# README: CRUD Python for AAC

## About the Project/Project Title

This project is a CRUD python-based module that creates, reads, updates, and deletes entries from a database using MongoDB.

## Motivation

This project was started as a way to collect information about certain types of dogs across different shelters throughout the US to be able to find candidates for certain types of disaster rescue operations. The goal of this module is to organize data from various shelters to identify and categorize these animals so that the selection process can be easier.

## Getting Started

In order to use this module, download and place the .py script into the source directory. It then can be imported into your python script as follows:



In order to be able to use the script in mongoDB, the database must be loaded into mongo first as well as authenticated as an administrator to read and write to the database. In the AnimalShelter class, the username and password to the database must be provided in order to access the database and use the module for its CRUD functions.

## Installation

Graphical user interface, text

Description automatically generated

## Usage

### Code Example

Text

Description automatically generated

### Tests

Once it is linked into your current file, you can use the create, read, update, or delete functions to add or find entries in the database.

Create:

*Text

Description automatically generated*

A picture containing graphical user interface

Description automatically generated



Read:

A picture containing calendar

Description automatically generated

*A picture containing graphical user interface

Description automatically generated*



Update:

A picture containing table

Description automatically generated

A picture containing diagram

Description automatically generated



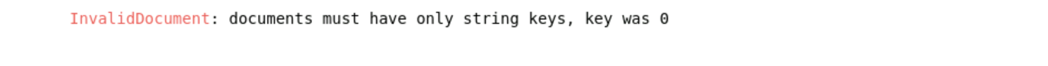
Delete:

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated



Filtering and Graph code:

Text

Description automatically generated

### Screenshots

Graphical user interface, text, application

Description automatically generatedImporting a CSV file:

Text

Description automatically generatedCreating an admin account, ensure authentication to the database and collection:

## Dashboard Screenshot:

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

## Describe the required functionality of the project. Include the screenshots or screencast taken while testing and deploying your dashboard (Step 6) as proof that you have achieved the required functionality.

The functionality for this code to load a database into MongoDB and use a CRUD python module in order to create, read, update, and delete entries in the loaded database. The module was then used in Jupyter to create a DASH dashboard to create an interactive filterable datatable, pie chart, and map of the different animals and shelters used by the company to find rescue animals for different scenarios.

## Describe the tools used to achieve this functionality and a rationale for why these tools were used.

## Be sure to explain why MongoDB was used as the model component of the development, including what specific qualities or capabilities it provides for interfacing with Python.

The tools used to achieve this functionality were MongoDB, Python, Jupyter, Dash, and Plotly. MongoDB is a good starting point as the model component of the development because it uses JSON structure for its overall structuring as a database. It works well with python because it has its own built in python driver called PyMongo, which makes interfacing with python much easier than other coding languages. This compatibility is used to create the CRUD module that is used in the beginning steps of this finished dashboard. After the module is made, a Dash framework is used to provide better structure to the dashboard and giver further functionality with datatables, graphs (in conjunction with Plotly), and interactive components such as maps.

## Explain the steps that were taken to complete the project.

As mentioned in the last paragraph, the project started with learning the basics of MongoDB, loading databases into the system that can be used by the CRUD python module that was to be created. Then after the python module was coded and tested using Jupyter, the module was imported into a Dash framework with its basic functionality focusing on the read function of the python module. This was then used to create a data table that was stylized using HTML components to provide a better view of the database, as well as options to make the data more user friendly. Finally, Plotly was used to create a pie graph of the data, and other Dash functionality was used to make all components interactive with the dataset.

## Identify any challenges that were encountered and explain how those challenges were overcome.

Honestly, this project was pretty tough overall and there are still parts that I am having trouble getting to work properly, even though I am pretty sure the code it written correctly. Most of the other challenges I overcome by using the resources section of the different modules, and delving into the book provided for the class. The online resources really helped when it came to the interactive components of the Dash framework and the last few weeks of the project. I would still like to figure out the issues I have been having with a few components of the finished product but overall I am pretty happy with how it works in terms of the functional pieces of code.

## Contact

Kyle Hake, kylehake@snhu.edu