# CS 340 README Template

## About the Project/Project Title

The project is to create a full stack development of a software application for the software engineering company, Grazioso Salvare. The goal is to determine and place available dogs into categories using existing data from the animal shelters. The purpose of the CRUD Python module is to develop a better understanding of the techniques found from the fundamental operations of CRUD. In this development, the CRUD Python module is used to create databases of document collections to seek out documents within the database system of the animal shelters. CRUD stands for create, read, update, and delete. These functions help in various operations and can assist many companies in managing data in their databases.

The module should be used in a manner that recognizes the functionality of the database and CRUD operation. Additionally, pymongo was utilized in this project and requires the programming language Python. Pymongo recognizes different tools that interact with the Mongo database. Python is an object-oriented programming language and is used widely by many software engineers as well as highly recognized in the tech industry.

## Motivation

The motivation behind the creation and maintenance of the project is to search for good candidates of dogs to train to become search-and-rescue companions. With MongoDB, the database from the animal shelters can be quickly and easily setup. Using the CRUD functionality for the animal database, the client, Grazioso Salvare can manage their navigation and records of data.

## Steps

To obtain a local copy and get the program up and running, please follow the following example steps:

1. Create a simple Mongo Database named “AAC”.
2. Create administrator account with full privileges to the “AAC” database.
3. Enable user authentication for the database using:

mongo --port xxxxx  --authenticationDatabase "admin" -u "admin" -p

1. Import data from the file “aac\_shelter\_outcomes.csv”.
2. Develop Python module in a PY file for CRUD operations in Jupyter Notebook.
   1. Create a method for create.
   2. Create a method for read.
   3. Create a method for update.
   4. Create a method for delete.
3. Create and run Python testing script in Jupyter Notebook.
   1. Make any necessary changes to ensure the code is functional.
4. Create Dashboard web application.
   1. Include chart of choice, geolocation map, widgets, and interactive data table.
5. Create callback for database queries and map updates.
6. Run the ipynb file and ensure the dashboard is fully functional and includes the necessary features, widgets, and tools.

## Installation

*List the tools you need to use the software and how to install them.*

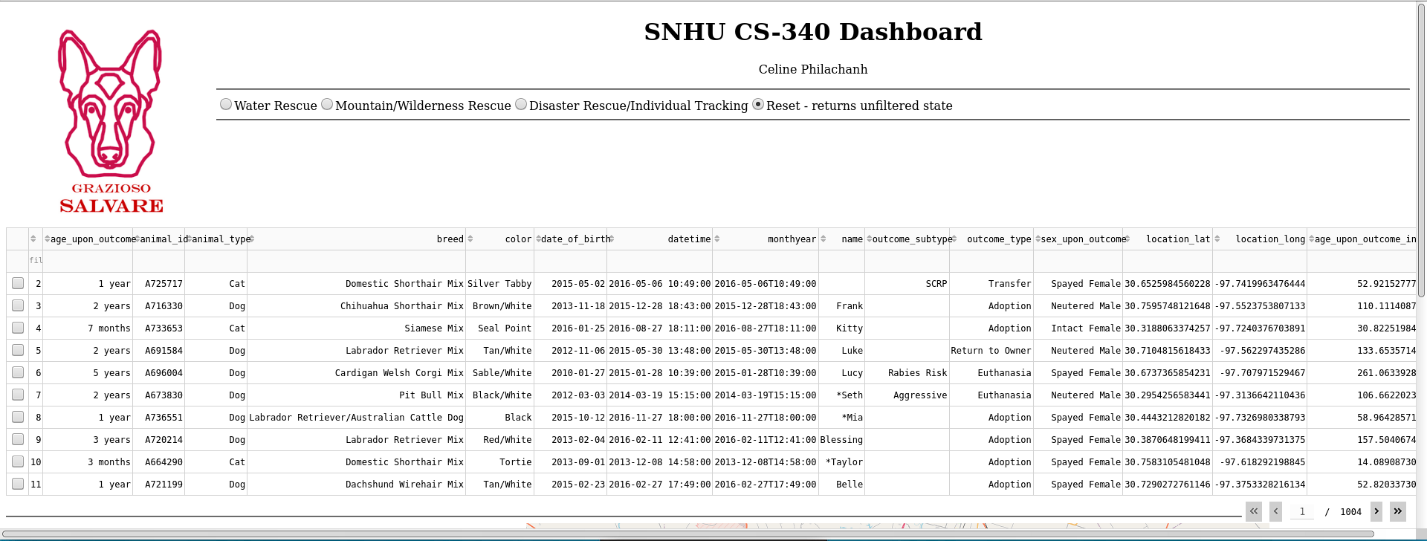
* MongoDB – To install, simply go to https://www.mongodb.com/docs/manual/installation/ and select the desired operating system.
* Pymongo – To install, simply go to <https://pymongo.readthedocs.io/en/stable/installation.html> and follow the manual instructions for installation.
* Python – To install, simply go to <https://realpython.com/installing-python/> which has detailed instructions on installing the program.
* Jupyter Notebooks – To install, simply go to <https://jupyter.org/install> and follow the installation instructions.
* Dash – This is a framework that can be installed by simply going to <https://plotly.com/python/getting-started/> to get started.
* Plotly – To create charts such as the pie chart, plotly is a tool that can assist with this need. Simply review the instructions and documents at <https://pypi.org/project/dash/> and follow the installation instructions provided on the website.

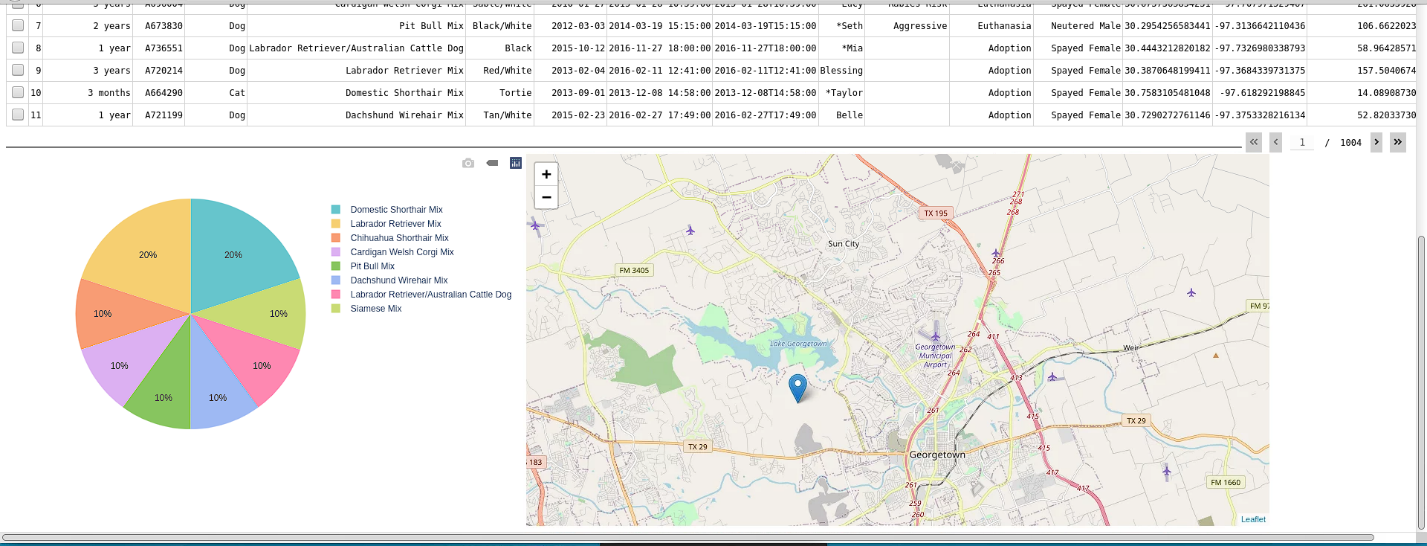
These tools were used because of their simplicity with navigating as well as being portable. Such that Jupyter notebook contains folders in which the user can better organize their files. MongoDB is also a great tool since it has everything that the company will need to store data. Python is one of the popular coding languages and with its use in this project, there is no doubt that it is easy to use and download. Furthermore, pymongo works with MongoDB in a manner that enables users to use the python language in the mongo shell. As for the Dash framework, it provides the view and controller structure for the web application. This framework helps ease the stress of the user since it is easy to use and has features and tools that make up the product. It is essential to have since it makes the application more appealing and aesthetic.

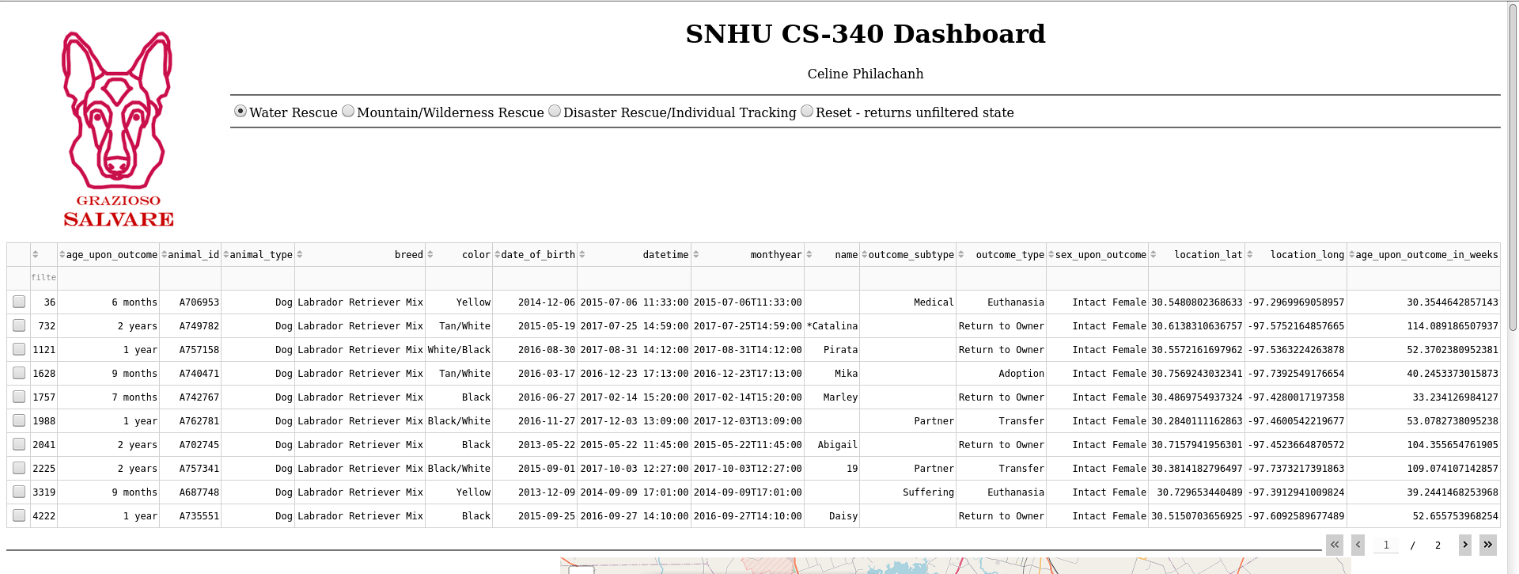
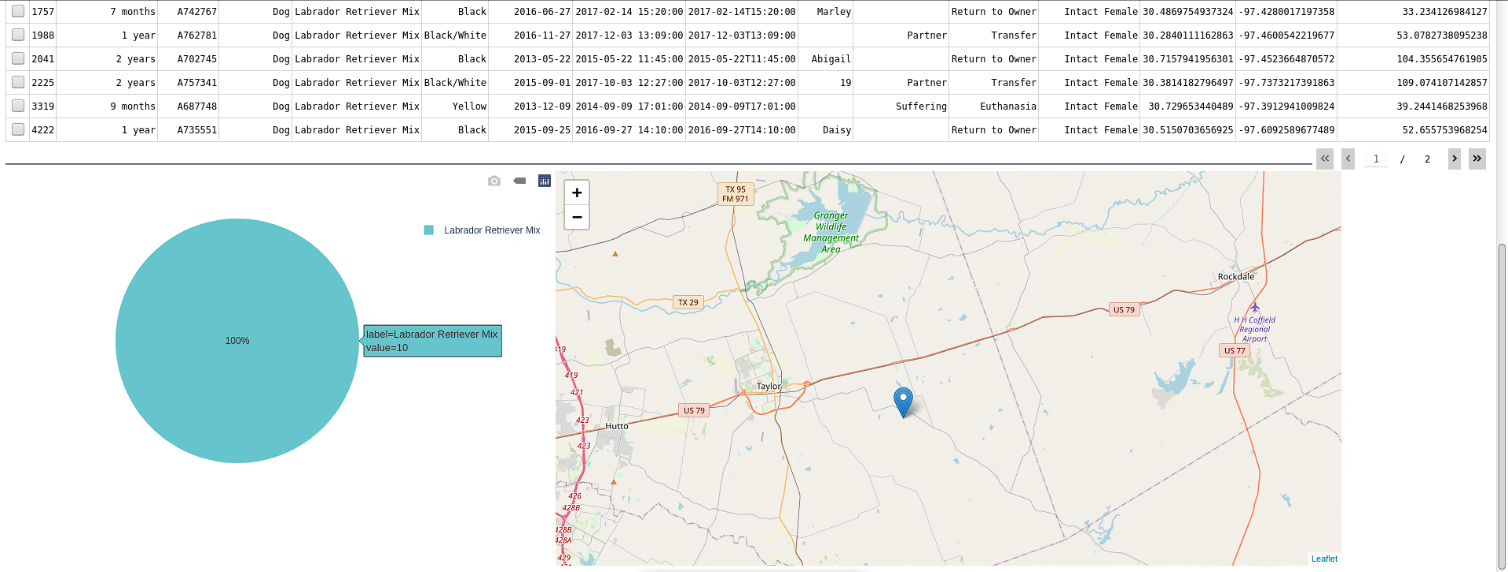
**Challenges**

While building the Dashboard web application, I encountered a challenge though, I was able to come overcome it. The challenge was formatting the chart and geolocation map so that they would be side by side. I was able to overcome this by going back to my readings and taking notes that I can refer to when necessary. Another challenge was Apporto itself. Apporto was not as reliable since it can get really slow at loading data to the screen. This in turn is time consuming when it could have been faster had I done the project on my personal PC. Despite the challenges, I had a great time constructing this web application and I do believe that my hard work paid off.

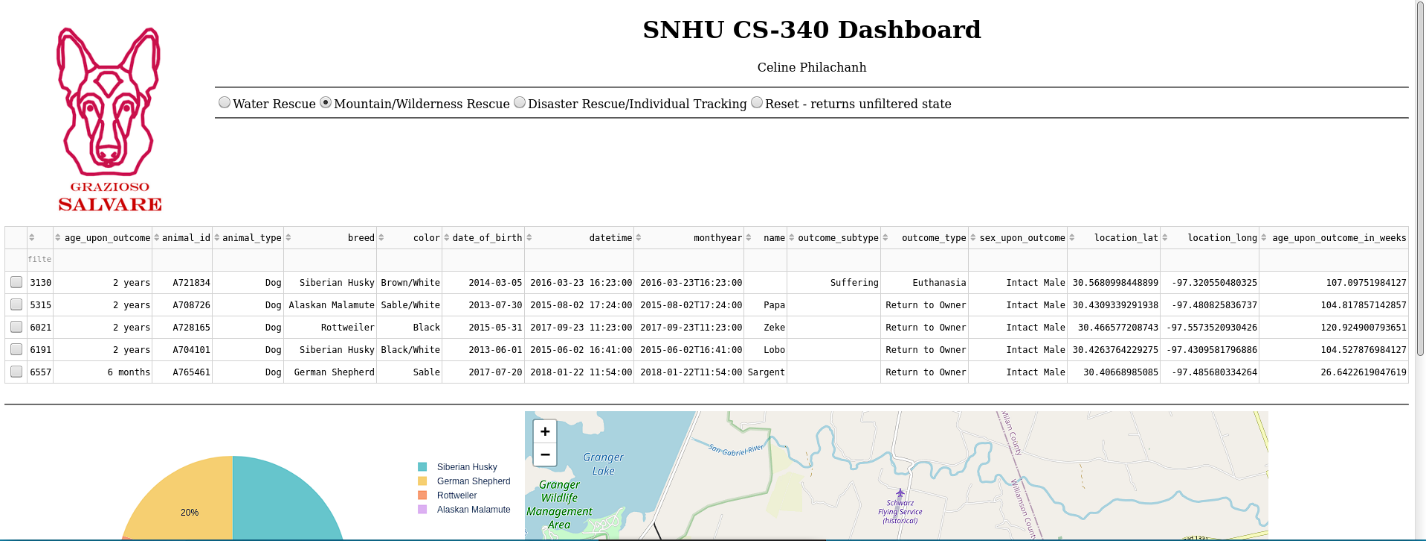
### Screenshots

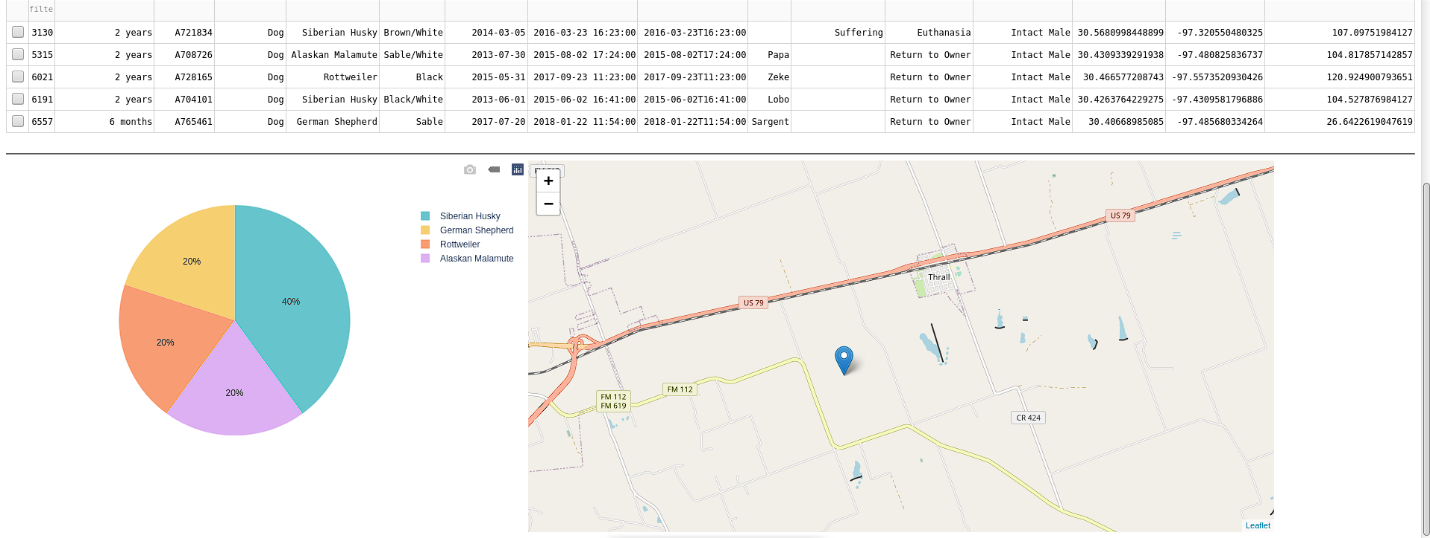
The 2 screenshots below are the starting state of the dashboard:

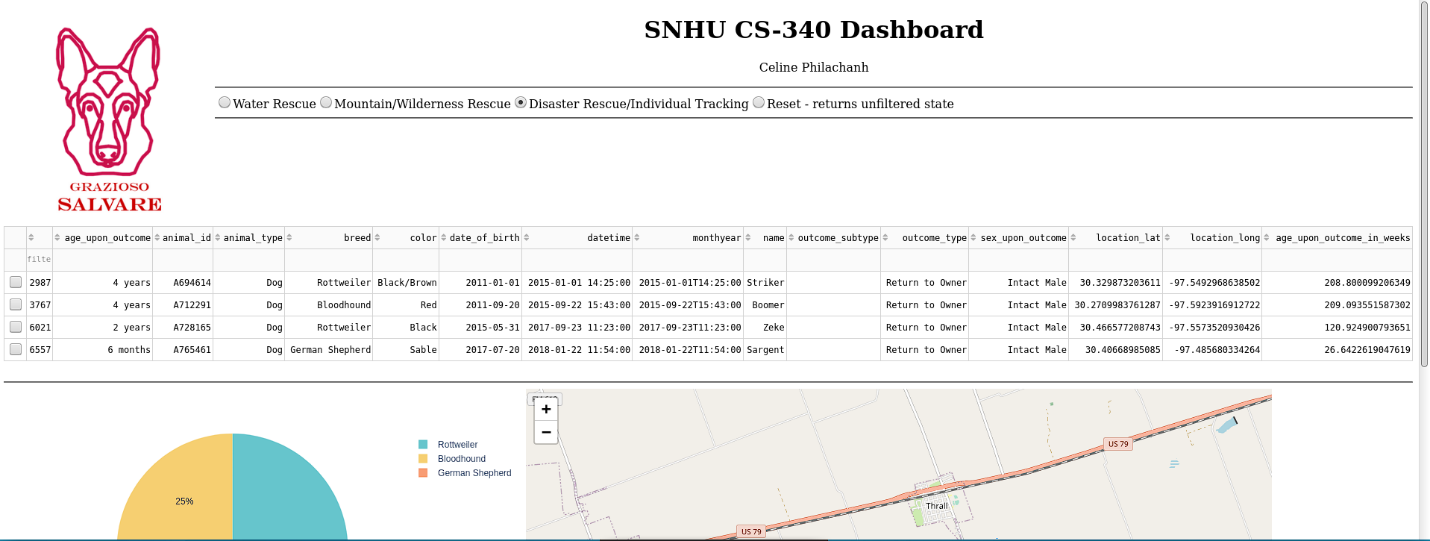


**Water Rescue execution**

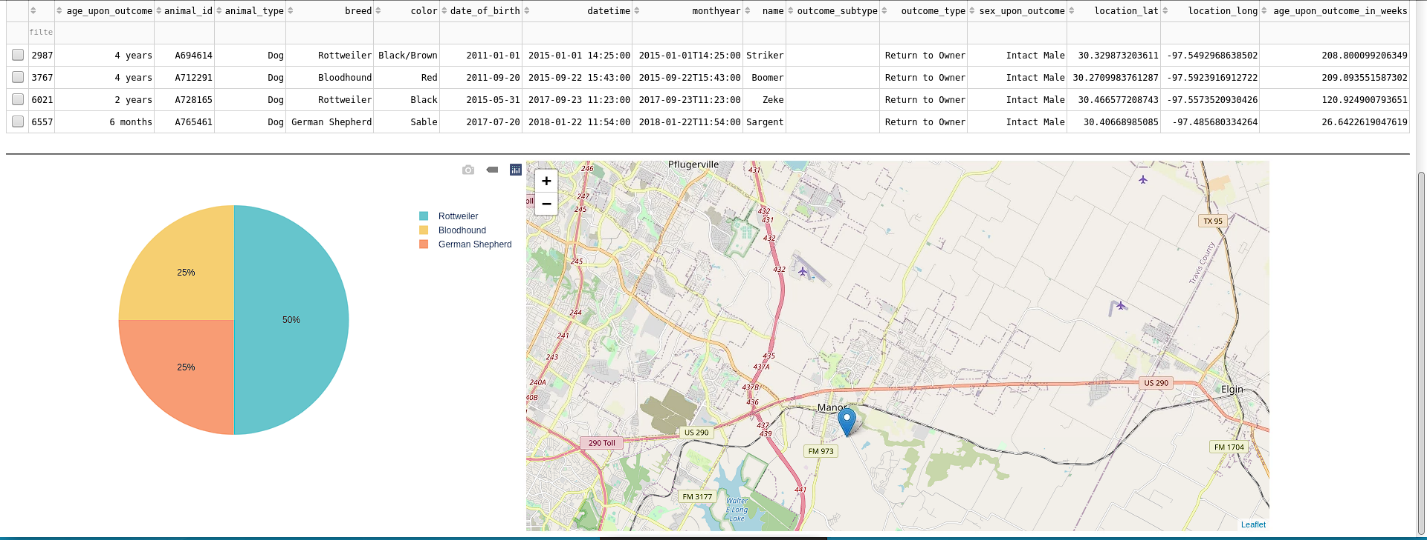
**Mountain or Wilderness Rescue execution**

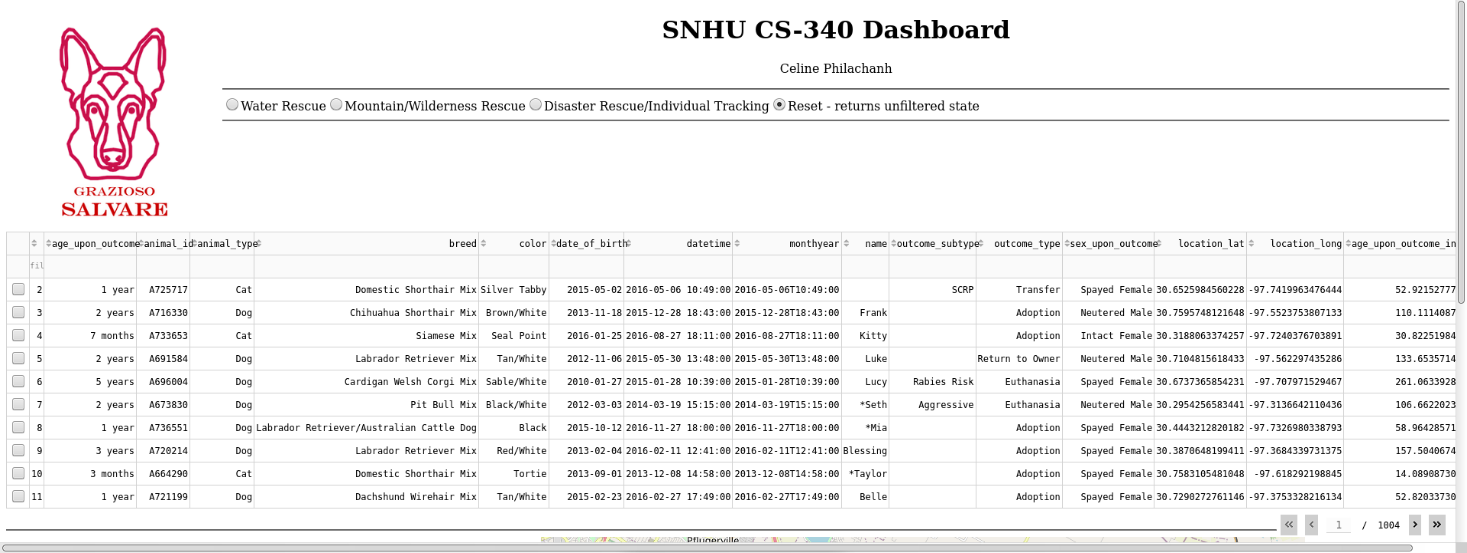
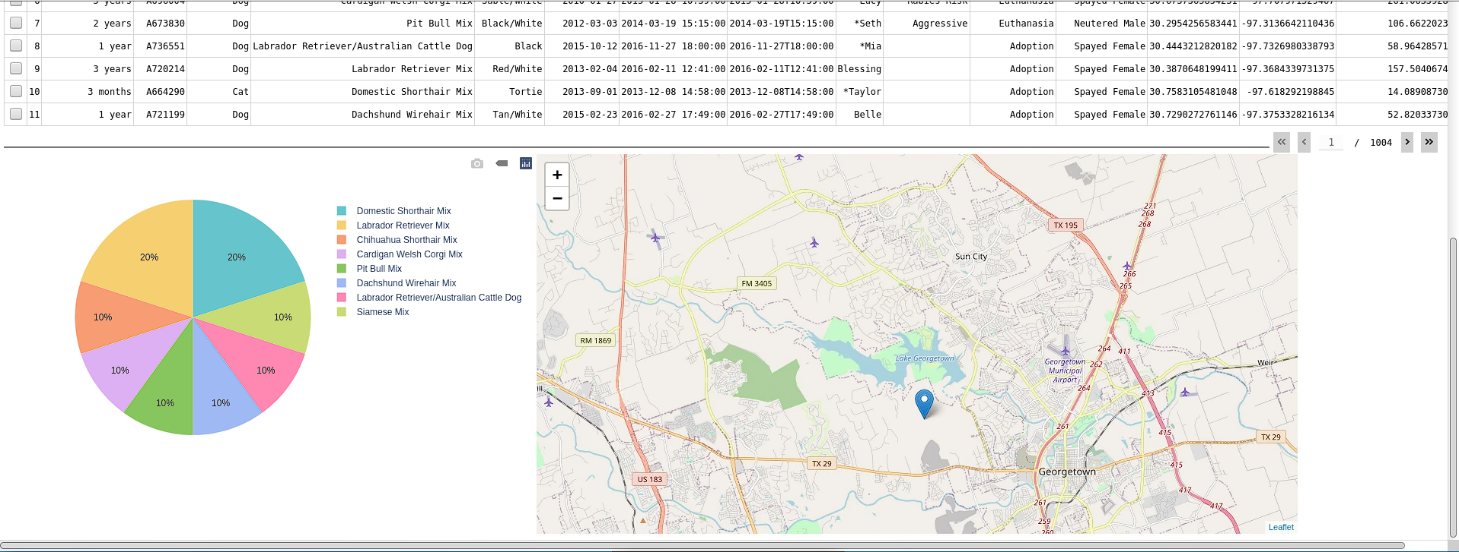






**Disaster or Individual Tracking execution**



**Reset (returns all widgets to their original, unfiltered state) execution**

## Contact

Your name: Celine Philachanh