

Full Stack Development with MERN

Project Documentaion formate

1. Introduction :

.project Title: learnhub : your center of skill enhacement

.Team Member : Garapati pranathi lasya (project making
,demo,making documentation)

2. Project Overview

- o **Purpose:** Build an interactive platform where users can register, browse and enroll in courses, track their learning progress, and download certificates upon completion.
- o **Key Features:**
 - o User registration and authentication
 - o Course listing and detail views
 - o Enrollment (with optional payment)
 - o Dashboard with progress tracking
 - o Certificate access after completion
 - o Admin control panel for course/user management

3. Architecture

- **Frontend (React):**
 - o Component-based design (Navbar, Footer, CourseCard, etc.)
 - o Page-based routing (Home, Login, Register, CourseList, etc.)
 - o Axios used for REST API communication
- **Backend (Django):**
 - o Modular apps: users, courses, progress, payments, certificates, etc.
 - o Django REST Framework exposes APIs for frontend
- **Database (SQLite):**
 - o Stores user credentials, course info, enrollments, progress, and certificates

4. Setup Instructions

Prerequisites

- Node.js and npm
- Python and pip
- Virtualenv

Installation

Backend Setup

```
python -m venv djangoenv  
djangoenv\Scripts\activate # Windows  
pip install django djangorestframework  
django-admin startproject learninghub .  
python manage.py startapp users # Repeat for other apps
```

Frontend Setup

```
npx create-react-app learninghub-frontend  
cd learninghub-frontend  
npm install react-router-dom axios bootstrap
```

5. Folder Structure

```
learninghub/  
├─ users/  
├─ courses/  
├─ progress/  
├─ certificates/  
├─ payments/  
├─ adminpanel/  
└─ learninghub/ # Settings and URLs
```

```
learninghub-frontend/
```

```
├── public/
│   └── index.html
├── src/
│   ├── components/      # Navbar, Footer, CourseCard
│   ├── pages/           # CourseList, Login, Progress, etc.
│   ├── api/             # Axios service files
│   ├── App.js
│   └── index.js
```

6. Running the Application

```
bash
# Django (from project root)
djangoenv\Scripts\activate
python manage.py runserver
```

```
# React (from frontend)
cd learninghub-frontend
npm start
```

7. API Endpoints (Example)

Endpoint	Method	Description
/api/courses/	GET	List all courses
/api/courses/<id>/	GET	Get course details
/api/register/	POST	Register new user
/api/progress/	GET	User's course progress

8. Authentication

- JWT or session-based login
- Protected routes handled with React (ProtectedRoute.js)
- Backend authenticates users and returns token/session data

9. User Interface

- Built with `Bootstrap` for mobile-first design
- Navigation bar, cards, progress bars, modals, and forms
- Smooth page transitions with `react-router-dom`

10. Testing

- **Frontend Testing:**
 - Manual testing using various browsers to verify routing, responsiveness, and form behavior
 - Component-level validation (e.g., ensuring `CourseCard` renders expected props)
- **Backend Testing:**
 - Django unit tests for models and views using `TestCase`
 - API testing with tools like Postman or Django's built-in test client
- **Integration Testing:**
 - API endpoints tested with dummy data and JWT authentication
 - React pages tested against live backend to confirm end-to-end flow

11. Screenshots or Demo

<https://drive.google.com/file/d/1ZfCB3GKGHvSNZmCwnAU2dTYC9vzf0US5/view?usp=drivesdk>

12. Known Issues

- API error handling not yet consistent across all endpoints

- Protected routes (`ProtectedRoute.js`) currently use static auth check
- Form validation is basic — improvement needed for usability and security

13. Future Enhancements

- Add JWT-based authentication for secure login and session handling
- Implement search and filter functionality on the course list
- Allow users to upload avatars and edit their profiles