# Meisner CS 340 README Template

## About the Project/Project Title

*This project utilizes CRUD, (create, read, update, delete) for interacting with MongoDB. We are utilizing Mongo to interact with animal data to properly take care of not only our database, but the data itself using filters in a mock “pipeline” which can otherwise be pictured as the entirety of our CRUD process.*

## Motivation

*MongoDB was chosen due to the capabilities that it has as a tool, and its configuration purposes with the CRUD process. Overall, it is important to consider the implications that using a specific tool, or a specific environment may have in deployment. In this case, MongoDB can be used for a variety of different tasks, in the addition to making it easy for customers to use and manage, unlike other database options. MongoDB also is easier to use and set up and use for individuals, and individual projects, unlike SQL which usually utilizes large amounts of data, databases, and is highly utilized in a company setting.*

*The instance in which solely motivates us from a customer perspective comes from applying what can be thought of as a “wrapper” around data that already exists, which is one example of how MongoDB can be applicable to many different instances, as well as malleable for individuals and customers to use alike. In this case, we are simply taking the already existing data, and imposing functionality upon the dataset to understand the exact situations necessary to best outline for the customer to identify issues, and help understand their data better, allowing better business procedures to be outlined and developed.*

## Getting Started

1. *Establish your MongoDB database and account / authorizations.*
2. *Open the CLI*
3. *Run the command: /usr/local/bin/mongod\_ctl start*
4. *Add the Python Module to your Desktop or another file folder of your preference.*
5. *Open this Module and change the 5 numbers located after “localhost” to your own server Port which can be seen when you run the MongoDB command to start the Database.*
6. *Continue to change the line below it, starting with self.database from the allocated database of ‘AAC’ to whatever your specified database is called.*
7. *Add/Create an ipynb file in the same location, if you are creating this yourself the file can be empty.*
8. *Open Jupyter Notebook.*
9. *Click on Desktop (or the location that your files live.)*
10. *Open the ipynb file.*
11. *Change the username and password variables to your own information.*
12. *Press the run button at the top of the page and wait for the output below.*

## Installation

*MongoDB*

*Python 3.6*

*Anaconda Navigator  
Jupyter Notebook*

***MongoDB***[*https://www.mongodb.com/docs/manual/administration/install-on-linux/*](https://www.mongodb.com/docs/manual/administration/install-on-linux/)

***WINDOWS*** *To begin, we need to have MongoDB installed in our local environment. Dependent on the computer you have, the installation changes. For example, if you had a Windows computer, it would be as easy as going to the MongoDB website and downloading their designated installer file. There are many different tools that MongoDB offers, such as MongoDB Compass for visually exploring and interacting with data, MongoDB for VS Code, which is a lightweight and easy to use IDE, and lastly, MongoDB Shell, which is what we primarily are interested in as it allows us to connect, configure, and work with our database. After we download the needed files, we must find this in our downloads folder, we will need to create a custom directory where MongoDB will store its files and run the command* ***md \data\db*** *to set MongoDB’s default location, as well as potentially specifying other locations using the* ***–dbpath*** *command. When this is done, we start MongoDB by running C:\mongodb\bin\mongod.exe or the path to the .exe file if it differs from the one given above. After this, we must open a separate command line, and connect to MongoDB using Mongo shell, i.e., running* ***C:\mongodb\bin\mongo.exe*** *As a special mention, because we are using Linux for our virtual environment, we can go over the Linux installation for Ubuntu as well, though there are also solutions provided for Red Hat Linux, and Debian, as well as SUSE and Amazon which are documented In the link above.*

***LINUX***

*For Linux, we start by importing the given public key used by their package management system.*

***wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc | sudo apt-key add –***

*The operation should respond with OK, but there is an additional chance that an error will be received specifying that gnupg is not installed, which you can install with a* ***sudo apt-get install gnupg*** *with the following attempt to re-import the key.*

*We then create a list file for MongoDB. This command differs dependent on the version of Ubuntu that you are running, which can be found by executing* ***lsb\_release -dc***

*Here is an example of creating the library for* ***Ubuntu 20.04 (Focal)***

***echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-6.0.list****The other commands can be found on the corresponding MongoDB page linked above under step 2.*

*We then continue to reload the local package database by issuing the following command:****sudo apt-get update*** *and finally, we can install the latest stable version using* ***sudo apt-get install -y mongodb-org***

***Python 3.6***[*https://docs.python-guide.org/starting/install3/linux/*](https://docs.python-guide.org/starting/install3/linux/)

*Python 3.6 should hopefully be a lot easier than MongoDB. This is because many different platforms come with some version of python installed already into the local system. Not every platform comes with the correct version of Python, and in some cases, if you are a Mac user, it is important to isolate your python language into a separate environment, so it does not misconfigure the version of python you already have downloaded, meaning you must download something separate, and run commands outside of your “local” environment. On the other hand, using Ubuntu Linux as an example, it is as easy as checking your Python version by running* ***python3 –version*** *and in the likely chance that it is not the correct version, we can then go ahead and follow these commands, which are further documented in the documentation provided on the Python website.****python3 –version***

***sudo apt-get update***

***sudo apt-get install python3.6***

*For those using a different version of Ubuntu, you will need to continue following the commands in the documentation outlined to download 3.8 instead of 3.6, which should also work perfectly fine for our purposes.*

***ANACONDA & JUPYTER***[*https://anaconda.org/anaconda/anaconda-navigator*](https://anaconda.org/anaconda/anaconda-navigator)

*To download Anaconda, all you need to do is follow the link above, and then run the following command, which is also located in the linked page. After this is done, we can use Anaconda to download Jupyter Notebooks, as Anaconda allows us to navigate an interface to launch and download necessary applications needed for different types of data analysis.***conda install -c anaconda anaconda-navigator** 

## Running the application.

After following the getting started part of the README, all you must do is run the application. For a full view version of the application, scroll down to the bottom of the output page, and you should see a link that will take you to a separate tab with the running application.

## Roadmap/Features

After following the run procedure, scroll down to the bottom of the page to interact with the application. You can select between four options, each which changes the output of the geo-location, and changes the animal based off the set requirements. In addition to this, you will see a pie chart outputting the different types of animals able to meet your needs on the left-hand side.

## Explain the steps that were taken to complete the project. The steps that were taken to complete the project involve the placement and imports of the HTML elements, such as my picture, the logo, and the title of the page. In addition, the configuration of the elements below including the data table, pie chart, and geolocation map, integrating the database in with these features additionally had to be created. Identify any challenges that were encountered and explain how those challenges were overcome.

One challenge for me was the placement of the HTML elements at the top. I had wanted to assure that each element visually made sense, and it was surprisingly difficult to dictate the exact area I had wanted everything to be. The way I overcame this was through the research of the HTML and CSS optionality that I had to choose from, and the different things I could use the dash\_table to do.  
  
A second challenge had been understanding the exact components that had been interacting with the different features of the application. To get everything functionally working appropriately, I had to understand the different elements and how to output them in the correct manner to assure everything was programmed correctly. Integrating little changes to assure that the fine detail had been met was important, and understanding how the HTML component integrated in with that was an additional challenge but was all mitigated with research, and laying the architecture out on paper to understand the fluidity of the database and the choices I was visually making when outputting the elements of the database variables.

Graphical user interface, text

Description automatically generatedGraphical user interface

Description automatically generated with medium confidenceGraphical user interface, application

Description automatically generatedGraphical user interface

Description automatically generated

## Contact

Zachary Meisner