# Maya Neely

# 2/3/2023

# CS 340 README Project 2

## About the Project

This web application will connect a client-side user interface to a database of animals that allows users to create, read, update and delete animals from the stored data.

## Motivation

Creating this project will help me to apply database systems, concepts and principles to develop a client/server application that interface client-side code with a database.

**Functionality**

This project produces a fully functional and user-friendly dashboard that is interactive; Allowing users to filter between different AAC outcomes with a pie chart and geolocation chart that dynamically respond to the filtering options.

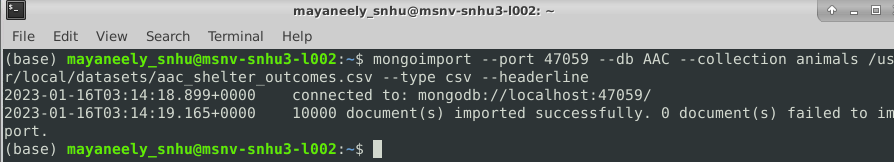
**Tools Used**

* MongoDB (<https://www.mongodb.com/>)
  + MongoDB was used as the model component of development by storing the projects database. Its efficiency and flexibility made MongoDB the most suitable tool to work with Python to develop this dashboard.
* Python (<https://www.python.org/>)
  + Python was used to create my CRUD file which enabled me to create, read, update and delete items within my database. This simple programming language was the middleware between MongoDB and Jupyter Notebook
* Jupyter Notebook (<https://jupyter.org/>)
  + Jupyter Notebook allowed for the creation of my .ipynb file which is ultimately the display of my dashboard and widgets
* Dash Enterprise (<https://plotly.com/dash/>)
  + Dash, combined with plotly, were the tools used to create and display my dashboard, pie graph and geolocation chart.

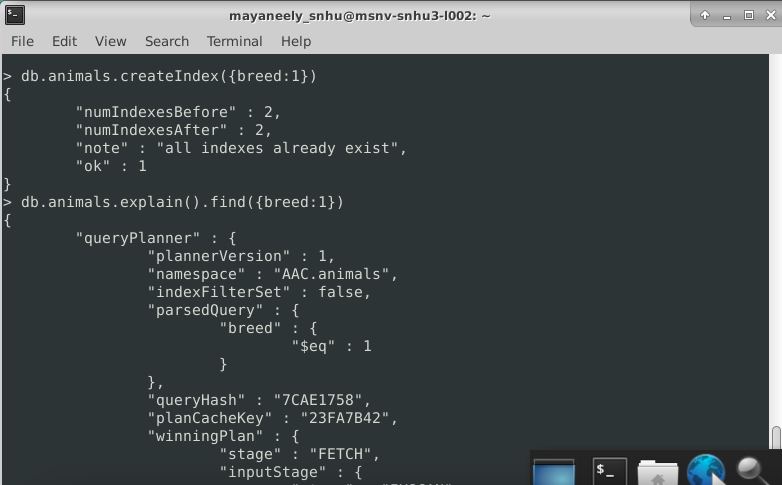
## Getting Started

To access the database,

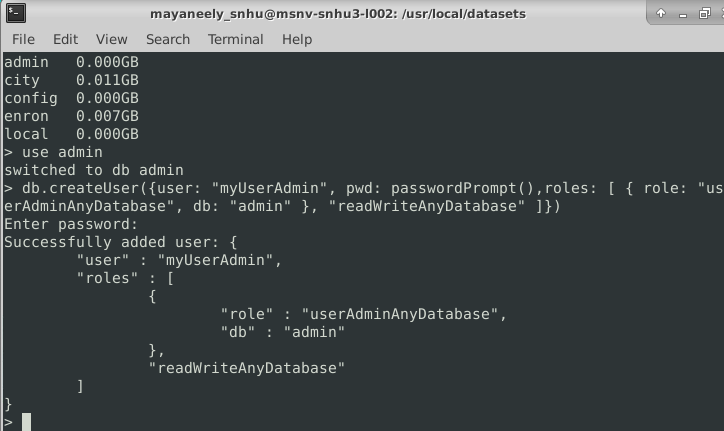
1. In the Linux shell, import the csv file aac\_shelter\_outvome.csv.

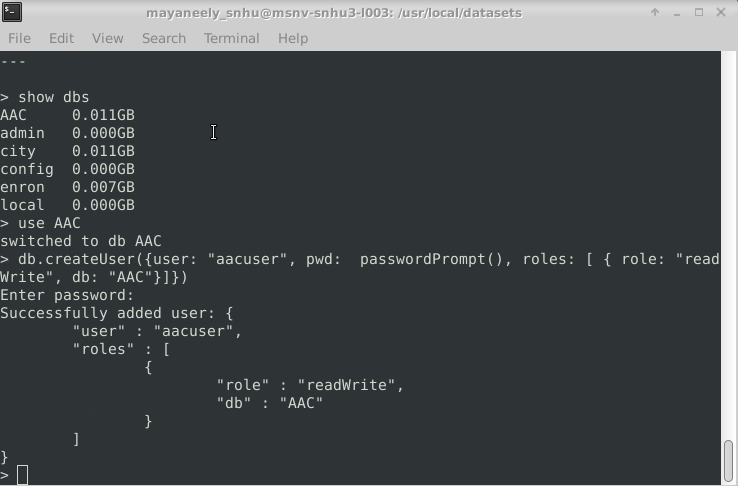


1. *Create a simple and compound index to test the data within the document*



1. *Create an admin and aacuser account to ensure user authentication to the database*



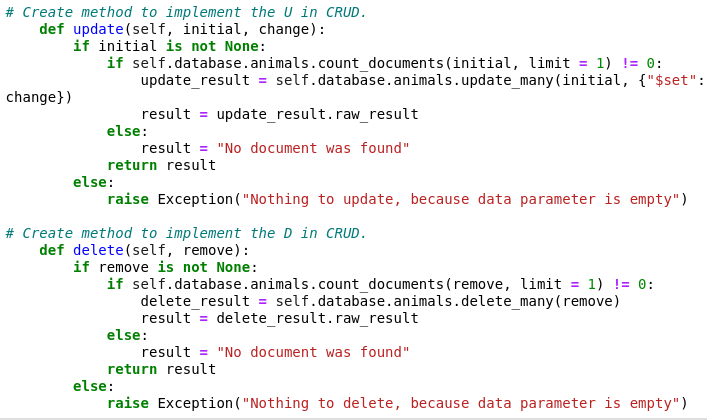


1. *Develop a python module to create and read functionality for the database*
2. *Create a Python testing script to call and test the create, read, update and delete instances*

### Code Example

Graphical user interface, text, application, email

Description automatically generated



## Screenshots of Dashboard

\*Please refer to the attached screen recording for a more accurate representation of my dashboard's functionality\*

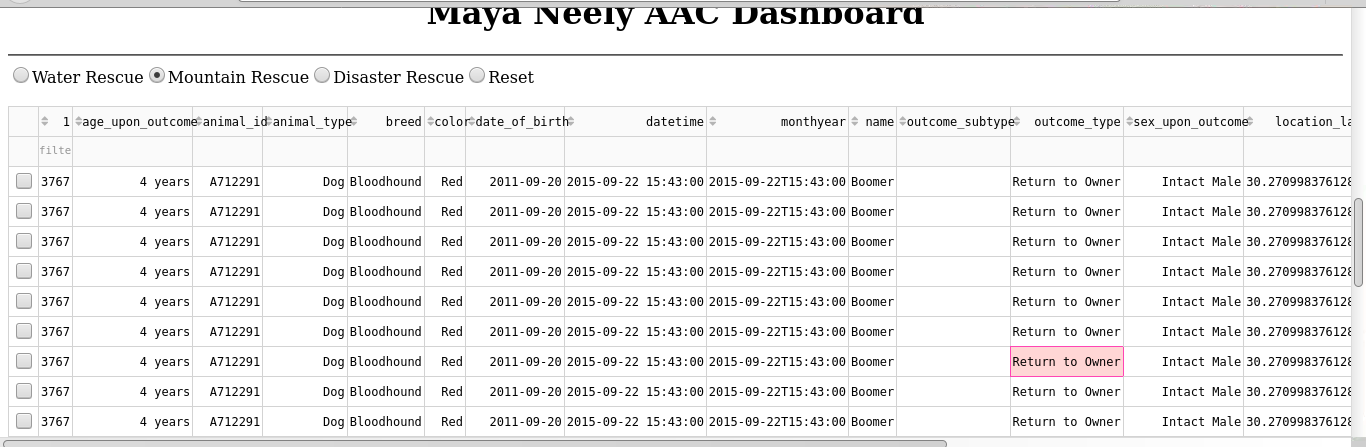
The initial screen shows the Grazioso Salvare logo and my unique identifier



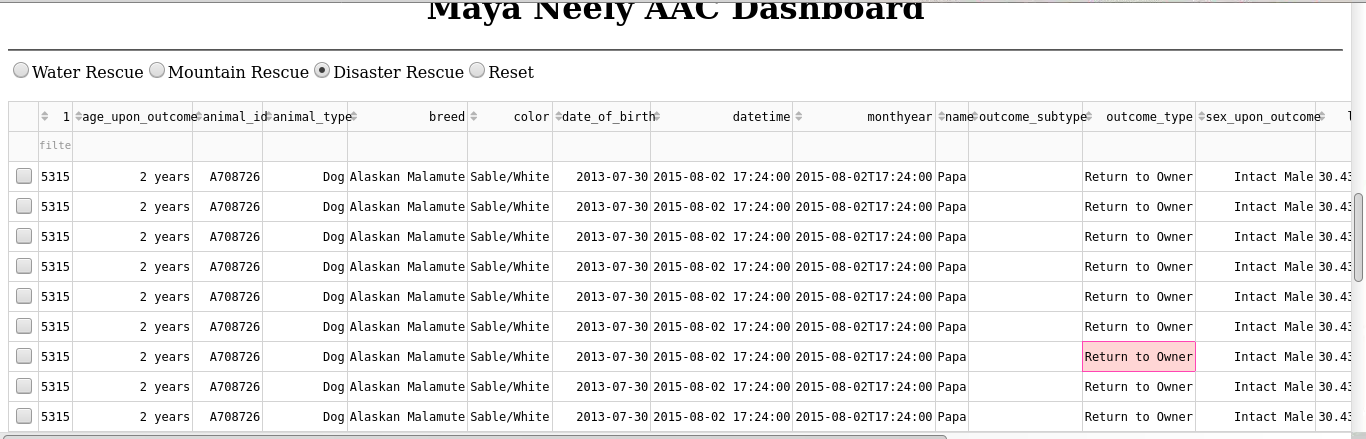
**WATER FILTER**



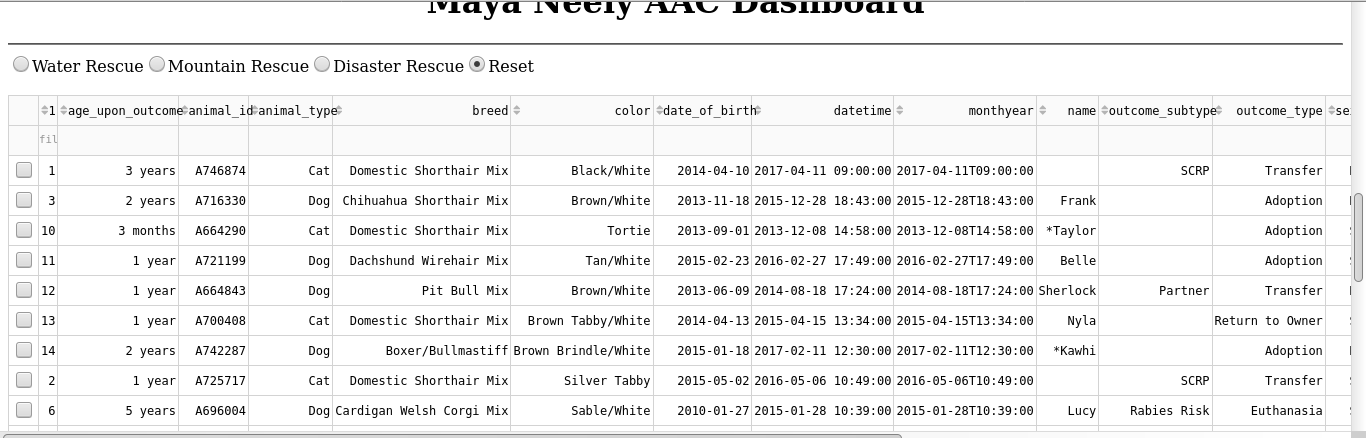
**MOUNTAIN FILTER**



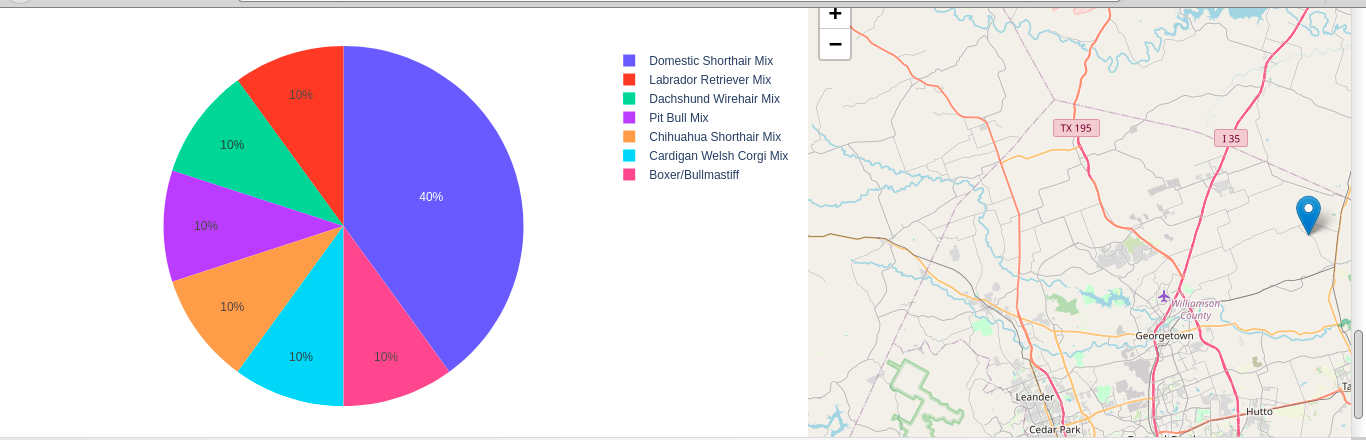
**DISASTER FILTER**



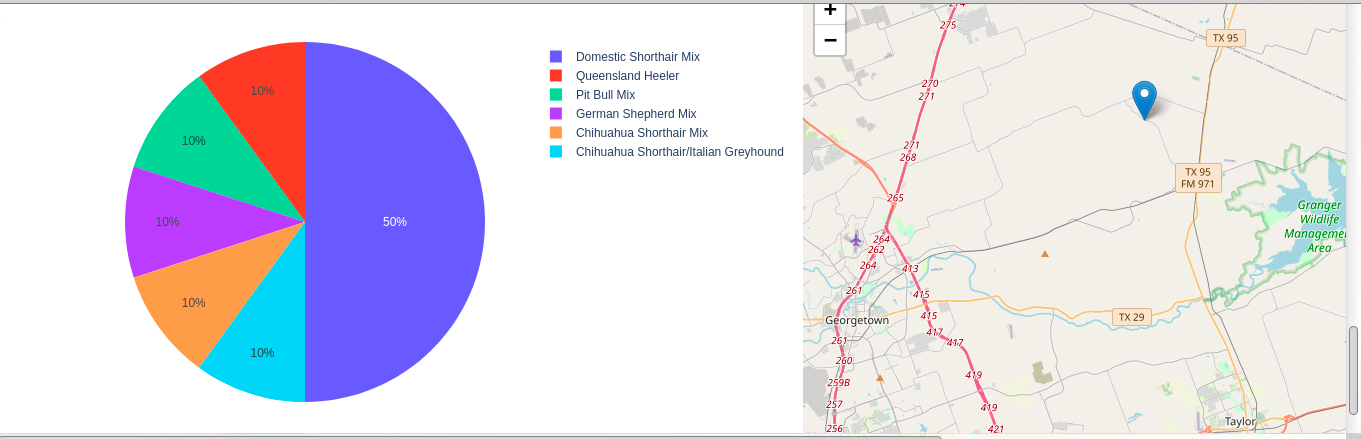
**RESET FILTER**



**Pie graph and map for Reset Filter**



**Pie graph and map for water filter**



## Challenges

I encountered many challenges during the past few weeks. The biggest challenge I had at first was displaying my data table. The simple tutorial provided in Module 6 to display a data table, would not work for me meaning there must have been an error in my CRUD.py even though this file accurately displayed my information in Module 5. With this being, I had to make changes to my CRUD file, essentially recreating the whole file. Another challenge I faced was that I was constantly getting error messages. After careful evaluation, I found that there were small errors in my code such as reiteration of functions and misspelling of words. Double checking my code solved this issue.

**Contact**

Maya Neely