# CS 340 README – Ryan Hubbuck

## About the Project/Project Title

This project works with an animal shelter database to identify and categorize available dogs through indexing for optimizing queries, authentication for database security, interactive dashboard creation, data visualization, and Create-Read-Update-Delete (CRUD) actions in MongoDB via Pymongo. My client, Grazioso Salvare, identifies dogs for search-and-rescue training. Grazioso Salvare is seeking a software application that can work with existing data from animal shelters to identify and categorize available dogs, provide data visualization of various animal categories, and include a geolocation feature to plot individual animals on a map. Finally, the dashboard allows users to filter animals by three main categories: Water Rescue, Mountain or Wilderness Rescue, and Disaster or Individual Tracking. The dashboard will be able to filter animals by these three categories and display a pie chart for representations of the demographics within that group.

## Motivation

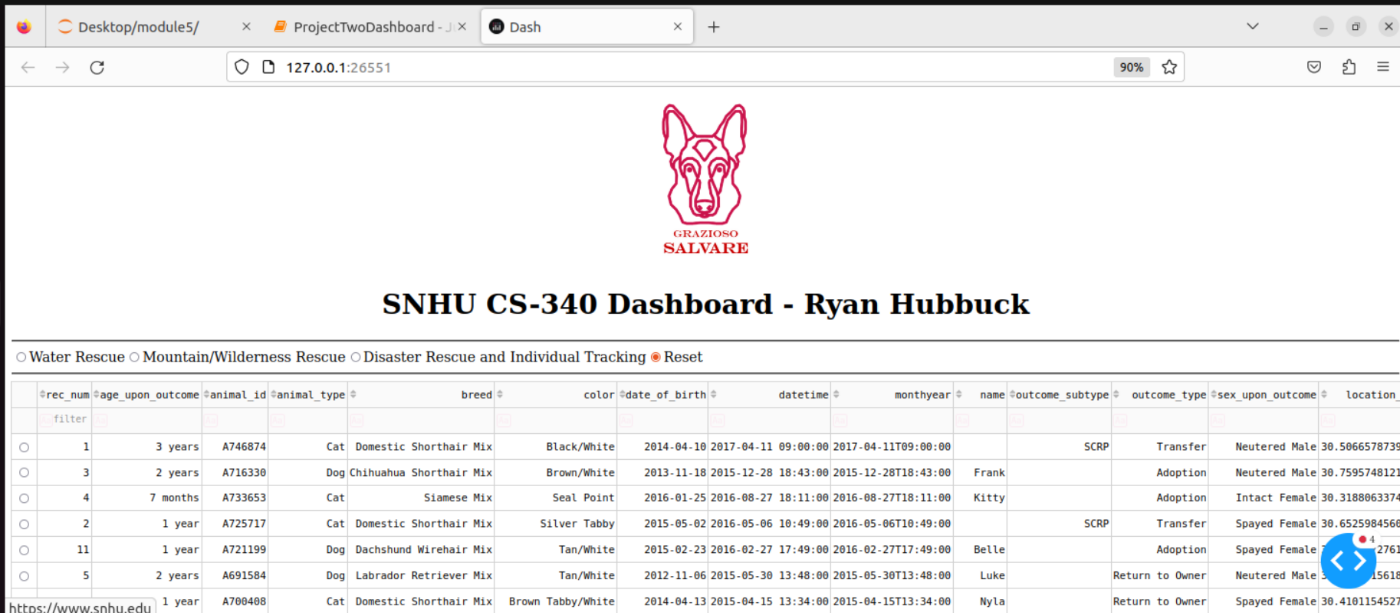
This program was designed to test my skill working with databases as a part of client/server development. To streamline the process of creating, reading, updating, and deleting data with each query, these operations have been abstracted to an AnimalShelter class with associated methods. The class ‘AnimalShelter’ will contain all the abstracted getter and setter methods needed for CRUD operations and MongoDB authentication. Additionally, Dash is used to test my skills in building an interactive dashboard that allows users to interact with, filter, and visualize data.

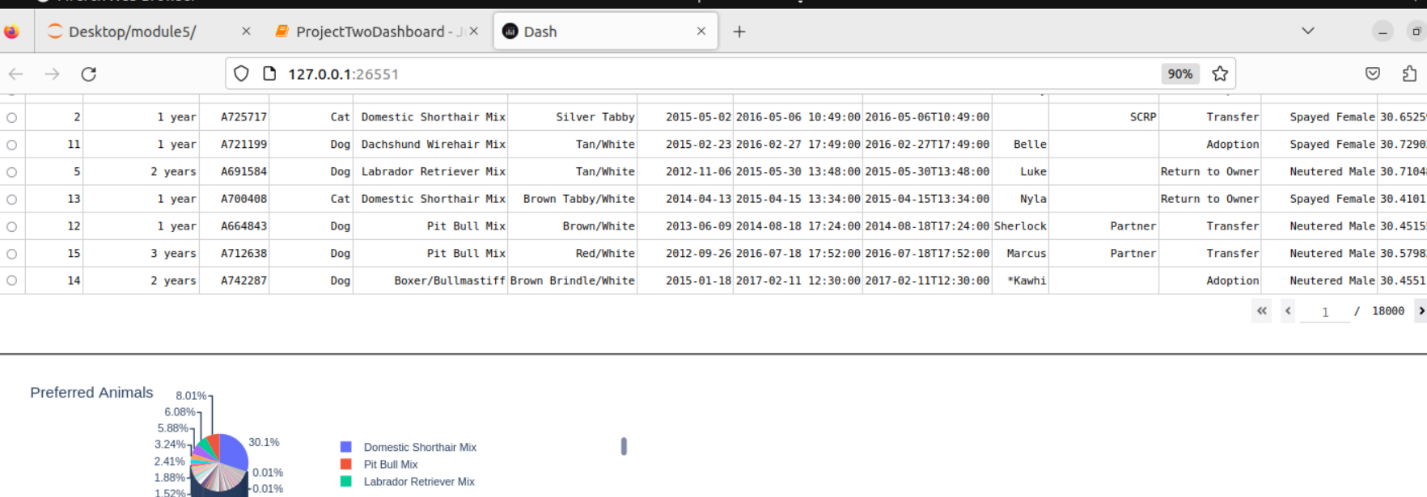
## Required Functionality

The first screenshot below shows the default dashboard. One can see the image/logo at the top that is a link to the SNHU homepage per the user’s request. Below the image is the title with my name included, followed by radio buttons that allow for data filtering. The table below is interactive. Selecting a specific animal will render and populate the geolocation feature, and the table’s contents are updated each time the table is filtered in a different manner. Finally, the pie chart in the lower-left corner displays demographics for the specified category.

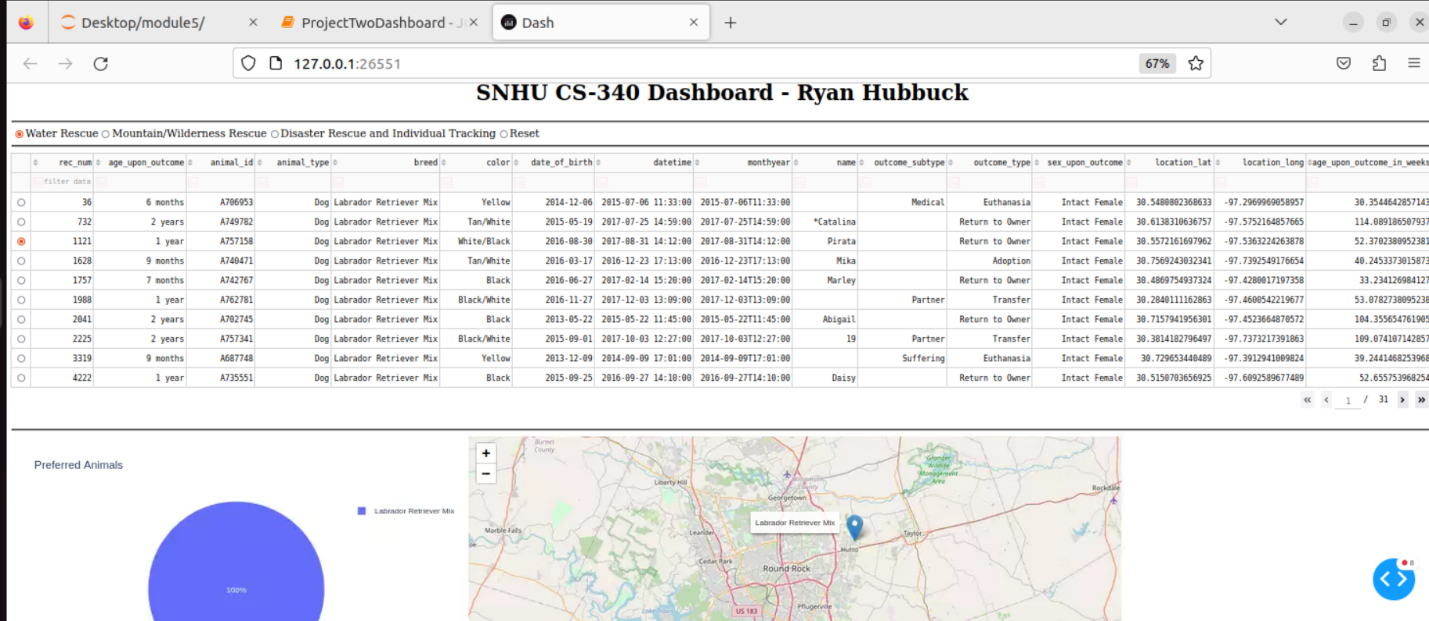
Note: the default category contains so many different breeds that the chart struggled to label each section in an attractive fashion, but selecting one of the three subcategories leads to a chart that is easier to digest.

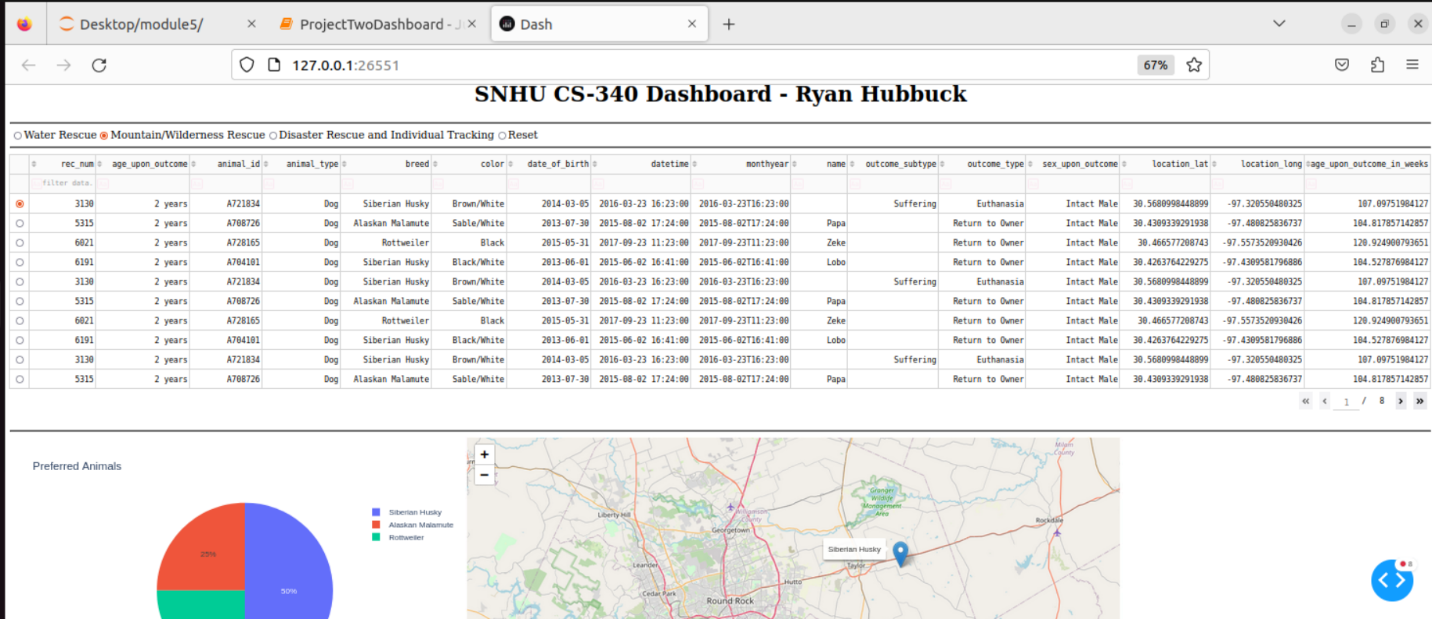
Note: Only the first screenshot includes the logo because I could not get all aspects into a single screenshot, but I made sure to include the header with my name in all of them.

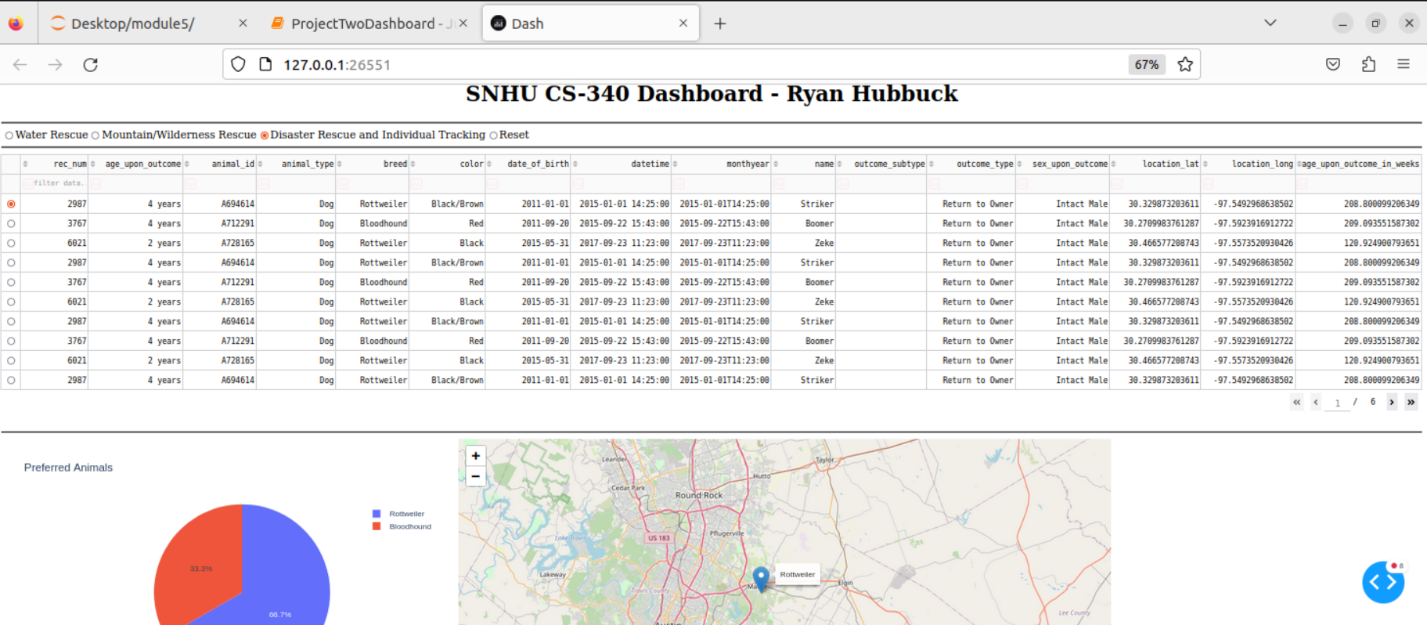


Th

The screenshot above shows the default, unfiltered pie chart without the selection of any animal to render the geolocation feature.







The three screenshots above show the page and chart for each category of animals.

## Tools Used

Python: The programming language this project is written in. Python is a popular language for data manipulation and visualization, as there are many useful packages used in this project like pyplot, numpy, pandas, and plotly. These packages are open-source and free to develop with.

Dash: Framework used to incorporate web-based features and build html components of the dashboard. Callback functions are utilized to ensure proper data transference and functionality. MVC is used which means that the logic is contained within a controller that chooses which view to display to a user.

Jupyter Notebook: Popular IDE for Python development.

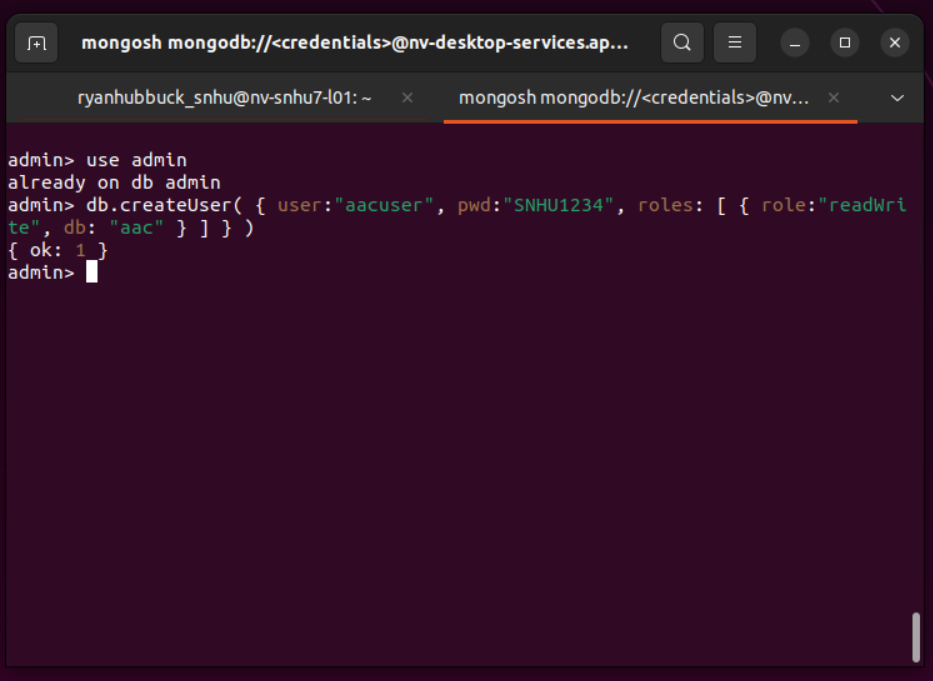
MongoDB: A unique database system that stores data in a JSON-like format. Data manipulation within MondoDB is written using calls like JavaScript and is very effective and intuitive. MondoDB is a scalable database architecture and allows the storage of structured or unstructured data. MongoDB interacts with Python quite well using packages like PyMongo. These can be used together to build and scale applications quickly.

## Steps to Complete the Project

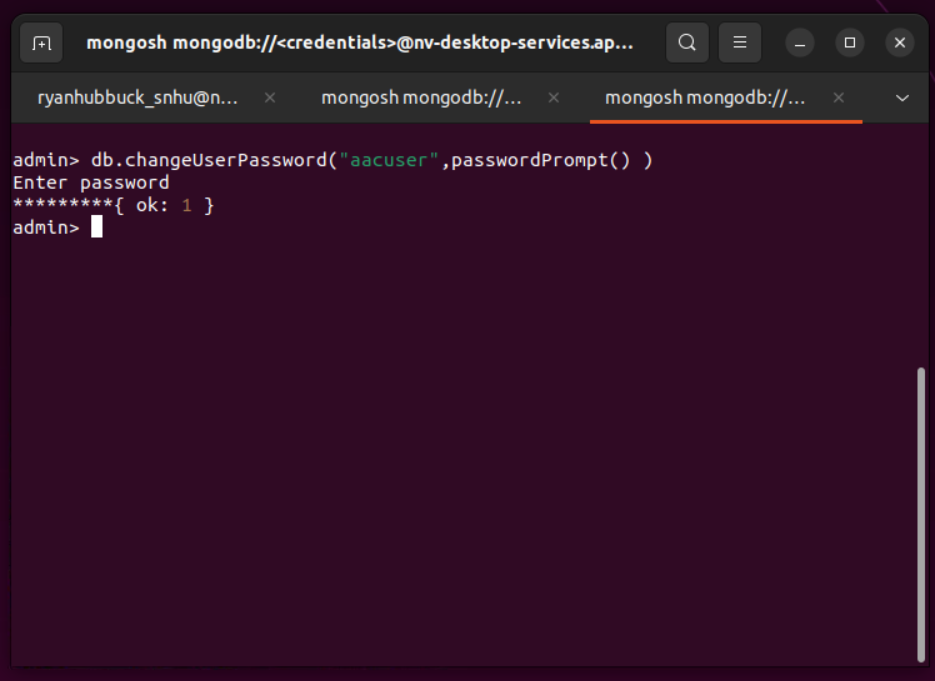
To get this program started:  
Enter a linux terminal and import the csv file aac\_shelter\_outvome.csv.

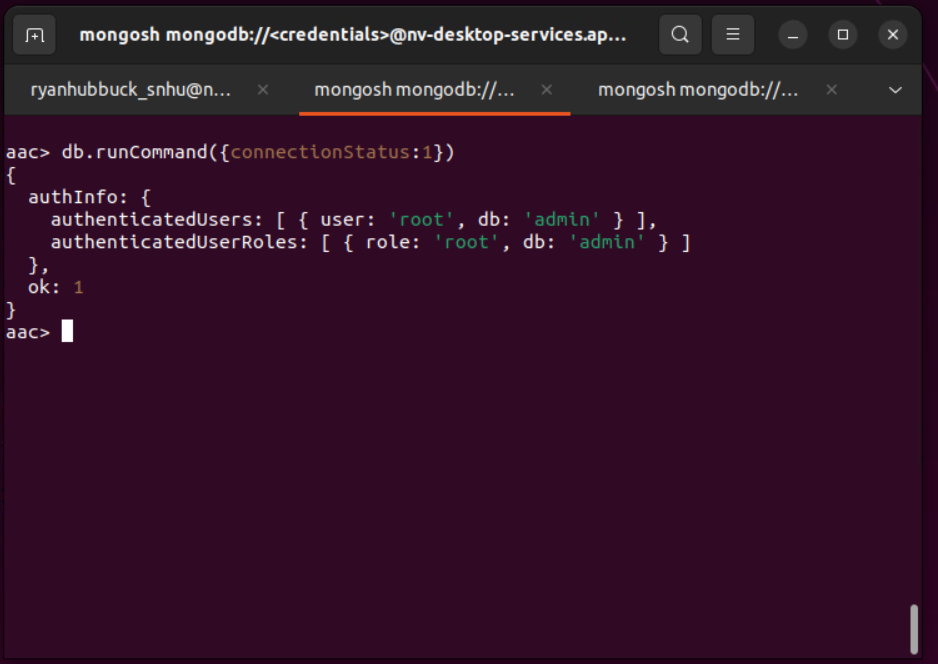


Create both an Admin account and an aacuser account to access the database.





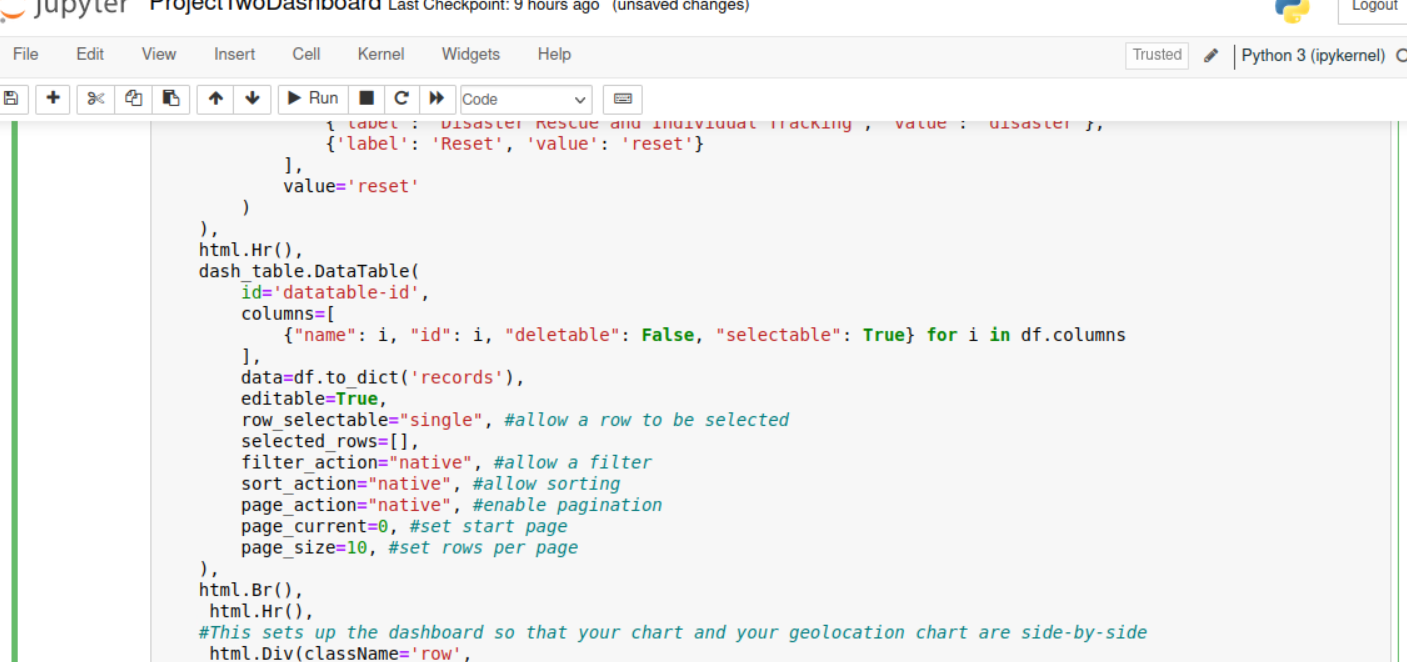


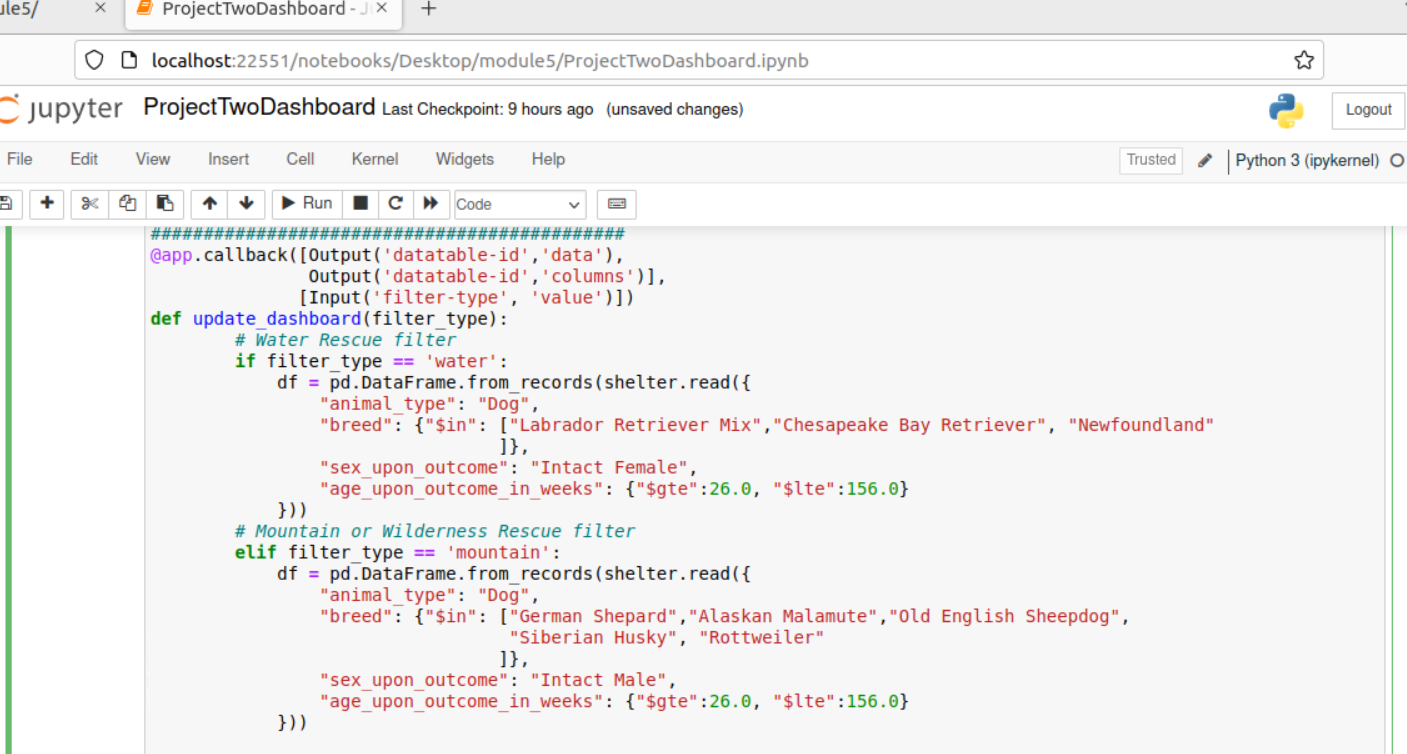


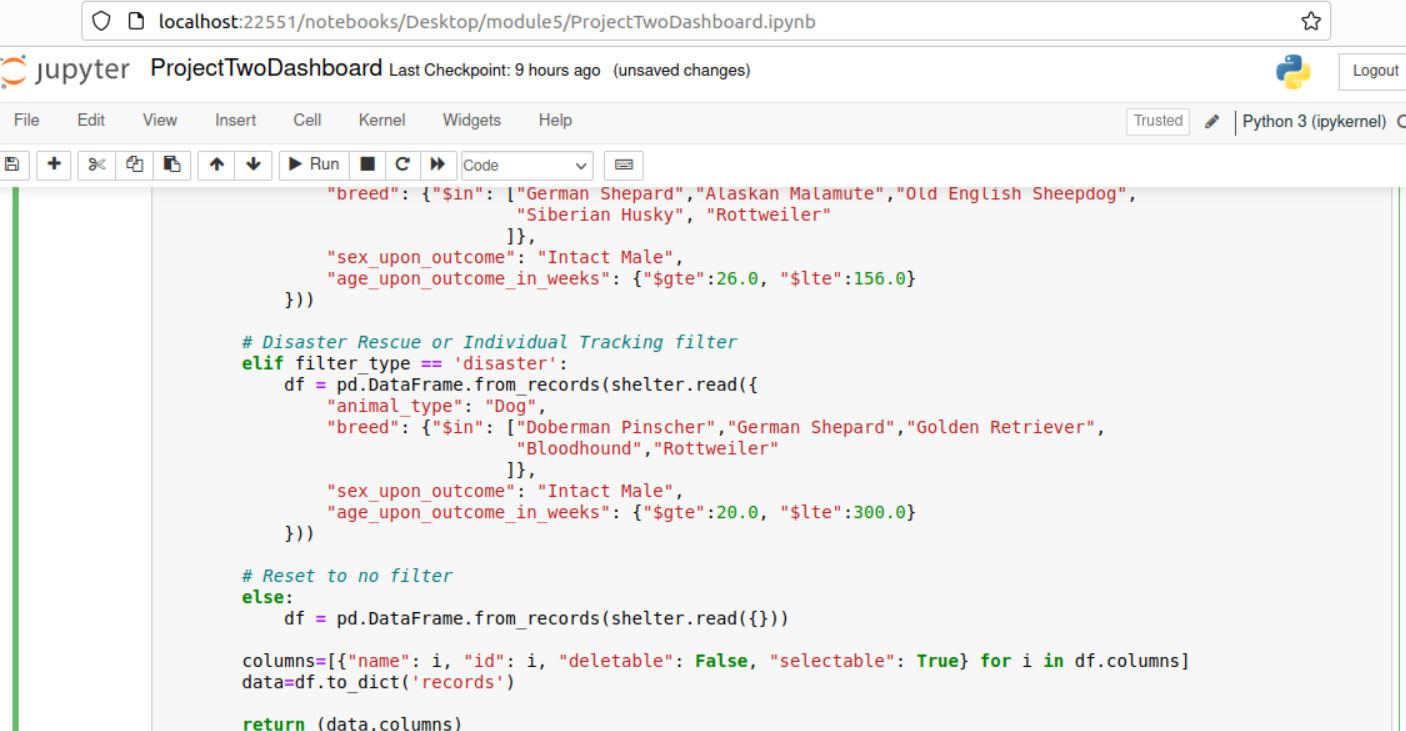
Import dependencies and CRUD file, then build out the functionality for the table, graph, styles, and geolocation within Dash Controllers.



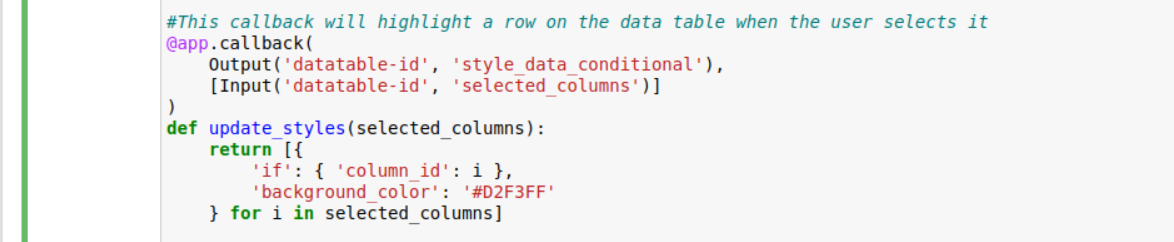
The above screenshot shows the radio buttons.



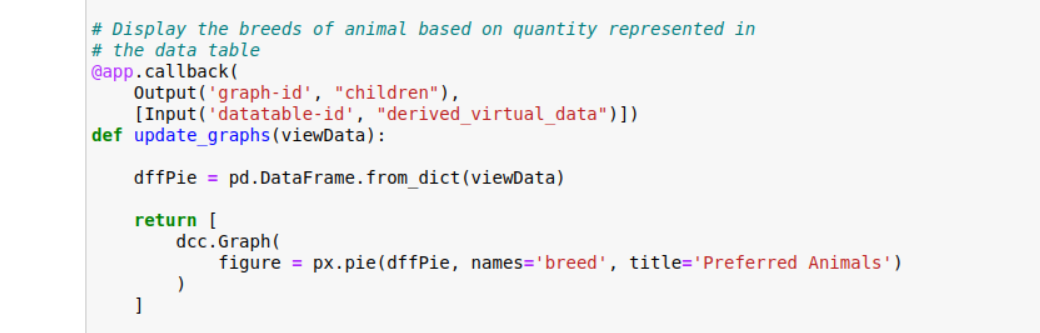




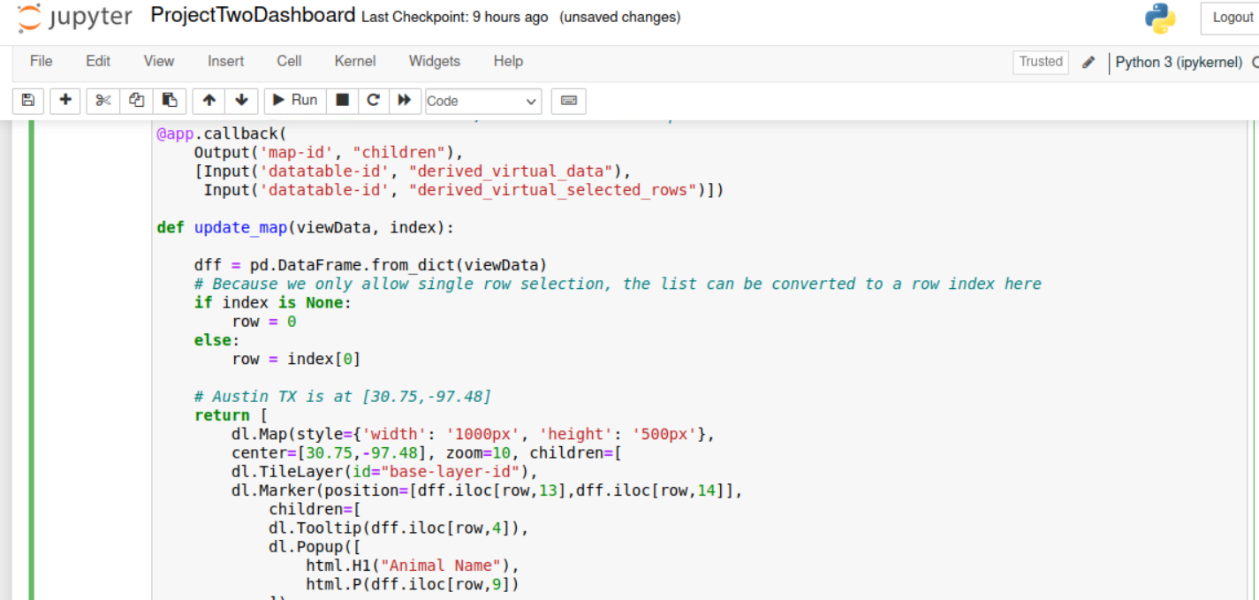
The two screenshots above show the table filtering controller.



The screenshot above shows the table styling controller.



The screenshot above shows the graph controller.



The screenshot above shows the map controller.

## Contact

Your name: Ryan Hubbuck