CS 499 Milestone Four Narrative

The artifact I selected for this enhancement is the AppointmentService component from my CS-320 Software Testing and Quality Assurance course project. The original version was developed in early 2025 and functioned as a backend service to manage appointments using in-memory storage. While it provided basic CRUD operations, the design cleared all data whenever the program restarted, which made it impractical for real-world use cases. For this milestone, I enhanced the artifact by integrating it with MongoDB for persistent data storage, replacing temporary in-memory storage with a robust, production-ready database solution. Since I am using the same artifact to enhance for all three of the project categories, the enhancements made for this milestone partially build on the enhancements made in the last milestone.

I chose this artifact for my ePortfolio because it shows my growth as a software engineer, from creating a simple in-memory service to designing a full-stack application with persistent storage, authentication, and a refined API. For this enhancement, I integrated MongoDB with Spring Boot to persist and efficiently query appointments, including chronological and exclusive range queries backed by indexes. I added JWT based authentication that protects create and delete operations while keeping reads open, implemented registration and login with Spring Security, and handled token storage and validation on the frontend. I expanded the API to cover all, upcoming, previous, and range queries and added CSV and JSON export, then updated the React UI with login and registration flows, client validation, and compact export controls. Together these changes demonstrate practical skills in database integration, secure API design, and end-to-end feature delivery across frontend, backend, and data layers.

Although my original Module One plan aimed to demonstrate only Outcome 3 and Outcome 4, this enhancement ultimately touched on all five course outcomes. For Outcome 1, I provided a collaborative environment where all users can view appointments, while only authenticated users can add or delete records. This distinction supports multiple stakeholders interacting with the same schedule in a controlled manner. For Outcome 2, I documented the authentication and database setup steps in the README and added professional, human-readable comments in the codebase to explain repository methods, validation paths, and error handling. For Outcome 3, I designed a persistent solution that leveraged derived repository methods for queries, supported exclusive date ranges, and enforced data integrity through unique indexes. For Outcome 4, I used industry-standard tools such as Spring Data MongoDB for persistence, Spring Security for authentication, and JWT for session management, which reflect technologies widely used in professional practice. Finally, for Outcome 5, I validated inputs, anticipated possible misuse, handled duplicate key errors predictably, avoided unsafe query construction, and protected write operations behind authentication. These practices demonstrate an ability to anticipate adversarial behaviors and mitigate risks to privacy and integrity.

The process of enhancing this artifact reinforced the importance of persistence and security in full-stack development. I learned how to design MongoDB schemas and indexes that align with the application’s most common access patterns and how to integrate them with Spring Boot using repository methods. One challenge I encountered was ensuring validation consistency across client and server, particularly when handling date ranges and duplicate keys. Another challenge was translating database errors into user-friendly error messages so that the frontend could surface clear feedback. These obstacles taught me to think critically about both the developer experience and the end-user experience. Overall, this enhancement strengthened my ability to design secure, persistent, and maintainable backend systems that integrate seamlessly with a modern frontend.