Jeremia Faust

CS-470 Full Stack Development II

Professor M. Shah Alam

February 26, 2023

CS 470 Final Reflection

CS 470 Project Two Presentation link

https://youtu.be/-_3ROmHlLn8

Experiences and Strengths:

What skills have you learned, developed, or mastered in this course to help you become a more marketable candidate in your career field?

This class taught me the concepts of containerization and using cloud services such as AWS to run applications such as web development. I learned about how the different services in AWS work together while converting a MEAN full stack to a serverless cloud-based program. This has introduced me to what is possible with cloud-based applications.

Describe your strengths as a software developer.

I may have little experience outside of what I have learned in college, but I have garnered the skills needed to advance as a software developer. I have become very good at debugging code although this still takes me time and with some time, I can use these skills to create new applications. I have spent many years working in manufacturing using lean methodologies. This converts well in the agile setting that many companies use today. With on-the-job training and my desire to learn more I will become an asset to any company that will hire me.

Identify the types of roles you are prepared to assume in a new job.

Computer scientist is a rather large field with many different roles that I could assume. I am not sure what role will fit me, but I have many options such as a full stack developer. I have the basic knowledge of both the frontend and backend that can be useful. This allows me to make correlations between the frontend and backend when working on a full stack application. The understanding of using MEAN and serverless cloud technologies makes me an asset to many companies.

Planning for Growth:

Identify various ways that microservices or serverless may be used to produce efficiencies of management and scale in your web application in the future. Consider the following:

- How would you handle scale and error handling?

  Using cloud technologies scaling is very easy as most services automatically scale based on the needs of the application. Using AWS, you can use AWS step functions to test the workflow. You can use step functions to create a workflow that supports error handling.

- How would you predict the cost?

  In serverless models you are priced on how much of the resources are being used. To predict how much is going to be used you will have to look at the application and how it is being used. Some sites such as Microsoft azure have calculators the help predict the costs. It is very important to set throttles and use the dashboards to monitor the costs. It has been known that an error in the program caused an unexpected outrageous bill.

- What is more cost predictable, containers or serverless?

  Containers tend to be more expensive due to it requiring more resources to run. Due to the nature of the containers you most like will be charged a fixed monthly bill. They can be scaled by increasing the computing resources. Serverless applications are event-based systems and are charged by triggered events. API calls tend to be what drives bills up. So, containers are more predictive the serverless, but each will cost less than if the company is running their own servers in the long run.

- Explain several pros and cons that would be deciding factors in plans for expansion.

  When deciding on expanding using cloud technology it is important to understand not only the great benefits but also the disadvantages. There are many benefits such as reducing infrastructure and manpower costs to data protection. The company needs to understand what could hinder the projects such as being locked in with a certain vender to having less control over how and where your data is being stored. These need to be thoroughly researched to see if it is a benefit for the company or not.

What roles do elasticity and pay-for-service play in decision making for planned future growth?

When using cloud services elasticity and pay-for-service are usually very important. There are many applications that see a rise in growth at certain times of the day or year. Instead of paying for resources that are not being used the company only must pay for what is needed. This allows for the company to plan out how they are going to grow the application because they can focus more on how the application is going to scale and less on the hardware needed.