**Professional Self-Assessment**

Throughout the Computer Science program, I have developed a versatile and industry-aligned skill set that has shaped both my technical strengths and my long-term professional goals. The process of completing coursework and compiling my ePortfolio has allowed me to reflect critically on how far I’ve progressed—from writing basic programs to building secure, full-stack applications using modern tools and methodologies. I now feel confident entering the field as a well-rounded developer who can collaborate, communicate, and deliver solutions that meet real-world needs.

A major strength that emerged during the program was my ability to collaborate effectively in a distributed environment. Weekly discussion boards functioned as informal team spaces, where I frequently exchanged feedback with peers and proposed solutions to shared technical challenges. This experience closely mirrors how asynchronous collaboration often occurs in remote or hybrid software teams, particularly when relying on documentation, issue tracking, or code reviews to maintain momentum. One particularly illustrative example came in Module 330: Computer Graphics, where students collectively explored the best way to implement OpenGL-based rendering pipelines and shader logic. The resulting conversations reflected real-world dynamics of debating design trade-offs and working across unfamiliar codebases. Over time, I also developed a stronger appreciation for the role of clear and consistent code commenting—not just as a self-reference tool, but as a communication channel for collaborators. I now make a point to leave purposeful comments throughout my code to explain logic, clarify edge cases, and support future maintainers. This practice has significantly improved both my own development process and my ability to contribute effectively to shared projects.

Communicating with stakeholders was another skill I cultivated, particularly through coursework that required translating technical recommendations into accessible language. In *Module 305: Software Security*, for instance, I developed a multi-part security plan tailored to a hypothetical client company. This involved justifying cryptographic choices, explaining authentication workflows, and mapping secure design principles to business outcomes—all while assuming the client had minimal technical expertise. Similar stakeholder-oriented deliverables throughout the program helped me develop the soft skills essential to bridging technical implementation and strategic decision-making.

In terms of software engineering and database design, I progressed from basic CRUD implementations to building modular, scalable applications backed by dynamic, schema-driven NoSQL databases. My coursework and projects emphasized maintainable design, version control workflows, environment separation, and testing strategies. In one example project, I developed a mock social fitness application featuring user profiles, progress logs, and goal tracking—all powered by a Node.js backend, Express API endpoints, and MongoDB collections for persistent data storage. The app also used middleware for login token verification and applied test-driven development practices for backend routes.

I also significantly advanced my understanding of data structures and algorithms. While foundational topics were taught early in the program, I continued to apply them in practical ways. For example, in a previous module, I developed a calendar-based task reminder tool that allowed users to set and retrieve tasks by date and priority. To manage this, I used a priority queue to ensure high-importance tasks were surfaced first, and a map structure to organize tasks by date. This allowed the application to efficiently retrieve upcoming tasks in O(log n) time while maintaining a clean separation of concerns. Choosing these structures helped me meet the performance needs of the app while ensuring scalability—important lessons in balancing algorithmic design with user expectations.

Finally, security was a recurring theme that became central to my thinking as a developer. Across several modules, I learned to approach software not just as a functional product, but as a potential vector for attack. I applied this mindset in projects where I implemented password hashing, token-based authentication, and secure storage practices. I also became familiar with automated vulnerability testing using tools such as Dependency Check, and learned how to advise nontechnical users on security trade-offs and best practices.

Together, these experiences have prepared me to contribute meaningfully in real-world environments. They’ve instilled not only technical confidence, but also the ability to adapt to new tools, balance trade-offs, and think like an engineer responsible for both systems and people.

**The project: Tilted**

The artifacts included in this portfolio have been carefully selected and enhanced to showcase the breadth and depth of my capabilities as a computer science graduate. Together, they demonstrate my proficiency in backend development, frontend design, data handling, algorithmic thinking, and secure software engineering. Each artifact highlights a specific skill area, while collectively, they form a cohesive narrative about my growth as a full-stack developer.

The centrepiece of this portfolio is Tilted, a full-stack web application built using the MERN stack (MongoDB, Express.js, React, and Node.js). This project began as a more conventional travel booking app built with Angular during an earlier course. For the capstone, I reimagined and refactored it into a news analysis platform with greater social relevance and real-world utility. The shift not only reflects my personal interests in politics and media literacy but also demonstrates my ability to assess user needs and align software design with broader societal trends. By selecting a problem space with genuine public value, I aimed to create a product with long-term potential and stakeholder appeal—an approach that underscores my mindset as a developer who considers both end-user value and long-term strategy.

Three supporting enhancement narratives—focusing on software design and engineering, databases, and algorithmic thinking—further illustrate the breadth of my technical abilities and my capacity to think holistically about application architecture.

The first of these narrates how I modernized the frontend and adopted React with Vite and Tailwind CSS, replacing the original Angular foundation. This transition showcases my ability to implement current industry standards, evaluate tooling trade-offs, and architect a flexible, scalable UI layer. The updated frontend demonstrates proficiency in building reusable components, applying responsive design principles, and delivering intuitive, user-focused experiences.

Second, to complement the frontend, the MongoDB backend and data ingestion system highlight my skills in automation, scripting, and database design. Using Node.js and Puppeteer, I created scripts that scraped publication URLs, cleaned and transformed data, and securely inserted records into MongoDB. I implemented Mongoose schemas, validation logic, and environment variable handling, which helped me create a clean, scalable, and secure data pipeline. This artifact demonstrates my ability to integrate multiple tools and libraries to solve real-world data problems.

Thirdly, In terms of algorithmic thinking, I enhanced the application with a feature that calculates the average political bias of a user’s saved articles. This involved collecting numerical bias ratings, filtering and validating data, applying aggregation logic, and dynamically updating the frontend with calculated bias categories and visual feedback using a rotating arrow and bias meter. This enhancement showcases my ability to write efficient data-processing logic, design visual indicators based on algorithmic output, and implement user-facing features that derive insight from raw data.

Although there is no dedicated narrative for this aspect of the application, I want to make note that the application is grounded in secure development practices. All user credentials are hashed using bcrypt, and authentication is handled using JWT tokens. Protected routes on both the frontend and backend ensure secure user experiences. The application includes route-level middleware, environment-based configuration, and frontend logic to conditionally display UI components based on authentication state. This demonstrates a strong understanding of full-stack security and the importance of incorporating protective measures throughout the entire application lifecycle.

Overall, these artifacts come together to tell a clear story: I am a capable and thoughtful developer who understands the full stack—from secure data ingestion and processing, to frontend UI design, to the implementation of meaningful logic that brings value to users. These projects not only demonstrate what I’ve learned in the classroom, but how I’ve applied it creatively, securely, and professionally in building software with real-world relevance.