John Nguyen

September 21, 2025

CS-499

Enhancement One: Software Design/Engineering

I chose to work with my Travlr Getaways full-stack web application. I built this project in CS 465 as my final project, where I had to create a travel booking site with a customer-facing website and an admin panel. The project uses Express.js and Handlebars for the main site, a REST API for the backend, and an Angular app for the admin side. I first built it last term, and it has been one of the larger projects I’ve worked on.

I decided to include this project in my ePortfolio because it shows my skills. It covers both backend and frontend development, and it has features to demonstrate software design principles like modularity and layered architecture. For my improvement, I worked on making the code more secure and reliable. Some of the changes I made include adding a real JWT authentication middleware instead of the placeholder one, adding input validation for API requests, improving the error handling so it’s consistent, and setting up structured logging. These updates make the project feel like something I could show to an employer.

These updates helped me move forward on the outcomes for software engineering and secure coding. In Module One I planned to use this artifact to show that I can design and implement improvements using software engineering practices. I showed that I can recognize weaknesses in code and then fix them in a way that improves security and maintainability. I’m also making progress toward the outcome of having a security mindset by finding and fixing vulnerabilities, like the unprotected routes. My original plan hasn’t changed, but now I feel confident that this project will strongly represent me in the ePortfolio.

Enhancing this artifact taught me a lot. When I first looked at the old code, I realized how important it is to think about security early on. The original version had wide-open routes that anyone could access. By adding authentication, I saw how safer and more professional the app became. I also got more comfortable with error handling and logging, and I realized how easy it is to troubleshoot when the app gives useful information instead of vague errors.

The hardest part was integrating input validation into the existing code without breaking things. I had to carefully test each route to make sure that valid requests still worked while bad requests were blocked with messages. It was also tricky making sure the Angular frontend still worked with the new backend rules, but I worked through it by testing both layers together. I learned a lot about balancing new features with not breaking old ones.

I think the project is in better shape now, and it feels like something I would be proud to show in an interview. It shows not just that I can code, but that I can think about improvements and apply best practices to make my work professional.