Day 3

?? Day 3 ? Apply CQRS and Validation

Objective

Refactor the existing codebase to adopt the Command Query Responsibility Segregation (CQRS) pattern. Introduce a validation layer for both commands and queries to ensure proper data integrity and structure.

Tasks

1. Apply the CQRS Pattern

- Introduce the CQRS pattern to separate reads (queries) from writes (commands).

- Create distinct classes for each command and query related to products and orders.

- Use a mediator library to dispatch commands and queries.

- Ensure existing endpoints are updated to use CQRS handlers.

2. Organize Handlers and Models

- Separate logic for handling queries and commands into their own handler classes.

- Group handlers logically by feature (e.g., Products, Orders).

- Ensure command handlers perform all data-modifying operations, and query handlers only perform reads.

3. Introduce Validation

- Add validation for each command and query using a validation library.

- Ensure validation is performed before the handler logic is executed.

- Include meaningful validation messages for common scenarios:

Day 3

- Missing or invalid input
- Invalid references (e.g., product not found)

4. Test Updated Endpoints

- Ensure all endpoints function correctly with the CQRS structure.
- Verify that validation is triggered appropriately.
- Ensure the structure is consistent and maintainable for future growth.