# CS 340 README Template

*Use this template to complete your README file. When completing the template, keep the headings as they are so that your document has a clear organization. Remove the italicized prompt text after you have completed each section for a polished final document.*

## About the Project/Project Title

This project is a dashboard web application which was built for the Grazioso Salvare company. The application is designed to utilize the Austin Animal Shelter (AAC) database api in order to display animal information. Grazioso Salvare is a company that identifies animals that are good candidates to be trained as rescue dogs. This web application can filter the AAC data for dogs that meet the criteria for being rescue animals. The web application also uses a table, pie chart and a geolocation graph to help expedite searches for rescue dogs.

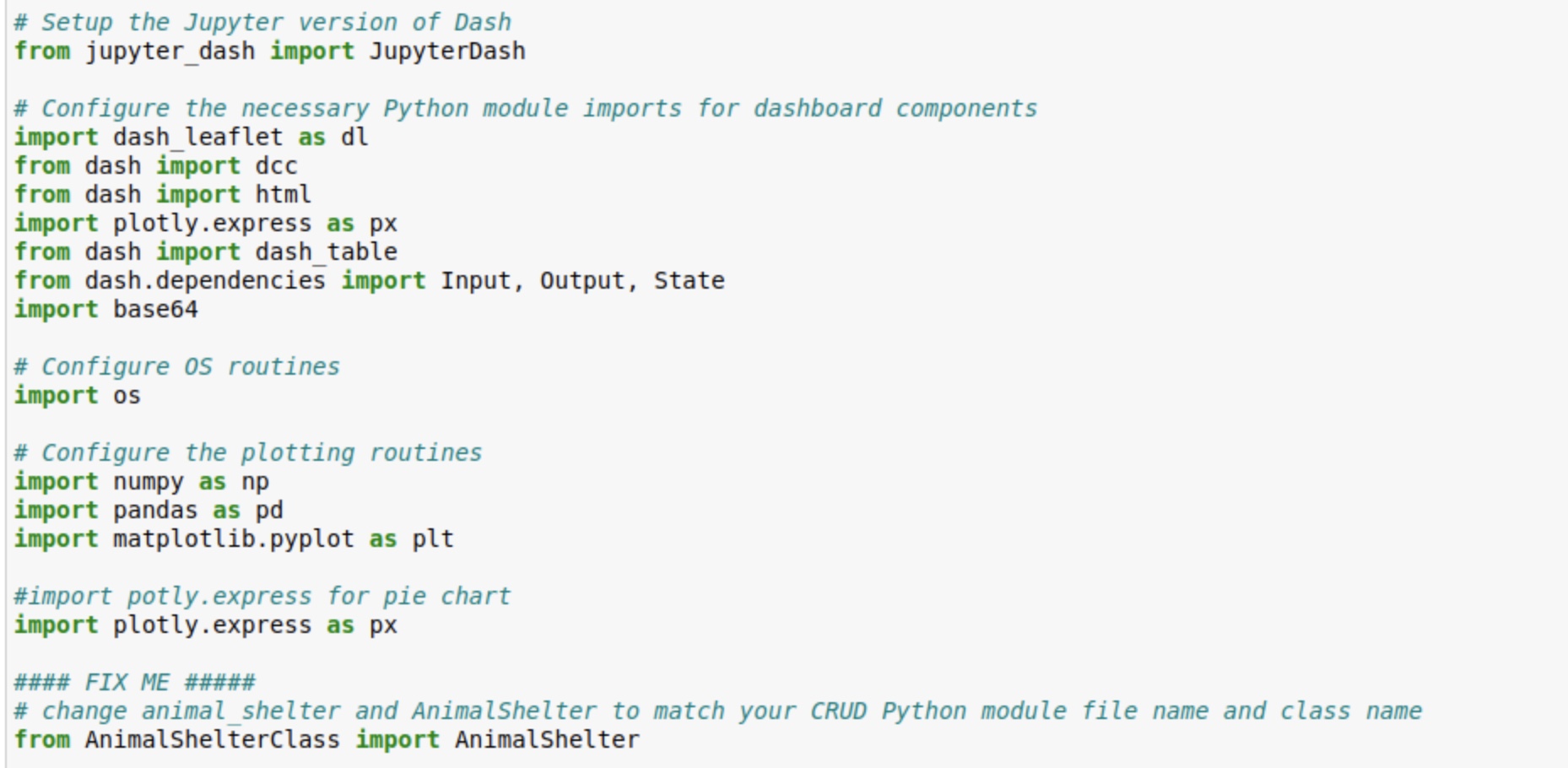
## Motivation

The motivation to build this app is to give Grazioso Salvare a tool to help quickly parse through AAC data to find rescue dog candidates. This will help Grazioso Salvare quickly and efficiently find dogs that are good rescue candidates in the Austin area. This will also help increase the positive outcomes of animals. This web application can also be utilized as Grazioso Salvare grows and expands and data from other geographical areas can be added increasing the range that Grazioso Salvare can get rescue dogs. Grazioso Salvare has made the source code open source so people can use and modify the code to meet their needs. For example if a company wishes to build an app that helps identify animals that are good fit as service animals they don’t have to start from scratch. They can modify the search parameters to meet their needs.

## Getting Started

To get started with creating your own you will need to download the AnimalShelterClass module to use the AAC database api. You will import this at the beginning of your app.py file.

## Installation

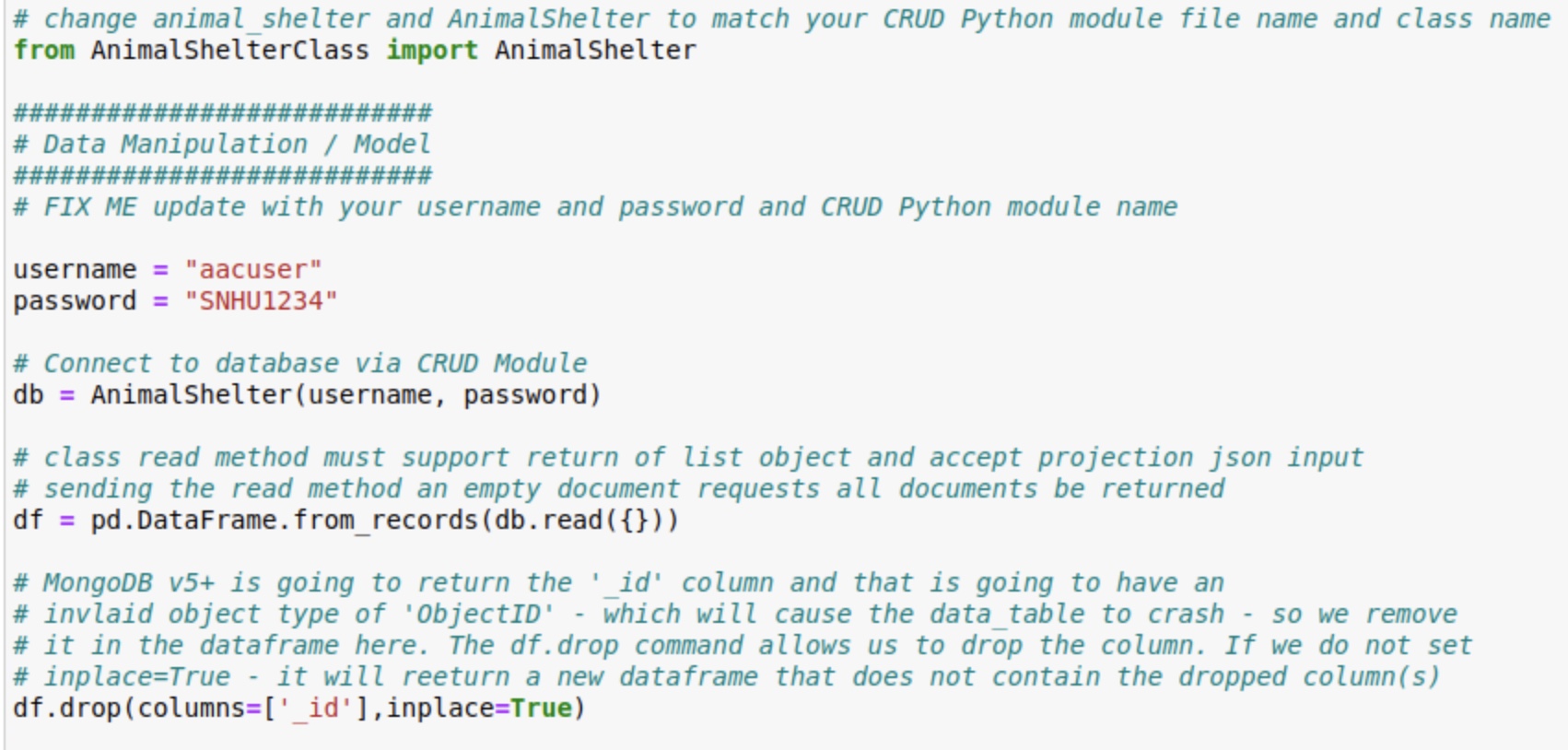
In addition to the AnimalShelterClass module this application also uses dash, plotly.express, numpy, and pandas. A screen shot of all of the dependencies used is below.

## Usage

This web application uses the AnimalShelterClass’s read() method to query the AAC database. For each of the radio buttons a read request is called passing in the specific parameters for that type of rescue dog. Once the the data for that type of rescue dog is received it is used to populate the table, pie chart and geolocation graph.

### Code Example

The code example below shows an instance of the AnimalShelterClass being created passing in the database username and password as arguments.

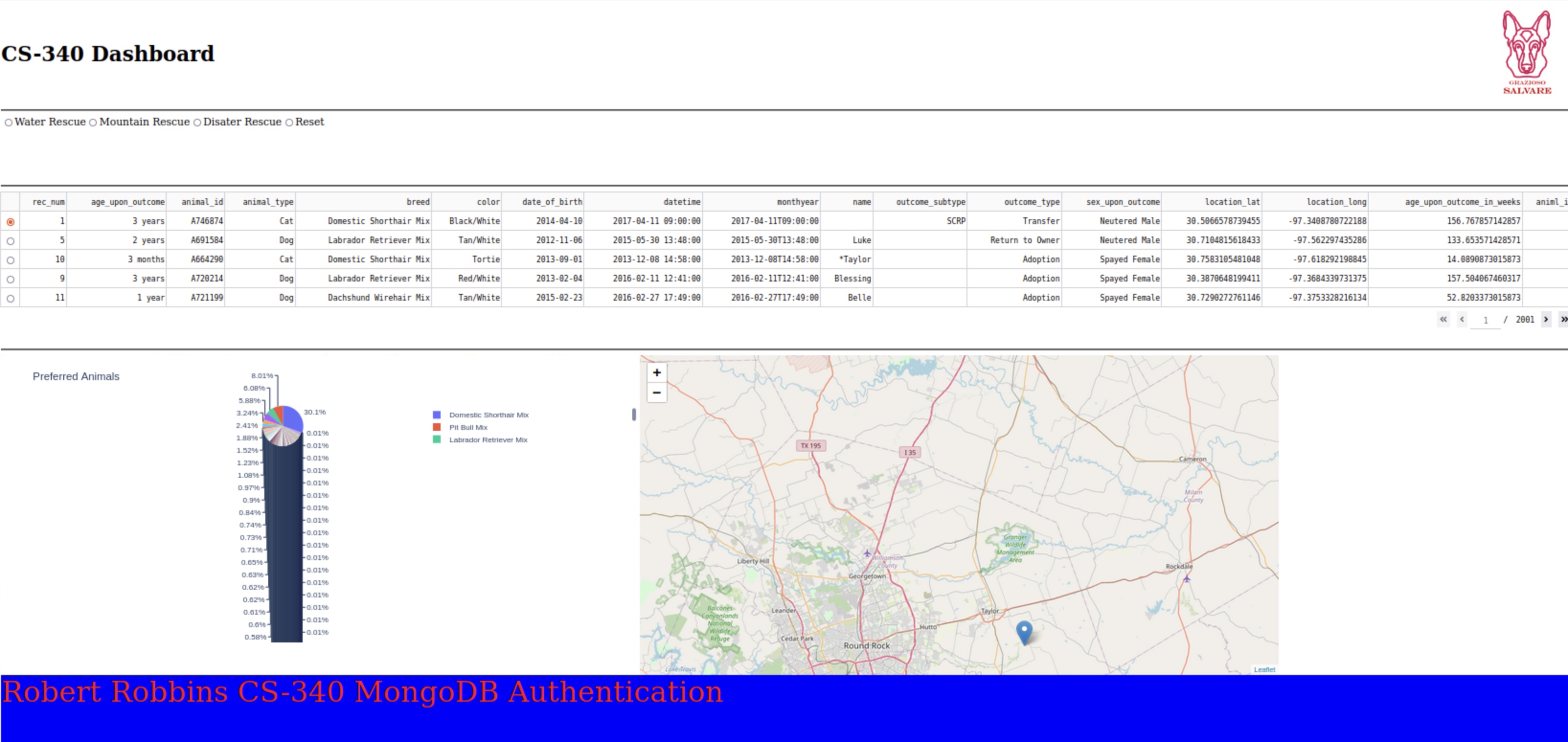
The instance of the class is then used to create a pandas data frame using the AnimalShelterClass read method. The read method has an empty dictionary as its argument which will get all documents from the database. This is the default data for the web application, and will be what is shown when the reset radio button is selected.

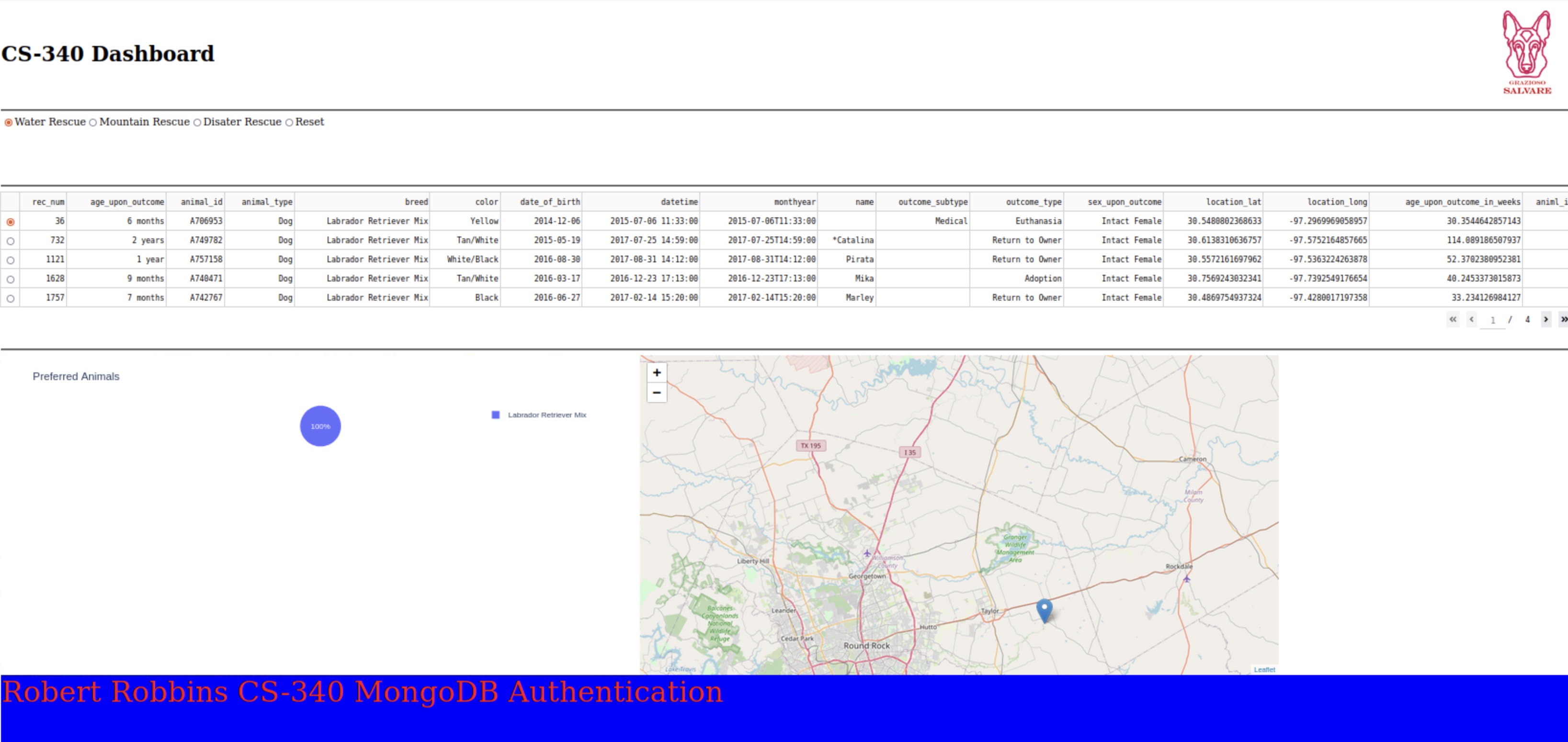
The next screen shot shows the radio button callback method. This method is used to filter the data based on the button the user selects. For this use case there are four rescue dog types. When the corresponding radio button is selected the update\_dashboard callback function is called passing in the filter type. A read request is then called passing in the search query for that type of rescue dog. This method can be modified for different types of user interfaces and search queries. To change the filter to show all cats you can change the dictionary to be {‘aniaml\_breed’: ‘Cat’}.

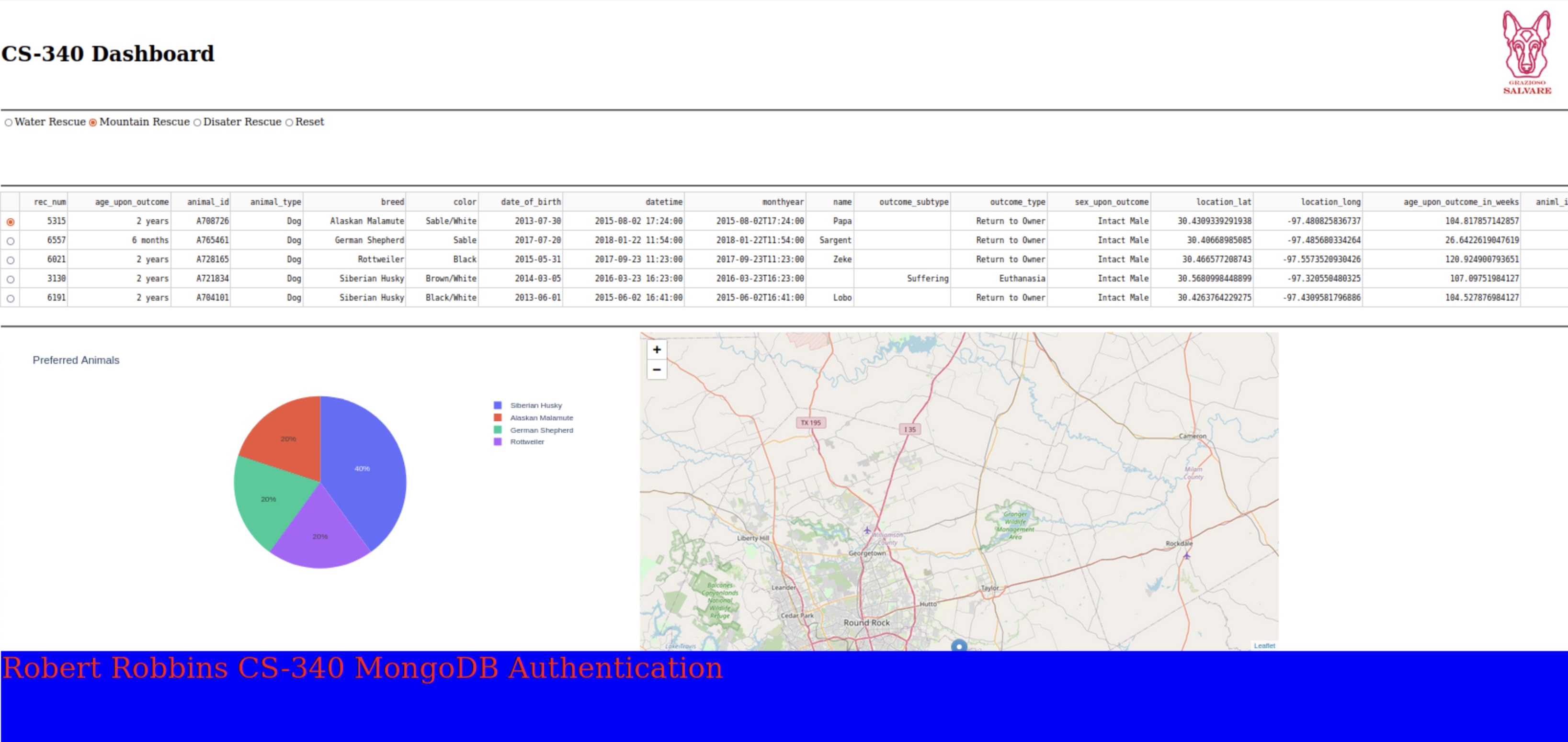
### Tests

This web application has four different states based on the filter buttons. The four data states are water, disaster, mountain and default/reset. To test that each of this works I selected the button to make sure that the right data populated the graphs.

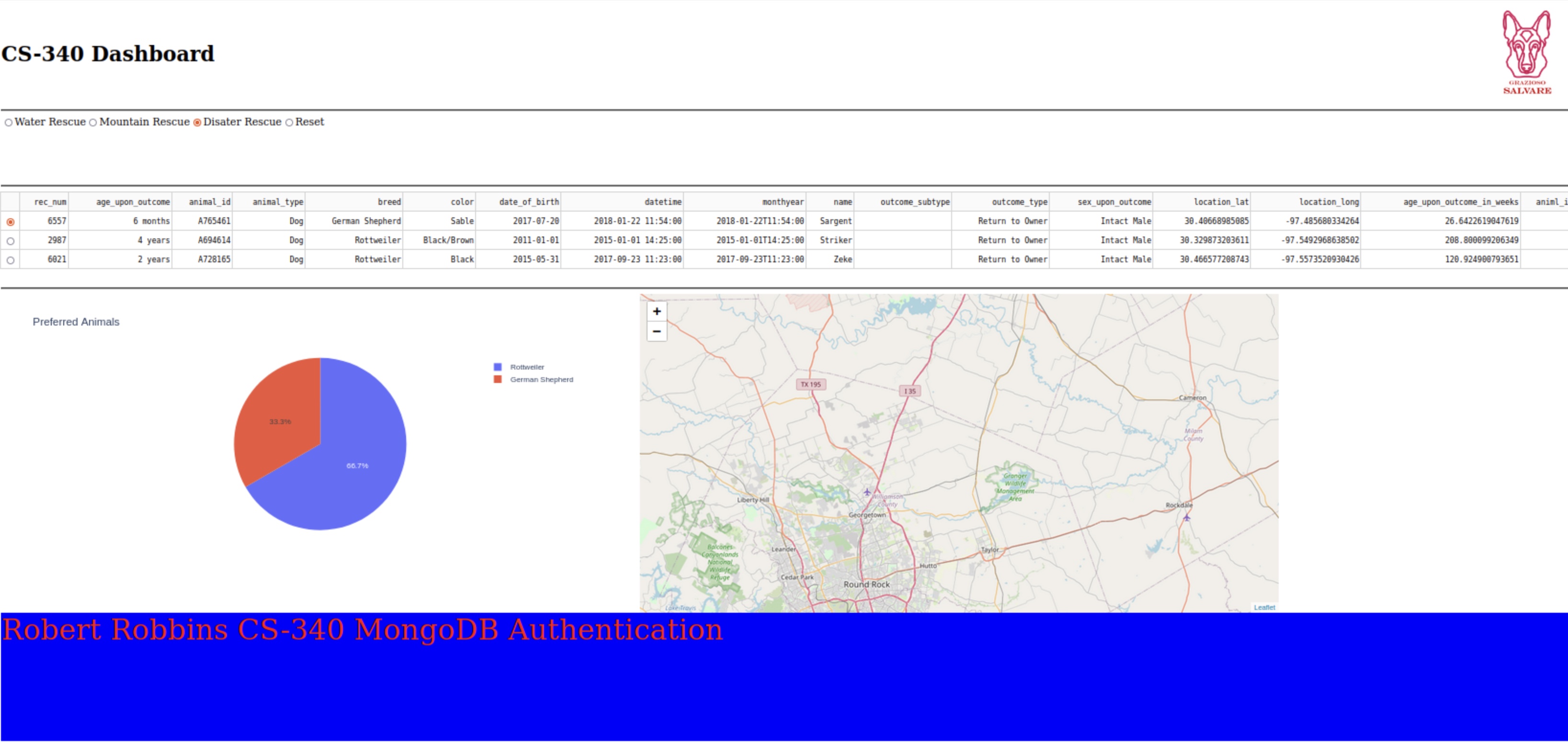
Below is the default state showing all animals in the database.



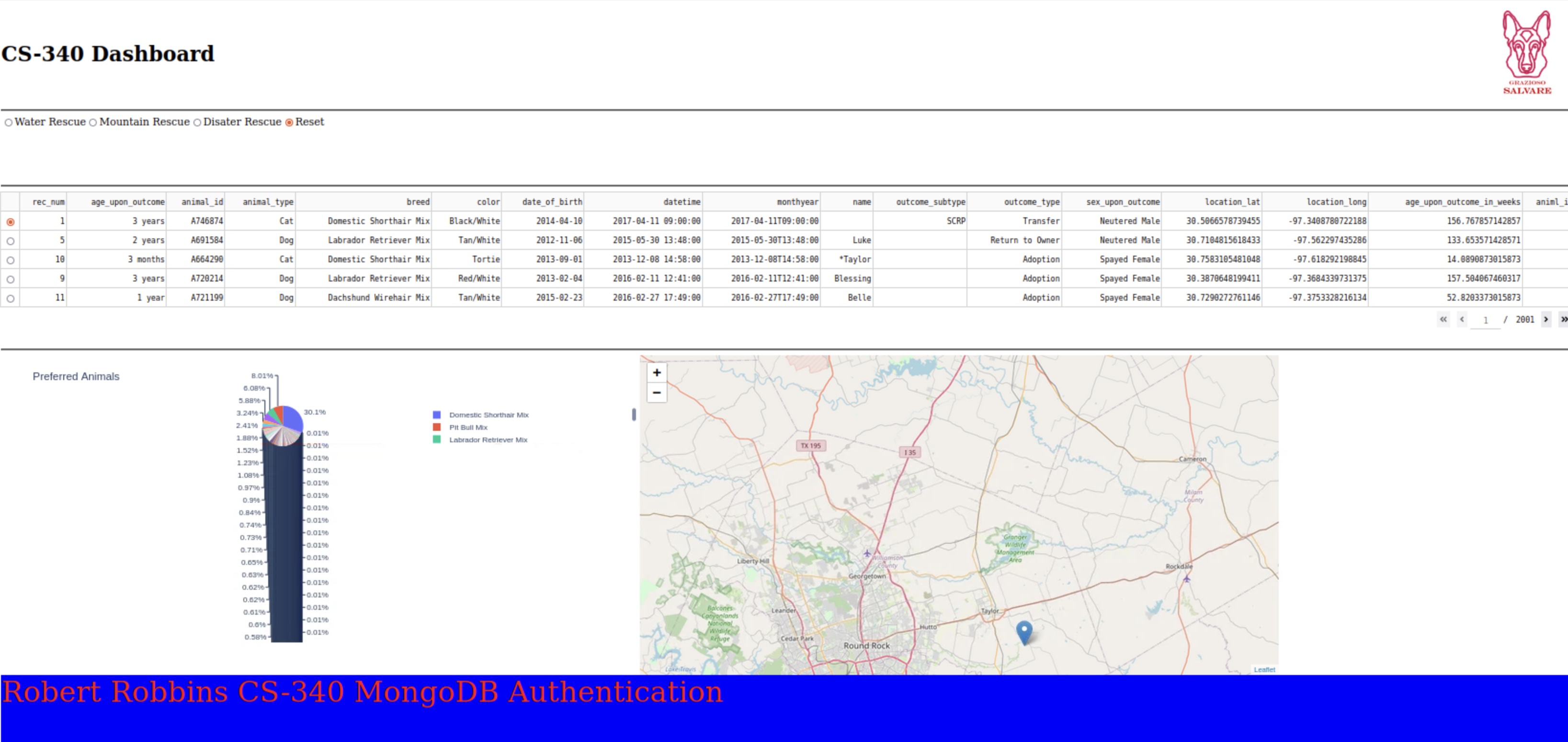
Below are the results for the Water Rescue filter.

Below are the results for the Mountain rescue filter.

Below are the results for the Disaster Rescue filter.



And finally the results from the Reset filter button.



## Contact

Robert Robbins