Milestone 2 Narrative

Sean Born

CS-499

**Description:**

The artifact I selected is the Animal Shelter Dashboard that I originally created for CS-340. This project consisted of two main parts: a Python module called AnimalShelter.py that providing CRUD functionality for a MongoDB database, and a Jupyter Notebook dashboard that used Dash and Plotly to visualize animal shelter data. The original artifact was created in 2023 as part of coursework focused on data-driven application design and database integration.

**Justification:**

I chose to include this artifact in my ePortfolio because it demonstrates my ability to integrate databases, software engineering practices, and user interface design into a working application. It shows off my skills in both backend and frontend development, combining secure database access with a clear, user-friendly visualization layer. The enhancements I added improved the artifact in several areas. The first is the structure. I refactored the code into multiple modules that include data.py, layout.py, callbacks.py, and app.py. The second is the security. I replaced hardcoded database credentials with environment variables, validated inputs before database operations, and added error handling for CRUD functions. The third is the efficiency of the program. I changed the use of numeric indices for database fields to meaningful column names, and structured the dashboard to allow multiple filters and multi-marker map support. Lastly, I enhanced the usability of the program. The dashboard now supports multi-row selections in the table and displays multiple markers on the map at the same time. This artifact highlights my skills in secure coding, modular software design, database integration, and user experience design, making it a strong demonstration piece for my portfolio.

**Outcome Coverage:**

This enhancement aligns with the outcomes I planned in Module One. It aligns with course outcome 3 because I demonstrated modular design by separating the database, layout, callbacks, and main entry point into distinct files. It aligns with outcome 4 because I improved readability and maintainability by replacing magic numbers with descriptive names which makes the code easier to debug and extend. Lastly, it aligns with course outcome 5 because I addressed security flaws by moving credentials into environment variables, validating inputs, and adding error handling.

**Reflection:**

Enhancing this artifact taught me the importance of designing for maintainability and security from the start. The original single-file structure worked for a classroom assignment, but it became inefficient as I tried to add more features. Breaking the project into modules made it easier to test, reuse, and expand. One challenge I faced was balancing security with usability. For example, requiring environment variables for credentials improved security but added a setup step that needed documentation. Another challenge was replacing numeric indices with column names. While it increased readability, it also required me to carefully inspect the database schema to make sure they were consistent with each other. Through this process, I gained confidence in applying real-world practices like secure configuration management, modular design, and defensive programming. These are transferable skills I can apply to future projects, especially those involving database-driven applications.