Eskinder Kassahun

Grazioso Salvare Dashboard Project

Introduction

This document highlights the development of the Grazioso Salvare Dashboard, an interactive tool designed to explore and visualize the Austin Animal Center Outcomes dataset. Built with Python, Dash, and MongoDB, the dashboard offers dynamic filtering, data visualization, and exploration features through various widgets, charts, and tables.

Required Functionality

The Grazioso Salvare Dashboard enables users to:

1. Filter data interactively based on rescue types, such as Water Rescue, Mountain Rescue, and Disaster Tracking.
2. View a data table that updates dynamically based on selected filters.
3. Visualize the data through a pie chart and a map widget for better insights.

Tools and Rationale

The following tools were chosen for their specific strengths:

* Python: For implementing the dashboard's logic and backend functionality.
* Dash: To build an interactive and user-friendly web-based dashboard.
* MongoDB: For efficient storage and querying of the animal outcome dataset.

Development Steps

1. Setup MongoDB and Python CRUD operations for database interaction.  
2. Developed an unfiltered data table displaying all animal outcomes.  
3. Created interactive filtering options using radio buttons.  
4. Built a pie chart and map widget for dynamic visualization.  
5. Tested the dashboard and took screenshots for documentation.

Challenges and Solutions

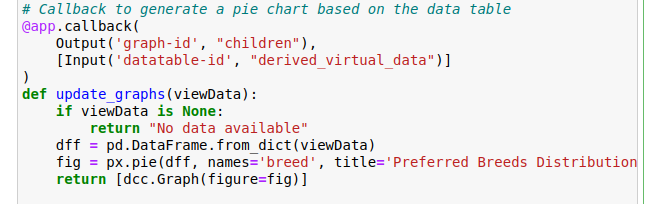
During the development of the Grazioso Salvare Dashboard, the following challenges arose:

1. Map Widget Integration:  
   Adding the map widget was challenging as it failed to display properly, likely due to missing or mismatched geolocation data. Unfortunately, this issue remains unresolved, and the map feature could not be fully implemented.
2. Pie Chart and Table Integration:  
   Setting up the pie chart and data table was initially difficult due to compatibility issues with the MongoDB dataset. Errors occurred when filtering data, mostly because of inconsistencies in the data structure.

Solution:  
The pie chart and table issues were resolved by cleaning the dataset, renaming columns to match Dash requirements, and ensuring data was compatible with filtering and visualization. With debugging and testing, both widgets were successfully integrated into the dashboard.

Screenshots

Screen shot for Pie Chart



Screenshot for Geolocation

A screen shot of a computer code

Description automatically generated

Screenshot for Queries for Filtering

A screenshot of a computer

Description automatically generated

Screenshots of Interactive Filtering Options

A screenshot of a computer

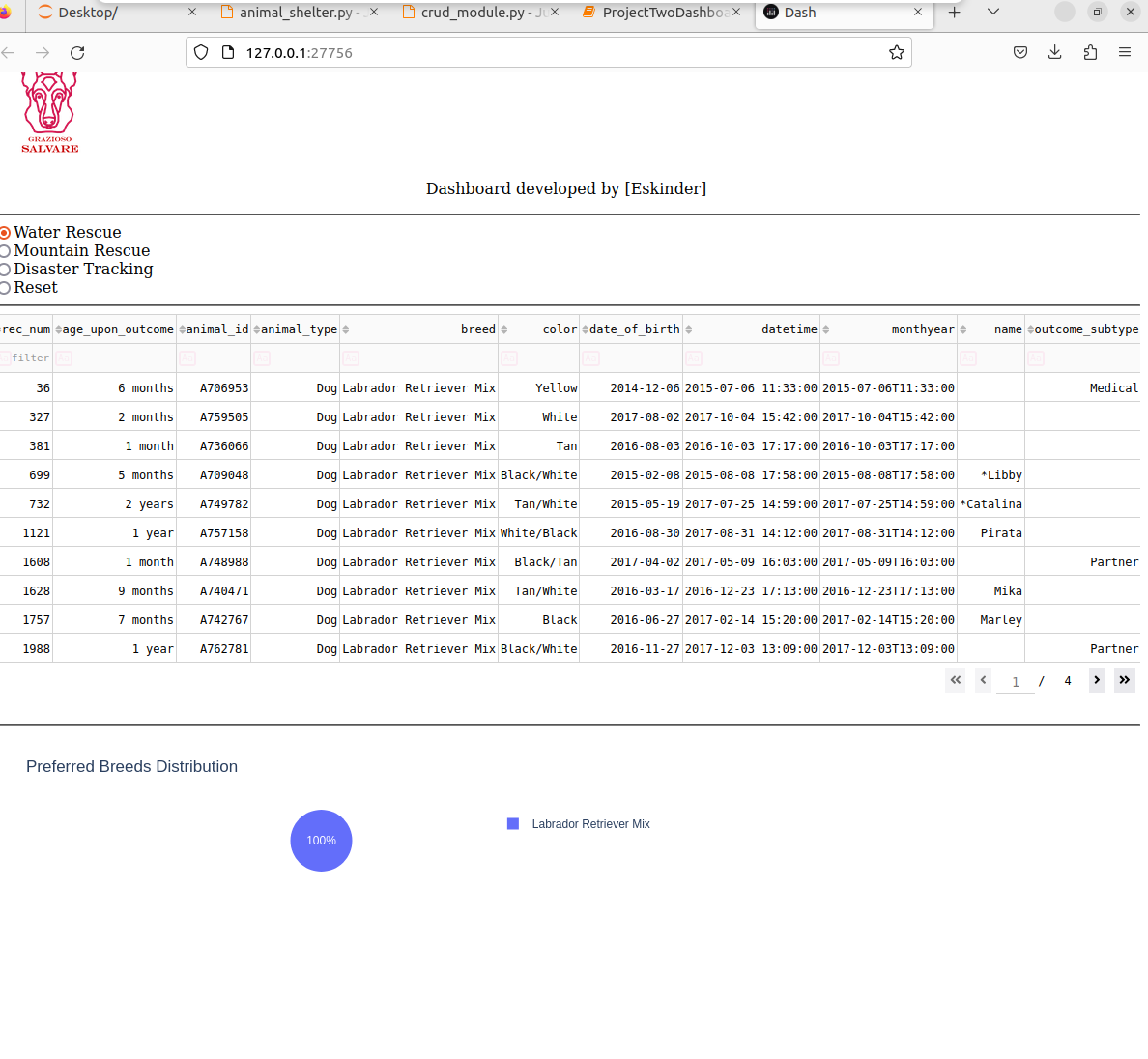
Description automatically generated

Dashboard with Reset filter applied.

A screenshot of a computer

Description automatically generated

Dashboard with Water Rescue filter applied.



Dashboard with Mountain Rescue filter applied.

A screenshot of a computer

Description automatically generated

Dashboard with Disaster Tracking filter applied

A screenshot of a computer

Description automatically generated

Filtered data in the dashboard table.

A screenshot of a computer

Description automatically generated

Animal Shelter py

A screenshot of a computer screen

Description automatically generated