# CS 340 README Template

## About the Project/Project Title

While this project does not have a specific name/title, the purpose of this project is to implement CRUD functionality for the Austin Animal Center (AAC), allowing users to successfully create and read with a python executed MongoDB job. With the addition of the user interface, it can now be used to compare data and keep track of the data in a far more organized fashion than previous iterations (samples below).

## Motivation

This project exists in order to provide a simpler and more convenient solution for database management when it comes to the Mongo Database system. It will carry out CRUD operations for the end users.

## Getting Started

To get a local version of this up and running you will first have to implement the specified Austin Animal Center (AAC) database into the Mongo system. Afterwards you will need to create an administrator account in the mongo shell by following steps #2–3 of the MongoDB [Manual Enable Access Control](https://www.mongodb.com/docs/v4.2/tutorial/enable-authentication/) tutorial. Then exit the mongo shell.

You can verify that you have enabled user authentication by accessing MongoDB with your new username/password. Type the following command into the Linux shell to start mongo:

mongo --port xxxxx --authenticationDatabase "admin" -u "admin" -p

Where xxxxx = your individual port number for mongo.

Then you will create a new user account called “aacuser” for the database AAC in the mongo shell. Refer to steps #6–7 of the MongoDB Manual Enable Access Control tutorial, linked above, to help you with this task. You will need to modify the commands so that the account name is “aacuser”.

## Installation

The tools needed are:

Mongo – Installation: <https://www.mongodb.com/docs/manual/installation/>

Jupyter Notebook – Installation: <https://jupyter.org/install>

Pymongo - <https://pymongo.readthedocs.io/en/stable/installation.html>

Python – Installation: <https://realpython.com/installing-python/>

Plotly – Installation: <https://www.journaldev.com/19692/python-plotly-tutorial#:~:text=Installation.%20To%20install%20plotly%2C%20open%20a%20terminal%20window,to%20install%20to%20collect%20dependencies%20and%20download%20them%3A>

Dash – Installation: <https://pypi.org/project/dash/>

Pandas – Installation: <https://pandas.pydata.org/pandas-docs/stable/getting_started/install.html>

The “AnimalShelter.py” file and the “Main.ipynb” file are required.

The Austin Animal Center dataset file is also required.

## Usage

This application can be used for several things, the first being the sorting of data based on things like breed, age, and sex for different kinds of dogs under the Grazioso Salvare umbrella. The buttons should be interactable and have functionality that allows for database queries which return updated data frames. The existing reset function will reset the data frame to an unfiltered state. The application also has a map with dynamic updates built in. If the user selects items, the map will change to allow for them. The final function is a dynamic pie chart that changes based on the selected data.

## Code & Image Examples

*Text

Description automatically generated with medium confidence*

### A picture containing chart Description automatically generated

A picture containing chart

Description automatically generated

A picture containing chart

Description automatically generated

### A picture containing diagram Description automatically generated

A picture containing text

Description automatically generated

### Tests

*Describe and show how to run the tests with code examples.*

Run tests by entering example data that you would like to insert and then search for it on the Mongo platform to make sure that it exists. To run the tests, add the test code to a Jupyter notebook and ensure that the test data for the create function is different each time or delete the added record between tests.

**Graphical user interface, text, application, email

Description automatically generated**

Create and configure a new Dash web application.

Populate initial data frame with all of the data.

Program the database queries based on the desired breeds.

## Update the map with the first item on the list of a given category until the user selects otherwise

Create a pie chart from the chosen data.

## Roadmap/Features (Optional)

## Contact

Your name: Joshua Perez

SNHU CS-340-H7316