CS340 7-2 Project 2 README

Michael Ross

Department of Computer Science, Southern New Hampshire University

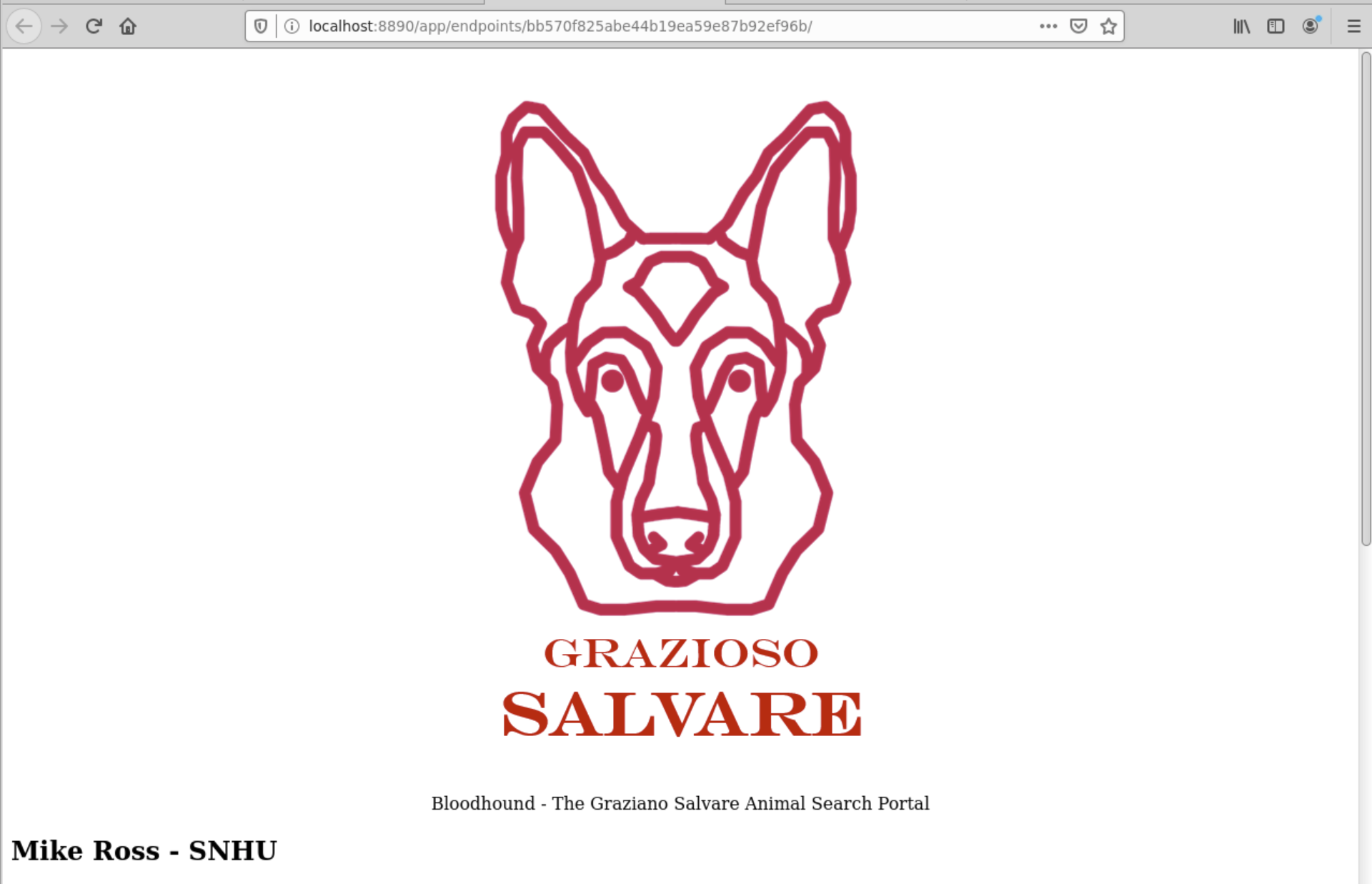
CS340: Client Server Development

Prof Kellogg

April 16th, 2023

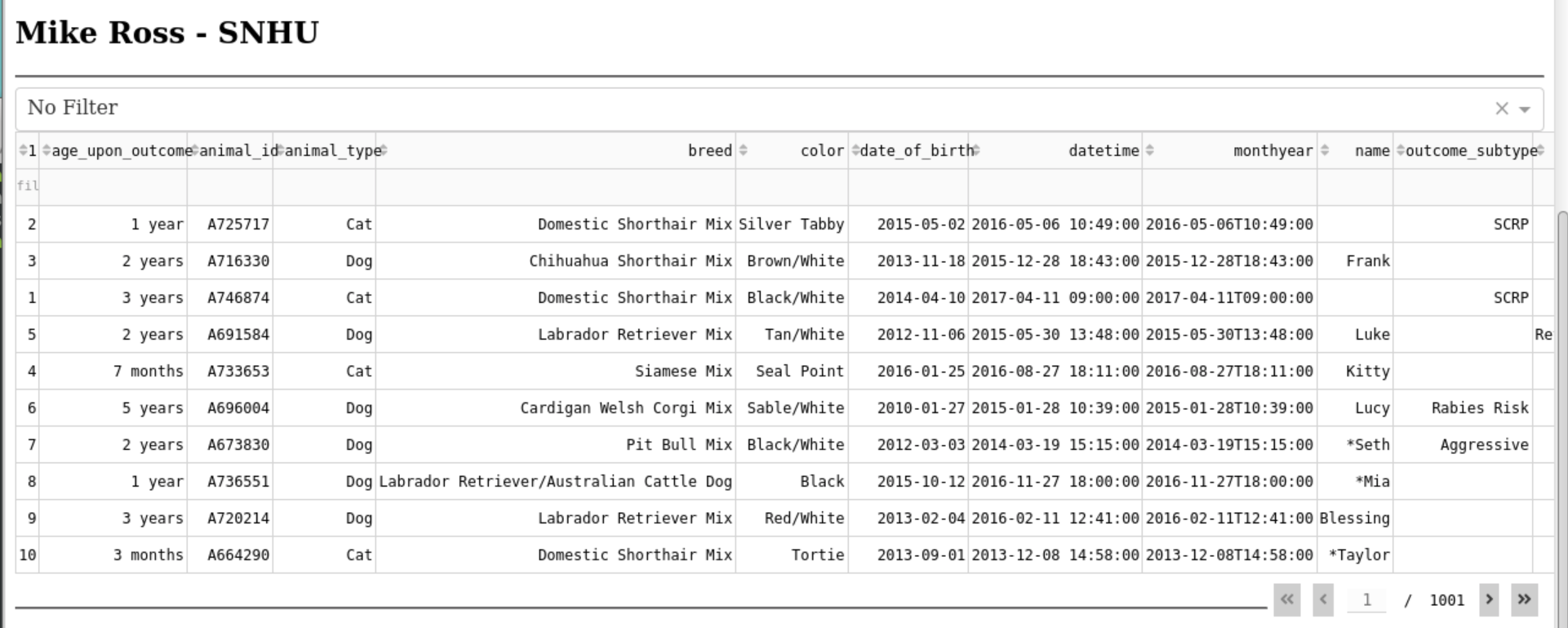
# README

The BLOODHOUND Animal Search Service by Grazioso Salvare

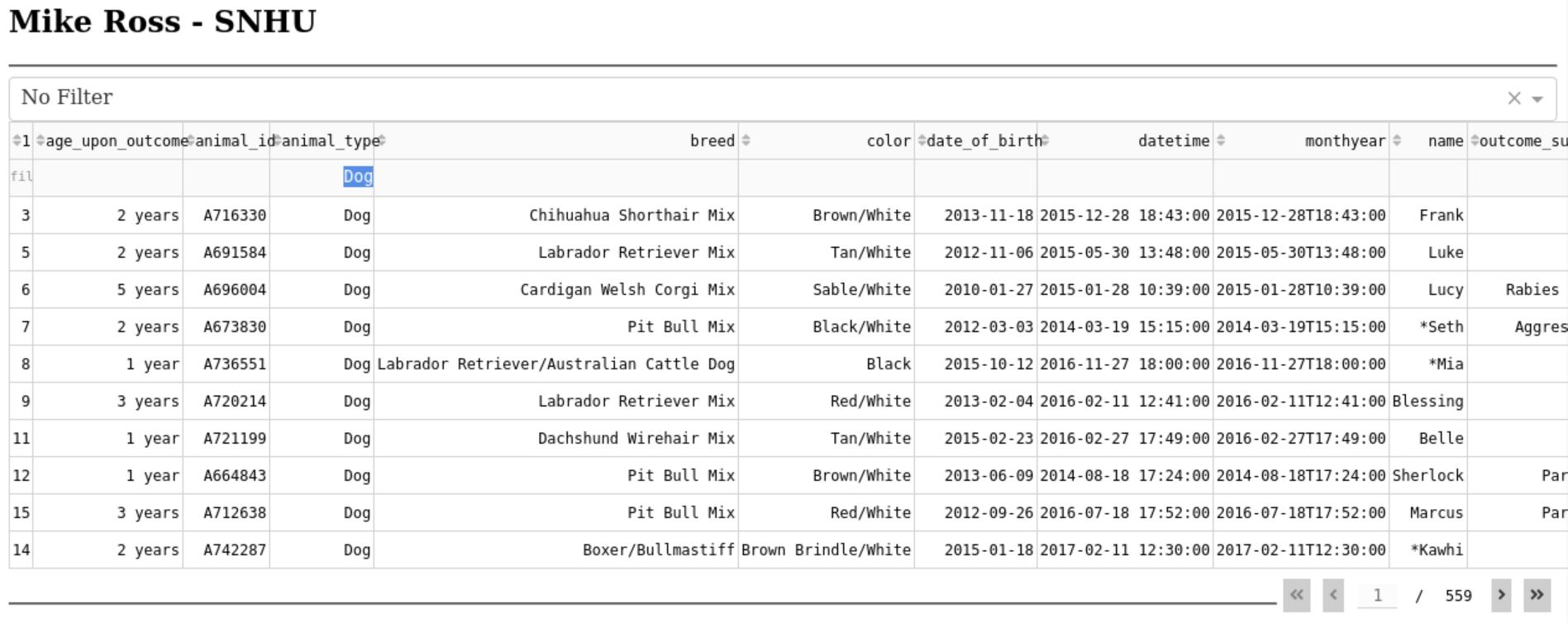


The BLOODHOUND service is a portal for use by authorized GS personnel for searching and identifying animals for use in various search and rescue organizations. The portal can:

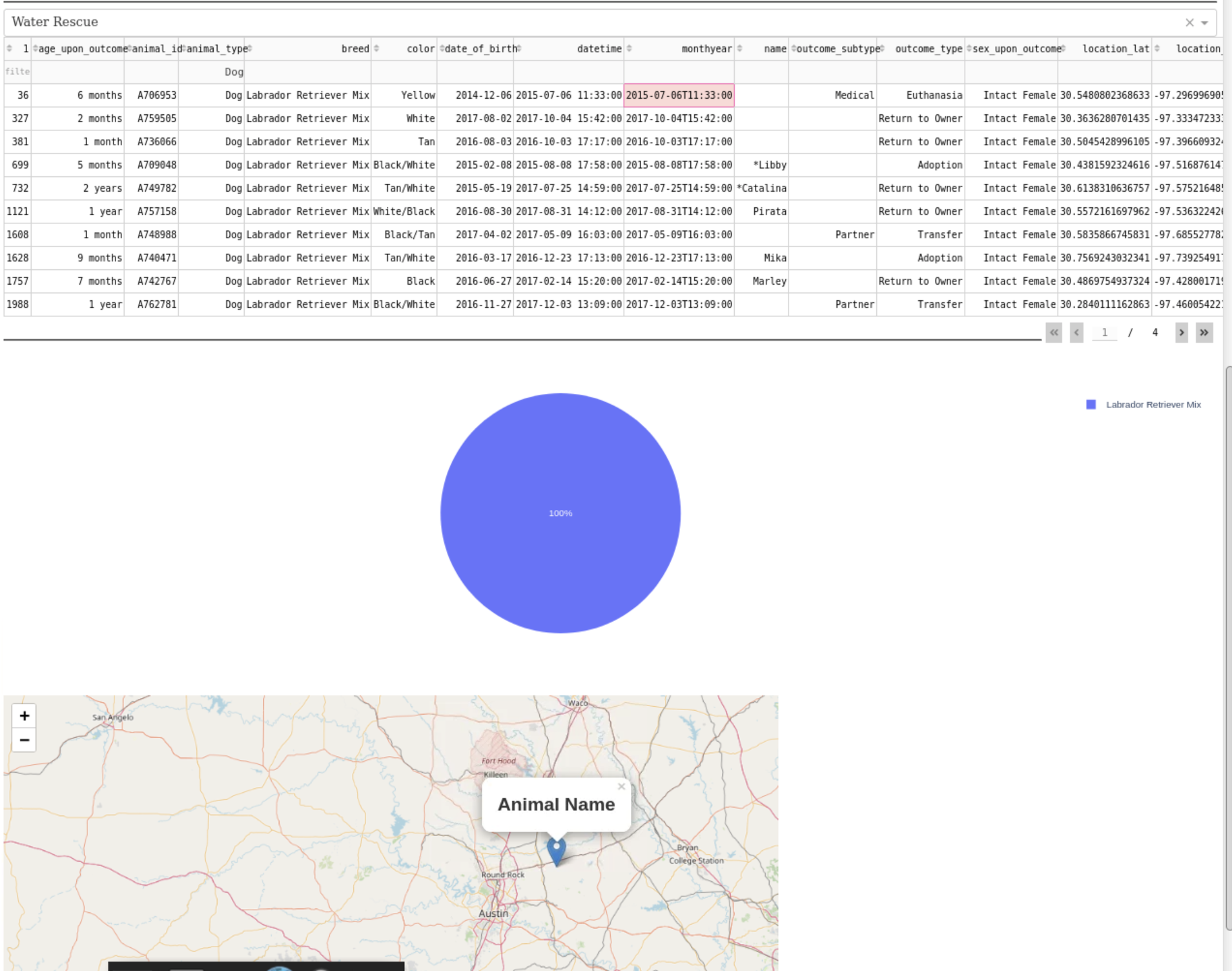
* List the data from various shelters in the area



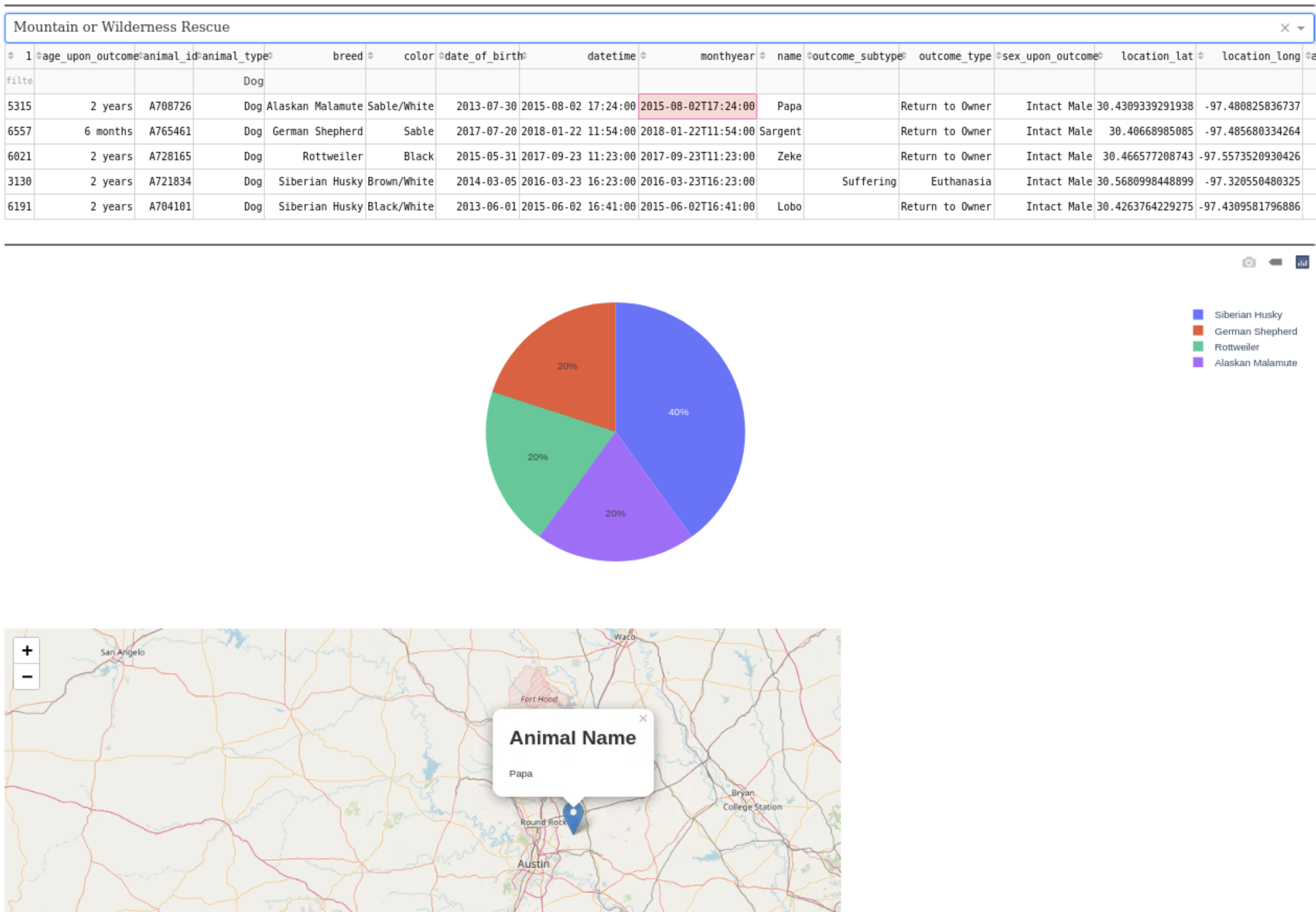
* Filter and search the data



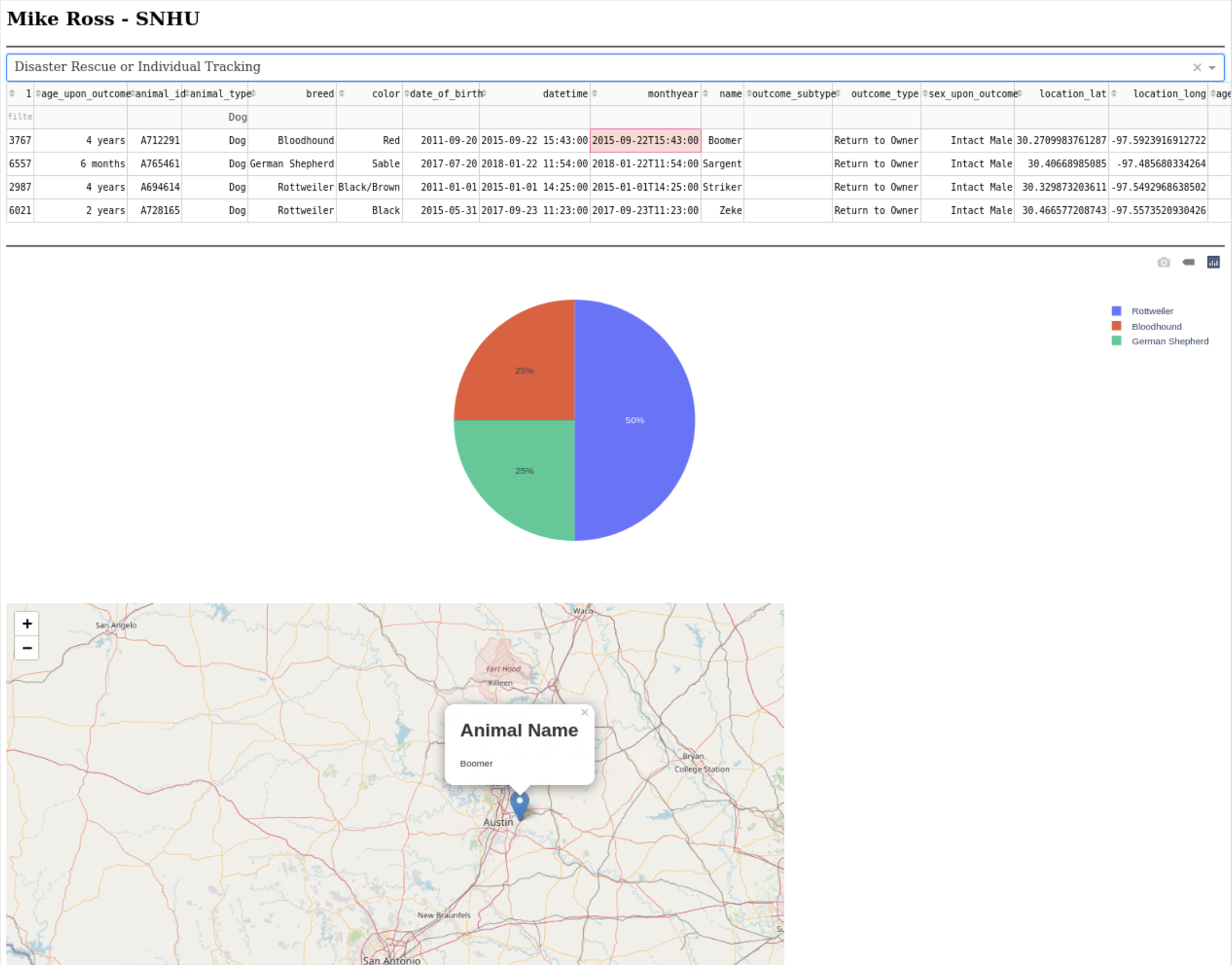
* Quick-search for several commonly used use-cases:
  + Water Rescue



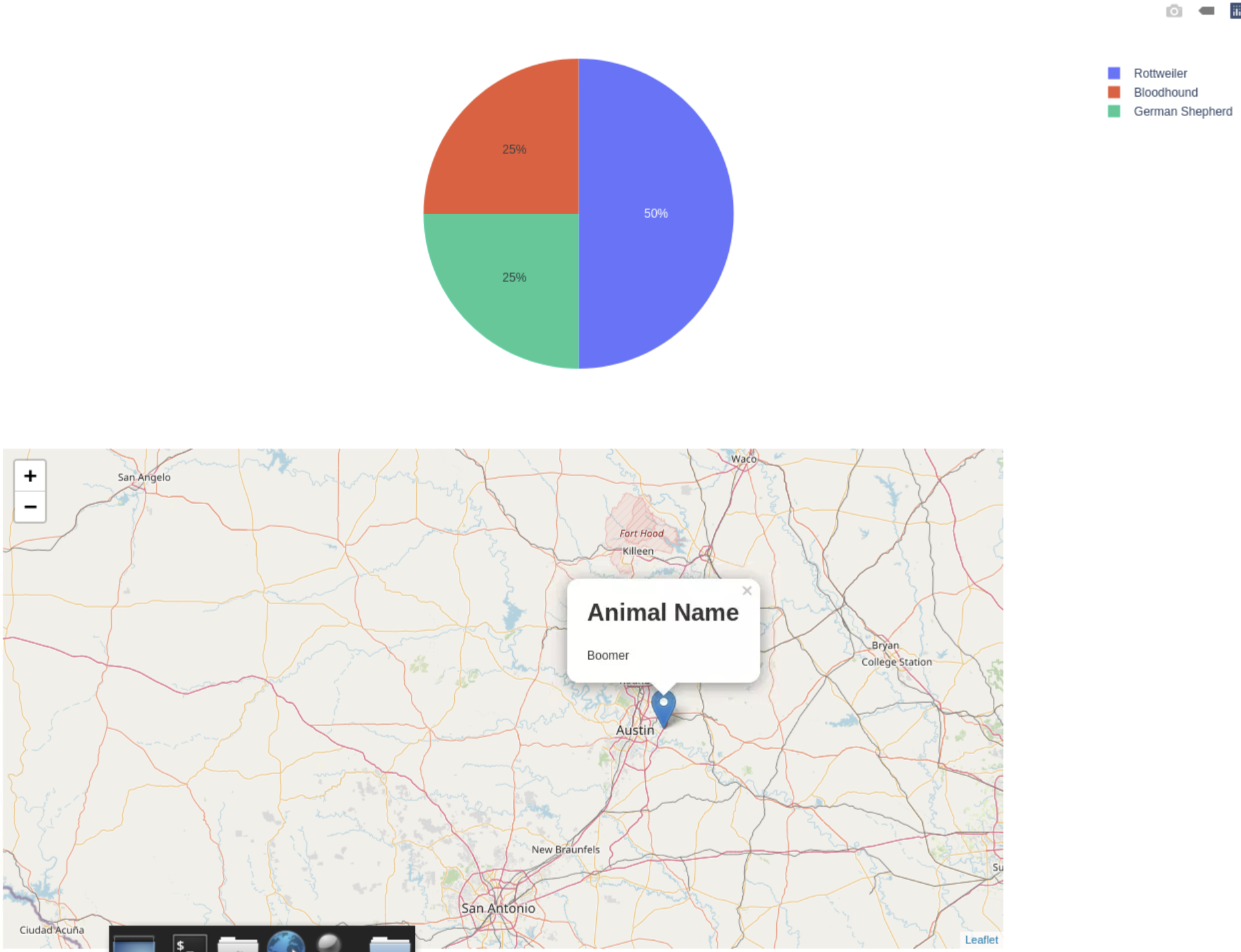
* + Mountain or Wilderness Rescue



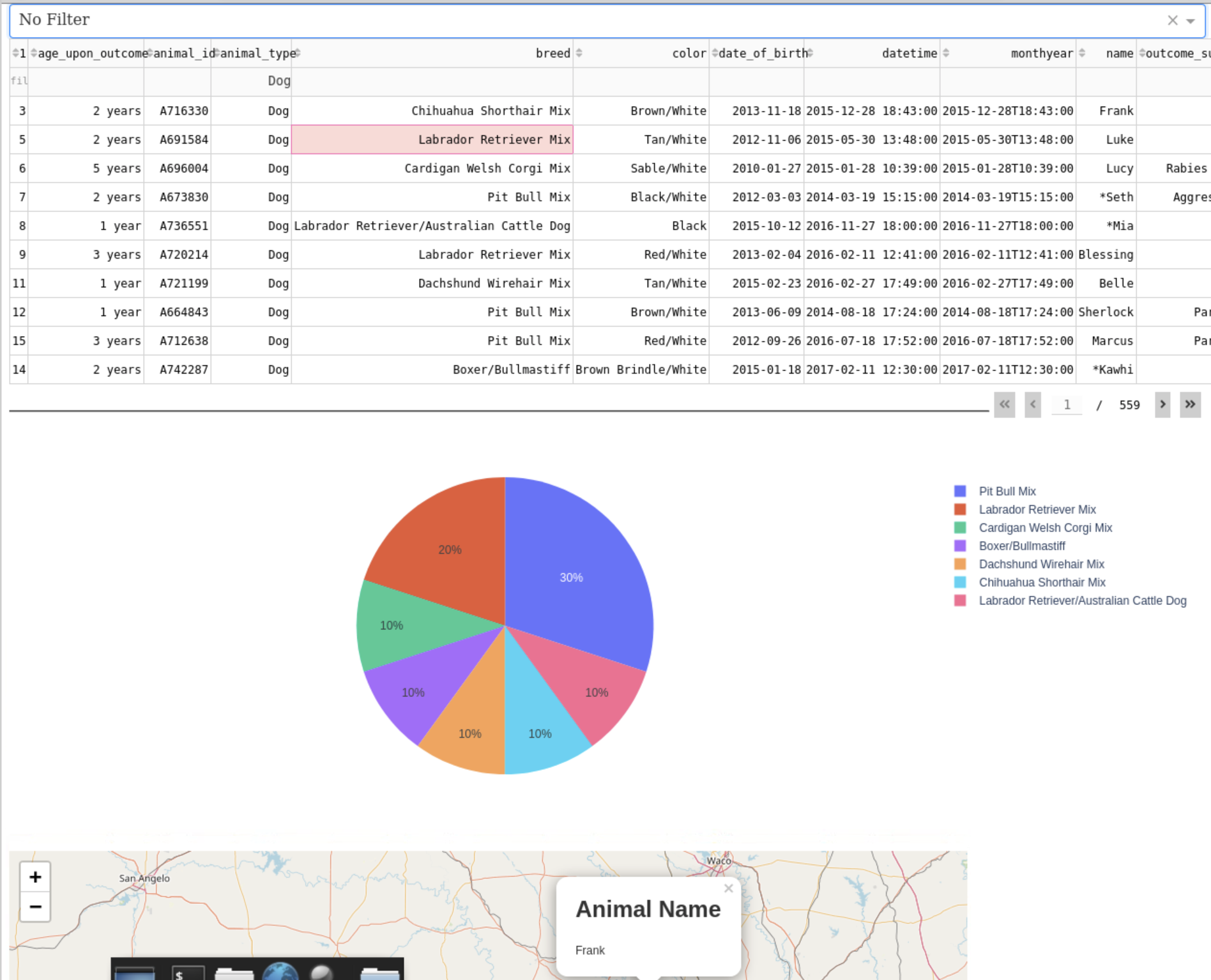
* + Disaster or Individual Tracking



* Displays the location of the animals searched as well as the histogram of the various breeds in the current data set.



* Allows the filters to be reset easily for normal use.



## Tools Used

The BLOODHOUND service was created using the MVC or model, view, controller framework. Using this framework ensures that the service can remain modular and flexible as new functionality is added to the application.

MongoDB was used for the Model portion of the application. Using MongoDB in particular and NoSQL databases in general ensures that the schema for the application can be modified as new fields need to be added to the database. It also ensures that the application remains performant as MongoDB provides a high level of performance and reliability. MongoDB is also supported in Python through use of the PyMongo library.

For the View portion of the application, to provide the user interface, Dash was utilized. Dash is a library for Python which makes creating data interfaces easy, reliable and powerful. Charts, images, data tables and more can be created through Dash. It’s native integration with Python makes it a good choice for this application.

Finally, Python was used as the middleware or glue layer for the application. Given its heavy use in data analytics workloads and native libraries for large data sets, UI integration, and middleware uses it was a strong choice. The middleware layer consists mostly of a class which interfaces with the database to perform CRUD type operations.

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

Python: <https://www.python.org/>

MongoDB: <https://www.mongodb.com/>

## Steps Taken

The application was developed in phases.

* In phase 1, the MongoDB database was configured and authentication was enabled.
* In phase 2, the middleware layer to interface with the database was created and tested for all four of the CRUD operations.
* In phase 3, the front end was developed and utilized the class developed in phase 2.

## Challenges Faced

There were many challenges faced while developing the application. Most were minor but there were a lot of challenges and some very severe challenges in working with Pandas, dataframes and Jupyter.

Jupyter provides a very limited troubleshooting interface and it made it difficult to properly diagnose and fix errors. Finding out about starting a new Jupyter notebook so the logs could be examined was a big breakthrough and helped development to continue much quicker.

Pandas and dataframes in particular proved very challenging as well. The syntax for most commands was foreign and the types of queries and manipulations had to be learned through a lot of reading and trial and error.