University Spring Boot Project

Introduction:

You should create a Maven Spring Boot project with the Spring Initializr (using Java 17) with the following dependencies:

- Spring-Boot-Devtools
- Lombok
- Spring Web
- Thymeleaf
- Spring Data JPA
- H2-Database
- Spring Boot Actuator

Project Requirements

You will create a project, representing a class management system for a university.

There are **3 Departments**: IT, Physics and Mathematics
In each department there are **4 courses** that are taught:

💻 IT

- **Software Development** (*Mandatory, 20 Credits, Duration 50 hours*)
- **Object oriented Programming** (*Mandatory, 30 Credits, 60 hours*)
- **Computer Architecture** (*Optional, 10 Credits, 30 hours*)
- **Algorithms and Data Structures** (*Mandatory, 20 Credits, 40 hours*)

🔬 Physics

- **Introduction to Mechanics** (*Mandatory, 25 Credits, Duration 40 hours*)
- **Astronomy** (*Mandatory, 30 Credits, Duration 50 hours*)
- **Quantum Physics** (*Optional, 10 Credits, Duration 25 hours*)
- **Theoretical Physics** (*Optional, 20 Credits, Duration 30 hours*)

📈 Mathematics

- **Algebra** (*Mandatory, 30 Credits, Duration 60 hours*)
- **Information Theory** (*Mandatory, 20 Credits, Duration 45 hours*)
- **Probability and statistics** (*Mandatory, 15 Credits, Duration 40 hours*)
- **Dynamics** (*Mandatory, 20 Credits, Duration 35 hours*)

Further requirements:

1. Each course is taught by a professor. There are 3 professors in each department.
2. Each student must enroll to mandatory courses and can choose some optional ones but must have at minimum of 70 credits to pass the semester.

Create a Spring boot project with a **h2 database**.
Create the tables and import the students/courses/professors via an .sql file in the h2.
Create all the required spring components (entities, repositories, services etc.) to manage the university system.

Create a **GUI using Thymeleaf**, it should have at minimum:

1. A page for each course presenting all the information regarding the course, such as credits, duration and which students have enrolled to the course

2. A page for each student with his personal details and information such as enrolled courses, total credits, total hours of courses and if the student is passing the semester

3. A page for each department including the courses.

4. Each of these pages should link to each other.
   e.g. The Physics department page should show all the courses and when the user clicks on Astronomy should be redirected to the Astronomy course page and so on.

Note: After completing the project, write the appropriate amount of **JUnit tests** (goal is to cover all the logic and the different cases in each part).

Cloud:

Useful Infos ---> Cloud.pdf

- After completing the University Project and you have it running locally you can start with the containerization of the app
- Start by building an image to the local registry using a generic base image, such as alpine for instance
- Build the image using a Dockerfile
- After building the image, create the deployment YAML file using this image
- Start the local cluster with minikube start (after some time it should start without any errors)
- Then create the corresponding Service for the backend
- When the application is deployed on the cluster, try to access it through your browser