# CS 340 Project README

## About the Project/Project Title

**CRUD Using Python Commands**

This project is about making CRUD (Create, Read, Update, Delete) database commands in Python for use in a MySQL database. It also creates a web app that utilises this Python code to display potential rescue dogs from the AAC database.

## Motivation

The Python base project makes reading and creating items for the database easier allowing for safer and easier access by more employees. It also allows for more programs to be built on top of it that can interface with the AAC database. The web application allows for easy sorting of the database for the specific use case of finding dogs to recruit for specific rescue jobs.

## Getting Started

To start the project you need to get the files into a Jupyter notebook file on a system that has or has access to the AAC database. Then run the IPYNB file on Jupyter Notebook.

## Installation

For installation, you will need Jupyter Notebook which can be downloaded at <https://jupyter.org/install> and you will need access to the database. If your system does not have access to the database or all of the required libraries contact IT.

## Usage

The module uses MongoClient from PyMongo to connect and interface with the database. This tool was chosen for its ability to connect securely through Python to the MongoDB database. This web program logs into the AAC database and displays the total list of animals. The user can then choose to customize searches in the chart itself or use the premade searches. Premade searches are made to filter for specific rescue dog use-cases. The web app also displays a chart of the outcome of the dogs left in the filtered chart. The user can also select a dog on the table to display its location in a geolocation app.

**Tools Used**

*MongoDB/Python*

Mongodb in conjunction with Python were chosen as the model component of the program because of their flexibility. MongoDB, as a NoSQL database, has a lot of flexibility on how it is used and scaled. PyMongo also has good tools to leverage MongoDB with its library.

*Dash*

Dash is a strong HTML framework that can utilise the tools built in Python to display the database directly to the user. This allows users to interact with an ever updating version of the database, pulling and displaying information in real time.

**Steps to Complete Project**

1. Build the Model - created a Python program with the PyMongo library MongoDB that can perform the basic CRUD operations and log into the AAC database
2. Test the Model - tested model with basic Python code by pulling the the Python code as a library in a Jupyter Notebook file
3. Build Model Functionality - built in specific customer functionality into the base Python CRUD library for specific searches
4. User Web App - created a user web application that utilizes the base CRUD module to sort the AAC database with preset sort functions and custom filtering options the web App includes a Pie chart of shelter outcomes and geolocation feature at the bottom of the page.

**Roadblocks Encountered**

The Python and CRUD and creation and testing went pretty smoothly and performed well when iterated on when creating the new functionality for the web application. The dash application works, but a bit slowly. The load-times are probably because it seems to be reloading the database into memory every time it has to interaton on it/ search it. A solution could be to load it into memory once and then use the loaded database to search but that would prevent the database from showing real time changes to the database after the page is loaded the first time. This solution would most likely lead to a significantly larger overhead in programing time for this and further iterations.

## Contact

Tyler Owens: Owenstylerlucas@gmail.com