Software Design and Engineering

Lab Document

High Level Purpose Statement:	I am developing a prototype for an optimized course scheduling system to help students graduate on time. My focus is on setting up a backend using Spring Boot with a PostgreSQL database and creating a RESTful API. Today, I made significant progress by resolving dependency issues, refining the database configuration, and implementing basic login functionality. However, I encountered challenges related to CORS, missing modules, and PostgreSQL syntax errors.
Experimental Design:	Experimental Design The project involves creating a Spring Boot application in IntelliJ IDEA , configuring it with necessary dependencies, and integrating it with PostgreSQL. The process includes: Designing a basic course entity to represent course data. Implementing a repository layer for database interactions using Spring Data JPA. Building a service layer to encapsulate business logic. Creating a controller to expose RESTful endpoints for managing course data.
Resources Available:	Spring Boot : For building the backend and exposing RESTful APIs. PostgreSQL : As the relational database for storing course data. Maven : For managing project dependencies. Online Documentation & Tutorials : Including Spring Boot and React documentation, YouTube tutorials, and Stack Overflow for troubleshooting.
Time Estimate:	The project took definitely more 10 hours, over 3 days , including: Setting up the Spring Boot project and learning Maven. Configuring PostgreSQL and resolving connection issues. Designing the entity, repository, service, and controller layers. Testing the application using Postman.
Experiment Notes:	Setting up the project in IntelliJ IDEA was straightforward using Spring Initializr . Configuring PostgreSQL required careful attention to database credentials and connection settings. Encountered minor issues with dependency management in Maven, which were resolved by updating the pom.xml file. Implementing the repository pattern simplified database interactions significantly.

	Testing with Postman confirmed that the API endpoints were functioning as expected.
Results:	Successfully created a functional prototype of a course scheduling system. The backend exposes RESTful APIs for managing course data. The application interacts seamlessly with the PostgreSQL database.
Consequences for the Future:	This project lays the groundwork for a more advanced scheduling system. Future improvements will include: Integrating a **React frontend** for a user-friendly interface. Enhancing the scheduling logic to optimize course recommendations based on student preferences and graduation requirements. Revisiting security features to implement proper authentication and authorization mechanisms if needed. Optimizing database queries for better performance and scalability.