Project scoring condition:

- • No compilation errors present
- • Implement the given requirements
- • Clean code & Readability - try to follow DRY principle (DRY = Do not repeat yourself)
- • The code follows JAVA coding styles and conventions.
- • All tests passed

I. Define a system of your choice.

1. **Define 10 business requirements for the chosen business domain**
2. Prepare a document based on the 10 business requirements containing a description of 5 main features your project should contain for the MVP (minimum viable product) phase.
3. Create a repository in GitHub for your project and commit the document for review.

II. The project should consist of a Spring Boot Application containing:

1. **Beans for defining repositories** One repository per entity.
2. REST endpoints for all the features defined for the MVP. You should define at least 5 endpoints.
3. Beans for defining services (implementing business logic). One service per feature. Define at least one interface implemented by minimum two classes.
4. The data within the application should be persisted in a database. Define at least 6 entities that will be persisted in the database, and at least 3 relations between them.
5. For any configuration, you should take the properties from the application.properties file. Do not hardcode any string, use a constants class to define all the strings as static final and use them within the application.
6. Write unit tests for all REST endpoints and services and make sure all passed.
7. Document the functionalities in the application such that anyone can use it after reading the document. Every API will be documented by Swagger. You can also export the visual documentation and add it to your main document presentation.
8. The functionality of the application will be demonstrated using Postman or GUI (Thymeleaf, Angular, etc).

Very important:
- **Define custom exceptions,**
- **Handle exceptions in a ControllerAdvice,**
- **Each endpoind should return a ResponseEntity<YourObject> with relevant statuses**
- **Define custom validations for fields(ex: only letters, only numbers etc.)**