# Project 2: Grazioso Salvare README

Byron Dewayne Staton

CS-340-T1192 Client/Server Development

Southern New Hampshire University

Dr. Tad Kellogg

October 16, 2022

**Animal Shelter**

## Motivation

The project will help identify specific types of dogs within a certain age range that can be trained for search and rescue missions. Dogs that are ideal candidates will be trained in water rescue, mountain and wilderness rescue, and disaster rescue missions.

## Getting Started

To get a local copy up and running, follow these steps:

* MongoDB must be installed on your computer.
* Start MongoDB from the command line/terminal by tying the “**mongod\_ctl start**” command.
* Type the “**mongo**” command to open the Mongo Shell.
  + You may have to start the Mongo Shell in a separate command line/terminal window.
* Next, create an “AAC” database by typing the command “**use AAC**”.
  + Verify that the database has been created by typing the command “**show dbs**”.
* Type the command “**exit**” to close the Mongo Shell.
* Import the “animal shelter” CSV file into MongoDB using the following command:
  + “**mongoimport --type csv -d AAC -c animals --headerline --drop [your csv file name]**”
* Open the Mongo Shell again and test that the import was successful by typing the following command in your terminal or command prompt: **db.animals.findOne();**

## Installation

To get started, you will need to install the following:

* **MongoDB**

To install MongoDB on your system, first visit <https://www.mongodb.com/docs/manual/installation/>. Download the Community or Enterprise edition for your machine. Run the executable file and follow the installation guide. Refer to the Community or Enterprise tutorials for further guidance.

* **Python 3**

To install Python on your system, first visit <https://www.python.org/downloads/>. Download the appropriate file for your system. Run the executable file and follow the installation guide. If you’re using Windows, add Python to the environmental variables. Ensure you install the package installer for Python (pip).

* **Jupyter Notebook**

Once Python and pip is installed, run “**pip install notebook**” then run “**jupyter notebook**” to start the program.

* **Plotly Dash**

[Plotly](https://dash.plotly.com/introduction) is a Python web application framework and must be imported directly into your *.py* and/or *.ipynb* file. Once imported, you will be able to utilize the Dash Core Components in Jupyter Notebook.

* **Pandas**

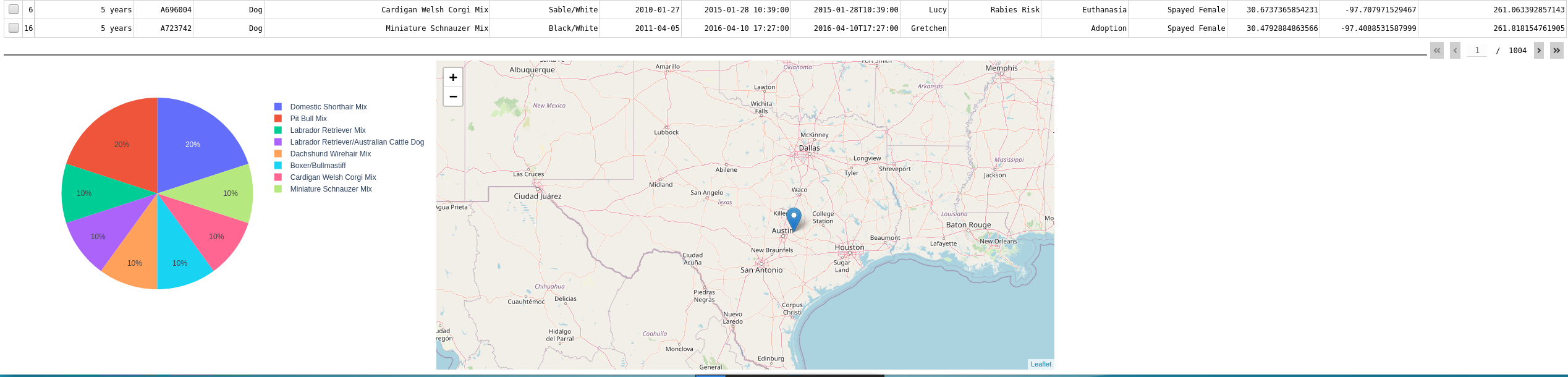
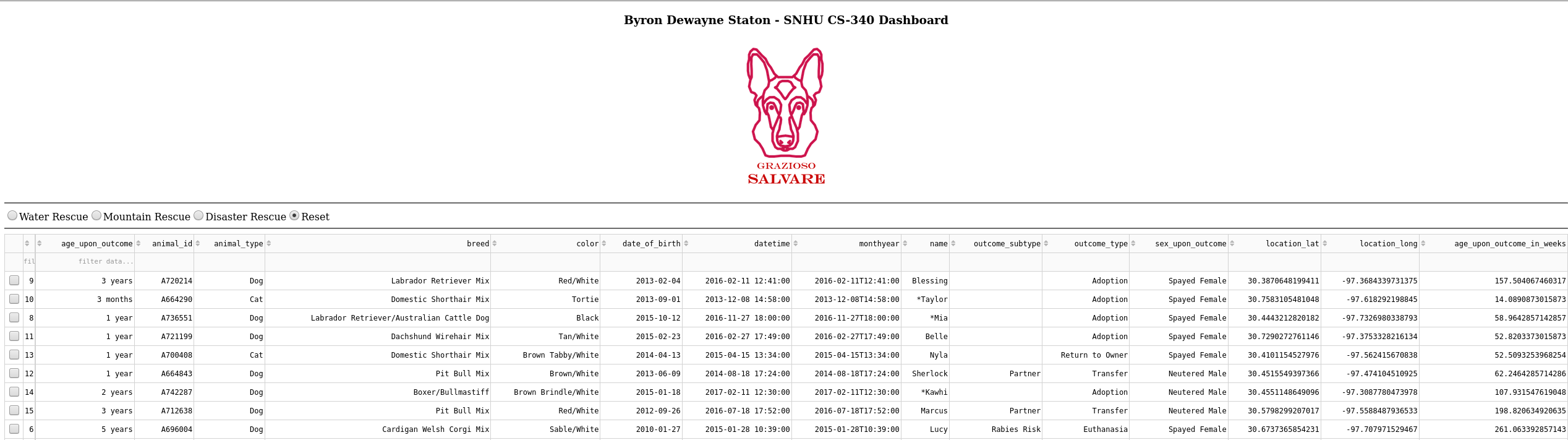
[Pandas](https://pandas.pydata.org/) is a Python library that is used for creating and displaying data frames in the browser. Pandas must be imported directly into your *.py* and/or *.ipynb* file.

* **Dash Leaflet**
  + [Leaflet](https://dash-leaflet.herokuapp.com/) is a Python library that is used for generating maps that are displayed in the browser. Leaflet must be imported directly into your *.py* and/or *.ipynb* file.

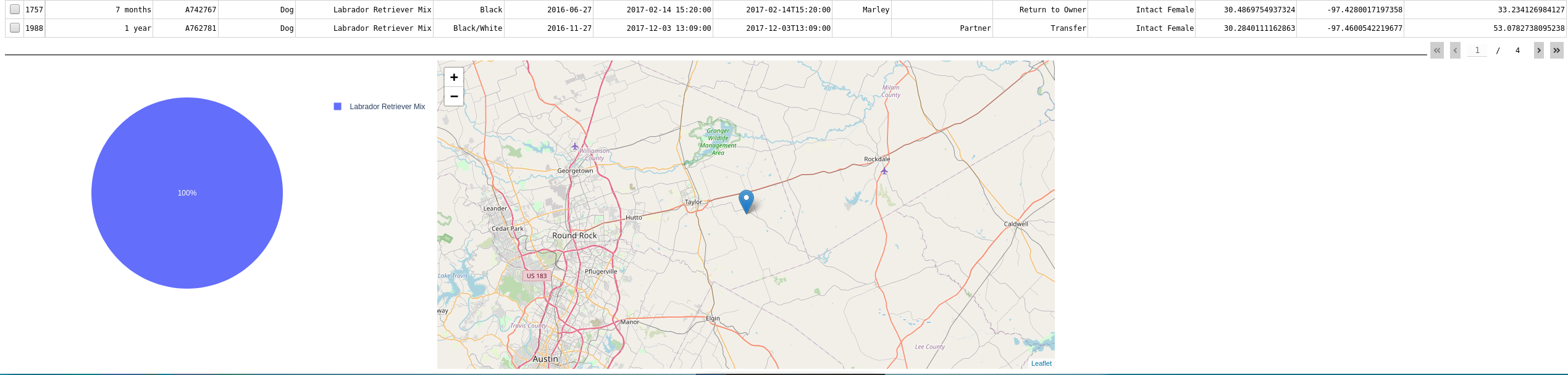
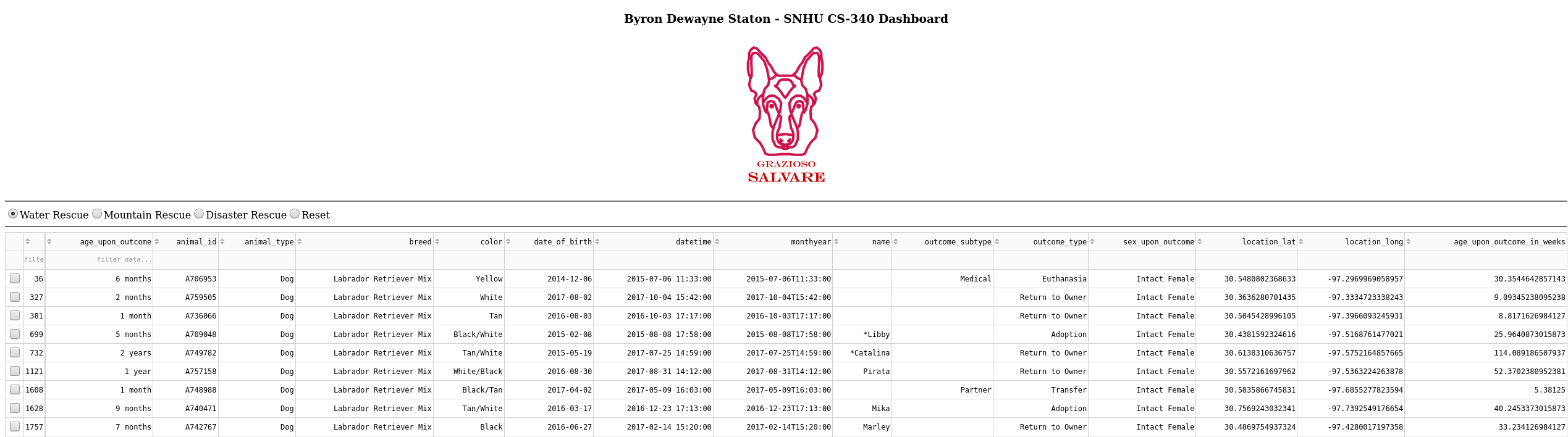
## Usage

The project takes in a CSV file containing data set of animal center outcomes via the MongoDB shell. MongoDB stores the data from the import and interfaces with Python by importing the PyMongo library. With the help of the Dash framework, we can create an HTML layout which displays components such as a heading, a logo, and radio buttons. The data from MongoDB appears in a table. With this data, we use radio buttons to filter the lists of animals to search for ideal rescue training candidates. The pie chart updates are based on your filtered data. A leaflet map pinpoints the location of the animal(s) and returns their breed and names when you click on the pin-drop.

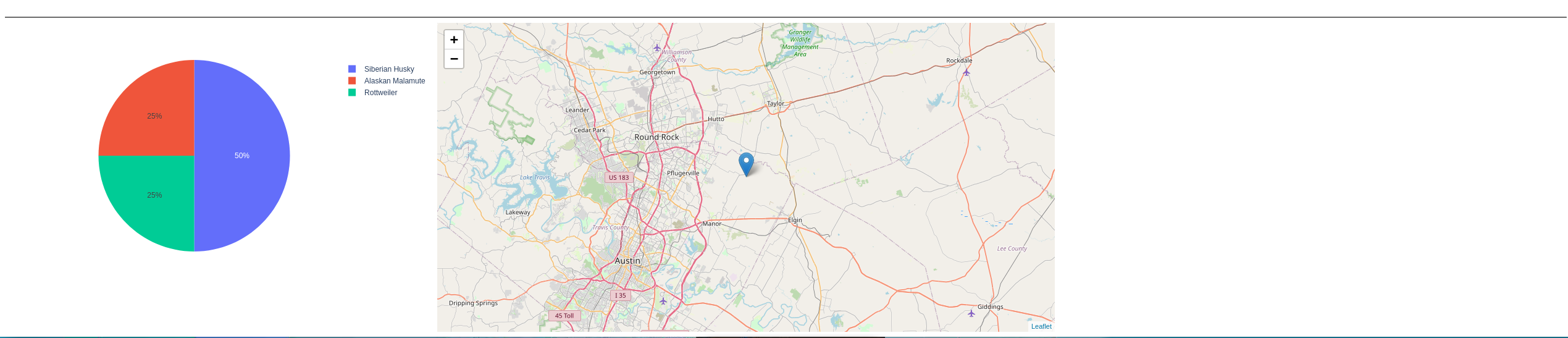
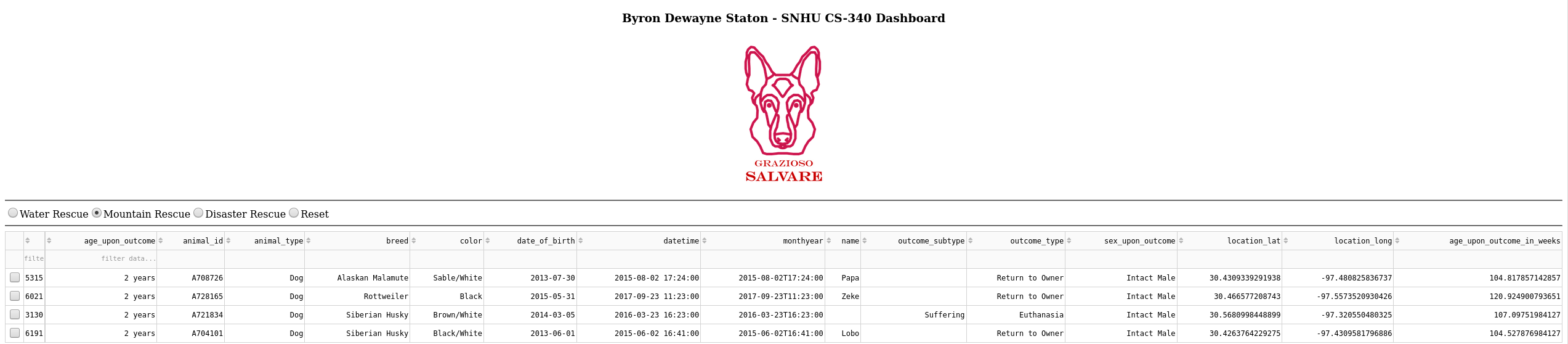
**Unfiltered / Reset Data Results**



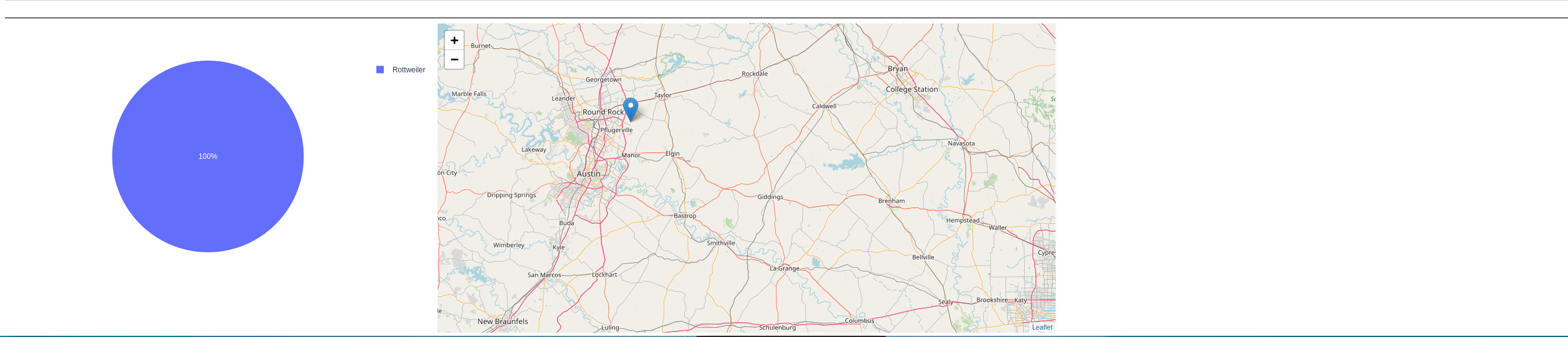
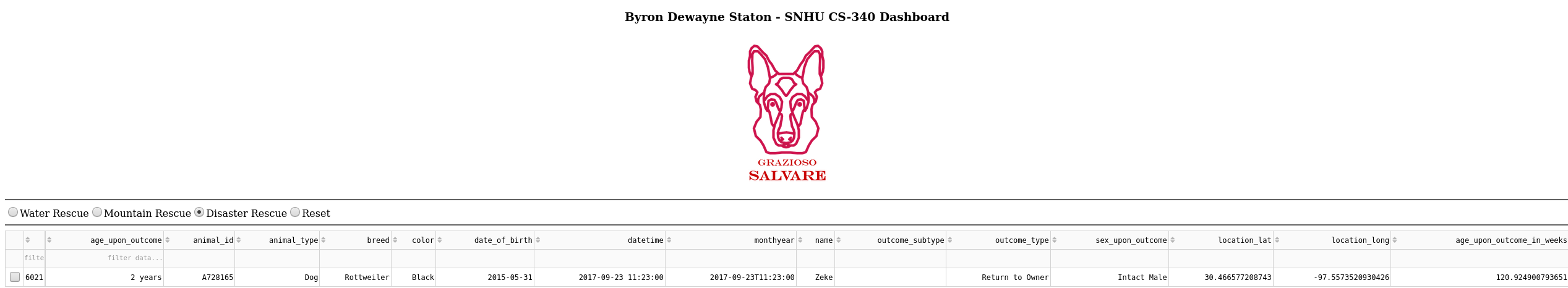
**Water Rescue Data Results**



**Mountain and Wilderness Rescue Data Results**

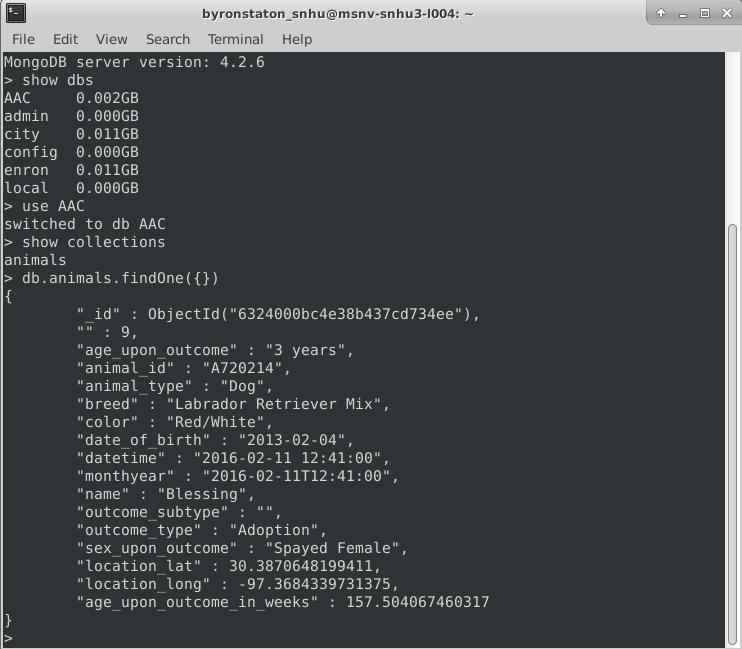


**Disaster Rescue Data Results**



**Additional Screenshots**

*Sample of a Query in Mongo shell after import of CSV file*



*Connecting Python and MongoDB in AnimalShelter Class*



*Testing CRUD Functionality*



*Sample of a Complex Query for Filtering the Interactive Table*



*Example of Dash Framework HTML Layout*



*Example of Leaflet Map Update*



**Steps Taken to Complete the Project**

In order to complete this project, I first created an “AAC” database and “animals” collection in MongoDB. Next, I imported a CSV file containing the animal data. After verifying that the data was successfully imported, I created added authentication to protect the data.

I used Jupyter Notebook to create an “AnimalShelter” class in Python file named “animal\_shelter.py”. The class is used to define the program’s CRUD functionality and return exception messages in case of failure. I then created an “animal\_test.py” file to test the CRUD functionality using hard-coded data. I created a dictionary called “data” which contained an animal type, breed, color, and name. I passed “data” as a parameter for the rest of the CRUD functions (read\_data, update\_data, delete\_data).

After clearing out errors and bugs from the program, I began putting it all together to create the Grazioso Salvare program.

**Challenges**

Throughout the development of the program, there were challenges in understanding errors in Jupyter Notebook. Many of the error messages were not obvious which made it difficult to find and fix bugs. To overcome this challenge, I first went to the line where the error was thrown and scanned the code going upward from that point. If I was unable to locate the bug, I would write multiple print statements to help me narrow down where errors occurred.

Ensuring I used indentions correctly was also a challenge at times. I found that the code would run without error, but nothing would appear on the screen. What I found is there would be something out of place, even if by just one space and the program would not load.

## Contact

*Byron.Staton@snhu.edu*