**About the Project**

The purpose of this application is to provide a user interface to view data stored in a MongoDB database and return desirable combinations of breed, sex, and age to training rescue dogs. This project contains a database, API and a dynamic dashboard.

**Why MongoDB**

I chose to use Mongo as it provides quick setup of the database from a csv file and has a very python friendly interface. Python can utilize tools like SQL, but due to syntax it can be difficult switching from one to the other. The selection tools when using CRUD functions of a database are more complex than those contained within SQL allowing for more freedom with how we create the database and interface.

**Getting started**

To create a local copy of this program you can follow this guide. First you will need to create a mongo database. Then you will need to create a Python CRUD module to access the said database. Finally, we must create a dash web application that uses the python crud module.

1. Create Mongo database called AAC
2. Create user profile with read/write privileges in that database
3. Import data from aac\_shelter\_outcomes.csv file
4. Update the accuser and password to the username and password you created
5. Run the tests and add code to Jupyter notebook to ensure that the create function is different each time and to test the delete funciton
6. Create a new dash web application dashboard and configure it to suit your needs, ensure that there are appropriate ids for the dataframe, map, and chart
7. Create app callback to populate the initial dataframe
8. Create radial options and program the database queries based on the desired breed
9. Create application callback to update the map with first item of a given category until the user selects an item.
10. Now create functionality that determines the user selection and shows it on the map instead
11. Create a pie chart from the displayed data on the screen at any given moment. With the pie chart, create an application callback that updates said chart with data that is being filtered via the users chosen display filters

**Installation**

You will need Jupyter Notebooks, Python, and MongoDB installed to run this project.

**Usage**

This application has four primary functions. Firstly, it obtains data from the user regarding what to enter the database or what to search for. The second function is running the creation and search function. The obtain functions ask the user for specific input. The create function operates via a python directory being passed into the MongoDB as a new item and returning Boolean True if it is successful or False if it isn’t. The read function returns false if nothing is available or returns data if matching data is found.

**Code Samples from Dashboard**

Complex Query:

A screenshot of a computer screen

Description automatically generated

Map Markers:

A screenshot of a computer code

Description automatically generated