Code Review of Sudais Ballah's Week 6 Lab

By: Elias Adjetey

✅ What you did well:

- The project correctly uses a microservice architecture with logical service boundaries.
- The core infrastructure, including the discovery-server, config-server and api-gateway is set up correctly.
- The services correctly use DTOs for data transfer. Also, DTOs are properly annotated to validate inputs.
- The project implements a global exception handler which handles exceptions accurately including custom exceptions.
- The use of kafka for asynchronous communication is well implemented demonstrating solid understanding of event-driven architecture.
- The addition of refresh token mechanism in auth-service is commendable as it improves session longevity and user experience.

⚠️ Areas you can improve:

- The register-admin endpoint in the auth-service is publicly accessible, allowing anyone to create an administrator.
  Action: The endpoint must either be removed or secured to require existing administrative privileges for access.
- The restaurant service layer lacks transactional management which can lead to data inconsistency if an operation fails midway.
  Action: All service methods that perform database write operations must have the @Transactional annotation.
- The jwt secret is hardcoded in plaintext in the config-repo's auth-service.properties file which is a critical security vulnerability.
  Action: Hardcoded sensitive data must be replaced with environment variables.
- While asynchronous communication worked, some message processing logic could be made more defensive.
  Action: Incoming events must be verified to ensure data integrity.

💡Suggestions for future work:

- Add Resilience4j circuit breakers and fallback methods to handle service failures gracefully.
- Integrate distributed tracing e.g. Zipkin for better observability.
- Explore containerization with Docker compose to simulate deployment in a production-like environment.