Randy Ortiz

Professor Kraya

CS-499 Computer Science Capstone 2025

July 19, 2025

The artifact that I selected for this milestone is the animal rescue dashboard that I originally developed in course CS-340. It integrates a Dash web interface with MongoDB to manage and visualize animal shelter data. This application is designed with the intent of allowing users to view adoptable dogs using filters like rescue type, breed, and age. The code interacts with the backend animal\_shelter class and the frontend Dash interface.

I selected this artifact to include in my ePortfolio because it is a full-stack application that uses data access, business logic, and a user interface which provides a lot of opportunities for enhancement. It also provides opportunities for improvement in terms of code modularity, input validation, and error handling. The enhancements I made were done by refactoring the create() method in the animal\_shelter.py to validate the required fields, type check and convert for the necessary variables, check for duplicates animals, and stripping the \_id fields to prevent insert errors. I also added test cells to confirm that validation works using missing fields or invalid types. I also added cells to validate functionality using print statements, DB document count before and after insertion, and removing any duplicates.

These enhancements display my ability to improve software design by applying validation, and ensuring a database is consistent. These enhancements align with course outcomes 3 and 4. For course outcome 3, the redesigning of the create() method in the animal\_shelter class to include input validation logic ensures fields are present and are properly typed to avoid bad data entering the database. This reflects improving reliability, safety, and flexibility without making much of a tradeoff. For course outcome 4, applying real world development practices with MongoDB and Python to enhance the data insertion logic with the validation demonstrates my software engineering skills.

Throughout this enhancement, I learned to anticipate edge cases and ensure clean, defensive programming, especially if working with real world data in MongoDB. One challenge I faced was confirming that my input validation was working in practice not only theory. Initially, I struggled triggering and observing the validation logic, so I added specific test cells with print statements to ensure errors were caught and handled. I also had to troubleshoot insertion errors that were related to duplicate \_id fields. This taught me that small details like leaving MongoDB keys in a test input can cause issues. I had to learn to strip the \_id fields before insertion to help me avoid those conflicts. This ended up strengthening my understanding of database constraints.