BSIS Program Review

Completing the Bachelor of Science in Information Science with a concentration in Web Design Technologies has pulled together a lot of different threads in my academic and professional life. I came into the program already working as a full stack software engineer, but the BSIS coursework pushed me to think more carefully about why we design systems the way we do, how information moves through them, and what responsibilities come with building technology for other people. This review looks back at the courses and assignments that shaped me the most and how they connect to my career goals in full stack development, UX-focused design, and eventually technical leadership.

Across the upper-level BSIS curriculum, several courses stand out as especially important for my future career. Early on, LIS 3261 Introduction to Information Science and LIS 3353 IT Concepts for Information Professionals gave me the language and frameworks to think about information systems beyond just “code that works.” Those classes introduced the idea of the information life cycle, the social impact of technology, and the need to design systems that are usable, accessible, and ethical. Later, LIS 4414 Information Policy and Ethics deepened that foundation by focusing on privacy, intellectual property, and regulation. That course directly influenced how I now think about data collection in the apps I build and why the question “can we do this?” is very different from “should we?”

The design-oriented courses were a perfect match for my Web Design Technologies concentration and for my day-to-day work as a developer. LIS 3361 WWW Page Design/Management helped me refine my front-end skills and pushed me to think about

consistency, responsive layouts, and long-term site maintenance instead of just getting a page to look good one time. LIS 3352 Interaction Design took that a step further by grounding design decisions in user-centered methods such as cognition, personas, prototyping, and usability testing, rather than relying on personal preference. LIS 3783 Information Architecture tied everything together by focusing on structure: navigation, labeling, and content organization so users can actually find what they need. Together, these courses shifted my mindset from simply building screens to designing full experiences that make sense to real people.

On the more technical side, courses like LIS 4482 Networks and Communication, CIS 3360 Principles of Information Security, and CIS 4365 Computer Security Policies gave me a stronger sense of how the systems I build sit on top of real infrastructure and real risks. Networks and Communication helped me understand how data actually travels and where performance bottlenecks or single points of failure can appear. The security courses were especially relevant for my current work with cloud services and APIs. They made concepts such as least-privilege access, encryption, and policy enforcement feel less abstract and more like everyday design constraints that I need to respect. Finally, LIS 4930 Intro to Python added another programming language to my toolkit while reinforcing basics such as modularity, file I/O, and data handling in a way that was tied directly to practical projects.

Several assignments from these courses had a bigger impact on my learning than I expected. In WWW Page Design/Management, I built and iterated on a full website and then wrote reflections such as the Final Summary Reflection and the Reflection for Project 2. Those assignments made me step back from the code and evaluate my own

design choices, including structure, navigation, typography, and how well the site communicated its purpose. They also mirrored the kind of client-facing documentation I have to write at work, where explaining why you made a design decision is just as important as the implementation itself. That blend of hands-on building and honest reflection helped me see each project as part of a longer professional story instead of just a one-off homework assignment.

In Interaction Design, the usability evaluation of the Xbox Series X controller was one of the clearest bridges between theory and practice. Using Rogers, Sharp, and Preece's six usability goals (effectiveness, efficiency, safety, utility, learnability, and memorability), I evaluated how the controller actually performs for real players. Writing that paper taught me how to use formal usability concepts in plain language, which translates directly into design reviews and conversations with stakeholders who are not designers. It also reminded me that good usability is not an abstract ideal. It shows up in very specific details like button layout, menu depth, and how forgiving a system is when users make mistakes.

From Intro to Python, the Project Summary: Fitness Tracker stands out as a simple but powerful example of how code, data, and user needs intersect. I designed a command-line app that lets users log daily steps, store them in a CSV file, and compare progress to a personal goal, all using modular Python scripts and basic time-series style tracking. Even though the project was small, it pulled together important concepts such as data modeling, file storage, input validation, and feedback loops that encourage behavior change. It also echoed real-world products like Fitbit and Apple Health, which made the assignment feel like a prototype for something that could actually matter to users. That

experience reinforced my interest in building tools that help people understand and act on their own data.

Looking back across the program, the common thread is that the BSIS curriculum taught me to see technology as more than just stacks of software and frameworks. The courses gave me conceptual tools for thinking about ethics, usability, security, and policy, while the assignments gave me concrete practice applying those tools to real designs and real problems. As a full stack developer, I now approach projects with a more holistic mindset and find myself asking questions such as: Who are the users? What policies and constraints apply? How will this system scale, stay secure, and remain understandable over time?

Overall, the BSIS program has prepared me not just to write code, but to think like an information professional. It has strengthened my technical skills, sharpened my design instincts, and deepened my awareness of the social and ethical dimensions of the systems I help build. As I move forward into roles such as full stack engineer or solutions architect, and possibly into project management or security-focused work, I will be leaning on the mix of theory, practice, and reflection that this program has given me.