# 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

This project uses a CRUD method with Python to sort data that has been uploaded to a MongoDB to allow an automated search through the data based on certain parameters. This project is used by importing the required files into your project, altering the files to fit your MongoDB and to declare the database you are using and then create the data you want to import or read from the MongoDB database. This project used Pymongo which works alongside MongoDB but uses the Python commands in a way that is easier to read while easily calling on the database that is being used. The CRUD functions let us Create, Read, Update, and Delete records in the database. This project mainly utilizes the Read function to find and pull up information about the dogs that are available for training.

## Motivation

This project was created to be able to find and identify dogs for training in different fields to be able to find and help in the rescue of humans or other animals. The different training areas are Water Rescue, Mountain or Wilderness Rescue, and Disaster or Individual Tracking. This project allows the search to be ran under specific parameters that the company requires as there are specific parameters that the dogs must fit to be able to go through the training such as being a specific breed or a specific age range. Using this project makes finding these animals easier as it lists out the animals that match the parameters in a data table that is interactive based on the training type. This allows us to see which animal fits the trainings parameters while showing us where the animal is on a geographical map to retrieve them and start the training.

## Getting Started

To run this program, the user will first need to install the necessary tools. First, the user will need to install and set up MongoDB. This is best done on a Linux system as python is the default application which allows for a quicker installation. The user needs to start the MongoDB using the correct command (For example: /usr/local/bin/mongod\_ctl start). **Important note: MongoDB will terminate if the window is closed so it is essential to be sure that mongo is running once the application is reopened.**

Once the required tools are installed, the user will be able to execute commands to add, delete, and modify records. This can be done by importing a dataset to work with. (For example: mongoimport –port #### (where the # is the specific port for the user) --db AAC --collection animals --file /usr/local/datasets/aac\_shelter\_outcomes.csv --type csv). After the dataset is imported, the user will need to select the database to work in to perform the tasks of adding, deleting, modifying, and searching the dataset.

To use the CRUD Python file in your own project: Download the script and place it in the root directory, then import it into the location you want to use it with by using the following command:

from AnimalShelter import AnimalShelter

## Installation

This project utilizes two primary tools, Python and MongoDB. Python is a default program for Linux, but it can be installed on Windows machines by using the default packages. I also used Jupyter Notebook for this project to run and test the code.

You can install Jupyter Notebook by utilizing a program called Anaconda.

1. Open a web browser
2. Navigate to anaconda.com/products/distribution
3. Download the Anaconda Distribution file
4. Install the Anaconda Distribution file
5. Run the Anaconda Distribution file
6. Click on the Install Jupyter Notebook option
7. Follow the installation wizard for Jupyter Notebook
8. Launch Jupyter Notebook
9. Congratulations, Jupyter Notebook is now installed!

You can install Python to your system using the following steps:

1. Open a web browser
2. Navigate to python.org/downloads/
3. Select the system you are using (For example: If you use a Windows system you can select Windows and be redirected to the download files specifically for Windows)
4. Download the latest version of Python that matches your systems criteria (For example: Windows installer (64-bit) for a Windows 64bit system)
5. Follow the prompts to install Python to the system
6. Congratulations, Python is now installed!

You can install MongoDB to your system using the following steps:

1. Open a web browser
2. Navigate to mongodb.com
3. Select the version of MongoDB to download
4. Select the platform (For example: Using a Windows system you will want to select Windows)
5. Select the package
6. Download the MongoDB installer
7. Run the MongoDB installer
8. Follow the installation wizard steps to install MongoDB to the system
9. Congratulations, MongoDB is now installed!

Set up MongoDB by utilizing the MongoDB installation manuals on the mongodb.com/docs/manual/installation website. This site shows the different manuals for each system specific needs to get MongoDB running correctly on the downloaded system.

For the AnimalShelter CRUD python script you will need to install pymongo and you can install it directly by using the command

pip install pymongo

or you can install pymongo into your environment by using the project requirements command

pip install -r requirements.txt

## Usage

This project can be used to filter through the animal records from the imported dataset along with allowing the user to create, read, update, and delete records in the dataset. Some tasks that can be performed are:

1. Importing the dataset into the system
2. Creating new users
3. Adding new animals to the dataset
4. Searching for animals in the dataset
5. Updating animal data in the dataset
6. Removing animal data form the dataset

For this specific project, we created a data table, geolocation chart, and pie chart that populates with the specific parameters to filter through the animal dataset and show the information for the dogs that fit the training parameters that the company is looking for.

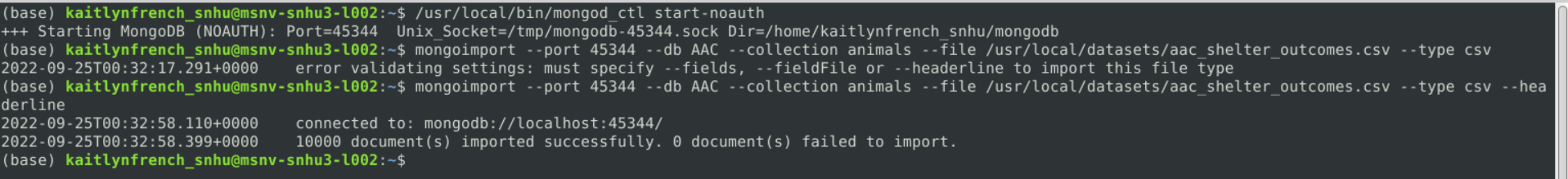
This project can utilize these tasks either directly through the MongoDB or by using the CRUD Python file in the project in Jupyter Notebook.

### Code Example

1. Import the database you are working in along with the collection

Graphical user interface, website

Description automatically generated



1. Create a new user and set your login credentials for the database you will be working in

**Text

Description automatically generated**

1. Log in to MongoDB using your authentication login and your database. This allows you to read or write in the given database

**Text

Description automatically generated**

1. You can use this database library to search through different breeds in the animal collection or to add your own. To search for a specific breed, we can use the following command:

db.animals.find({breed:”GermanShepherd”})

To get further details on this breed we can use the explain function as well

db.animals.find({breed:”GermanShepherd”}).explain()

Your output will look like this:

Text

Description automatically generated

### Screenshots

Graphical user interface, application

Description automatically generated

Chart

Description automatically generated with medium confidence

Graphical user interface

Description automatically generated with medium confidence

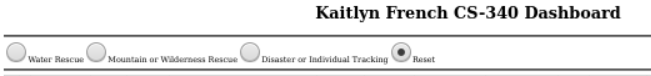
This project can be run through Jupyter Notebook using the CRUD python file.Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Import the CRUD Python file into the Jupyter Notebook code to utilize the CRUD python file. In this project, we did this to create a data table, a geolocation chart, and a pie chart that populates from the animal records dataset.



We created radio buttons to show different dataset parameters used to populate the data table, geolocation chart, and pie chart:

Water Rescue:

A picture containing graphical user interface

Description automatically generated

Mountain or Wilderness Rescue:

A picture containing application

Description automatically generated

Disaster or Individual Tracking:

A picture containing chart

Description automatically generated

We also included a rest button to reset the table and charts to the original output that we receive when running the program initially.

Table

Description automatically generated with medium confidence

### Tests

We can run tests using the Jupyter Notebook application to ensure that functions are working in the database. This can be done for each function of CRUD and Jupyter will let you know if there is something missing or wrong with the way the python script file was altered so you can easily switch back to the python script file and update your alterations accordingly. Another good reason to run tests on the code is there are multiple ways to get the desired output and testing this code allows you to see which way works best and alter the code accordingly to fit the best output.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

## Contact

Your name: Kaitlyn French