Fayaz Shaikh

10/03/2025

CS 499

**Enhancement Three Narrative**

1. Briefly describe the artifact. What is it? When was it created?
   1. The initial artifact in this final enhancement was called “Travlr SPA”, a single page application created throughout the CS465 course, Full Stack Development I. The artifact I uploaded included the final submission at the end of the class with all the modules completed. It was created along with a flowchart detailing a full implementation of a travel site with a customer and admin portal. The customer could book trips from the databases, while the admins could add trips to the database. Customers could manage their existing bookings and earn points. None of these features, however, were fully completed in the class.
2. Justify the inclusion of the artifact in your ePortfolio. Why did you select this item? What specific components of the artifact showcase your skills and abilities in software development? How was the artifact improved?
   1. Because none of the features were fully implemented in the class, I thought this would be an excellent example of something I could improve by creating what the flowchart showed, with functionality behaving similarly. I selected this item so that I can both show my ability to implement these features but also implement them in an effective manner. I intend to rewrite this from a single page application to a multipage domain with routing, with a full server and client along with a SQLite database holding all the information. I will use a table for accounts and a table for trip data to store everything. I think this improves it by fleshing it out, and while ambitious, will showcase my skills in full stack development.
3. Did you meet the course outcomes you planned to meet with this enhancement in Module One? Do you have any updates to your outcome-coverage plans?
   1. The course outcomes I planned to meet with this enhancement are related to security. With my inclusion of JWT tokens to authenticate all traffic with REST APIs, I was able to successfully meet this course outcome in the enhancement for databases. Overall, I was now able to touch on all of the classes course outcomes, incorporating those elements into my three enhancements I made.
4. Reflect on the process of enhancing and modifying the artifact. What did you learn as you were creating it and improving it? What challenges did you face?
   1. The process of enhancing and modifying the artifact started with a new Next.JS project. Like in previous weeks, I used Tailwind CSS to speed up my UI creation, and created a new page.js landing page. Things began to differ however, as next I worked with app routing to create subdomains for features such as register, login, and profile. As this is a massive project to undertake, I referenced some work from my previous personal projects, specifically code I wrote during a hackathon and then later repurposed into a full stack app I created last month for online Bingo. Then, I wrote a new home page that fetches and neatly displays all the trips. A new tripsService talks to the backend and handles all of the trip creation, deletion, and editing, and booking. All of the user account (signing in, authentication, registration) was pulled from a previous project I worked on. If you are logged in as an administrator, then you have a button to delete, edit, or add new trips. Customers (and admins for now, to make it easier) can book a trip, which currently adds the trips to their home page. The registration process allows users to choose if they are an admin. In a full implementation, this would not be allowed, and would instead be done on the server side, but it is made simpler for now for testing purposes to showcase the full ability of the full stack application. This was a giant rewrite but in the end I’m happy with how it turned out.

Similar to prior weeks, I have delivered the enhancement in a folder containing Next.JS files. If you would like to run them locally, once again the proper node modules are needed (I delete them before uploading as they are hundreds of small files that take a while to compress but also take up almost half a GB). Keep in mind all the code is not in page.js anymore, but instead scattered across many page.js files for every subdomain. There are also common controllers for handling API routes. After changing directory into the folder called “enhancement”, you need to open BOTH the client and the server folders, and type “npm install” to install the dependencies, then “npm run dev” on the client to host it on localhost:3001, and “npm run start” on the server to host it on port 3000. There is also an included example .env file, which contains JWT\_SECRET. In production, this will be filled in by the user, and not be shared. If you delete database.db, the sever will create the database on the fly, but I’ve included an example one that has a user account with username and password “fayaz”, and an example trip. Alternatively, you can go to the client on a web browser and click register to create a new account (and check admin to have full control). This database is local to your instance. You can test all the pages, creating trips, editing them, deleting them, and booking or unbooking them to appear on your home page. All of this has been tested to be fully working! This week, I did not deploy it to my portfolio again, but I will when I get the chance. This future version will also contain existing example trip data in a database you can view.