# CS 340 README FINAL

## About the Project/Project Title

Project Two - Animal Rescue Dashboard

This is a dashboard designed to aid Grazioso Salvare in identifying dogs that are good candidates for search-and-rescue training. Specifically the dashboard categorizes dogs in the animal shelter database and serves them to the user depending on which category of rescue dog they require. Implementation contains CRUD framework as well as a Dash dashboard.

## Motivation

In an effort to save dogs from shelters Grazioso Salvare has reached an agreement with a non-profit agency to adopt dogs that may serve well as rescue dogs. To this end Grazioso Salvare needed a software application that can identify the dogs that they would be able to use, to fit this need I have developed a Dash dashboard that interfaces with a MongoDB database to manage and filter the database of animals to the specifications required.

## Getting Started

To use this application you first need to establish a Mongo database that you wish to engage with. Once you have chosen your database and created a valid user account the application can be run on your preferred Python IDE.

Note: The user account credentials are hardcoded into the main class of ProjectOne.py and will need to be updated accordingly.

## Installation

Required Software / Extensions:

MongoDB

PyMongo

Python

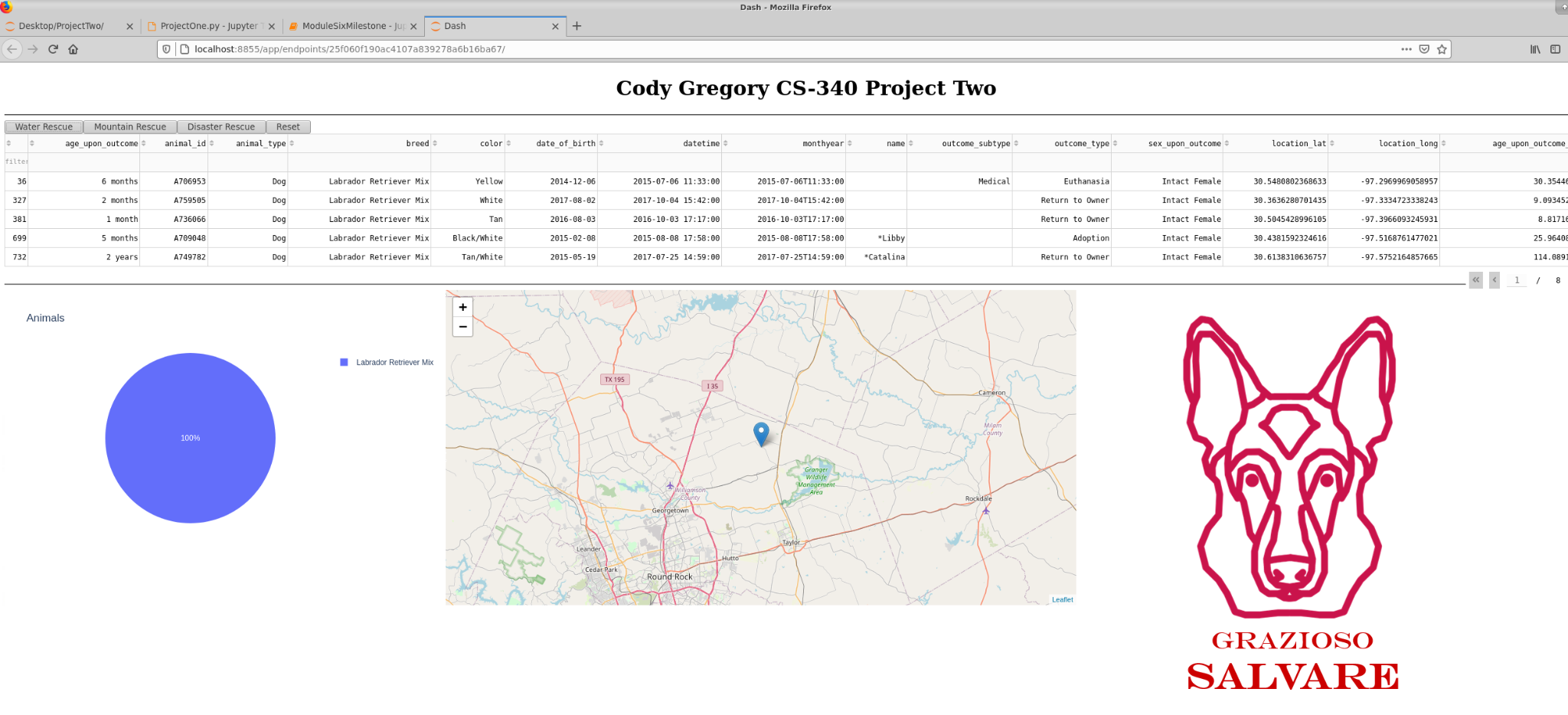
Dash

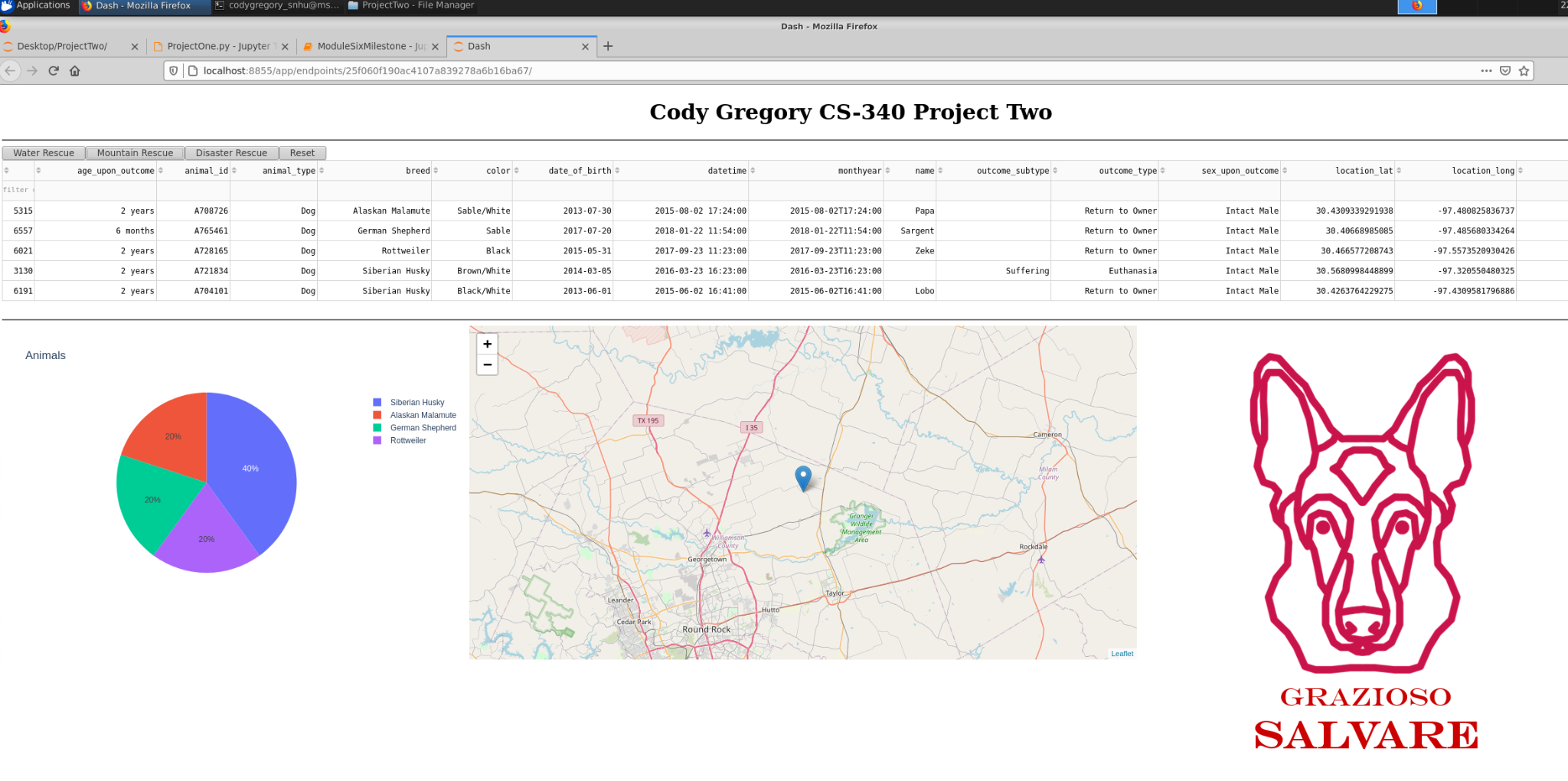
Your preferred Python IDE

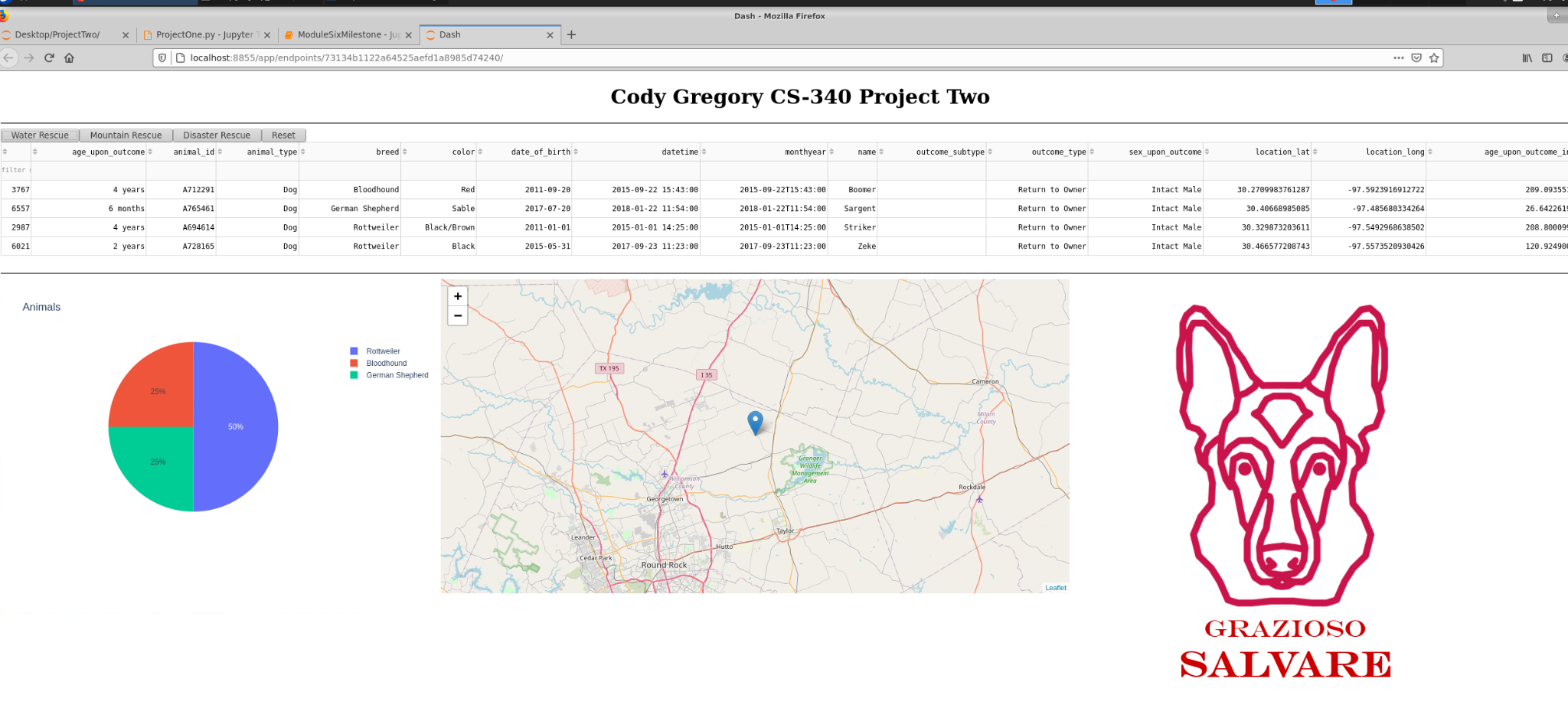
## Usage

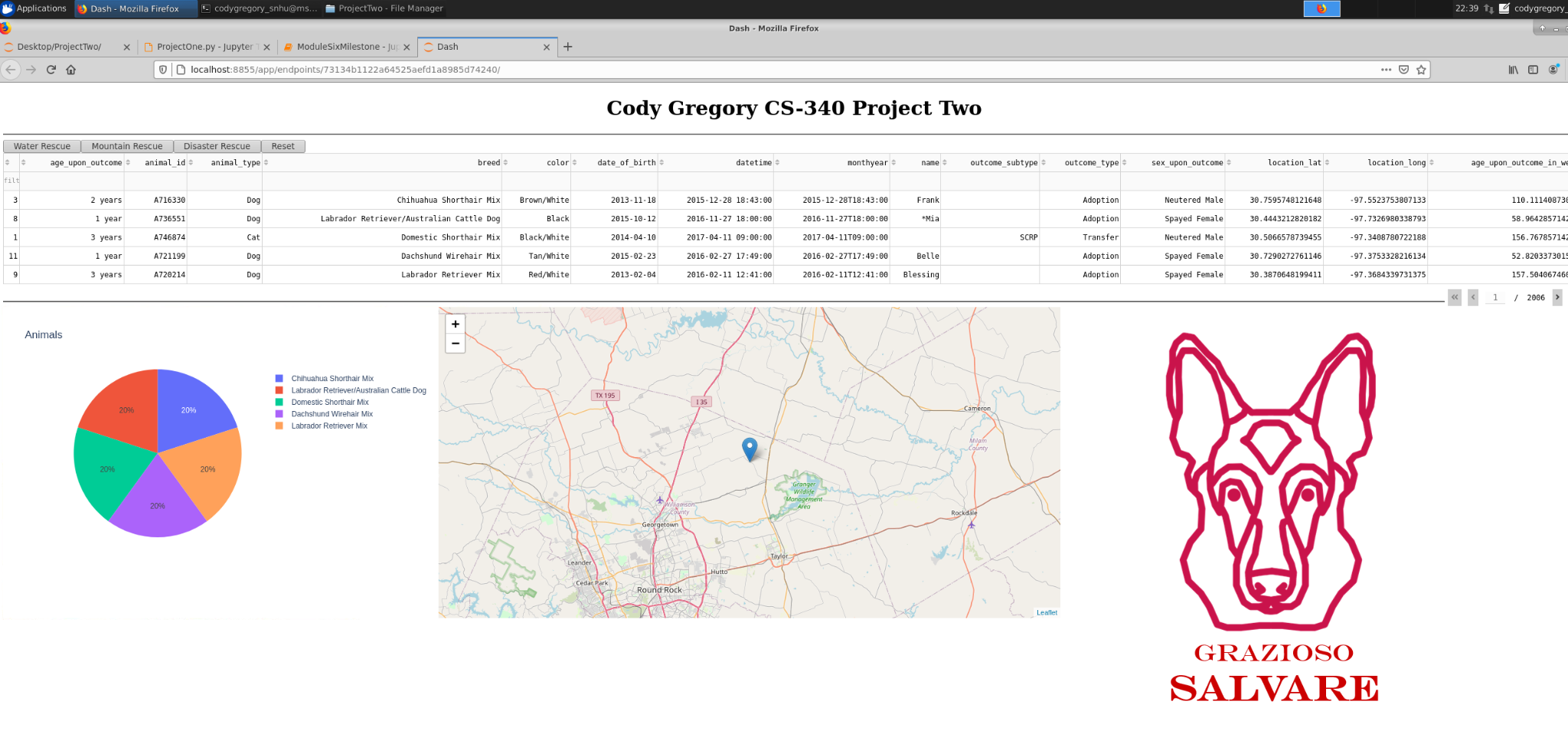
Filter dogs by rescue operation:

Water Rescue:

By toggling a button in the upper left hand corner you can filter your results by dog from specific breeds that are naturally inclined to that type of rescue. Sex status, and age are also considered by the filters.

Mountain Rescue:

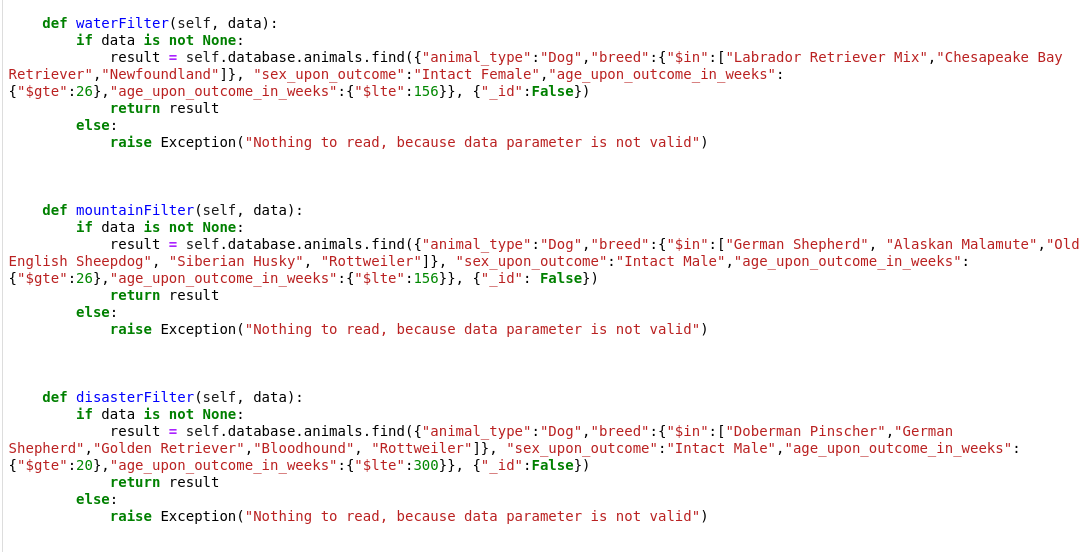
Disaster Rescue:

Or view all of the animals in the system:

While the functionality of the dashboard is designed for rescue dog operations the entire database of animals exists for further customization should the need/use arise.

Dash was used extensively in this implementation. Through Dash we constructed the entire dashboard, allowing us access to features such as dash\_table that provide the main source of information displayed. It also included dependencies to structure the buttons and widgets that are used to interact with the data and generate the pie chart as well, through another dependency called Plotly.

### Code Examples

Custom Filters:

### Using MongoDB query language, custom filters are available to specify search results for specific rescue operations.

**Steps**

The process of completing this project consisted primarily of trial and error. I began by utilizing code that I already had constructed, specifically the milestone six assignment as well as the project one .py file. From there it was a matter of implementing the necessary components and features, one by one until all of the components were there.

My first course of action was to complete the html components. Specifically I needed interactable buttons of a sort in order to engage the filters that will be discussed further on. To accomplish this I started with the course resources and developed standard buttons. These buttons work very effectively, but lack a straightforward approach to maintain functionality, as they operate by comparing the number of times they have been clicked. To combat this I implemented dash radio buttons. Unfortunately while the radio buttons in html exist, and are still included in the code I could not get the callback function to operate with them and the filters without totally breaking the dashboard. Leaving me to stick with standard buttons that work inefficiently.

Once the html was down, I proceeded to generate a graph. Specifically I decided on a Pie Graph for its visual appeal and simplicity. To accomplish this task was relatively straightforward, reading up on the dash core component literature and establishing a figure with values, which generated the chart you see on the dashboard.

Finally I began work on the filters, which I included as separate functions in the .py file for ease of use and to keep the .ipynb file clean. To accomplish these was the matter of determining the correct queries in regards to the specification of each rescue operation. Assign those filters to a button and the assignment stands completed.

## Contact

Your name: Cody Gregory