Tyler Pruitt
10/22/23
CS-470

CS 470 Final Reflection

Presentation link: https://youtu.be/YDFlv8PKyLg

This course has taught me so much about web development which I had never learned before. Before this course, I had no experience with any Amazon Web Services, Docker, or even containerization for that matter. This experience will make me significantly more marketable because the skills will provide me with a useful baseline for full-stack development as well as insight into other forms of web development that I did not have before. My strengths as a software developer lie in my perseverance. Once I have a vision for a project, I don't give up on realizing it easily. I have also learned that this can be a weakness as well. I can get too fixating on making everything right that I will get stuck and slow down to fix it. Within a new job, I would likely assume the role of bug fixing, where the main goal would be to solve specific problems, that way I can use my strengths and mitigate the weaknesses.

On the subject of my knowledge in cloud services, I find there are many things to consider to produce efficiencies of management and scale in my web application. For example, I must handle scale and errors. Auto-scaling does significantly well at handling database scaling, however traffic scaling isn't always so simple. This is where I would use roles, policies, load dispersal, and more to ensure the web application does not get overloaded. These are all tools I have learned about this term. Error handling can be done the same way. Using gateways, I can create responses to certain error codes that I determine are possible, such as 404, 403, and more. I can also create a default response to errors as well, to ensure all bases are covered. Cost can also be a problem to consider. Using cloud-based development, the cost would be based upon the resources I use, however I may not always know how much the application is using. Thankfully, analytics is also possible and easy to implement in cloud development. I would use analytics in an attempt to predict figure costs. Ultimately, containers will be easier to predict the costs. While there is more waste because you are paying when not using them,

their costs are consistent. Pay-as-you-go can be inconsistent and unpredictable if resource usage is volatile. Expansion should be decided upon based on use case and necessity. The pros to expansion means being able to handle more user traffic and data. Conversely, this also means more costs, more data to protect, and in many cases, more maintenance. Elasticity and pay-for-service can make a significant difference in the decision to expand. Having elasticity can mean being able to compensate for varying levels of usage without wasting money on extra available resources. The same could be said about pay-for-service, where regardless of usage, you only pay for what you use. With these cost schemes in mind, planned future growth can be much more flexible than it once was, with less waste as well!