# animal\_shelter.py README

# *Animal Shelter Web App by Grazioso Salvare / Developed by Charles Haines (SNHU)*

## About the Project

Animal Shelter is a web application used to browse information in the database for the Austin Animal Center in Austin, Texas. The application was developed using Python 3, Dash, Plotly, Pymongo and the AnimalShelter Python 3 module. MongoDB database. The database used for the webapp is MongoDB. It has features to help users visualize the data contained within the database such as the ability to display a map of the location of any animal in the database and a pie chart depicting the breakdown of animal breeds that are being displayed.

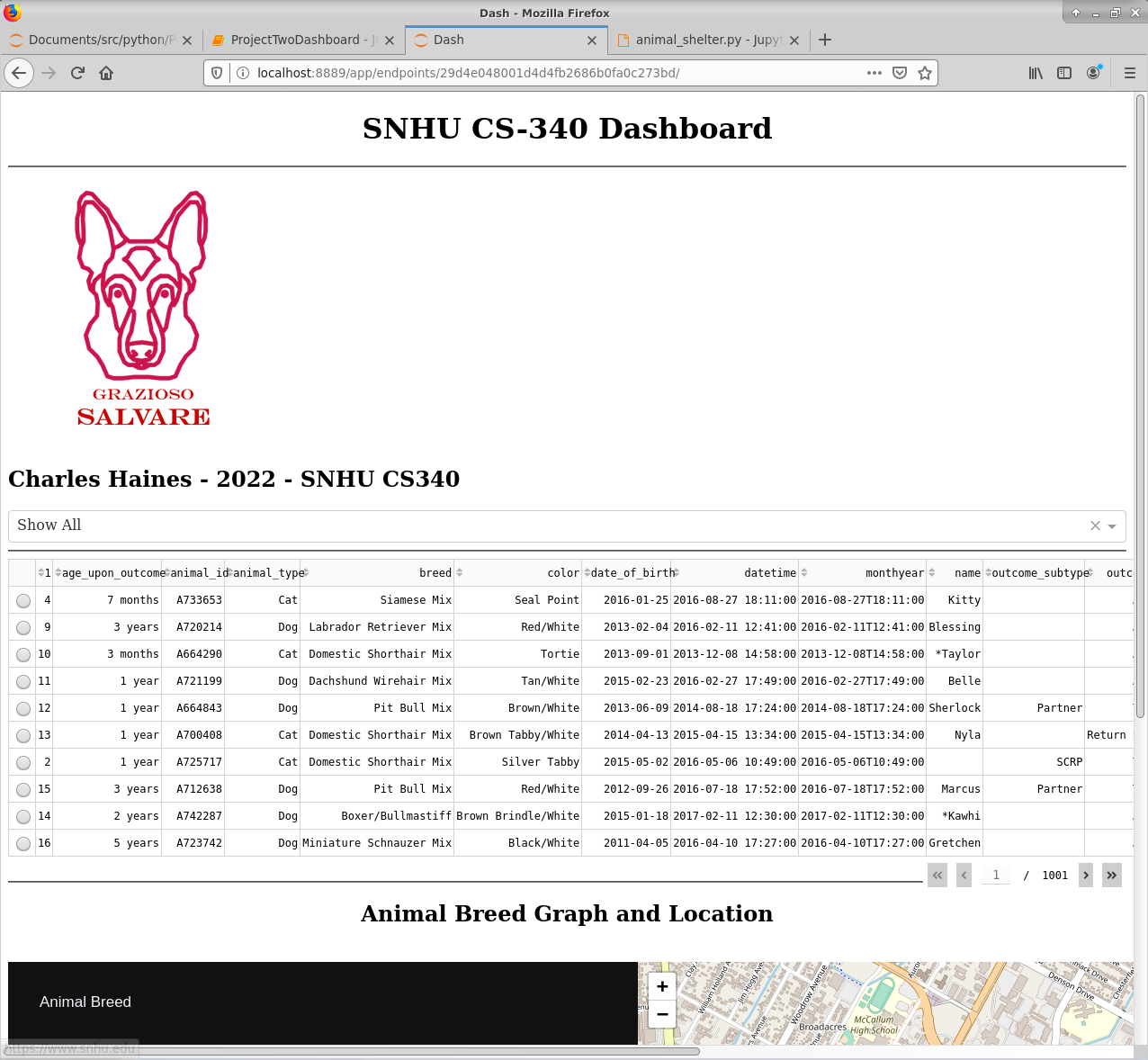
## Motivation

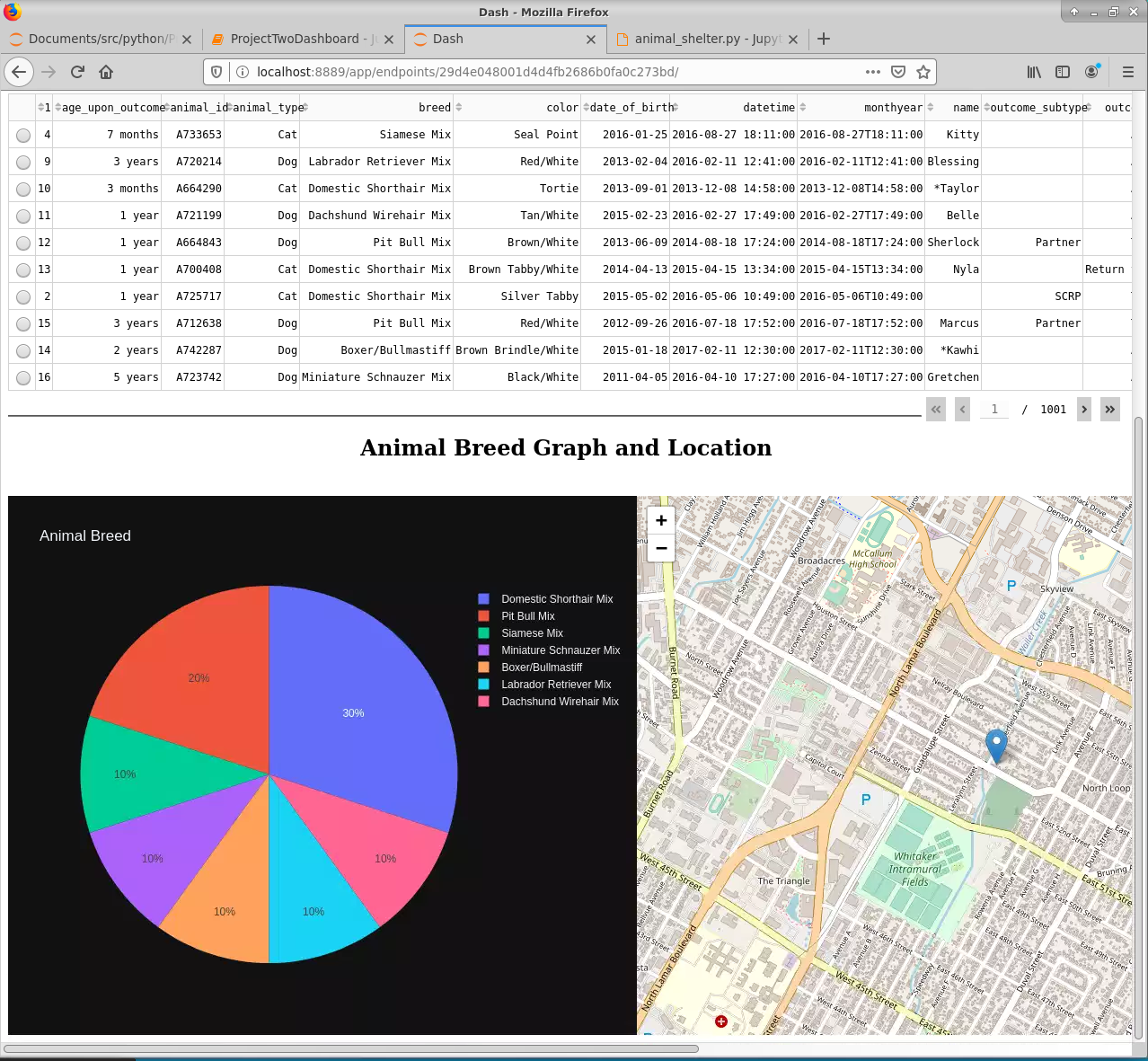
A local animal shelter hired me to create a web application that will allow them to view and manage data for local animal shelters. As an animal lover I could not be happier to be in a position where my work could help local animal shelters by providing an interface to see and interact with their data.

## Getting Started

To get started, open your browser, and navigate to the Animal Shelter web application URL.

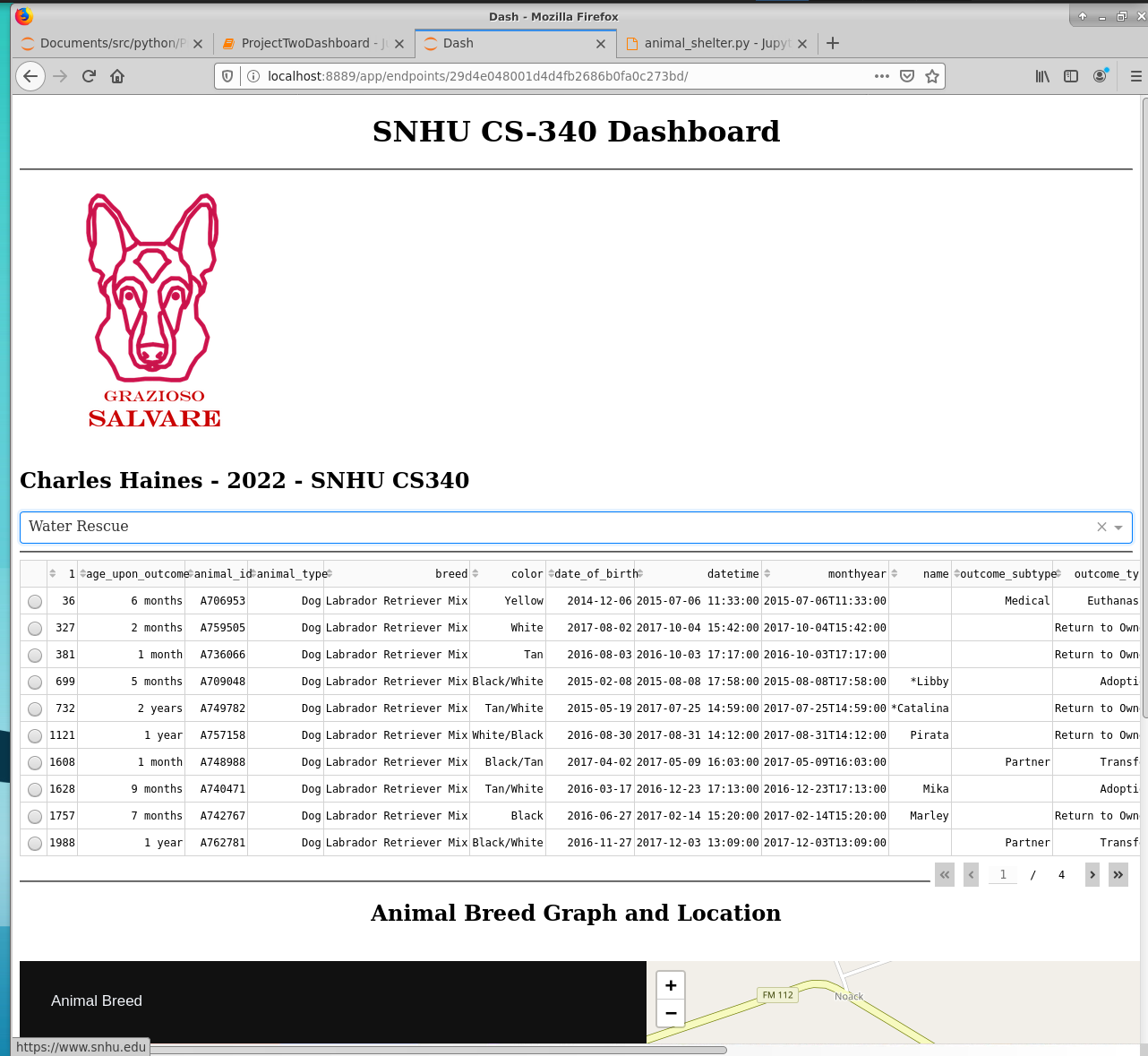
By default, all of the animal data from the database will be loaded and the graph will display information about the various animal breeds. Please note that the pie chart only displays data for the current page of animals being displayed. Each page shows information about ten animals and pages can be navigated by using the arrow buttons at the bottom right of the data table as seen in the screenshot below:

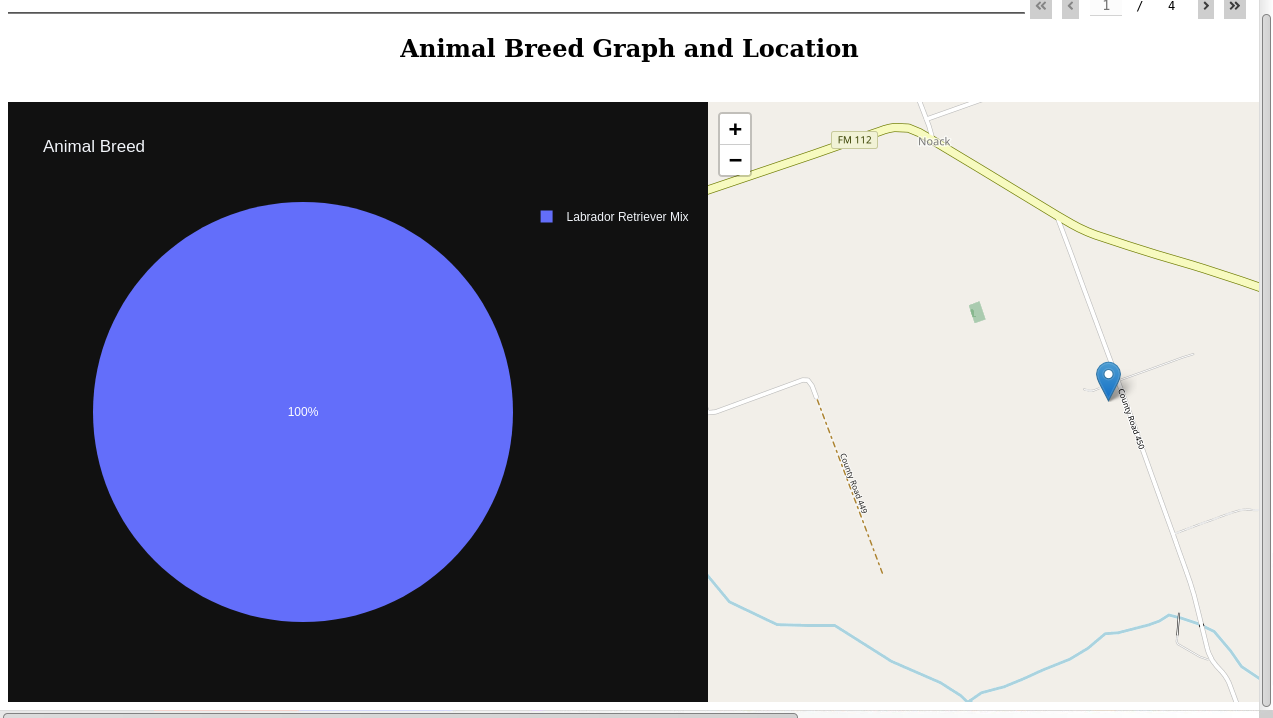




The dropdown box above the table allows you to filter the results being displayed. Here is a sample of the available filters:

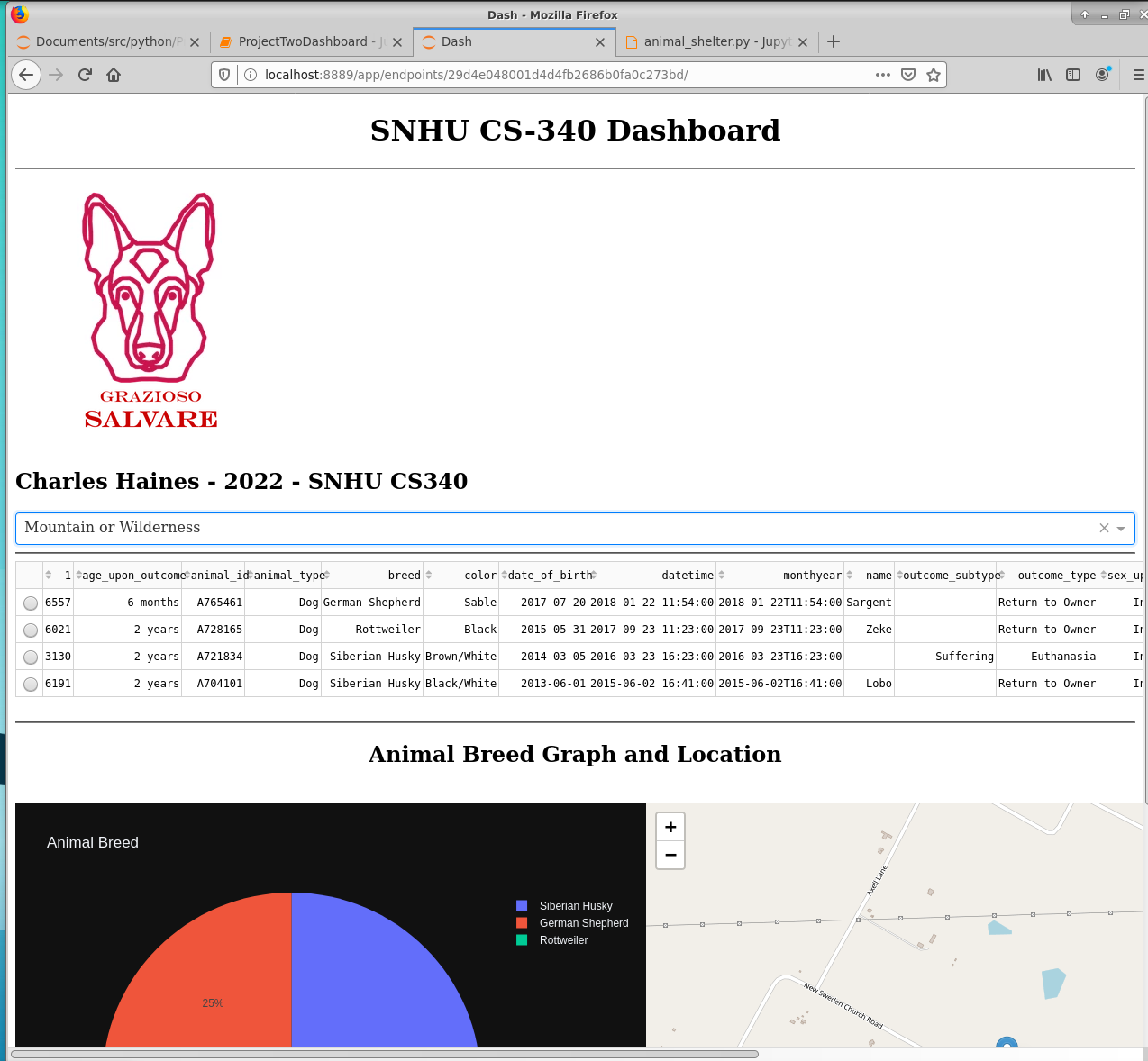
Water Rescue:



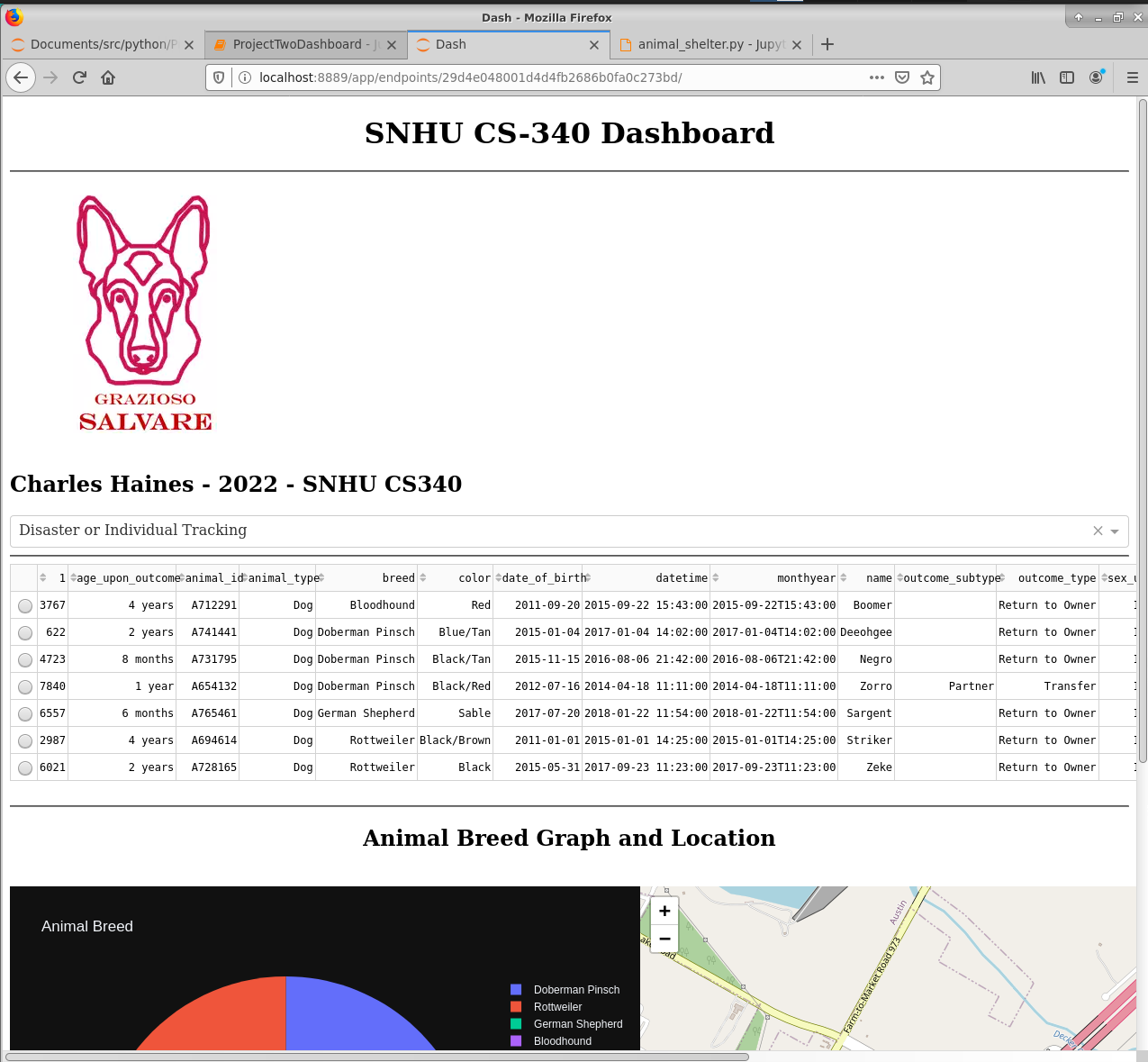


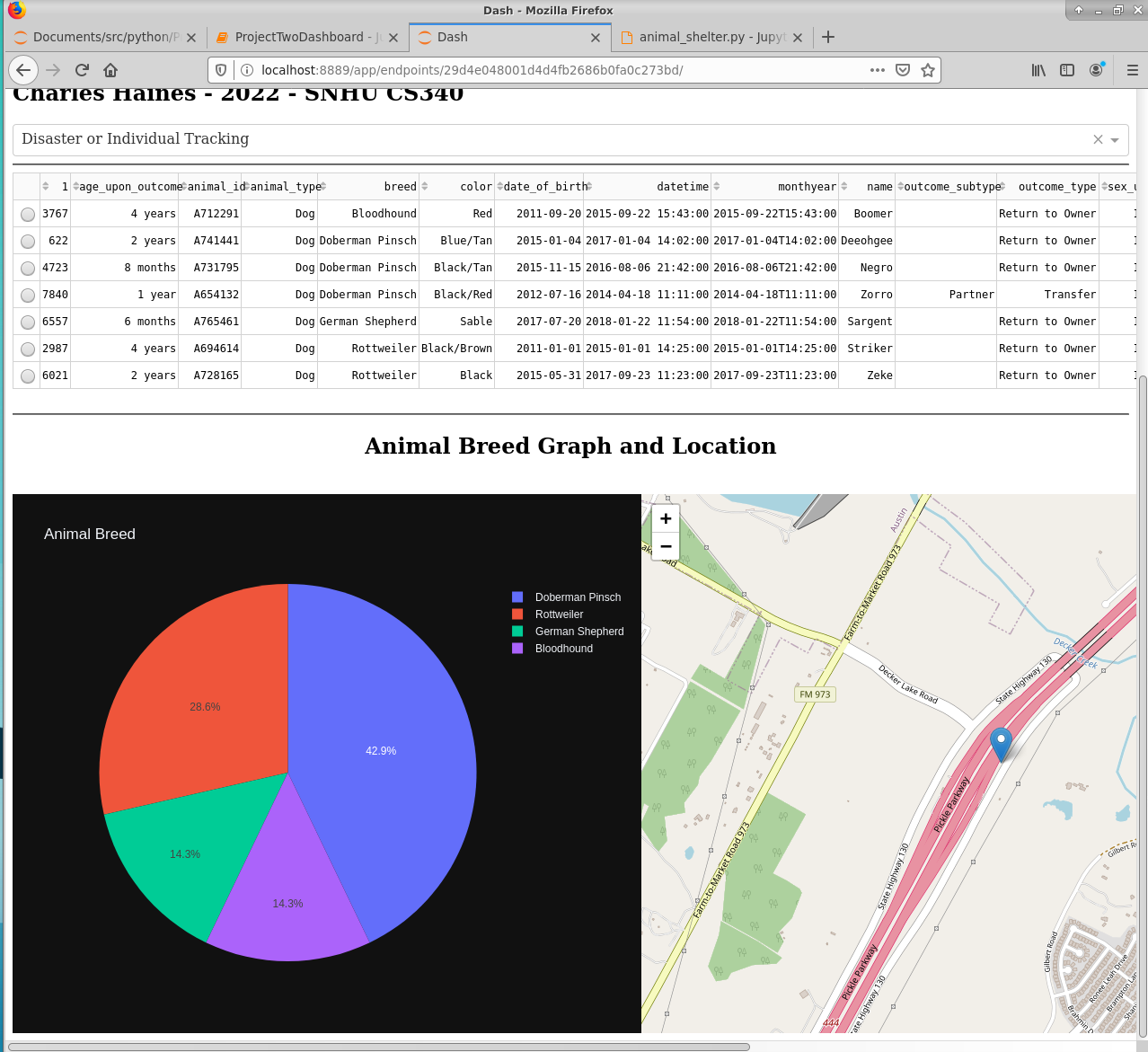
## Tools Used

Mountain or Wilderness Rescue:



Disaster or Individual Tracking:





As you can see in the above examples, the chart and map updates based on the data currently being displayed in the data table.

**Tools Used:**

* **Python 3**
  + Python is a popular programming language that can be used to quickly create powerful applications that can work with data from a variety of sources. Python was chosen due to its ease of use and many plugins and utilities available for working with and visualizing data.
  + [Python Homepage](https://www.python.org/)
* **MongoDB**
  + MongoDB was chosen because it is a reliable schema-less document-based database that can easily contain and manage the animal data for the Austin Animal Center. MongoDB was chosen as the model component since it has an official driver for Python 3 that can be integrated within a Dash web application.
  + [MongoDB Homepage](https://www.mongodb.com/)
* **Pymongo**
  + Pymongo is the official Python driver for interacting with a MongoDB database.
  + Pymongo allows a Python application to store, retrieve, and modify data stored in a MongoDB database.
  + [Pymongo Documentation](https://pymongo.readthedocs.io/en/stable/)
* **Dash Framework**
  + Dash is a framework that allows Python developers to easily create and manage a web application with the benefit of allowing developers to create a web application without using any raw HTML.
  + Dash contains functions that can create the HTML code needed to create the front-end of the Animal Shelter web application.
  + Dash easily integrates with Plotly which allows developers to easily create and display interactive charts and graphs.
  + dash\_tables is a feature included with the Dash framework that allows developers to create interactive data tables that can be used to display data in a user-friendly format that is highly configurable.
  + [Dash Framework Homepage](https://plotly.com/dash/)
* **Plotly**
  + Plotly is a tool that allows Python developers to create interactive visual representations of data quickly and easily.
  + Plotly can create and display custom graphs and charts for various data types.
  + Plotly easily integrates with Pandas which allows developers to use Pandas DataFrames to store and manipulate data once it has been retrieved from the database.
  + [Plotly Homepage](https://plotly.com/)
* **Pandas**
  + Pandas is a data analysis library that can be used to store and analyze data from various sources. Pandas was chosen since it offers the ability to store data in memory once it has been loaded from the database. Pandas is used to analyze the animal shelter data prior to creating the graphs that are displayed in the application.
  + [Pandas Homepage](https://pandas.pydata.org/)

## **Steps Taken to Complete This Application**

* Import data into a MongoDB database server
* Setup authentication for the database server to protect the database from modification by unauthorized parties or individuals
* Created a Python3 CRUD module which can be used to create, read, update, and delete records from the animal shelter database.
* Integrated the Python 3 CRUD module within a new Dash web applications
* Developed the Dash application layout which defines how the application looks and the various web components used to display the application in a web browser.
  + Added interactive components such as the dropdown box used to filter the data displayed in the data table and graphics.
* Define various callback functions used to update the layout based on the current selections of the interactive components such as the filter selection dropdown. This allows the components like the map and chart to update in real time based on the current filter selection.
* Tested the various features and options to make sure that they are working as intended.

**Challenges and Solutions:**

There were a few challenges I encountered when developing the application. This is a quick breakdown of what the challenges were and how I resolved them:

* Make each animal in the table selectable and update the map based on the current selection
  + The challenge I faced in this task turned out to be a simple mistake. I was treating a variable for the selected row as an integer when it is a list.
  + After recognizing this simple mistake and updating the code to work with a list instead of an integer got this feature working as expected within a few minutes of making this discovery!
* Some breeds were not being displayed when selecting various filters
  + After looking through the data, I discovered that some of the breed names in the database were shortened.
  + This issue was resolved easily by updating my filters to use the shortened names that appear in the database as opposed to the full names listed in the applications specifications document.

## Roadmap/Features

* Implement better security
  + Currently the database credentials are hard coded into the web application code.
  + This is a huge security issue and needs to be corrected before the application enters production.
  + This situation is not a big deal when developing and testing a new application but will have huge security implications if not corrected before deploying this application for production use.
  + SSL should also be configured within Dash or by using a reverse proxy to ensure that data sent between the browser and the application is encrypted.
  + SSL needs to also be set up and configured for the MongoDB server that will be used when the application is entered into production. This ensures that communication between the database server and the back end of the application is encrypted.
  + SSL also provides the benefit of allowing the various servers and clients involved to know that they are talking to the correct entity and not an imposter. This will eliminate the possibility of an attacker using a man-in-the-middle attack to impersonate the application or database server.
* Package web application into a container so it can be easily deployed.

**Final Remarks from the Developer**

I had a lot of fun creating this application by creating the various layers and pieces needed to achieve the end result. I want to thank Dr. Kellogg at Southern New Hampshire University for his support, tips, and advice over the past seven weeks, as they were essential in the successful development of this application.

## Contact

Your name: Charles M Haines II <charlesmhainesii@gmail.com>