Brigitte McColl

06/27/25

CS 470 Final Reflection

Link to presentation: https://youtu.be/8n-eZSyE-es

The main skills that I have learned in this course is with the use of Amazon Web Services (AWS). I have learned how to create databases, APIs, and Lambda functions in AWS and how to connect them. I think that knowing this helps me become a more marketable candidate because it shows that I can work with AWS and create projects using it. My strengths as a software developer is with my ability to debug and solve problems that I face. I am able to debug code and find where errors are and also create tests to be able to make sure that there is proper error handling so that if a user inputs something incorrectly they are notified, and the application doesn't break. Besides being able to fix problems on my own, I also have the ability to ask for help when needed because I can get a bit tunnel visioned on problems that I miss the big picture. So having someone else look at the work or even allow myself to take a break before returning to the code shows my ability to solve problems but also try to solve them faster if I can't see the problem right away. The role that I am prepared to assume in a new job is a back-end software developer role. I have experience working on writing tests and also finding and solving all of the errors that I found. I can also see myself in a role where I am testing code and adding in error handling and test cases to make sure my code is covered. I also see myself in roles where I do performance testing and testing before releasing code and making any changes where there are problems.

When planning for growth there are many factors to keep in mind, some of them should be predicting the cost and seeing if that price is feasible and how we can handle many people using the application at once. When it comes to handling any errors, I would make sure that there is error handling in the lambda functions that I write to make sure that the errors get caught. This will help make sure that the application is running smoothly. When it comes to talking about scaling, I would use the autoscaling feature that is built into AWS to be able to scale the application up or down as needed. One of the ways that I would predict the cost of the auto-scaling would be to use the AWS pricing calculator that gives estimates based on specific cases. This means that I can check the case that I am running to be able to see the cost of what it could be. When choosing the run either a server serverless or with containers depends more on the project we are trying to create. Containers offers a more up-front cost based on the use of resources while serverless is a pay as you go model which is great for variable workloads. So, I think choosing which one is better will depend on the project that you want to create.

When I think about the pros and cons of plans for expansion. Some of the pros are that you can make sure that your application doesn't crash when there is a higher volume of people using the application and there can be infrastructure built so that we aren't scrambling when we do get more users. A con would be the time and cost to build the infrastructure and logistics of the expansion. Another con would be that if we build this infrastructure and the application doesn't grow we could lose some cost. However, with elasticity and pay-for-service can help with some of the cons. If we can set up our application to scale up and down as needed with elasticity we can make sure that our application would be ready for any kind of growth that our application would face. With the pay-for-service this will allow us to have elasticity set up but

we only pay for what we are using keeping costs lower. So with the use of both features, setting up some future growth plans is easier and more manageable.