Sarah Hodge
June 25, 2024
CS-470 Final Reflection
https://youtu.be/bOZvcLs8EdA

Experiences and Strengths

How will CS-470 Full-Stack Development help you reach your professional goals?

CS-470 Full-Stack Development has equipped me with comprehensive skills in both front-end and back-end development. It has enabled me to understand the entire web development process, from designing user interfaces to managing databases and deploying applications. This comprehensive knowledge is crucial for my goal of becoming a proficient full-stack developer.

What skills have you learned, developed, or mastered in this course?

Through CS-470, I have learned advanced concepts in frameworks (such Node.js for server-side scripting), database management, RESTful API design, and deployment strategies (including Docker and AWS). These skills are essential for developing scalable and efficient web applications.

Describe your strengths as a software developer.

My strengths lie in my ability to conceptualize and architect complex systems, my proficiency in problem-solving and debugging, and my dedication to writing clean and maintainable code. I am also adept at collaborating within teams and communicating technical concepts to non-technical stakeholders.

Types of roles prepared to assume in a new job:

With my skills in full-stack development, I am prepared for roles such as Full-Stack Developer, Front-End Developer, Back-End Developer, or even DevOps Engineer, where my understanding of deployment strategies and infrastructure can be valuable.

Planning for Growth

Synthesis of knowledge about cloud services:

Cloud services provide scalable and flexible infrastructure, allowing applications to handle varying workloads efficiently without upfront investment in hardware.

Ways microservices or serverless can produce efficiencies:

- Scale and error handling:
    - *Microservices allow for independent scaling of components, while serverless platforms handle scaling automatically. Error handling in both cases can be managed through centralized logging and monitoring.*
- Cost prediction:
    - *Serverless can be more cost predictable as it charges based on execution time and resource usage, whereas containers require estimating resource needs and scaling accordingly.*
    - *Containers are less predictable due to manual scaling and resource provisioning. Costs can fluctuate based on resource usage and scaling needs.*
- *Pros and cons influencing expansion plans:*
    - Microservices:
        - Pros: Scalability, independent deployment, technology stack flexibility.
        - Cons: Increased complexity in managing multiple services, potential overhead in communication between services.
    - Serverless:
        - Pros: Automatic scaling, reduced operational overhead, cost efficiency for low to moderate workloads.
        - Cons: Vendor lock-in, potential cold start latency issues, limited control over underlying infrastructure.
- Role of elasticity and pay-for-service in decision-making:
    - Elasticity: Enables applications to automatically adjust resources based on demand, ensuring performance and cost efficiency.
    - Pay-for-service: Aligns costs directly with usage, promoting cost-effective resource allocation and scalability.

Conclusion

In conclusion, the CS-470 Full-Stack Development course has equipped me with the skills necessary to excel in various roles within the software development field. Understanding cloud services, particularly microservices and serverless architectures, allows me to plan for efficient scaling, robust error handling, and cost-effective operations in future web application deployments. By leveraging elasticity and pay-for-service models, I can ensure that future growth plans are not only scalable but also optimized for cost and performance.