**Module 5: Project One.**

Dr. Tad Kellogg

Southern New Hampshire University

CS-340 Client/Server

Trevor Skillman

October 15, 2022

# CS 340 README Template

*Use this template to complete your README file. When completing the template, keep the headings as they are so that your document has a clear organization. Remove the italicized prompt text after you have completed each section for a polished final document.*

## About the Project/Project Title

For this week’s assignment, we created code from the CRUDs method using create(), read(), update(), and delete() methods and test out the functionality of the code using Python to ensure it is executing and running properly.

## Motivation

This project exists to be able to create a document inside of the AAC database of new animals being inserted and able to search for the document easier by using PyMongo and able to have the end user query with ease inside of MongoDB.

## Getting Started

To start we need to be inside MongoDB with no auth by following these steps.

Cd /usr/local/datasets

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

*Text

Description automatically generated*

When inside of “/usr/local/datasets$” we can go ahead and import the CSV file that we want to use and the following snip will show the exact command to enter and its execution after the command is executed. Showing “1000 document(s) imported successfully”

Text

Description automatically generated

then from here you want to enter “mongo” and that will then bring you into the mongo shell like so

*Text

Description automatically generated*

*Text

Description automatically generated*

Type in the command “show dbs” and then want to enter in “use ADMIN” to use the admin database then we want to jump to the next 2 snips of the commands for when creating an admin user and a regular user account:

Text

Description automatically generated

The previous snip showed the creation of the admin user, the next snip will be the commands to create a regular user. Be sure to not have any typos as you can see in the next snip that I receive an error on my first attempt due to a typo, but below that is where I successfully created the user account.

Text

Description automatically generated

In the next snip, you will see the login process for both accounts on one snip, read the lines carefully to make sure you notice each of the logins for the admin and the user account, the user account has a username of “aacuser”. You will also need to exit the mongo shell to login to the appropriate accounts.

Text

Description automatically generated

Notice the commands entered to log in using the admin account, if the command is successful the system will ask you to enter your password so be sure to write down or remember it. Upon successful login, you can test to see if the admin account has access to all the databases by entering “show dbs,” you will see the databases the admin account can work with. Type “exit” to leave the account. We will now log in under the “aacuser” account. As you can see, it is the same command to log in as the admin would. After successful login, enter the command “show dbs,” which should show only one database that the “aacuser” account can access and maintain.

## Installation

You will need to install Jupyter NoteBooks via <https://jupyter.org/install>. You would also need to install MongoDB via [www.mongodb.com](http://www.mongodb.com). Jupyter is where you can input the code to be used within MongoDB, you would also use the Jupyter execution file labeled “example.ipynb” file. You would copy and paste your code inside of that file to execute the code.

## Usage

*Use this space to show useful examples of how your project works and how it can be used. Be sure to include examples of your code, tests, and screenshots.*

### Code Example

With the example below, you want to start by importing the mongo client and then Initialize the MongoClient be sure to put in the specific port number you will be using for the mongo shell. By entering it after “/localhost:”

*Text

Description automatically generated*

With the following snip, I have included the create method. In which we create self and data objects to create a new document inside of the animal’s collection, if there is something to input, then it will come out to true and add the document, else it would come up with an exception with no data entered.

Graphical user interface, text, application

Description automatically generated

For the next snip, you can follow along with a simple read method, in which we initialize the curser to read a document that was created or in the database, and it then returns the curser holding the data to be read.

A picture containing text

Description automatically generated

### Tests

The following snip is where I added the test script to see if the input data will be outputted as a result. We had to initialize the username and password for authentication purposes and pass that in the parameter inside of the shelter object. Then the data being passed to create a new document of a new animal will be created. Next, we want to read what was created.

Graphical user interface, text, application

Description automatically generated

### Screenshots

The following snip shows the output being a successful desired outcome.

*Graphical user interface, text

Description automatically generated*

*Graphical user interface, text, application, email

Description automatically generated*

In the next couple of snips, I will show the update and delete methods used for project one.

Graphical user interface, text, application, email

Description automatically generated

The following snip is the test script for the update and delete method.

Graphical user interface, text, application

Description automatically generated

The following snip is the execution of the code. You will notice the update to 10 years old and the color gray, which then deletes instances of the name “Spike” and animal type of “Dog.”

Text

Description automatically generated

I could not get the delete result in the execution, but I will provide a snip of searching for “Spike” and “Dog,” in which I get a return of “null,” which tells me deletion was successful.

Text

Description automatically generated

## So, now we have established our CRUD methods. I will show you a code for the final project for Grazioso Salvare Dog Rescue. With the following snip, you will see code starting out in HTML format that adds a logo to the dashboard, and the logo was provided for the project. Within the code block, you will see an “href” which specifies a URL link. If you were to click on the company logo, it would open a new window going to the SNHU website.

Text

Description automatically generated

With the next snip, I will show the execution and the logo opening a new window to SNHU.com when the logo is clicked.

A picture containing text

Description automatically generated

When the logo is clicked, you will be brought to this website.

Graphical user interface, website

Description automatically generated

With the next couple of snips, I will show the code blocks for creating the filtered option of radio buttons that will filter out the certain types of dogs that are available for the specific type of rescue.

A picture containing scatter chart

Description automatically generated

The next would be the code that tells you what happens when each radio button is clicked and filtered. If “Water Rescue” is pressed, then it will show you the dogs that are available for water rescue. If “Mountain Rescue” is clicked, then the dogs that are available for this rescue will be filtered. Same with “disaster rescue.” Then finally, if “RESET” is pressed, then the data table will show the entire database of the animal shelter.

Text

Description automatically generated

Now I will show several snips of each radio button and the outputs it will show within the data table.

A picture containing text

Description automatically generated

Text

Description automatically generated with medium confidence

Text

Description automatically generated

A picture containing text

Description automatically generated

In the next snip, I will add in the code block I created for making a pie chart to show the percentage of available dogs and what breeds are available with their ages after each radio button filter is pressed. The title of the pie chart is shown with “dog breeds available,” which will have the breed and age in weeks.

Graphical user interface, text

Description automatically generated

The next snip will be the execution of the pie chart. The following snip is upon pressing the radio button “RESET.”

Chart, pie chart

Description automatically generated

Water Rescue. When the radio button of “Water Rescue” is pressed you will see the pie chart of the available dogs, in this case there is only one dog available, that being a lab retriever mix.

Chart, bubble chart

Description automatically generated

Mountain Rescue. When the mountain rescue radio button is pressed, you will see the available dogs with in the pie chart, and if you hover over each section of the pie chart, it will show you the age in weeks along with the breed.

Chart, pie chart

Description automatically generated

Disaster Rescue. When disaster rescue is pressed, we get the pie chart of the available dogs for this particular rescue, that being three dogs a rottweiler, bloodhound, and German shepherd.

Chart, pie chart

Description automatically generated

Finally, I will show the code for the geolocation map of where the shelter is located in Austin, Texas, by receiving the latitude and longitude of the dog’s location.

Text

Description automatically generated

I will show how each tooltip will change location with every radio button that is pressed for the particular rescue, and it will show the location of the first dog on the list. Like water rescue below.

Chart, map

Description automatically generated

Now, for mountain rescue’s location of the first dog on the list will show a different spot on the map.

Chart, map

Description automatically generated

The next snip will show the first dog on the list available for disaster rescue at a different location near Austin, Texas.

Map

Description automatically generated

With this information provided you can be on your way of creating your own “full stack” project.

## Contact

If you are in need of assistance, please feel free to contact me, Trevor Skillman.