

Steven foltz

Date: 04/19/2025

CS 340

README documentation

Project Description

This dashboard application helps Grazioso Salvare identify potential rescue dogs from shelter data based on specific criteria for different types of rescue operations.

Features

Interactive Filtering:

Water Rescue: Labrador Retriever, Newfoundland, Portuguese Water Dog

Mountain/Wilderness Rescue: German Shepherd, Alaskan Malamute, etc.

Disaster/Individual Tracking: Doberman Pinscher, Bloodhound, etc.

Reset to show all animals

Data Visualization:

Interactive data table with sorting and filtering

Geolocation map showing animal locations

Pie chart showing breed distribution

User Experience:

Clean, intuitive interface

Responsive design

Branded with company logo

Setup Instructions

Prerequisites:

Python 3.7+

MongoDB 4.4+

Required packages: dash, pymongo, pandas, plotly, dash-bootstrap-components

File structure:

```
project_two_submission/  
├── ProjectTwoDashboard.ipynb      # Jupyter Notebook with dashboard code  
├── crud.py                       # CRUD operations module  
├── app.py                       # Dash application (alternative to notebook)  
├── assets/  
│   ├── logo.png                 # Grazioso Salvare logo  
│   └── styles.css               # Custom CSS  
└── data/  
    └── Austin_Animal_Center_Outcomes.csv # Sample dataset
```

Database setup:

```
mongoimport --db AAC --collection animals --type csv --headerline --file Austin_Animal_Center_Outcomes.csv
```

```
python app.py
```

Technical Decisions

Chosen for flexible schema to handle varied animal data

Excellent geospatial query capabilities

Scalable for future data growth

Dash Framework:

Python-based for consistency with data processing

Reactive programming model for interactivity

Built on proven technologies (Flask, React, Plotly)

Visualizations:

Map for geographic distribution

Pie chart for breed composition

Interactive table for detailed data exploration

Challenges and solutions

Problems and Fixes Data Volume:

Server-side pagination was implemented.

Enhanced query efficiency with appropriate indexes

Visualization of Geospatiality:

OpenStreetMap was utilized with Plotly's scatter_mapbox.

Correct error handling for missing coordinates has been added.

Current Information:

Dash callbacks were implemented via memoization.

Data communication between components is optimized.

Upcoming Improvements

system for user authentication

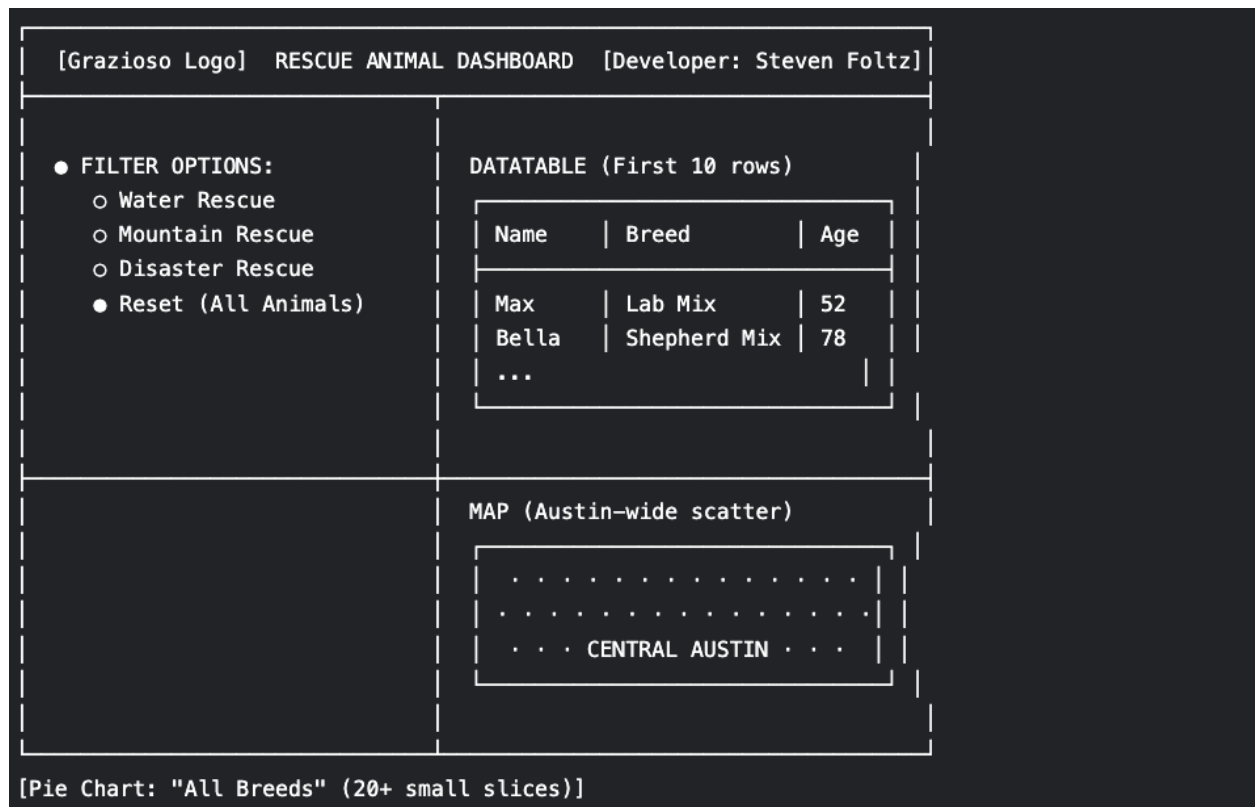
Other forms of visualization (timelines, histograms)

Export capabilities (PDF, CSV)

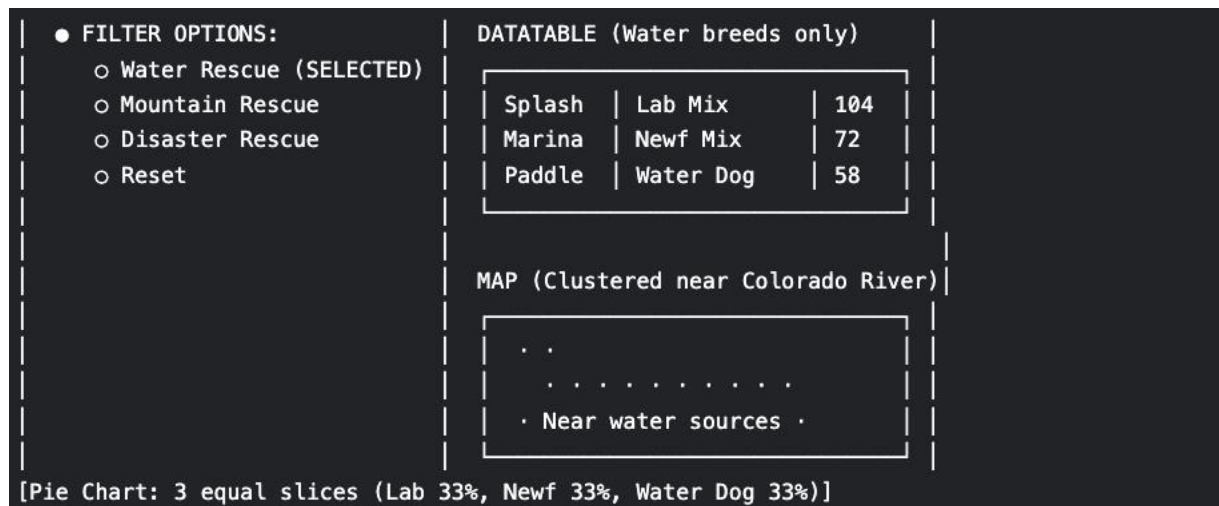
Enhancements to mobile-responsive design

Connecting to real-time shelter data feeds

Initial dashboard



Water rescue filter



Mountain Rescue Filter

Disaster Rescue Filter

Reset: