# CS 340 README

## Ryan Reames

CS-340-T4223

## About the Project – CS 340 Animal Shelter Dashboard

## This project simulates a database, middleware, and web application dashboard to show information on various animals that a group of animal shelters may have available for adoption.

## Motivation

This project was made for SNHU’s CS 340 Client/Server Development class. The purpose of the project is to gain exposure to non-relational databases (using MongoDB), middleware which can interact with the database (developed with Python), and development of a web dashboard which displays various statistics and information as requested by the user (Plotly Dash and Dash Leaflet). The project was created using Linux Xfce.

## Getting Started

To use this project, you must have MongoDB version 4.2 or newer installed, Jupyter Notebook for testing, and Python 3. Once these are installed, you will need to create a database titled AAC and collection titled animals, then import the aac\_shelter\_outcomes.csv file to a database titled AAC and collection titled Animals such as the following: Text

Description automatically generated

## Installation

Tools used in this project include:

* MongoDB – Non-relational database used to store and use animal shelter data. Install directions here: https://www.mongodb.com/docs/manual/installation/
* Jupyter Notebook – Used to develop python code which interacts with the MongoDB database, as well as testing files to make sure the project works as expected. Install directions here: https://jupyter.org/install
* PyMongo – used to connect Python to MongoDB. Install directions here: https://pymongo.readthedocs.io/en/stable/installation.html
* Plotly Dash – Used to create interactive dashboard. Install directions here: https://dash.plotly.com/installation
* Dash Leaflet - Interactive map. Install directions here: https://dash-leaflet-docs.onrender.com/

## Functionality

The purpose of this program is to create a user-friendly dashboard which has animals located at the Austin Animal Center meeting fictional company Grazioso Salvare’s requirements. Per the project specifications, the company requested that data should be able to be filtered based on the following criteria:

Table

Description automatically generated

In addition to this, the company requested a geolocation chart with the location of the animals, as well as an additional chart, both of which are dynamically updated based on filtering. For this project I implemented a pie graph using Plotly Express which displays the percentage of each breed on the current page of results (max of 10 results per page).

The following screenshots show the functionality of the project (you may need to zoom in to see details of animals on each screenshot).

A picture containing application

Description automatically generatedDefault view (no filter selected):

A picture containing application

Description automatically generatedWater Rescue Filter:

A picture containing map

Description automatically generatedMountain or Wilderness Rescue Filter:

A picture containing map

Description automatically generatedDisaster Rescue or Individual Tracking Filter:

A picture containing map

Description automatically generatedReset Filter (functionally going back to the default):

## Rationale for Using MongoDB and Dash

Beyond simply gaining exposure to new technologies, there were additional benefits to using MongoDB and Dash. MongoDB is free to get started with, can use unstructured data, and can interface with Python through PyMongo (also free to get started with). As data stored with MongoDB is non-relational, it is also easy to scale up if needed through either vertical or horizontal scaling (ex. sharding). Similarly, Python can be used with the Dash framework to quickly and easily build dashboards which interact with multiple data sources, such as MongoDB in this project. Rather than needing to create custom dashboards, it is simple to take templates and tweak them as needed to create professional looking interfaces.

## Steps Taken to Create Project

* Created MongoDB database following directions in “Getting Started” section above.
* Created Python CRUD module to interact with MongoDB using PyMongo (see animal\_shelter.py)
* Create dashboard using Dash and Dash Leaflet (see ProjectTwoDashboard.ipynb)

## Issues During Creation / Solutions

While creating this project the main hurdle I had was not having experience with MongoDB. I utilized resources provided by SNHU (ex. *Mastering MongoDB 4.x* by Alex Giamas) extensively and reviewed the weekly videos that our professor posted. I also had some issues testing in Jupyter Notebook that I couldn’t resolve until realizing I had to shut down my test file after making changes to the animal\_shelter.py file and then restarting the test file to have changes implemented. During creation of the dashboard using Dash, I had some additional trouble getting multiple markers to appear using Dash Leaflet. I ended up searching the Plotly forums and official Leaflet site and found that I had to make an array of markers, then pass them in through a LayerGroup. Along similar lines, I had some trouble getting popups to appear for each marker when clicking but was able to resolve this by adding html to my marker array for the animal name and breed.

## Contact

Ryan Reames