CS 470 Final Reflection

Drew Shepard

3/2/2024

Experiences and Strengths:

This course, CS 470, has been instrumental in shaping my journey toward achieving my professional goals in the software development field. Through this course, I have acquired and honed various skills that have significantly enhanced my marketability as a candidate in the tech industry.

One of the key skills I've learned is proficient API development, which is essential in creating robust and scalable web applications. Additionally, I have gained a deep understanding of cloud computing concepts, including Docker, AWS services such as S3, Lambda, and DynamoDB, as well as containerization and container orchestration. These skills not only make me versatile in modern software development practices but also align perfectly with the current trends in cloud computing, making me an attractive candidate for roles requiring expertise in these areas.

As a software developer, my strengths lie in my adaptability and problem-solving abilities. I excel in understanding complex systems and breaking them down into manageable components, allowing for efficient development and troubleshooting. Moreover, I am a strong communicator, capable of articulating technical concepts to both technical and non-technical stakeholders, fostering collaboration and understanding across teams.

In terms of roles, I am prepared to assume positions such as Full Stack Developer, Cloud Engineer, or DevOps Engineer. My proficiency in both front-end and back-end development, coupled with my knowledge of cloud services, equips me to contribute effectively to various aspects of software development and deployment within an organization.

Planning for Growth:

In planning for the future growth of my web application, leveraging microservices or serverless architecture can lead to significant efficiencies in management and scale.

For handling scale and error handling, I would implement auto-scaling mechanisms provided by cloud platforms, such as AWS Auto Scaling or Kubernetes Horizontal Pod Autoscaler for microservices, and AWS Lambda scaling for serverless functions. Additionally, I would incorporate comprehensive monitoring and logging solutions to identify and address errors as they occur quickly.

Predicting the cost would involve analyzing the application's usage patterns and estimating resource consumption. While containers offer more granular control over resource allocation, serverless functions follow a pay-per-execution model, making cost prediction more straightforward in serverless architectures.

Containers and serverless have their pros and cons. Containers offer more flexibility and control over the environment but require more management overhead. Serverless, on the other hand, abstracts away infrastructure management, allowing for rapid development and scalability but may introduce vendor lock-in and performance limitations.

Elasticity and pay-for-service are crucial factors in decision-making for planned future growth. Elasticity ensures that the application can dynamically scale resources based on demand, optimizing performance and cost-effectiveness. Pay-for-service models align costs with usage, enabling efficient resource allocation and budget management as the application scales.

Presentation:

C:\Users\drew1\OneDrive\Documents\SNHU\Full Stack Development 2\CS 470 Project Two Presentation .ppsx

Script:

C:\Users\drew1\OneDrive\Documents\SNHU\Full Stack Development 2\Mod 7 Project Two Script.docx