# CS 340 README

## About the Project/Project Title

With this project you can access and edit a previously existing MongoDB database and link those actions to elements within the project you create around it. This program includes the actions in the CRUD method to Create, Read, Update, and Delete the entries in the MongoDB database provided by the client for the purpose of this program. By utilizing the functions in the CRUD\_Module.py file it is much easier to retrieve the required information needed by a user-friendly front end which will display the

## Motivation

This program was developed for Grazioso Salvare for the purpose of easily identifying and categorizing high potential animals from several animal shelters around the Austin, Texas area to be trained as search and rescue dogs. By utilizing the user-friendly front end and the sorting/search features available a user is able to quickly comb through the large amount of data quickly to find the best candidate for search and rescue training.

## Getting Started

To get started with this project, first download the zip file containing the CRUD\_Module.py file, ProjectTwoDashboard.ipynb, and Grazioso Salvare logo, and extract it to the your project location. Then import the module by adding the following code to your main python file at the top with the rest of your imports.

from CRUD\_Module import \*

Note that I simply used the import all method instead of directly importing the animalShelter class due to issues with Jupyter Notebook not finding the class correctly.

After this we can import our data into our database if we haven’t done that already by using the mongoimport command as shown below.

A screenshot of a computer

Description automatically generated

While the current module doesn’t use authentication it is possible to include authentication into the module by making user accounts for the database using the terminal and using that authentication in the module.

Text

Description automatically generated

Now, start the mongoDB service in the terminal using the command below and capture the port number.

/usr/local/bin/mongod\_ctl start-noauth

In the CRUD\_Module.py file change the port number in the \_\_init\_\_ function to match the port number of your port and change the database and collection names to match the database/collection of your choice.

Text

Description automatically generated

Now you should be all set to utilize the module to make modifications to your database via code instead of command line.

## Installation

Because the installation for each of the tools listed below may vary over time and OS I recommend following the instructions for installation provided by the links below to install each tool using the common method.

***Jupyter Notebook***

{Development Environment and Data Front-End visualizer}

[*https://jupyter.org/install*](https://jupyter.org/install)

*­*

***MongoDB***

{Database to store the animal shelter information}

<https://docs.mongodb.com/manual/installation/>

To ensure that you’ve installed MongoDB correctly, open a terminal and enter the following command to start MongoDB with no authentication.

/usr/local/bin/mongod\_ctl start-noauth

Text

Description automatically generated with medium confidence

NOTE: Write down or make note of the port number listed above as it will be necessary in future steps.

Now type mongo into the command window and hit enter. The outcome should look similar to the window below.

Text

Description automatically generated

If your terminal window looks similar to the window above Mongo has been correctly installed.

***PyMongo***

{Tool for interacting with a MongoDB database from Python}

Follow the link below for instructions to install the latest version of PyMongo.

<https://pymongo.readthedocs.io/en/stable/installation.html>

***Dash Framework***

{Library to create the data visualization in Jupyter Notebook}

To install the Dash Framework open a terminal window and enter the following command then press enter.

pip install jupyter-dash

Dash also recommends installing Pandas since a large amount of their examples require this additional component. You can install this similarly to the Dash Framework by entering the following command.

pip install pandas

***AAC Shelter Outcomes Data***

{Sample data setoff animal center outcomes}

Once you have installed MongoDB it is time to import the sample data set for the animal shelters.

First download the aac\_shelter\_outcomes.csv file and make sure not to rename it at all. Then you can open a terminal and navigate to where your csv file is stored by copying the following command into your terminal and replacing everything in the file path with what is in your files file path.

cd /usr/local/datasets/

Then import the csv file by copying the following command which should result in your terminal window appearing like the one in the screenshot below. Be sure to replace the port number below with the port number you wrote down when installing MongoDB

Mongoimport –port 52858 –db AAC –collection animals –type csv –headerline –file aac\_shelter\_outcomes.csv

A screenshot of a computer

Description automatically generated

To check and see that you are able to access the AAC database correctly first start Mongo without authentication using the command below.

/usr/local/bin/mongod\_ctl start-noauth

Then type in the command below and hit enter.

mongo

From here type the command “show dbs”, hit enter, and you should see the AAC database appear in the database list.

Text

Description automatically generated

This means that you have successfully entered the database into MongoDB and are able to access it through the terminal window.

## Usage

### Code Example

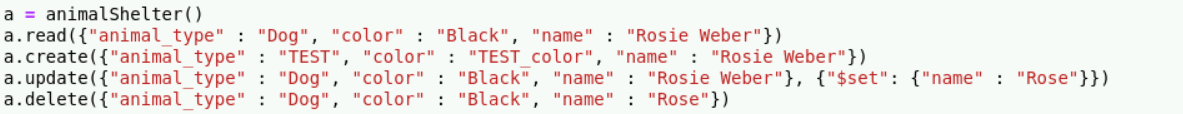
The four functions shown below are the four functions that are able to be called from this module. The screenshot below shows how each of the functions should be used in code.

create(query={})

read(query={})

update(query={}, updates={})

delete(query={})



### Tests

Below is an example of how to run a test to ensure that the module is working correctly. In this example we are testing to see if we can read from the database correctly. First we get an instance of the class, run the read function which should return a Cursor object, and check to ensure that data was returned.

Text

Description automatically generated

Below is a screenshot of the test results. In addition to the test result of OK for each test I’ve included print outs of the return values to ensure that I understand what the data looks like.

Table

Description automatically generated

### Screenshots

CRUD Test Screenshots

Text

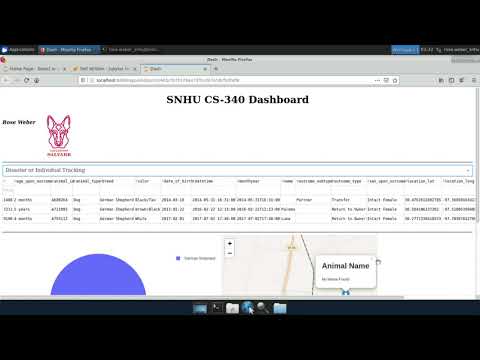
Description automatically generated

Results

Table

Description automatically generated

Below is a screencast of the usage of the dashboard as a whole which showcases the dynamic filtering and updating of the pie chart, map, and data table. I chose to include a screencast for this section instead of screenshots to aid in the demonstration of the dashboard.

[](https://www.youtube.com/embed/cbt1BZJRX3o?feature=oembed)

Please note that if you are unable to view the video above please follow [this link](https://www.youtube.com/embed/cbt1BZJRX3o?feature=oembed) to view it in a browser.

## Contact

Your name: Rose Weber

Rose.weber@snhu.edu