# CS 340 Project README

## About the Project/Project Title

This project is meant to create a Python module to interact with a MongoDB database applying CRUD functionality. The Python code allows the user to Create, Read Update and Destroy documents in the database. A Dash interface uses the Python code to allow the user to interface with the database in a user-friendly way.

## Motivation

This example is made to access the Austin Animal Center database, which is a branch of Grasioso Salvare, an organization the identifies dogs to use for search and rescue. Access to the AAC database is useful for them so they can identify candidates for training. This client-server set up is to help them search the database efficiently by using a simple interface to access the data. A MondoDB shell is used for the server side. Python code is used by the Dash interface to access the database. Ideally Grasioso Salvare would filter the animals to find animals that meet specific criteria to best suit the needs of three main types of training: Water Rescue, Mountain or Wilderness Rescue, and Disaster or Individual Tracking. The same idea could be used for other databases and organizations.

## Installation

This uses MongoDB, Python and Jupyter Notebook for testing. Examples of how to use these are shown above and below. Once the Mongo shell is available, use an admin account to gain access to a database. The above example uses the Grasioso Salvare Austin Animal Shelter database. A new user should also be created to access the database.

The Python file is the CRUD.py file. This needs to be available in the Jupyter Notebook. This can be done by downloading the file into the notebook. The ProjectTwoDashboard.ipybn file needs to be available as well. The user and password needs to be changed to the user created that can access the AAC database. The above set up shows examples of how to accomplish this.

## Getting Started with the MongoDB Server

The Mongo shell needs to be accessed. It also needs to have a user set up to access the AAC database.

A computer screen with white text

Description automatically generated with low confidence

MongoDB user must be set up through the Mongo shell. Make sure that they have the option to read/write so that they can use all the functionality of the CRUD.py file. Once the user is set up check this by using db.runCommand({connectionStatus:1}). It should say that the user can readWrite.

A screenshot of a computer

Description automatically generated

**Accessing the Server with Python**

### The CRUD.py file needs to be downloaded. Change the parameters to connect to your local MongoDB shell using the correct MongoClient. To access the animals collections the following need to be done in MongoDB:Access collection:

### *A screenshot of a computer program Description automatically generated with low confidence*

### Python Code:Create function:

A screen shot of a computer code

Description automatically generated with low confidence

Read function:

A screenshot of a computer code

Description automatically generated with low confidence

Update Function:

A screen shot of a computer code

Description automatically generated with low confidence

Delete Function:

A screen shot of a computer code

Description automatically generated with low confidence

**Dash Interface**

In the ProjectTwoDashboard.ipybn file the CRUD object needs to be imported from the CRUD.py file. The username and password needs to be changed to match the user created to access the AAC database. A CRUD object is then created using the account details given.

A screenshot of a computer

Description automatically generated with medium confidence

There is a readAll() function in the CRUD.py file that will read from the entire database. This function is called to pass the data from the Mongo shell to the client side interface.



## Usage

The ProjectTwoDashboard.ipybn file can now run and should be able to access the database. The Interface should open to the Grasioso Salvare logo:

A red line drawing of a dog

Description automatically generated with medium confidence

A table with the information about all the animals in the database will be visible by scrolling down.

A picture containing text, screenshot, font, number

Description automatically generated

Below the table there is a pie chart and a map visible. The pie chart can be changed by using the dropdown menu. It is default set to a reset tab which shows all the animals in the database.

It can be changed to help find animals that would be suitable for different types of candidates. The criteria for each type of training are coded in the Dash file.

**Examples of Candidates:**

**A red circle with white text

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with medium confidence** Water Rescue

Mountain or Wilderness Rescue

A screenshot of a computer

Description automatically generated with medium confidenceA picture containing screenshot, text, diagram, design

Description automatically generated

Disaster or Individual Tracking

A screenshot of a computer

Description automatically generated with low confidenceA picture containing screenshot, text, circle, diagram

Description automatically generated

A specific animal can be selected using the circles to the side of each animal. When selected, the animals geographic location will be marked on the map.

A screenshot of a computer

Description automatically generated with medium confidenceA map with a blue point on it

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidenceA map with a blue point on it

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with medium confidenceA map with a blue point on it

Description automatically generated with low confidence

**Rationale**

Mongo DB is a noSql data store. This is used so that the database can store large amounts of data and can be easily accessed. Python is used to access the database and as the connection between the database and the interface because it is efficient at sorting through large amounts of data. Creating an interface makes accessing data and using it much easier for the user. Dash is used because it is simple to develop the interface. Creating interactive widgets and displaying data is simple and efficient when built upon the dash interface.

## Contact

Zac McBride:

|  |  |
| --- | --- |
| [zachary.mcbride1@snhu.edu](mailto:zachary.mcbride1@snhu.edu) | 801-898-6452 |