Michael Xavier Canonizado
BSCS 3A
Software Engineering 2 - Async Task 2

ShopMaster launched its online retail platform in 2012. Like many startups, they built a monolithic application. All business logic, customer management, product catalog, and payment processing lived in a single codebase and database.

Over time, this architecture led to several pain points:

- Slow Development Cycles: Even minor changes required rebuilding and redeploying the entire application.
- Scaling Issues: Traffic spikes (like Black Friday) would overload the whole system, not just the checkout process.
- Deployment Risks: A bug in one feature could bring down the whole platform.
- Onboarding Friction: New developers struggled to understand the massive, intertwined codebase.

As the company grew, innovation slowed and operational costs soared. Leadership recognized the need for a more flexible, scalable solution.

Based on the given scenario, answer the following questions:

1. ShopMaster struggles with various pain points. How could breaking the monolith into smaller, independent services improve scalability and deployment reliability?
2. What architectural alternatives could ShopMaster consider?
3. Considering operational costs, infrastructure, and developer time, how would you evaluate whether migrating to a new architecture is worth it?
4. If you were a startup in 2026, would you still start with a monolith?

1) Breaking down the monolith into individual services helps for multiple reasons. Firstly, each service now scales independently, therefore only high-load services can be given extra resources and the service can use technologies that can improve its performance. Secondly, independent services can be deployed independently, this reduces the risk of the whole system going down when a particular service/module crashes in a monolith system.
2) ShopMaster should not immediately migrate to a full microservices unless they need the performance and have the manpower to do so as it is a big overhead to the company. They can initially switch to a modular monolith first, where they get some benefits of a microservice, then in the future, when they can easily switch to a microservice.
3) Migration to a new architecture is primarily a business decision, they should evaluate the cost factor, their manpower(developers), their current operational metrics, and if it will affect ROI.
4) If I were a startup in 2026, I would start the software as a monolith. As a startup, it is very important to prioritize the product and acquiring customers before considering a microservice. Startups won't have the manpower and funding to handle a microservice.