# CS 340 README

## About the Project

This project will be able to implement a CRUD Python module in MongoDB to create, read, update, and delete data for a collection, “animals”, in a database “AAC.” This module will be able to take animal data from a file and input it into the intended database, also allowing the user to update the information when needed. In this project we use PyMongo as the driver, since python works well MongoDB. This is because Python has dictionaries that make it easy to query through the database and conduct data collection.

## Motivation

This Python module will be part of a web application that connects a client-side user interface (such as a dashboard) to a database, it will enable CRUD functionality for data connection. This module was created for easy data collection into a database and is intended to be very versatile with the usage of global variables. Additionally, though a specific database and collection are coded, they can be easily changed since they are global variables, for other databases if need be.

## Getting Started

Ensure you have the latest version of Python and MongoDB running on your computer in order to use this module. Follow the steps below:

1. Verify that you have the latest and least buggy Python version by typing the following commands on the terminal:
   1. Windows: python –version
   2. MacOS/Linux: python3 -V

This should display the latest version you have on your machine

1. If you have an outdated version of Python downloaded on your machine, download the latest version at <https://www.python.org/downloads>.
2. After updates, please verify again the steps from step 1, to ensure you are running the latest version.
3. Download MongoDB to create a database based on the provided code. Please follow the instructions from the [online MongoDB manual](https://www.mongodb.com/docs/manual/installation/).
4. Ensure you have all the databases and collections created into MongoDB. Below is an example import for a database with a .csv file that was used for this project:

mongoimport --username=”${MONGO\_USER}” \

--password=”${MONGO\_PASS}” --port=${MONGO\_PORT} \

--host=${MONGO\_HOST} --db AAC --collection animals \

--authenticationDatabase admin --type csv \

--file aac\_shelter\_outcomes.csv --headerline

1. To run the script type the following:

cd <the desired directory where you python module is>

python <name\_of\_file.py>

1. Ensure your new database and collections were created and imported your new files:

mongosh > show dbs > show collections > db.collection\_name.find({“name”:”file\_name”})

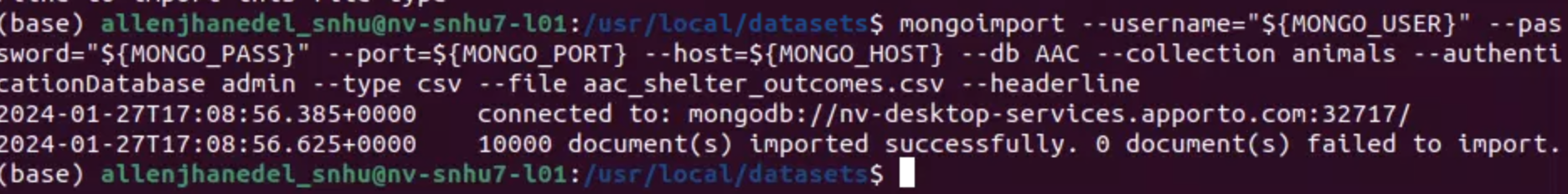
## \*See Installation Instructions for further details

## Installation

1. To install the latest version of Python on your machine, go [here](https://www.python.org/downloads).
2. To install the latest version of MongoDB on your machine, go [here](https://www.mongodb.com/docs/manual/installation/).
3. For a tutorial on how to get started with MongoDB, go [here](https://www.mongodb.com/docs/manual/tutorial/getting-started/).

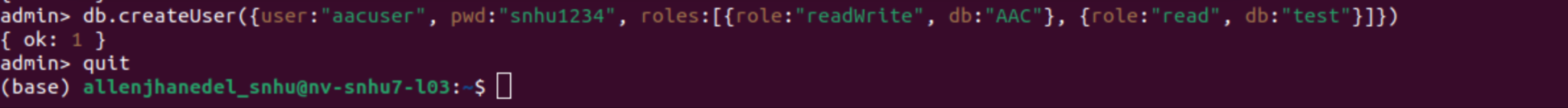
Before implementing CRUD for a database, we need to ensure that a database is created.

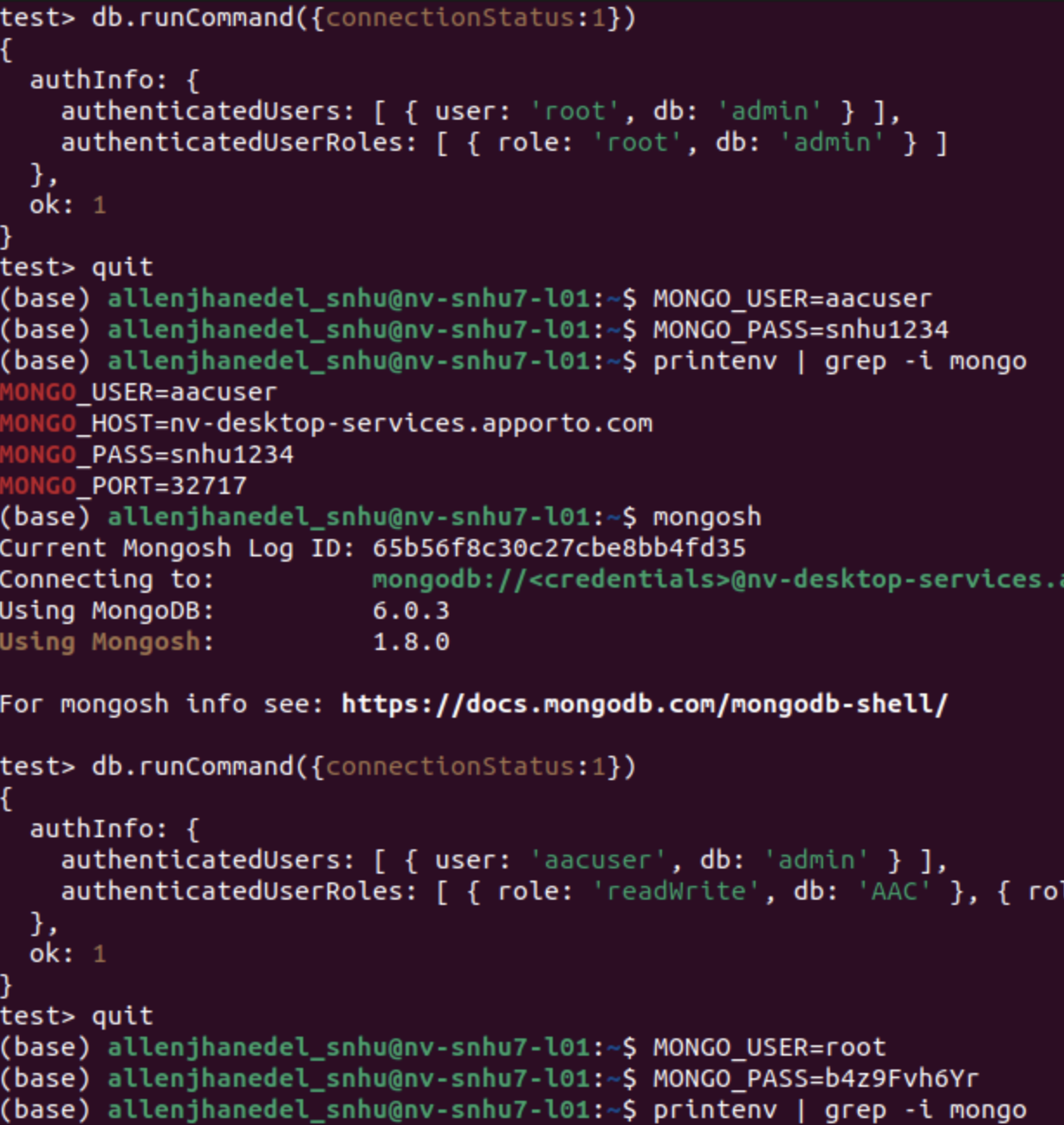
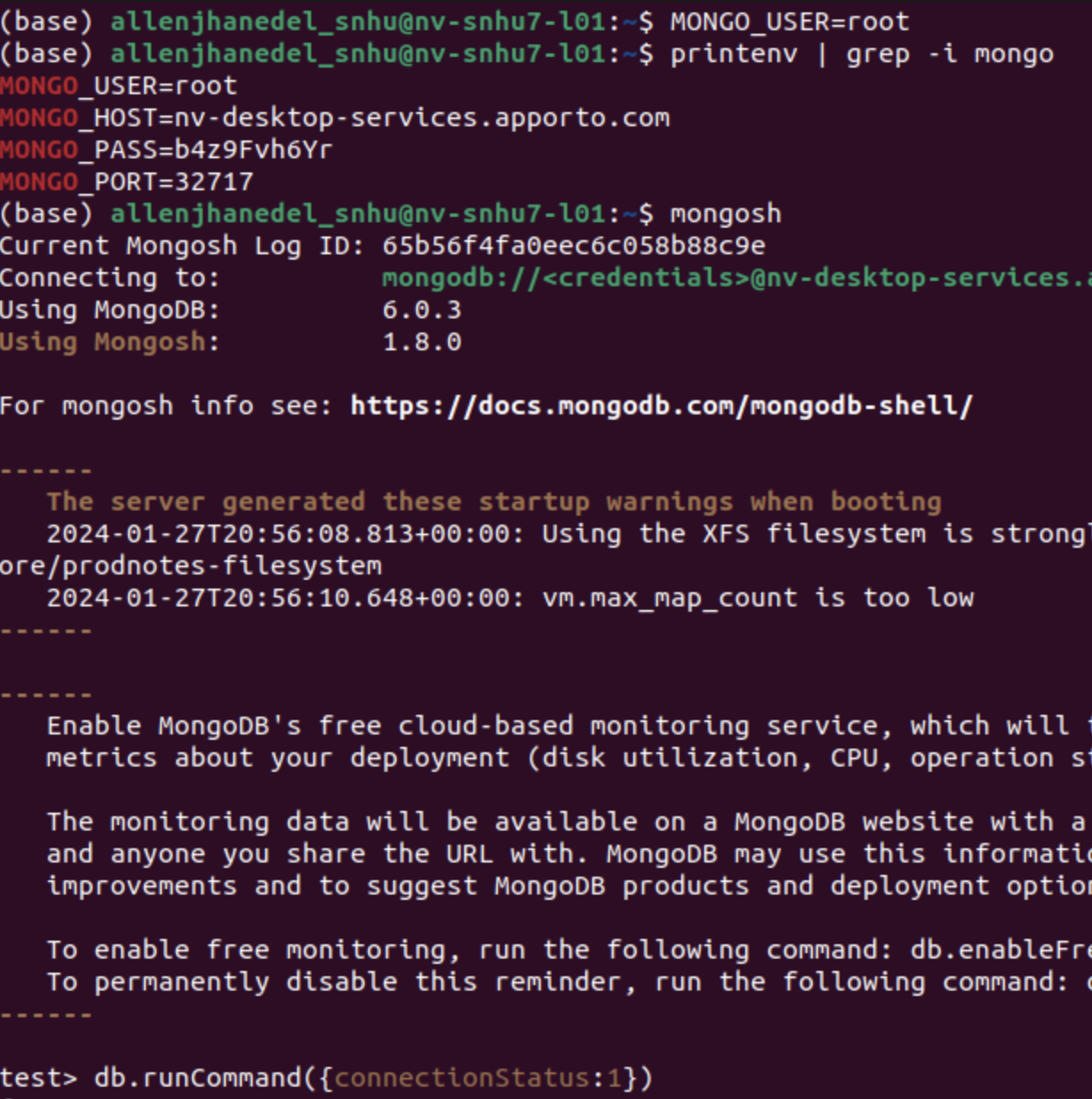
Below is an example of data import and database creation in MongoDB through the terminal:



After database creation, we need to create a user that is authorized to perform CRUD actions other than the default/root user.

Below is an example of creating a valid user for our use case and it’s confirmation:





 These screenshots are read left to right, top to bottom.

## Steps Taken to Complete This Project

In order to complete this application, we had two projects to work on over the course of the semester: the CRUD module and the Dashboard.

During the first half of the semester, we focused on learning how to use MongoDB by setting up the environment that would hold all the data from the input data file. We learned the basic requirements using mongosh and applied those skills towards uploading the data into the database and creating a user account that we would later need. Once we confirmed that our environment was set up correctly, we developed a CRUD module to create, read, update, and delete data from the database. For this CRUD module, we had to connect it to the database and provide login information once connected. By connecting to the database, we were able to directly edit the information in the database through the logged-in account. After confirming that all aspects of the CRUD module worked, we completed the first half of the application and moved on to creating the dashboard.

For the Dashboard, we first learned how to set up the environment using basic HTML tags through the Dash Python packages. We learned how to display and arrange the HTML components. Then, we learned how to connect to the database from the CRUD module and display the information in a table on the dashboard. Additionally, we learned how to add a map to the dashboard so we could visualize the locations of the animals based on database information. After completing the basic functionality of the dashboard application, we added widgets to make it easier for the user to filter through the database and visually see the data. After creating this, I ensured that all components worked correctly and displayed properly for the user so that it could be a deployable application.

## Challenges

When creating this project, I encountered many problems. The first issue I faced was connecting the CRUD to the MongoDB server. There were several issues with my connection, and I had to edit some of my code as it was causing a slight problem. However, after consulting with my instructor, we identified that I used the wrong port to connect to it. After replacing the port with the one I used in a previous module instead of the one in the template, the module worked, and the server was successfully connected to my code.

Later on, while trying to complete this project, I encountered a JSON serialization issue. My code couldn't convert the read information into a filtered table. Just as I was about to give up after hours of research, I came across a Plotly post that mentioned we had to remove the ID object at 0 since it's an object part of the list that cannot be serialized. After removing the object from the read columns list, the code worked perfectly!

I encountered some issues with displaying the dashboard. However, after some research, although tiring, I was able to display all components on the dashboard as I liked.

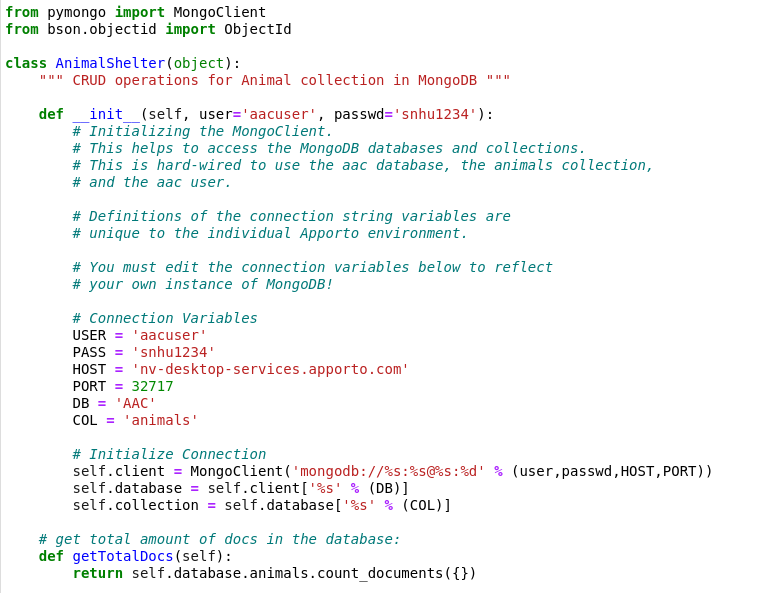
## Usage

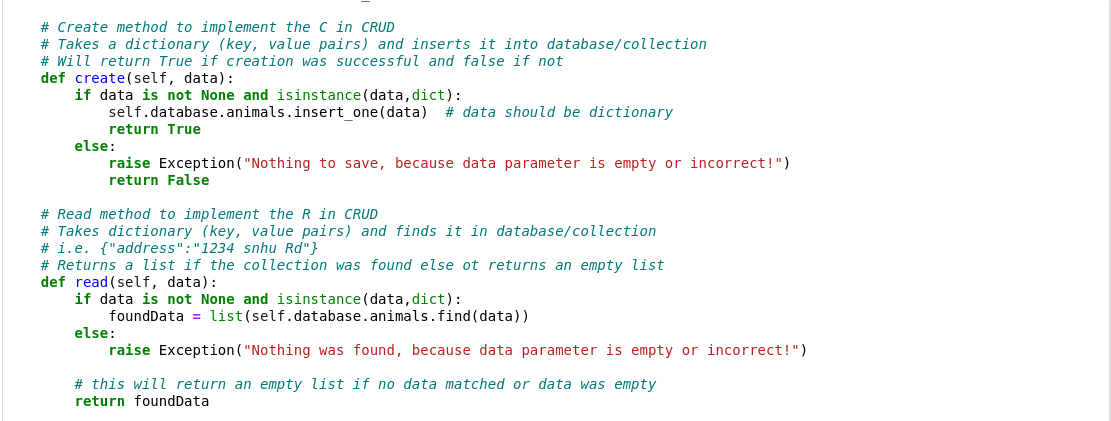
### Tools Used for CRUD Code

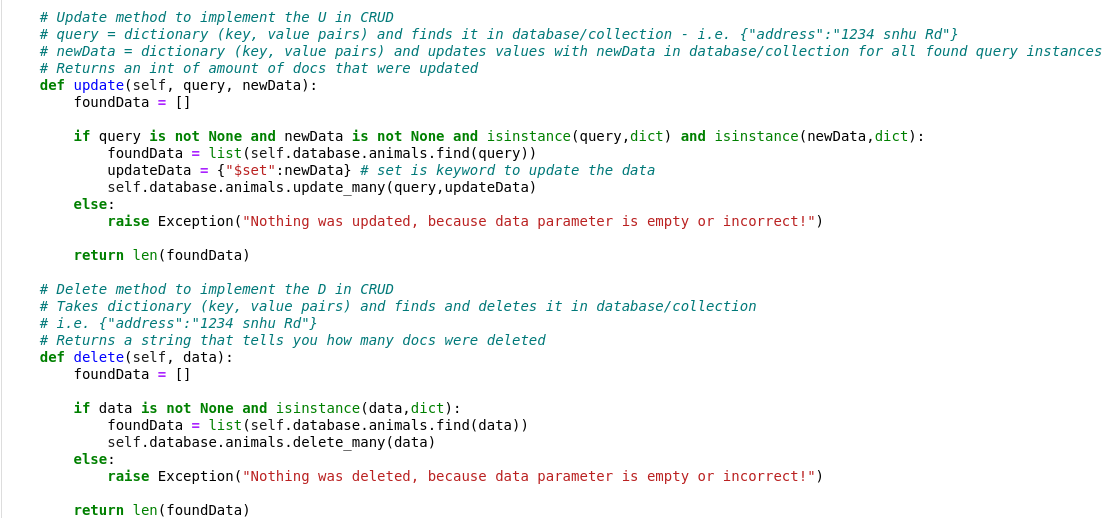
I used pymongo and bson in order to manipulate the database we created and connected to in mongosh. These packages allowed me to manipulate and create a CRUD module that can directly interface with the mongoDB system on out virtual machine

### CRUD Code Example

Below are screenshots of code that implements CRUD in order for a user to create, read, update, and delete data they choose as long as they are on an authorized account.

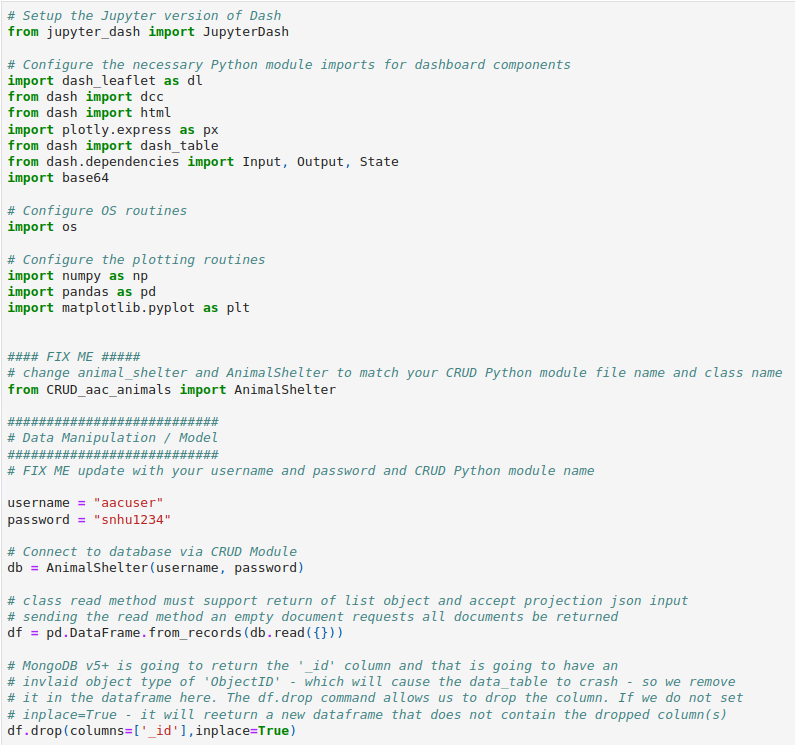
Screen shots of an AnimalShelter obj that will be used for CRUD:





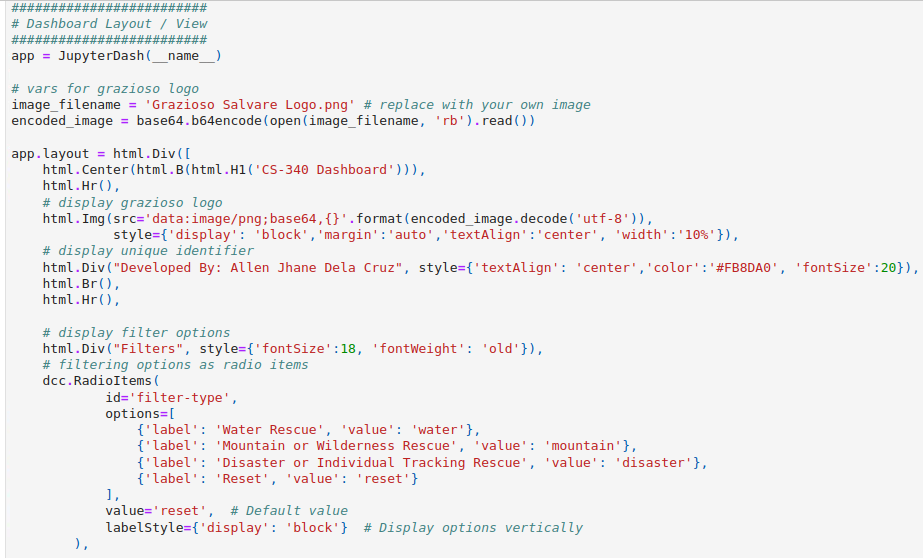
### Tools Used for Dashboard Code

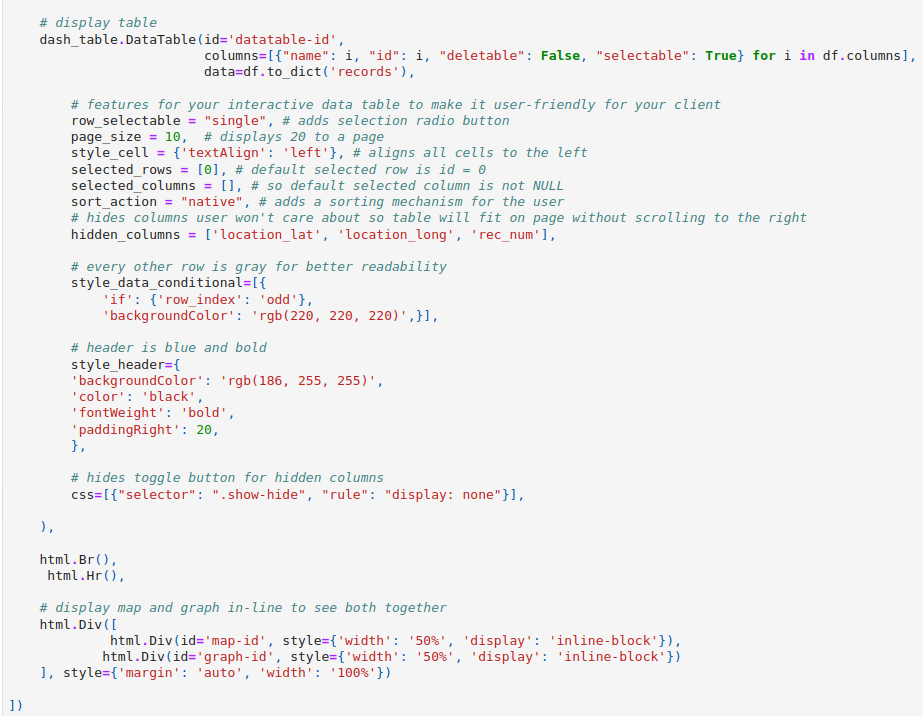
I used all the packages listed below. jupyter\_dash and all the dash components listed allowed me to created and display database information from our mongosh database onto a localhost website. Additionally, plot and pandas was used to display a graph and table onto the dashboard, respectively.



### Dashboard Code Example

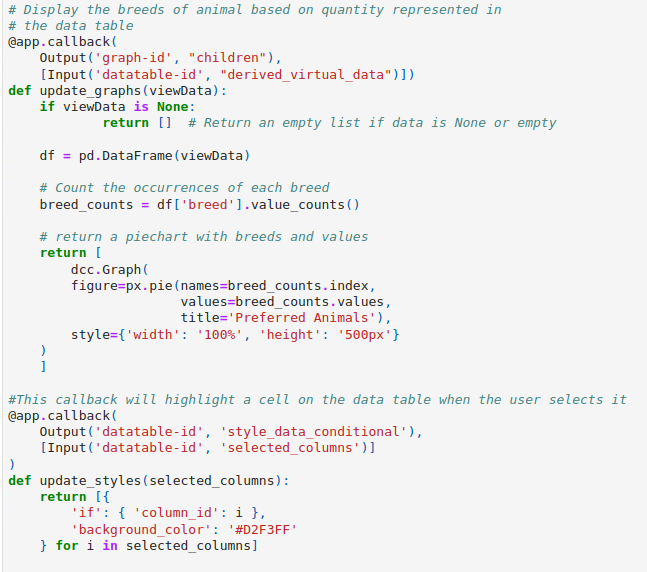
Below are screenshots of the code that was used to create the dashboard application that implements the CRUD module and provides filtering in a GUI for a user-friendly view.







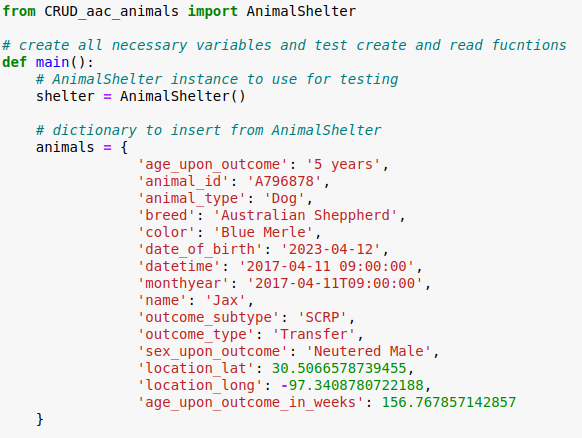


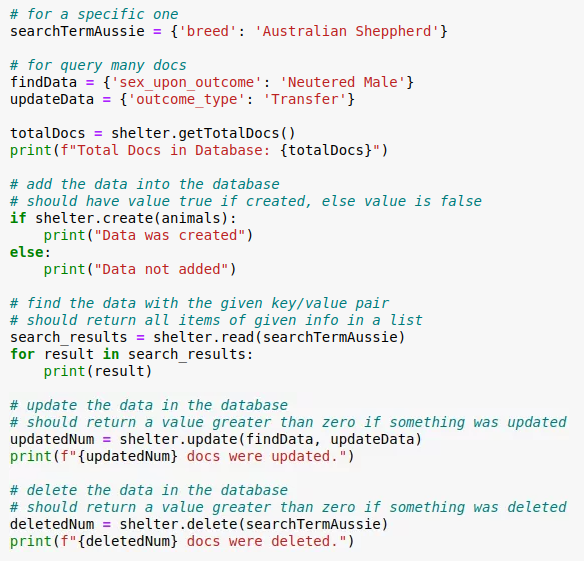




### Test CRUD obj AnimalShelter

Test class with sample data that shows if data was created, found, updated, or deleted.





Below are the results when the test was ran:



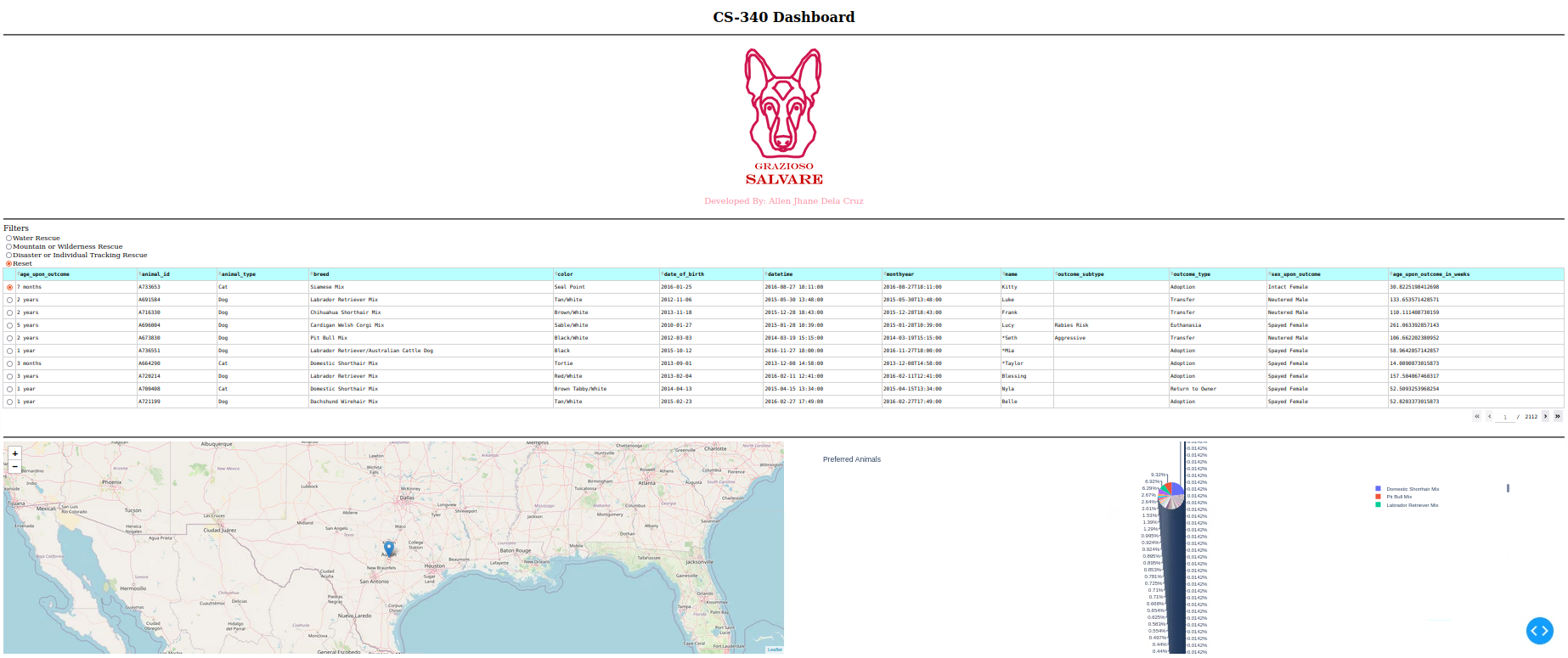
**Test Dashboard that implements CRUD**

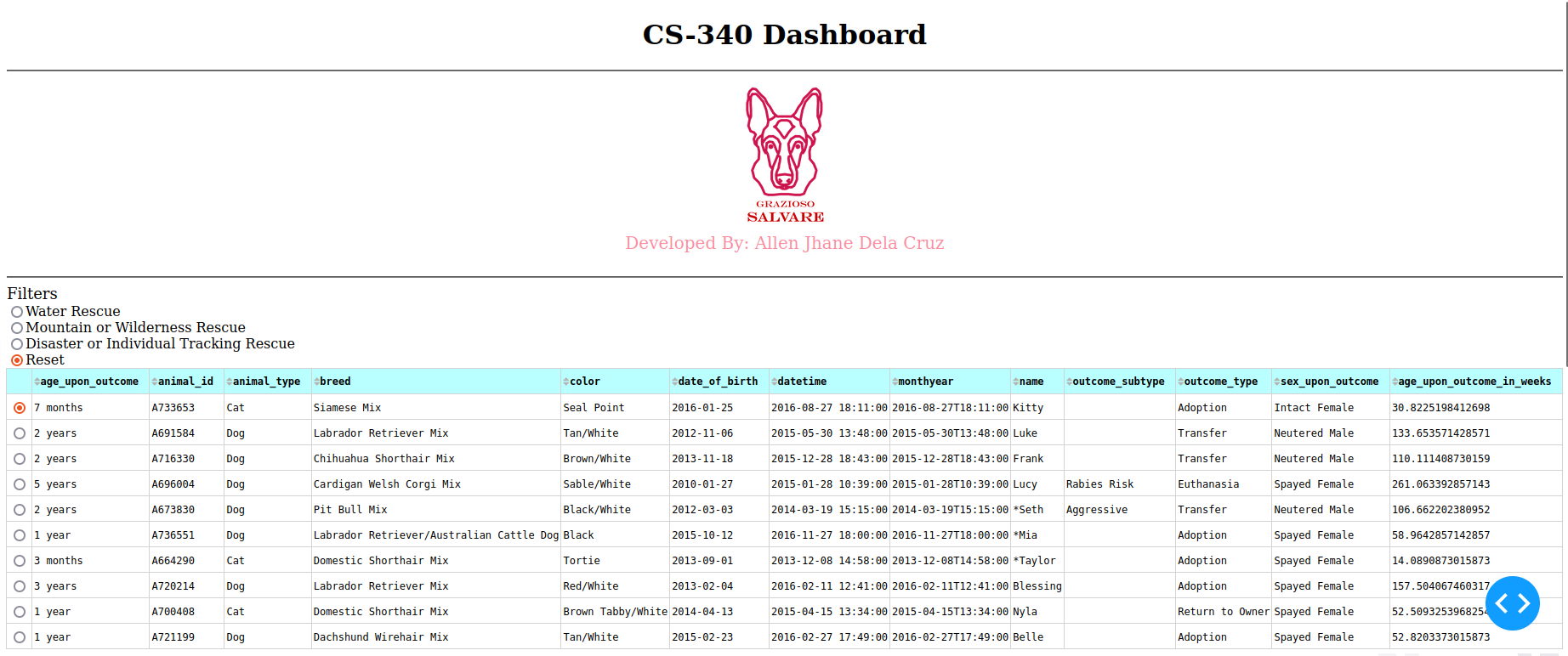
Below are the screencast and screenshots of filtering tests for the dashboard.

In the video it shows all filtering options from top to bottom.

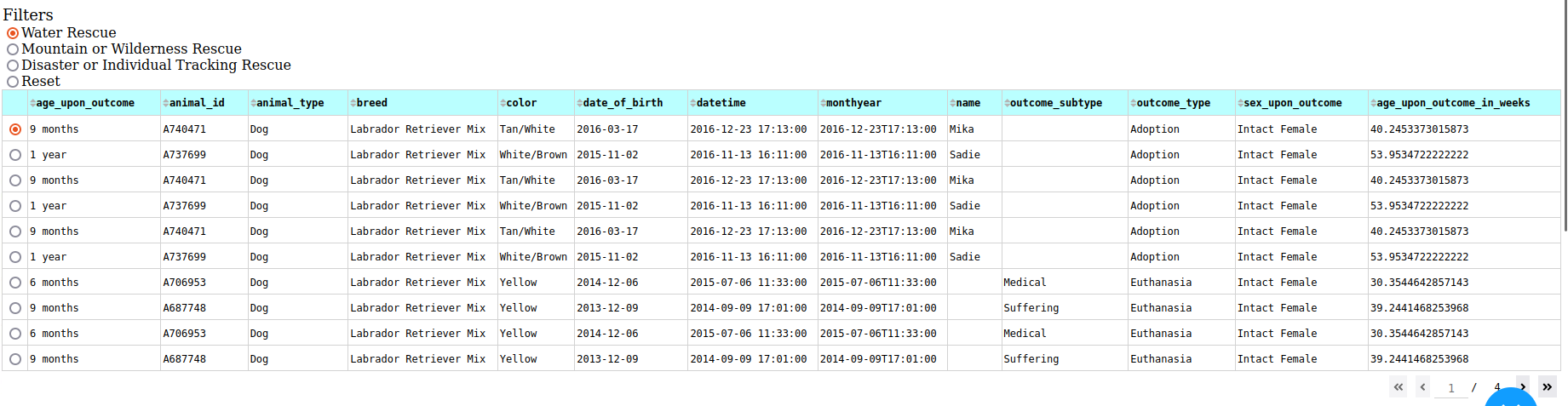
[Screencast from 02-25-2024 10:09:06 AM.webm](https://drive.google.com/file/d/1JuNCZDn8AjLX06gq-2gk1sU12j3q0VaH/view?usp=sharing)

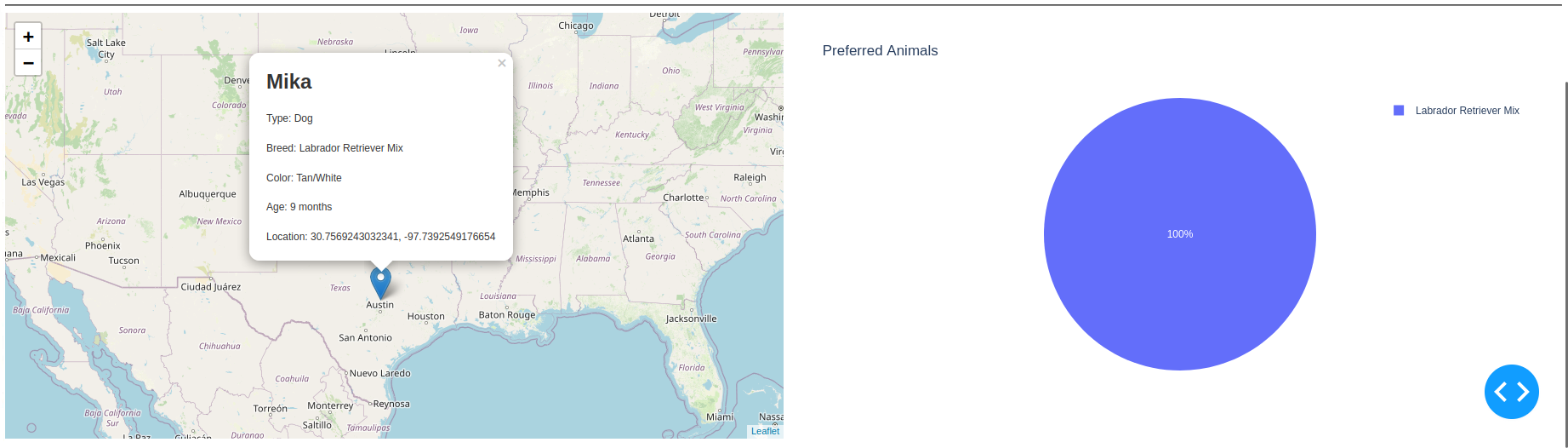
<https://drive.google.com/file/d/1JuNCZDn8AjLX06gq-2gk1sU12j3q0VaH/view?usp=sharing>



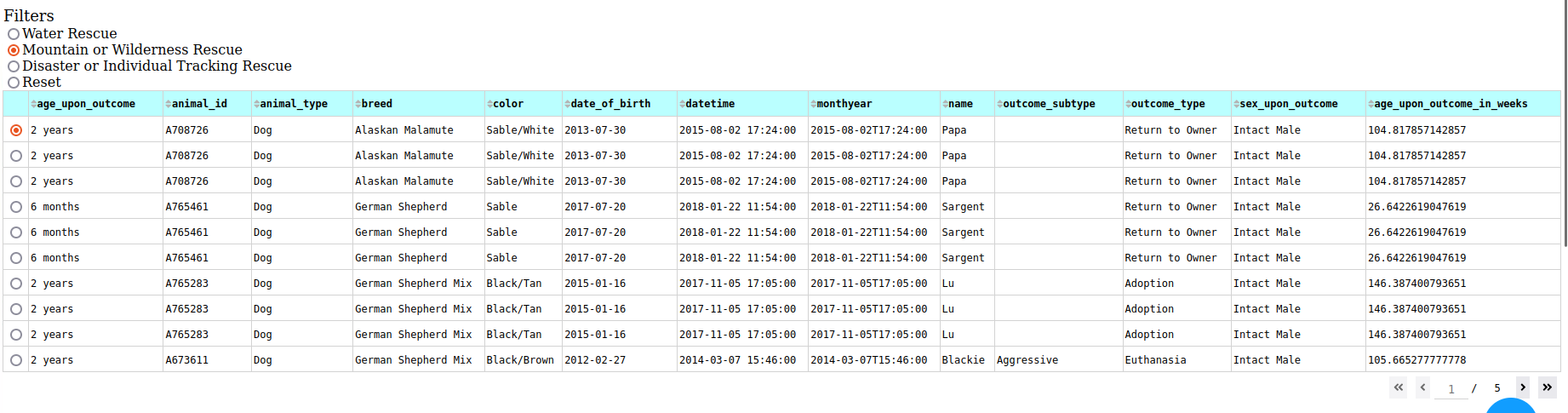


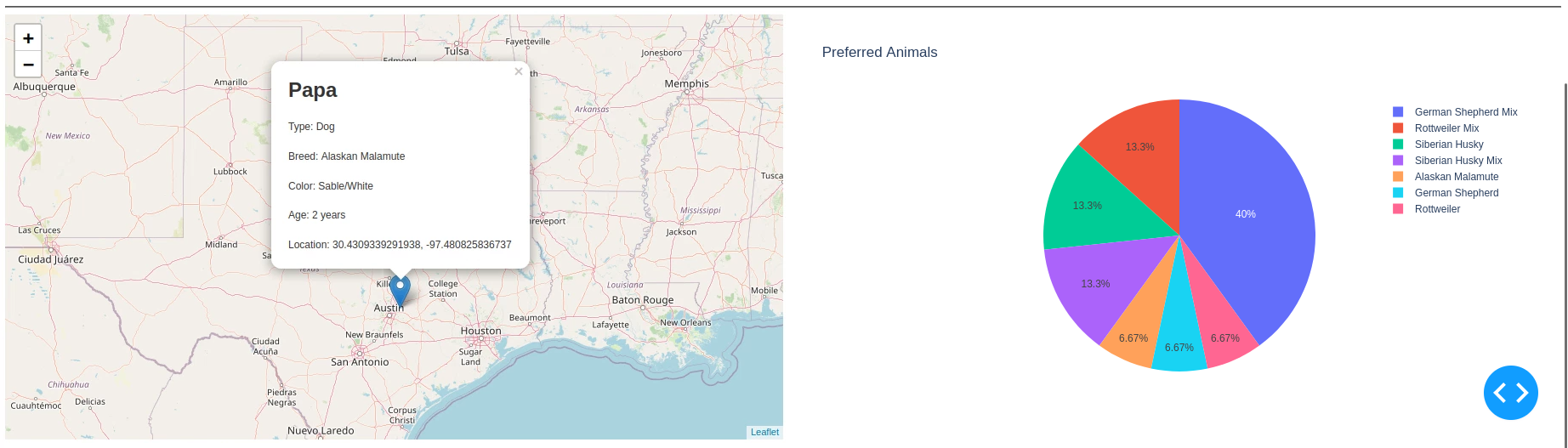
**Filter Water Rescue Dogs**



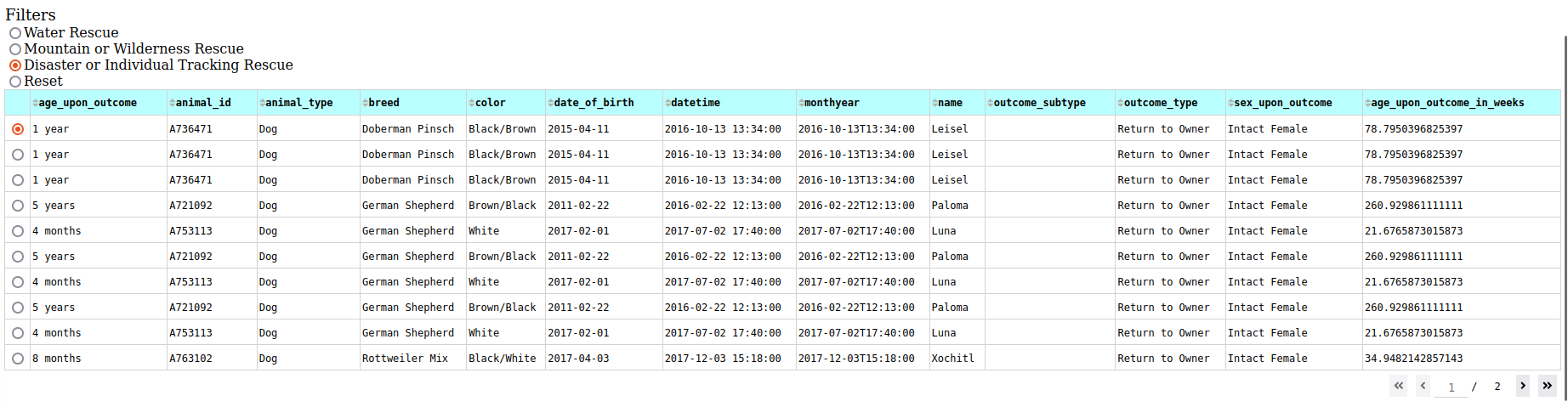


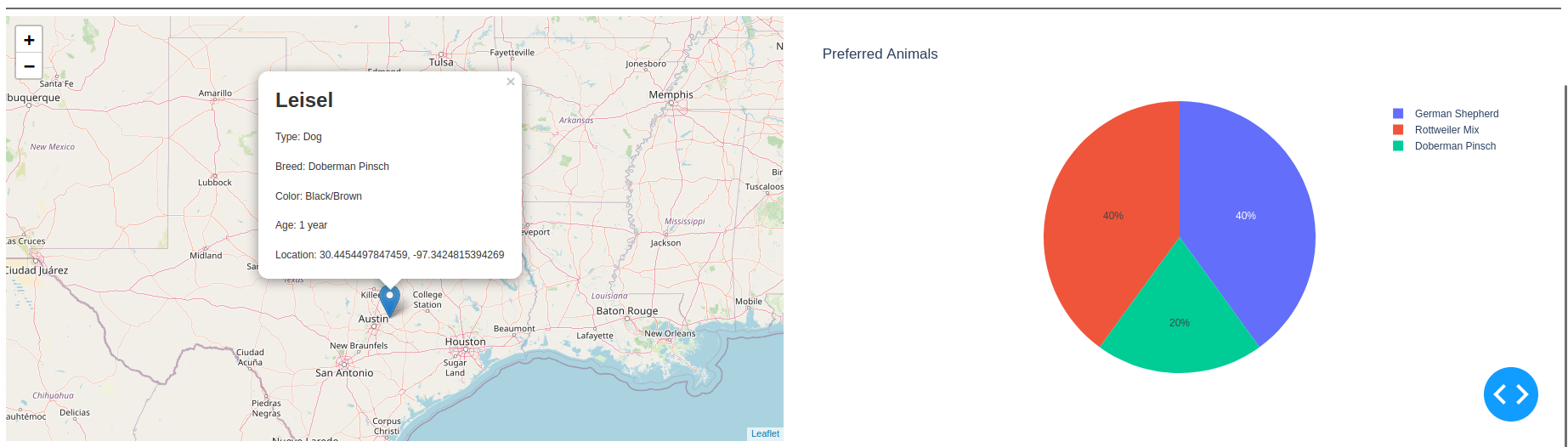
**Filter Mountain/Wilderness Rescue Dogs**





**Filter Disaster/Individual Tracking Rescue Dogs**





## Contact

Your name: Allen Jhane Dela Cruz