Overview

This project involves the development of an interactive dashboard for Grazioso Salvare, a company that specializes in training animals for rescue operations. The dashboard enables users to filter and visualize animal data based on specific rescue scenarios, such as water rescue, mountain or wilderness rescue, and disaster or individual tracking. The project utilizes Python, Dash, Plotly, and MongoDB for data handling and visualization.

Project Requirements

The dashboard must fulfill the following functionalities:

CRUD operations for filtering animal data from a MongoDB database.

Interactive filtering options for different rescue scenarios.

An interactive data table that responds to user inputs.

Charts that display data dynamically based on user selections.

Clear documentation and reproduction instructions.

Tools and Technologies

Python: Used for backend data processing and CRUD operations.

Dash: A Python framework for building analytical web applications.

Plotly: A graphing library used for creating interactive charts.

MongoDB: A NoSQL database used to store and query animal data.

dash-leaflet: Used for creating interactive maps.

Jupyter Notebook: For developing and running the code interactively.

Setup Instructions

Clone the repository: Ensure you have the project files.

Install dependencies: Install the required Python packages using pip:

sh

Copy code

pip install dash dash-leaflet pandas numpy plotly pymongo base64

Set up MongoDB: Configure MongoDB connection details in the animal\_shelter.py file:

python

Copy code

username = "aacuser"

password = "newpassword"

mongo\_host = "nv-desktop-services.apporto.com"

mongo\_port = 30316

Run the application: Execute the Jupyter notebook (ProjectTwoDashboard.ipynb) to start the Dash server and display the dashboard.

Functionality and Usage

CRUD Python Module Queries for Filtering

The animal\_shelter.py module implements CRUD operations for MongoDB, enabling efficient data retrieval based on specified filters. The dashboard allows users to select from predefined rescue scenarios, triggering the appropriate database queries to filter data.

Interactive Filtering Options

The dashboard features radio buttons for selecting different rescue scenarios:

Water Rescue: Filters for Labrador Retriever Mix, Chesapeake Bay Retriever, and Newfoundland breeds.

Mountain or Wilderness Rescue: Filters for German Shepherd, Alaskan Malamute, Old English Sheepdog, Siberian Husky, and Rottweiler breeds.

Disaster or Individual Tracking: Filters for Doberman Pinscher, German Shepherd, Golden Retriever, Bloodhound, and Rottweiler breeds.

Reset: Resets the filters to display all data.

Interactive Data Table

The data table component (dash\_table.DataTable) dynamically updates based on the selected filters. It supports sorting, filtering, and pagination, enhancing user interactivity.

Charts

The dashboard includes a pie chart that visualizes the breed distribution of the filtered data. The chart updates dynamically based on user inputs.

Map

An interactive map (dash\_leaflet.Map) displays the location of selected animals based on their latitude and longitude. The map updates when a row in the data table is selected, showing the specific location of the chosen animal.

Challenges and Solutions

Data Filtering and Queries

Challenge: Implementing efficient and accurate queries to filter data based on user selections.

Solution: Utilized the MongoDB query language and the CRUD Python module to create specific filters for each rescue scenario.

Interactive Components

Challenge: Ensuring that the data table, charts, and map update dynamically based on user inputs.

Solution: Employed Dash callbacks to link user interactions with data updates, ensuring seamless interactivity.

User Interface Design

Challenge: Designing an intuitive and responsive user interface.

Solution: Utilized Dash layout components and CSS styling to create a clean and user-friendly interface.

Reproduction Instructions

To reproduce this project, follow these steps:

Set up the environment: Install the necessary Python packages.

Configure MongoDB: Ensure the MongoDB server is running and accessible with the provided credentials.

Run the Jupyter notebook: Open ProjectTwoDashboard.ipynb and run all cells to start the Dash server and view the dashboard.

Test the functionality: Use the interactive filters to verify that the data table, charts, and map update correctly based on user inputs.

Conclusion

This project demonstrates the integration of data processing, interactive visualization, and user interface design using Python, Dash, Plotly, and MongoDB. The dashboard provides a powerful tool for analyzing and visualizing animal rescue data, enhancing decision-making for Grazioso Salvare.

Screenshots

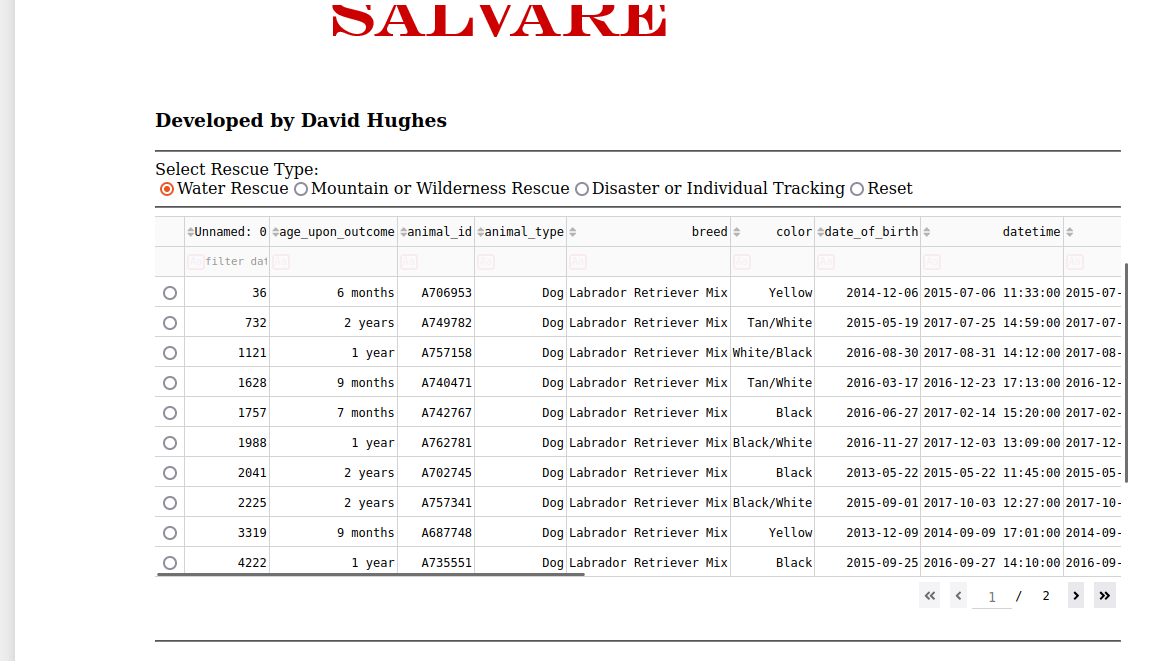
1. Dashboard Header



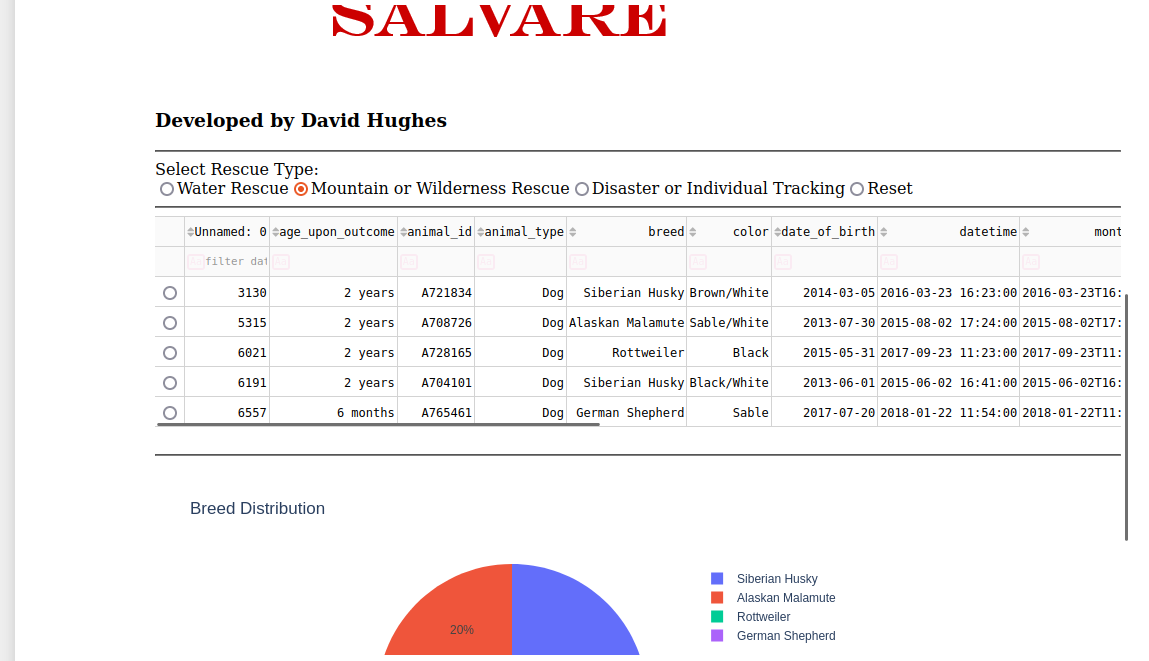
Description: This screenshot shows the header of the dashboard with the Grazioso Salvare logo and the title "CS-340 Dashboard".

2. Data Table with Default View

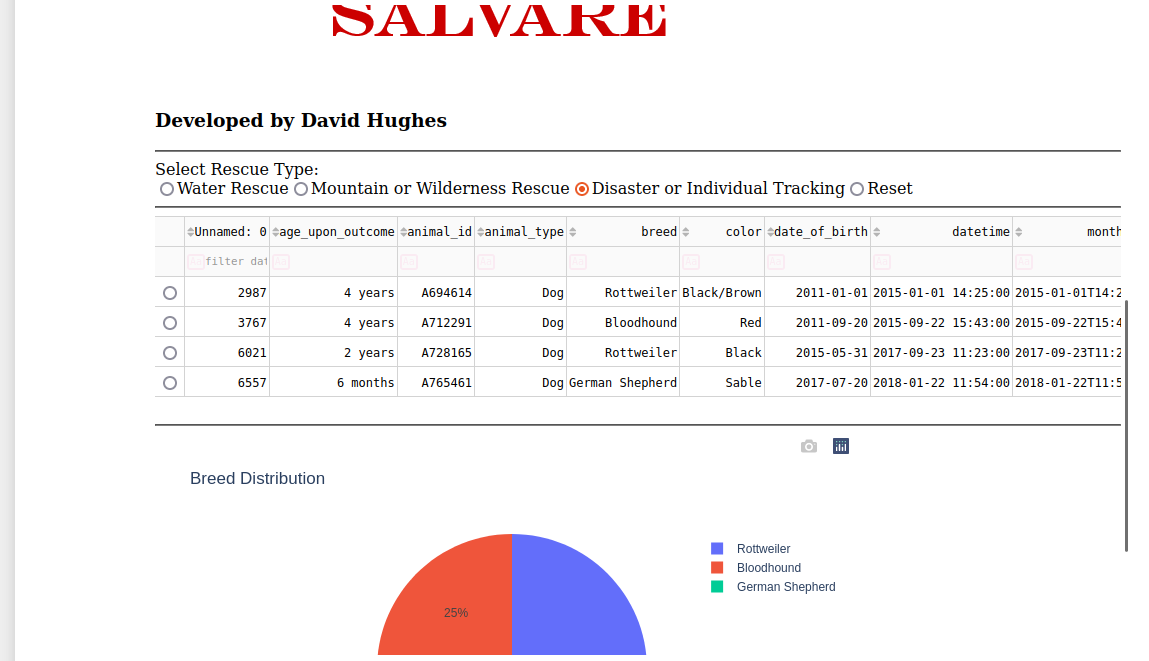
Description: This screenshot displays the data table in its default state with no filters applied.

3. Water Rescue Filter

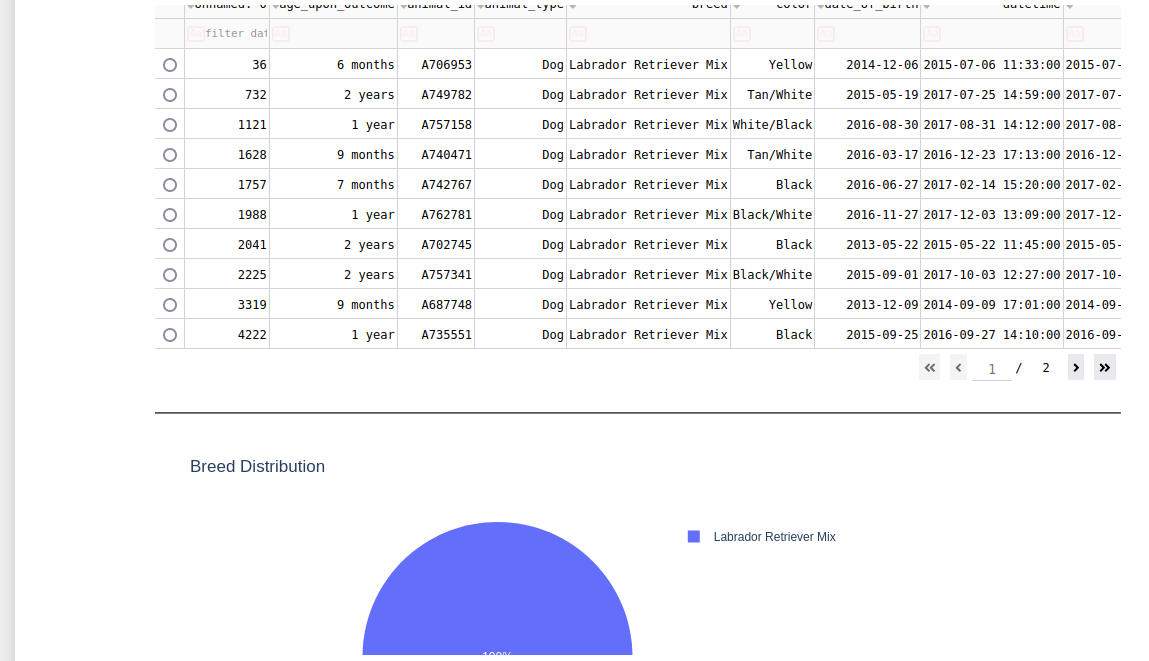
Description: This screenshot shows the data table filtered by the "Water Rescue" option, displaying only relevant entries.

4. Mountain or Wilderness Rescue Filter

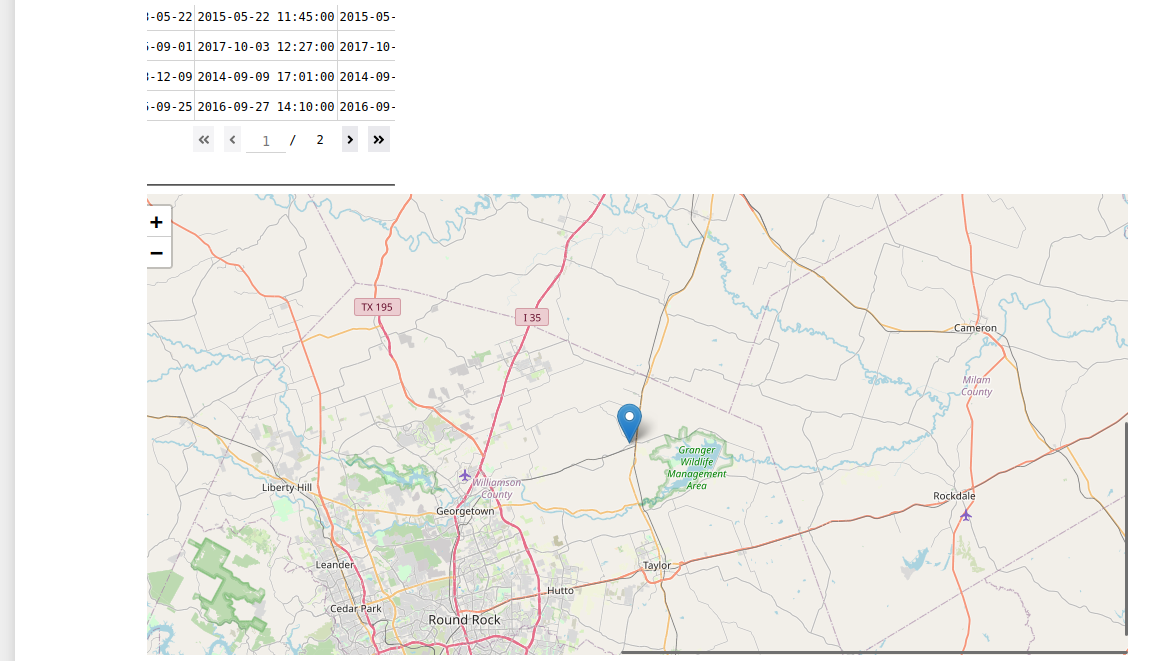
Description: This screenshot shows the data table filtered by the "Mountain or Wilderness Rescue" option.

5. Disaster or Individual Tracking Filter

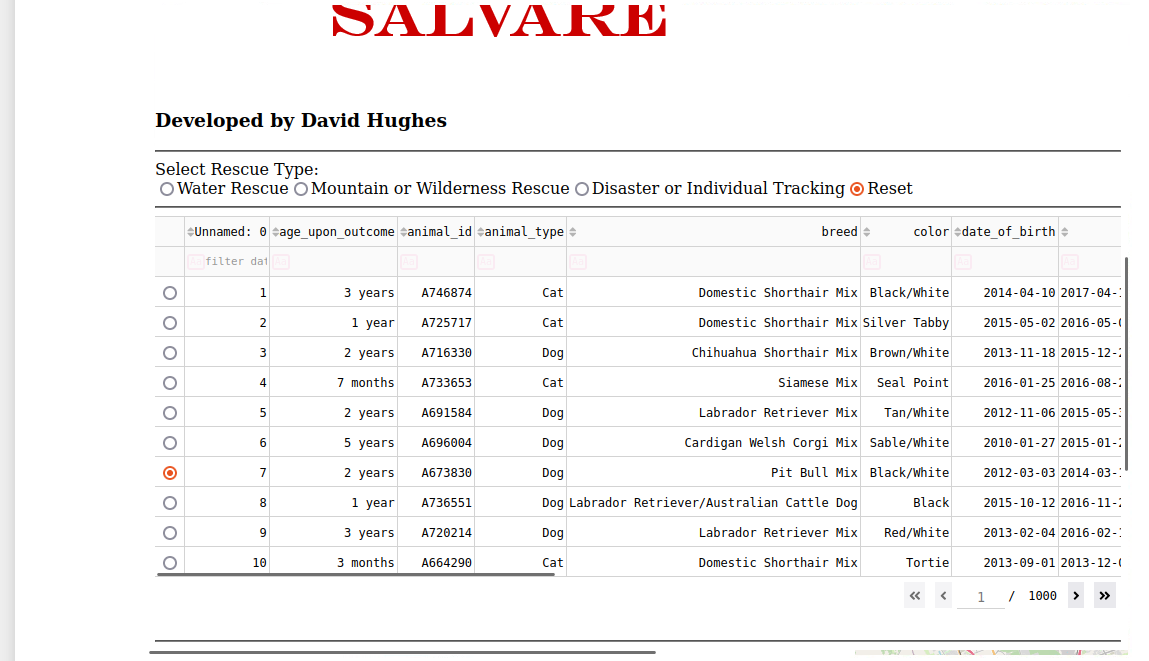
Description: This screenshot shows the data table filtered by the "Disaster or Individual Tracking" option.

6. Breed Distribution Graph 

Description: This screenshot displays the breed distribution graph based on the filtered data.

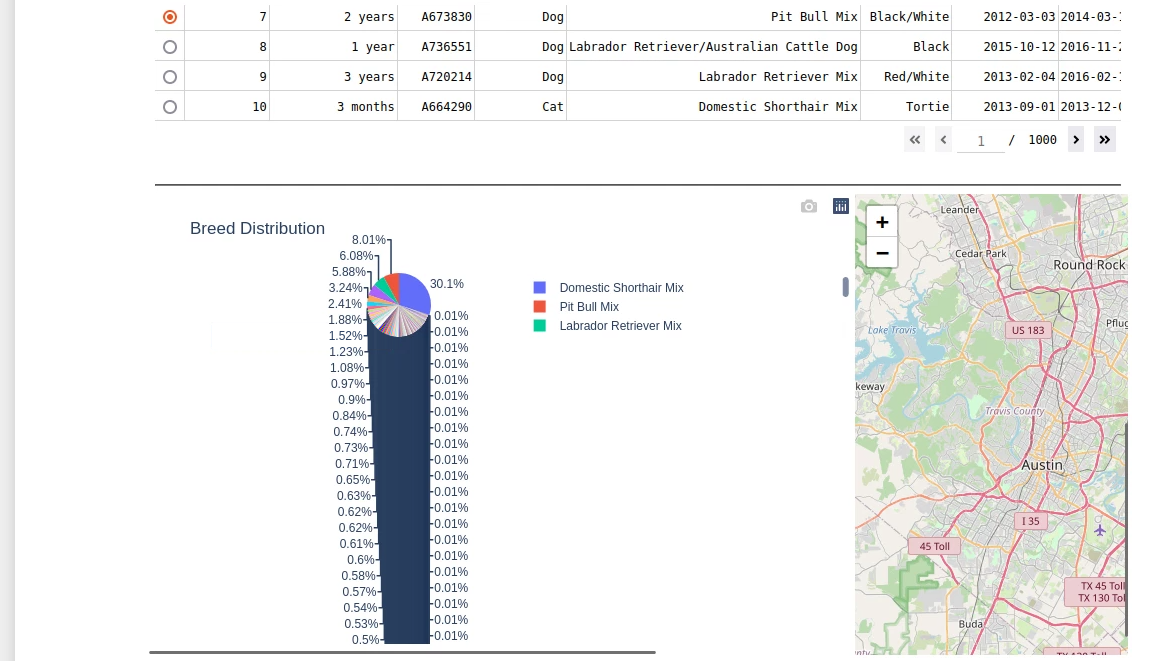
7. Map with Selected Animal Location 

Description: This screenshot shows the map with the location of the selected animal from the data table.

8. Data Table with Selected Row 

Description: This screenshot has no filters and highlights a selected row in the data table, which updates the breed distribution graph and map.

9. Full Dashboard View



Description: This screenshot shows the full dashboard with the data table, breed distribution graph, and map.