# CS 340 AAC Data application README

## About the Project/Project Title

This project is called *Search for Beethoven*, it is a CRUD application that uses python to interface with a NoSQL database. This project was created for Grazioso Salvare. The underlying data contains outcome records for several animal shelters in the Austin area. This project is currently in the very early stages of development and has yet to reach a user-friendly state.

The project currently uses MongoDB, a NoSQL document-oriented database system and PyMongo, a python library with MongoDB tools. Mongo is where we will store our data, and PyMongo will be used as the layer that links the data to the interface. The interface will use the Dash Python framework to create a data dashboard. Dash is a quick and easy framework that can be used in tandem with PyMongo to create a data driven application.

## Motivation

This project was created to assist Grazioso Salvare in their search for suitable search and rescue dogs. The outcome data for an animal shelter can be very dense and difficult to go through manually. Thus, to make things smoother for Grazioso Salvare, we have created this application.

## Installation

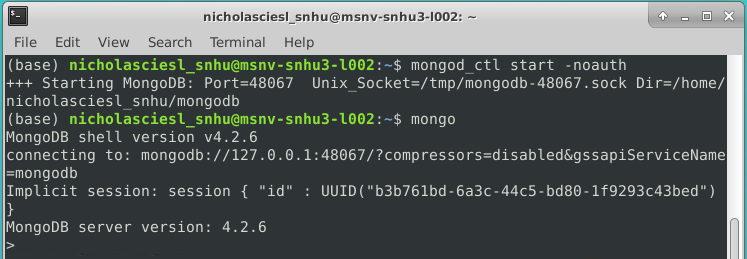
Most importantly, you must be using a machine that has the underlying data installed as it has not been distributed yet. You also must have a copy of the python files.

To install **Jupyter**, enter “pip install notebook” in your command line and hit enter. Jupyter will be used to view, edit, and test the python files that are needed for this project. It is a simple web-based IDE that lends itself well to the workflow we have chosen to use for this project.

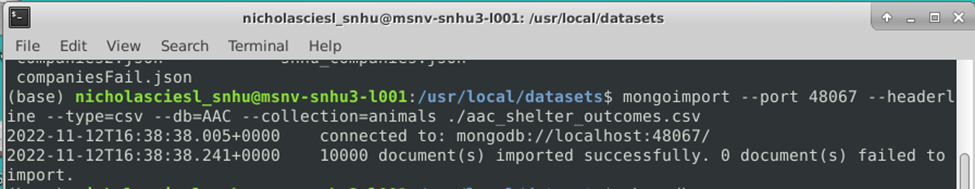
You will also need to install MongoDB. This can be a somewhat involved process and is out of the scope of this document. Information on this install can be found [here](https://www.mongodb.com/docs/manual/installation/).

## Getting Started

Currently, to get a local copy running, you must follow these steps:

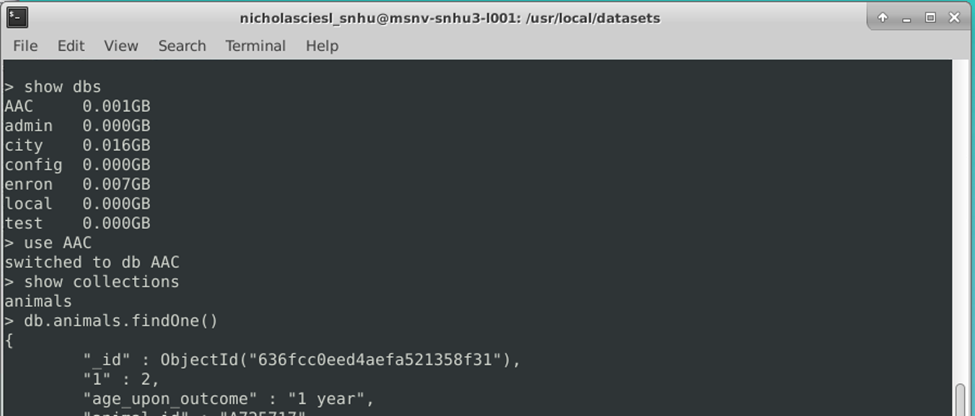
1. Start the MongoDB server by typing “mongod\_ctl start-noauth” in your command line application and hit enter. Then, connect to the server by typing “mongo”. You will know that you are connected to the mongo server when the system username goes away and the “>” operator appears before your command instead of the “$” character.
2. To load the data into your mongo database, exit Mongo and return to the terminal. This is achieved by typing “exit” and hitting enter. Once you are in the terminal, navigate to the folder where the csv file is stored using the “cd” command. Once you are in this folder, execute this command:

Mongoimport --port \*port number from previous screen\* --headerline --type=csv --db=AAC --collection=animals ./\*filename\*

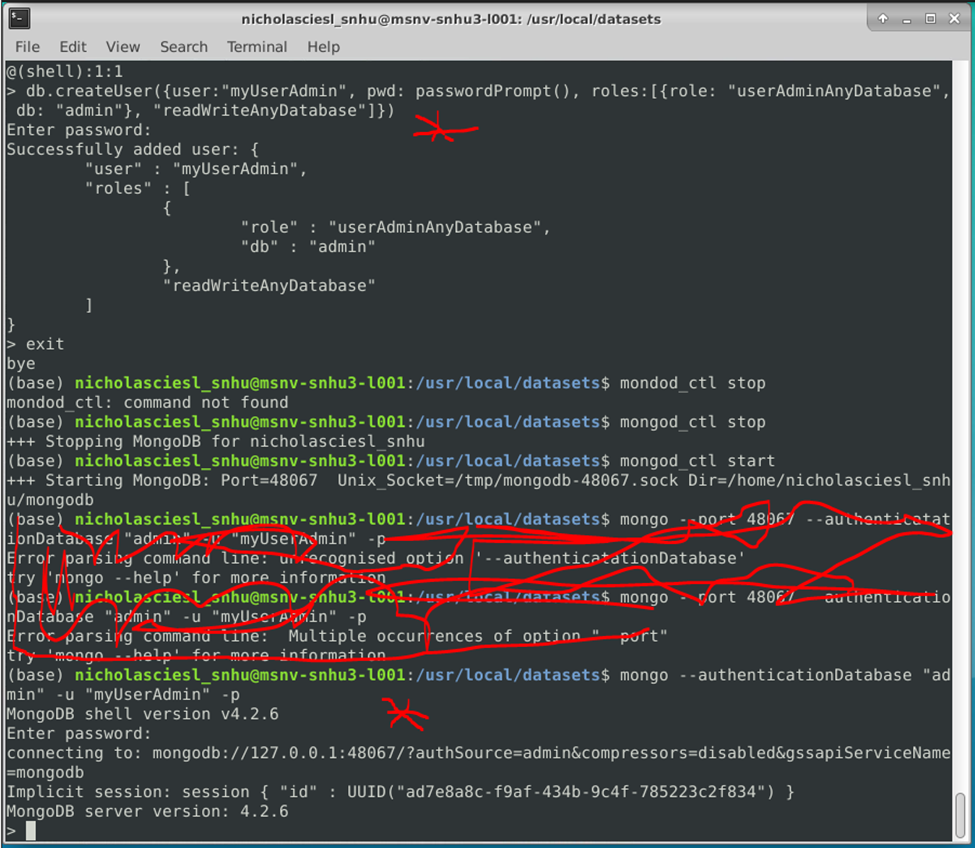


Be sure to replace the text in asterisks \*\* with the port number your machine shows during step one and the name of the file you wish to import.

1. Reconnect to mongo and type execute the “show dbs” command, if the import was successful, you will see the “AAC” database in the list.

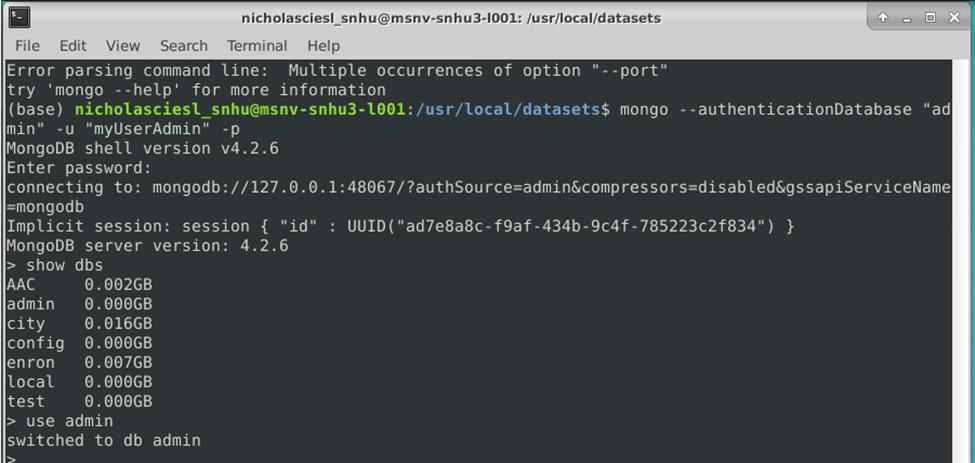


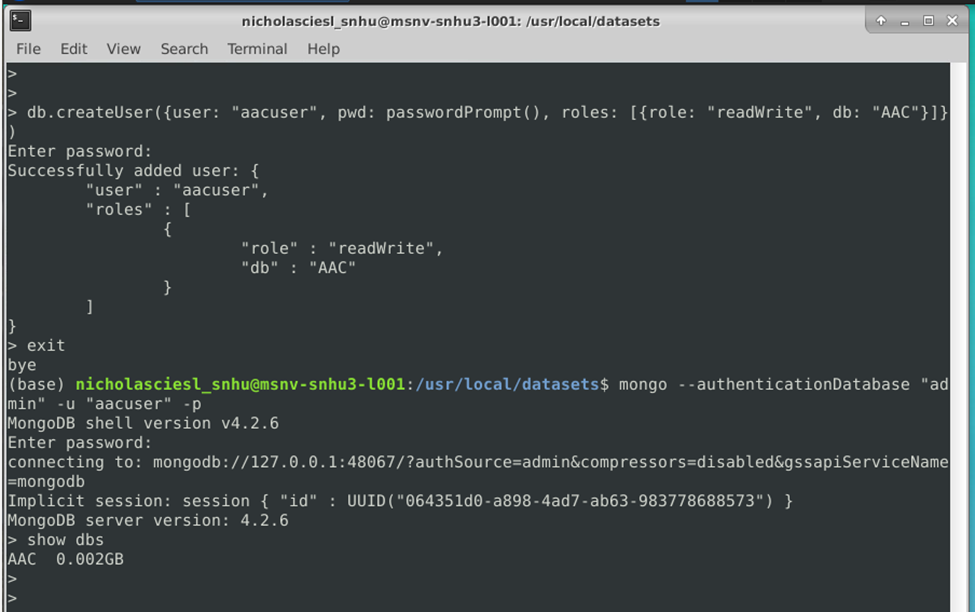
### Authentication Setup

1. Authentication is important to any project as it helps ensure the security of your data. We will start by setting up an admin account. First, reenter mongo if you are not already in mongo. Then execute the “db.createUser()” command shown in the image below. You will be prompted to enter a password. Any characters you type will not appear, but they are being accepted.
2. To login as this user, exit mongo and stop mongo with the command “mongod\_ctl stop” and restart it with “mongod\_ctl start”. This starts mongo with authentication enabled. You can now connect to the database with the command :

mongo --authenticationDatabase “admin” -u “myUserAdmin” -p”

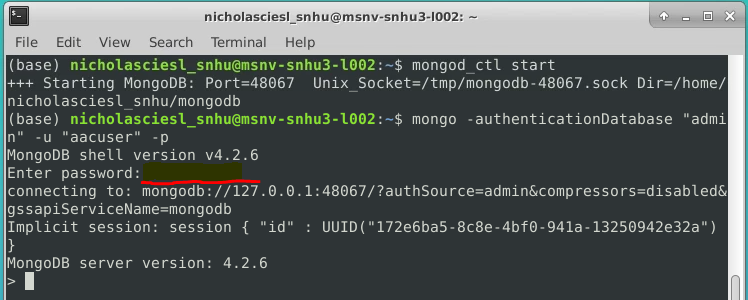
You will be prompted to enter the password you have just created.



1. Now that you have successfully created and logged into an admin account, you should create a user account. This will be very similar to the creation of the admin account, but the user should only have permission to access the database that they will need for app operation. This command can be seen in the screenshot below:

You can now exit mongo and login to this user account with the following command.

mongo --authenticationDatabase “admin” -u “aacuser” -p

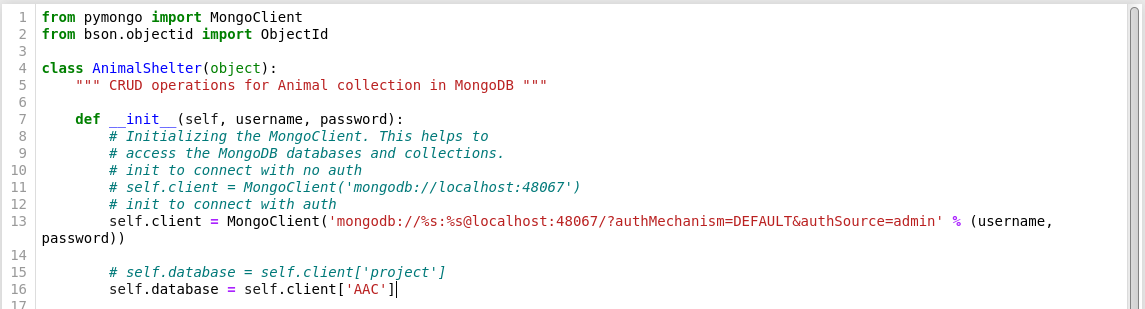


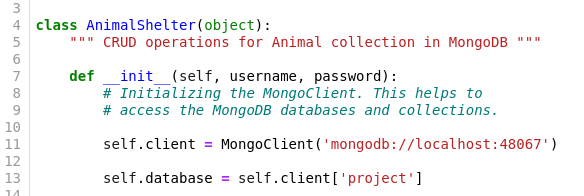
if you use the “show dbs” command, you will see that the only db available will be the AA C db we just imported. This will allow for better database security and data integrity.

1. Place animal\_shelter.py and ProjectTwoDashboard.ipynb in the same folder and start Jupyter Notebook.
2. Navigate to this directory in Jupyter. If you do not have Jupyter installed, please reference the “Installation” section below.
3. Open ProjectTwoDashboard.ipynb and click run.

## Usage

### Code Example (Database Driver)

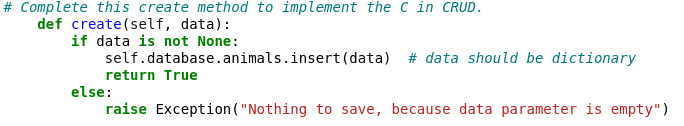
The module connects to the database on startup with a username and password the username and password are arguments for the constructor:



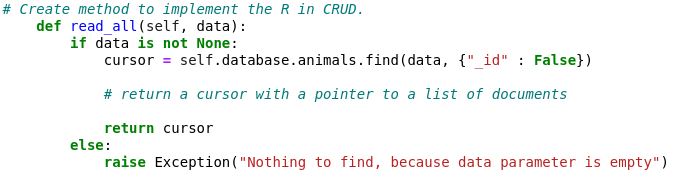
Although, it is not difficult to revert to connection without authentication for development purposes.

there are five functions available to any importing program. These functions are create, read\_all, read\_one, update\_many, and delete\_many. All these functions accept data as a parameter. The update\_many function also accepts the values to be updated. This data should be in the form of a dictionary, but there is currently no check for this. We do, however, make sure that data is not empty.

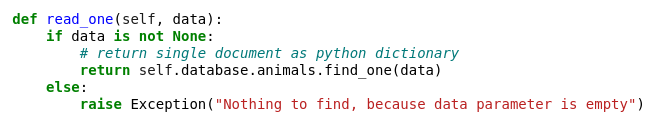
create:



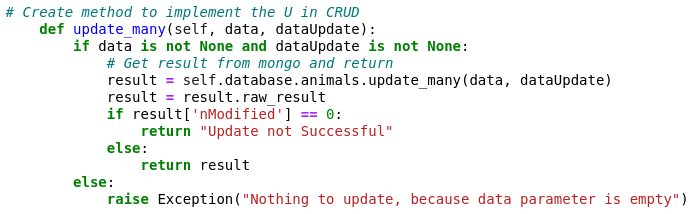
read\_all:



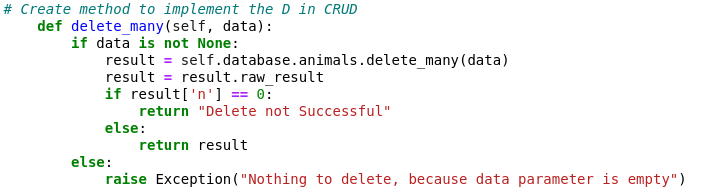
read\_one:



update\_many:

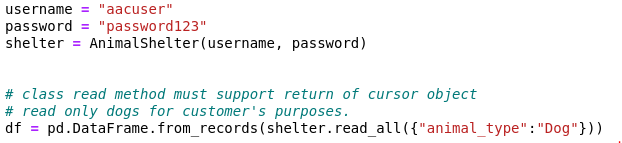


delete\_many:

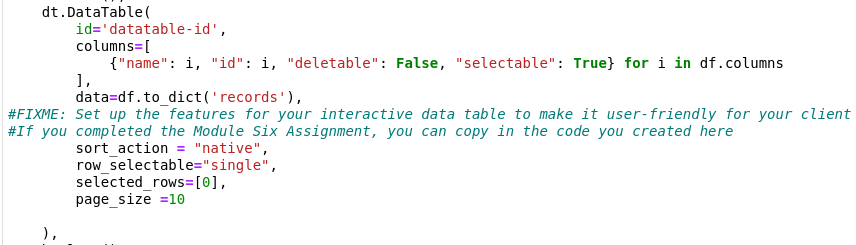


### Code Example (Dashboard)

The dashboard connects to the database and logs in with the username of the accuser account.



### This data is then loaded into an interactive table



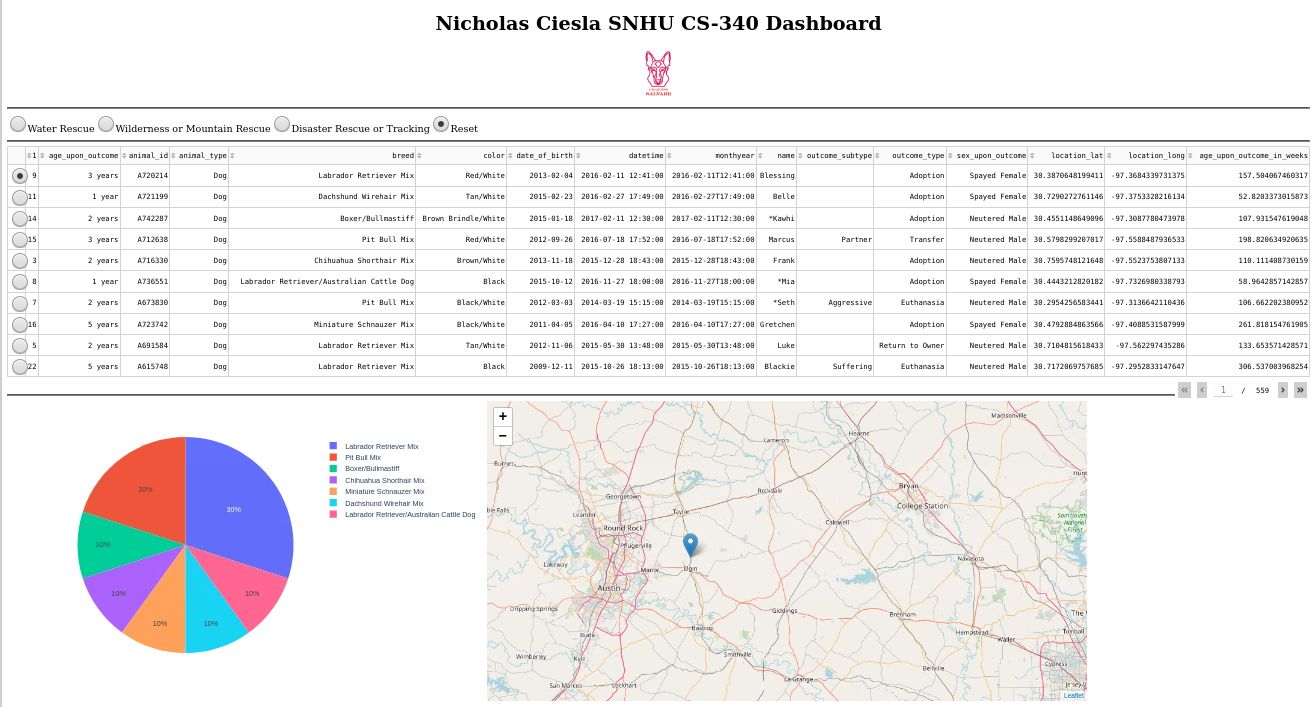
From here, there are several ways to interact with the table that we will get into in the testing section. One of these is selecting a row with a radio button located on the left side of each row. When this is done, the map updates with the location of that dog.



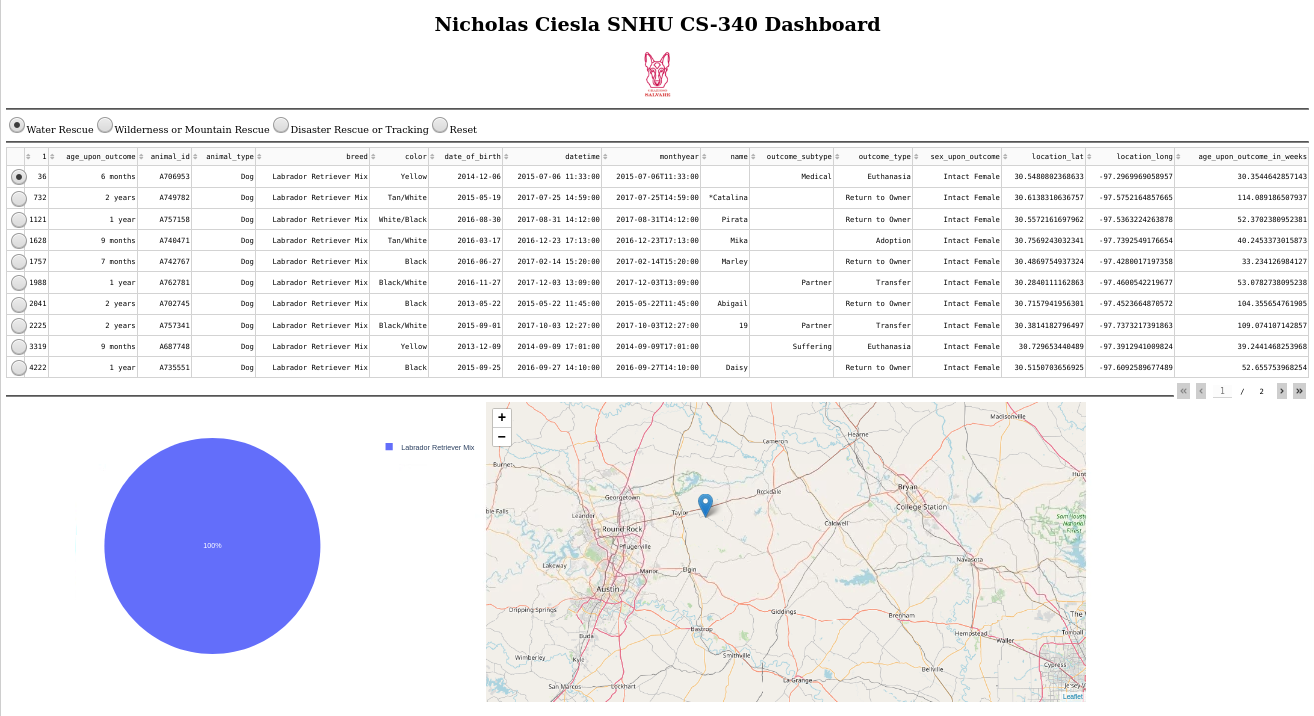
### Tests

The current dashboard file exists as a Jupyter notebook file. This test can be run by opening the file in Jupyter Notebook and clicking run. You can run the file again by copying the text and pasting it in the box at the bottom. This also allows you to make as many changes as you like to the test file. However, if a change has been made to the animal shelter, it may not be reflected until you start a new notebook. This test is the deployment ready dashboard.

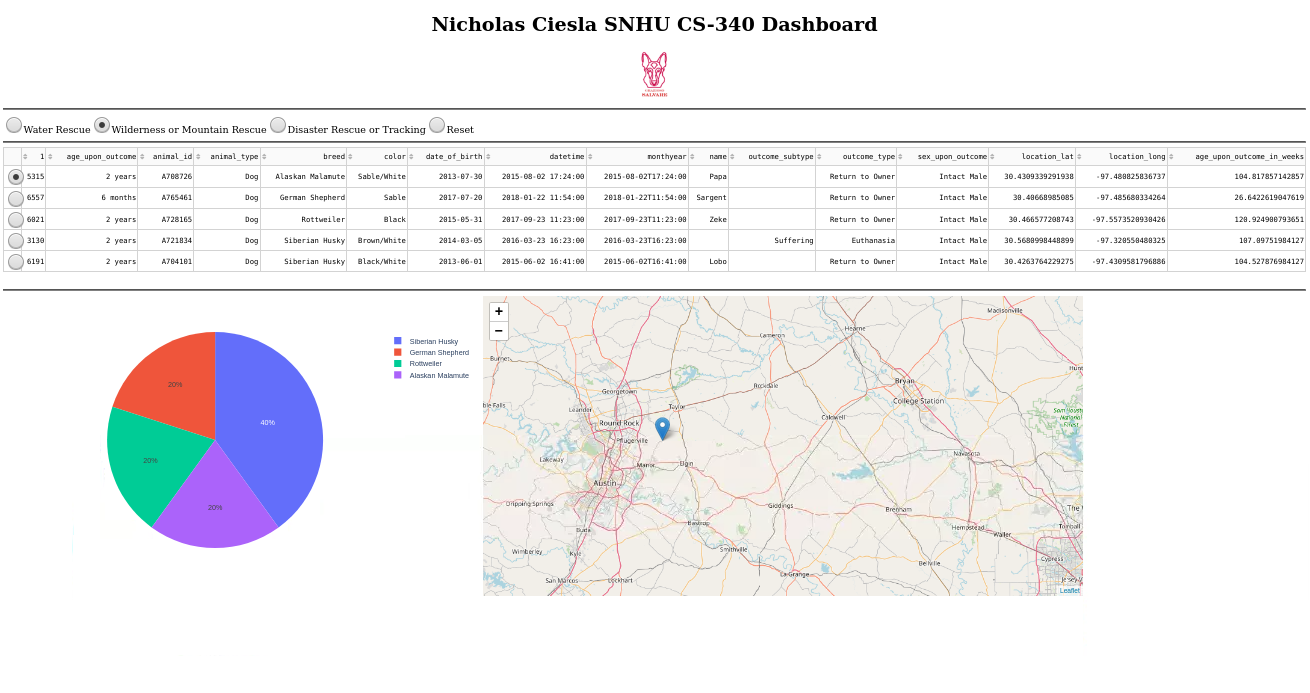
### Initial load, all dogs with breed pie chart and location of selected dog. The pie chart only displays the breeds for that page as things get difficult to interpret otherwise. This can easily be changed if need be:



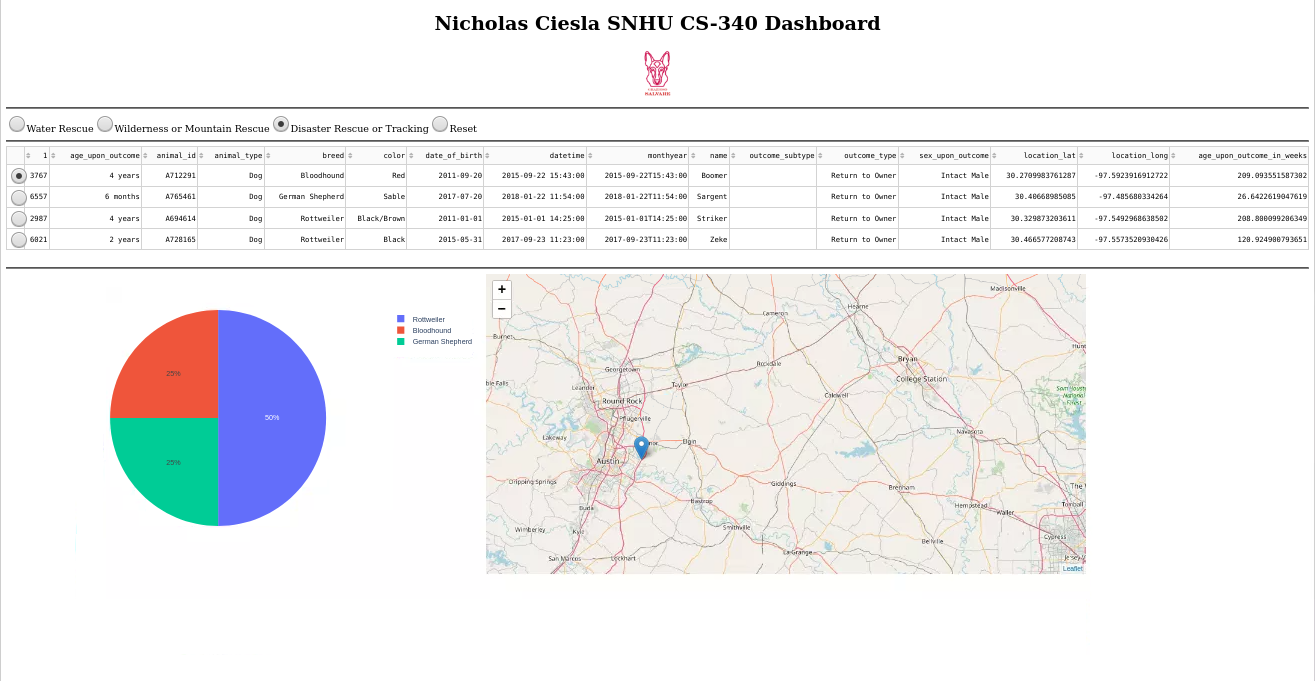
Water Rescue Filter selected:



Wilderness or Mountain Rescue Filter selected:



Disaster Rescue or Tracking selected:



## Roadmap/Features (Optional)

The Dashboard is currently in a ready to deploy state. If user testing with Grazioso Salvare goes as planned, we can make the dashboard public, and use can begin. There are a few small adjustments that can be made to improve the app as well if need be. These include filter adjustment and layout changes.

This project began as a mongoDB collection, became a python driven database, then dashboard functionality was added, and now we have a fully functional dashboard. This dashboard allows the user to view the entire AAC outcomes dataset, filter it as they please with custom filters, use preset filters, and view graphics based on data viewed.

The largest challenge that we encountered when creating this dashboard was the general lack of beginner friendly documentation on the Dash framework. This may prove to be troublesome when an update is needed, but we will persevere as we did this time. Another challenge that we encountered was the fact that the data leaflet documentation site was down through the entire development of this dashboard. This made things a little difficult, but code samples were found that helped us learn the correct syntax just fine.

## Contact

Your name: Nicholas Ciesla

Email: Nicholas.ciesla@snhu.edu