# CS 340 README Custom Dashboard

## About the Project/Project Title

The project was done to build a web-based dashboard for Grazioso so that they can review animals at the Austin Animal Shelter.

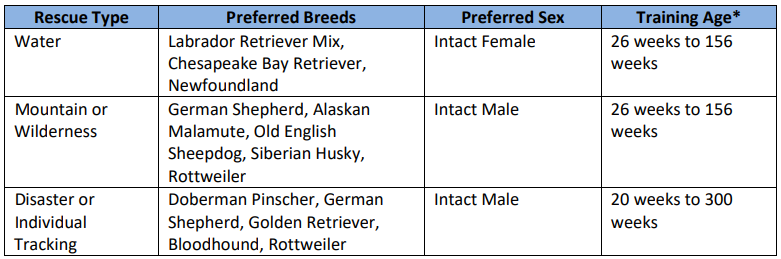
* 1. There are layers to this project to make funcitonality
     1. MongoDB that contains the data.
     2. Python used for control.
     3. A Plotly, Dash, Leaflet layer to make the html portion.

The Python portion first connects to a mongoDB server and then uses the Create, Read, Update, and Delete functions made in the AnimalShelter class.

## Motivation

Grazioso requested specific aspects of the html server to have; these requirements are shown below.

* Branding with logo
* A link to their website
* Interactive filter options (buttons, drop-downs) to filter the Austin Animal Center Outcomes data set by:
  + Water Rescue
  + Mountain or Wilderness Rescue
  + Disaster Rescue or Individual Tracking
* Reset (returns all widgets to their original, unfiltered state)
* A data table which dynamically responds to the filtering options
* A geolocation chart and options and chart of your choice (such as a pie chart) that dynamically respond to the filtering options



## Project requirements

You must have the following software installed on your machine to use:

* Python (most recent version)
* Pymongo
* MongoDB
* Plotly Dash
* Dash Leaflet
* Data imported via CSV
* User accounts with read/write permissions

## Component Selection

MongoDB was the selected part of the development since is has a nice ease of use and a ton of flexibility. One of the features we can take advantage of is that the database platform will not allow any errors during an update or creating operation.

Python was chosen for its availability to connect to the back end and the front end of the of the environment. Python also allows for the use of java based dashboards.

Plotly-Dash, Dash Leaflet libraries are wrappers for JavaScipt tools that work with Python. With these we are able to use the (CRUD) class of Python and work on the GUI (graphical User Interface) more attractive and user friendly.

## Creating a user and password

First open the terminal in the Linux shell, then type “mongosh” this allows you to go into the mongo database. Then, type what is below, the database will respond with “{ ok: 1 }” showing you have made the user and their restrictions properly.



## Code for CRUD Operation

The image below displays the actions made to connect to the database. In this we used the environment variable to allow connection as this was made on a virtual machine since available computer was windows operating system. The username and password for testing were “aacuser” and “qwerty”. We connected to the database named “AAC” with the “animals” collection.

A screen shot of a computer code

Description automatically generated

Next is the create operation to be able to add an animal to the database.

A screen shot of a computer

Description automatically generated

Below is the read function to display the desired or selected animal in the database

A screen shot of a computer

Description automatically generated

Next is the update or possibility to change the stored information in the database.

A screen shot of a computer

Description automatically generated

Finally, is the delete operation to take animals out of the collection in the AAC database.

A screen shot of a computer

Description automatically generated

## Code for html Operation

First, we see the code that imports the code and the class for the CRUD operation



Next, we look at the input of the username and password that are the variables we used to access the html server

A black background with white text

Description automatically generated

Below is the input of their logo which was requested.



We can now see the functionality of a drop box to select the variables they requested.

A screen shot of a computer program

Description automatically generated

This is an included chart to make the variables more visual and data easier to read.

A screen shot of a computer code

Description automatically generated

This is the code for the map that shows location of the animal.

A computer screen shot of text

Description automatically generated

### Problems

In the creation of the project the main problems came form the CRUD class. Lack of skill in a database and lack of skill in a Linux vm has humbled me as the creator of this app. After seeking professional help from my professor I was able to learn my mistakes and correct them. This caused some problems to occur with the html server connection which I was able to work on and write code to get the displays.

## Contact

Matthew Ellsworth