Week 18: LMS (EduHub) - Authentication & Admin Infrastructure

Day 1: Security and User Authentication Strategy

Theoretical Overview

Security is a critical pillar of the **EduHub** platform. We implemented a token-based authentication system using JSON Web Tokens (JWT) to manage user sessions without storing state on the server. This aligns with the "Session State" concepts discussed in Appendix B of the syllabus.

The Authentication Flow

- **Password Hashing**: Before storing user data in MongoDB, passwords are encrypted using Bcrypt to ensure that even if the database is compromised, user credentials remain secure.
- **Token Generation**: Upon a successful login, the server generates a unique JWT containing the user's ID and role.
- **Client-Side Storage**: The frontend React application stores this token in localStorage or sessionStorage, attaching it to the header of every subsequent API request to verify the user's identity.

Day 2: Protected Routes and Role-Based Access Control (RBAC)

Securing the Frontend

Using **React Router**, we developed "Private Routes" to protect sensitive areas of the application, such as the Instructor Dashboard.

- **Auth Guard**: A wrapper component checks for the presence of a valid token before rendering the requested page.
- **Role Verification**: The system differentiates between "Student" and "Instructor" roles. If a student attempts to access the instructor-only /admin path, the router automatically redirects them to the home catalog.

Day 3: Instructor Dashboard - Content Management UI

Building the Administrative Interface

The Instructor Dashboard is a high-complexity interface designed for course creation and management.

- **State Management**: We utilized the useReducer hook to manage the complex state of the course creation form, which includes multiple fields for titles, descriptions, and lesson arrays.
- **Form Logic**: The dashboard features a multi-step form where instructors can input course metadata before moving to the lesson uploader.
- **Control State**: We implemented localized state to handle toggles for "Draft" vs. "Published" status of the courses.

Day 4: Backend Controllers for Course Management

Logic and Middleware Integration

The backend for the dashboard utilizes specialized middleware to ensure that only authorized instructors can modify content.

- **Authorization Middleware**: A custom function intercepts requests to /api/courses, decodes the JWT, and verifies that the userRole is "Instructor" before calling next().

- **POST and PUT Handlers**: Developed controllers to process course data sent from the dashboard, saving new records or updating existing ones in the MongoDB "Courses" collection.
- **Input Validation**: Implemented server-side checks to prevent the creation of courses with empty titles or missing descriptions.

Day 5: User Feedback and Communication State

Enhancing the Admin Experience

- **Toast Notifications**: Integrated a notification system to provide instant feedback when a course is successfully saved or if an error occurs during upload.
- **Loading States**: During API calls, the "Save" button enters a loading state to prevent duplicate submissions, following the "Communication State" best practices.
- **Error Handling**: Detailed error messages are passed from the Express backend to the React frontend, informing the instructor if a specific field fails validation.