



*Abdisalam Jimale*

*14/12/2023*

*CS 470 Final Reflection*

[https://youtu.be/qYlck\\_zWLWQ](https://youtu.be/qYlck_zWLWQ)

## **Experiences and Strengths:**

My goal in starting the Full Stack Development course went beyond just broadening my knowledge and included building a personal project, an important step towards becoming a professional software engineer. This course gave me a solid grasp of full stack engineering; at present, I am deeply involved in writing and debugging code for my personal project using skills that I acquired from the course.

***Learned Skills:*** The course has not only augmented but solidified my understanding on coding, stressing upon security and networking infrastructures. Dipping into serverless architectures has made me bold enough to talk about them especially for the project I am working on currently.

***Strengths as a Developer:*** Detail orientation, keen sense of curiosity defines my core strengths. In addition to my immediate needs for project purposes, I always code throughout, trying out new and old programming languages. The project is one aspect, while my passion extends to developing different kinds of stuffs like web apps and games.

***Roles Prepared to Assume:*** Engaged deeply in steering the course of my personal project, I am prepared to wear various hats related to full-stack development. The inherent flexibility of personal projects aligns seamlessly with my inclination towards creating innovative solutions independently. Remote work opportunities, a prospect I am keenly interested in, further resonate with my aspirations for the ongoing project.

## **Planning for Growth: Synthesis of Cloud Services Knowledge**

In anticipation of the future evolution of my web application, I've critically assessed the knowledge acquired in cloud services, focusing on the utilization of microservices and serverless architectures. Addressing key questions guides the formulation of effective strategies for scalability, error handling, cost predictions, and overall expansion plans.

***Handling Scale and Error Handling:*** The scalability and error handling of the web application are pivotal considerations for its future growth. In incorporating microservices or serverless architectures, the approach to handling scale involves implementing dynamic scaling mechanisms. Microservices, with their modular structure, allow for targeted scalability, adjusting specific components as needed. Serverless architectures, on the other hand, inherently scale with demand, efficiently managing spikes in traffic. For error handling, both microservices and serverless architectures enable centralized logging systems, aiding in identifying and troubleshooting issues across the application.

***Predicting Cost:*** Cost predictions for my project depend on the complexity of tasks; minor issues find efficient resolutions, while major changes may involve substantial costs. This approach aligns with the dynamic nature of personal projects, allowing me to adapt and reassess based on financial considerations.

***Containers vs. Serverless:*** While serverless solutions may entail management fees, the overall cost-effectiveness derived from reduced backend management aligns well with the resource constraints typical of personal projects. Managing containers demands specialized teams, making serverless architecture an optimal choice for streamlined development.

***Pros and Cons for Expansion:*** The prospect of expanding my project introduces the dual facets of increased storage capacity and the potential for resource wastage. The project's architecture is designed to accommodate anticipated demand, yet the cautionary note centers

on the inherent risk of overestimating storage needs. The dichotomy of serverless architecture, slightly costlier yet inherently elastic, emerges as the preferred approach. The flexibility offered by serverless architecture mitigates the risk of financial misjudgment and prevents unnecessary resource expenditures.

***Roles of Elasticity and Pay-for-Service:*** Thinking about the future of my project, elasticity for storage needs introduces a nuanced dilemma. Predicting future requirements involves inherent uncertainties, raising the spectre of potential financial losses. A pay-for-service model is practical. This approach ensures charges align with actual usage, eliminating stress about predicting the future. In personal projects, elasticity, despite potential advantages, may cost more over time due to resource constraints.

### **Conclusion:**

The CS 470 course has been a significant contributor to my growth as a developer, providing not just a theoretical framework but tangible skills that I am using actively in my personal project. As I handle the ups and downs of my project, the knowledge gained from the course becomes a guiding force, informing decisions and refining the trajectory of the project.

This reflection, blending my experiences with course knowledge, captures the essence of growing through Full Stack Development. My ongoing project is proof of how the course is not just about skills but also about changing how I approach development, scalability, and balancing innovation with practicality.