# CS 340 README Template

## About the Project/Project Title

Rescue Training Lookup will help Grazioso Salvare to maintain a database of dogs through 5 different locations from their home office.

## Motivation

This database will help them to narrow down the certain breeds and ages for effectively training rescue dogs.

## Getting Started

1. Import CSV into DB
2. Create user that has the ability to only read/write to that DB \*not admin acct\*
3. Once account is created attempt to connect to the DB. You should make sure that the user and pass are the same as when you created the user account. Also be sure to check host and port numbers are correct.

Once these steps are done you will be able to run the module after importing it. Again Host and Port numbers should be changed in the module to reflect your current values. For the create function it will time out if any of the connection variables are incorrect. The create function takes your dictionary and inputs it into the DB using insert\_one function from pymongo. It should return true or false if the item was added or if it wasn’t. The read function acts as a search function this function parameter is a key/value pair, it uses a for loop to input the find() cursor into a list. If there are no values that match the pair it will return an empty list

**Getting Started with dashboard**

To connect to the dashboard you will need to know the Username, Password, host, port, database, and collection. This will be passed into the parameters for building the DB object. It is also important to drop the \_id field when loading in the DB as it will cause the Dashboard to crash. I have added a fields list. Using this we can choose which columns the table will display that way we only get the information that we need. We get all this functionality from using the Dash framework as well as plotly, numpy, as well as pandas. These are needed to create the table, charts, and help create the data frame for the interactive data.

## Installation

To run you will need to install python and MongoDB

Use Python’s Mongo Driver

To install type pip install pymongo

Install an IDE of your choice

For the Dashboard you will need Dash

Plotly, Numpy, and pandas

## Usage

### Code Example

In This code we want to use the MongoClient from PyMongo. To do this we need to import it from PyMongo. Once imported we can create our object and connect using our DB host, username, pass, and port. Our functions require to create a dictionary and to search a key value pair. Example: new\_dog\_search = { “name” : “Zues”} then run rescue.read(new\_dog\_search) this will find any dog with he name Zues in the name field. The create Function is similar except it will use a dictionary that has more than 1 key/pair value such as name, type, breed, and age.

**Dashboard Code:**

If you want to change the column fields you can do this by updating the fields variable. Queries are constants and should not be modified unless it is done within the variable. The pie charts and Geo Location takes the data from the table using the breed, and coordinates **\*note: coordinates are not listed in the columns but can be added as stated above\*** The table is filtered using radio buttons these can be modified to add more or changed for different filtering.

### Tests

To run the test import AAC.py and it is nice to use pprint.

Create your object such as rescue = AnimialShelter()

You then need to define variables for your terms to either create or search

Create will be can be a BSON file or another type of dictionary

The search field should be key:value pairs

**Dashboard Test:**

To test the dashboard the main thing you will need to do is to import your DB. Making sure the connection variables match your information. After this you will need to change the fields variable to match the fields that are in your DB. If you wish to show all your fields you can change this line of code: **columns=[{"name": i, "id": i, "deletable": False, "selectable": True} for i in df.loc[:,fields]]** to **columns=[{"name": i, "id": i, "deletable": False, "selectable": True} for i in df.columns].** This will also need to be changed in the filter function as well. After making these changes you should be able to run the dashboard.

### Screenshots

Import DB: A screen shot of a computer

Description automatically generated

Test Script:

A white background with black text

Description automatically generated

Fields and Queries:

A screenshot of a computer code

Description automatically generated

Dashboard Screenshots:

(Disaster Filtered): A screenshot of a computer

Description automatically generated

(Mountain Rescue): A screenshot of a dashboard

Description automatically generated

Screenshot of functions:

A computer screen shot of a code

Description automatically generated

## Roadmap/Features (Optional)

*Provide an open issues list of proposed features (and known issues). If you have ideas for releases in the future, it is a good idea to list them in the README. What makes your project stand out?  
  
Note: This section is optional for the purposes of this assignment. If you choose not to fill out this section, remove it from your final README file.*

## Contact

Your name: Cody Phelps