

Douglas Rowland

CS-499 Capstone

Self-Assessment

Over the course of this program, I've grown in ways I didn't fully expect when I first started out. What began as a curiosity about how things work under the hood has turned into a solid understanding of software development, algorithms, database management, and even system security. Building out my ePortfolio was a great way to reflect on everything I've done and how far I've come. It gave me the chance to go back through past projects, improve on them, and see how much I've learned. In doing so, I've started shaping where I want to take my career and what kind of developer I want to be.

Working on the ePortfolio helped me zero in on my strengths and where I've made the biggest leaps. One area where I really noticed this was in the Appointment Management System project. When I originally submitted it, it worked fine, but there were clear areas for improvement in terms of test coverage and overall structure. By going back in and writing more complete unit tests, I was able to push the coverage over 98 percent and make sure every core feature was reliable. This process taught me the value of detailed testing and clean code. It also made me realize how important it is to anticipate edge cases and design software that's cleaner and more reliable, not just functional.

The Inventory App, which covers my other two enhancements algorithms and database work, was another major step forward for me. I integrated sorting features, search functionality, and created a Test Data Generator that allows admin users to populate the database with 100+ items instantly. This was super helpful for debugging, but it also gave me a better grasp of how

scalable systems should behave when working with large datasets. I also added an admin-only log viewer to track database activity like additions, deletions, and updates. Building in this functionality really got me thinking more critically about user roles, access control, and the need to keep sensitive data separated from general users.

Security is something that I've grown more mindful of, especially as I started layering in admin authentication to protect certain app features. While I didn't implement full encryption or anything complex, just working through these features made me more aware of how important it is to verify users and restrict access to key operations. It's also made me want to explore this area more in-depth in the future, especially if I plan to work on apps that deal with real user data or payment systems.

While I didn't participate in much group work throughout the program, I still learned a lot about the importance of clear communication and thinking through how others might interact with my code. Since most of my assignments were independent, I had to rely on being organized and writing code that was clean, well-documented, and easy to follow later (especially for myself when I look at old projects). I've come to appreciate how good documentation and clear structure make a huge difference when revisiting or building on a project down the line. It also made me think more critically about how I would explain my work to a teammate or stakeholder in a real-world setting. Even without formal peer feedback, reviewing and improving my own code helped sharpen my ability to think from multiple perspectives and plan ahead in a more thoughtful way.

When it comes to talking to non-developers, I've learned to simplify technical terms without losing meaning. In a few classes, I had to present solutions and explain my logic in a way that made sense to instructors and peers with different levels of technical experience. These

situations helped me realize that technical communication isn't just about being precise, it's about understanding function. I used simple comparisons and analogies a lot to convey how things work in a concise and easy-to-understand way. That skill is going to be vital when I need to discuss progress with stakeholders or explain a system's behavior to someone in another department.

In terms of course content, I'd say the areas I connected most with were software engineering, database systems, and mobile app development. I enjoyed designing user interfaces that are simple and functional, and thinking through how data flows between components. I also got a lot out of learning how to make my code modular and reusable. One specific improvement I'm proud of is how I switched the Inventory App's data access from a Cursor-based approach to using a List of objects. That change made the app smoother and more efficient, and it also made the logic easier to follow and test. It might seem like a small tweak, but it really cleaned up the codebase and made future enhancements easier to implement.

Overall, these three projects in my portfolio work together to show what I can do across different parts of the development process. The Appointment Management System shows my attention to detail and ability to write maintainable code with high test coverage. The Inventory App shows my progress with database management, user interface enhancements, and algorithmic thinking for features like sorting, searching, and bulk data generation. And throughout all of them, I've worked to think more critically about security and access control.

Putting everything together in the ePortfolio helped me see my own progress in a more concrete way. It also gave me a collection of real, working projects that I can confidently share with potential employers. These aren't just school assignments, they're real projects I've put real effort into improving and polishing so they reflect what I'm capable of today. This experience

has made me excited for what comes next, and I feel like I now have the foundation I need to step into the field professionally. Whether it's front-end, back-end, or full stack work, I know I can take on new challenges, adapt to what's needed, and keep improving. This self-assessment might mark the end of the program, but it feels more like the start of something. I'm leaving with a toolkit full of skills I can build on, and I'm excited to start putting it to use.