CS-470 Final Reflection
June 2024
Michael Rutherford
My YouTube URL: https://youtu.be/9qaugsWZaNw

- **Experiences and Strengths:** Explain how this course will help you in reaching your professional goals.

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

I have learned many skills, but the most helpful skills I have learned are regarding containerization, lambdas, APIs and JSON. Those were some areas that I was lagged in, and this course was a great way to get me up to speed with what those concepts were and how to implement them.

    - *Describe your strengths as a software developer.*

I now realize that my strengths as a software developer lie primarily in my abilities to make programs that do exactly what they need to without unnecessary clutter. These days, there are many programs that are overdesigned and have feature creep, but I know when to stop working on a project. This allows my work to speak for itself on the merits of base functionality.

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

Having now done this course, I am ready to tackle basic full stack developer jobs. The types of roles I feel most comfortable in at this point in my education would be more back-end developer jobs, but this course did give me a great deal of knowledge about how to integrate front-ends and back-ends.

- **Planning for Growth:** Synthesize the knowledge you have gathered about cloud services.

    - *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?*
        - *How would you predict the cost?*
        - *What is more cost predictable, containers or serverless?*

Given that the applications I tend to design do not harbor a very large userbase, I would use cloud-hosted microservices like AWS for them. This would help them scale well, and the cost will be predictable because I will not have to pay for a lot upfront and, barring internet virality, I will not see my costs skyrocket. Of course, containers are far more cost predictable because its not pay as you go. If someone shares my website on social media and I get a million people

using it, my AWS bill would be through the roof, whereas my containerized program would simply be handling what I paid to have it do initially.

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

The pros of containerization would be that you can know pretty definitively what the costs will be upfront. You can pay for the server infrastructure and host it yourself, and that makes it very attractive to those on a budget. The downside is that your usage may exceed capacity, but at least you will not be financially penalized. The pros of cloud hosting is that you pay as you go, which works better for the smaller programs. The cons of this is that you can get hit with a large bill for sudden spikes in interest to your website.

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

As stated above, the aforementioned pay-for-service and elasticity factors harbor major pros and cons. For example, elasticity means not having to worry about scaling up as your program grows. This means you can safely deploy a website and not have to worry about the hosting logistics, which is attractive if you are the type who likes to focus on functionality first and foremost. The downside of this lies in the pay-for-service nature of it, meaning although you won't have to worry about the logistics of hosting, you will have to worry about the cost of hosting.