CS-470 Full Stack Development II – Final Reflection

Chance Roy

10/26/2024

Experience and Strengths

This course has taught me the foundational principles of developing a full stack web application in the cloud. From creating buckets and tables to managing APIs and permissions, it's opened my eyes to an exciting new career path that I can't wait to explore further.

Throughout the course, I discovered that my strengths lie in my ability to learn new skills quickly. I have a keen eye for spotting important details and can dig up reliable resources when I'm in search of answers to complex problems.

This experience has prepared me for a range of roles, from frontend and backend development to cloud and software engineering. That said, I'm now drawn toward a career in cloud cybersecurity. I believe focusing on a niche like this makes me even more valuable in today's job market, and I'm eager to specialize in securing the cloud.

Planning for Growth

Thinking about growth in the cloud has really shown me how microservices and serverless tech can be game-changers when it comes to scaling and keeping everything efficient. With serverless platforms like AWS Lambda, scaling is as simple as letting the cloud handle it, adjusting resources only when needed. This makes it easy to maintain performance even as an application grows, while services like AWS CloudWatch help keep things smooth by monitoring each component. Learning about how isolated microservices can keep errors contained has been an eye-opener too, and these insights are invaluable for building a robust app.

Then there's the cost side of things, which I hadn't thought about as much before this course. Serverless pricing is flexible, charging by request and execution time, which is ideal for variable workloads. But if traffic is high, containers, with their fixed computer resource costs, can offer more predictability, which can make a big difference in long-term budgeting. Each has its pros and cons—serverless is effortless on scaling but can surprise you with costs, while containers are predictable but need a bit more manual management.

Finally, I've come to see elasticity and pay-for-service as two huge benefits of cloud architecture. Elasticity means the app adjusts as demand does, which is perfect for unpredictable growth. The pay-as-you-go model is cost-efficient, especially when it's tied to the serverless setup. But for steady workloads, containers could offer more consistency. In the end, I could see a hybrid approach being a great choice—using serverless for flexible, high-demand tasks, while letting containers handle the steady, long-running functions. This kind of setup would keep things efficient and make sure costs stay predictable, setting things up for solid growth in the future.

Presentation

https://www.youtube.com/watch?v=dBhEn00bDGM&ab_channel=ChanceRoy