**Grazioso Salvare Dashboard**

Matthew Tyson

Southern New Hampshire University

CS 340: Client/Server Development

Jeff H. Sanford, MS

October 15, 2023

**Grazioso Salvare Dashboard Project**

**Description**

The Grazioso Salvare dashboard is a tool designed for visualization and data interaction tailored for shelter animal data. The tool provides insights into different breeds suitable for various rescue missions, namely Water Rescue, Mountain or Wilderness, and Disaster or Individual Tracking. With an interactive user interface, users can filter and view specific data sets and visualize them in forms of tables and graphs.

**Required Functionality**

The dashboard provides the following functionalities:

1. View and filter animal data based on rescue mission types.
2. Reset filters to view the entire dataset.
3. Visualize preferred animals in a pie chart format.
4. Direct links to Grazioso Salvare's official website via its logo.

**Screenshots**

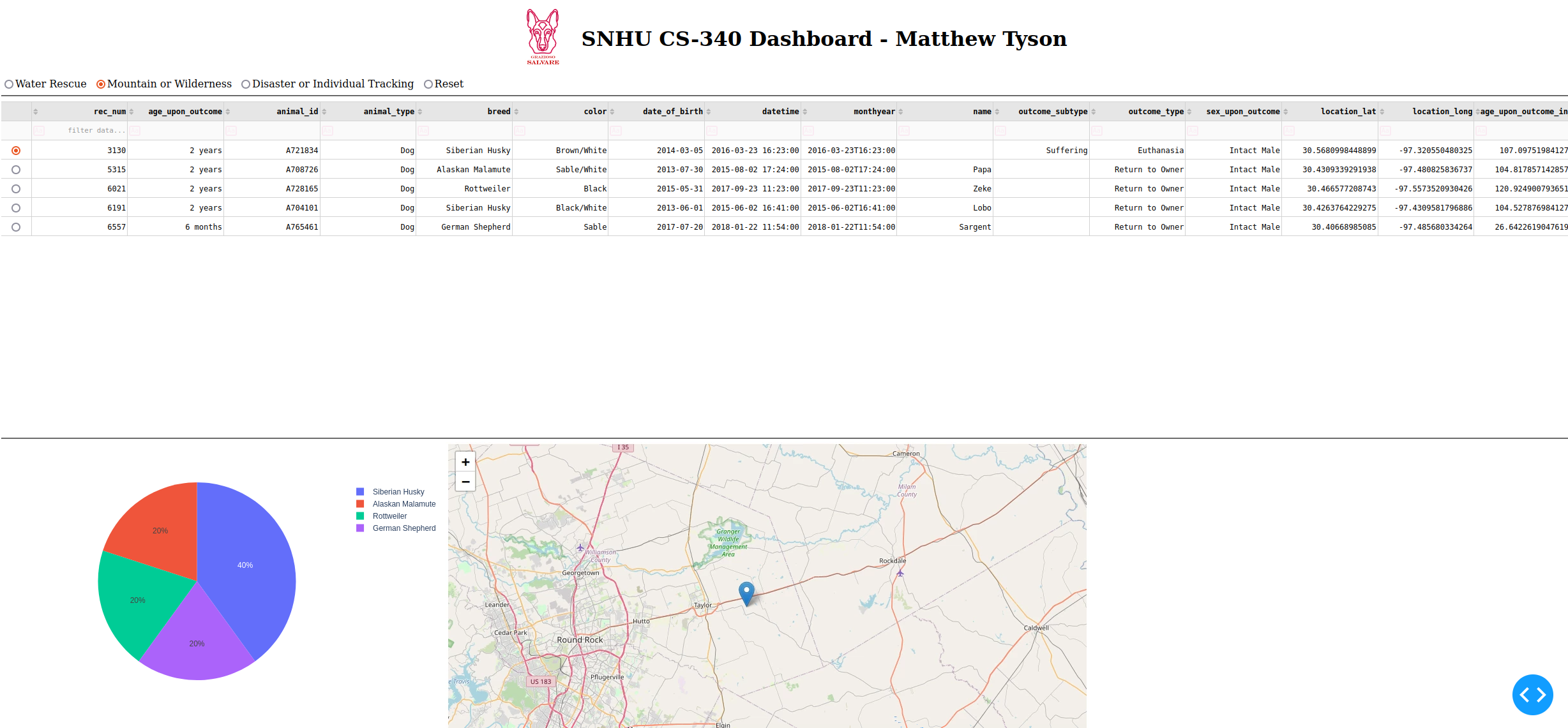
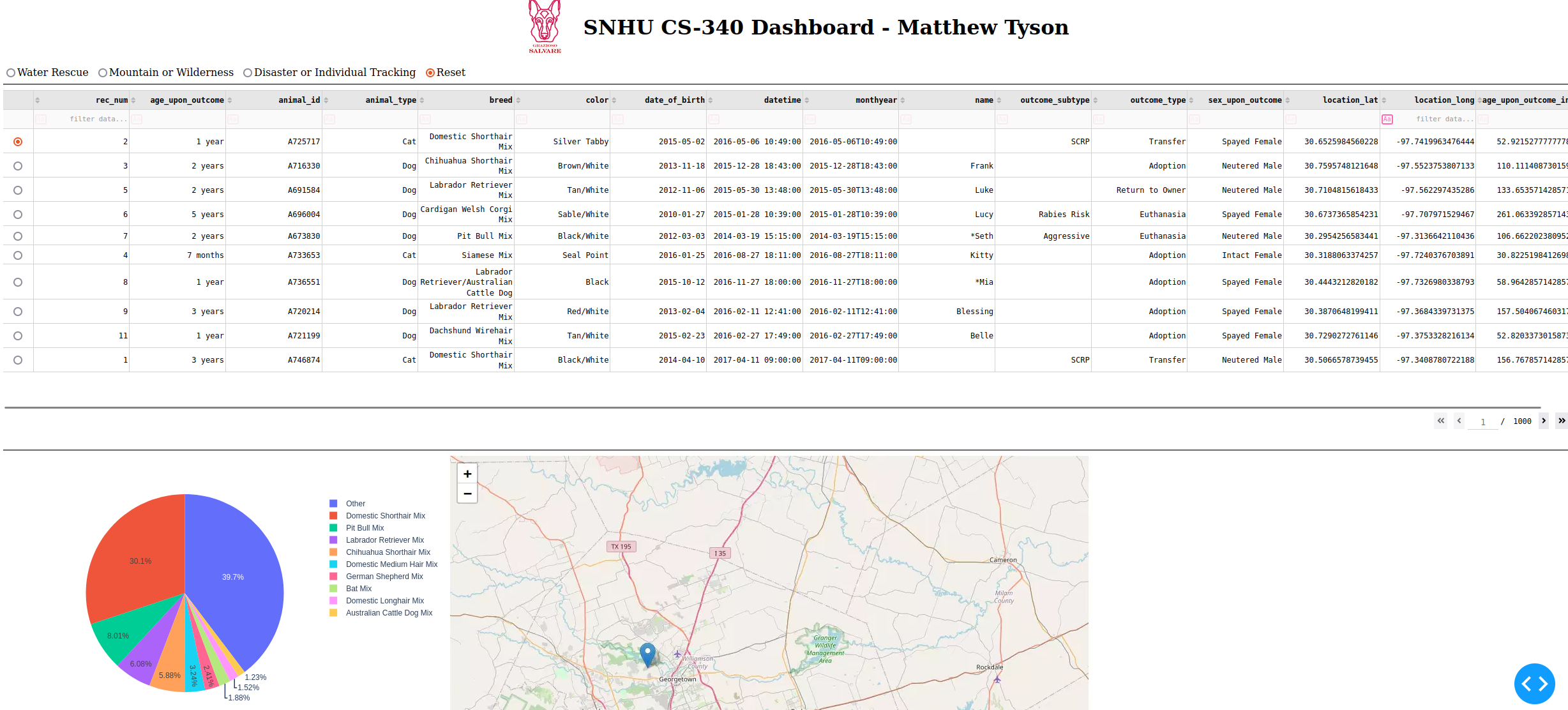
****

Figure 1 Filter animal data based on rescue mission type.



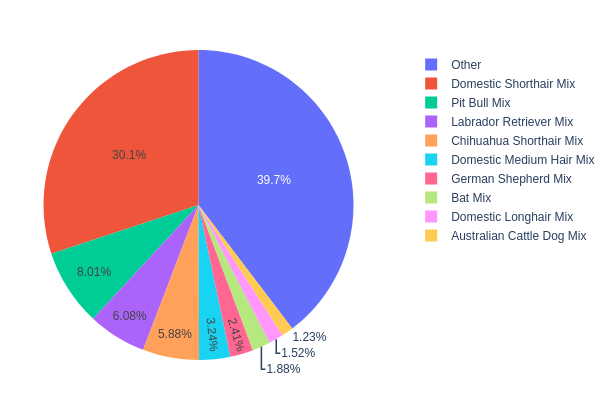
Figure 2 Clicking reset will reset all filters.

Figure 4 When there are more than 9 animals it creates an other category.

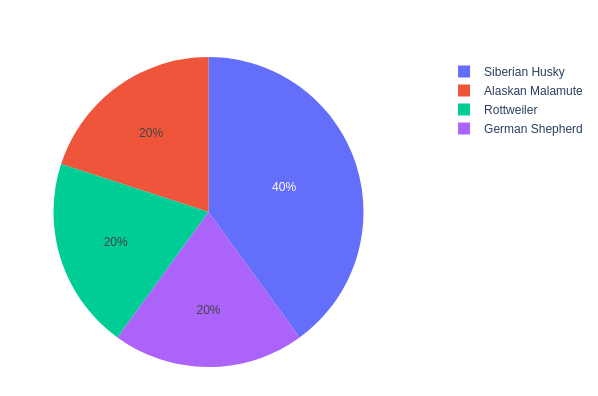


Figure Visualize preferred animals.

****

Figure 5 Clicking logo brings user to SHNU website.

**Tools and Technologies Used**

***MongoDB***

MongoDB was chosen as the model component of the development due to the following reasons:

1. **Schema-less Nature**: MongoDB being a NoSQL database, is schema-less which means any data can be stored in MongoDB directly without first defining its structure.
2. **Flexibility**: The dynamic schema allows for flexibility in data storage and modifications.
3. **Python Compatibility**: MongoDB offers strong support for Python, making it easier to interface with Python web frameworks.

***Dash***

Dash by Plotly was utilized as it provides a web application framework primarily for analytical web applications. It serves as both the view and the controller in our MVC architecture. Reasons for choosing Dash:

1. **Interactivity**: Dash provides components that enable users to interact with visualizations.
2. **Simplicity**: Building a web-based dashboard using Dash is straightforward with Python.
3. **Integration**: Dash integrates seamlessly with Plotly visualizations and pandas for data manipulation.

***Resources and Software Applications***

* [Dash by Plotly](https://plotly.com/dash/)
* [MongoDB](https://www.mongodb.com/)
* [pandas](https://pandas.pydata.org/)

**Steps to Reproduce the Project**

1. **Setup MongoDB**: Initialize a MongoDB instance and populate it with the provided shelter data.
2. **Setup Python Environment**: Install necessary libraries such as Dash, pandas, and pymongo.
3. **Run the Application**: Use the Python script provided to launch the Dash application.
4. **Interact with the Dashboard**: Use the radio buttons to filter animal data based on rescue mission types.

**Challenges and Solutions**

***Data Filtering Issues:*** There were challenges in ensuring the 'Reset' button cleared all filters. This was resolved by scrapping the button and instead just added the ‘Reset’ functionality to the radio selection.

***Styling and Alignment:*** Achieving a proper layout, especially aligning the logo with the title was a bit tricky. This was addressed using CSS flexbox for alignment.

***Data Visualization:*** Including only the top 9 breeds in the pie chart required a data manipulation logic that groups all other breeds as 'Others'. This was handled using pandas dataframe operations.