# CS 340 README Template

## About the Project/Project Title

Grazioso Salvare identifies dogs that are good candidates for search-and-rescue training. This project is the implementation of a dashboard application that pulls information from local animal shelters stored in a database and allows for Grazioso Salvare to filter the data to help improve the search for successful candidates. The dashboard also includes a pie chart representing the dog breed distribution per filter and a map showing the exact location of the animal.

## Motivation

This projects’ main goal is to learn to build a client/server infrastructure, allowing for filtering of local animal shelter data. The results from these queries are meant to help Grazioso Salvare find quality search-and-rescue candidates. This application has been built from the database all the way to the front end.

## Getting Started

* First using the mongoimport statement create the database, collection, and import the dataset being used.
* To create a new user open the Mongodb environment then enter the admin database
* Enter “db.createUser” and enter the username and password and set the roles that the user is allowed to have and what databases the user has access to. This username and password will be necessary when connecting to the python CRUD file.
* In the Mongodb environment run the “printenv” statement to get the host and port values.
* Next import the AnimalShelter python file into your project using “from animalShelter import AnimalShelter”
* Create an AnimalShelter object with arguments for the username, password, host, port, database, and collection that are being connected to.

Once these steps are complete you should be connected to both the python file and the Mongodb client giving full CRUD functionality. If there are issues with the connection double check that all the arguments passed into eh AnimalShelter object are correct this was an issues during initial development having enter the wrong port.

A screenshot of a computer

Description automatically generated

The “aacuser” was created with “readwrite” authorization for the “AAC” database allowing this user to use the CRUD functions from this project. The collection being used is animals and was initially created with the import of a csv document. Ensure that the “Col” variable is set to “animals” as the “AAC” database has multiple collections. Below are screen shots showing the creation of the “aacuser”, created with read and write authority to the AAC database. I have also included a screenshot of the mongo import method showing how to import a csv file into the correct database and collection.

A computer screen shot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

## Installation

Jupyter Notebook – this tool makes it easy to test small portions of the code making it ideal for testing various different project types.

Mongodb – Mongodb is a user friendly database system that is easy to implement with Python, with very detailed documentation. Mongodb is suited for working with potentially inconsistent data making Mongodb a great database choice when you can’t plan for all information that you may incorporate into you application.

https://www.mongodb.com/docs/

Mongoimport – the mongoimport tool is a useful tool to allow for a user to import their own data into the mongo shell.

https://www.mongodb.com/docs/database-tools/mongoimport/

Python – the python file should be saved in the same file as the Jupyter Notebook file in order for the import statement used to properly operate. Python has numerous libraries dedicated to data analysis making it a great language to use when working with large amount of data. There is also Pymongo a framework specifically built for working with MongoDB making Python a very easy language to implement due to official support.

https://www.mongodb.com/developer/code-examples/python/python-crud-mongodb/

Dash- the Dash framework is an easy to implement framework giving the user access to numerous interactive components allowing for simple implementation of UI elements.

https://dash.plotly.com/dash-core-components

Plotly- open source graphing library for Python making it easy to incorporate data visualization into an application, with extensive documentation.

https://plotly.com/python/pie-charts/

## Usage

The AnimalShelter python code allows a user to create, read, update, and delete from a Mongodb database. A user can connect this file to their application and easily interact with their database. The dashboard portion of this project offers a way to display and filter data from a database. There is also the addition of data visualization tools i.e. a pie chart and a geolocation chart to help convey the data being presented from the database. This dashboard can be easily modified allowing a user to import their own database information straight in to this dashboard.

### Tests

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

First the import for the python class is called (the pprint call was for testing purposes). Once this is done a class object is created in this case animal and a dictionary for the testing of the create method. Next the create function is called using the animal class object and the testAnimal dictionary, if the create is successful there will be a print of the result followed with “True”. Next the read statement is called using the animal class object and the parameters being queried this returns all results. Then the update statement is tested using the animal class object the first dictionary is the key/value pair to search for, the second dictionary is the index being updated and the information that needs to be inserted followed with a read statement to show how the results have changed. Finally there is the delete statement using the animal class object, using a dictionary as the key/value pair to find the document that is to be deleted and a read call which returns an empty dictionary proving that the item has been deleted.

A screenshot of a computer

Description automatically generated

## This shows the pie chart and map displaying one dog when the Water filter is selected.

A screenshot of a computer

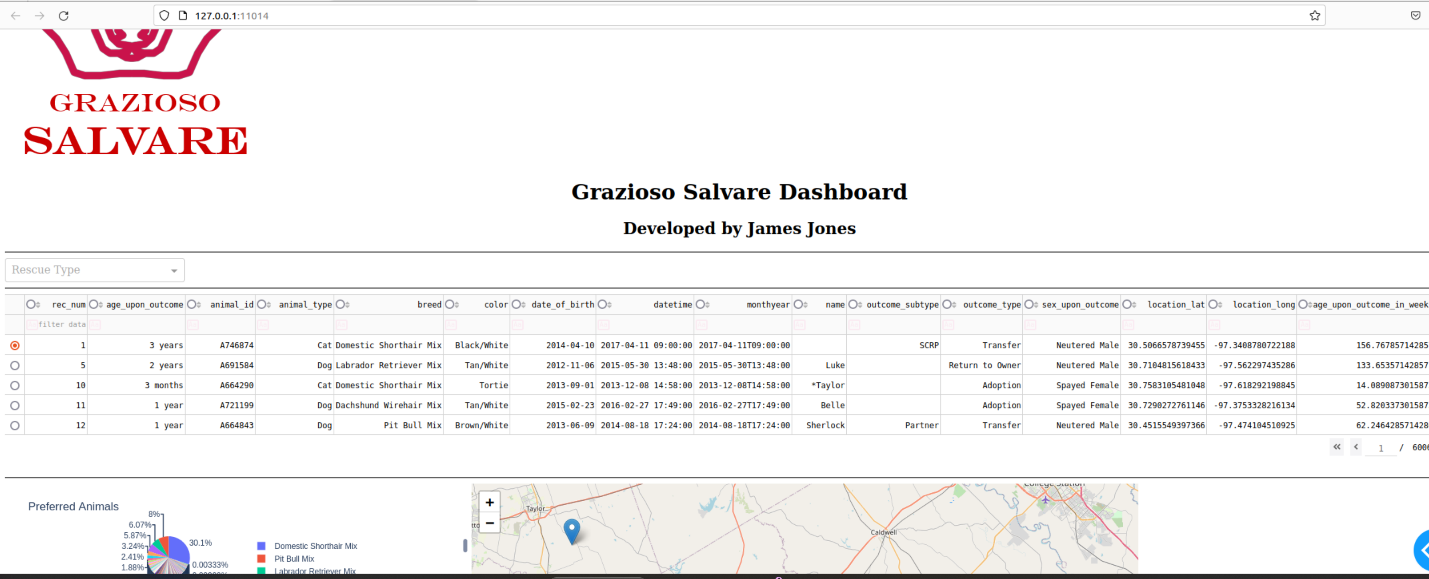
Description automatically generated

This shows how the dashboard has changed when using the Mountains/Wilderness filter

A screenshot of a computer

Description automatically generated

Next is the change implemented in the Disaster/tracking filter.



Finally this shows the removal of the filters which can be accomplished by clicking the “x” on the drop down menu.

## Contact

Your name: James Jones