Course Management Platform

CSCI 44092 – Enterprise Application Development

Student Number – CS/2019/016

‘

Introduction

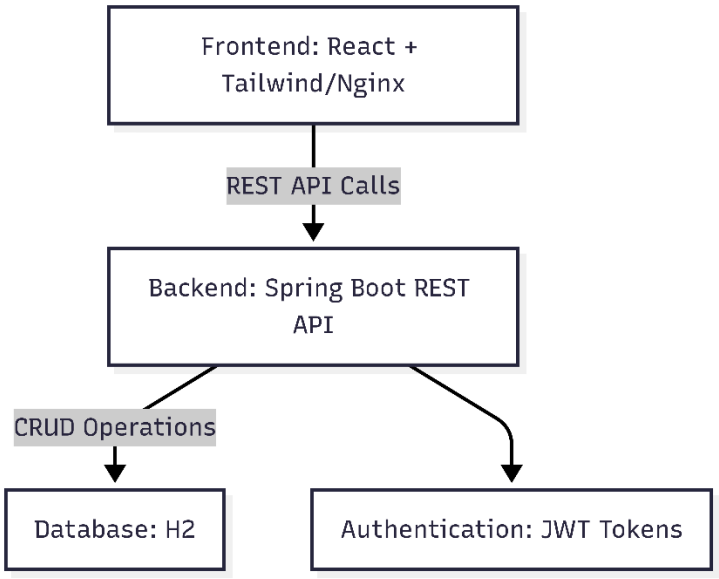
This project is a Course Management Platform that allows students and lecturers to manage courses in a simple and efficient way. It supports user registration, authentication, and role-based access, where students and lecturers can log in and access different features of the system.

The backend is built with Spring Boot, the frontend with React.js, and data is stored in an H2 database. The system is containerized using Docker and Docker Compose, making it easy to deploy and run. This platform provides a foundation for managing academic courses and can be extended with more advanced features in the future.

System Architecture

The system follows a three-tier architecture with frontend, backend, and database components, all containerized using Docker. The frontend and backend communicate via REST APIs, while the backend handles business logic, database operations, and user authentication using JWT tokens.

Architecture Diagram



Component Details

1. **Frontend**
    - Built using React for interactive and responsive user interfaces.
    - Tailwind CSS is used for styling.
    - Served via Nginx inside a Docker container.
    - Sends HTTP requests to the backend REST API for retrieving or modifying data.

2. **Backend**
    - Developed using Spring Boot as a REST API server.
    - Processes requests from the frontend and executes business logic.
    - Handles authentication using JWT tokens:
        - On login, the backend generates a token and returns it to the frontend.
        - Subsequent API requests include the token for user validation.
    - Interacts with the database for CRUD operations.

3. **Database**
    - H2 Database with files for permanent storage.
    - Stores courses, user accounts, and other persistent data.
    - Accessible via the backend only, frontend interacts indirectly through REST APIs.

4. **Deployment**
    - The entire system is containerized with Docker, and services are orchestrated using Docker Compose.
    - Ensures a consistent environment across development, testing, and production.
    - Each component runs in its isolated container for better modularity and maintainability.

Test Users

The system comes with pre-created user accounts for testing purposes. These accounts allow you to experience the system as both a student and a lecturer.

Email	Role	Username	Password
john@example.com	STUDENT	john	1234
sam@gmail.com	STUDENT	sam	1234
jane@gmail.com	LECTURER	jane	1234
hana@gmail.com	LECTURER	hana	1234

- Use any of the above email addresses and their corresponding password to log in.
- These accounts are intended for testing and demonstrate the features available to different roles.