**Databases Narrative**

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

This artifact is an animal shelter database application originally created in my CS-340 Client/Server Development course. Initially developed in a Jupyter notebook, the application was designed for a fictional company, Grazioso Salvare, to identify suitable rescue dogs by filtering animal shelter data based on breed, age, and sex. The tool used MongoDB for data storage and featured a basic dashboard with pie chart visualizations, filtering options, and an interactive map showing animal locations based on shelter data from the Austin, Texas area.

My refactored version significantly expands the original concept—transitioning from Grazioso Salvare’s narrow focus on specific rescue criteria to a full-fledged animal shelter dashboard designed for public adoption use. Users can now filter by species, breed, age, sex, and other key attributes to help match prospective adopters with pets that best fit their needs and preferences. The goal was to transform a specialized academic tool into a broadly usable and realistic application for shelters and adopters alike.

***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?***

I selected this artifact because it offered a strong foundation for demonstrating a wide range of technical and design skills while also allowing me to transform a narrowly scoped academic project into an industry-relevant solution. In my opinion, the enhancement process reflects significant progress in both software engineering practices and user-focused design.

Key improvements include:

* Architecture & Security
  + Migrated from Jupyter notebook to a Python application
  + Implemented secure credential management with environment variables
  + Added comprehensive error handling and input validation
* User Experience Transformation
  + Redesigned interface from rescue-specific tool to general animal adoption platform
  + Created modern filter system with dropdowns, sliders, and quick-select buttons
  + Improved visual design with professional color scheme and layout
* Data & Performance
  + Integrated CSV data processing for 10,000+ animal records
  + Optimized performance with smart loading limits and toggle features
  + Enhanced data presentation with user-friendly labels and readable formats
* Visualization Improvements
  + Streamlined map interface with adopter-focused animal information
  + Removed unnecessary chart components for cleaner, more purposeful design
  + Added interactive features like map toggling and enhanced popups
* Code Quality
  + Established modular project structure with separated concerns
  + Replaced hardcoded values with configurable constants
  + Improved maintainability through reusable functions and clear documentation

***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?***

In my Module One planning, I aimed to meet the following program outcome:

*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.*

I believe the enhancements made to this project strongly align with that outcome. By refactoring the original rescue-dog filter into a more scalable and intuitive adoption platform, I demonstrated my ability to design, build, and refine a practical solution that addresses real-world needs. The result is a production-ready application that could realistically assist shelters and adopters alike.

***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?***

This project challenged me to grow in both technical capability and design thinking. On the technical side, I expanded my knowledge of modern Python development using the Dash framework, learned best practices for project structure and modular design, and handled real-world data processing at scale. Moreover, implementing environment variable management and removing hardcoded credentials also helped me adopt better security hygiene.

The most challenging aspect was redesigning the user experience and data flow. Since this was a previous school project, much of the original logic for filtering rescue categories was already hardcoded into the system; accordingly, transforming the application from a specialized tool with predetermined rescue categories to a flexible platform required me to develop new skills in software architecture design and learn how to balance maintaining existing functionality while implementing entirely new user requirements.

Perhaps the most valuable takeaway from this enhancement was the importance of iterative development and constant evaluation. Every improvement was guided by a focus on user experience, not just feature expansion. I’ve learned that building a good application isn’t just about writing functional code—it’s about making that code serve people effectively and intuitively.