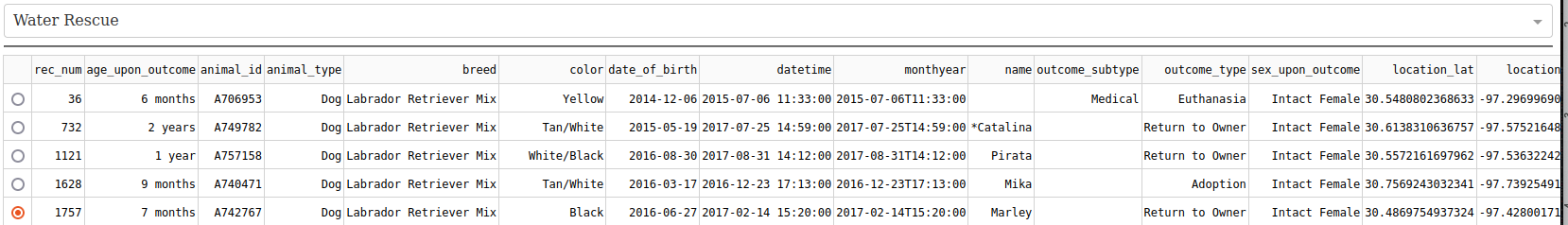
**Project Overview:**

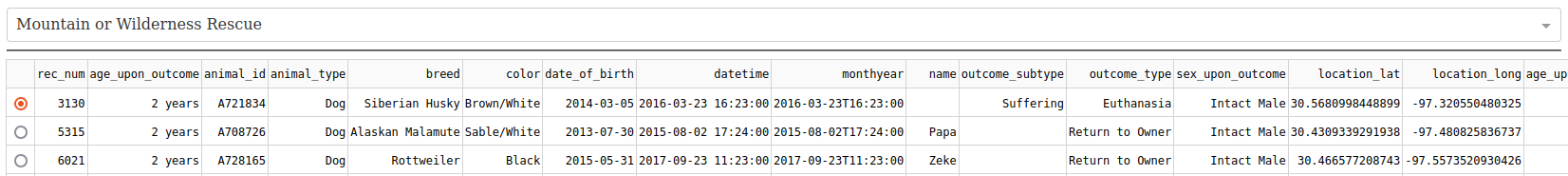
This dashboard was created for Grazioso Salvare to allow users to interactively filter and visualize data related to animal rescues based on certain conditions and types. Also allowing a filter for certain types of rescue dogs required. It provides an intuitive user interface where data can be visualized through tables, pie charts, and interactive maps.

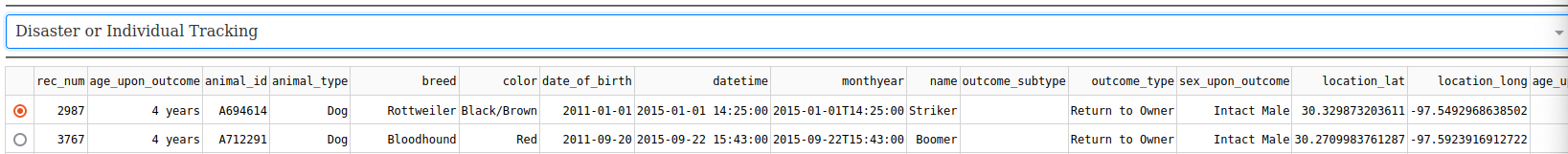
**Required Functionality:**

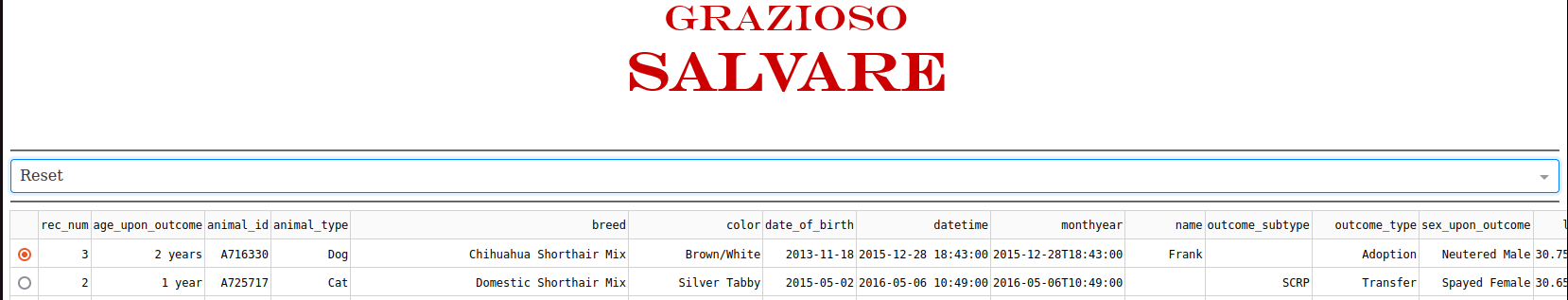
The dashboard should allow users to:

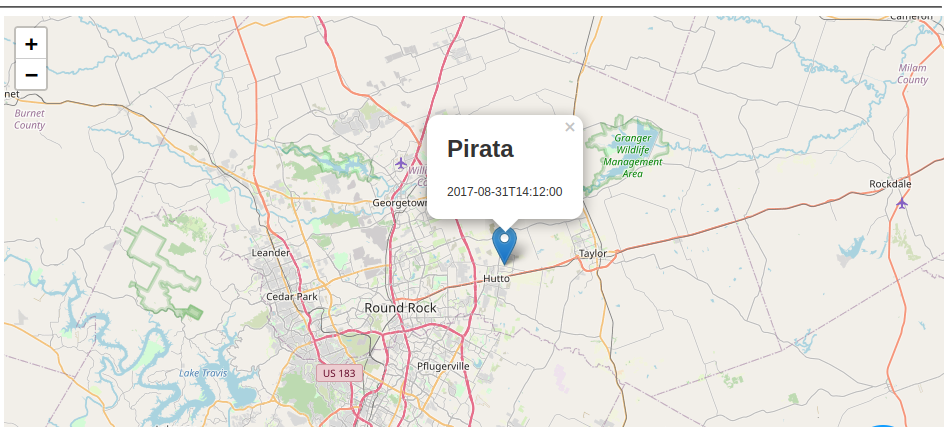
1. Select a specific type of rescue from a dropdown (e.g., "Water Rescue", "Mountain or Wilderness Rescue", etc.).
2. Display the corresponding filtered data in a table.
3. Visualize the preferred animals for a particular rescue type using a pie chart.
4. Show the location of the selected animal on an interactive map.

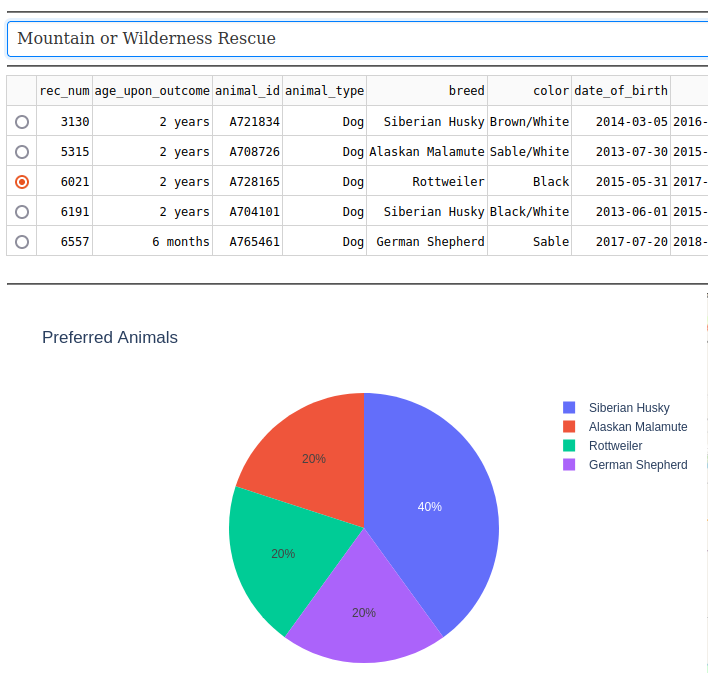
**Water Rescue Filter Screenshot:**

**Mountain or Wilderness Rescue Filter Screenshot:**

**Disaster or Individual Tracking Filter Screenshot:**

**Reset (displays all data, cut off for screenshot size) // Also note logo (can see full logo in code)**

**Map working (Displays animal name along with other information):**

**Pie Chart**

**See commented code for proper naming conventions and further detail.**

**Tools and Technologies:**

**MongoDB:** A NoSQL database was chosen for its flexibility in storing structured and semi-structured data. It allows for easy integration with Python and can handle large amounts of data efficiently. MongoDB provides the model component for the dashboard.

**Rationale:** MongoDB's dynamic schema nature means we can store documents without a predefined schema, making it suitable for our varying data needs.

**Dash by Plotly:** This Python framework allows for the creation of web applications using pure Python, without the need for front-end technologies like JavaScript. Dash handles both the view and the controller components of our dashboard.

**Rationale:** Dash provides a simple and intuitive way to build interactive web applications directly using Python. This streamlines the process and reduces the need for multi-language expertise.

**Other tools/libraries:**

**Pandas:** For data manipulation and analysis.

**Plotly:** For creating visualizations.

**dash\_leaflet:** For map visualizations.

**Resources & Software:**

[MongoDB Official Documentation](https://www.mongodb.com/docs/)

[Dash by Plotly Documentation](https://dash.plotly.com/)

[Pandas Documentation](https://pandas.pydata.org/docs/)

[Plotly Documentation](https://plotly.com/python/)

**Project Steps:**

1. Established a connection to the MongoDB database and retrieved animal rescue data.
2. Used Pandas for data manipulation and cleaning.
3. Setup the Dash app layout, incorporating a dropdown for rescue type, a table for displaying data, a pie chart for visualization, and a map to show animal locations.
4. Implemented callbacks in Dash to handle interactivity and data updating based on user input.
5. Styled and fine-tuned the UI for better user experience.
6. Tested all functionalities and made necessary adjustments.

**How to reproduce from my files:**

1. Launch mongodb and import the aac\_shelter\_outcomes csv (not provided)
2. In Jupyter Notebook place the 3 files provided in a folder
3. Launch the notebook (ensure mongodb aac database is running)
4. View the Dash App running locally on your machine
   1. Enjoy!

**Challenges and Solutions:**

**Challenge:** Encountering issues with map markers not displaying correctly.

**Solution:** Adjusted the callback logic to fetch the correct data rows and ensured that latitude and longitude data were correctly parsed and converted.

**Challenge:** Integrating MongoDB with Dash. Had problems in previous modules as well.

**Solution:** Used PyMongo to fetch data from MongoDB and converted it into a DataFrame, making it easier to use with Dash and Plotly.

**Challenge:** Displaying the Grazioso logo on the dashboard.

**Solution:** Used base64 encoding to incorporate the logo image directly into the Dash layout.