# CS 340 README by Chloe Ninefeldt

This project is about making a database system that can have the animal shelters information uploaded to it, then easy accessed by an end-use. It is to track the animals and find those that can be used for search and rescue. This full-stack application gives a visualization of the animals, where they are located, and what breed they have in the shelters.

The motivation behind this project is to give the client, Grazioso Salvare, capabilities to find search and rescue dogs. Mr. Salvare would like to filter dogs by their breed as certain breeds are useful for certain types of search and rescue missions. Certain dogs can be used for water rescue, mountain or wilderness rescue, and disaster or individual training. The dogs also need to be under 2 years old. From this information, I was able to make code that could filter through the different search and rescue type missions that brought up the certain breeds for the certain missions. Being able to do this makes it easier for Mr. Salvare or whoever is using the database to be able to find dogs faster for certain missions.

In order to have all of this information, the data was stored using MongoDB due to efficiency and reliability. From there, a CRUD module needed to be created using Python for the end-user to have abilities to create, read, update, or delete entries. MongoDB and Python work rather well together making this a smooth project for both to communicate with one another. For the geolocation chart, Dash Leaflet was used and it a useful tool to have. Then Plotly Graph Objects was used to create the pie chart. There were some calculations to be done by plotly, but it makes the end-user see visuals which helps when analyzing data. HTML was used to create the dashboard to again, give the user a visualization of the data. In order to actually view the application, the Dash Framework tool was used. All of this comes together by having the HTML elements rendered within Python, making the dashboard able to come. Pandas is also used to bring up the data visualization functions for the end-user.

Here are some links that helped me learn about all of the criteria above:

1. For understanding Dash Core Components - <https://dash.plotly.com/dash-core-components>
2. Understanding how Dash Datatables functioned - <https://dash.plotly.com/datatable>
   1. This was super useful as it helped me set up the table all together. Including how columns were made, how to style it, formatting, making it editable, and filtering.
3. I also watched YouTube with understanding MongoDB overall - <https://www.youtube.com/watch?v=E-1xI85Zog8&t=563>

To get this project running, there were many milestones to be achieved. First the information needed to be uploaded to MongoDB. Using mongo, an index for breed was created and the aacuser was created with reading and writing capabilities. Switching over to Python, the CRUD module needed to be created. While doing this, there was a testing script also created to verify that the CRUD module did indeed work properly. After testing was completed successfully, the dash application code was started. Importing the CRUD module, hardcoding the username and password for aacuser, creating filters and radio buttons, adding the Grazioso Salvare Logo, creating features, charts, and adding a geolocation completed the table. The application was finally ready for use, and I will walk you through how to use it below.

To use this application, it starts out looking like the image below. This image has the data at the bottom of all the animals in the shelter, with the filtering options at the top of the data, then more filtering options in the top left. The Grazioso Salvare image is on the right-hand side with the title at the top.

Text

Description automatically generated with low confidence

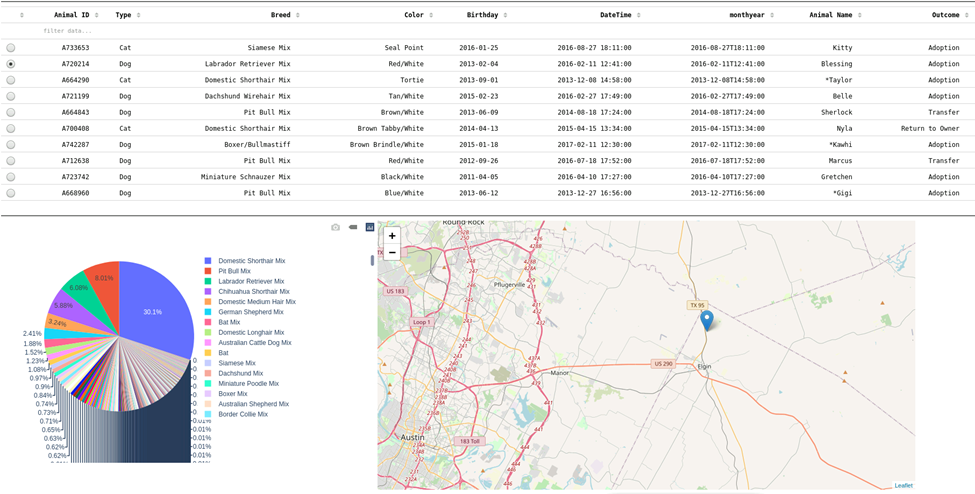
If the user clicks “Water Rescue” to filter, this will show up next:

A picture containing text

Description automatically generated

The options displayed are the breeds that can be used for water rescue, as you can see the data list has changed from the first image to this one above. The user can switch between any filters, add age in the bottom, or reset the data completely and start over.

If you want to know the animal’s location, simply select the animal on the left. Once selected, the pie chart and geolocation map will appear. Showing the percentages of the breeds in the shelter, as well as where the animal selected is at. See image below for an example.



The user can reset the information and continue selecting animals based on what they are looking for. Again, can be filtered in many different ways so the user can find exactly what they need.

Here is the full photo of the dashboard when an animal is selected, as well as showing the data being reset rather than filtered by “Water Rescue”.

A picture containing table

Description automatically generated

This project was a rather challenging one as error codes popped up left and right. There were syntax errors, indentation errors, or the code just not working in general. I was rather unfamiliar with a lot but learned a ton as the milestones went on to get me to the final project. There were many times I had to break this project down into tiny pieces in order to find said errors as the error message would say “line #” is incorrect, when it had nothing to do with that line, but there was an error 20 lines above it making it not function. I reached out to Stackoverflow a couple times to figure out what I had done wrong and majority of the time I had typos that I couldn’t see. Analyzing my own work is difficult, but I figured it out to make a project that’s usable and functions. There are many things I would like to add, like more filtering options, better charts, etc but I would need to learn more. It’s something I can expand on.

## Contact

Your name: Chloe Ninefeldt