Janice Ainembabazi

Southern New Hampshire University

CS-340-18683-M01 Client/Server Development

Prof. Jeff Sanford

December 19th 2024

**Grazioso Salvare Dashboard**

**About the Project**

This dashboard is a project I created for my CS-340 project. The goal was to design an interactive tool for Grazioso Salvare that helps visualize Austin Animal rescue data. The dashboard lets users explore and filter the data based on rescue types (like water rescues or mountain rescues) and see the results in a table, charts, and even on a map.

**What the Dashboard Can Do**

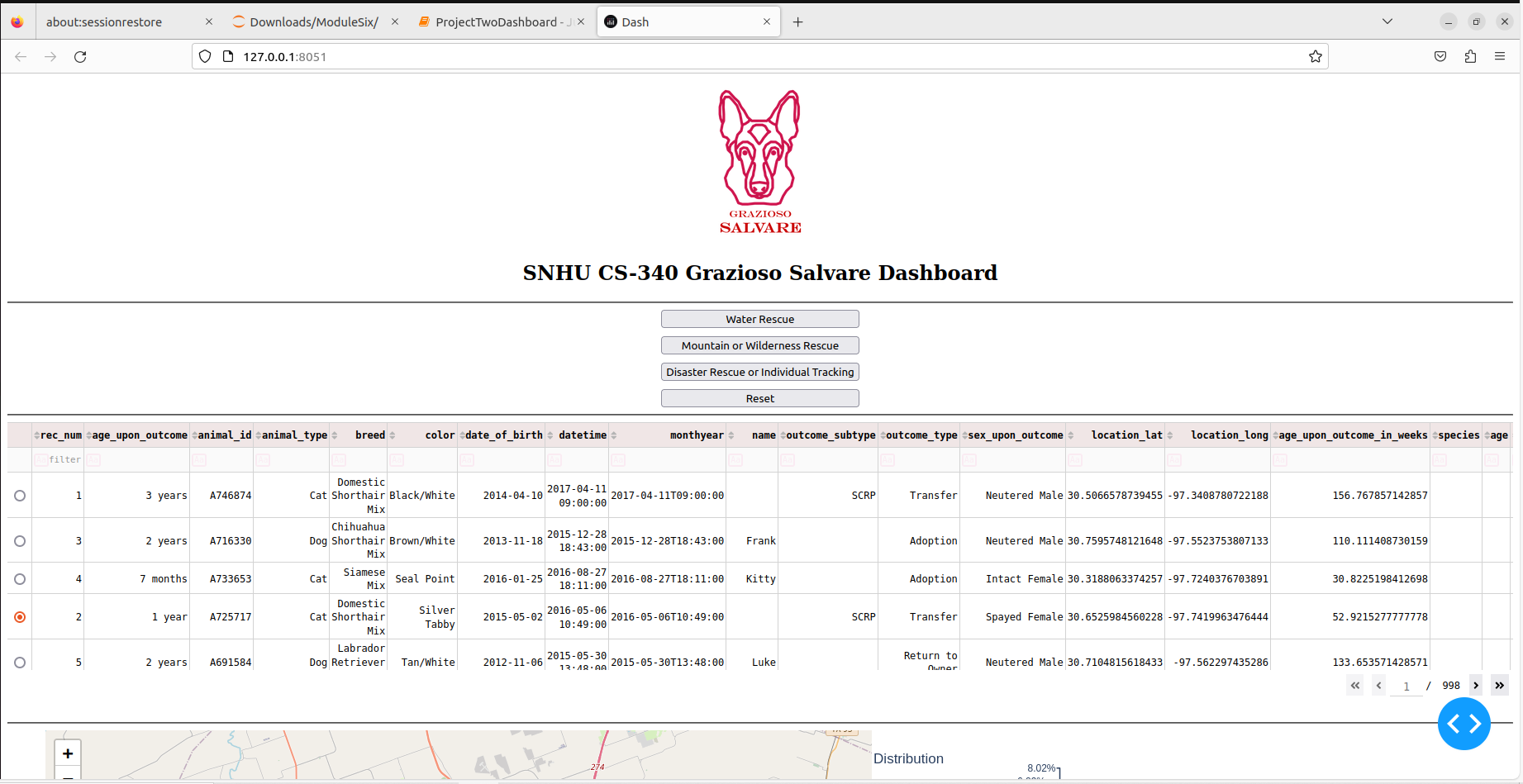
Here’s what the dashboard offers:

* Filters: Buttons let you filter the data into categories like Water Rescue, Mountain or Wilderness Rescue, Disaster or Individual Tracking, or you can reset everything to show all the data.
* Interactive Data Table: The table updates instantly based on your selection, making it easy to find specific information.
* Charts: The pie and bar charts provide a quick visual breakdown of the data.
* Map: The map shows the location of rescues and adjusts to the selected data.

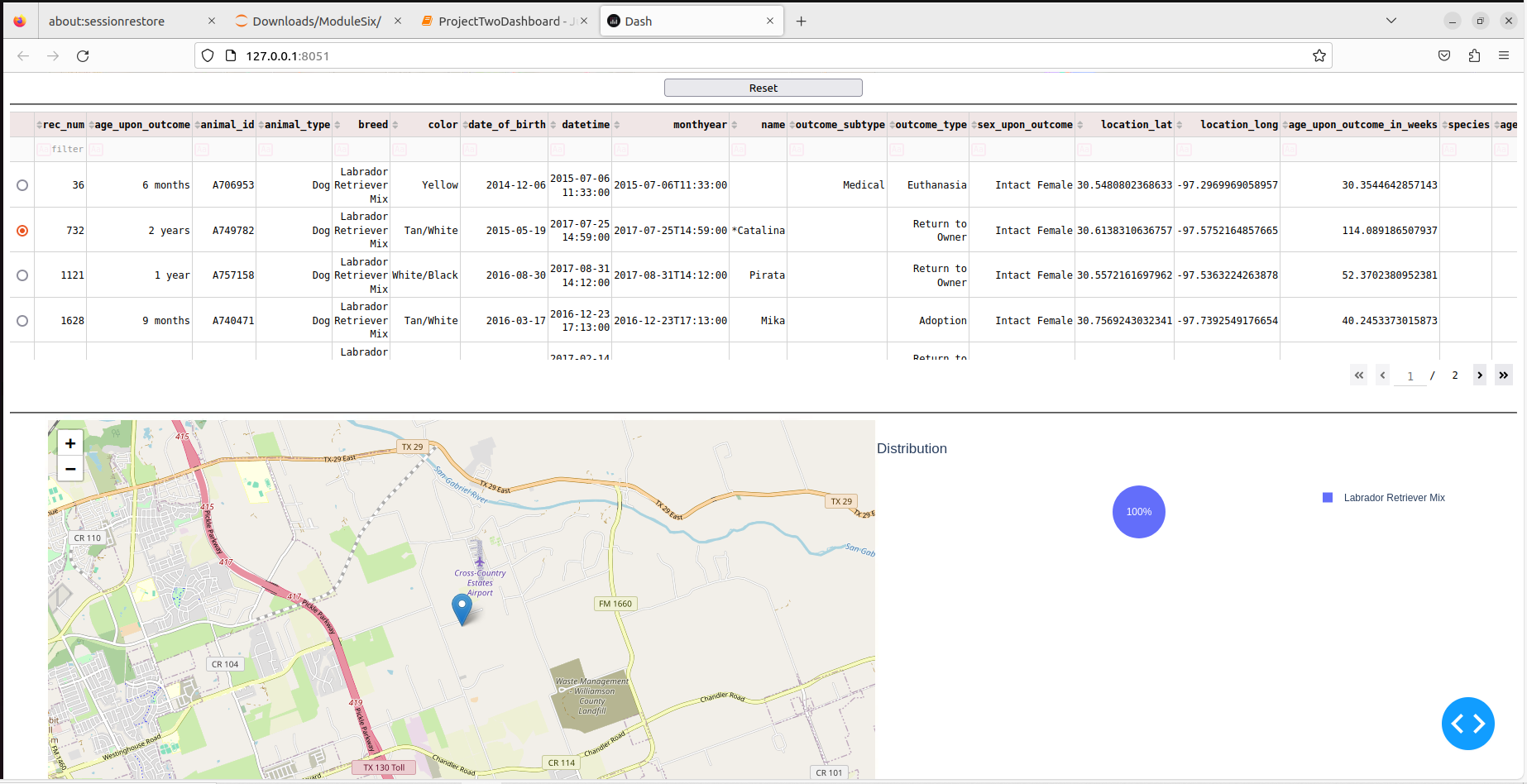
**Proof of Functionality**

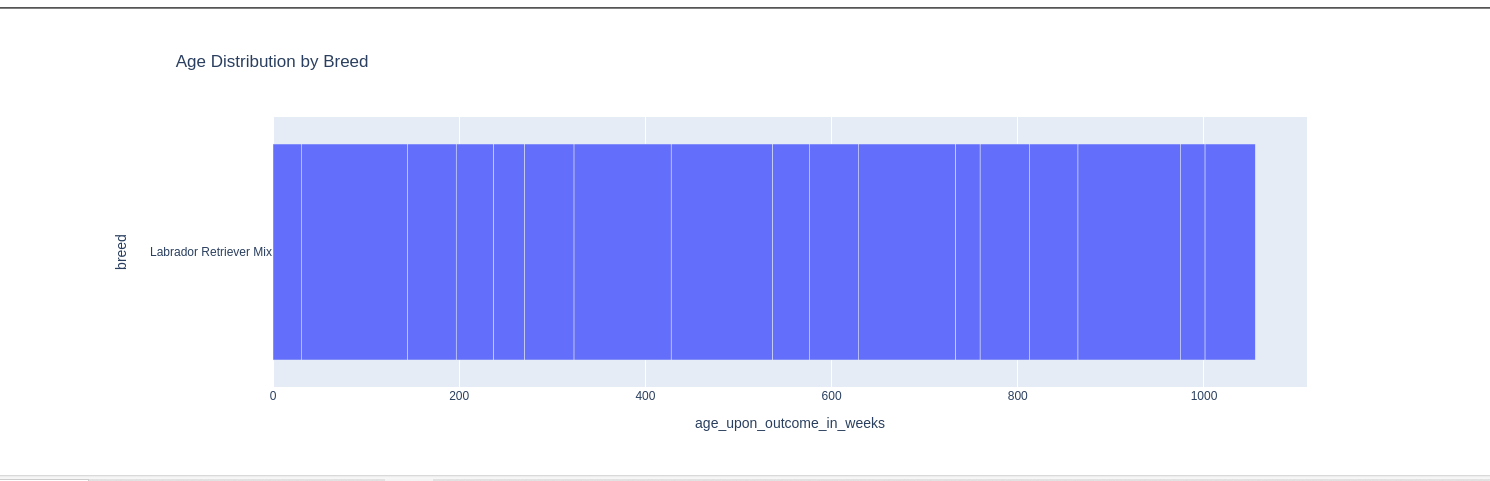
During testing, I captured screenshots to show that the dashboard works as expected. These include:

1. Starting View: Everything visible—filters, table, charts, and map in their default state.

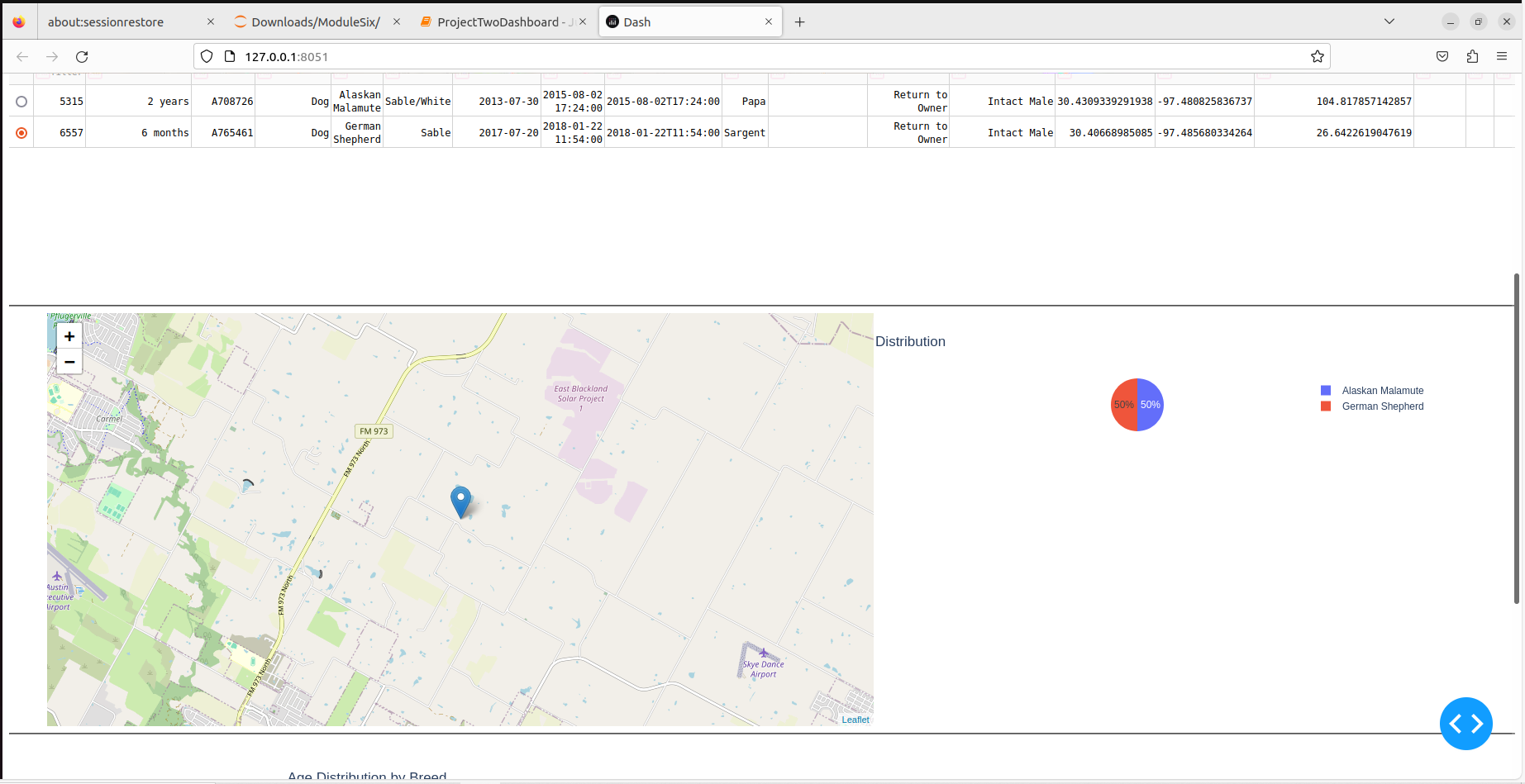


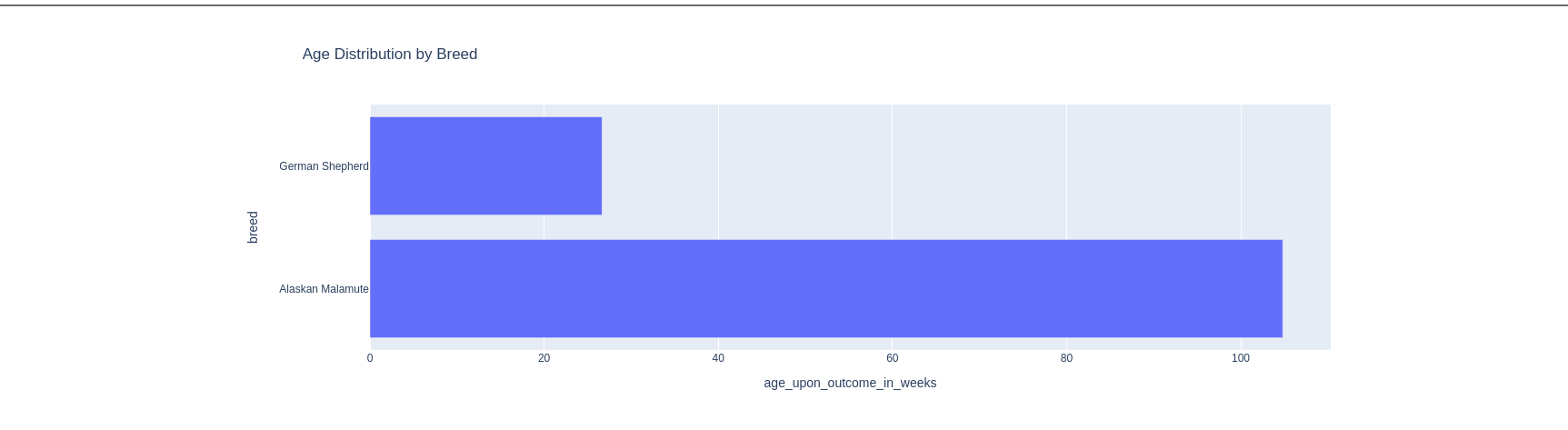
1. Water Rescue Filter: Clicking this button updates the data to show only water rescues and adjusts the visuals accordingly.



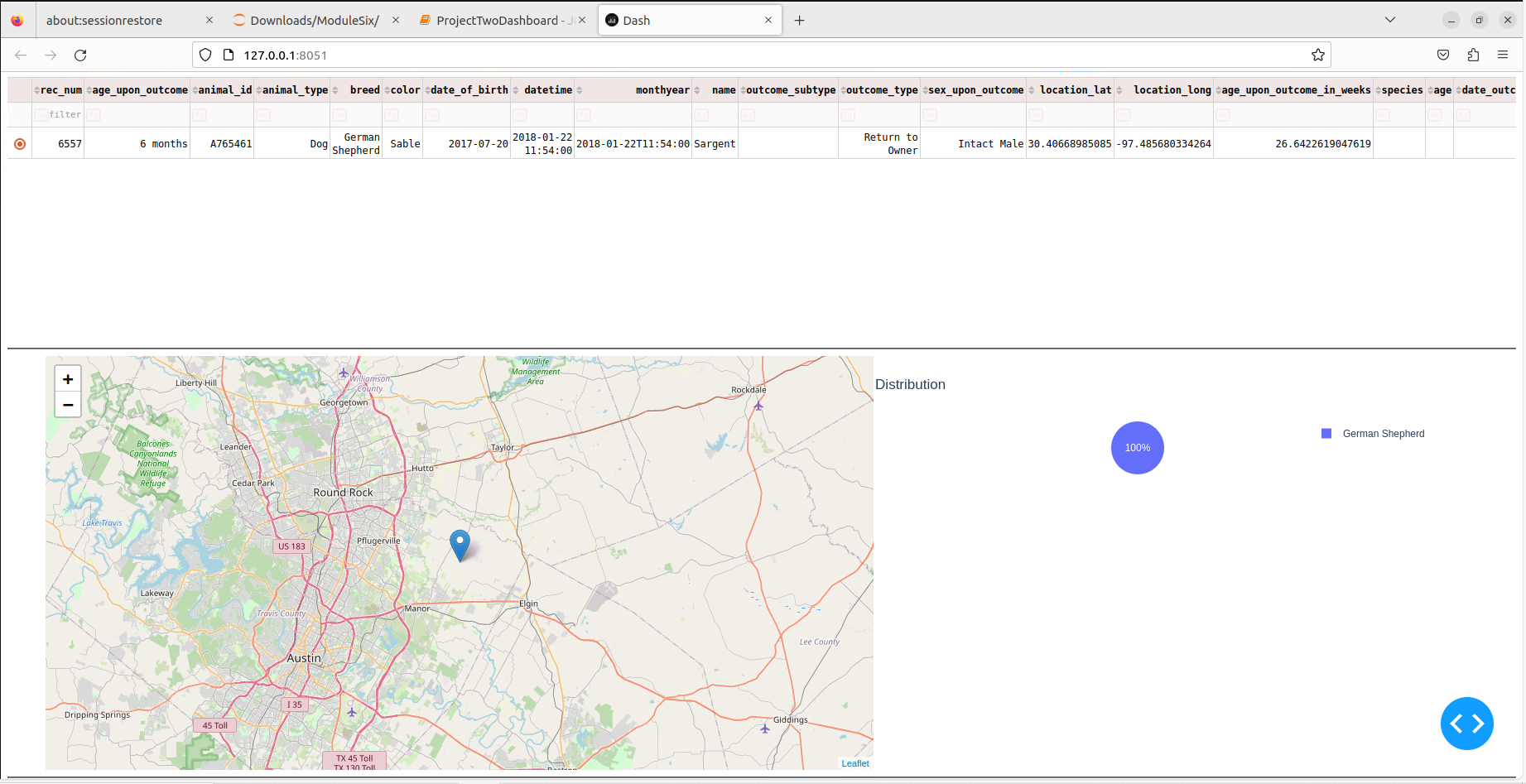


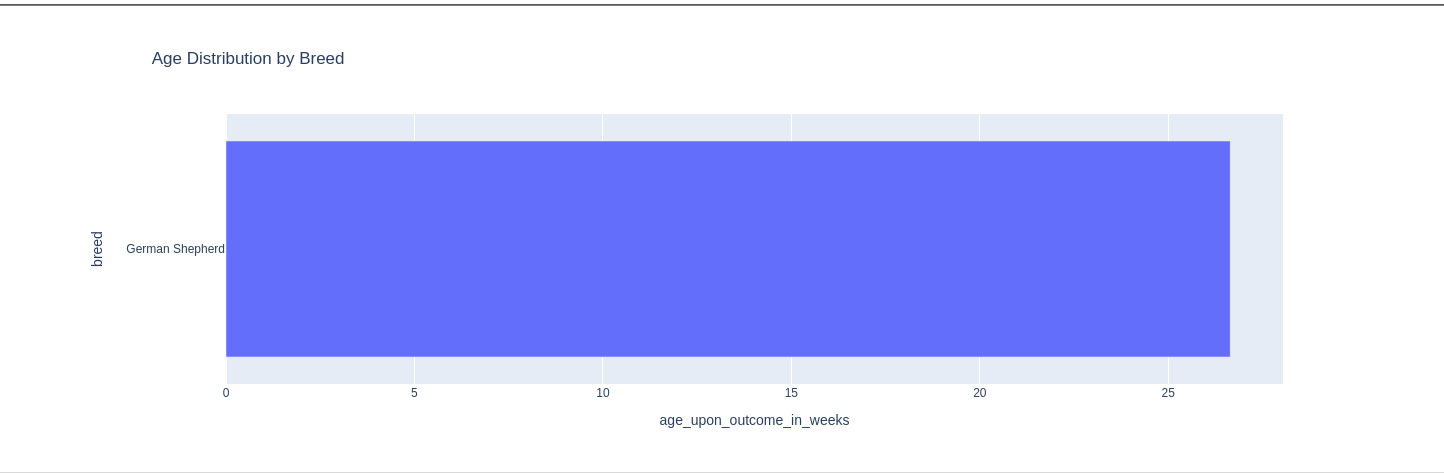
1. Mountain or Wilderness Rescue Filter: The table and visuals now reflect only mountain or wilderness rescues.



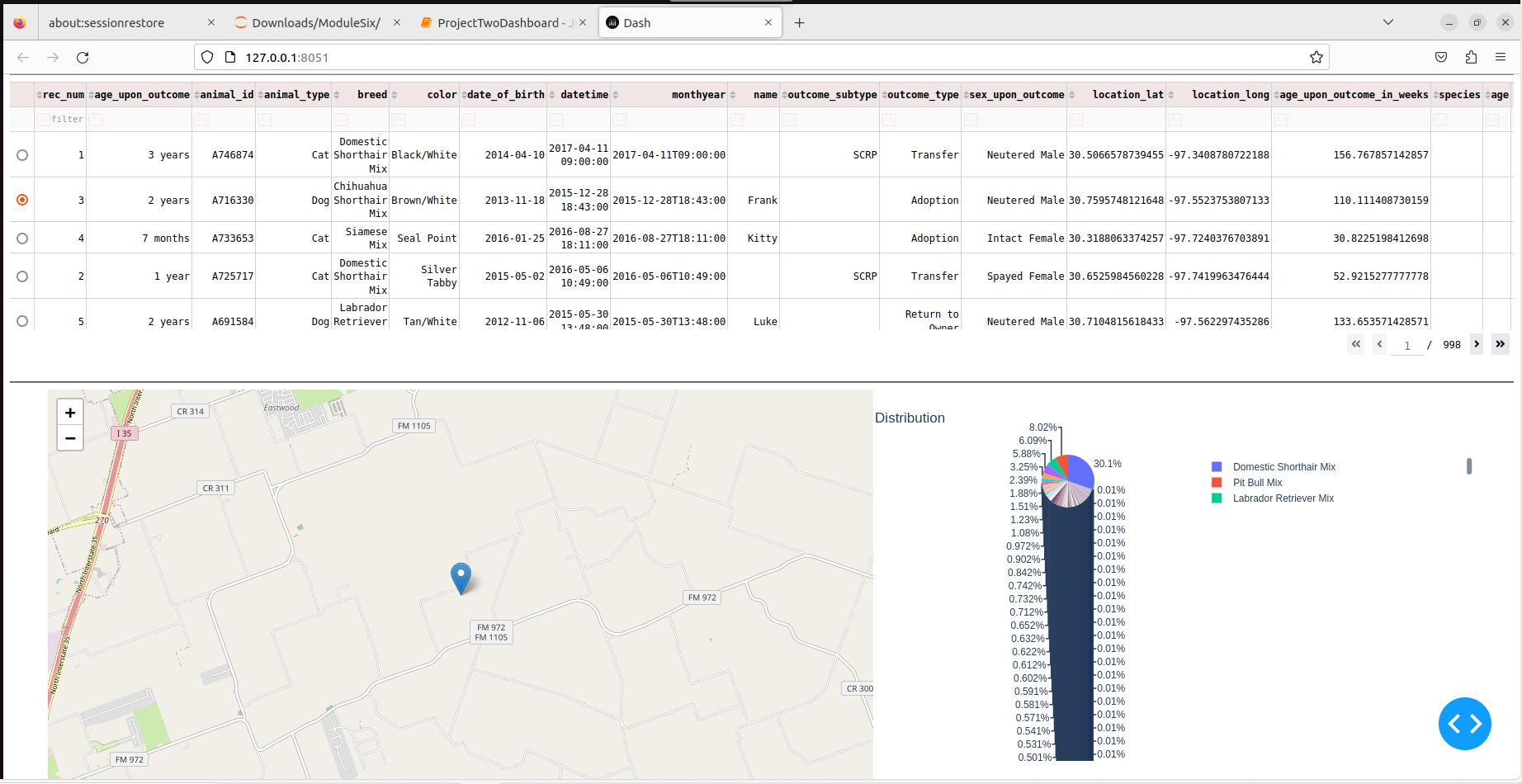


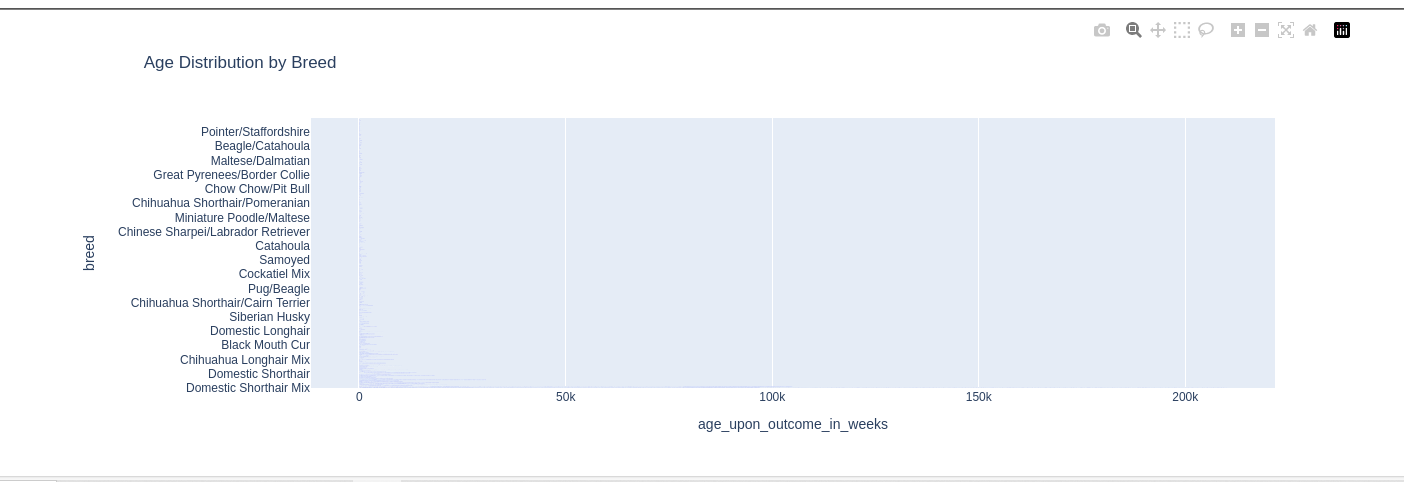
1. Disaster or Individual Tracking Filter: This shows just the disaster or tracking-related rescues.





1. Reset Button: This clears all filters and brings the dashboard back to its original state.





**Tools I Used**

For this project, I used tools that worked well together and made the process straightforward:

* Dash: Used to build the dashboard because it allowed me to combine interactivity with visuals in one place.
* Plotly: Perfect for creating the pie and bar charts, which are interactive and easy to use.
* Dash-Leaflet: Added the map feature, which was crucial for showing rescue locations.
* Pandas: Helped with organizing and filtering the data efficiently.
* MongoDB: Served as the database to store and retrieve rescue data.

**Why These Tools?**

* MongoDB: I used MongoDB because it’s great for handling large and unstructured datasets like the rescue information. It’s flexible and integrates well with Python, making it easy to query and filter the data.
* Dash: Dash was a good choice because it makes creating interactive web-based dashboards simple without requiring a lot of front-end work.

**How I Built It**

1. I started by connecting to the MongoDB database to retrieve the rescue data.
2. I created a layout for the dashboard, including the Grazioso Salvare logo, buttons for filtering, the data table, charts, and a map.
3. I added interactivity by linking the buttons to filters that update the table, charts, and map whenever a filter is applied.
4. I tested everything by running the dashboard multiple times to make sure the buttons, visuals, and map worked as expected.

**Challenges I Faced**

* Map Functionality: Setting up the map to display rescue locations properly was tricky. I fixed this by linking the table’s selected data to the map.
* Filter Performance: Initially, the dashboard was slow to update when filters were applied. I solved this by optimizing how the data was handled and filtered.

This project helped me better understand how to combine coding with real-world applications. I enjoyed building something interactive and useful while learning how to work with tools like Dash and MongoDB. It was also a great opportunity to practice solving challenges and improving my coding skills.

**Contact**

Janice Ainembabazi