Here's a simple Spring Boot mini-project question It involves CRUD operations, validation, exception handling, and MySQL integration — perfect for beginners:

Spring Boot Assignment: Sales Data Management System

© Problem Statement:

You are working as a backend developer for a **Software Company**. The company wants to keep track of its **product sales data**. Your task is to build a simple **Spring Boot REST API** to manage sales records.

Requirements:

- 1. Create a Sale entity with the following fields:
 - id: Long (Primary Key)
 - productName: String (not null, minimum 2 characters)
 - customerName: String (not null, minimum 2 characters)
 - saleAmount: Double (must be greater than 0)
 - saleDate: LocalDate (must not be in the future)
- 2. Implement the following endpoints:
 - o POST /sales Add a new sale
 - o GET /sales Get all sales
 - GET /sales/{id} Get sale by ID
 - PUT /sales/{id} Update an existing sale
 - DELETE /sales/{id} Delete a sale
- 3. Add **Validation** using JSR-380 annotations (e.g., @NotBlank, @Size, @Min, @Past0rPresent, etc.)
- 4. Add Global Exception Handling:
 - Handle invalid input and return proper HTTP error messages

- o Handle resource not found when fetching/deleting a sale
- 5. Use **MySQL** as your database.

M Bonus:

• Add an endpoint: GET /sales/total that returns the **total amount of sales** made.

📚 What You'll Learn:

- Spring Boot CRUD
- REST API development
- Bean validation
- Exception handling
- MySQL integration