# CS 340 README

## About the Project/Project Title

This project uses a MongoDB database and integrates it with a python module and client side application to filter a list of animals to specific search criteria. It also uses a pie chart to show distribution among the search results and a map to show the general area of selected results.

## Motivation

Grazioso Salvare trains dogs for search and rescue operations. They have an agreement with a non-profit agency that runs multiple animals shelters in the Austin, TX area. They needed a system that utilized a database and a web based application dashboard to more easily search through the available animals to find suitable candidates for training. MongoDB was chosen over other database languages for its versatility when working with unstructured information, such as unordered lists of animals from various shelters. Additionally, as most of the functionality needed is CRUD methods, PyMongo is a very powerful solution. Because of this, MongoDB and python were chosen as the ideal programming languages for this application. The visual dashboard is made using the Dash framework, specifically a version made to work well with python when using Jupyter Notebook.

## Getting Started

To get a local copy running, first you must either download the AnimalShelter.py file or you can use your own python module that contains a read method, though it is recommended to at least use AnimalShelter.py as a guideline for your own module. Make sure to load the python module file into your project folder. Finally, make sure to add the line “from AnimalShelter import AnimalShelter”. When creating an instance of the AnimalShelter object in your own project, make sure to use a valid mongodb username and password as parameters, i.e. AnimalShelter(“username”, “password”) with the quotes. Note, that if you choose to alter any of the source code, knowledge of MongoDB, python and html is strongly recommended.

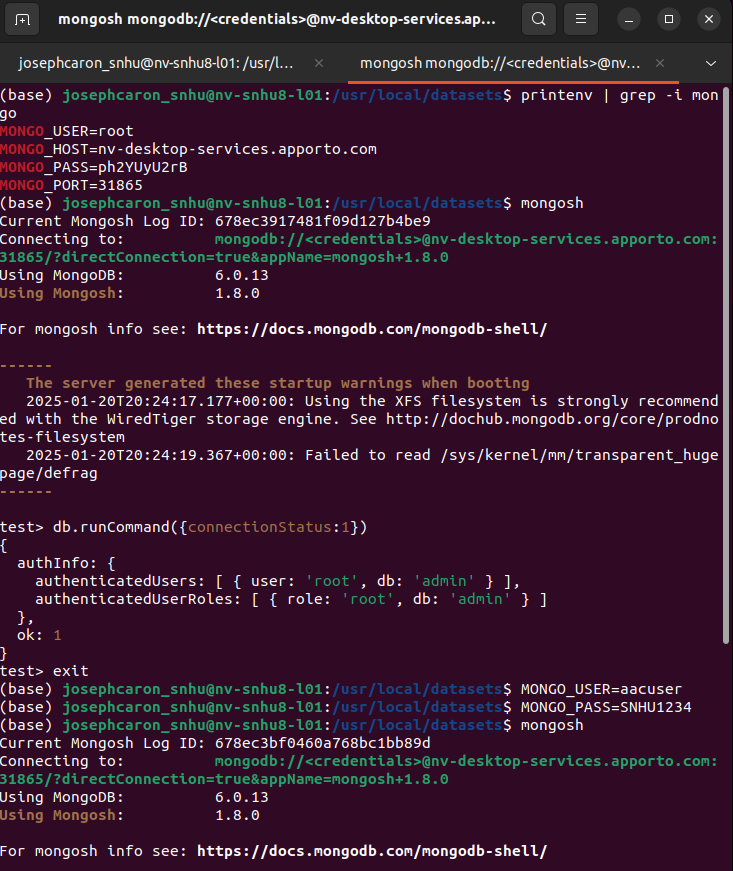
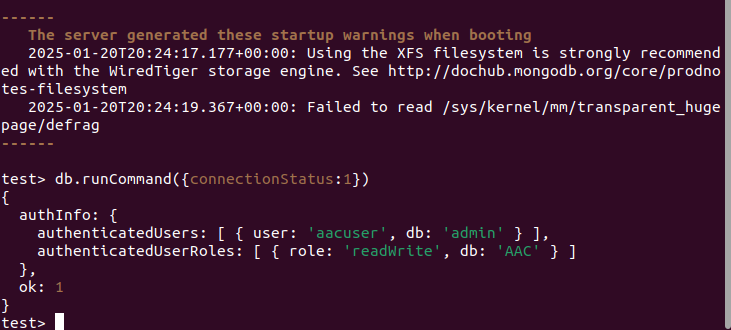
## Installation

This project was created and tested using Jupyter Notebook and a command terminal. It requires the Pymongo and Objectid libraries. It was created after importing a csv into MongoDB as shown below:

A screenshot of a computer

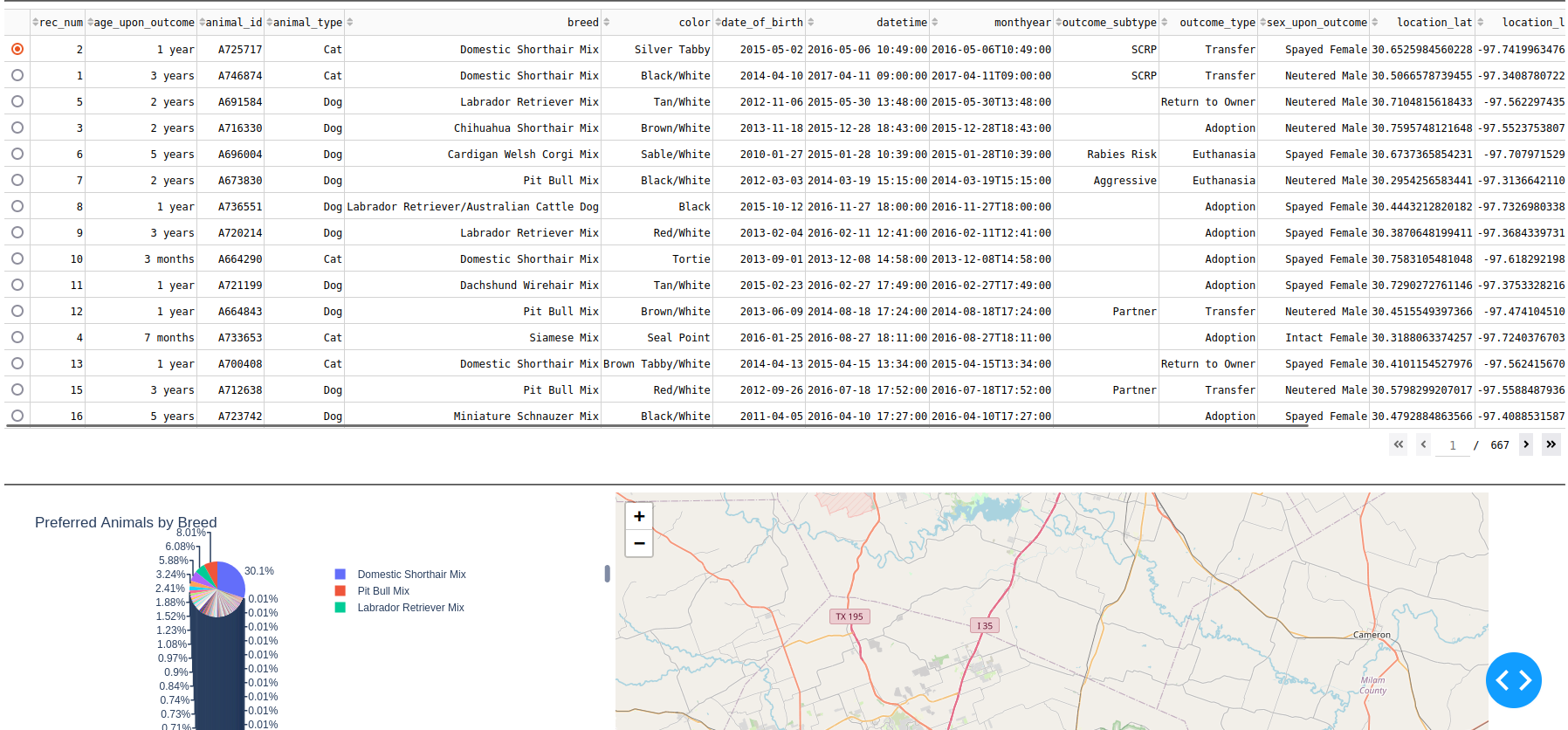
Description automatically generated

I also created a new user account to use with this project as shown below:



## Usage

### The project is used to tie together a database of animals in and around the Austin, TX area and a web based application dashboard. This allows Grazioso Salvare users to search through this database, using pre-determined filters, to find dogs with the preferred traits to be trained for search and rescue operations. After loading the application dashboard, a user needs only to open the dropdown menu and select the type of search and rescue operation they are looking for training candidates for. The datatable will then update to show only dogs that meet the preferred criteria. This will also update a pie chart widget that shows the distribution of breeds in the filtered results. Clicking on an animal in the list also updates a map widget to show that animal’s location.



### Code Example

Below is the code for creating new documents in the collection and reading documents from the collection. A close-up of a code

AI-generated content may be incorrect.

Below is the code for updating and deleting documents in the collection. A screenshot of a computer code

AI-generated content may be incorrect.

This is some of the code for the application dashboard

*A screenshot of a computer screen

AI-generated content may be incorrect.*

This is some of the code for the different filter options and the pie chart

A screenshot of a computer

AI-generated content may be incorrect.

### Tests

The below shows the input to test creating a document and the confirmation that it was successful.

A screenshot of a computer code

Description automatically generated

The below shows the input to test reading a document from the collection, as well as the document to show it was successful.

A screen shot of a computer code

Description automatically generated

The below shows the input to test the update function for updating a single document and many documents as well as the confirmation of how many records were updated to show it was successful.A screenshot of a computer

AI-generated content may be incorrect.

The below shows the input to test the delete function for a single document, as well as the confirmation dialog to show it was successful.

A close-up of a computer code

AI-generated content may be incorrect.

The below shows the input to test the delete function for multiple documents, as well as the confirmation dialog to show it was successful. A close-up of a computer screen

AI-generated content may be incorrect.

This was a test of the filter options. As you can see, choosing the water rescue filter had drastically changed the datatable and pie chart

A screenshot of a computer

AI-generated content may be incorrect.

Another test with the Mountain and Wilderness Rescue option

A screenshot of a map

AI-generated content may be incorrect.

**Challenges**

The largest issue I had with creating this application was with the AnimalShelter.py module. I had erroneously created the read method to require a key-value parameter, otherwise it would throw an exception. This ended up being a major issue, as the MongoDB command find() has the capability to show the entire database collection, but only if you run it with an empty parameter field. But of course, the way I wrote the program, this was breaking the application. It took a long time to figure out what was causing the problem, though once I did it was very easy to edit the program to work as intended.

## Contact

Joseph Caron