**Grazioso Salvare Animal Rescue Dashboard**

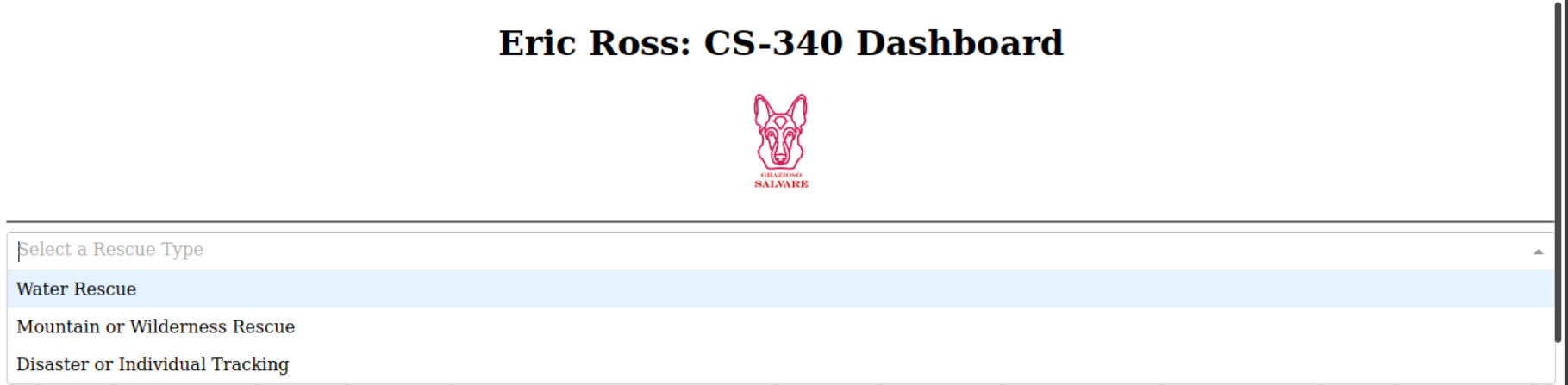
**Overview**

This project involved the creation of an interactive dashboard for Grazioso Salvare to streamline and enhance animal rescue efforts. The dashboard displays data from an animal shelter database, allowing users to filter, visualize, and interact with the data effectively.

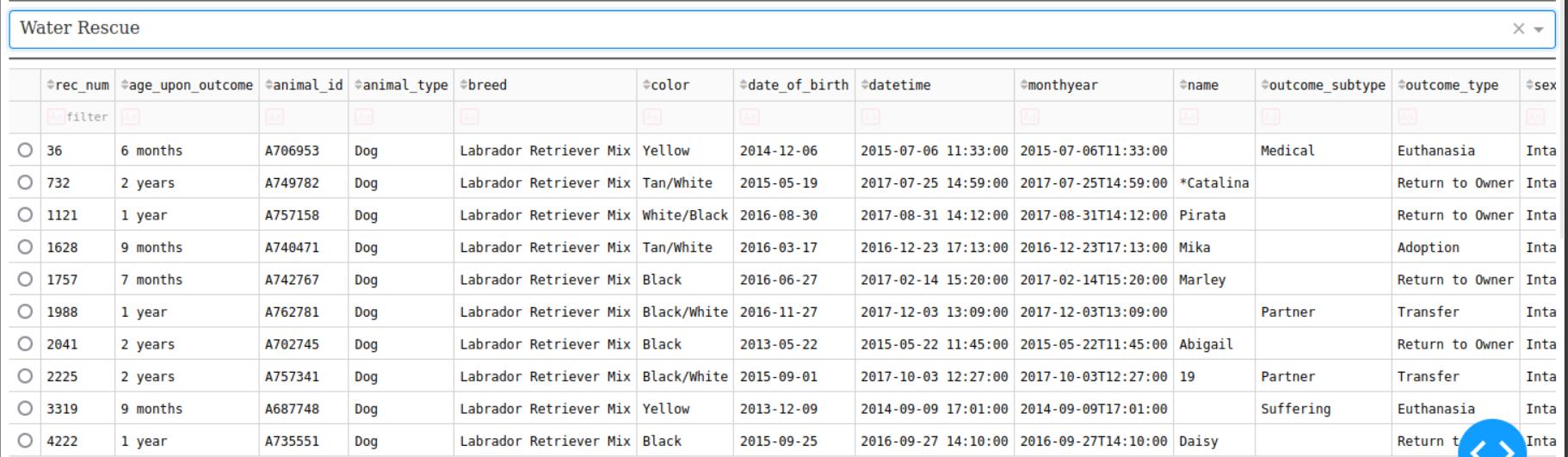
**Required Functionality**

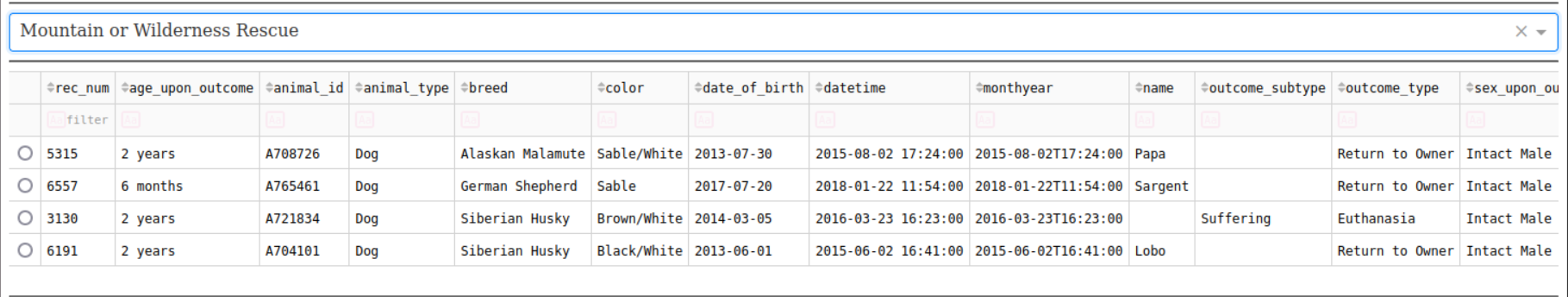
**Features**

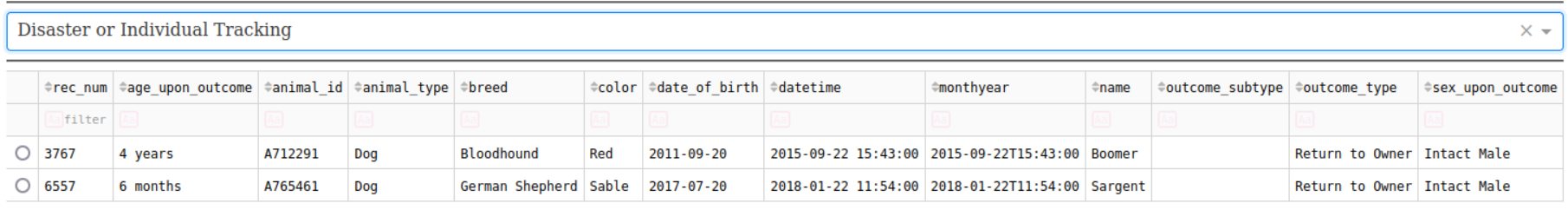
Interactive Dropdown: Users can filter animals based on rescue types: "Water Rescue," "Mountain or Wilderness Rescue," and "Disaster or Individual Tracking."

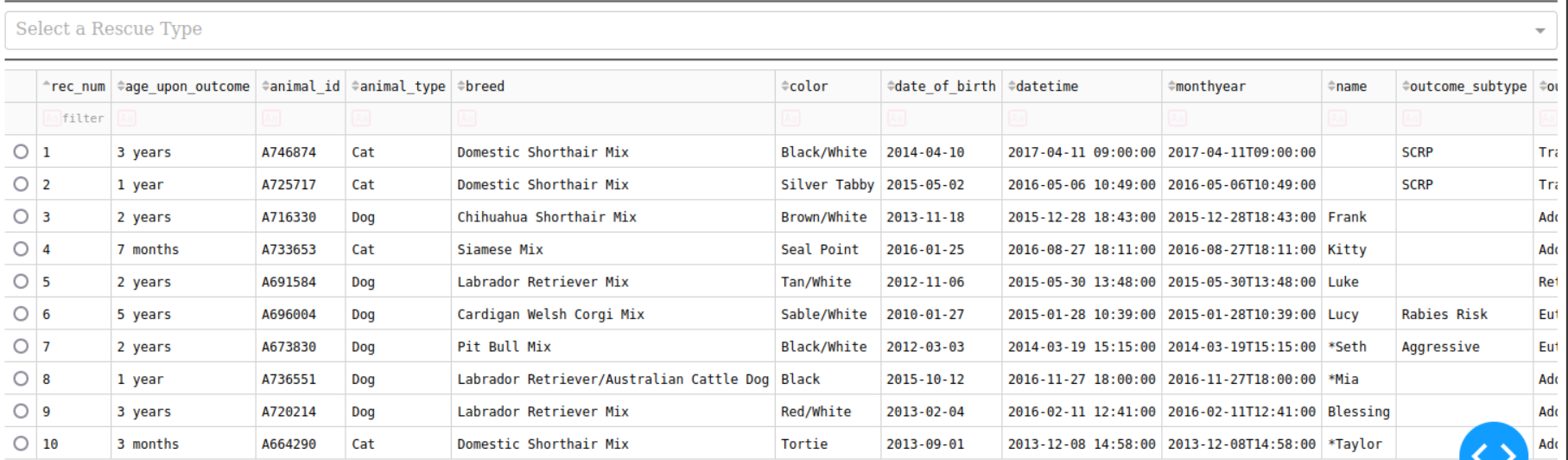


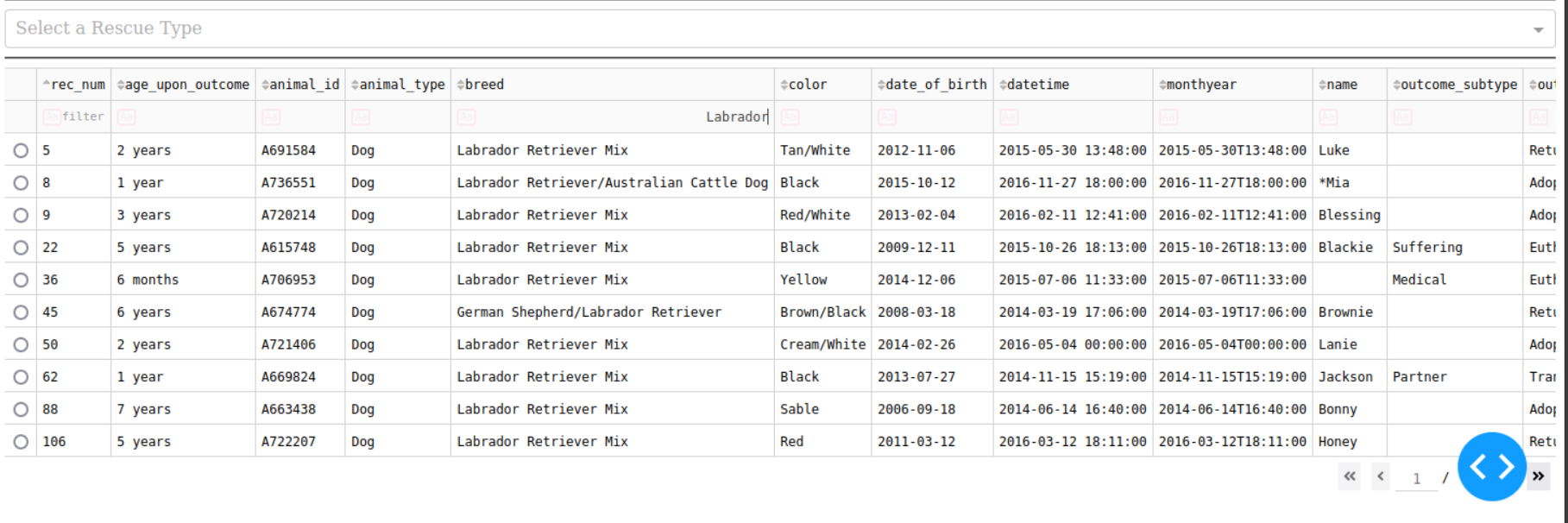
Data Table: Displays animal data dynamically based on the selected filters. It includes sorting, filtering, and row selection functionalities.

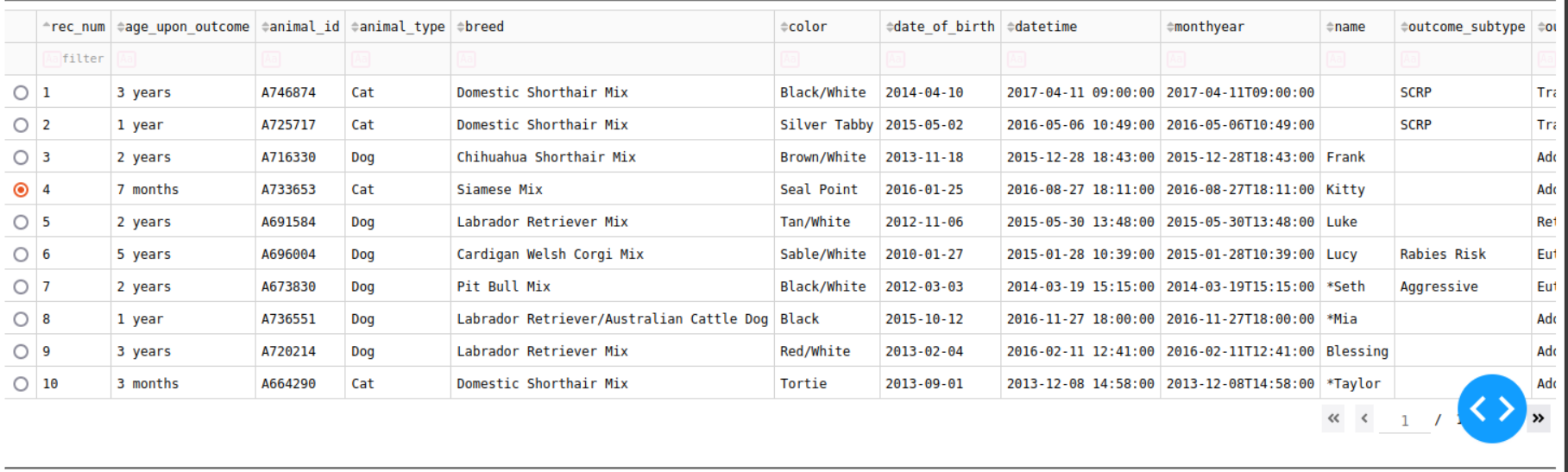




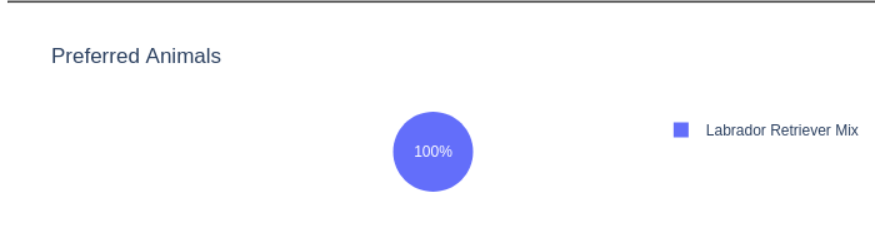


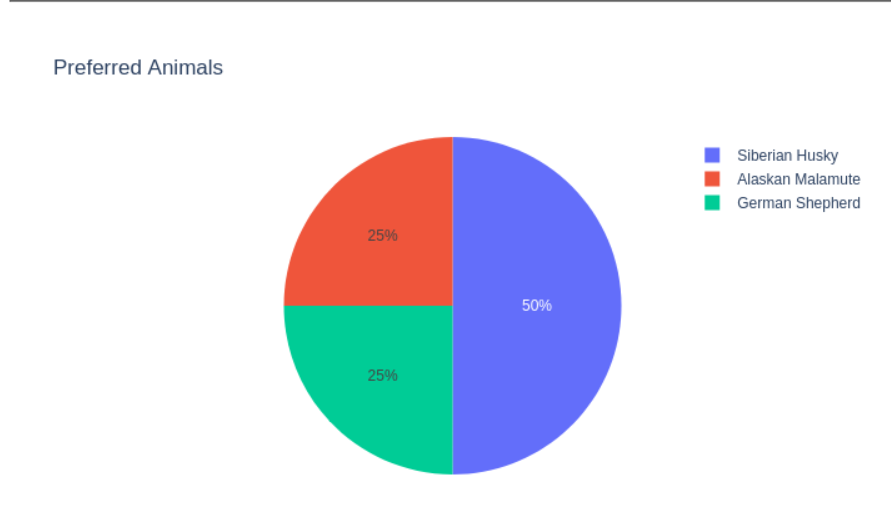


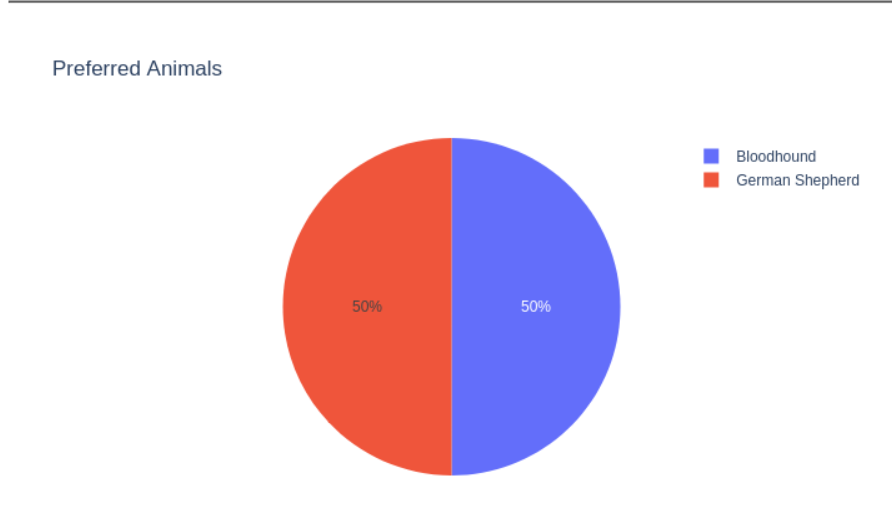




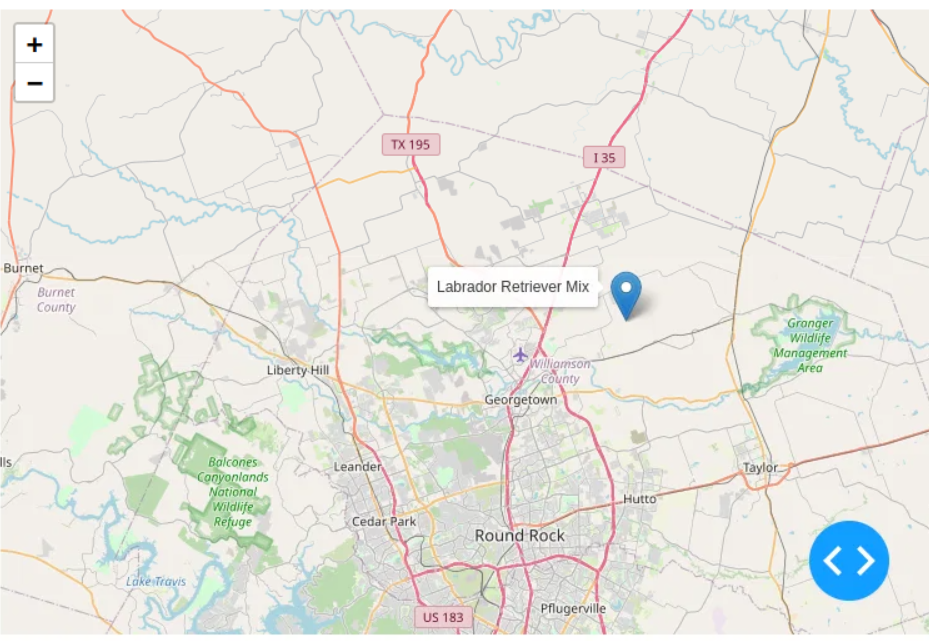
Pie Chart: Visualizes the distribution of animal breeds based on the selected filter.







Geolocation Map: Displays the location of a selected animal from the data table on an interactive map with markers and tooltips.



**Tools Used**

MongoDB: MongoDB served as the model component, providing a NoSQL database to store and retrieve animal shelter data. Its key features are that it handles unstructured data effectively and has native support for JSON-like documents allowing for seamless interaction with Python. MongoDB’s scalability and compatibility with Python’s PyMongo library made it ideal for this project.

Dash Framework: Dash was used for building the dashboard’s user interface and integrating interactivity. Its key features are that it’s easy to integrate with Plotly for data visualization and the use of callback functions to enable dynamic interactions between components. Dash simplifies web application development which makes it accessible for data scientists and developers.

Other Tools: Python was used for all data manipulation, server interactions, and application logic. Jupyter Notebook simplified testing and debugging during development. Dash Leaflet enabled the creation of interactive maps.

**Steps Taken**

Setup: The necessary libraries were installed including dash, dash-leaflet, plotly, and pandas. Next, MongoDB was configured, and I imported the animal shelter data.

Backend Development: I created a Python class to handle CRUD operations for MongoDB. Then I queried and cleaned data to remove unnecessary fields such as ‘\_id’.

Frontend Development: I designed the dashboard layout using Dash components such as dropdowns, tables, charts, and maps. I then implemented callback functions to enable interactions between components.

Testing: I debugged and verified the functionality of each component. Edge cases, such as empty or missing data were also addressed.

Deployment: Finally, I fine-tuned the application layout and styling.

**Challenges and Solutions**

The first challenge was data compatibility. The initial data included fields like ‘\_id’ that are incompatible with Dash’s data table. This was resolved by dropping the field during preprocessing. Another challenge was creating dynamic seamless interactions between dropdown filters, data table, and visualizations. This was resolved by using structured callback functions with clear inputs and outputs. The geolocation map created another challenge where the map markers were not displaying due to missing latitude and longitude data. This was resolved by validating data and adding error handling for missing coordinates. Finally, I addressed UI enhancements by adjusting the layout for responsiveness and simplicity.