Java Assignment

Background

You're building a **Product Catalog Microservice** for a retail company. This service will manage product information and publish events when products are added or updated. The service will be deployed in a cloud-native environment.

Tech Stack Requirements

- **Java 21**
- **Spring Boot**
- **PostgreSQL** or **MongoDB**
- **Maven**
- **Pub/Sub (GCP)** — use emulator or mock if not on GCP
- **GitLab** for version control
- **Optional**: RabbitMQ, PL/SQL, Docker, GKE

Assignment Tasks

Core Functionality

1. **Create REST APIs** using Spring Boot:
   - POST /products – Add a new product
   - PUT /products/{id} – Update existing product
   - GET /products – List all products
   - GET /products/{id} – Get a product by ID

2. **Product entity** must include:
   - productId (UUID), name, description, category, price, availableStock, lastUpdated

3. **Data persistence**:
   - Use JPA for PostgreSQL or Spring Data for MongoDB.

❖ Event-Driven Design

1. On **create/update**:
   - ➢ Publish an event to a **GCP Pub/Sub topic** (product-events)
   - ➢ Event structure:

json

CopyEdit

```
{
  "eventType": "PRODUCT_CREATED" | "PRODUCT_UPDATED",
  "timestamp": "2025-04-23T12:00:00Z",
  "product": { ... }
}
```

2. **Optional**: Add RabbitMQ support as an alternative to Pub/Sub.

❖ Cloud & Deployment

1. **GCP Integration**:
   - ➢ Simulate Pub/Sub locally (use emulator or pubsub mock library).
   - ➢ Optionally dockerize the application and prepare for deployment on **Cloud Run**.

Deliverables

- ➢ Source code (hosted on GitLab or GitHub)
- ➢ README with:
  - Setup instructions (local DB, how to run)
  - Pub/Sub setup (mock or emulator)
  - API usage examples (curl/Postman)
- ➢ (Bonus) Dockerfile and sample docker-compose.yml
- ➢ (Bonus) Unit tests (JUnit + Mockito)

Evaluation Criteria

- Clean architecture and code quality (use of services, DTOs, etc.)
- Knowledge of Spring Boot best practices
- Proper use of JPA or MongoDB queries
- Correct integration with Pub/Sub
- API documentation (Swagger/OpenAPI preferred)
- Git commit hygiene and project structure