Abigail Santoro

CS-470

Full Stack Development II

Journal

Although this course has been demanding and difficult, it has also been tremendously gratifying. I was able to push myself beyond my comfort zone, and as a result, I've improved as a software engineer. I will be better prepared for my future work because to the abilities I've learned in this course, which increase my marketability to employers who are looking for people with such skills. I now understand several design patterns and how to use them in applications. I've always found debugging to be intuitive, but this course helped me understand the value of debugging and the reasons why it's crucial to learn how to do it well. I had a great time in this training. I was able to expand my understanding of software programming and gain a ton of new things. It also gave me the opportunity to teach other students the material we were given and give them input on how to improve their code and become better programmers. I've gained a lot of knowledge from

this course. I am now able to build dynamically generated websites and databases that are linked to the website using APIs.

I believe that my capacity for collaborative work is my best asset as a software developer. Because I have the ability to work with people of diverse skill levels, I am better able to grasp what each individual needs and how their various traits might improve teamwork. I also take very seriously the obligations I have to the business and its customers.

Scale is one of the key benefits of microservices. Writing your code as a set of distinct services is essentially what microservices are. This has various benefits, one of which is that they can be scaled independently of one another. Instead of changing anything else to manage the increased traffic, a service that requires more capacity can simply spin up a new instance to handle it. Other services that are active and not impacted by problems in other services won't be affected if a service has an error. This would be beneficial in my opinion.

My approach to handling errors will also play a role in my selection. I would need to consider this because it would be quite challenging to follow tried-and-true procedures for addressing mistakes in big volumes of code. However, if there were other methods for addressing errors, I would like to

choose the most effective one. This can be accomplished by studying the solutions of other businesses and learning the best practices from them. Cloud services come at a variety of prices. Other services have a monthly subscription price, while some are free. You determine your own unique demands to determine the annual cost of operating a service. You may simply test out cloud functionalities before you start paying for anything because they are free and offer an unlimited amount of features.

Permissions and other limits on the function can be set up in advance and readily handled using the provider's tool when using PaaS. Since there are so many unknowns associated with renting virtual server space, I believe this will be incredibly helpful because it will make managing my program and keeping track of what is being paid for. Compared to virtual servers, containers and serverless both have better cost predictability. Because they can be configured to use more resources than they actually require, virtual servers can behave very erratically. Therefore, it is recommended to spin up more virtual machines than necessary and to keep track of the resources that are being consumed. Because containers have all the code they need to operate once they start, there is less potential for errors, confusion, etc., and there is less chance that you would

overuse a resource and end up paying for something you aren't utilizing. As a result, containers are likely to have more predictable costs.

Continuous delivery will undoubtedly play a significant role in determining a company's destiny. Errors won't be passed on to customers who are already paying for your product because it makes it simple to install and test new features. Additionally, this will make it less expensive to deploy new features and easier for you to add new ones as required. It also aids in lowering the number of bugs that slip through testing. Software, hardware, and other system difficulties fall under this category. The cost of something like an elastic cloud platform will need organizations with a significant growth strategy to be extremely capable of handling it. This calls for competent management, the ability to manage increased traffic, and a thorough knowledge of the logistics involved in flying a few new employees from the company's headquarters. To make the shift easier for your staff, you must plan ahead and educate yourself on how everything operates. Due to the possibility of future problems stemming from having too much knowledge, I believe it is beneficial for newly established businesses to avoid learning too much about everything.