Names: George Adriane Dean P. Salise
Teaching Facilitator: Sir. RB Angelou Cobosa
Course and Year: BSIT 2nd Year
Date: March 14, 2025

Activity Instructions: Building Microservices with Database Migrations and GraphQL CRUD Endpoints (Reflection)

1. What do database migrations do and why are they useful?
   Database migrations create a template for creating databases. Not only does this allow developers to recreate the entire database to a new one with relative ease, but also create copies of the database templates. This is particularly useful since databases naturally evolve over time as the system adds, updates, and removes entire rows, columns, or even tables. This system allows them to rollback to previous versions of the database in the event something goes wrong or the change is deemed unnecessary. Lastly, this allows developers to test new features of the system safely without using the main database as they are able to share the artifacts/files created by the migration.

2. How does GraphQL differ from REST for CRUD operations?
   What makes GraphQL different from REST is how it doesn't rely on multiple endpoints to fetch or modify data as it only makes use of a single endpoint to do the same job. This is because, unlike REST's resource-based system, GraohQL makes use of a query-based system wherein clients are able to specify their data requirements. This also means that GraphQL does not suffer from overfetching and underfetching which not only improves the system's performance but also makes it easier to gather the client's data requirements. This also means that it is easier to modify the existing data structure which reduces the need for versioning. Lastly, GraphQL is capable of sending real-time updates to the client through the subscription system only when updates do occur in the system instead of having to periodically poll the server for updates.