

Assignment 1: Event-Driven Microservice

Task:

Build a microservice that simulates a simple ticket booking system for a hypothetical event (e.g., a concert or conference).

1. Requirements:

- Use **Python (FastAPI)**, **Java (Spring Boot)**, or **Go** to implement the service.
- Implement RESTful APIs for:
 - **Booking a ticket**: Users can book tickets for an event.
 - **Cancelling a ticket**: Users can cancel bookings.
 - **Fetching user bookings**: Retrieve a user's booking details.
- Use an event-driven architecture:
 - When a ticket is booked or canceled, emit an event to a message queue (e.g., Kafka, RabbitMQ, or a mock implementation).
 - Process events to update ticket availability in a separate service.
- Store data in both SQL (e.g., PostgreSQL) and NoSQL (e.g., MongoDB or Redis) databases.
- Dockerize the application and ensure it can be deployed in a Kubernetes cluster.
- Write unit and integration tests for key components.

2. Key Evaluation Criteria:

- Code structure and readability.
- Proper use of microservices architecture.
- Implementation of event-driven communication.
- Effective database design and usage.
- CI/CD readiness with Docker and Kubernetes.
- Quality of test coverage and documentation.

3. Optional Extensions:

- Integrate authentication and role-based access control (e.g., Admin vs. User roles).
- Implement rate limiting for APIs.