Learning Navigator

Problem Statement

Develop a **RESTful API** service using **Spring Boot** to manage the exam enrollment process for a Learning Management System (LMS). You are required to use **MySQL** to persist the data.

Problem Description

The exam registration service is a critical component of a Learning Management System. Generally, exam registration requires thorough Authentication and Authorization. For this assessment, your task is to develop a simplified version of the exam registration service that meets the specified requirements below.

Requirements

- The **API** must handle CRUD operations for Students, Subjects, and Exams
- Each Student has the following fields:
    - Student Registration ID (Unique Identifier)
    - Student Name
    - List of enrolled Subjects
    - List of registered Exams
- Each Subject has the following fields:
    - Subject ID (Unique Identifier)
    - Subject Name
    - List of registered Students
- Each Exam has the following fields:
    - Exam ID (Unique Identifier)
    - Subject
    - List of enrolled Students
- The entities must use **Foreign Key** relationships wherever necessary
- Students can register for the exam only after enrolling in the corresponding subject
- Handle common errors gracefully and return **appropriate HTTP codes** (Ex. 404, User not found)
- Use **GlobalExceptionHandler** and **@ControllerAdvice** to organize and streamline Exception Handling
- Include basic unit tests while making use of **MockMvc** and **Mockito** (Minimum 3)

Additional Requirement

Easter Egg Feature

- In software development, an "Easter egg" refers to a hidden feature, message, or joke intentionally inserted by the developers into the software.
- These Easter eggs are typically meant to be found by users who explore the software thoroughly or stumble upon them by chance.
- **Your task is to introduce an easter egg feature using the Numbers API to generate random facts about numbers.**
- This feature must be triggered whenever a user sends a GET request to a hidden endpoint.
- The endpoint is defined in the “Endpoints” section below.
- You will have read through the Numbers API documentation to achieve this feature.

Endpoints

- Design RESTful endpoints based on the requirements
- You can use **Spring Data REST** to streamline this process if required
- POST /exams/{examId} - Registers a student for the specific exam
- **Easter Egg Feature:**
    - GET /hidden-feature/{number} - Generate a fact about the number which is passed as the path parameter

Publishing and Documentation

- Publish your code to a public **GitHub** repository
- Write meaningful, **incremental** commit messages
- Include a descriptive **README.MD** for your application codebase
- Do not include the easter egg feature in your documentation
- Create and add a public **Postman Collection** in the README.MD (Optional)

What to Submit?

- You will be submitting your GitHub code repository for this assignment.
- Note: An activity will be part of your program to collect this submission.

Additional Resources

- Local Environment Setup - Backend - For setting up your local environment **(Added section on macOS MySQL Workbench installation)**

- Setting Up Applications Using Spring Initializr - To learn about generating boilerplate code with Spring Initializr, adding dependencies, integrating databases, and Spring Boot best practices **(Added section on MySQL integration)**
- Template for Backend Takehomes
- What is a Foreign Key Constraint? Understanding Primary & Foreign Keys
- Spring Boot Global Exception Handler | by Ahmet Emre DEMİRŞEN | Medium
- Make sure to initialize a new repository for every project on GitHub. Use one of the below for the necessary steps:
    - Installing Git and Creating a Repository OR
    - How to Add a New Project to GitHub Repository with Visual Studio Code
- Postman Collections - Getting Started and Postman Collections - Learning More
- Basic writing and formatting syntax for README.MD and Markdown Cheatsheet
- If you are new to building Spring Boot Applications using Spring Data JPA and MySQL, you can take a look at this project:
  Spring Boot Project | Banking Application using Spring Boot 3, Spring Data JPA (Hi…