Ericka Resendez

Southern New Hampshire University

CS-499 Computer Science Capstone

Artifact Narrative: Grazioso Salvare Dashboard

1. Briefly describe the artifact. What is it? When was it created?
    a. This artifact is the Grazioso Salvare Dashboard, which was created for the CS-340 Client-Server Development course during February and March of 2025. It consists of a Python-based animal shelter management system that utilizes CRUD operations and an interactive dashboard that was built with Jupyter Dash. It originally utilized a virtual MongoDB database that was hosted on Apporto services. The system was designed to help Grazioso Salvare, an animal rescue organization, to help identify and manage dogs suitable for search and rescue training through interactive filtering, data visualization, and mapping capabilities. The enhanced version introduces a comprehensive data export and reporting module, allowing users to download filtered results in multiple formats (CSV, JSON, Excel) and automatically generate professional PDF reports with statistics and animal data for rescue team briefings.

2. 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 did the enhancement improve the artifact? What specific skills did you demonstrate in the enhancement?

a. I selected this project for my ePortfolio because it demonstrates my ability to engineer a practical, user-focused software solution across multiple technical domains. I refactored the entire data layer from MongoDB to CSV-based storage while maintaining the original API interface. This required engineering a custom filtering system (_filtered_data method) that replicates MongoDB query operators ($in, $gte, $lte) using pure Python, demonstrating my ability to abstract database functionality and create portable solutions. This enhancement added five new export methods to the AnimalShelter class: export_to_csv(), export_to_json(), export_to_excel(), generate_rescue_report_pdf(), and generate_rescue_report_pdf_from_data(). I integrated the ReportLab library to generate professional PDF reports with formatted tables, rescue-type statistics, and metadata. The reports include calculated metrics (water rescue count, wilderness rescue count, disaster rescue count) using custom helper methods that apply specific breed, sex, and age criteria. It showcases software design and adaptability through the custom AnimalShelter CRUD class, which was engineered to emulate a MongoDB interface while operating on a portable CSV file.

b. The enhancement transformed the dashboard from a read-only visualization tool into a complete data management system that supports operational workflows. Teams can now export filtered datasets for external analysis and generate printable briefing documents with mission-relevant statistics. These enhancements directly fulfill Course Outcome 4 by demonstrating the use of well-founded techniques and tools (Dash, pandas, ReportLab, CSV processing) to implement professional-quality features that deliver immediate operational value.

3. Reflect on the process of enhancing 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? Which course outcomes did you partially or fully meet with your enhancements? Which do you feel were not met?
    a. This enhancement process allowed me to gain a deeper understanding of library integration and data format conversion. I learned how ReportLab structures PDF documents and how to apply TableStyle objects to create professional-looking reports. I also gained experience with pandas' export methods (to_csv(), to_json(), to_excel()) and how to handle in-memory data transformations between Dash's dictionary format and DataFrame objects. A significant challenge was creating a reliable data flow from the Dash frontend's filtered view to the backend PDF generator. This required carefully passing the filtered dataset through callbacks and ensuring the generate_rescue_report_pdf_from_data method could process it correctly. Another hurdle was mastering the ReportLab API to create a well-structured, styled PDF document from dynamic data, which involved managing page layout, tables, and paragraphs programmatically.
    b. I was unable to incorporate feedback as I fell behind very early in the term and did not turn in the assignments for feedback within the permitted timeframe. The artifact evolved from a single-purpose visualization tool to a comprehensive data management platform. The CSV migration made the system portable and eliminated external dependencies (MongoDB, Apporto services), while the export features enable real-world workflows like sharing data with field teams, generating briefing materials, and archiving operational snapshots. I fully met Course Outcome four through this

enhancement. The enhancement demonstrates industry-standard practices including library integration, multi-format data export, professional documentation generation, and user-centered design. The PDF reports follow operational briefing conventions with clear headers, summary statistics, and formatted data tables. I believe I also partially met outcome two as the modular class allows for future extensibility, though I did not conduct formal algorithmic analysis or complexity evaluation. I did not mee course outcome five as the dashboard and CRUD operations as I did not address security concerns like input validation, authentication, or access control. In a production environment, features like user permissions for exports and data sanitization for PDF generation would be critical.