**Grazioso Salvare Dashboard README CS-340**

**Project Overview**

This project provides Grazioso Salvare with a fully interactive, web-based dashboard to streamline the identification of animals from the Austin Animal Shelter for a variety of rescue operations. The dashboard utilizes MongoDB, Python, and Dash visualization tools for a well-rounded user experience.

**Motivation**

The purpose of this dashboard is to provide an interactive tool for analyzing animal shelter data. This project is meant to emphasize database management and visualization techniques to real-world scenarios. By filtering and visualizing rescue animals based on specific criteria like rescue type or location, the dashboard allows for efficient data exploration. The ultimate goal is to showcase the ability to create a user-friendly application that bridges data retrieval and meaningful insights, while adhering to professional design principles.

**Key Features**

1. Brand Recognition

* Grazioso Salvare’s logo is prominently displayed at the top of the dashboard.

1. Custom Filters Based on Client Needs

User can apply filters to target animals that fit Grazioso Salvare’s rescue categories:

* Water Rescue: Dogs aged 26-156 weeks, breeds like Labrador Retriever Mix.
* Mountain Rescue: Dogs aged 26-156 weeks, breeds like German Shepherd, Siberian Husky.
* Disaster Rescue: Dogs aged 20-300 weeks, breeds like Doberman Pinscher, Golden Retriever.
* All: Displays all animals.

1. Interactive Data Table

* Displays the filtered results in a clean, interactive table.
* Features include sorting, inline filtering, and pagination.

1. Visualization via Pie Chart

* Displays breed distribution based on the filter.
* In the **"**All**"** filter, breeds representing less than 1% are grouped under “Other” for better visualization (less slices, more visibility).

1. Interactive Geolocation Map

* Displays location of the selected animal from table.
* The default map center is the Austin Animal Shelter if no animal is selected.

**Technologies Used**

1. Mongodb : Stores animal data in a flexible, JSON format.
2. Python - Pymongo and Custom CRUD Class: Handles database interaction for querying and filtering data.
3. Visualization - Dash & Plotly: Dash for creating the dashboard components, Plotly for generating dynamic charts, Dash Leaflet for the geolocation map.
4. Additional Imported Libraries: Numpy and Pandas for data manipulation. Matplotlib for additional plotting support.

**Code Contains**

1. Custom Filters Logic: Filters are implemented using MongoDB queries with specific criteria for breed, age, and sex. The “All” filter retrieves all records without restrictions.

2. Pie Chart Customization: For the "All" filter, breed counts below 1% are grouped under "Other" to improve chart readability.

3. Geolocation Map: The map updates to show the location of a selected animal based on its latitude and longitude. If no selection is made, the default location, Austin Animal Shelter, is displayed.

4. Company Logo Integration: The company logo is encoded in base64 and rendered at the top of the dashboard.

**Installation Requirements**

1. Prerequisites:

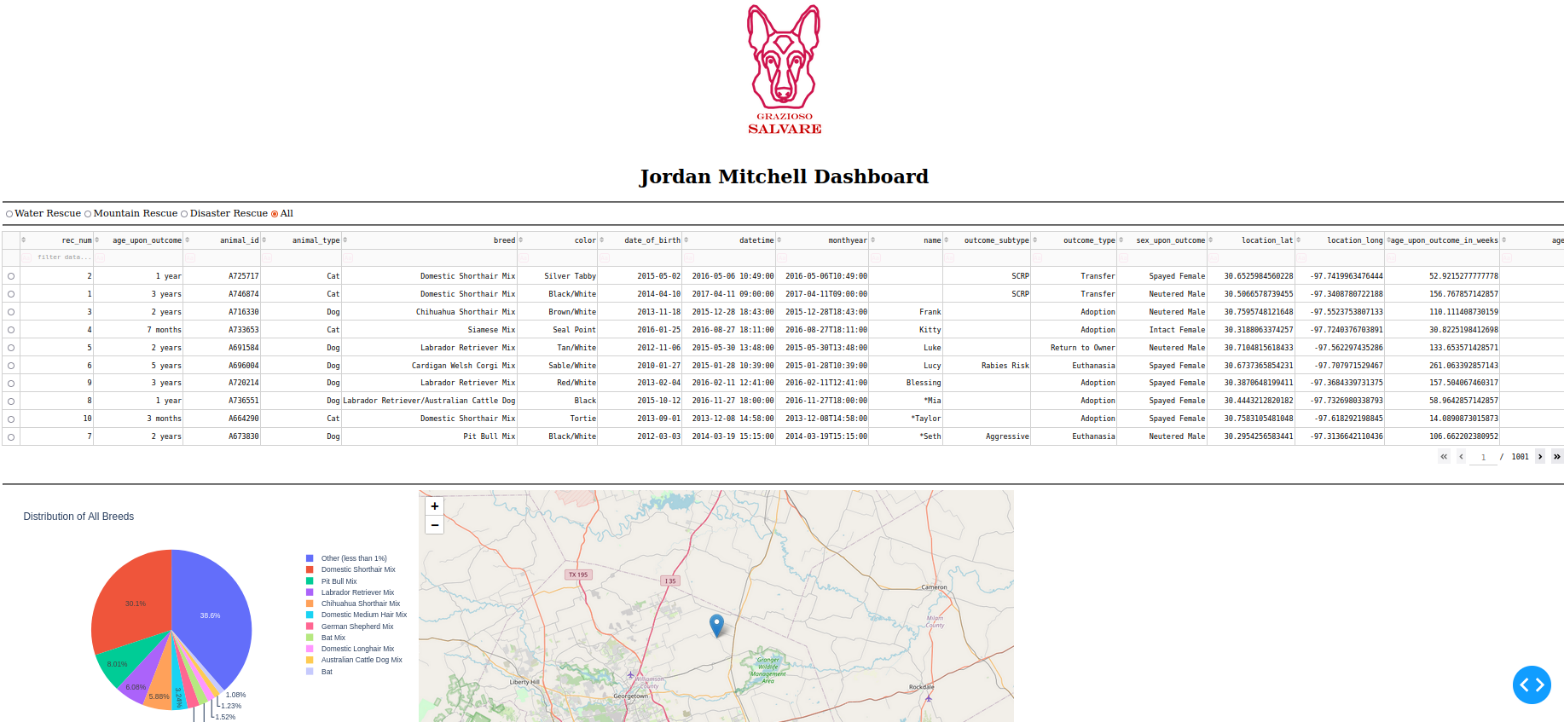
* [Python](https://www.python.org/downloads/)
* [MongoDB](https://www.mongodb.com/docs/manual/installation/)
* Required Python Libraries:
  + pip install pymongo plotly dash dash-leaflet pandas numpy

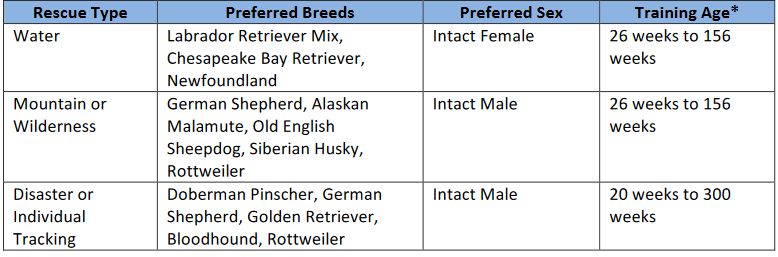
1. Setup Instructions:

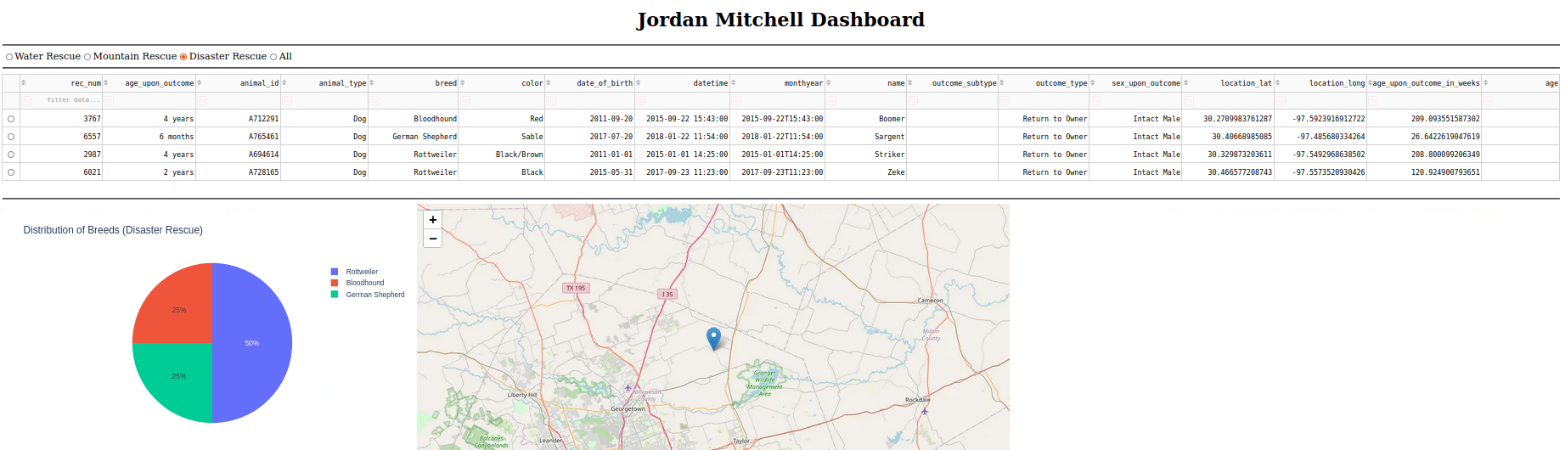
* Ensure the MongoDB database is running and loaded with the animal shelter dataset.
* Update the username, password, host, and port in the Python script for database connectivity.
* Place the Grazioso Salvare logo in the same directory as the script.

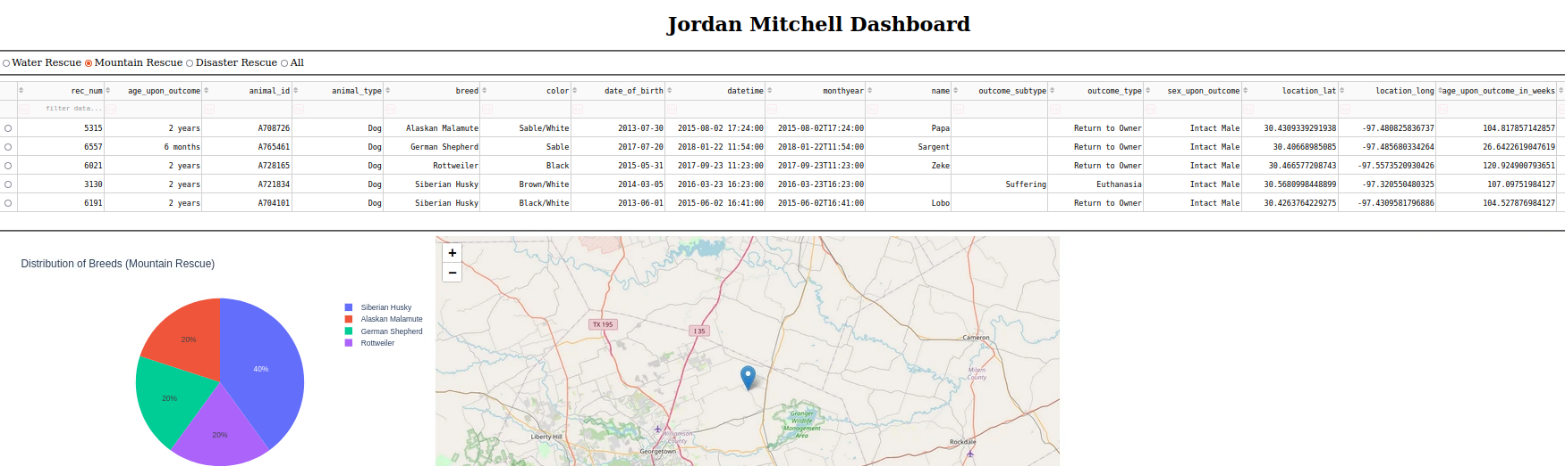
1. Run the Application:

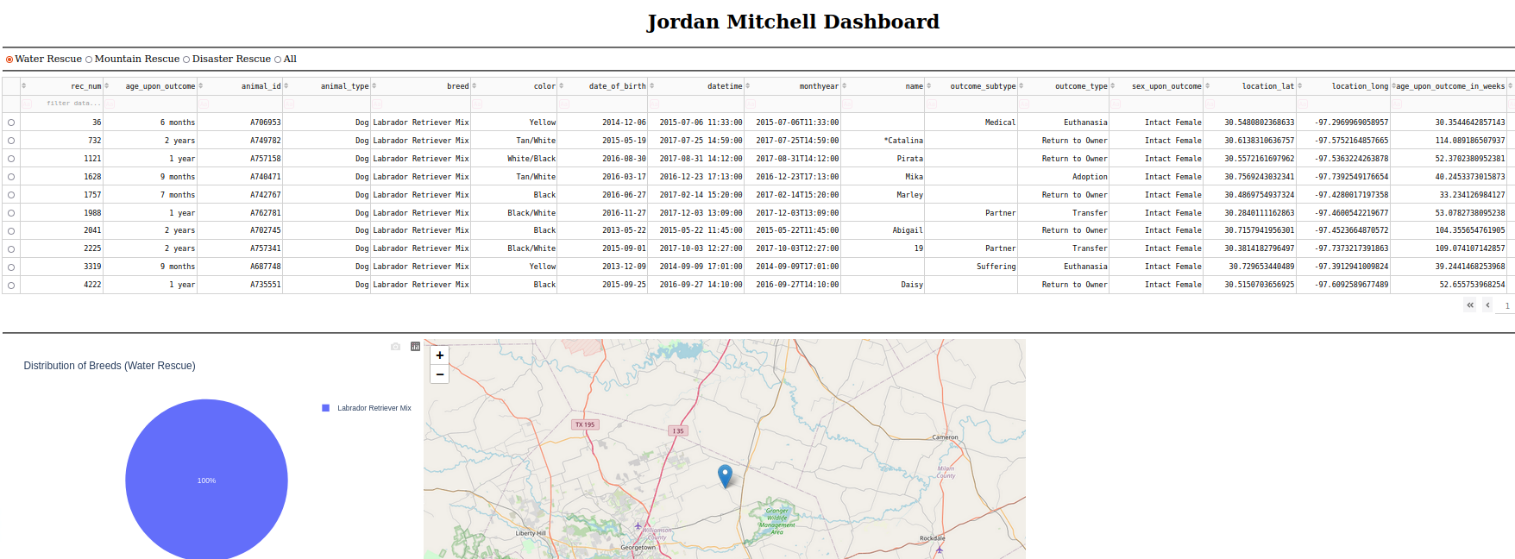
* Execute the script in a Python environment with Dash and MongoDB support:
  + python dashboard\_script.py

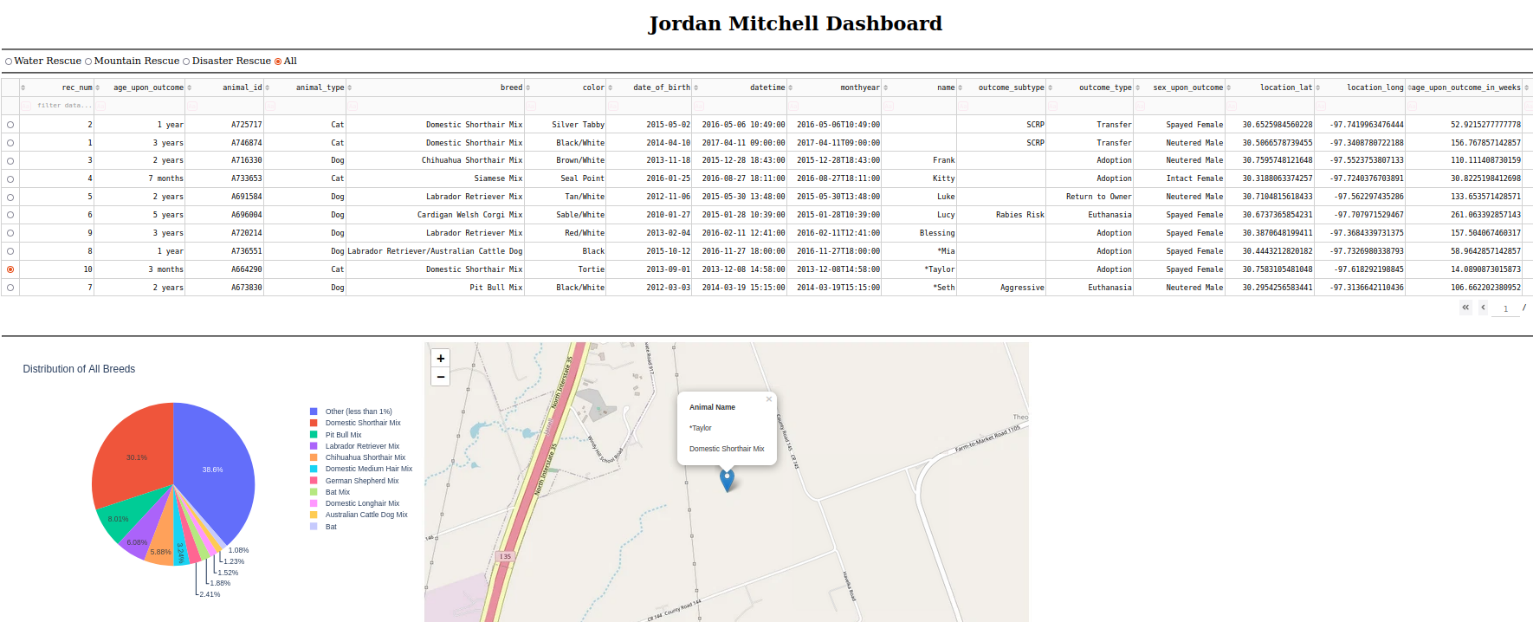
An image of the dashboard with the ”All” filter selected. Company logo visible, along with geolocation map default set to Austin Animal Center and a pie chart depicting the various breeds.

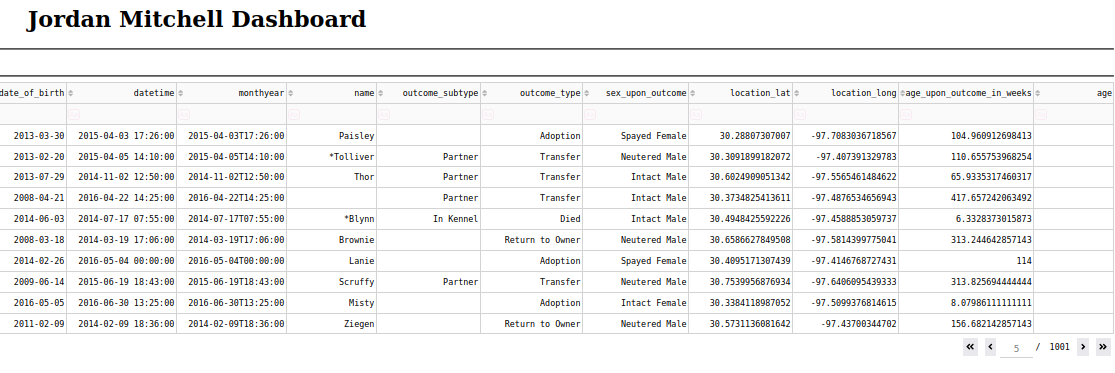
  
Each custom filter set to these specifications by client.

Example of the Disaster Rescue filter’s output and effect on the chart and table.

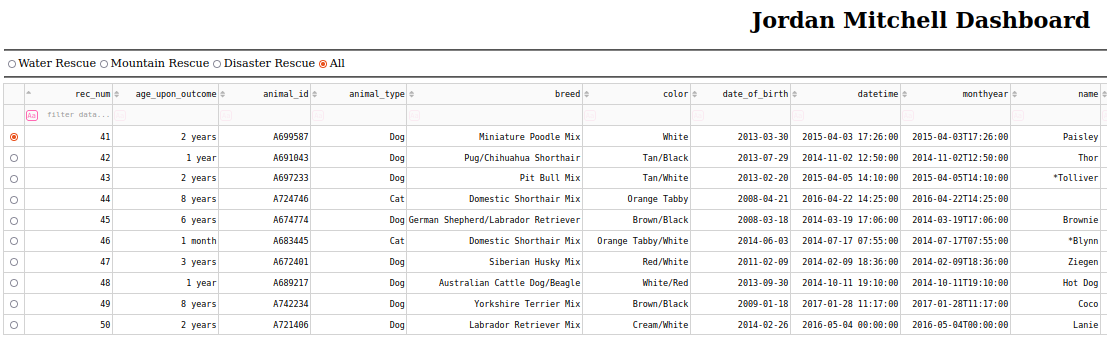
Example of the Mountain Rescue filter’s output and effect on the chart and table.

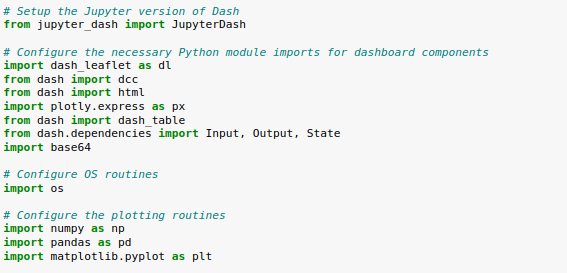
Example of the Water Rescue filter’s output and effect on the chart and table.

If a specific animal is chosen on the table, the map will update with its location, name, and breed.

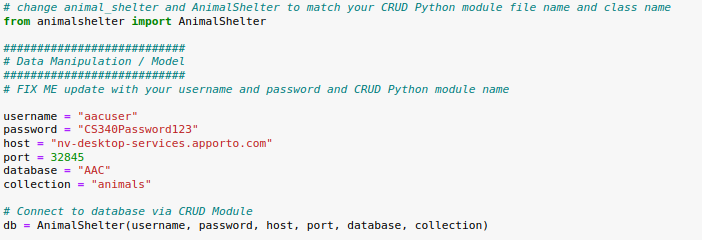
Example of the table’s pagination

Example of radio button on table

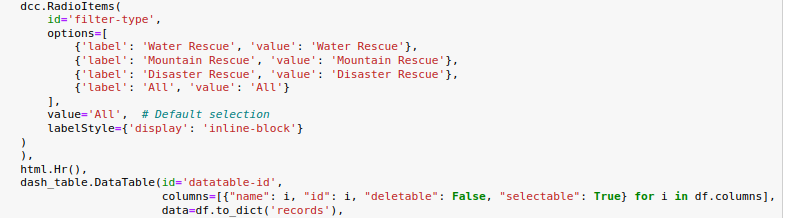
Sorting information from rec\_num column in ascending order starting from 41.

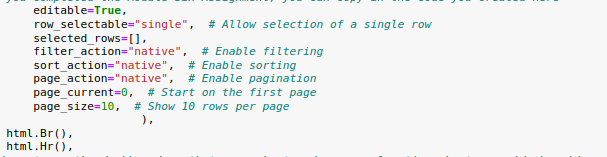


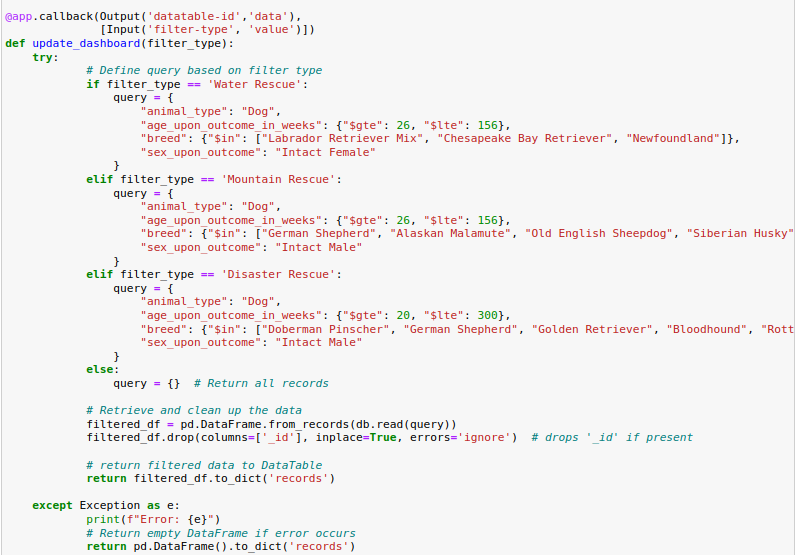
Initial dashboard setup. Importing relevant libraries

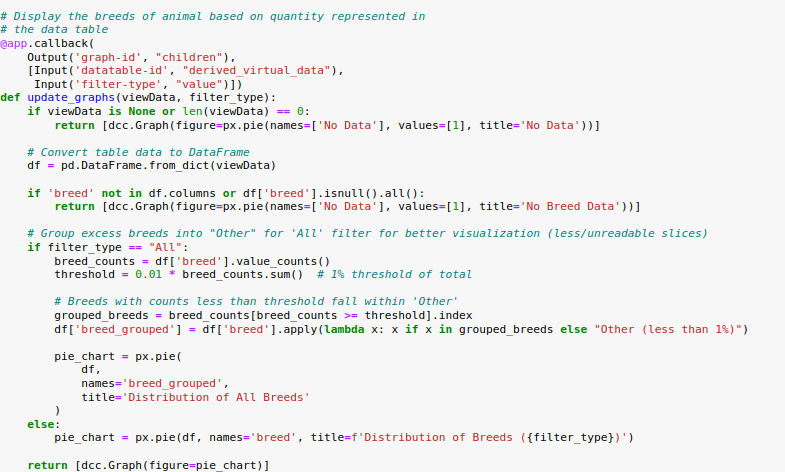
Import and instantiate CRUD module. Update credentials

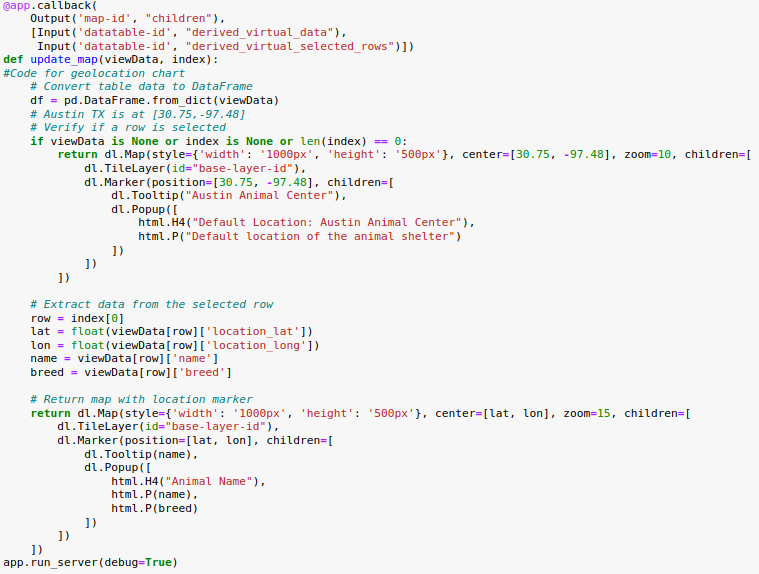
Add in company logo and center at the top of the dashboard

Code for adding interactive filtering options

  
Set up features for interactive data table for easier navigation

  
Code to define filter types based on requirements,

  
Code to filter interactive data table. Enable threshold of 1% to allow better readability.

Code for geolocation map