Matthew Minton

August 21, 2025

CS 470 Final Reflection

CS 470 Project Two Presentation Matthew Minton

Experiences and Strengths

When I first started this course, I had a general understanding of software development, but CS 470 really pushed me to think in terms of cloud solutions and full-stack applications. I learned how to use Docker to run MongoDB inside containers, which taught me not only the value of containerization but also how consistent environments make development and deployment much easier. From there, I built on that foundation by working with AWS services, especially Lambda functions and API Gateway. Writing, testing, and deploying those Lambdas gave me hands-on experience with serverless computing and showed me how quickly and effectively you can create scalable solutions in the cloud.

The skills I've gained in this course will help me stand out in my career. I can confidently say that I now understand containerization, serverless architectures, and cloud deployment. I also learned how to connect cloud backends to a web application, which is an important skill for a full-stack developer.

In terms of strengths, I'd describe myself as adaptable and resourceful. I've learned how to troubleshoot when things don't work as expected, and I've developed the persistence to break problems down and keep iterating until I find a solution. This course also helped me strengthen my ability to document and present technical work, which is just as important as writing the code itself.

With these skills, I feel prepared to take on roles such as full-stack developer, backend developer, or even a cloud-focused position like a DevOps engineer. I now have the technical foundation and confidence to step into a team and contribute effectively.

Planning for Growth

One of the most valuable lessons I took from this course is how to think about growth and scalability in cloud applications. It's one thing to build an app that works for a few users, but it's another to plan ahead for when that app needs to handle thousands.

For my project, I can see microservices and serverless functions playing a big role in managing growth. Serverless is especially appealing because it allows automatic scaling functions to only run when triggered, and I do not have to manage servers directly. This makes it easier to handle spikes in traffic while keeping costs tied to actual usage. Error handling can also be improved with built-in logging and monitoring tools like AWS CloudWatch, which would allow me to quickly detect and fix issues as they come up.

When it comes to cost, containers tend to be more predictable since you're essentially paying for reserved resources. Serverless, on the other hand, is harder to predict at very large scale but is often more cost-effective in the early stages. Both have their advantages: containers are stable and portable, while serverless is flexible and efficient. Choosing between them depends on the size of the user base and the need for predictable costs.

Elasticity and pay-for-service are at the heart of planning for growth in the cloud. Elasticity means the system can automatically expand or shrink resources based on demand, so I don't have to worry about under- or over-provisioning. Pay-for-service ensures I only spend money on what I actually use, which makes growth manageable without creating unnecessary

financial strain. Together, these concepts allow me to design an application that can start small and scale gracefully as more users come on board.

Closing Thoughts

Overall, CS 470 has been one of the most impactful courses I've taken. It gave me hands-on experience with technologies that are highly relevant in today's job market and helped me practice the mindset of building for scale. I now feel more confident in my ability to design, deploy, and plan for the future of web applications in the cloud. This is the kind of experience that will continue to shape my professional growth moving forward.