# CS 340 README Template

## Project Title

Interactive Dashboard for Grazioso Salvare, populate database information and filter the output with user selection including an interactive pie chart and an interactive geolocation map using Dash by Plotly.

## Getting Started

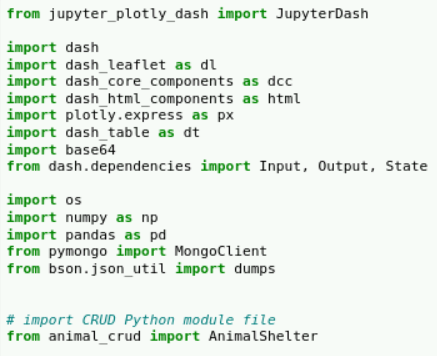
To get your database up and running with python, ensure you do the following:

Make sure you have pymongo installed (you can also refer to this tutorial for help getting started and with CRUD operations <https://pymongo.readthedocs.io/en/stable/tutorial.html> )

Put your socket number from mongodb here for connection ‘mongodb://%s:%s@localhost:#####’, refer to the code below or the tutorial link.

From the linux shell, get in mongodb before you run your python code, example “/usr/local/bin/mongod\_ctl start”, don’t login or start with no-auth, the python initialization will do that.

Import required components to work with Dash, plotly, etc for required functionality:



Create a py file with authentication, setup of database, and CRUD operations to use/import in support of the app – This particular dashboard will need the authentication and Read functions.

Code in username and password, instantiate CRUD class to authenticate and use database:

# username and password and CRUD Python module name

username = "Username"

password = "Password"

shelter = AnimalShelter(username, password)

Create Your Dataframe from your database records:

# class read method must support return of cursor object

df = pd.DataFrame.from\_records(shelter.read({}))

Include Radio Items, data table, graph, and map in layout

I began with the data table functionality, testing it to make sure it updated with the different selections. Then I went to the map to update with data table and row selection, and lastly the pie chart callbacks and functions for updating with radio items.

## Installation

MongoDB is a general purpose NoSQL database with the following key characteristics: flexible schema design, high availability, feature rich, scalability and load balancing, aggregation framework, native replication, security features, JSON objects for storing/transmitting documents, MapReduce, multi-document ACID transactions, and mature tooling (Giamas, 2019). MongoDB has successful cases in supporting applications in the following areas: IoT, mobile applications, real-time analytics, personalization, catalog management, and content management (Giamas, 2019). To connect to MongoDB using Python, an official MongoDB low-level driver, PyMongo, is used(Giamas, 2019). Using Python with MongoDB, you can create an application that connects to your database and perform mongo operations such as creating a database, creating a collection, performing CRUD operations in a database, and indexing among others.

Alex Giamas. (2019). *Mastering MongoDB 4.x : Expert Techniques to Run High-volume and Fault-tolerant Database Solutions Using MongoDB 4.x, 2nd Edition: Vol. 2nd ed*. Packt Publishing.

You will need:

1. Python, can download from <https://www.python.org/downloads/>

2. PyMongo, can install with: <https://www.mongodb.com/languages/python> or <https://pymongo.readthedocs.io/en/stable/installation.html>

3. If you want to use Jupyter as in this demonstration, you can use in browser or download <https://jupyter.org/>

4. Dash is an easy to use open source library for building data apps with customized user interfaces with Python, <https://dash.plotly.com/introduction>

5. Dash core components, <https://dash.plotly.com/dash-core-components>

6. Dash Data Table, <https://dash.plotly.com/datatable>

7. Dash Leaflet for map, <https://dash-leaflet.herokuapp.com/>

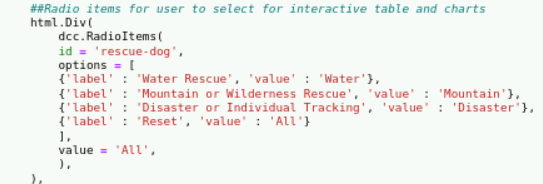
8. Pie charts with plotly express, <https://plotly.com/python/pie-charts/>

## Usage

Populate a data table from a chosen database and display filtered results from user options, in this case, specific breeds for different types of rescue capabilities.

### Code Examples

Options are used as Radio Items, in the app layout:

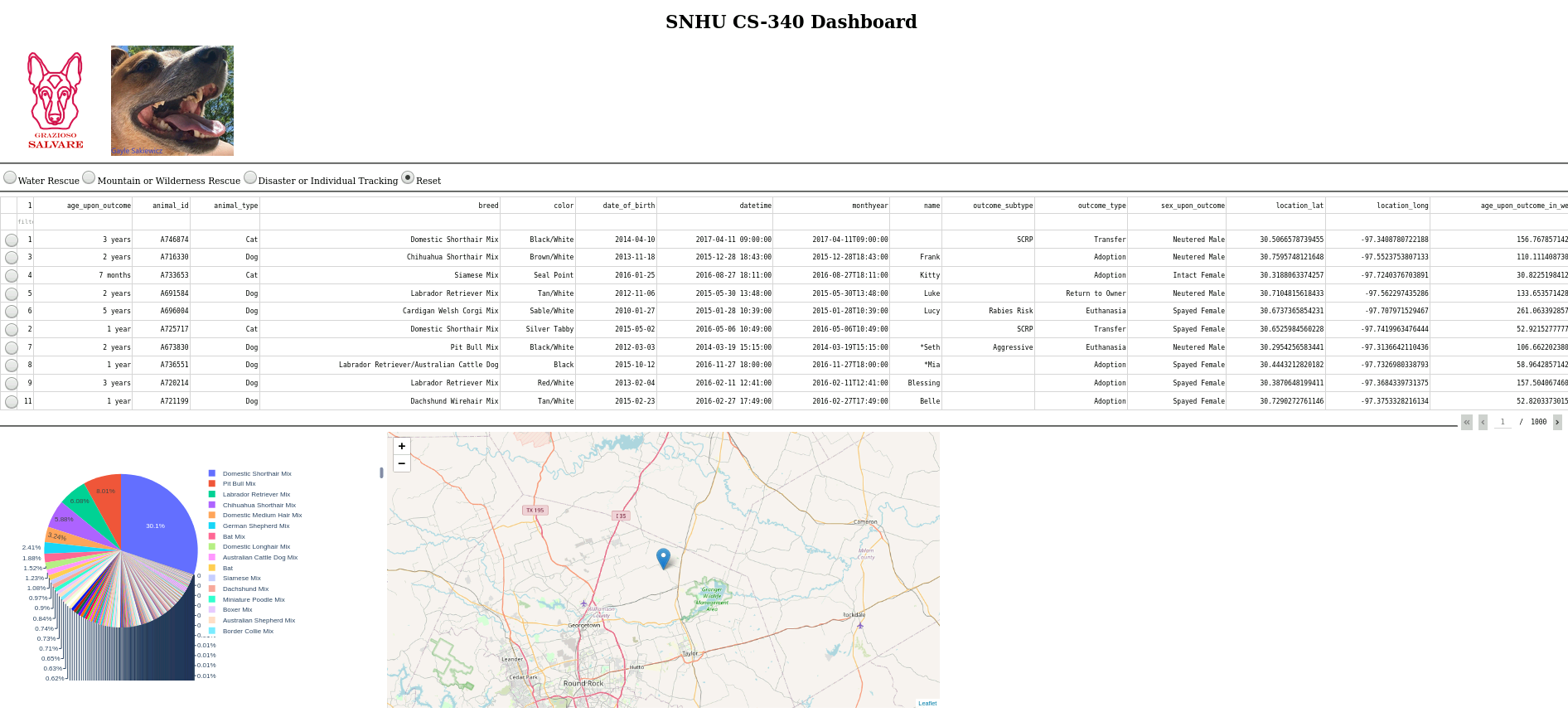
**

Radio options are used in the data table and pie chart call backs to update:

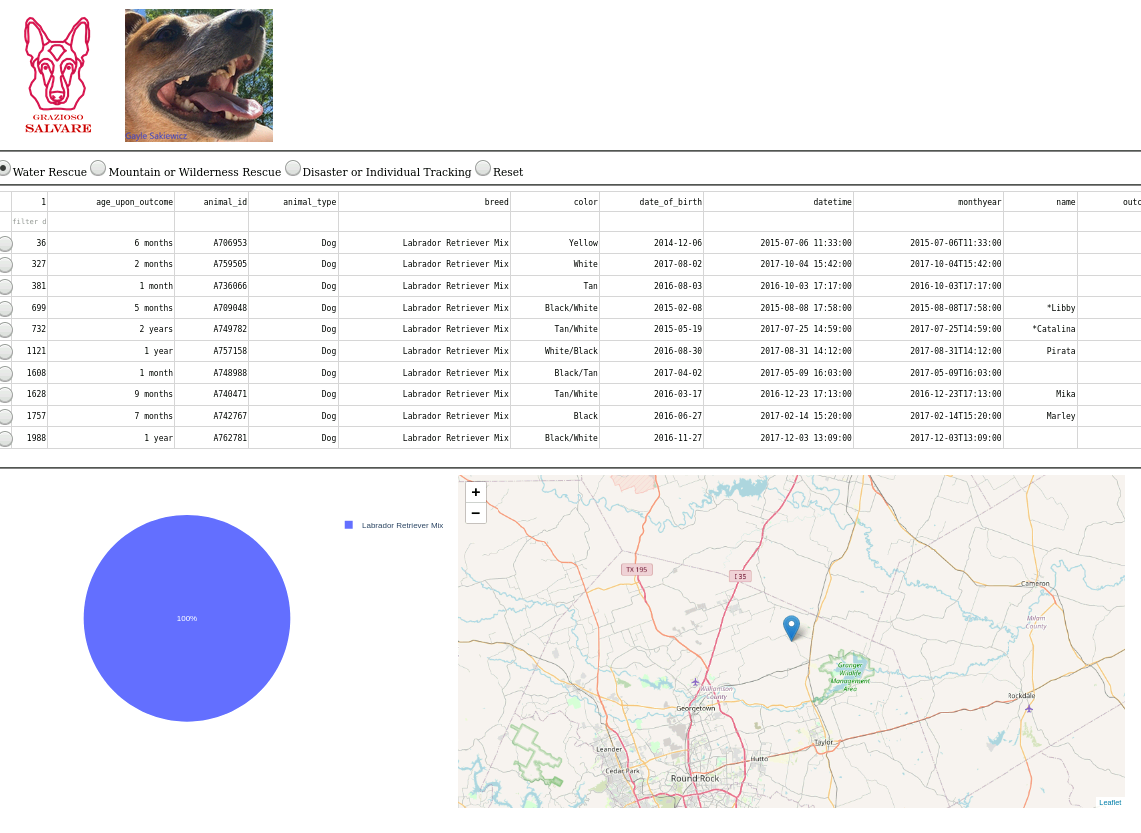


### Tests

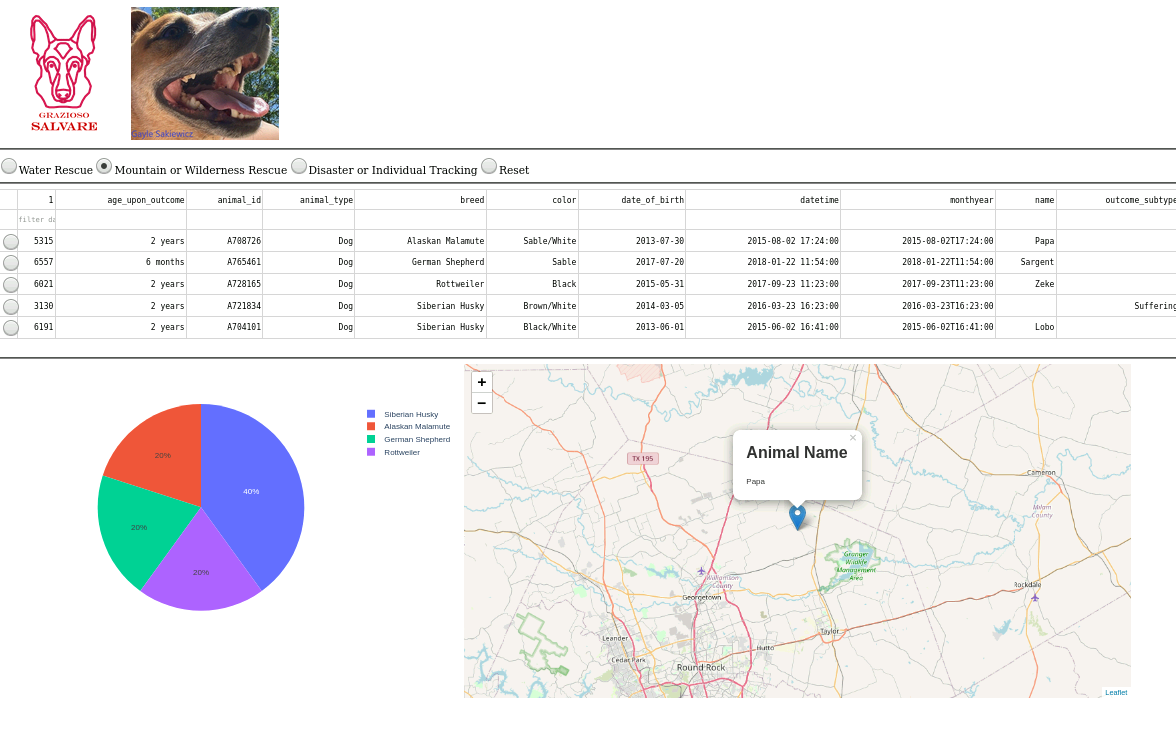
When first run data table shows all animals, pie chart shows all breeds, and map shows the first animal in database:



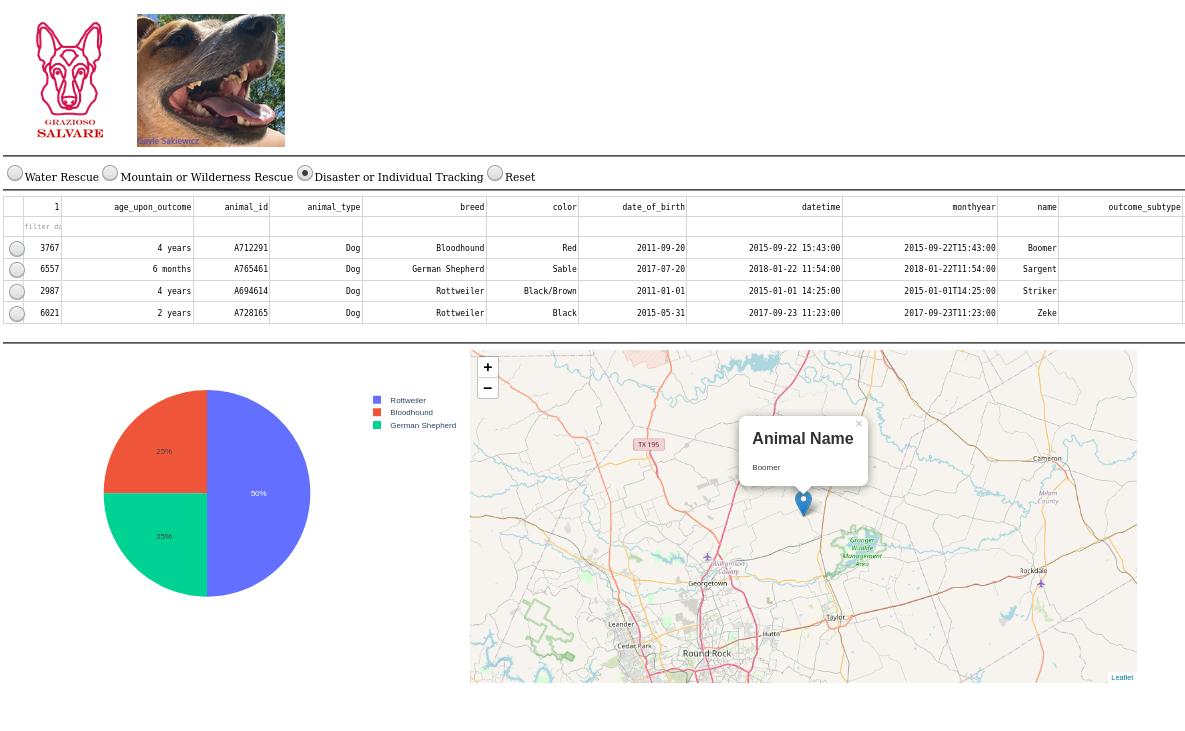
When “Water Rescue” is selected:



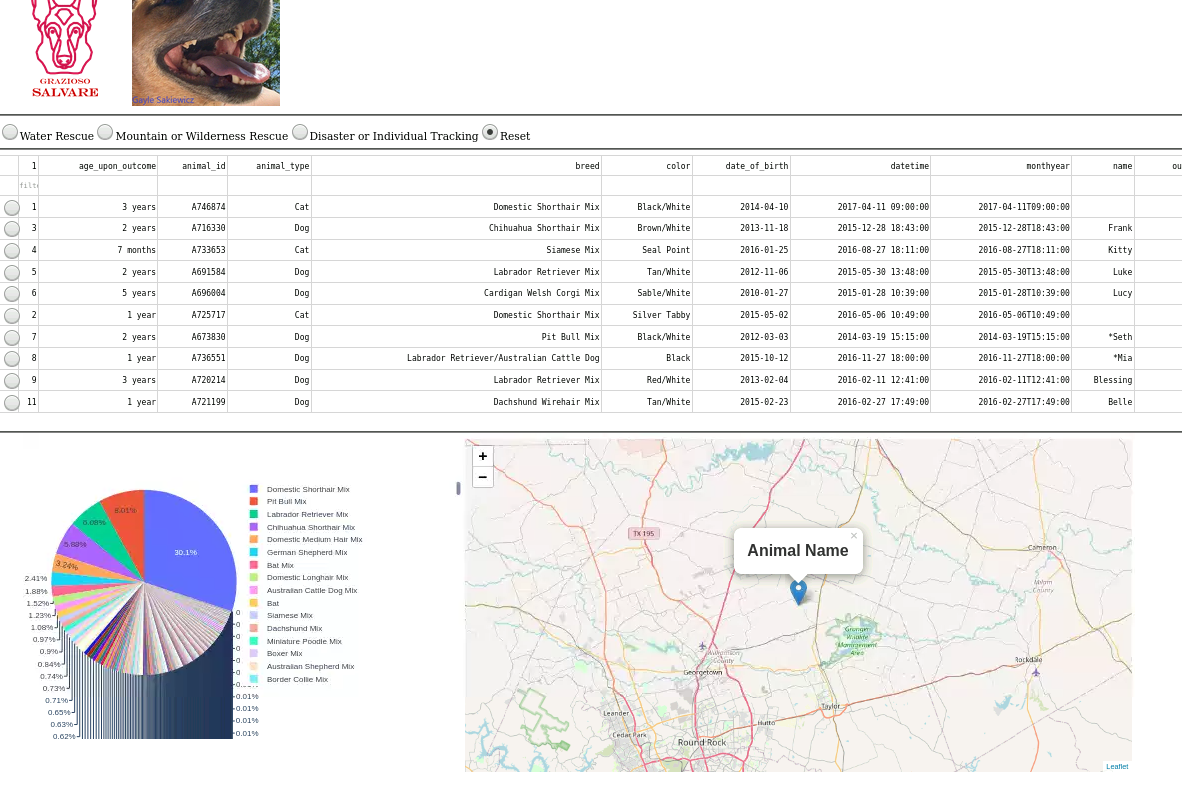
“Mountain or Wilderness Rescue”:



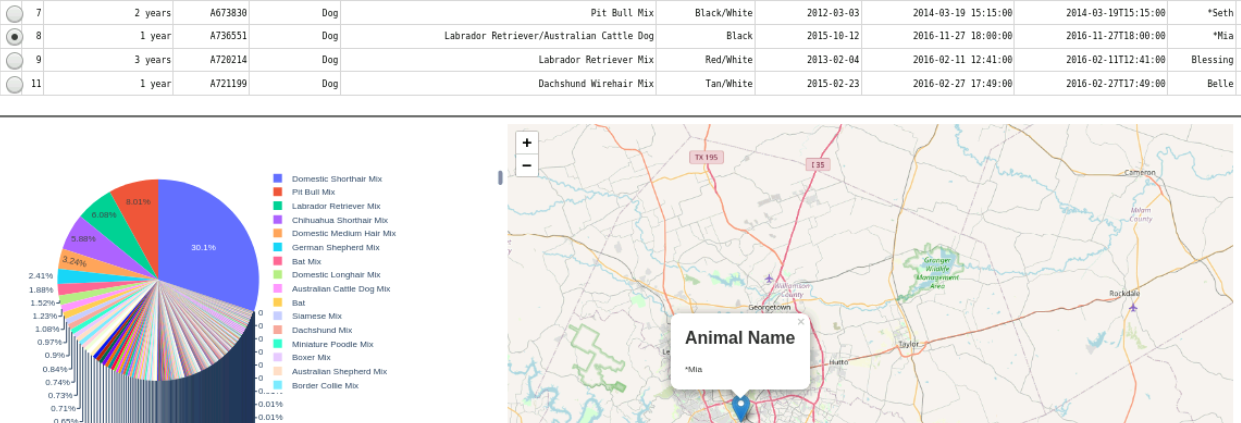
“Disaster or Individual Tracking”:



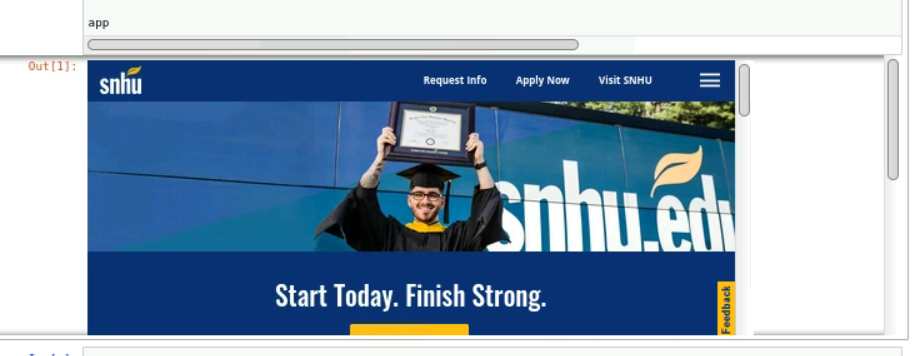
Reset:



Select a row and it updates on the map (with any radio option):



Company logo opens their website when clicked:

## Challenges

I faced challenges when populating the pie chart, at first it took many tries just to get one to show, not having the variables done correctly – in this case, I didn’t need “values” as they were implied and for “names” used ‘breed’ from the database. Then I couldn’t get it to update with the radio options, so I made the dataframe for each one a list with the read/find method from my CRUD module.

## Contact

Gayle Sakiewicz