# 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

The purpose of projects one and two is to have students familiarize themselves within the connections between MongoDB and Python and how the CRUD functionality works within a database.

## Motivation

The goal of the project is to have the Austin Animal Clinic store the data of all of the patients in a new and centralized database.

## Getting Started

To start, we open the terminal and navigate to the correct directory, in this case, its usr/local/datasets. Then we create a python file that will set the parameters for CRUD the AAC data. Then we will use a main file that imports the code from the first file to make sure that it is running as intended.

## Installation

To get this project started, we will need the Linux terminal in Apporto, MongoDB, Python, and Juypter Notebook.

## 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

*from pymongo import MongoClient*

*from bson.objectid import ObjectId*

*class AnimalShelter(object):*

*"""CRUD operations for Animal collection in MongoDB"""*

*def \_\_init\_\_(self, username, password):*

*if username and password:*

*print("Username and password are:", username, password)*

*# Initializing the MongoClient. This helps to*

*# access the MongoDB databases and collections*

*self.client = MongoClient('mongodb://%s:%s@localhost:44037/AAC'%(username,password))*

*self.database = self.client['AAC']*

*# Complete this create method to implement the C in CRUD.*

*def create(self, data):*

*if data is not None:*

*self.database.animals.insert\_one(data) # data should be dictionary*

*return True*

*else:*

*def update(self, searchData, updateData):*

*if searchData is not None:*

*result = self.database.animals.update\_many(searchData, {"$set": updateData})*

*else:*

*return "{}"*

*return result.raw\_result*

*def delete(self, deleteData):*

*if deleteData is not None:*

*result = self.database.aniamls.delete\_many(deleteData)*

*else:*

*return "{}"*

*return result.raw\_result*

*return False*

*def read(self, data):*

*if data is not None:*

*return self.database.animals.find\_one(data)*

*else:*

*raise Exception("nothing to read because data parameter is empty")*

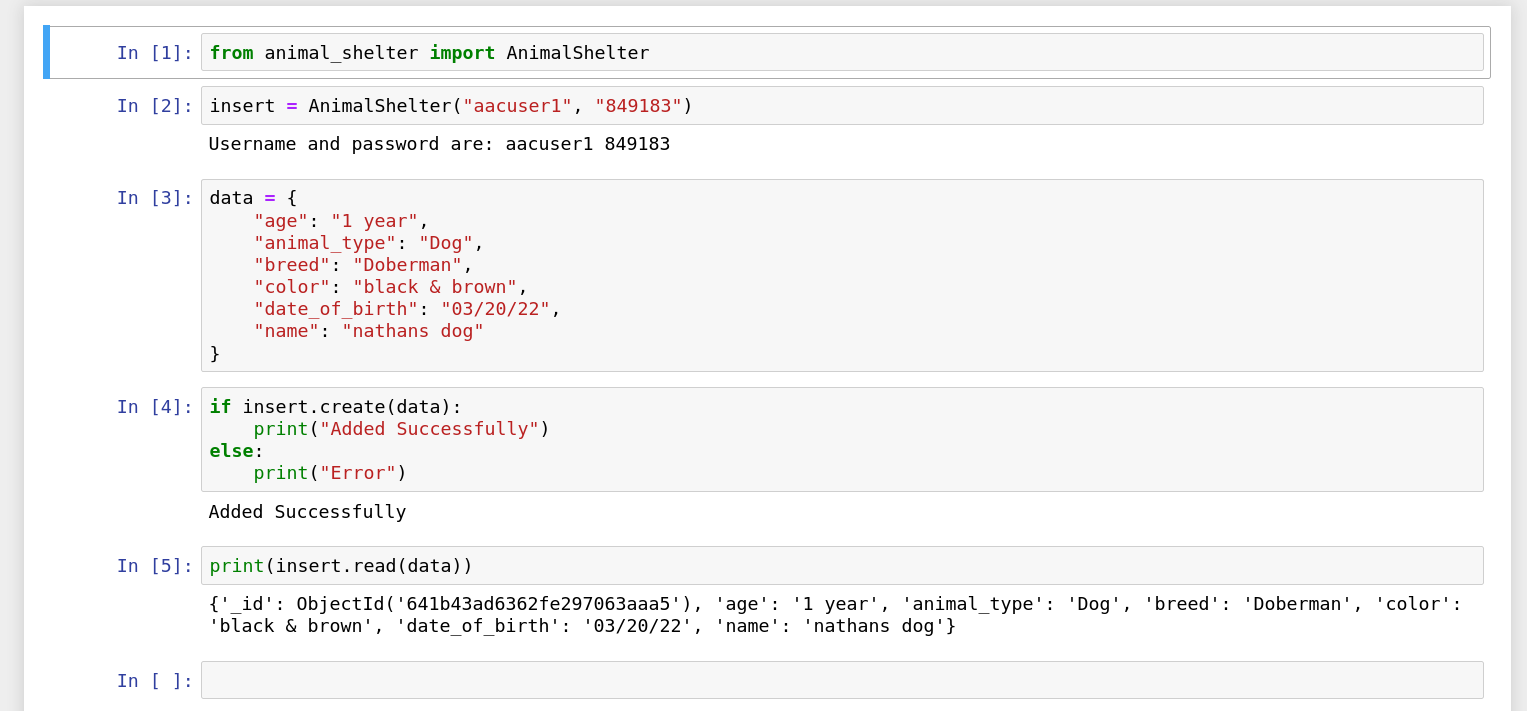
Above is the code needed to initialize and connect the file to the database. We start by importing the PyMongo and bson. Then we create a class called AnimalShelter which initializes and connects the database as well as have all of the CRUD methods within itself.

### Tests

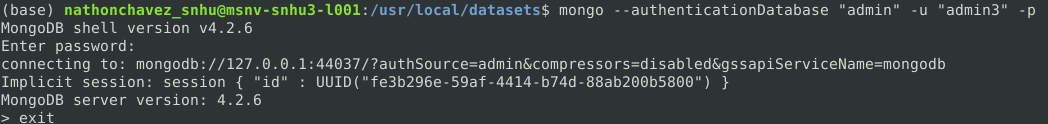
Testing is far more simple, we will create a separate file and then import the class from the first file into second. Then we will create a variable that accesses the data within the database, create a new object in the form of a dictionary, and return that object. If everything is good, I will not see any errors, otherwise, its not good and needs reworking.

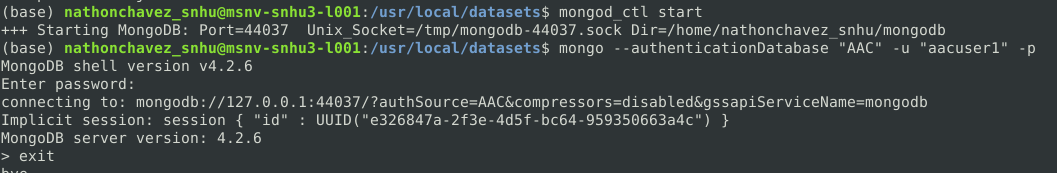
### Screenshots

*Provide screenshots that demonstrate your work.*

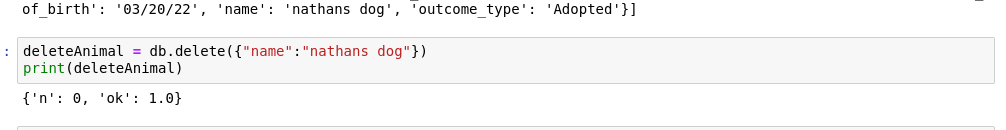
**

## Project One:









CRUD stands for Create, Read, Update, and Delete, and it is a set of basic operations that are commonly used in relational databases to manage data. In Python, CRUD operations can be performed on a database using various libraries such as SQLite, MySQL, or PostgreSQL. In the project we created a new animal, read its data, update its info and deleted the data.

Within the program, we used pymongo and bson object. We used pymongo to connect and utilize the functionality and assets of MongoDB and bson module which is a library for pymongo which gave us functionality within for the code to work. There are several working parts within the CRUD code. Each segment started with a method that had two or three parameters. If the data parameter existed and was not empty, then we would CRUD the data otherwise return with an error message.

## Contact

Your name: Nathon Chavez