# CS 340 PROJECT README

## About the Project/Project Title

This project is a python program and web application that allows easy access to data on various dog breeds considered for different rescue operations. It accesses a database of animals from local shelters and displays details, location, and makeup of the population of the general collection, as well as dogs preferred for water rescue, mountain/wilderness rescue, and disaster rescue or individual tracking.

## Motivation

Creation of this project is motivated by the need to pass CS 340 and gain the skills to create similar projects in the future. In a professional setting it would be driven by a need to create and maintain a space for storing different datasets on animals available within a single collection of similar items and have a method of accessing and grouping this data for ease of use.

## Getting Started

To get started with using this project, you need to change the parameters within the AnimalShelter.py file to match your settings. This includes the database, collection, host, port, username, and password under the \_\_init\_\_ function, if they are different from the defaults included. You should then run the AnimalShelterTest file to make sure there are no errors.

## Installation

For this project to run properly, you need to open the ProjectTwoDashboard file in a python IDE, preferably Jupyter Notebook. Using Jupyter allows you to run the tests that will throw exceptions without having to comment them out to run the other tests. In addition, you will need to install MongoDB and the Mongo Shell for your database if you haven't already and set up the database and collection that you wish to use to store data gained from using the AnimalShelter program.

## Usage

### Code Example

A screenshot of a computer code

Description automatically generated with low confidence

Code for switching between preferred rescue situations.

### Tests

Running the tests is as simple as opening the ProjectTwoDashboard file and running the program in debug mode(app.run\_server(debug=True)).

### Screenshots

A screenshot of a computer

Description automatically generated

Starting state with logo, drop-down, and unique identifier.

A screenshot of a computer

Description automatically generated

Reset choice with unique identifier (UID).

A screenshot of a computer

Description automatically generated with medium confidence

Water rescue with UID.

A screenshot of a computer

Description automatically generated

Mountain/Wilderness Rescue with UID.

A screenshot of a computer

Description automatically generated

Disaster Rescue or Individual Tracking with UID.

## Roadmap/Features (Optional)

Future versions will include update and delete functionality and testing for both. Tools used to create this program and web app were MongoDB, the Dash framework, and Jupyter Notebook. Jupyter ran the actual python code, and allowed for development of the Dash framework that became the front end of the application. Dash made the HTML portion of the application flow seamlessly with the python bridge and Mongo backend. Finally, Mongo was used as it has a well-defined library of interactions in python and is easy to manipulate by itself as well.

Steps taken to complete this project were to set up the Mongo database with the required information, create the CRUD functionality in the Python bridge so that interacting with the database could be done in a separate program, and finally to develop the HTML front end for the required usage of searching the given data for matches, based on pre-defined terms the client provided. The main challenge with this project was getting the data table to update successfully. Finally, after many attempts and testing of individual sections, it turned out that a line of code was missing that was needed to initialize the local variable in the function for changing the table.

Resources:

* <https://www.mongodb.com/>
* <https://jupyter.org/>
* <https://plotly.com/python/pie-charts/#pie-chart-in-dash>
* <https://developer.mozilla.org/en-US/docs/Web/CSS>
* <https://dash.plotly.com/>

## Contact

Jonathan Boeglin

jonathan.boeglin@snhu.edu