Chetachi Ezikeuzor  
Jun 22, 2025

Professional Self-Assessment

As I look back on my journey through the Computer Science program at Southern New Hampshire University, I see more than just a collection of assignments and code—I see growth. I see a clearer sense of direction, a stronger technical foundation, and the shaping of values that now define how I approach problems as a developer. Completing this program and building this portfolio have been a huge part of that. It gave me the chance to revisit older work with new eyes and level up those artifacts in ways that reflect the skills I’ve gained along the way.

This program helped me build a solid understanding of the core areas of computer science, but it also made me more intentional about how I approach my career. I’ve always been curious, and this major gave me the space to explore different topics—everything from software engineering to cybersecurity to mobile development. More importantly, it helped me figure out where I thrive. I realized I’m drawn to work that combines clean architecture, thoughtful design decisions, and clear user experiences. That discovery didn’t happen all at once. It evolved as I tackled projects, received feedback, and pushed myself through the tougher parts of the curriculum.

One of the most valuable aspects of the program was learning how to collaborate effectively. I’ve worked in team settings before, but courses like CS 250 (Software Development Lifecycle) and CS 405 (Secure Coding) taught me how to contribute meaningfully in technical environments. I learned how to ask better questions, give actionable feedback, and adapt my communication depending on who I was talking to—whether it was a peer, stakeholder, or project lead. I’ve carried that forward into my professional experiences at T-Mobile and Verizon, where I supported teams with automation and application monitoring tasks. Sometimes, that meant shifting between technical operations and broader communication with non-technical teammates, which challenged me to be more flexible in how I explain and justify my work.

When it comes to communication, I’ve grown a lot. Early in the program, I focused mostly on whether my code worked. Now, I put just as much effort into clarity and documentation. Through assignments like the code review video and narrative writing, I’ve learned how to explain the *why* behind my work, not just the *what*. That skill has proven incredibly valuable, especially when justifying design trade-offs, walking others through architecture, or presenting enhancements that improve scalability or security.

In terms of technical growth, this capstone is where everything started to click. For the software engineering enhancement, I applied clean architecture principles to reorganize the backend of my full-stack app, breaking up monolithic logic into modular routes and middleware. That was a direct result of what I learned in CS 320 (Software Testing and QA) and CS 340 (Software Engineering). For the algorithms enhancement, I refined backend utilities by writing more efficient data sorting and filtering logic. I leaned on skills from CS 260 (Data Structures and Algorithms) and practiced making trade-offs between readability and performance. And for the database enhancement, I implemented JWT authentication and optimized MongoDB queries—an area I explored further in CS 405 and CS 350 (Computer Architecture and Operating Systems), where I started thinking more critically about data flow and security from the ground up.

Security is something I’ve developed a much stronger mindset around. Earlier in the program, I saw it as something separate, something you “add on” after building the main functionality. But through courses like CS 405 and the security-focused work I did during this capstone, I realized that security starts at the architecture level. Thinking about data validation, access control, and user authentication as part of the initial design process has made my applications more resilient and user aware. Even when using third-party libraries or frameworks, I now approach them more critically, asking: How does this handle user data? What vulnerabilities does this open?

This capstone project ties all those lessons together. I chose to enhance a full-stack travel planning app that I originally built in a previous course. Each enhancement—software design, algorithms, and databases—builds on the last. Together, they demonstrate not just a wider range of technical skills, but also a more mature perspective on how to maintain and scale an application over time. My goal with this portfolio wasn’t just to check boxes. I wanted to produce work I’d be proud to show to a hiring manager, recruiter, or engineering peer.

Ultimately, this program helped me bridge the gap between learning and doing. It taught me to think like a developer, not just a student. And while there’s still so much I want to learn and explore (especially in full-stack development and cloud infrastructure), I feel more confident stepping into roles that require not just technical ability, but also thoughtful communication, collaboration, and decision-making. My ePortfolio is the best reflection of that growth.