# CS 340 README Project Two

## About the Project/Project Title

Project Two builds upon the work of previous projects and milestones to create an interactive table that displays lists of animals based on specific filters.

## Motivation

Grazioso Salvare is an internal animal training company. Part of its work involves the identification of dogs that are good candidates for rescue training. Due to this work, a need exists for a dashboard framework that can filter animal shelter lists to find the dogs for different types of training and display the data in an easy-to-understand format.

These dogs should be filterable by breed, and listed dogs should fall within the ideal age and sex categories for training.

## Getting Started – Part One

Note: Part One is an example of how to utilize the Mongo tool to acquire a dataset. Those with pre-existing data or familiar with the use of MongoDB should refer to Part Two.

Install MongoDB

* See [this page](https://www.mongodb.com/docs/manual/installation/) for installation instructions based on OS

Start with the mongoimport tool to import your dataset

A screen shot of a computer screen

Description automatically generated

Note: Your login information may be different, based on inputs from prior mongo installation. CSV filename is aac\_shelter\_outcomes.csv.

Run mongo



Use command mongosh to start MongoDB.

Create a simple index



This is and the two steps below are for testing purposes. A complex index with more than a single query is also possible.

Verify index creation



Ensure that you have the correct dataset with a quick lookup



Create a new user account



Use the admin database.

A computer screen shot of a program code

Description automatically generated

Enter the username and password you want to use. Ensure that you add the readWrite permissions for the database you want to use as well. Permissions are set after ‘role’, and the intended database is set after ‘db’. Both are shown in the image above.

Logging in

A computer screen with text and numbers

Description automatically generated

Quit out of MongoDB so that you’re back at the regular terminal. Use the cd command to navigate to your datasets. Login with your previously set username and password. Finally, you can use printeng | grep -i mongo to print information pertaining to your login.

## Getting Started – Part Two

Run the Dashboard



Using either Jupyter notebook, or any other Python IDE, run the ProjectTwoDashboard\_King file. Then, follow the given address to the dashboard itself. Ex:  
  


Using filters



The dashboard contains two filters to sort through animal shelter lists. The first is a dropdown menu with the options of cats, dogs, and other animals. The second is a radio button with training types. The latter only applies to dogs, and other animals will not appear on the list when they are selected.

The dog training type filter results will only be those with ideal characteristics for training, specifically age and sex.

Viewing the data

A screenshot of a map

Description automatically generated

Below the data print out, a pie chart and geolocation map are clearly visible. The pie chart will display the distribution of breeds in any given category, including those that aren’t dogs. The geolocator will display the location of any selected animal on the current list, along with their name and identification number.

To select an animal, simply click on the button next to their row.

Potential issues

Be aware that all usernames, passwords, collections, database names, and other fields of information are case sensitive. Setting a user role with permissions on a database named aac will not give that role permissions for a database named AAC.

When initially importing, you will have to replace fields such as “${MONGO\_USER}” with your actual information.

When using the database, it may take some time to load depending on the number of entries.

## Installation/Tools

* MongoDB (see getting started for installation information, or [click here](https://www.mongodb.com/docs/manual/installation/))
  + MongoDB is used to manage to imported database
  + MongoDB has a wide variety of useful tools from creating new databases, managing user permissions per database, and editing information in existing databases
* Any IDE capable of running .py files and .ipynb files, such as PyCharm or Jupyter Notebook.
  + The IDE will be crucial to editing/maintaining the code
  + [PyCharm](https://www.jetbrains.com/pycharm/download/?section=windows) download (scroll down to find a free community edition)
  + [Installing Jupyter](https://jupyter.org/install)
* Dash framework (for more information about the Dash framework, [click here](https://dash.plotly.com/))
  + Dash is a good framework for creating interactive web applications
  + Dash allows for dynamic updating of UI elements with minimal backend coding
  + Dash tools such as Dash-leaflet allows for integration of maps
* Plotly Express (for more information about Plotly Express, [click here](https://plotly.com/python/plotly-express/))
  + Plotly Express is used for a variety of graph objects, and works well with Dash

### Functionality examples:

Filter 1: Regular Dashboard (with logo)

A screenshot of a computer

Description automatically generated

Note: A reset filter was possible, but I opted to simply have it as the default instead of requiring a manual reset.

Filter 2: Water Rescue

A screenshot of a computer screen

Description automatically generated

Filter 3: Mountain or Wilderness Rescue

A screenshot of a computer dashboard

Description automatically generated

Filter 4: Disaster or Individual Tracking

A screenshot of a computer

Description automatically generated

**Project Steps**

Review the specs

I began the project by engaging in a review of the prompt, supporting materials, and starter code in that order.

Begin dashboard

The dashboard was built and tested prior to engaging in further steps, primarily to ensure that any errors would not compound later in the project. Exception handling was added to most functions during this stage.

Develop queries/Build interactive options

The first filter was a drop down, meant as a test to check for functionality. After completion, the radio buttons were added in and both filters were tested together.

Add widgets

A graph was added using the Plotly Express framework, and Dash-leaflet was utilized to include a geolocator for listed animals. The pie graph pulls from the breed category of the dataset. The geolocator takes the name, id, and location coordinates from a selected animal.

Test the dashboard

The dashboard was testing numerous times to ensure that filters interacted correctly, and that data was being displayed in a complete format.

**Challenges**

A slew of errors

Initial builds required a lot of exception handling to avoid empty fields causing problems.

The Great Cat Invasion of 2024

The single most frustrating issue with the build was an insistence by all the cats on the list that they were to be trained alongside dogs. Anytime the filter would be set to dogs or cats, and any sort of training, the cats list would also appear. The dogs notably did not invade the cats list.

This issue took an inordinate amount of troubleshooting to resolve, but the cats have finally agreed to respect their canine counterparts. In the end, there was a single logic statement that had been incorrectly inverted.

## Contact

Your name: David King