# CS 340 README Template

## Required Functionality

This project required a web interface that allows users to easily filter a database to find certain types of animals. It also required a geolocation chart and another data visualization chart (a pie chart in this case) that automatically update based on the filtering.

## Tools

MongoDB was used as the backing database system, with PyMongo being the glue between the database system and the web interface. This combination allowed for a quick development time.

Dash was used for building the HTML web interface, as it provided built in functions to create a clean data table and connect it to the data visualizations. It is also written in Python, which made it easy to connect to PyMongo.

Dash DataTable: <https://dash.plotly.com/datatable>

Dash Graph: <https://dash.plotly.com/dash-core-components/graph>

Dash RadioItems: <https://dash.plotly.com/dash-core-components/radioitems>

Dash Callbacks: <https://dash.plotly.com/basic-callbacks>

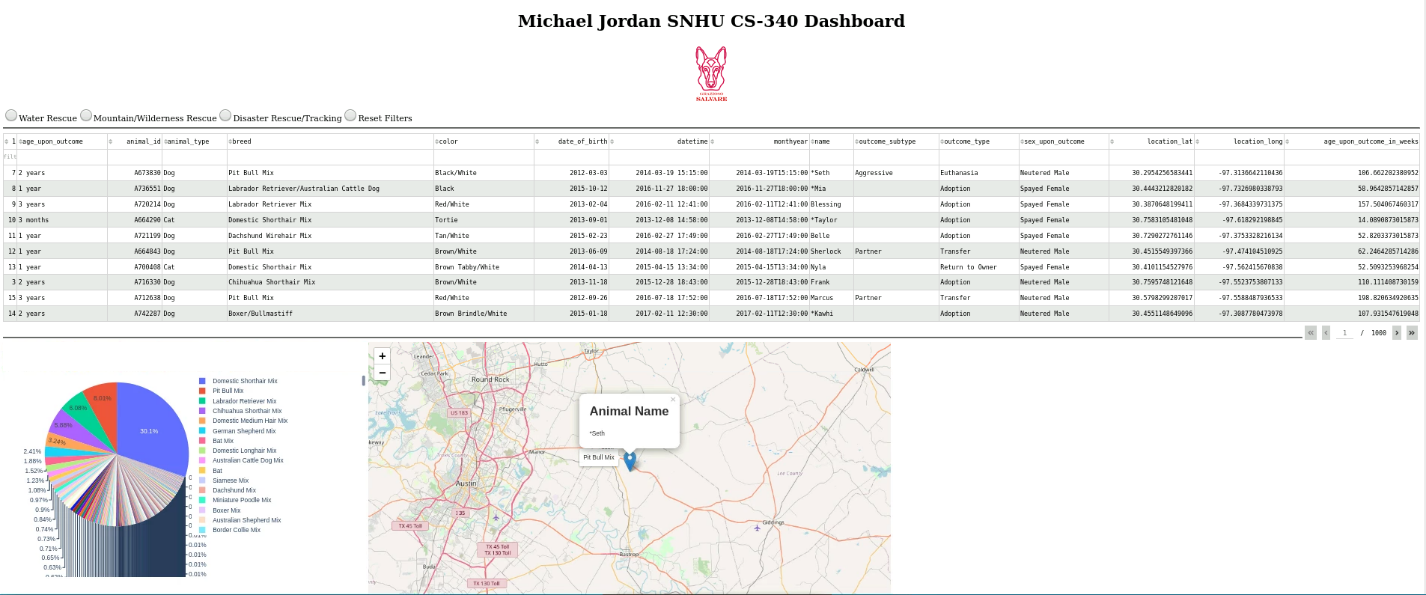
Plotly Pie Chart: <https://plotly.com/python/pie-charts/>

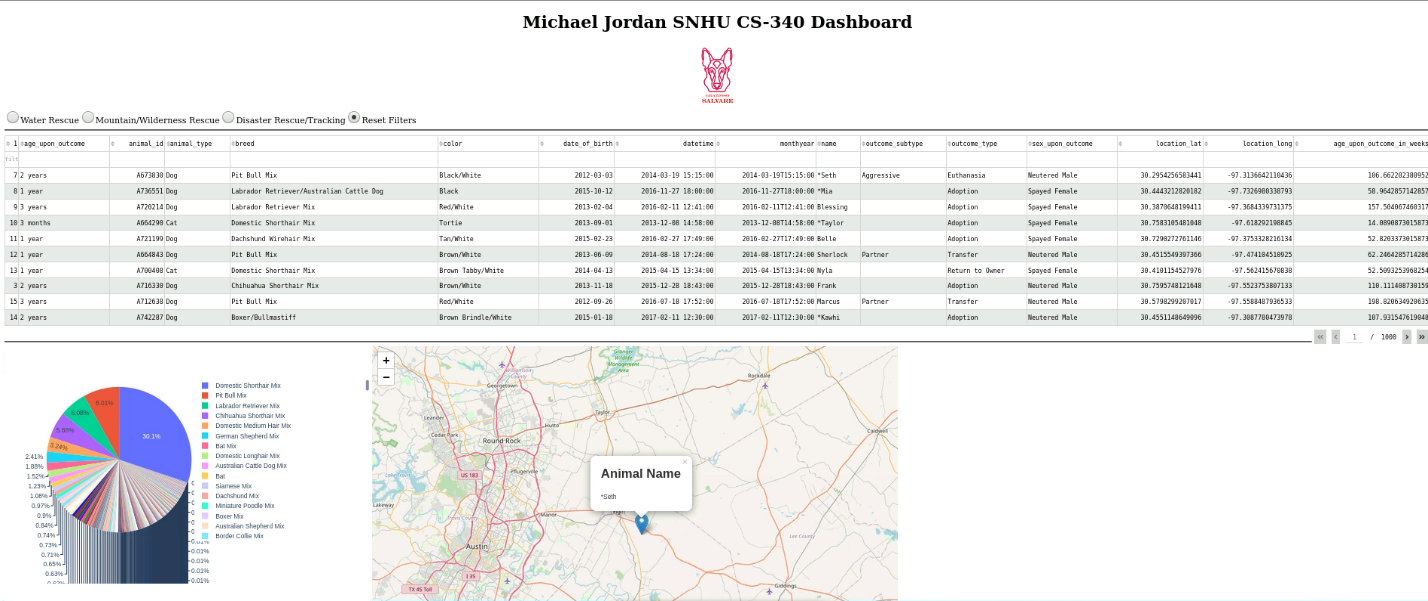
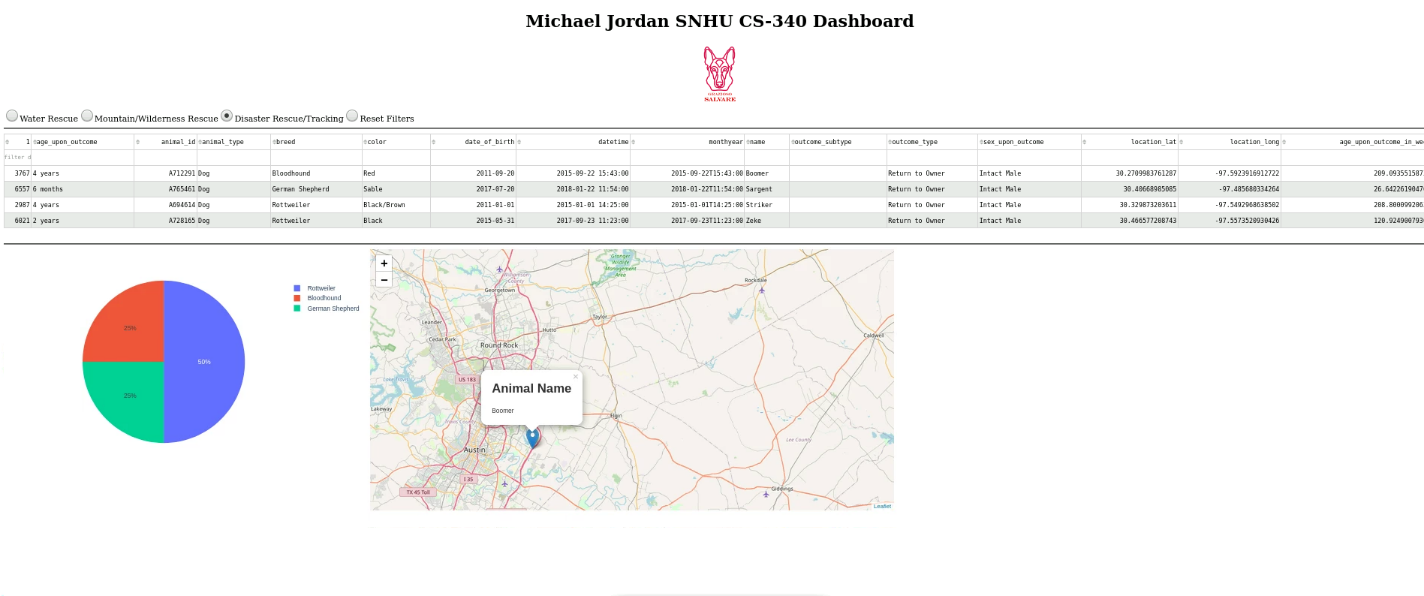
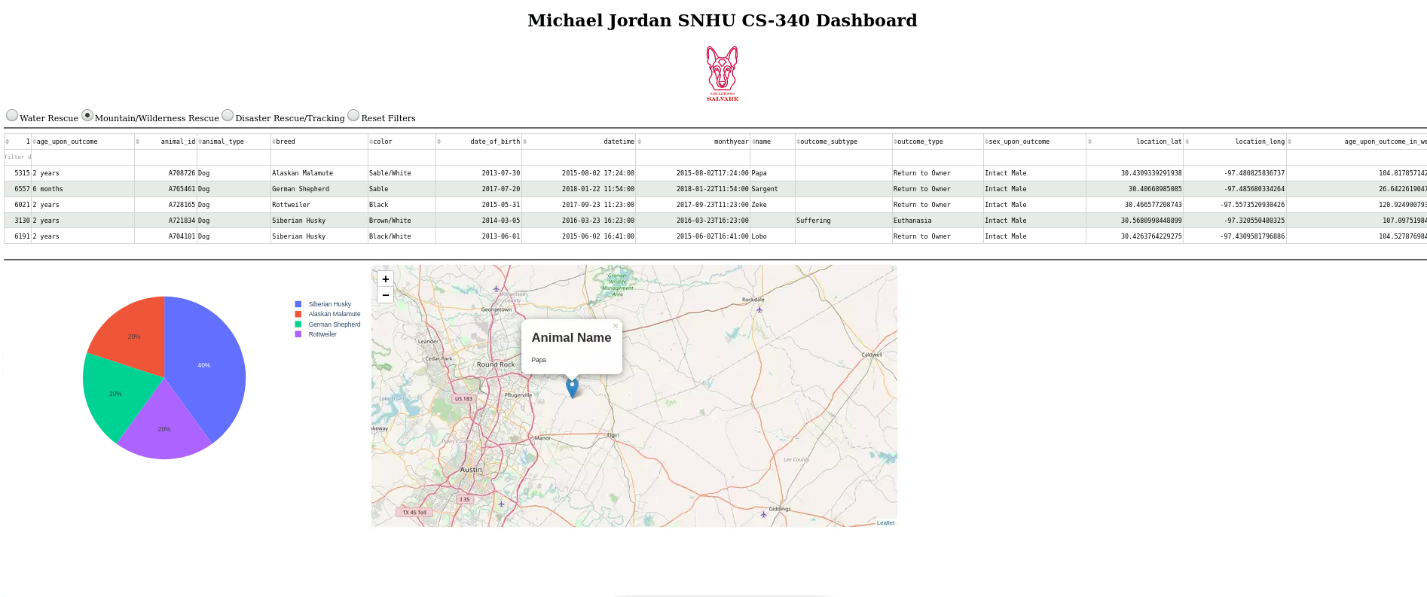
Dash Leaflet: <https://dash-leaflet.herokuapp.com/>

## Completing the Project

The first step was getting the data table to work properly, since it is the component that everything else relies on. The Dash framework makes connecting components simple with callbacks, so the next task was to build the radio buttons. Next was creating the geolocation chart. A challenge here was getting the marker to properly update based on the filtered data table. I had to investigate the indexing properties of the underlying pandas data frame to overcome this challenge. The final item was the pie chart. Initially, attempting to add this item to the interface was not working at all, and would also cause the geolocation chart to not display. I believe the underlying issue that caused this behavior was related to the Dash callback being applied at the wrong time, leading to an attempt to iterate over data that did not exist. This was fixed by checking if the data existed before attempting to iterate over it.

### Screenshots

Chart

Description automatically generated with medium confidence

## Contact

Your name: Michael Jordan