

Nathaniel Burt

CS 499

5/23/2025

Software Design and Engineering Enhancement Summary

Artifact Overview

The artifact I selected for this enhancement is a full-stack, containerized web application originally built from a monolithic Node.js project developed in CS 320. The application was expanded and redesigned between Modules One and Three of the capstone course to demonstrate key software design and engineering principles. It now consists of three microservices—auth-service, appointment-service, and a React frontend. It communicates securely through REST APIs and orchestrated using Docker Compose.

Why This Artifact Was Chosen

This artifact was selected for inclusion in my ePortfolio because it showcases my ability to build modular, scalable, and secure systems using modern tools and best practices. The redesign of the original code into a microservices architecture demonstrates my proficiency in system decomposition, containerization, and API design. Specific improvements include:

- Refactoring into separate services for authentication, appointment handling, and frontend rendering
- Implementing JWT-based secure login and protected API endpoints
- Containerizing services with Docker and coordinating them using Docker Compose
- Adding interactive Swagger documentation for the appointment API

- Developing a responsive React frontend with secure integration, filtering, and visual feedback

These enhancements highlight not just code-level knowledge, but architectural decision-making and deployment readiness.

Course Outcomes Alignment

The enhancement meets the intended course outcomes I identified in Module One, including:

- Designing and evaluating computing solutions using algorithmic and architectural principles
- Applying innovative tools (Docker, Swagger, React, JWT) in real-world software practices
- Delivering professional-quality visual and written communications (e.g., API docs and UI design)

The enhancements also reflect a strong focus on maintainability, scalability, and real-world deployment, all of which are skills I plan to carry into system administration and backend engineering roles.

Reflection on the Enhancement Process

Enhancing this artifact helped me deepen my understanding of full-stack architecture beyond theory. Converting a monolith into microservices required careful separation of concerns, interface definition, and data flow design. I learned how to balance security (via JWT), usability (through a React frontend), and maintainability (by using Docker and service-level abstraction).

One key challenge was making sure there was a smooth communication between services while maintaining security boundaries. I also had to troubleshoot cross-origin and token-passing issues when integrating the frontend. These obstacles helped me develop stronger debugging and configuration skills, especially within containerized environments.