**Enhancement Three Narrative**

Tyler Blanchard

2/7/2025

For the third enhancement, I have continued to improve upon the functionality and security of my project made for enhancement one, the recreated Graziosos Salvare Animal Shelter Dashboard, by implementing back and front end components and methods to manipulate and control a database based on the core CRUD principles without ever needing to directly access the database. The original artifact for this project was a program written in Python in early 2024 that managed a pre-built database to display data about sheltered animals around the Austin Texas area, allowing users to select and view the location of animals as well as search for dogs that are suitable for multiple types of rescue training. This artifact was initially enhanced for the first enhancement, where it was rewritten into a full stack application using JavaScript, to display a list of sheltered animals. However, this initial enhancement did not include any functions to filter through the database nor did it have any components on the dashboard that allowed for the database to be manipulated besides from simply displaying data, requiring any changes that wanted to be made to the data to be made either directly through the database using command lines or tools such as MongoDB Compass or through direct API calls using tools like Postman.

I chose to further enhance this project in enhancement three as while the initial enhancement recreated the basic functions of the original artifact to neatly display a table of sheltered animals, there was no further functionality, essentially making it a static web page that only displayed the data it was given and prevented the data from being easily changed. This project had much more potential than just a simple tool to display sheltered animal, and I wanted to fulfill this potential by introducing add, edit, and delete features in addition to security features to allow this application to be used by theoretical administrators of the Graziosos Salvare team to securely manage the data within this application on the application itself, preventing the need to directly access the back-end to do so. To improve upon this project and to further fulfill its potential, I implemented the last three features of the CRUD functionality (create, update, and delete) by creating new components to dynamically add. remove, and update animal entries all in one page. To make sure these features are secure to prevent guests/non admins from modifying the database, I implemented a login feature along with authentication security components in the front and back ends to authenticate admins and ensure that only admin users with valid credentials and a valid JSON Web Token can view and access these CRUD features. Additionally, I have implemented a custom filtering feature to allow users to find animals that are not only candidates for rescue training but are available for adoption, allowing for the application to be used in a more universal manner to view animals that are available to be adopted in various shelters in Austin. I also have implemented plenty of minor features such as a selection display to make the application look more responsive and better seeding to allow for a more realistic display of the many different potential animals in Austin shelters. With these improvements, I demonstrate my skills in creating core database features to manage/control a database based on my implementations of the remaining CRUD features, and in understanding and implementing basic security with the login and authentication functions to prevent unknown users from being able to change the database. I also demonstrate my skills in being able to implement interactions between a server and a client based on my implementations of the CRUD and security features as well, as both sides need to interact with the other end to allow requests to be verified and processed.

The two course outcomes that I had planned to meet for this enhancement were outcomes 4 and 5:

Demonstrate an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals (software engineering/design/database)

Develop a security mindset that anticipates adversarial exploits in software architecture and designs to expose potential vulnerabilities, mitigate design flaws, and ensure privacy and enhanced security of data and resources

Based on the improvements I have made for this third enhancement, I believe that I have sufficiently fulfilled these two outcomes. Outcome 4 was fulfilled by utilizing my skills and knowledge on the CRUD principles and the given software environment to develop the front end components to add, update, and remove data from the database, while Outcome 5 was fulfilled by my understanding of basic security principles to implement a functional administrator login feature to allow administrators to be authenticated to use the add, update, and remove features.

This process of improving the project came with a few challenges, and I have learned quite a bit by facing and overcoming these challenges. One thing I learned was the importance of understanding the tools or libraries you use so you can write a solution for it to make it work the way you want it to. During the implementation of the custom filtering for example, I needed to update the data table so it correctly displayed the new filtered data queried from the database, but inserting the data and attempting to display it would only display the original set of data. This was due to the data table library I was using being static, such that no data could be added or changed after implementation. By understanding this, I was able to implement a solution that saved the filtering data into storage and reimplemented the data table whenever a new filter was selected to allow for the new data to be correctly displayed on the dashboard. Another thing I learned was the importance of including security not only in the front end to hide or show features, but also in the back end to control the use of important functions that may change data. While simply hiding features from users can help prevent non authorized users from seeing sensitive functions, they can still be accessed in other ways such as by entering the correct URL extension to route to a component. To ensure that sensitive functions are secure, it is critical to add security to the server side or onto any APIs to check all requests for validity and prevent any requests or processes that were not made by a valid or authenticated user. By better understanding these concepts, I can further improve on my problem solving skills and understand even more advanced concepts of security to make the projects and databases I program now and in the future more functional and secure.