Grazioso Salvare Dashboard – README

Eric L Foster.

SNHU- CS340

Project Overview and Functionality

This application creates a data dashboard for Grazioso Salvare, a search-and-rescue dog training organization. The dashboard is designed to interface with real-time data from animal shelters using MongoDB. It enables users to filter and visualize adoptable dogs based on their suitability for rescue training in three categories: Water Rescue, Mountain or Wilderness Rescue, and Disaster or Individual Tracking. A reset filter is also provided.

Each record displayed includes fields such as animal type, breed, age in weeks, sex upon outcome, and geolocation data. All columns are fully searchable and sortable in the data table, allowing users to refine results even further based on their own preferences or qualifications and the pie chart visualization summarizes breed distribution among filtered results. Breeds representing less than 1.0% are excluded to maintain legibility and relevance.

The dashboard was developed using Python, which served as the base programming language. Dash was used as the primary framework for building the interactive web-based dashboard, supporting both the view and controller components of the application. MongoDB was selected for its flexibility and strong querying capabilities, making it well-suited for serving as the model layer. To manipulate and prepare data retrieved from MongoDB, the Pandas library was used to translate records into the needed DataFrame structures that were compatible with Dash components.

The Visualizations were created using Plotly Express, which allowed for the efficient generation of responsive charts like the breed distribution pie chart. The map component was implemented using Dash Leaflet, which enabled the display of geolocation data through dynamic markers. JupyterDash was used to allow the application to be run and tested inside Jupyter Notebook. Finally, Base64 encoding and OpenCV were used to load and embed images such as the Grazioso Salvare logo and the Paper Rabbit Labs rabbit icon.

**Why MongoDB?**

MongoDB was used as the model component because of its ability to store large volumes of semi-structured animal outcome data. It allows for dynamic queries, complex filtering conditions, and easy integration with Python via the PyMongo library. MongoDB’s document-oriented structure allows developers to perform quick filtering based on field values, including nested attributes.

**Why Dash Framework?**

Dash was selected for its seamless integration with both interactive Python backends and modern HTML/CSS visual layout. It enables rapid creation of responsive interfaces, dynamic callbacks, and scalable deployment via Flask-style components.

Steps Taken to Complete the Project

1. Designed a CRUD module to interface with the MongoDB AAC database.

2. Built a dashboard layout in Dash using the MVC pattern.

3. Added interactive filters (radio buttons) for each rescue type.

4. Configured a searchable, sortable data table to update with each filter selection.

5. Developed map and pie chart components that dynamically reflect filtered results.

6. Styled the dashboard to meet Grazioso Salvare’s branding requirements, including logo and personal identifier.

7. Tested the dashboard by running the IPYNB file and capturing required screenshots of functionality.

Screenshots of Dashboard Functionality

**A screenshot of a computer

AI-generated content may be incorrect.Initial Dashboard View (Unfiltered**):

**A screenshot of a computer

AI-generated content may be incorrect.Water Rescue Filter Applied**:

**A screenshot of a computer

AI-generated content may be incorrect.Mountain or Wilderness Rescue Filter Applied**:

**A screenshot of a computer

AI-generated content may be incorrect.Disaster or Individual Tracking Filter Applied**:

**A screenshot of a computer

AI-generated content may be incorrect.Reset Filter (All Results)**:

**Challenges and Solutions**

Challenge: Filtering breeds while maintaining readable charts.

Solution: Applied a 1% minimum threshold to filter out low-frequency breeds in the pie chart.

Challenge: Layout and component overlap in smaller screen sizes.

Solution: Adjusted CSS and container widths to ensure layout responsiveness and visual fit.

Challenge: Image and unique identifier positioning.

Solution: Used flexbox and inline styling to align and style the GSL logo and Paper Rabbit Labs branding.

Resources Used

Dash documentation: <https://dash.plotly.com>

MongoDB documentation: <https://www.mongodb.com/docs>

Python official documentation: <https://docs.python.org/3>

SNHU Virtual Environment Resources and Code Examples

Austin Animal Center. (2020). *Austin Animal Center Outcomes* [Data set]. City of Austin, Texas Open Data Portal. https://doi.org/10.26000/025.000001