Elliot Collins

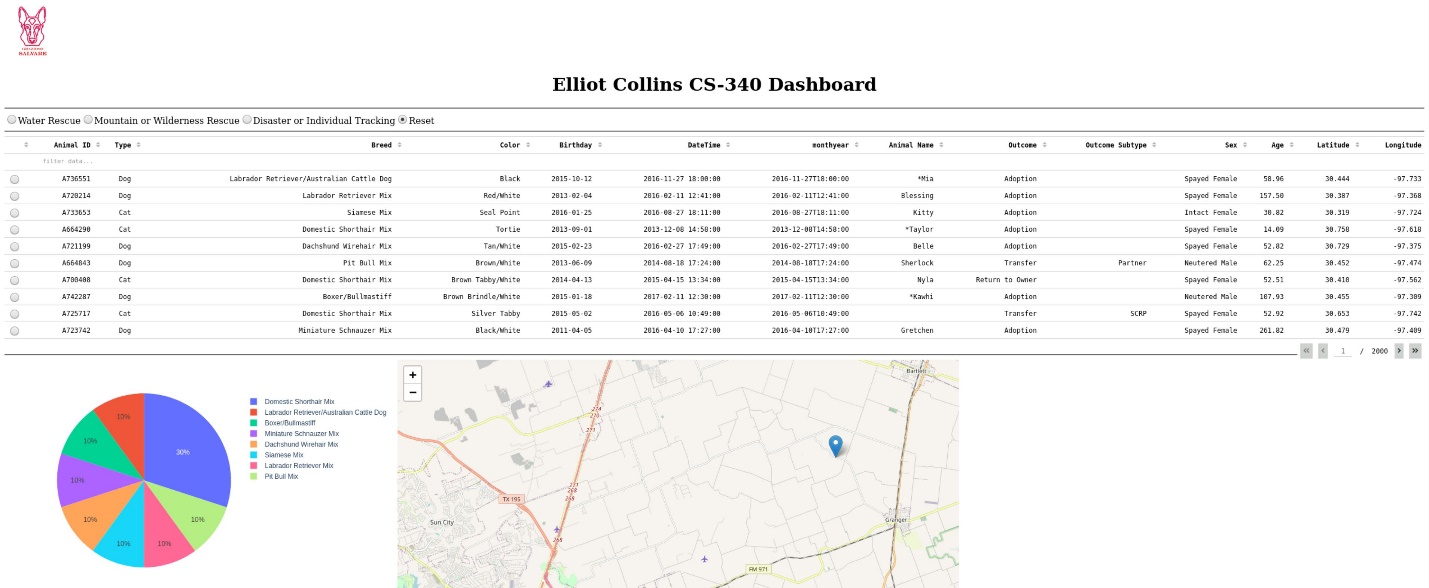
CS340

Project Two

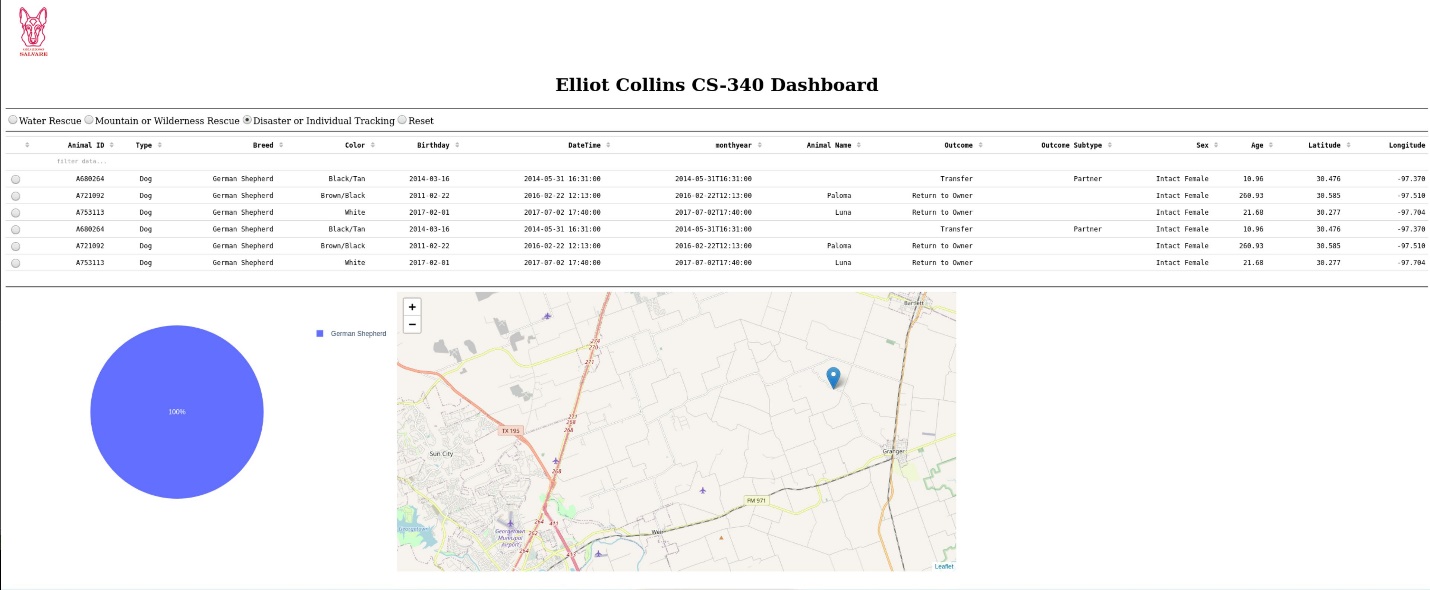
**Required Functionality**

I, a developer for Global Rain, was hired by Grazioso Salvare to provide an interactive database dashboard for an animal shelter in Austin Texas. The purpose of this program is to provide a user-friendly display of all of the shelter’s animals and the ability to filter the animals to meet the company’s needs. The program meets the requirements of, an interactive data table of the database, a graph view of the data, a geolocation map of the data, and the ability to filter the data. In this case we provided filtering option that would display animals suited for certain disaster rescue scenarios, using radio items above the data table display.

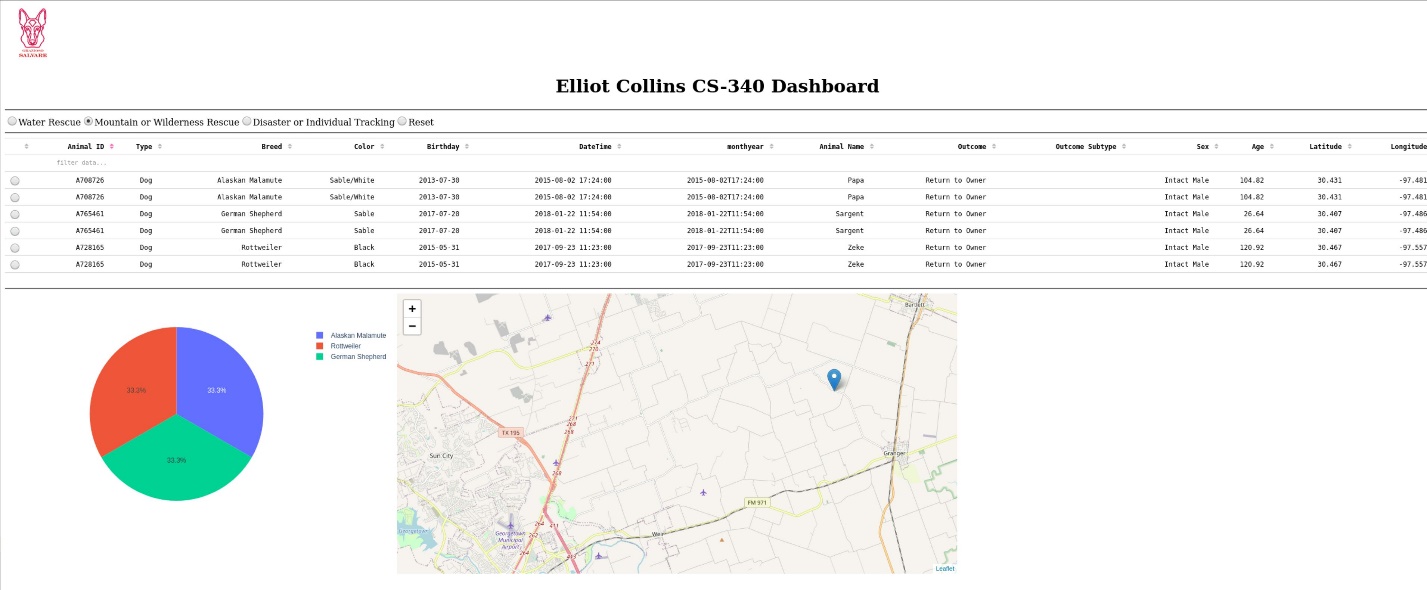
**Dashboard without Filtering**

****

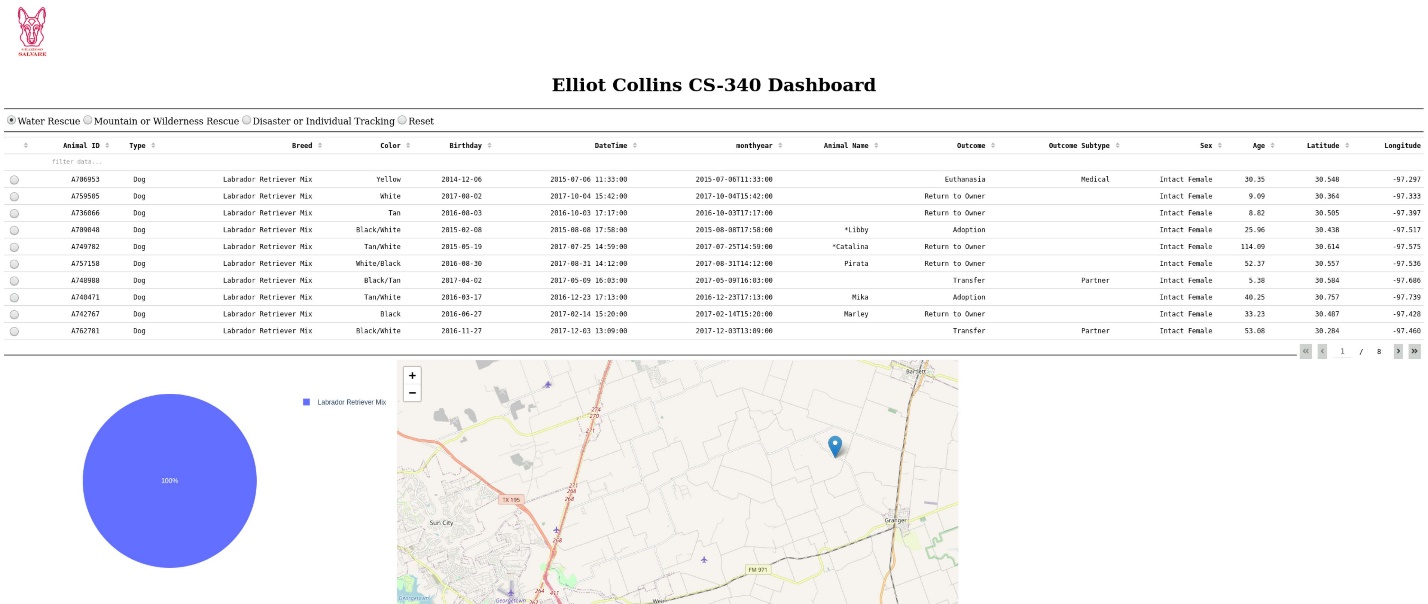
**Dashboard with Disaster or Individual Tracking Filtering**

****

**Dashboard with Mountain or Wilderness Rescue Filtering**

****

**Dashboard with Water Rescue Filtering**

****

**Tools Used**

We chose to use MongoDB as the home for the data for our program because it is a database that can easily be interacted with via python, this interactive relationship is known as PyMongo. We are able to use python in hand with the Dash frame work to import the data from MongoDB and build UI/UX tools. Python is also used to query data from the database and feed it into the user interface functionalities.

[PyMongo](https://pymongo.readthedocs.io/en/stable/)

[Dash Framework](https://plotly.com/dash/)

**Steps Taken**

A lot of the work done throughout this course has been reused and/or implemented into this final dashboard project. The nature of the project’s modularity really lent itself to working this way, completing different pieces of functionality as assignments and then combining them into a larger program at the end.

The first step taken was creating the AnimalShelter Python file that initialized all of the CRUD methods, allowing us to start manipulating and accessing the data in the MongoDB animal database. The next step was building the project one dashboard, which was a more bare version of the project 2 dashboard, In project one we created a data table that presented the data from the MongoDB using the python CRUD methods. It also interacted with a geolocation map and had some simple filtering options, I created buttons that filtered by animal type.

The final step was project two. The functionality required in addition to the data table and the geolocation map, was the ability to filter data with more specific queries with multiple conditions, as well as adding a chart view that updated as the data table was updated. Creating the filtering options was the most complicated step as I did not expect how much work was needed to create working queries with so many conditions. For example, I thought that I could simply create queries with my “read all” command and all of the conditions for each query, as show below:

Text

Description automatically generated with low confidence

However, I continued getting error messages saying I needed to create an index in order for this to work. So then , after some research I learned how to, and implemented an index for my data:

Text

Description automatically generated

Once the functionality of these queries was coded into the UX of the radio items, I was finally able to update my dashboard correctly, changing the data table to the appropriate animals, and having the charts change accordingly.