Name: Timothy Johnson  
Date: February 21st, 2025  
CS 340  
Professor Iles

**CS340 Project Two Grazioso Salvare Dashboard**

The goal of this project was to create an interactive dashboard for Grazioso Salvare, an animal shelter organization. The dashboard allows users to filter and view information about rescue animals, their breed, sex, age, and geographical data, all while displaying maps and charts. The users can interact with the dashboard through the widgets such as radio buttons and tables. The dashboard uses MongoDB as the database to store animal data, Dash for creating the web application interface, and Plotly for visualizing data.

**Dashboard features:**

Widgets for Filtering Data: Users can filter rescue dogs based on three categories: Water Rescue, Mountain or Wilderness Rescue, and Disaster or Individual Tracking.

Interactive Data Table: Displays filtered information in a table format that users can scroll through and select.

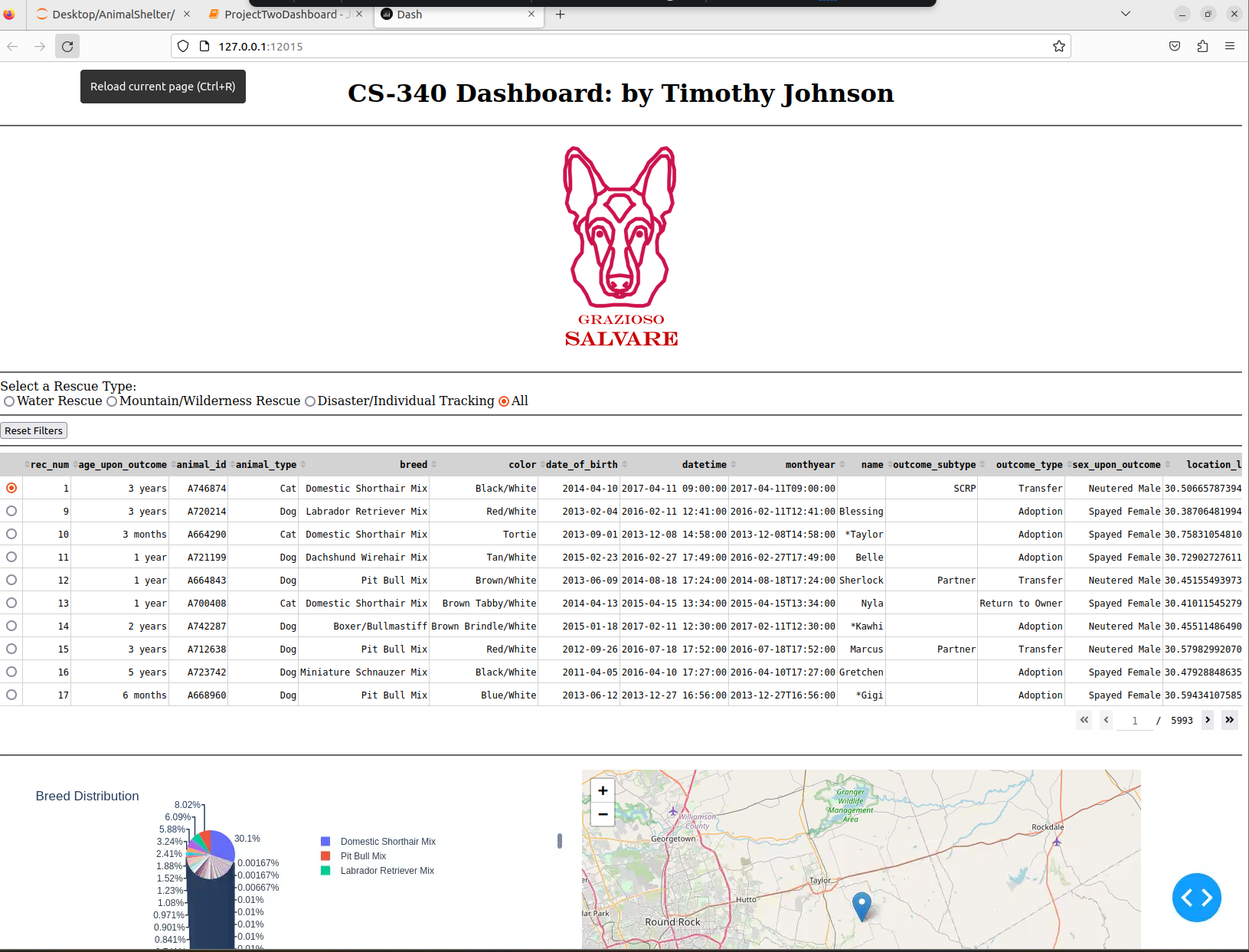
Charts: After applying filters, users can view a pie chart showing the distribution of breeds of the selected rescue animals.

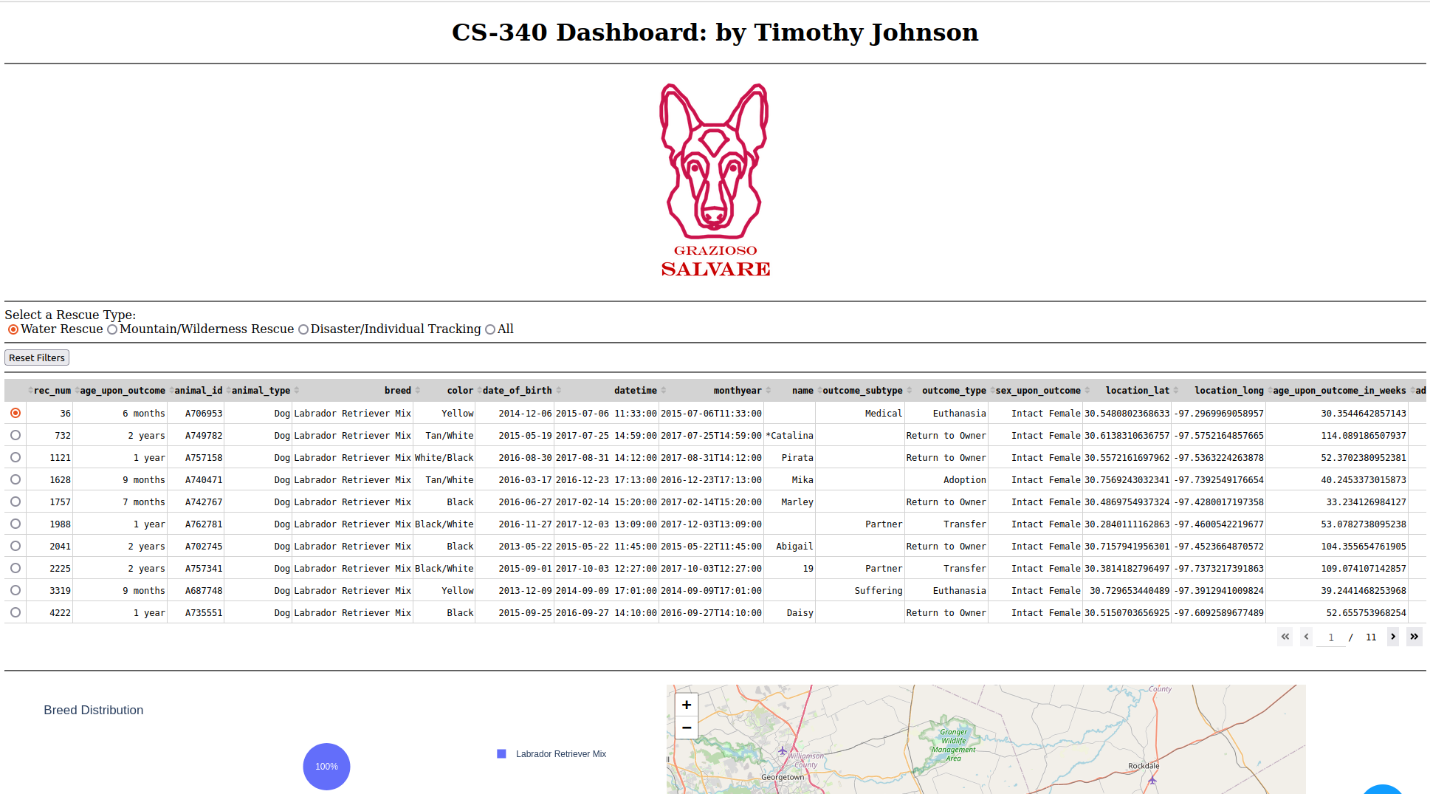
Geographic Map: Clicking on a specific animal in the table brings up a map showing the animal's geographic location with a marker.

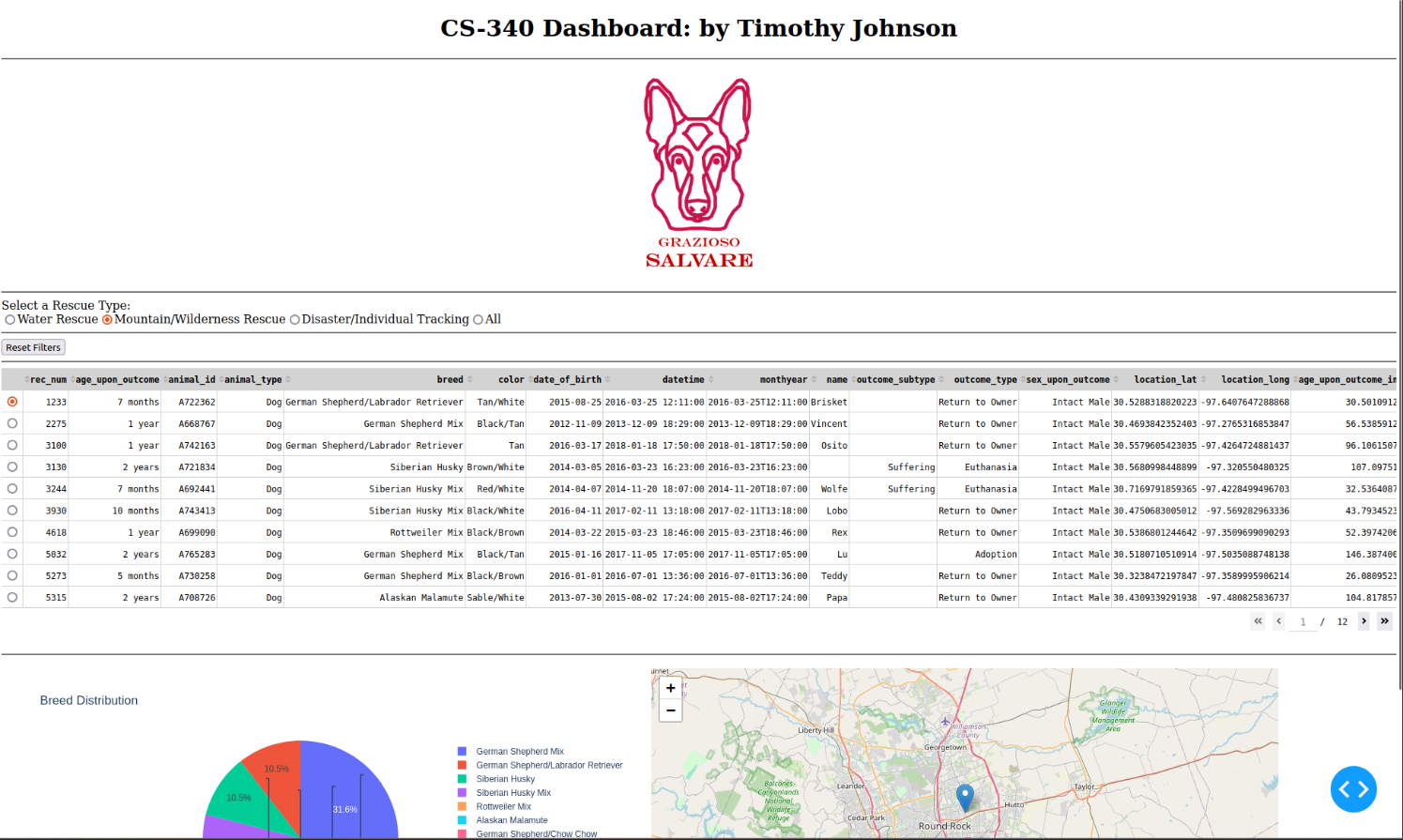
Reset Button: Resets all filters and restores the dashboard to its initial state.

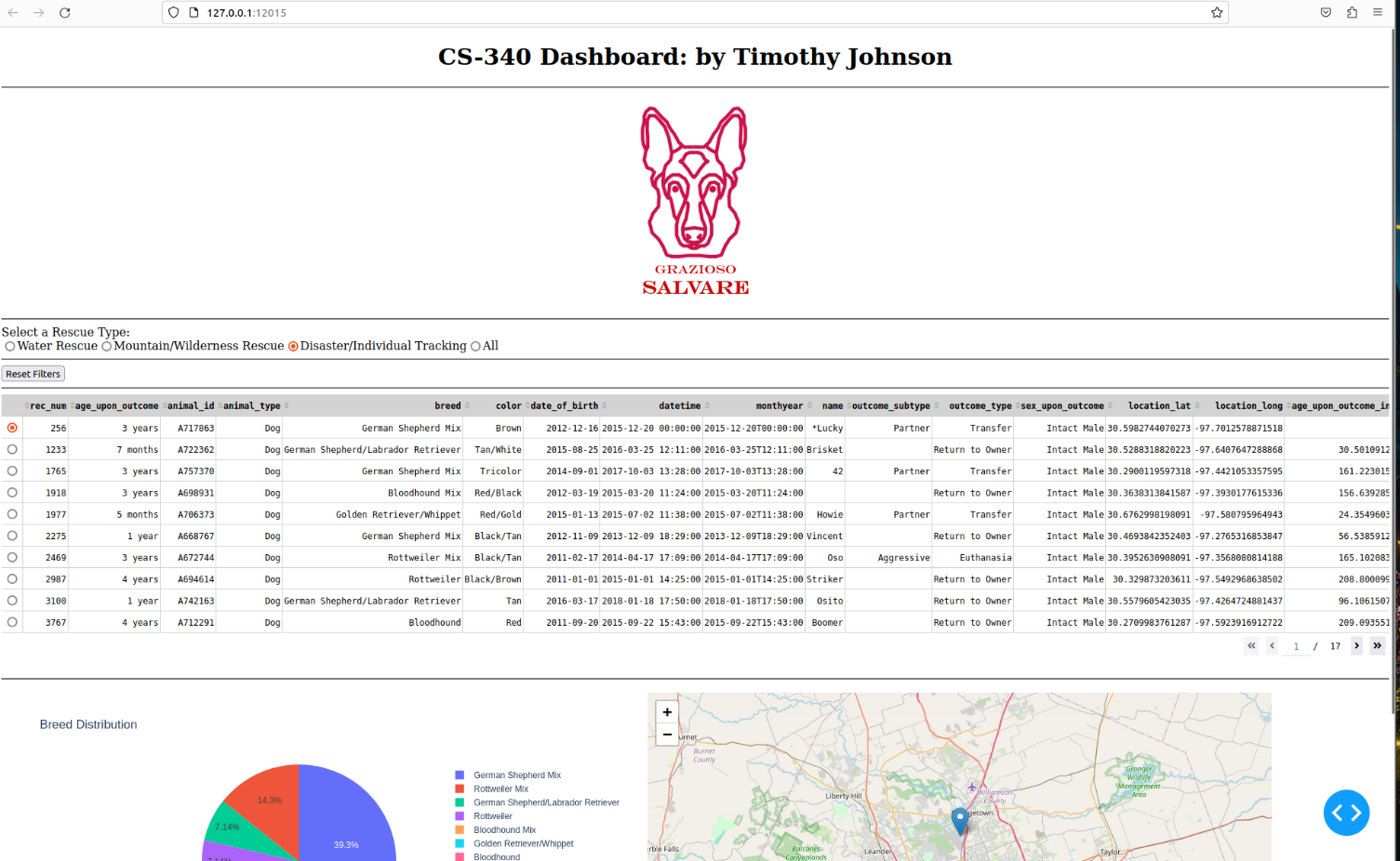
**Screenshots**

Below are the screenshots demonstrating the functionality of the dashboard, including the starting state and after each rescue filter is applied:









* Starting State: Shows the dashboard with all widgets unfiltered, displaying the full dataset.
* Water Rescue Filter Applied: After selecting "Water Rescue" from the widget, the dashboard shows the filtered data of Water Rescue dogs.
* Mountain or Wilderness Rescue Filter Applied: After selecting "Mountain or Wilderness Rescue," the dashboard shows only the relevant dogs.
* Disaster or Individual Tracking Filter Applied: After selecting "Disaster or Individual Tracking," the dashboard shows the relevant rescue animals.

**Tools**

MongoDB: MongoDB was used as the database for storing animal data. This NoSQL database is perfect for handling animal data. The pymongo library was used to interface MongoDB with Python, providing an efficient way to interact with the database.

Dash Framework: Dash, a Python framework for building web applications, was used to create the interactive dashboard. Dash provides a powerful interface to design web applications with minimal HTML/CSS knowledge. With Dash, I was able to create interactive widgets like radio buttons, dropdown menus, and data tables.

Plotly: Plotly was used for generating the pie chart visualization of breed distributions and other interactive plots. Plotly’s integration with Dash made it easy to add charts that are responsive and visually appealing.

Python and Pandas: Python and Pandas were essential for processing and manipulating the data before it was sent to the Dash application. Pandas helped clean, filter, and manage the data pulled from MongoDB and allowed me to efficiently handle the complex filtering logic for the dashboard.

Leaflet: Dash-Leaflet was used for embedding interactive maps in the application. The map displays the location of rescue animals based on their latitude and longitude.

Pymongo: Pymongo was used for interacting with MongoDB. It helps with inserting, updating, and retrieving documents from the database. Commands like insert\_one, update many, and read() were used for creating and querying the shelter database.

**Installation**

To get the Grazioso Salvare dashboard running on your local machine, follow these steps:

Install MongoDB:  
Download MongoDB from the official site: [MongoDB Installation Guide](https://www.mongodb.com/docs/manual/installation/).

Start the MongoDB service

Install Python:

Ensure you have Python 3.8+ installed on your machine.

Install Required Libraries:

You'll need to install the necessary Python libraries to run the project. Open your terminal. & connect to the database.

MongoDB Interaction: This project relies on PyMongo for accessing the MongoDB database. You will also use bson.objectid.ObjectId to handle object IDs in MongoDB.

Set Up MongoDB Database:

After MongoDB is running, make sure the animal data is inserted into the database. You can use the CSV file or manually populate the MongoDB collections using MongoDB Compass or a MongoDB client.

Run the Dashboard Application:

Once MongoDB is set up and the required libraries are installed, navigate to the project directory in your terminal, and start the Dash app.

**Challenges Encountered**

Complexity in Filtering: Initially, the logic for filtering the rescue animals based on different categories was complex. I had to ensure that the correct criteria were applied (e.g., breed, sex, age) and that the data was properly updated in real time. This challenge was overcome by thoroughly testing the filtering logic.

Handling Missing or Inconsistent Data: There were instances where data had missing values or inconsistent formats, especially with animal sex and breed information. These issues were handled by adding validation checks and filtering out invalid or missing data before displaying it in the dashboard.