Week 17: Major Project - EduHub (Learning Management System)

Day 1: Requirement Analysis and System Scope

Theoretical Overview

The initiation of **EduHub (LMS)** requires a comprehensive analysis of the "Problem Domain." A Learning Management System is a complex, data-driven application designed to facilitate the delivery of educational content, track student progress, and manage administrative tasks.

Core System Features

- **User Personas**: The system must support at least two distinct user roles: Instructors (content creators) and Students (consumers).
- **Course Management**: Instructors require a full CRUD interface to create, update, and organize course materials, lessons, and assignments.
- **Student Dashboard**: A personalized view for students to see their enrolled courses, progress tracking, and grade reports.
- **Persistent Storage**: All user data, course content, and enrollment records must be stored securely in a MongoDB database.

Day 2: Data Modeling and MongoDB Schema Design

Designing the "EduHub" Database

Using the MongoDB Data Model principles, we designed a non-relational schema that optimizes for high read performance, as students will frequently access course content.

Collection Structures

- **Users Collection**: Stores user profiles, hashed passwords, and roles (Admin/Student).
- **Courses Collection**: A document-oriented structure containing the course title, description, instructor ID, and an array of lesson objects.
- **Enrollments Collection**: A mapping collection that links User IDs to Course IDs, tracking the date of joining and completion status.

Day 3: State Management Strategy (Appendix B Concepts)

Defining the State Hierarchy

For a project as large as **EduHub**, we must categorize our data based on the "Types of State" outlined in the syllabus.

- **Data State**: Managing the actual educational content fetched from the **Gadget API**-style backend.
- **Session State**: Handling user authentication status (logged in vs. guest) and user preferences.
- **Control State**: Managing UI elements like open/closed sidebars, loading spinners, and active tabs in the course viewer.
- **Location State**: Using **React Router** to track where the user is within the LMS (e.g., specific lesson ID or course category).

Day 4: Backend Infrastructure with Node.js and Express

Architecture Setup

The backend for EduHub follows the event-driven, non-blocking I/O model to ensure the platform remains responsive even with multiple concurrent users.

- **Express Server**: Initializing the application with standard middleware for JSON parsing and CORS.
- **Modular Routing**: Setting up separate route files for /api/auth, /api/courses, and /api/enrollments to keep the code maintainable.
- **Environmental Variables**: Implementing a .env file to store the MongoDB connection string and JWT secret keys safely.

Day 5: Frontend Scaffolding and Routing

Building the Skeleton

- **React Project Setup**: Initializing the frontend using the standard structure found in the **"React New"** folder.
- **Primary Navigation**: Implementing a **BrowserRouter** with high-level routes for the Home page, Course Catalog, and User Profile.
- **Private Routes**: Developing a wrapper component that uses the "Session State" to redirect unauthenticated users away from the dashboard.