# CS 340 README

## About the Project

This project was created to deliver a user-friendly dashboard used to display information that is stored in a MongoDB database, in this case used to store information about rescue animals. The required functionality for this project was a sortable data table, a pie chart that can be used to show a breakdown of a desired data parameter, and a geolocation map with pointers showing the location of entries in the database.

## Motivation

MongoDB was chosen as the database management system because of its ease of use and integration with Python. Pymongo delivers much of the same functionality that is available in the Mongo shell, while also allowing integration of other frameworks like Dash. Dash callbacks were used to build a dashboard that calls Python methods from an included class. Documentation for Dash can be found at <https://dash.plotly.com/>.

## Getting Started

A MongoDB database with user authentication must first be created. More information on how to achieve this can be found at:

<https://www.mongodb.com/basics/create-database>

<https://docs.mongodb.com/manual/tutorial/enable-authentication/>

The Pymongo driver application will need to be installed to connect the dashboard to the database. The recommended installation method is by using the pip tool. More information can be found at <https://docs.mongodb.com/drivers/pymongo/>.

In the Jupyter notebook file used to display the dashboard, change the username and password variables to your credentials. The variable and class names used to connect to the database (shelter and AnimalShelter) can be changed, but the project’s Python file will also need to be updated accordingly. The image file path in the Jupyter notebook file should be changed to your own file or removed. The dropdown menu can be updated by changing the label and value parameters of the Dropdown object with the id rescue-dropdown. The filtering that is done by this dropdown menu can be changed in the update\_table method, using Pymongo queries.

## Usage

By default, the dashboard displays results for all animals within the database. The dropdown menu above the data table can be used to select different filters.

Unfiltered:

![Graphical user interface, application

Description automatically generated]()

Water Rescue:

![Graphical user interface

Description automatically generated with medium confidence]()

Mountain Rescue:![Chart, application

Description automatically generated]()

Disaster Tracking:![Chart, application

Description automatically generated]()

Pressing the reset button returns the dashboard to its original state, displaying results for all animals.![Application

Description automatically generated with medium confidence]()

To complete this project, an incremental development approach was taken, with the MongoDB database being created first, followed by the Python class that would be used to access the database. Then, each element of the dashboard was created in a Jupyter notebook file and brought together into an intuitive and user-friendly interface. Challenges were regularly encountered, but were able to be overcome by becoming more familiar with the frameworks being utilized. The Plotly, Dash, and MongoDB documentation was referred to extensively, in addition to many StackOverflow posts.

## Roadmap

Combine data within the pie chart if it is below a certain threshold, creating an “other” group to reduce the number of pie slices.

Add functionality to update the map pointer based off of which data table row is selected.

## Contact

Your name: JT Bolton, Jeffrey.bolton@snhu.edu