# CS 340 README

## About the Project/Project Title

*The purpose of this application is to provide a navigable interface with which the user can sort through the large quantities of data present in the animal rescue database. It features a number of filtering options, which are arranged generally by breed, as well as the option to sort by the various columns that the .csv has. Finally, there are a couple of visual aids in the form of a geolocation map and a pie chart that displays the breed population for a given selection.*

## Motivation

*This file exists due to the man-hours that can be saved due to scripts which can use it to interact with the dataset.*

## Getting Started

*To get started, download this class, as well as the accompanying AAC\_CRUD.py. Setup a user account for your Mongo Database, then enter the account information into the class under the Data Manipulation heading. Run the script and test out the various sorting and filtering options to get a hang of the project.*

## Installation

*Requires Python3 and mongodb 4.2 or later. A number of necessary libraries are included in the main file, such as plotty.*

**PyMongo**

[*https://www.mongodb.com/docs/drivers/pymongo/*](https://www.mongodb.com/docs/drivers/pymongo/)

*PyMongo is the official python driver for connecting to and interacting with mongodb databases. This means that the driver is supported and will continue to function in future versions. Motor is the other official driver for mongodb. The difference is that PyMongo is meant for synchronous applications and motor for asynchronous. As we don’t need the additional functionality that motor provides, and PyMongo has generally been reported to be the faster of the two, it makes more sense to utilize PyMongo for this application.*

## Usage

RadioItems

A picture containing table

Description automatically generated

Here we utilize radio items for filtering. These items are assigned a numerical value. This value ties into the update\_dashboard function which we use to filter out data.

Text

Description automatically generated

DataTable

Text

Description automatically generated with medium confidence

The primary datatable displayed in the app. Using page size, we limit the items per page to 10, else it would likely take up the entire screen with thousands of items. I looked into other sort options, but was unable to find them; native seems to do what you’d expect from a sort however.

Widgets

Text

Description automatically generated

This controls the formatting on the widgets, with the map on the left and the chart on the right.

### Screenshots

Images:

This is a snip of my splash page. I tried to center the image through the use of html style and padding.Table

Description automatically generated with medium confidence

An unfiltered view with the radios and chart untouched.

Graphical user interface, chart, application, pie chart

Description automatically generated

Water Rescue Filtered

Graphical user interface, chart, application

Description automatically generated

Mountain Rescue Filtered

Chart

Description automatically generated

Mountain Rescue Filtered. I also sorted by breed so that I could get a different breed to show up on my widget, as there are so many german shepherds.

Graphical user interface, chart, application

Description automatically generated

Reset

Chart, application

Description automatically generated

## Roadmap/Features

*No further plans to build on functionality at this time.*

## Contact

Nick Nealeigh

nicholas.nealeigh@snhu.edu