# CS 340 README

## About the Project/Project Title

**Grazioso Salvare Dog Categorizing App**

A full stack application that allows the company Grazioso Salvare to identify and categorize available dogs from different animal shelters. These dogs will be trained for different types of rescue missions. This app is needed because training is more effective for dogs that are no more than two years old. Also, certain breeds of dogs are more proficient at different types of rescues. These may include, water rescue, mountain, or wilderness rescue, locating humans after a disaster, or finding a specific human by tracking their scent. This application allows the company to search through the animal shelters database. They can search by animal ID, age, animal type, breed, color, outcome type, gender, and more. The application is linked with the client-side server. The company can search, add, update, and delete any of the animals in the database.

## Motivation

This application exists to organize different dogs from five separate animal shelters. These dogs are organized by age, breed, gender, etc. Having database that contains all these dogs makes it easy to identify which dogs are suitable for training in search-and-rescue missions.

## Getting Started

After the user installs the appropriate packages explained below. They will have to run MongoDB on the appropriate port. While mongo is running, they can access their python code and perform any actions within the animal database. These actions may include adding, removing, updating, and searching. After copying the python code provided ensure that the local host port is changed to the user’s port. There are four functions in the python file. These functions include create, read, update, and delete. The create function will take a dictionary entry entered and add it to the animal database. The read function will take data entered in a key: value format and look through the database for it. If found it will return the entire animal dictionary. With the find one it will only look for the first available entry and output that. The update function will take a dictionary key: value input and replace it with the new value. The update one will only allow one key value to be updated. Finally, the delete method will look up the dictionary value and if found delete it.

## Installation

The application utilizes python and MongoDB to run. Both python and MongoDB packages can be installed online.

python install link - <https://www.python.org/downloads/>

mongoDB install link - <https://docs.mongodb.com/manual/installation/>

After both installs are made the first step would be setting up your mongoDB database. The first step would be typing /usr/local/bin/mongod\_ctl start. Afterwards, type mongo –authenticatioDatabase “admin” -u “admin” -p. Then create your admin password. After that you can create multiple accounts in a similar way. For more information on accomplishing that, visit the following link.

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

Now that the Mongo database is set up, we can set up the python file that connects to it. As a reminder whenever the python program is being used the Mongo database must be logged into with username and password. The python files require importing pymongo, json, MongoClient, and bson.objectid. After they are imported ensure that all the methods look like the pictures posted below. If the methods are not working, ensure that mongoDB did not lose connection. Also, ensure that everything has been typed correctly and that you are accessing the correct database and port number.

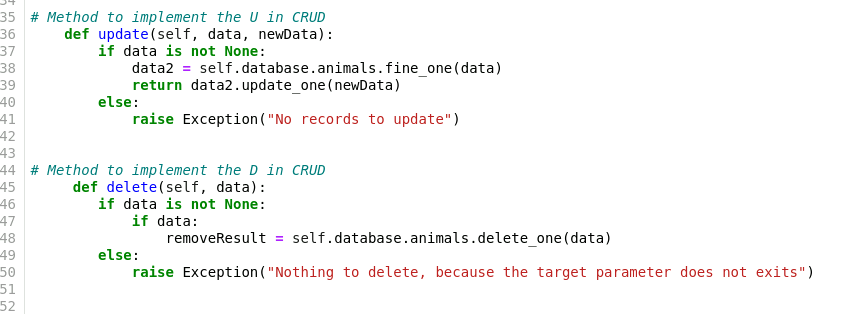
**Dash Framework**

Dash framework and Pandas data frame code was used to structure the program. Dash is used for structuring web page components for the dashboard. Inside my HTML code I developed radio buttons, data table, pie chart, and the map with user interactivity by utilizing Dash. Pandas is used for data structures and operations for manipulating numerical tables. Whereas Dash is used by taking that data and creating interactive web applications. Using both Dash and Pandas, we can create a very user friend client-side dashboard.

## Usage

### Code Example*Text Description automatically generated*

*Text

Description automatically generated*

### Tests

Logo

Description automatically generatedTable

Description automatically generatedA picture containing graphical user interface

Description automatically generatedA picture containing chart

Description automatically generatedA picture containing table

Description automatically generatedTable

Description automatically generated with medium confidence

**Challenges**

Challenges I faced when creating this program was ensuring that all my callbacks were using the correct data. The next issue I faced was ensuring that I was importing the correct dependencies to my app. The final challenge I faced was an issue where Mongo was running on a port from a previous used time. To solve this issue, I closed my web browser and opened a new browser and got it to work in the virtual lab.

## Contact

Your name: Thomas Cogley, lead developer