LEARNHUB – PROJECT PLANNING LOGIC

System Overview

LearnHub is a **MERN Stack based Online Learning Platform (OLP)** that allows:

- Teachers to create and manage courses
- Students to enroll and complete courses
- Admin to monitor and control the system

The system follows a **Client–Server Architecture**:

Frontend (React + Vite)
↓
Backend (Node.js + Express.js)
↓
MongoDB Database

Database Planning Logic

The application uses **MongoDB (NoSQL Database)** with two main collections:

🔹 **1. Users Collection**

Fields:

- _id (Auto-generated by MongoDB)
- name
- email
- password (encrypted using bcrypt)
- type (Admin / Teacher / Student)

Logic:

- type field determines role-based access.
- Password stored in hashed format for security.
- JWT used for authentication.

🔹 2. Courses Collection

Fields:

- _id (Auto-generated)
- userID (acts as foreign key → reference to Teacher)
- C_educator
- C_categories
- C_title
- C_description
- sections
- C_price
- enrolled
- Pre-requisites

Logic:

- userID links course to Teacher.
- enrolled keeps track of student enrollments.
- C_price determines if course is free or paid.
- sections stores learning modules.

Role-Based Planning Logic

Teacher Module Logic

Responsibilities:

1. Login using JWT authentication
2. Add new course
3. Add sections to course
4. Delete course (if no student enrolled)
5. Manage course content

Flow:

Teacher → Login → JWT Generated
Teacher → Add Course → Stored in Courses Collection
Teacher → Add Sections → Update Course Document

Student Module Logic

Responsibilities:

1. Register/Login
2. View available courses
3. Filter courses (by name/category)
4. Enroll in courses
5. Purchase paid courses
6. Continue from last section
7. Download certificate after completion

Flow:

Student → Login → JWT
Student → View Courses → Fetch from DB
Student → Enroll → Update "enrolled" field
Student → Complete Course → Certificate Generated

Admin Module Logic

Responsibilities:

1. View all users
2. View all courses
3. Modify course data
4. Monitor enrollments
5. Remove inappropriate content

Flow:

Admin → Login
Admin → Manage Users

Admin → Manage Courses
Admin → Monitor Enrollments

4️⃣ Backend Planning Logic

Backend uses:

- express
- mongoose
- bcryptjs
- jsonwebtoken
- cors
- multer
- dotenv
- nodemon

🔹 Backend Structure Logic

```
backend/
│
├── config/       → MongoDB connection
├── controllers/  → Business logic
├── middlewares/  → JWT authentication
├── routes/       → API endpoints
├── models/       → Database schemas
├── index.js      → Server entry
```

🔹 Authentication Logic

1. User registers
2. Password hashed using bcrypt
3. On login:
    - Password verified
    - JWT token generated

4. Middleware checks JWT for protected routes

🔹 Course CRUD Logic

Create → Insert document
Read → Fetch documents
Update → Modify fields
Delete → Remove document

Using Mongoose for DB operations.

Frontend Planning Logic

Frontend uses:

- React (Vite)
- Bootstrap
- Material UI
- Ant Design
- Axios
- mdb-react-ui-kit

🔹 Frontend Structure

```
frontend/
|
 ├— components/
|    ├— admin/
|    ├— teacher/
|    ├— student/
|    ├— common/
|
 ├— context/ → User state management
 ├— App.jsx
```

🔹 Frontend Logic

1. React Router handles navigation.
2. Axios connects frontend to backend APIs.
3. Context API manages login state.
4. JWT stored in localStorage.

Application Flow Planning

Step 1:

User registers (Student/Teacher/Admin)

Step 2:

Login → JWT token generated

Step 3:

Based on role:

- Student → Student Dashboard
- Teacher → Teacher Dashboard
- Admin → Admin Dashboard

Step 4:

Course creation and enrollment

Step 5:

Course completion & certificate generation

Security Planning

- Password hashing (bcrypt)
- JWT authentication
- Role-based access control
- Protected routes middleware
- CORS configuration

Deployment Planning

Frontend build using:

npm run build

Backend hosted on:

- Render / Railway / VPS

Database:

- MongoDB Atlas

Testing Plan

- Test login validation
- Test duplicate email
- Test role-based access
- Test course enrollment
- Test paid course logic
- Test certificate download

Future Enhancements

- Payment Gateway Integration
- Live Video Classes
- Chat System
- Ratings & Reviews
- AI-based course recommendation