CS 340 README FOR DASHBOARD

## About the Project/Project Title

*The Power of Python in MongoDB: The use of python coding to create a dashboard that access Mongo databases and interprets them*

## Motivation

*The reason for the creation of this README is to show the power of Python modules when working with Mongo and how it can help database entries in a timely and efficient manner.*

## Getting Started

*Get set up with a PC that has a Linux operating system or create a VM which runs a Linux operating system. Once this is done get the listed below software installed.*

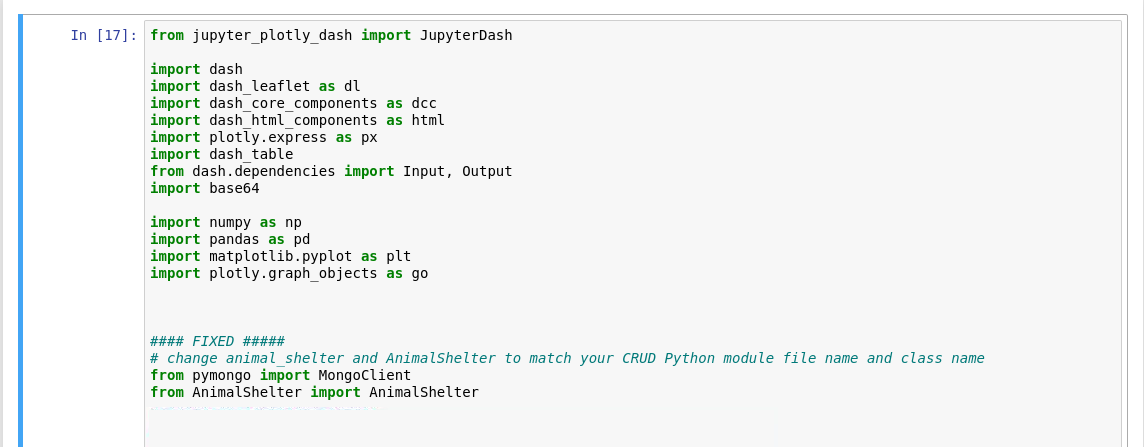
## Installation

* *MongoDB shell version v4.2.6 (currently)*
* *Python 3.6 (currently)*
* *Jupyter Notebook*

**Prequel**

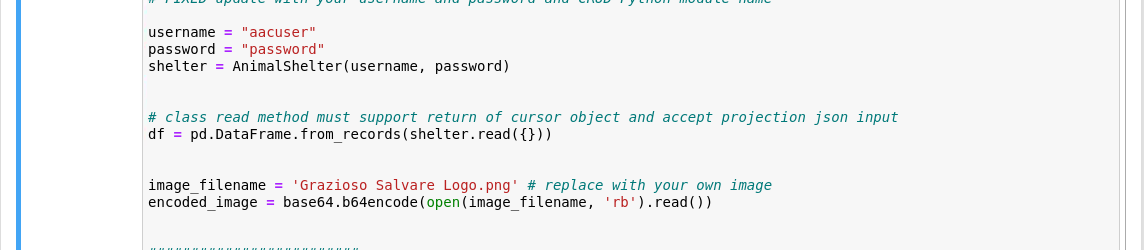
Please reference README FOR PROJECT 1 to make sure AnimalShelter.py is created and setup correctly.

## Usage



Make sure to import all libraries and files used for this project.

Also make sure the AnimalShelter.py is in the same location as the IPYNB file being worked on so that it doesn’t have issues accessing it.



We will now setup a username and password attribute to access the Mongo Database (for my Mongo database the username is aacuser and the password is password).

After doing this we will create an attribute for the Grazioso Salvare Logo so that it can be apart of the dashboard. Make sure the name of the png file is exactly the same and in the same location as the AnimalShelter.py and IPYNB file.

We will now work on the Dashboard Layout



The html.Img() line brings the encoded image we work on earlier into the dashboard.

The dcc.RadioItems() allows filters to applied to the code.

The above code also allows the dashboard info to be edit (from editable = True) and also be filtered (filter\_action = “native”). The code shown below html.Br() allows for the Map and graph to be a part of the dashboard(we will work on the code for the map and graph later).

The below code shows the app.callback for the Styles and Graphs. You should already have the code for update\_styles from the ProjectTwoDashboard.ipynb given to you so we will talk about the update\_graphs definition.



The first line in Update\_graphs pulls all the data from the AAC database and sets it up as variable dff.

Next we tell it to return a graph (ddc.Graph), in the px.pie function we first tell it to access the dff variable and then specifiy the parameter we want compared in the screenshot we see names = “animal\_type”, we can change this to names = “breed” or any other type of info you see as useful and lastly we specify the title of the graph.

The last screenshot of code will be about the map definition

Graphical user interface, text, application

Description automatically generated

To start off we do the same as the graph definition and pull the info into variable dff. In the code you can see we have the map starting location as Austin, Texas. It then pulls item 13 and then item 14 from the first variable in the dashboard. Once this is pulled a Pin will be dropped where their location is and when you hover over it you will see the breed. If the Pin is clicked on it will show the animals name. The below screenshot are of the IPYNB after being run.



Table

Description automatically generated

Graphical user interface, chart, application, pie chart

Description automatically generated

The dashboard can be filtered depending on what you are looking for, in the below screenshots you can see it being filtered by animals that are 7 years old.

Table

Description automatically generated

You can see the graph also updates based off the filtering.

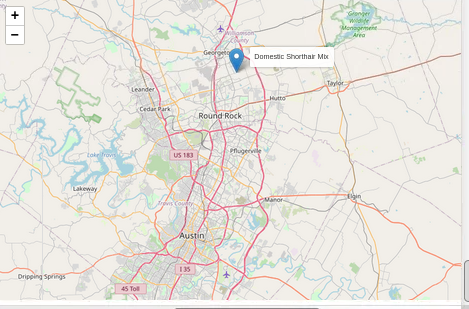
Chart, pie chart

Description automatically generated

The map will also pull the first entry in the dashboard so we can confirm this by going to the map and seeing if it’s Domestic Shorthair Mix.

Table

Description automatically generated



## Contact

Your name: Trenton Walty