Byron Laferriere

Professor Goggin

CS-499

**Enhancement One**

1. *Briefly describe the artifact. What is it? When was it created?*

The artifact selected for Enhancement One is a website that was created in my Full Stack Development One class. This class was taken over four months ago now and was meant to teach me about the difference between hardcoded static websites and the high functionality of dynamic websites. In this project I created a M.E.A.N. stack application that uses MongoDB for the database, Express and Node.js for the backend and Angular for the frontend development. This artifact required me to create a database to store company files, usernames and details, and available trip information, then link this database to a backend application that was comprised of Node.js and Express HTML. I used concepts like Bootstrap and Handlebars to make my codebase more efficient and less redundant. Then I created an Angular application that needed to communicate to the backend and the database to pull the information displayed on the page. I used Single Page Application (SPA) architecture to ensure that my page did not experience long load times and push customers away. This project also used Postman to check our API endpoints, which included the creation of JSON Web Tokens for Java. Here, we created an administrative account which had CRUD capabilities for the website.

1. *Justify the inclusion of the artifact(s) in your ePortfolio. Why did you select this item? What specific components of the artifact showcases your skills and abilities in software development?*

I believe that this artifact belongs in my portfolio as a software engineer because of its envelopment of many core practices in both design and development. The enhancements made to this artifact showcase my skills in creating clean and efficient code, ensuring the code base is well-documented using in-line commenting, and the ability to add new features and security updates to the original application. To create clean and efficient code, I went back through the entire project and removed all the commented-out lines of code that we had replaced along the way. To ensure that the project was well-documented, I worked through all of the files within the project and added code in TypeScript and HTML. This showcased my ability to quickly switch between languages as needed. Adding new elements to the home page took quite a lot of back and forth for cross-compatibility between static and dynamic websites. To create a user-friendly home page that had more than a login prompt, required me to showcase my ability to understand the functionality of the site and exactly what files needed to be edited. Finally, the last element that highlights this artifacts placement in this portfolio is the added security layer. Using JSON Web Tokens can still leave a website vulnerable to attacks. To ensure this does not happen, the website was secured by default with the HS256 hashing algorithm and a token revocation process was added. This process would check the current date/time when users logged in and give them a certain amount of time, before the token expired and required them to log back in.

1. *Reflect on the process of enhancing and/or modifying the artifact. What did you learn as you were creating it and improving it? What challenges did you face? How did you incorporate feedback as you made changes to the artifact? How was the artifact improved?*

Throughout this process I learned about the importance of researching new techniques that can be used and learning more about topics I have previously encountered. While performing the enhancements, I learned about the usage of polymorphism in Full Stack Development to a much further degree. Learning what pages needed to be edited to send data to a different page, seemed like a new concept, all though we had previously covered it. This entire process was a great refresher on the methods behind Full Stack Development. I also learned about the standard hashing security that JSON Web Tokens uses. One of the original enhancements I had suggested mentioned protecting against the ‘None Hashing Algorithm’. However, during this class, we added 256-bit encryption to our web tokens to protect against this. I was missing this security feature initially, because of the location of the file. I was not looking in the app\_api folder, which contained the backend security. Some of the challenges that I was able to overcome during this project related to both understanding the layout/communication of the project and how to edit the existing documents without damaging the applications ability to successfully load. I struggled with this greatly when implementing the Token Revocation method into the application. I had to perform extensive research to learn more about JSON Web Tokens and the default security settings that are in place. Feedback was given often, which was quite useful in the growth process. This information helped shape my outlook on the project and what the next steps to take could be. At the end of the enhancement on this artifact, the application now contained an appeasing home page for new clients, with all the proposed implementations, well commented/documented code bases, and a new additional layer of security to help prevent from tokens being stolen and reused.