# CS 340 README : Project Two

## About the Project/Project Title

The purpose of this project is to allow users to view the animals database from the Austin Animal Center(AAC) and interact with the data via filter settings, an interactive data table, a pie chart (that shows overall breed distribution for the current filter) and a geolocation map that shows the location of the actively selected animal.

## Motivation

This application was developed for Grazioso Salvare to their design requirements and specifications so that they may easily manipulate the vast swath of data contained within the AAC in order to find good candidates for their search-and-rescue programs. There are different filters that search the database for what have been described as good candidates for Water rescue, Mountain/Wilderness rescue, and disaster rescue/individual tracking.

## Getting Started

Depending on how well set up one is with MongoDB, there may need to be some setup steps taken before using the webpage.

MongoDB setup:

- Enter the MongoDB terminal

- Import the CSV file ‘aac\_shelter\_outcome.csv’

- Enter the necessary user credentials for authentication

\*Note that the python scripts assume the user is ‘aacuser’ and the password is ‘password’. You may need to adjust db\_manip.py and ProjectTwo.py with the proper user credentials if these are not your account credentials.

Python file setup:

- You will need two python files in order to operate/access the webpage, db\_manip.py (which is a helper script that interfaces with mongoDB) and ProjectTwo.py (which references db\_manip to access the database and also defines the different dash components needed for the webpage and runs the server.

- As previously mentioned, the user credentials will need to be updated with your credentials.

*-* There are a few dependencies that are necessary to run the python files, including dash, pymongo, pandas, numpy, etc. These dependencies can be found at the top of both python files and I would recommend installing any that are missing via pip.

It is worth noting that this project was originally going to be written with Jupyter Notebook, but I was getting increasingly frustrated by issues that were occurring due to the kernal staling out or configuration issues. Because of this, I opted to remove Jupyter Notebook from the equation entirely to simplify things, that being said, if the web server is not responding as expected, it may be worth terminating and re-running before assuming an error, I would do the same if *any* changes are made to either python script.

## Installation

The user will need access to mongoDB with an account with sufficient read/write access for the desired database. The user will also need to have python3 installed on their local machine as well. For the different dependencies used, they will need to be installed as well, which I recommend doing via the command “pip install <package>” in the terminal. The dependencies needed are listed here:  
- pymongo

- bson

- pandas

- dash

- dash\_leaflet

- plotly

- numpy

- matplotlib

Additionally, access to MongoDB is necessary. MongoDB is where and how the database is being hosted. Pymongo allows the interfacing between MongoDB and Python, pandas allows for easy management of large datasets via dataframes, dash provides the user interface and the controllers for the web application and plotly is used to build the different visuals.

## Usage

The webpage allows for the viewing and manipulation of displayed data based on the filter options selected. For more information on how db\_manip.py interfaces with MongoDB, I would recommend referencing the readme that was previously created for that program. The ProjectTwo.py script has a lot of different sections that I will break down here. The #Data Manipulation / Model section contains lines that may be adjusted to get the access to the database set up. Here you may input your user credentials as well as adjust the instantiation of AnimalShelter object (detailed greater in the db\_manip.py readme) and it also reads the database into a dataframe for manipulation in the script.

The #Dashboard Layout / View section is where the logic for the webpage is located. Essentially this is where the different parts of the website are built out and specified.

The #Interaction Between Components / Controller section contains the various callbacks and logic used to manipulate the data and update the webpage.

### Tests

To test the program, execute the ProjectTwo.py script by running something akin to “python3 ProjectTwo.py” in the terminal. While the webpage is up, you can reference the terminal for various debug statements, which I have written in to help with troubleshooting if there are issues on any sections. From here, you should be able to determine if the issues are rooted in the actual database access or in the program as it updates the webpage. By default, the webpage will run with debug=True, so errors and issues will also be displayed on the page itself.

Included in the .zip is a screencast that shows all the different aspects of the webpage operating as expected, please reference this for additional code examples and troubleshooting.

**Challenges**

Most of the issues that I faced early on were due to strange configuration problems I was having while attempting to get everything to work properly with Jupyter Notebook so I opted to remove the variable out entirely. After that I faced quite a bit of challenge working with dash as I am not super familiar with the library. In reviewing the documentation and going back and forth between editing and viewing the page, I found a good workflow for hacking away these issues though. Finally, I had some issues with the included line that was intended to drop the ‘\_id’ column. I found it was best to comment it out. I believe in my syntax for db\_manip.py, I already had ensured that this wouldn’t be an issue, so trying to double-down on the problem caused issues. I also would be remise if I didn’t mention the challenge I had in not having my go to IDE for these projects. I am a Linux user primarily as is, so working within the Ubuntu environment wasn’t that bad for me, but I didn’t have things like VS\_Code, so I was missing things I come to rely on such as auto-complete, auto-indentation, etc. Most of my work during this course was written in good ole vim, but I surprisingly started to really like the GNOME built-in text editor, it is surprisingly just feature rich enough to greatly simplify the process.

## Contact

Zach Beachy

zachary.beachy@snhu.edu