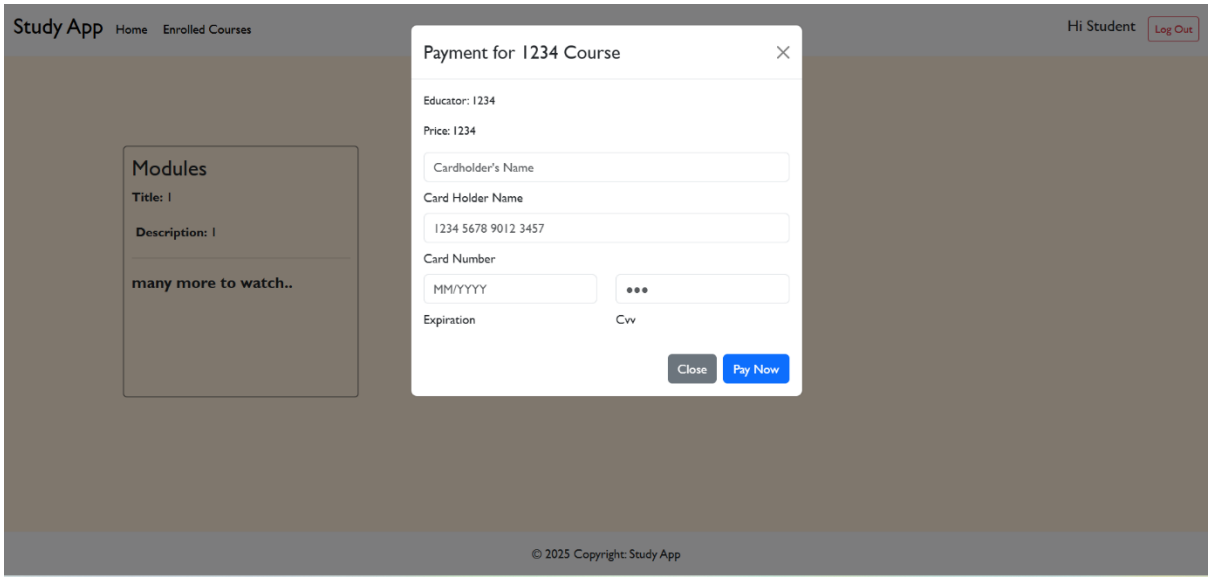


TEACHER DASHBOARD:



STUDENT DASHBOARD:

PAYEMENT PAGE:



10. Testing

Testing Strategy

Frontend:

- Tool: Jest with React Testing Library
- Scope:
    - Component rendering (NavBar, Login, CourseCard, etc.)
    - Form validations (login, register, course upload)
    - Auth flow (protected route redirection)

Backend:

- Tool: Mocha & Chai
- Scope:
    - API response testing (/auth, /courses, /enroll)
    - JWT-based auth validation
    - Error handling and status code verification

End-to-End (E2E):

- Tool: Cypress
- Scope:
    - Login > Browse > Enroll > View Progress flow
    - Role-based route access (student vs teacher)
    - Simulated real-user journey with course upload and enrollment

11. Known Issues

Course progress updates require page reload to reflect changes immediately.

File uploads are currently limited to PDF only.

Teacher approval (if admin control is added) is manual and not automated.

Dashboard tables and some layouts are not fully responsive on smaller mobile devices.

12. Future Enhancements

Integrated video sessions between instructors and students.

Course rating & reviews by enrolled students.

AI-based recommendation engine for personalized course suggestions.

Google Calendar sync for upcoming lessons or deadlines.

Multi-language support to reach global learners.

EdTech financial aid integration (scholarship & fee management).