Course: **Guided Research I**

Title: **Scalability experiment of microservice architecture on an online bookstore application**

Weekly Progress Report 4

Student: Tural Mehtiyev

Current Status of the project:

In the past week, I made significant progress primarily on the application side of the project. Here are the main accomplishments:

Application Logic Enhancement: I revamped the logic of the application and refined its implementation to ensure smooth user experience and reliable performance. This involved reevaluating existing processes and fine-tuning them to align with current project requirements.

Functional Testing: Leveraged Postman for testing the application's functionality. This covered various use-cases and scenarios to verify the application's responses under different conditions, thereby ensuring that all functionalities perform as expected.

Database Configuration and Deployment: I configured and deployed the databases integral to the application, ensuring optimal structure and performance. This involved setting up the appropriate tables, relationships, and data flow mechanisms to support application functions effectively.

Application Deployment: Successfully deployed the application to the target environment. This involved meticulous planning and execution to ensure that the application operates seamlessly in the deployment environment.

Performance Testing Initiation: Began performance testing of the application using Apache Jmeter. This critical step aims to assess the application's robustness, speed, and reliability under varying loads and high-stress conditions. The results of this testing phase will provide valuable insights for potential optimizations and improvements.

Planned next steps for the upcoming week:

Order Management Service Bug Fixing: The past week's results have shown that there are some internal server errors cropping up in the Order Management Service. Therefore, a small portion of the upcoming week will be dedicated to identifying and resolving these issues. This will involve comprehensive testing of each part of the Order Management Service to find the root cause of these errors. Debugging will then be conducted to fix these issues, paying particular attention to any errors in the logic implementation of the service. This crucial step will ensure that all processes within the service are working correctly and the application performs as expected.

Continued Performance Testing: Continue Performance Testing of the application using Apache Jmeter. Not only will testing continue, but all results will also be meticulously recorded for future reference and analysis. For the experiment design I will follow the plan that I present in my report 3.

Statistical Analysis and Visualization: Perform an in-depth statistical analysis based on the performance testing results. This will provide insightful data to better understand the application's performance characteristics. To complement the analysis, visualizations will be prepared to make data more understandable and to effectively communicate results.

Architecture Diagram Enhancement: Work on the enhancement and refinement of the architecture diagram, improving its clarity and presentability. The focus will be on illustrating the system's design and workflows more effectively, ensuring it's ready for inclusion in the final project report.

Final Report Preparation: As the project advances towards its culmination, preparation for the final report will begin. This report will provide a comprehensive overview of the project, covering all aspects from conception to execution. Preparing this report will involve consolidating all the work done over the duration of the project into a coherent, well-structured document that accurately and thoroughly represents the project's accomplishments.

Key Decisions Made During the Previous Week and Their Main Rationale:

Deployment of the Database: I decided to use Amazon RDS for deploying all three databases, based on its promise of scalability, security, and managed database service. By deploying all three databases in a single instance, I aimed to cut costs, streamline management, and ensure high availability. Amazon RDS takes care of automatic backups, patch management, and offers built-in security at both the infrastructure and database level. This allowed me to allocate more of my resources towards improving the application rather than managing the databases.

Deployment of the Services: The decision for the deployment platform was influenced by several factors including cost, compatibility with FastAPI, ease of use, and scalability. I evaluated various options, including both on-premise and cloud solutions. Heroku was an attractive choice due to its extensive support for a variety of languages and frameworks and a robust ecosystem of add-ons. However, its pricing structure was restrictive for me. In contrast, I found Deta, a newer platform specifically designed to support Python applications and microservices. Its user-friendly interface, straightforward deployment process, and most importantly, a generous free tier made it more cost-effective and time saving for my use case.

Usage of Docker in Deployment: While Docker initially appealed to me for this project, due to its capacity to streamline dependency management and ensure consistent performance across multiple environments, I ultimately chose not to use it. This decision wasn't made lightly; in fact, I initially started implementing Docker into my project. However, considering the simplicity and smaller scale of the services involved, I found that the additional layer of Docker wasn't necessary. By not using Docker, I could also avoid additional overheads and complexities, focusing instead on the core functionality and performance testing of the services.

Deliverables from the previous week:

Product Catalog Service:

https://github.com/ADA-GWU/guidedresearchproject-tmehtiyev2019/blob/main/app/product_catalog_microservice/README.md

Shopping Cart Service:

https://github.com/ADA-GWU/guidedresearchproject-tmehtiyev2019/blob/main/app/shopping_cart_microservice/README.md

Order Management Service:

https://github.com/ADA-GWU/guidedresearchproject-tmehtiyev2019/blob/main/app/order_management_microservice/README.md