Senior Java Developer - Take-Home Assignment

Overview

You are tasked with building a **RESTful microservice** for managing **Products** in an inventory system.The service will be **Dockerized** and use **PostgreSQL** for persistence.

This exercise is designed to evaluate:

- **Java & Spring Boot** skills
- **REST API design** and basic **database operations**
- **Docker** usage and basic containerization
- **Code quality & documentation**

The task should take **no longer than 4–5 hours**.

Requirements

1. Functional Requirements

Build a **Product Management Service** exposing the following REST endpoints:

Endpoints to Implement

- `POST /products` – Create a new product
- `GET /products` – List all products
- `GET /products/search?name=...` – Search products by name (case-insensitive)
- `PUT /products/{id}/quantity` – Update product quantity
- `DELETE /products/{id}` – Delete product
- `GET /products/summary` – **Get inventory statistics**, including:
  - Total number of products

- Total quantity of all products
- Average price
- List of out-of-stock products (quantity = 0)

Example Response:

```
{
  "totalProducts": 5,
  "totalQuantity": 78,
  "averagePrice": 219.99,
  "outOfStock": [
    { "id": 3, "name": "Monitor" },
    { "id": 5, "name": "Keyboard" }
  ]
}
```

Example Product JSON

```
{
  "id": 1,
  "name": "Laptop",
  "quantity": 10,
  "price": 1299.99
}
```

Additional Notes

- Use **JPA/Hibernate** for persistence.
- Return appropriate **HTTP status codes** (e.g., 201, 200, 404, 400).
- Add **basic exception handling** and meaningful error responses.
- Document the API using **Swagger/OpenAPI**.

2. Non-Functional Requirements

- Application runs with **Docker Compose** including:
  - **Java service container**
  - **PostgreSQL container**
- Include a **Dockerfile** and **docker-compose.yml**.
- Provide a **README** with:
  - How to build and run the project
  - Example API requests (**curl** or **Postman**)

3. Bonus (Optional)

- Use **Testcontainers** for integration tests.
- Implement **pagination** for GET /products.
- Add **basic unit tests** for service or repository layer.

Deliverables

- **Source code** in a GitHub repository or ZIP file.
- **Dockerfile** and **docker-compose.yml** for the app and database.
- **README** including:
  - Setup instructions
  - How to run the application
  - Example API requests