

# Full Stack Development with MERN - Project Documentation

## 1. Introduction

Project Title: LearnHub - E-Learning Platform

Team Members:

- Thatraju Sandhya - Full Stack Developer

## 2. Project Overview

Purpose: LearnHub is an online e-learning platform built to simplify access to digital education, enabling students to learn from anywhere through structured content and seamless interaction.

Features:

- User authentication and role-based access (Admin, Teacher, Student)
- Course creation, enrollment, and progress tracking
- Responsive UI with video lectures and quizzes
- Admin panel for managing users and content

## 3. Architecture

Frontend: Built with React.js (Vite), structured with reusable components and routes.

Backend: Node.js and Express.js to handle APIs and secure authentication.

Database: MongoDB Atlas using Mongoose for schema modeling.

## 4. Setup Instructions

Prerequisites:

- Node.js (v18+), npm, MongoDB Atlas (or local MongoDB)

Installation:

```
git clone https://github.com/sandhyathatraju/LEARNHUB.git
```

```
cd LEARNHUB
```

```
# Backend Setup
```

# Full Stack Development with MERN - Project Documentation

```
cd backend
```

```
npm install
```

Create .env file with:

```
MONGODB_URL=your_mongodb_uri
```

```
PORT=5000
```

```
npm start
```

# Frontend Setup

```
cd ../frontend
```

```
npm install
```

```
npm run dev
```

## 5. Folder Structure

Client (frontend):

```
frontend/
```

```
  public/
```

```
  src/
```

```
    components/
```

```
    pages/
```

```
    services/
```

```
    App.jsx
```

```
    main.jsx
```

Server (backend):

```
backend/
```

```
  config/
```

```
  controllers/
```

```
  models/
```

```
  routes/
```

```
  middleware/
```

# Full Stack Development with MERN - Project Documentation

index.js

## 6. Running the Application

Frontend:

```
cd frontend
```

```
npm run dev
```

Backend:

```
cd backend
```

```
npm start
```

## 7. API Documentation

POST /api/auth/register -> Register new user

POST /api/auth/login -> User login (returns JWT)

GET /api/courses -> List all courses

POST /api/courses -> Create a new course

PUT /api/courses/:id -> Update a course

DELETE /api/courses/:id -> Delete a course

## 8. Authentication

Method: JWT-based authentication

Flow:

- On login/register, the server issues a token
- Protected routes verify the token using middleware
- Role-based access ensures Teachers/Admins have extra permissions

## 9. User Interface

- Dashboard view for students, teachers, and admins

# Full Stack Development with MERN - Project Documentation

- Course details and interactive content display
- Responsive layout using Tailwind CSS

## 10. Testing

Strategy:

- Unit testing for backend routes using Jest
- Manual testing of frontend components and form validation

## 11. Screenshots or Demo

Include screenshots or demo video link to showcase the application

## 12. Known Issues

- Course video upload may fail on unstable internet
- Mobile view for teacher panel has minor alignment issues

## 13. Future Enhancements

- Add real-time chat for students and instructors
- Integrate payment gateway for premium courses
- Enable dark mode and accessibility improvements