# 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

The software, when fully complete, will have a database and a CRUD client-facing application dashboard created through MongoDB and Python. The application, will have the ability for animal shelter users to Create, Read, Update, and Delete.

## Motivation

The purpose of this project is to provide a full stack development of an application that will help identify dogs that would be applicable for search-and-rescue training. The original application is meant to take data from five animal shelters in the Austin, Texas region containing characteristics about these dogs and simplify identifying these possible candidates.

## Getting Started

*To get this file running, follow these steps:*

1. *Set up a database via MongoDB. In the case of this software, we used AAC, and the collection name was animals.*

*A screenshot of a computer

Description automatically generated*

1. *Update the PythonModule File. This will be dependent upon your server and user settings. Varying accounts from administrators to standard users will likely need to be set up via MongoDB. For the standard user, use db.createUser and give them only readWrite access. Example below:*

A screenshot of a computer screen

Description automatically generated

1. *Utilize the create, read, update and delete methods. For create, I have implemented a return true and return false depending on if the animal was able to be added. The update returns with how many files were updated. In delete, we allowed delete one or many depending on the situation. The only issues encountered for me were when I overlooked indentation. Since I was using Jupyter, the manual indentation was sometimes necessary.*

## Installation

For this project, I used Jupyter, MongoDB, and a CSV file. You will need to install Jupyter and MongoDB on your device or via the cloud. The CSV file can be mostly blank but will need to have predetermined headers that will be filled out by the user. MongoDB was chosen because it is opensource and easy to use and its usability with Python. Python is recognized as a generally easy to use language because of its simplicity. It is also commonly used when dealing with data. Overall, I believe it was the best language to use for this project. Dash framework was used in creating this program. It is easy to program with Python and is generally considered versatile to use.

## Usage

Below, you will see examples of the code used to create the Create, Read, Update, and Delete functions. Further, you will see the code developed to sort and organize the data within the database. These are specific to the rescues using this program, but the basics are the same and can be manipulated based on your organization’s needs. I have also demonstrated the code that will layout the map and pie charts as well as the code used to create the pie chart.

### Code Example

Below is the code for the CRUD portion of the files:

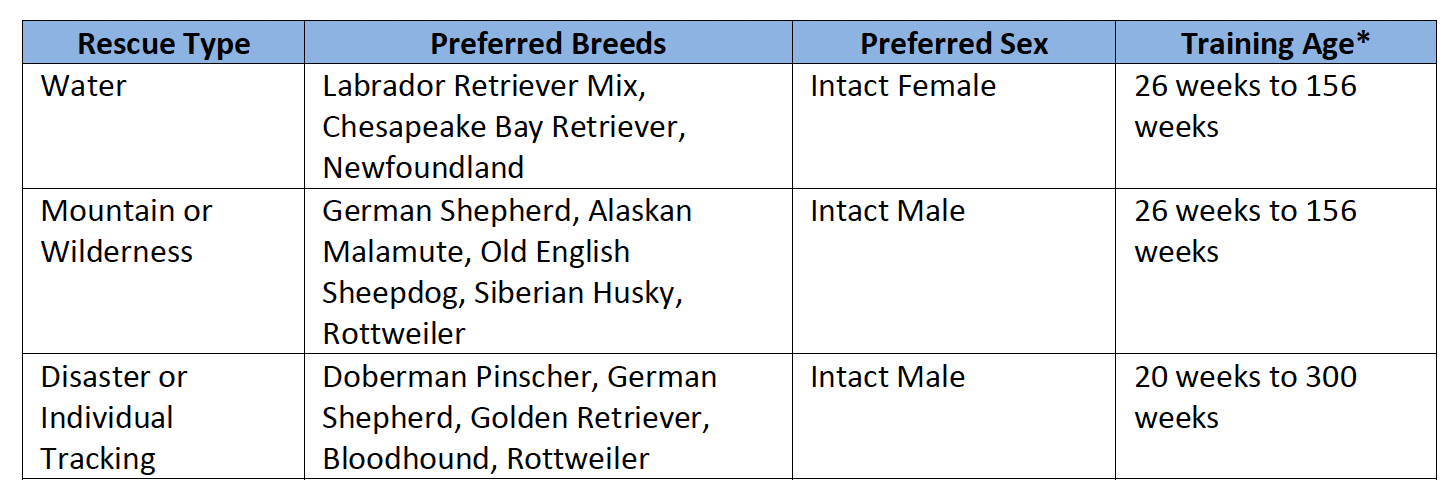
A screen shot of a computer code

Description automatically generated

*A screen shot of a computer code

Description automatically generated*

In the working model of the program, I have deployed several filters based on the below table.



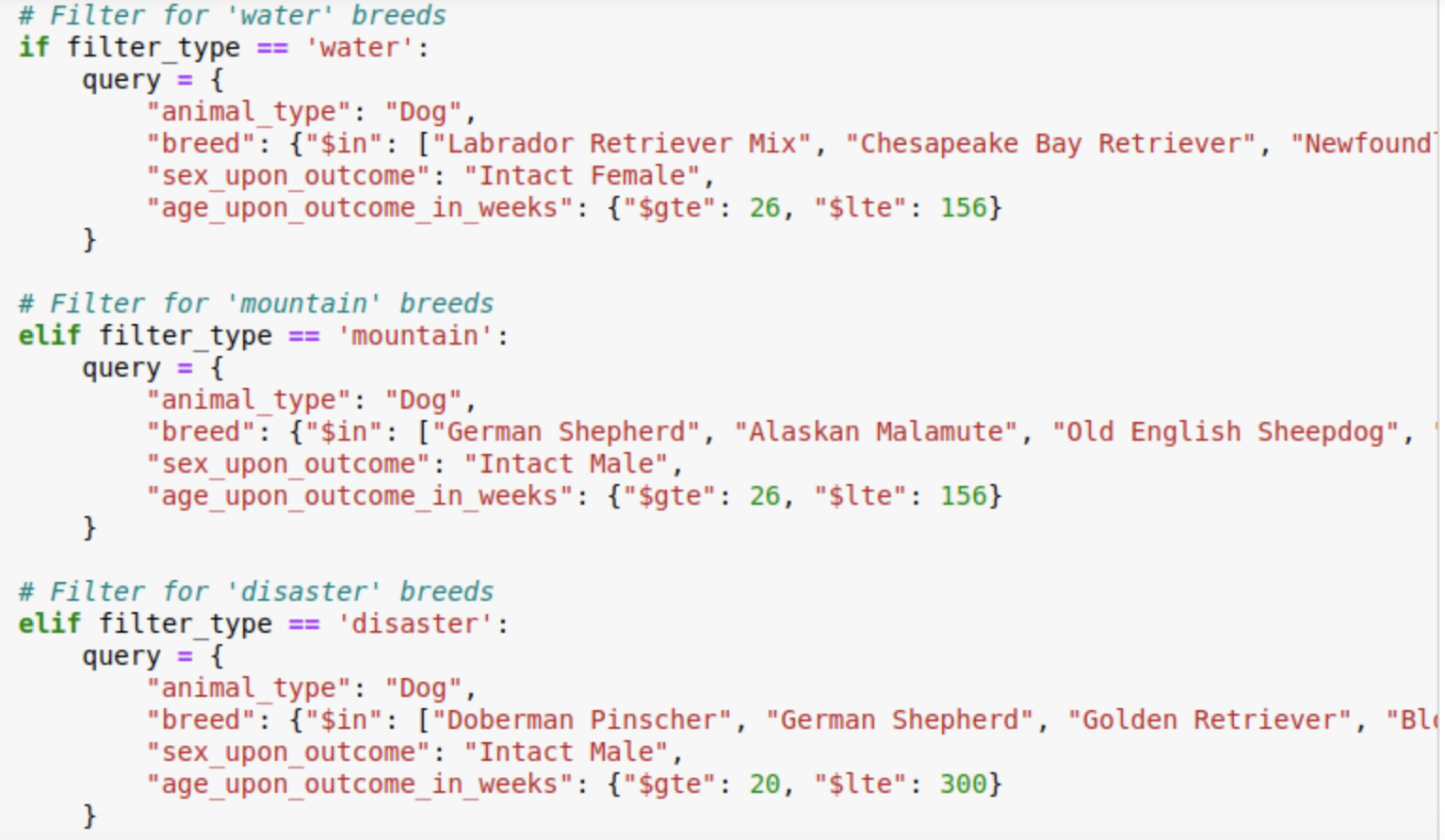
This can be catered to your organization’s needs based on the below code.

**Radio items**



Here, we created labels for the radio items. These will trigger below code to filter data based on the above table.

**Data filters for radio items/Data Table**







Here you can see each filter type from the radio items. While the “animal\_type” is redundant, it will ensure no improperly labeled animals will appear in the results.

**Layout for Map and Graph**



**Graph Code**

### Tests

Below, you will see tests that I ran to ensure that the C and R were working properly. To not skew any data in the database while I am using it, I was focused on birds. You can see that I created a new blue parrot. And was able to find that in the database

A screenshot of a computer

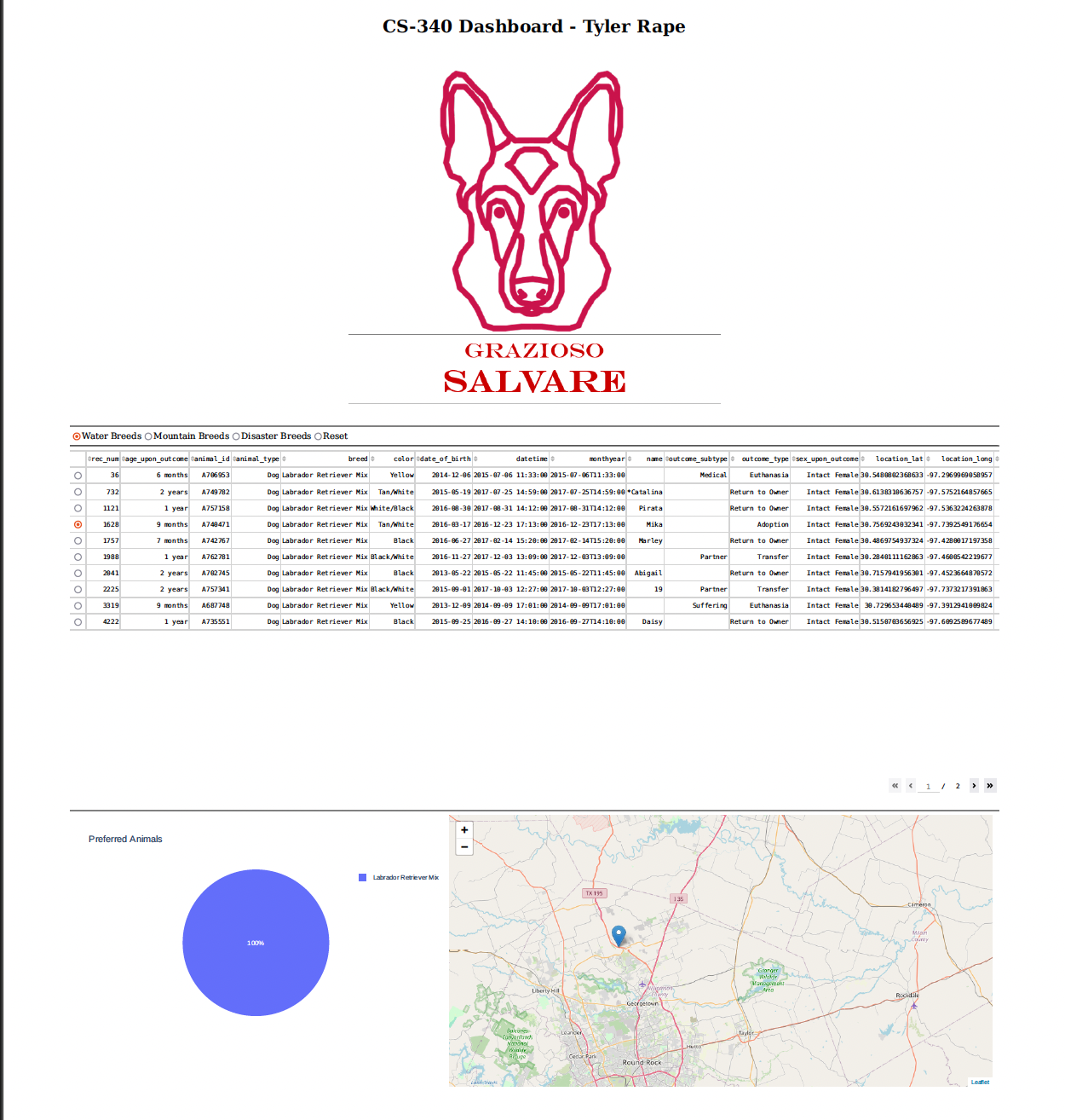
Description automatically generated

In the next section we will focus on testing the U and D. You will see that I worked with files I previously created.

A screenshot of a computer

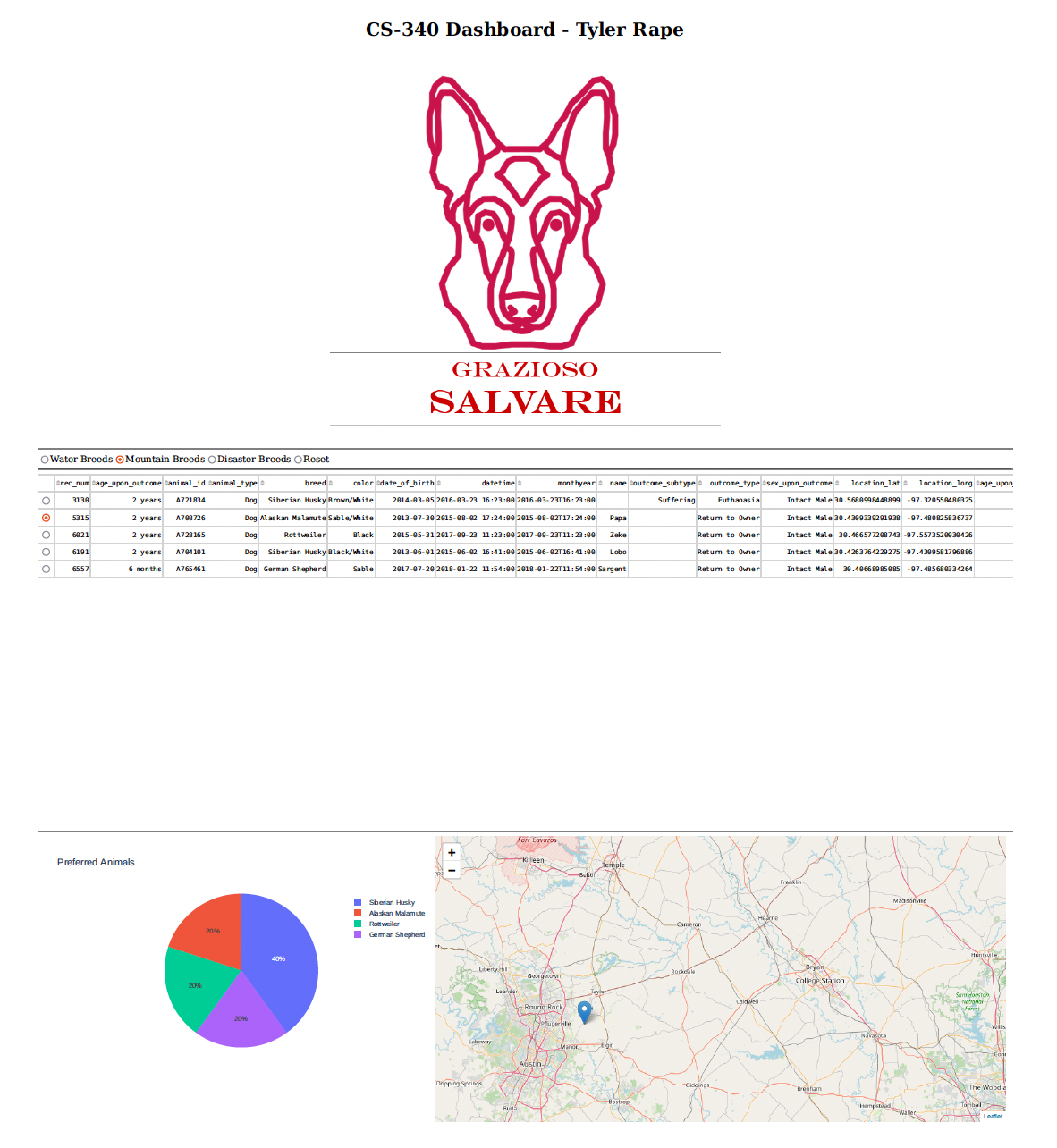
Description automatically generated

**Water**



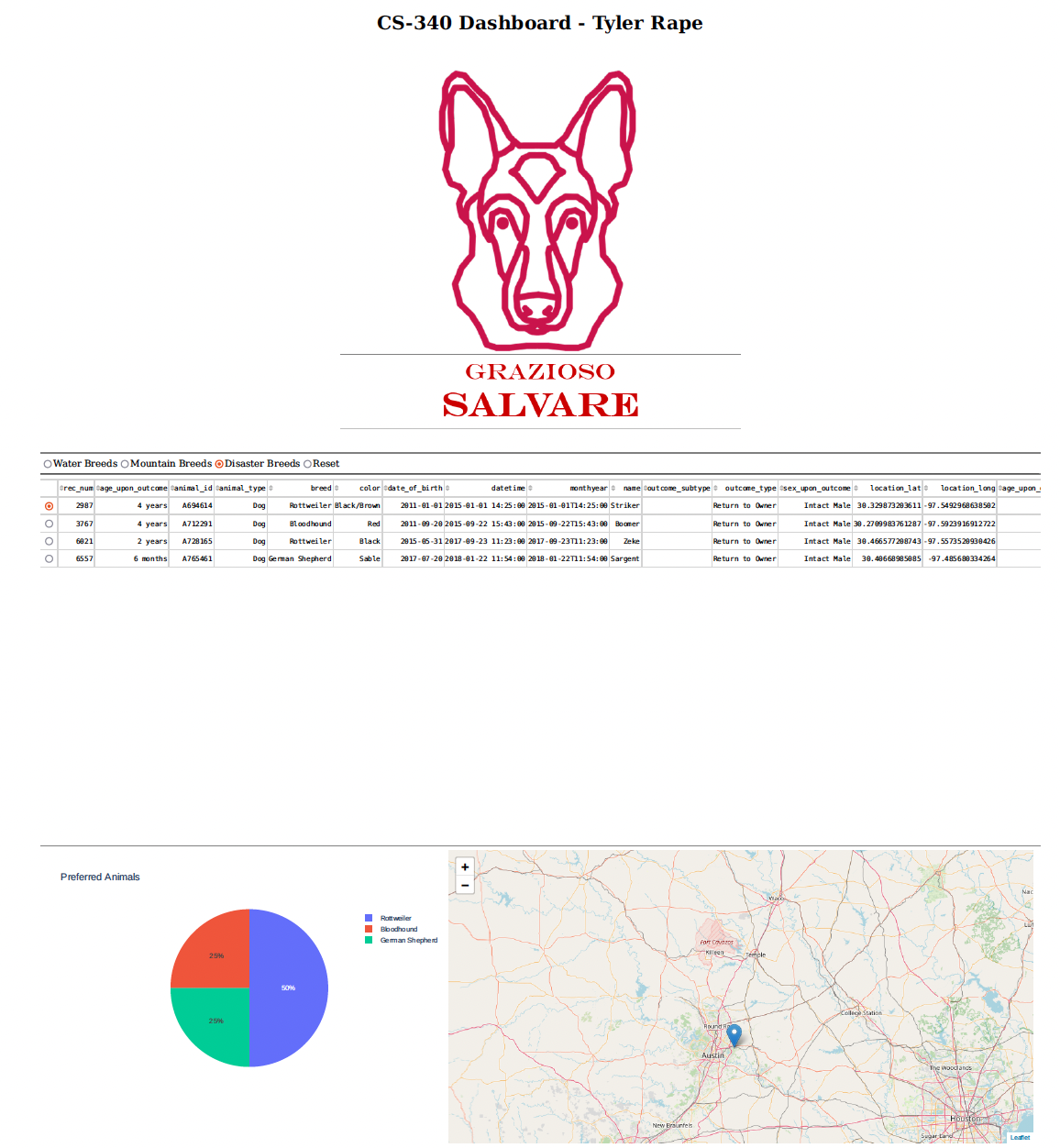
In the case of water, the database only found eligible Labrador mixes that would fit the criteria. You can see that one was selected on the map.

**Mountain or Wilderness**



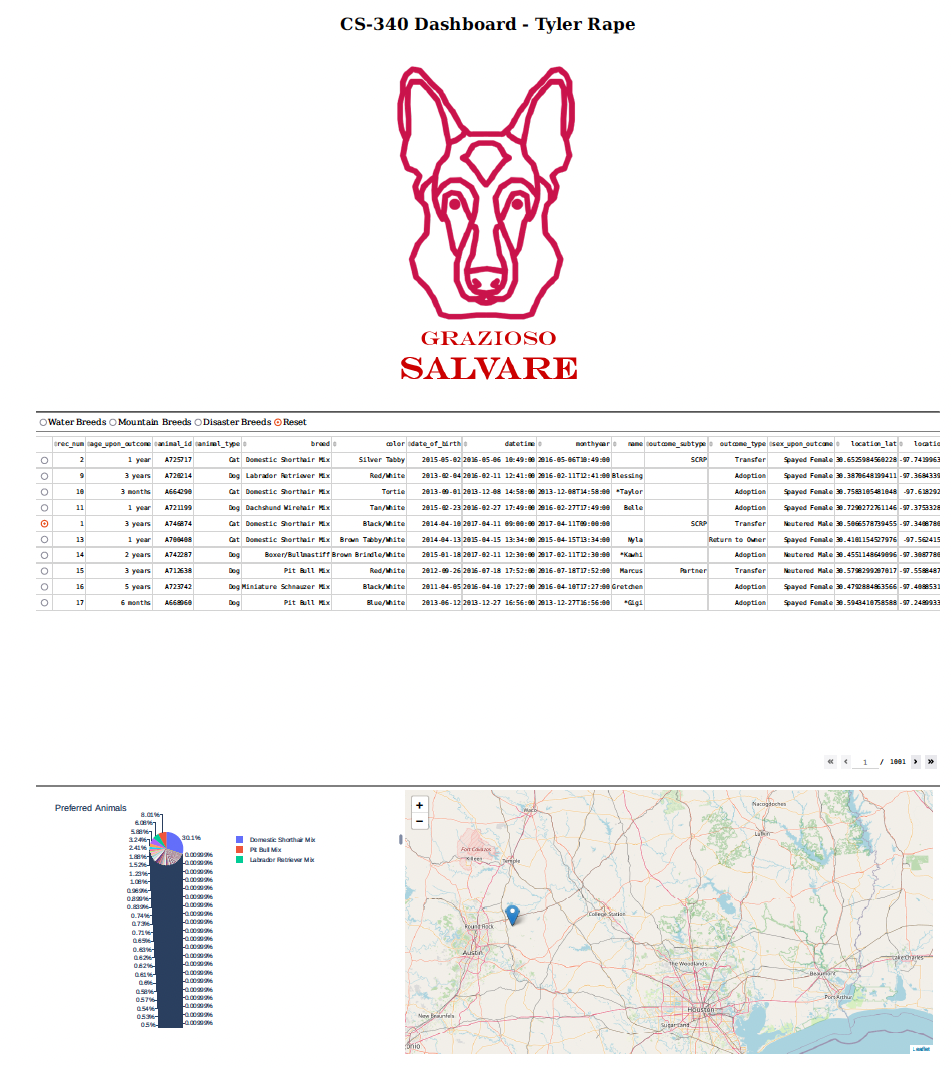
There are a few different breeds that fit for the mountain breeds. The pie chart represents each of these.

**Disaster**



The disaster breeds also have a few that qualify.

**Reset**



The reset shows all animals, even cats. That is why the pie chart is divided so much.

## Roadmap/Features (Optional)

While this program can be improved, there are no current issues with it. The issues encountered were minimal and mostly due to my lack of knowledge. It was a great learning process to develop this program. I overcame these issues by researching each error that came up. I found that many times, it was either an indentation issue with Jupyter or a common issue experienced by many others.  *Note: This section is optional for the purposes of this assignment. If you choose not to fill out this section, remove it from your final README file.*

## Contact

Your name: Tyler Rape