Christopher O’Dell

SNHU – CS340

Project 2

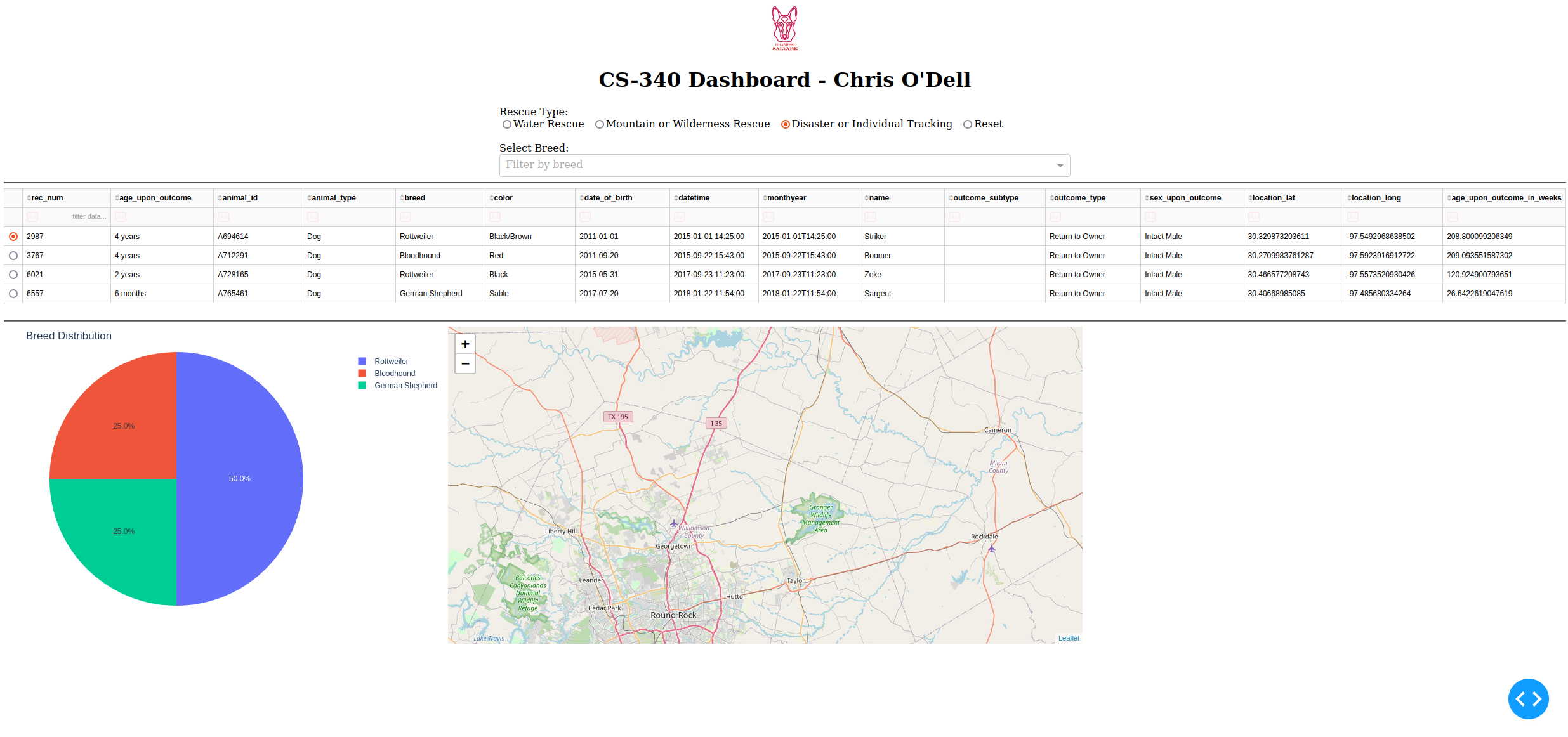
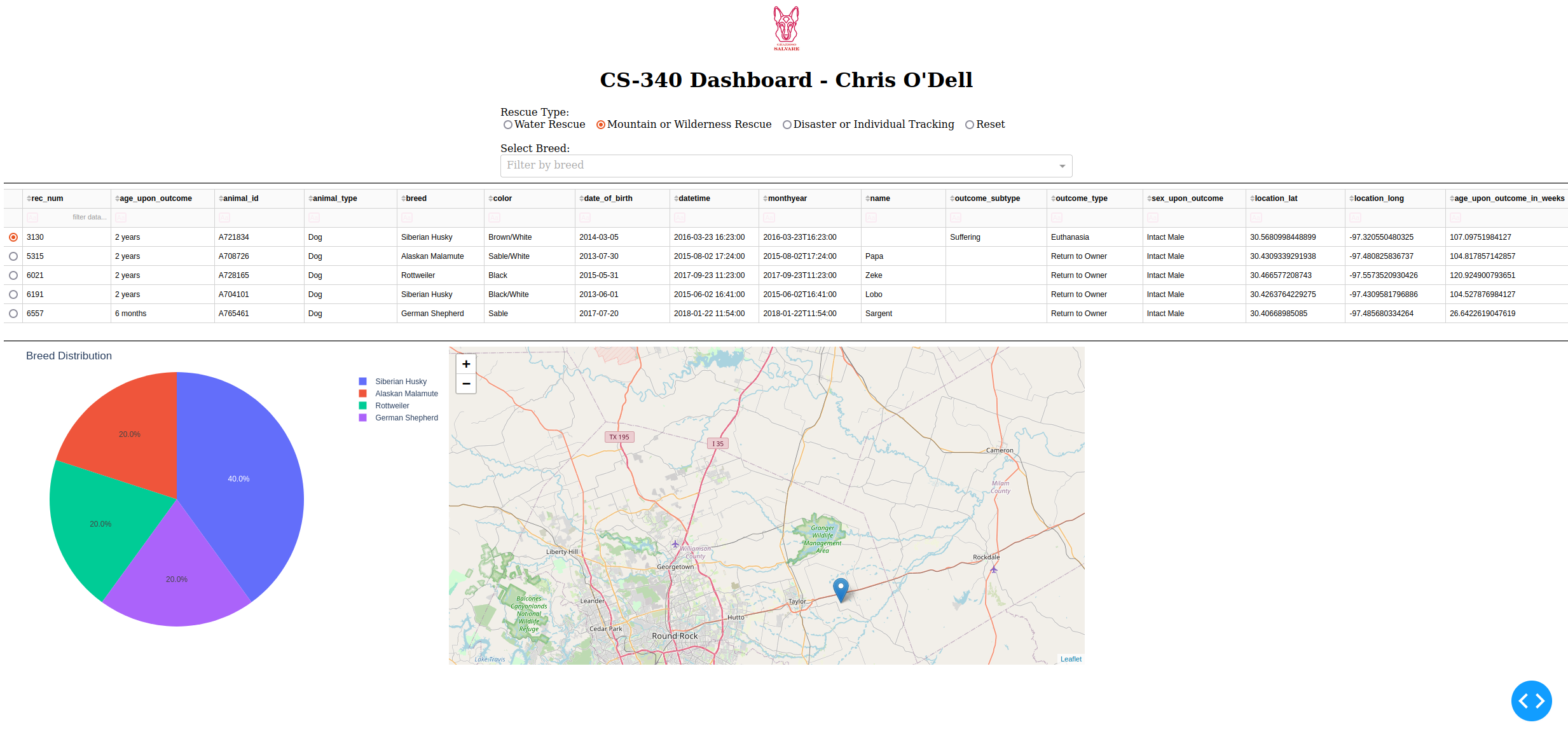
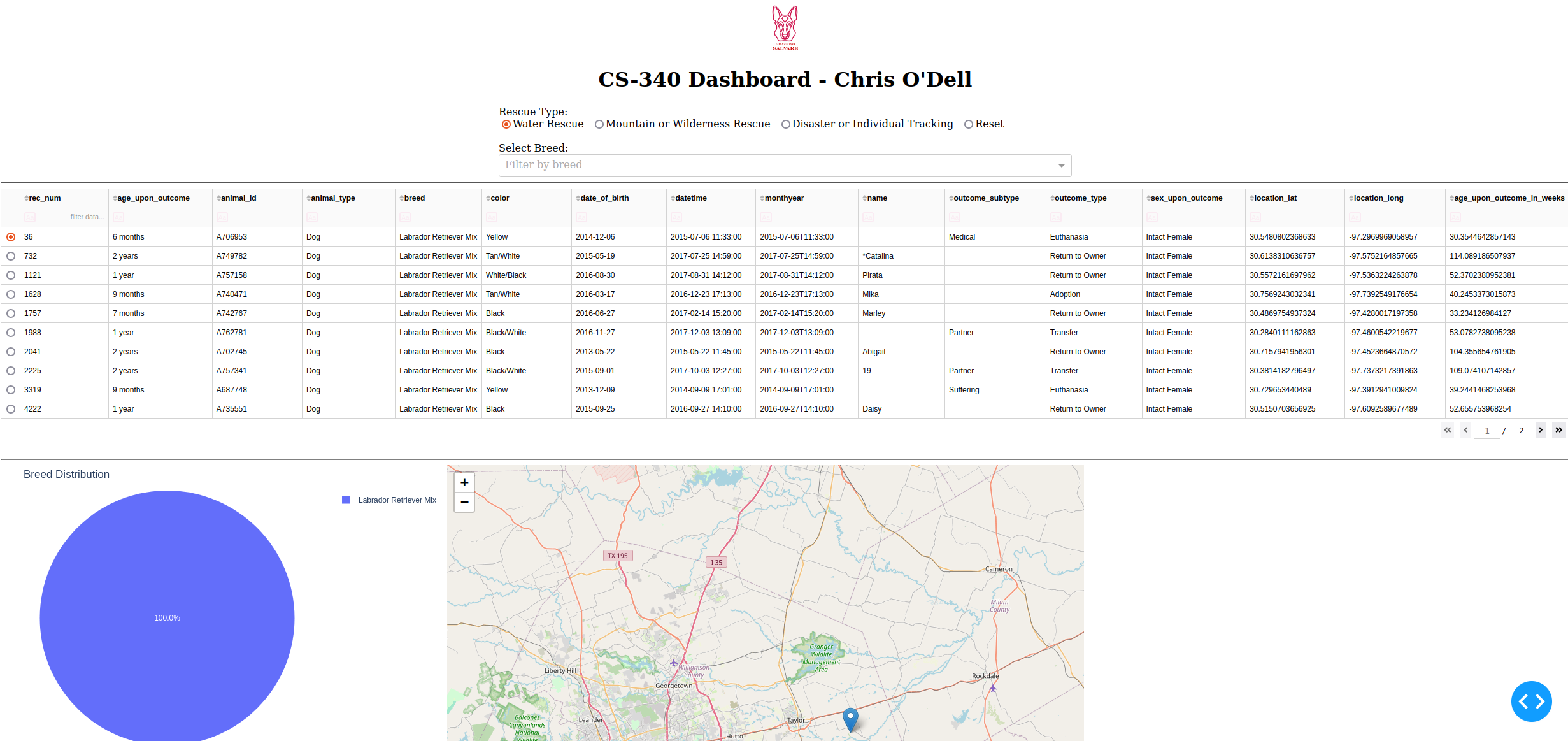
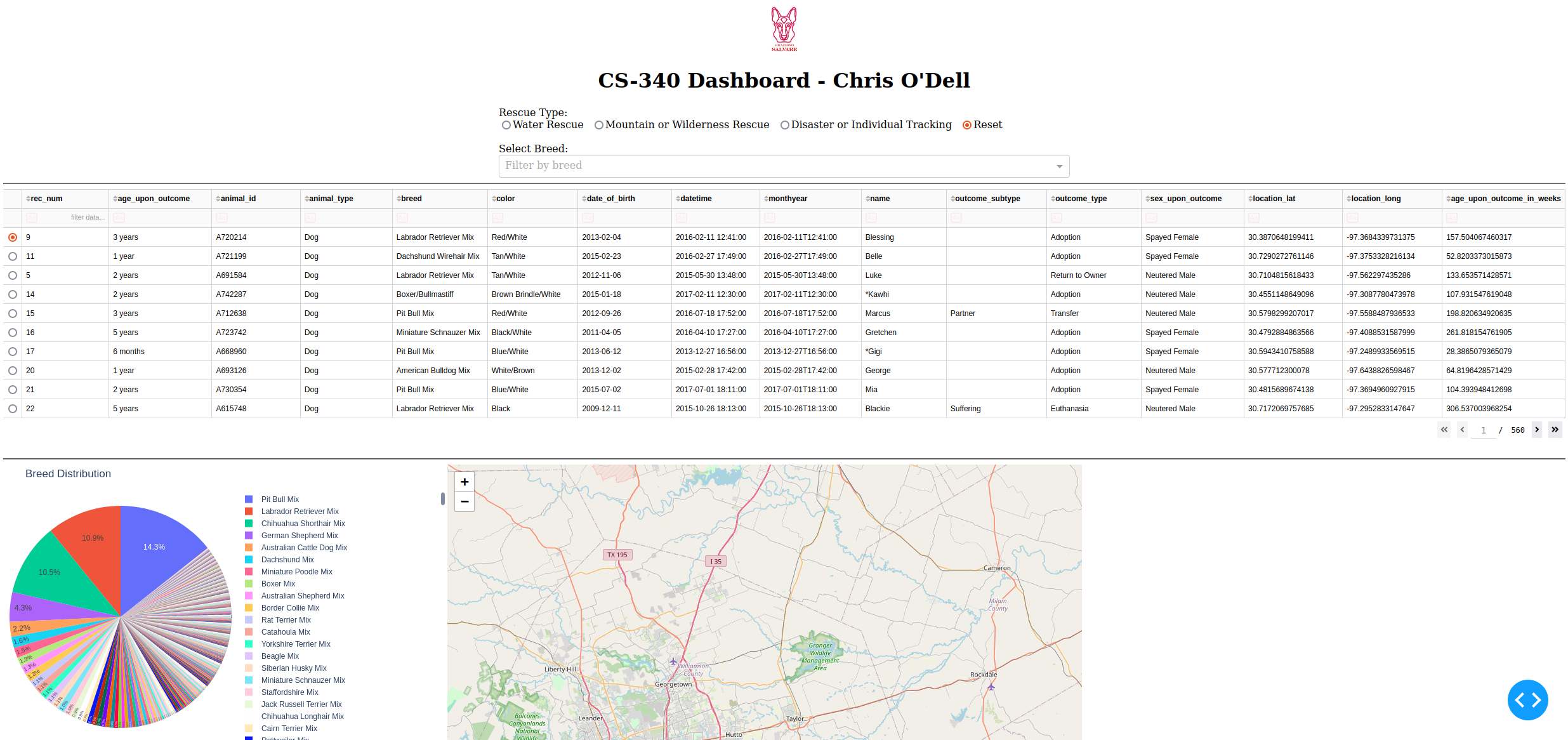
Dashboard ReadMe

* **Required Functionality**

The project provides an interactive dashboard for Grazioso Salvare, displaying visualization and filtering of animal shelter data from the Austin Animal Center. Users can filter animals based on rescue type via radio buttons or by dog breeds from the dropdown.

Required functionality also includes:

* Connection to the AAC database stored in MongoDB
* Interactive filtlering options (radio and dropdown)
* Live data table populated directly from the MongoDB
* Pie chart visualization showing distribution of breeds
* Map visualization showing location of animals
* Dynamic chart and map updates



* **Tools and Frameworks**

MongoDB was chosen as the model due to its flexible document-based storage. MongoDB provides efficient querying and filtering of large datasets and integrates easily with Python using Pymongo. The use of MongoDB also allows us to handle varying data formats without requiring a rigid scheme.

Python Dash framework provides the view and control for the application. Dash is perfect for building data dashboards for the web and integrates with both plotly for charts and leaflet for maps. Dash’s callback architecture allows for us to provide users with automatically triggered callback updates on the table, chart, and map.

Pandas is also used to allow us to handle transforming the AAC dataset into easily “digestible” visual representations. Plotly is utilized for building the pie chart visualization from our dataset.

* **Steps Taken**

1. Importing the AAC dataset into MongoDB including import verification and query testing.
2. CRUD Python Module to handle creating, reading, updating, and deleting data and allowing the dashboard to connect to the MongoDB.
3. Building the dashboard using the Dash framework.
4. Implementing filtering options with both radio buttons and dropdown menu.
5. Linking filter inputs to MongoDB queries through callbacks.
6. Adding an interactive data table for displaying data.
7. Implementing a pie chart using plotly.
8. Integrate a map using Dash Leaflet.
9. Continuous testing and updating through Jupyter notebook.

* **Challenges During the Project**

Implementing the filtering was by far the biggest struggle in this project. After doing some research into implementing the filter options and having its desired functionality working, it was evident that proper callback and query updates were necessary to provide updated, filtered results to both the data table and the pie chart.