Milestone 4 Narrative

Sean Born

CS 499

The artifact I selected for my Category 3 enhancement is my Static Code assignment that was originally developed during CS-405: Secure Coding. The original version was a static code analysis exercise written in C++ that focused on identifying and correcting unsafe coding practices, such as improper pointer handling, buffer overflows, and missing exception safety. For this enhancement, I changed that initial codebase into a fully functional secure database-integrated application that uses MongoDB for data storage and implements safe user authentication, role-based access control, and audit logging.

I included this artifact because it demonstrates my ability to apply secure coding principles to real-world database applications. The enhancement shows my growth from working on isolated code-level vulnerabilities to building an integrated, secure system that handles user data responsibly. It also shows my understanding of both software engineering and cybersecurity. These are very important competencies in today’s development environments. Specific components that showcase my skills include the implementation of parameterized queries to prevent injection attacks, role-based access control (RBAC) to restrict unauthorized actions, and audit logging that records invalid login attempts and unusual activity in a protected collection. These improvements strengthened the project’s robustness, scalability, and maintainability while aligning it with industry standards for secure software development.

I successfully met the course outcomes I planned for this enhancement. My Category 3 project aligns directly with Outcome 4, as it demonstrates my ability to apply innovative tools and techniques to implement secure software systems. It also aligns with Outcome 5, as it reflects a strong security mindset that anticipates and mitigates vulnerabilities such as injection risks and privilege escalation. I was also able to demonstrate qualities of Outcome 3 through the use of structured data handling and algorithmic design within the database layer. No updates are needed to my outcome-coverage plan since the enhancement achieved all of the objectives initially proposed in Module One.

Enhancing this artifact taught me how to bridge secure coding theory with practical database engineering. I learned how to design and configure a MongoDB environment with least-privilege user access and TTL indexes for automated log expiration, improving both security and performance. One of the main challenges I faced was making sure that the MongoDB C++ driver integrated cleanly into the existing codebase while maintaining platform compatibility and readability. Managing environment variables securely and testing role-based permissions also required careful planning. Through this process, I strengthened my understanding of secure database architecture, modular software design, and defensive programming practices. These skills are directly applicable to my career goal in video game development. Especially in building secure systems for online features, player accounts, and data-driven gameplay.