CS470 Final Reflection

Tanner W. Holbrook

August 13, 2023

Southern New Hampshire University

Dr. James Webb

Throughout this course, I have gained valuable insights and practical skills contributing to my journey of becoming a full-stack developer. For instance, one of the skills I learned is full-stack web development and integrating the front and back-end together to make a functional application. This understanding will enable me to contribute to many projects, like modern web frameworks, databases, and apps, to create dynamic and responsive web applications.

As a software developer, my main strengths lie in my ability to adapt and analytical thinking. I approach problems systematically, breaking them down into smaller components. This strength is valuable to software developers because it allows them to diagnose and address complex issues and concerns during development.

My journey here at SNHU has prepared me to take on multiple roles ranging from a front-end developer who can create a seamless user experience to a back-end engineer designing robust systems and APIs. Additionally, I have learned how to work as a full-stack developer, ensuring an efficient collaboration between the front and back-end.

When planning growth of my web applications, I will leverage the knowledge I have gained about cloud services to ensure scalability, reliability, and cost-effectiveness.

I can use the microservices architecture to enhance management and scale. For instance, I can isolate individual functionalities by breaking the application into smaller independent services, improving maintenance, and allowing for more accessible updates. In addition, error handling can be improved by incorporating redundancy and different failover mechanisms. Serverless technology can then be utilized for specific functions, reducing the need for manual resource management and ensuring automatic scaling.

To predict costs, I will analyze the usage pattern of my application and consider whether serverless architecture or containers will be more cost-effective. For instance, serverless architecture can be more cost predictable because it charges based on actual usage, whereas containers have more variable costs because of provisioning and scaling.

Pros for expansion plans include faster development cycles with serverless, efficient resource utilization with containers, and improved fault tolerance with microservices. Cons involve complexity in managing microservices, potential vendor lock-in with serverless, and operational overhead with containers.

Elasticity and pay-for-service are crucial in growth decisions. Elasticity ensures the application can handle the increased load, scaling resources dynamically. Pay-for-service models align costs with actual usage, optimizing spending.