EPICODE

Backend Programming

Project Overview

You are required to build a complete backend application using Spring and PostgreSQL, demonstrating your ability to design and implement robust server-side features, including request handling, data persistence, validation, authentication, business logic structuring, and seamless interaction with the underlying database and external services when needed. The project should be fully functional and well-structured, showcasing good coding practices

This practical project accounts for 50% of the final grade; the remaining 50% will be assessed through an oral examination

General Requirements

Project Theme

You are free to choose the theme of your application (e.g., an e-commerce store, a task manager, a social media dashboard, a movie database, etc.)

General Requirements

Entities

The application must include a domain model with at least eight tables, designed through coherent and meaningful relationships, and containing at least one inheritance structure that justifies a hierarchy within the domain

General Requirements

User Requirements

The application must include a complete user management system. Each user must have an email, a password, and a profile image that can be updated after registration. In addition to these core attributes, users must include all common personal details required by the application's domain, such as name, surname, registration date, or any additional information that contributes to a realistic and fully functional profile

General Requirements

REST APIs

The system must expose REST APIs that follow consistent principles for handling requests, responses, and errors, ensuring predictable and reliable interactions

General Requirements

Auth

The application must implement authentication and authorization based on JWT

The user model must include at least three distinct roles, each with its own permissions and access rules

General Requirements

Queries

Queries must be implemented to retrieve and manipulate data efficiently. These should include filtering, sorting, aggregations, and queries combining multiple conditions. JPA query methods, JPQL, or native SQL may be used

Queries should support real use cases within the application

General Requirements

Error Handling

The project must validate all incoming data and handle errors through structured and meaningful responses. The overall application should behave reliably and present consistent patterns for both expected and unexpected situations

General Requirements

3rd Party APIs

The backend must interact with at least two third-party APIs. The retrieved information must be incorporated meaningfully into the system and contribute to the application's internal logic or exposed functionality

General Requirements

Supporting Material

The project must be hosted on GitHub, including everything needed to run the application and a clear README.md explaining: project overview, running instructions, environment variables needed, features, etc.

General Requirements

Supporting Material

Students must also include a Postman collection in JSON format, containing all the requests needed to test every implemented feature. Each request must include example payloads, parameters, headers, and every detail required for immediate use

Any functionality not represented in the Postman collection will not be evaluated during grading

Project Overview

❌ All general requirements are mandatory. Failure to meet these requirements will result in penalties in the final evaluation

❌ Penalties may apply if security principles or best practices illustrated during the course are not followed

✅ Optional features can be implemented to enhance the project to gain extra points. These may include integration with additional third-party APIs not covered during the course, a dedicated section of the application accessible through GraphQL, the creation of particularly complex or optimized queries or other extensions that add meaningful functionality to the system

Good luck and happy coding! 🚀