## About the Project/Project Title

*Global Rain has been commissioned by Grazioso Salvare, an innovative rescue-animal training company, to create a software application that can work with an existing database from animal shelters to identify and categorize dogs. The animals in the database are filtered to identify which are good candidates for search-and-rescue. This project is for full stack development that will include a database and client-facing web application dashboard and allow the users at Grazioso Salvare to have access to this database. The code created for this application will be open-source and available on GitHub so that others may use this information to provide similar services. This project uses a CRUD Python module to perform Create, Read, Update, and Delete operations within the MongoDB database. Creating the project in this manner will allow for a simple interaction within the animal database to find animals for rescue training. The Crud Module is imported into Python code, written in the Jupyter Notebook, to create the data table, use interactive filtering, update charts and maps, geolocation mapping, data visualization, and a dashboard to run the functions necessary to have a fully functioning application.*

## Motivation

*The motivation behind this project is to create an application coded within Python using MongoDB as the database. This will allow CRUD access to documents in MongoDB. By using these steps and using the power of Python and Mongo, we can maintain create, read, update, and delete functions within a database. Using the PyMongo driver to access that database while using Python code allows for a better application and one that is well-supported. The motivation also comes from allowing for a simple, usable application for the end-user that is fast, convenient, and accurate. The dashboard is created with the customer and user in mind and uses the Dash Framework to ensure that application is updating in real time giving the user the best possible experience.*

## Getting Started

## To get started with this application we first imported our data from a CSV file called animal shelter. We did this within the Linux Shell and then proceed to work inside of MongoDB. Within the MongoDB terminal we create a CRUD Module with Python coding language. This Module, called Animal Shelter, created a Create, Read, Update, and Delete function that can be imported into out application later. This allows us to interact with the data that was loaded from the CSV file.

I then developed the dashboard from which the application would function. I designed the layout of the application with the base code that was provided to me. The application consists of a header in which I have identified the name of the dashboard. There is a creator identifier that must be placed within the header, along with the company logo. Within the company logo, I have embedded a hyperlink that takes you to the SNHU website. The application was then given a geolocation map which would update when the data from the table changed. A second chart was added, and it was a pie chart that would also update when changes were made within the data table functions. The data table was given the ability to be manipulated by functions at the top of the table. The type of dog was loaded into a widget that could be altered to find the exact data needed for the rescue needed.

The application was then tested within Jupyter notebook and changes were made whenever required. The application is hosted through port 30032 and functions as requested by the company. The charts and tables update accordingly, and the geolocation map changes as well.

## Installation

* *For this installation you need to have installed PyMongo and the Jupyter notebook.*

*The MongoDB Python driver, PyMongo, is used in this module because it allows you to interact with MongoDB but also use Python code. It is a mature and widely used driver that is easy to use, is well-documented, and provides support.*

* *The Python CRUD Module that was created, named AnimalShelter, must be imported within the Python code and is used as a bridge between the application and the database in which it pulls its information. This module allows the application to create, read, update, and delete information within the application and ensure that relevant data is used in the application.*
* *The Dash Framework is also used within this project and must be imported into our code within the Jupyter notebook. The Dash Framework allows a web application to run in Python. By using the Dash Framework, there is no need for HTML, which would normally be used to create a web application. Dash is based in Python and allows for the creation of interactive web applications.*
* *To be able to create and interact with the charts that are necessary within our application, we much install Plotly for data visualization. Plotly is a very powerful tool when it comes to data visualization and was the perfect choice for our application.*
* *Dash Leaflet is implemented to allow the functioning of a geolocation map for the application. Dash Leaflet can function within the Dash Framework and is a very straightforward and seamlessly integrated geolocation mapping function.*

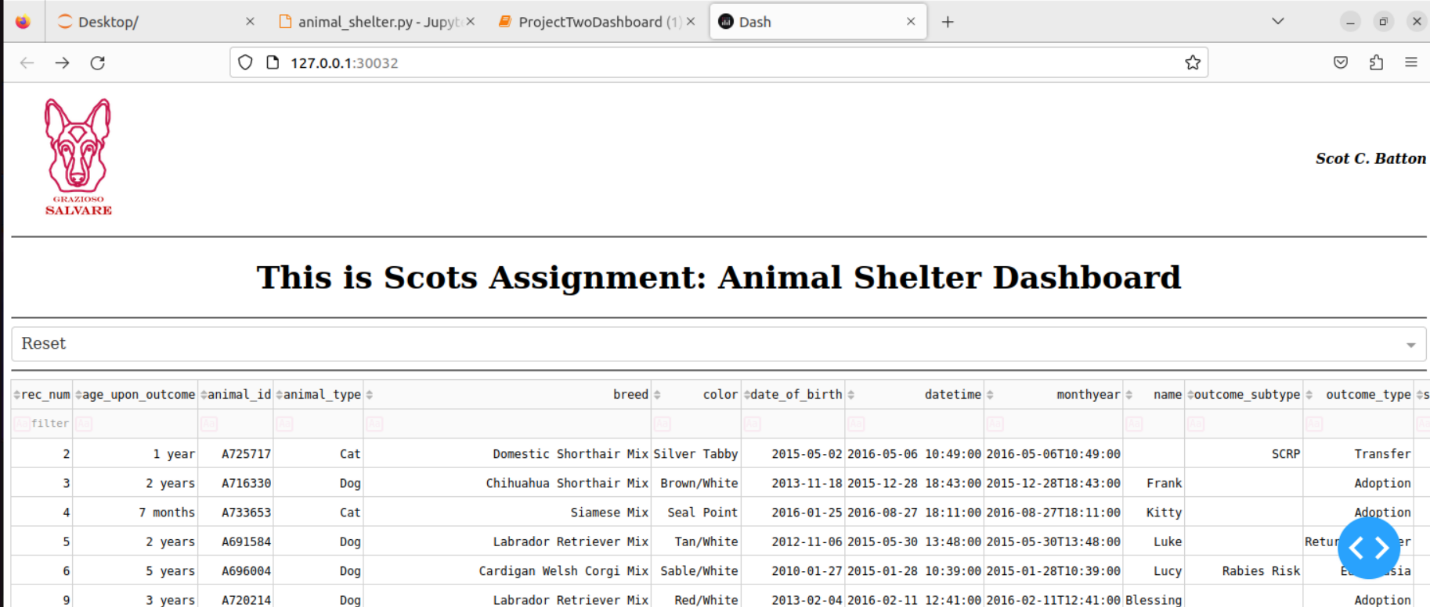
## Usage

### The applications usage is very simple due to the nature of the CRUD module and the Python code used to create the framework for the app. Once the application is reached there are drop downs that control the data within the table. The data table is modified by the type of dog that the trainers are looking for. As you use the dropdown menu, the data will change, and so will the pie chart and geolocation map. You can see the information change and get more specific each time the drop down is used.

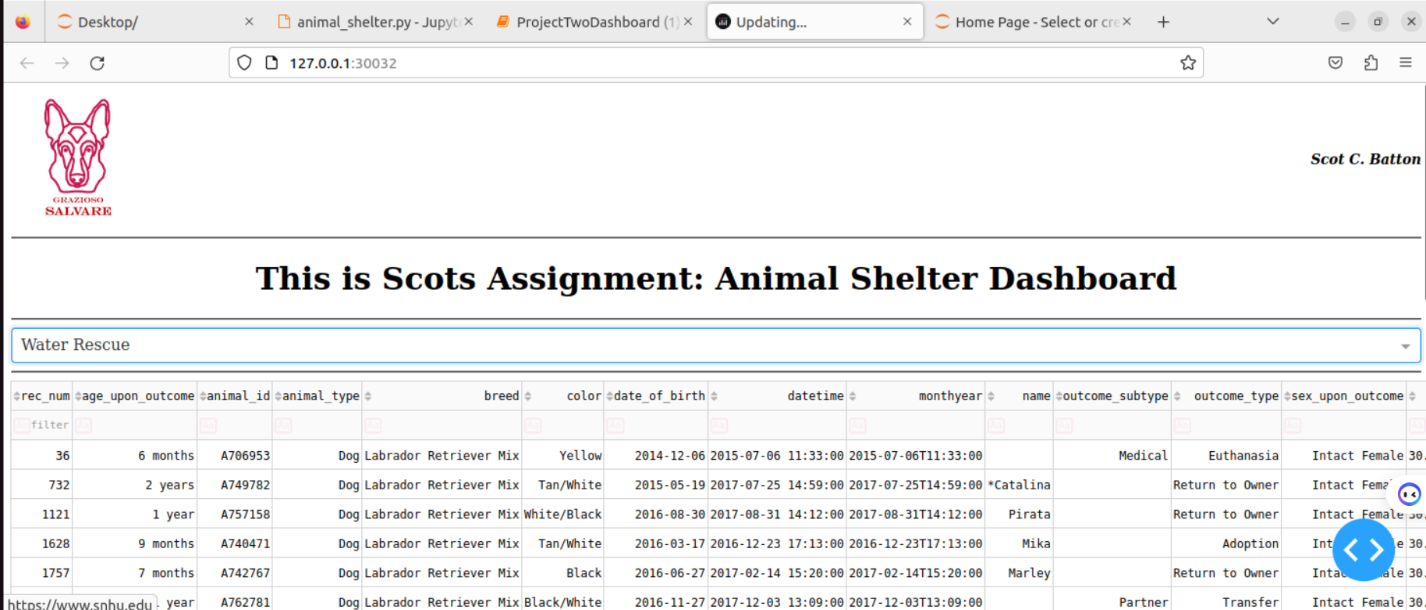
The geolocation map will allow you to see the locations of the animals within the shelter system and map the drive in real-time. This information that is received is also analyzed and shown in a way that allows the user to get as much information about the animal as possible.

**Screenshots**

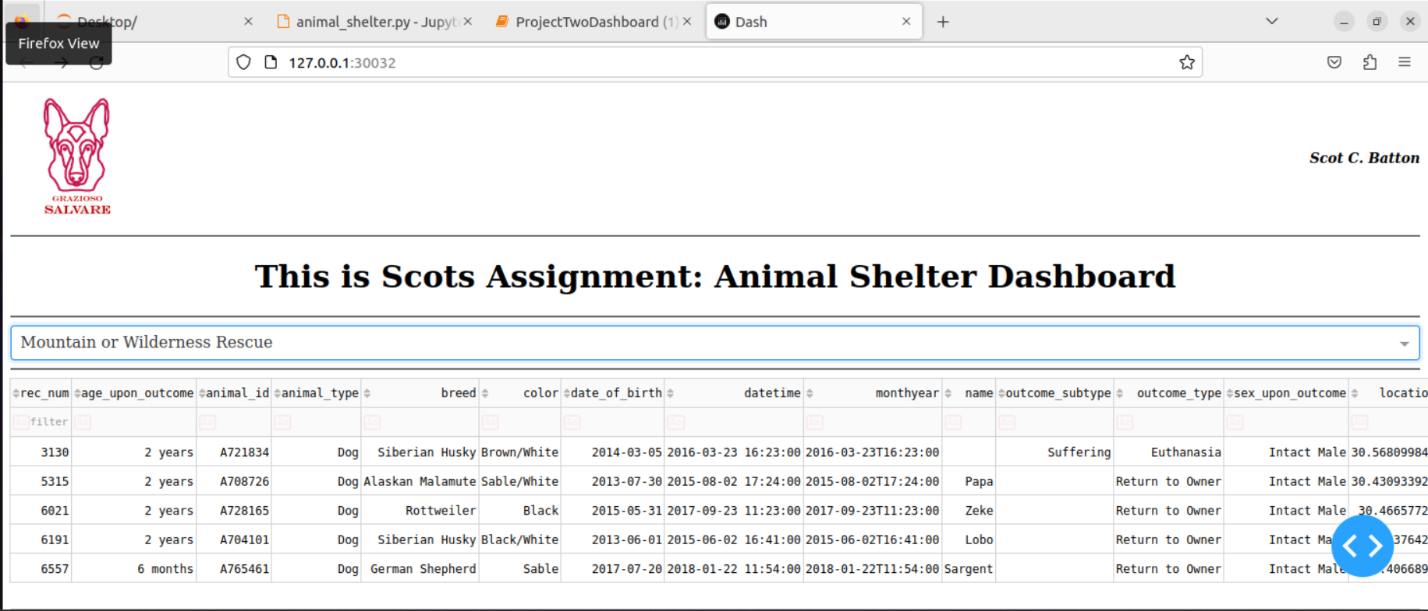
This screenshot shows the dashboard that was created within the Jupyter Notebook. The image shows the Grazioso Salvare logo in the left top corner and my identifier in the top right corner. Below this is my headline for the name of the Dashboard that I created. This initial screenshot shows the data table before it was filtered with any of the widgets that were created.

****

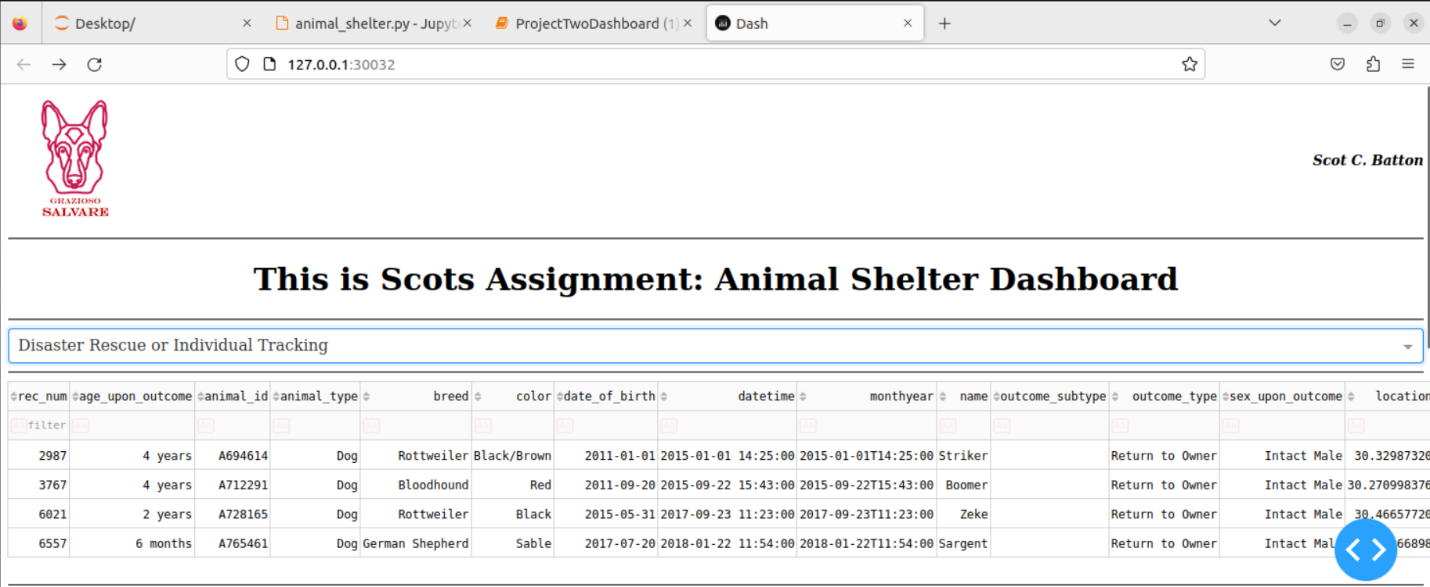
This screenshot shows the data table after it was filtered with the widget that filters only the animals that are deemed best for water rescue.

****

**This screenshot shows the data table after it has been filtered with the widget that shows animals that are deemed best for Mountain or Wilderness Rescue.**

****

**This screenshot shows the data table after it has been filtered with the widget that filters animals that are deemed best for Disaster Rescue or Individual Tracking.**

****

**Included in the submitted code file is the screencast that shows the functionality of the app that I have created for Grazioso Salvare that shows the widgets working and updating the data tables, pie chart, and map. Within this screencast, it shows that the logo includes a hyperlink to the SNHU website.**

### Resources

*All the tools that I implemented into this project our open-source and can be easily accessed. The documentation for all the tools can be found here.*

*MongoDB:* [*www.mongodb.com*](http://www.mongodb.com)

*Dash: dash.plotly.com*

*Plotly: plotly.com/python*

*Dash Leaflet: dash-leaflet.plotly.com*

## Contact

Scot C. Batton