

CS 470 Final Reflection

Joshua Henderson

December 16, 2022

Video link: <https://youtu.be/BsZbOG4rmOI>

Experiences and Strengths

This course has taught me how to develop a cloud-based full stack application and work with AWS services like Lambda, API Gateway and S3 buckets. I had to learn how to run containers in Docker Compose, create a static website in an S3 bucket, deploy an API, create and manage a database for my full stack application, and incorporate IAM roles and policies for better security. Because I've learned such a variety of full stack-related processes, I believe that I will be a more marketable candidate in the Computer Science field.

As a software developer, one of my biggest strengths is keeping my code well-formatted and organized. I believe that code should be readable by anyone, regardless of their technical experience. To accomplish this, in this course I used comments to label and section off lines of code and indented or made spaces whenever necessary to keep my code uncluttered. Another strength I have is my problem-solving skills—something which I feel that I improved in this course. To solve problems in this course, I had success by either Googling solutions or reaching out to my classmates. The latter takes courage, as it can be difficult for one to admit that they don't understand something. My takeaway from this class is that computer scientists are human, too; sometimes, they get lost despite their best attempts to solve a problem. I demonstrated this strength by calmly describing my problems to classmates in much detail—often using images for clarity—as well as posting as early as possible.

I feel that for a future job, I'm prepared to assume a regular software developer role. This course helped me practice my ability to follow directions, ask questions, and solve problems—all skills that are important when working on a software team. The assignments may have also prepared me to assume the role of a leader. For any successful group project in the Computer

Science field, there must be a good leader who can take initiative; by pouncing on problems as soon as they occur, a software team will reach their goals without much issue.

Planning for Growth

Error handling of a serverless application could be achieved with a program like Amazon API Gateway, which presents the user with a helpful dashboard that alerts them of any errors found in the execution. The cost of running a serverless application could be predicted based on several factors, such as how long the application must run, and which cloud computing service is being used (like Amazon AWS). Serverless applications are more predictable; oftentimes, developers need only to pay for whenever an application runs, rather than the entire time that a server is active. In contrast, containers run on other services, which could be inactive, thereby wasting a developer's money.

Elasticity and pay-for-service play a large role in decision-making for planned future growth. A cloudless full stack application's ability to be *elastic*—in which it can adapt to changing demands for more or less computing power—gives developers flexibility cost-wise; it allows them to pay only for whatever resources they're using at a given time. This ensures that fewer resources (and less money) will be wasted. The pay-for-service model guarantees that a developer will only pay when a service or application is being executed. This is critical to the decision-making process, a developer will have to pay more if their application has to run constantly, as opposed to once. Outside of cost, other cons that could be deciding factors in plans for expansion include developers not having access to fast Internet, and security risks that would require extra time spent on creating IAM roles and policies. In terms of pros, the idea of bringing a full stack application to a lightweight platform is appealing; it saves up resources on a developer's computer, as a cloud computing service takes care of those responsibilities.