# CS 340 Austin Animal Control README

## About the Project/Project Title

We are creating a Python module for the Austin Animal Control (AAC) for their shelter animals. It will allow them to perform CRUD operations in MongoDB where they keep track of their animals. This was created by importing their information into a MongoDB. From there, a python script and a Jupyter script were created to work together to give users easy access to the database information. It also allows them to filter the information in useful ways.

## Motivation

The reason for creating the code is so that the AAC will be able to create, read, update, and delete records of animals in their shelter. This will help them keep better track of the animals that they work with. CRUD functionality is essential for interacting with all databases. They currently store all their animal records in a CSV file and want to import them into a MongoDB database.

## Getting Started

To get started, you’ll need to have your MongoDB configured and run it with authentication. Next, you’ll need to have both the AnimalShelter.py and the CRUDscript.ipynb in the same folder and on the same device that you ran your MongoDB. You will need to alter both files to match your local MongoDB username, password, database, and port.

## Installation

To use the program, you’ll need to have MongoDB configured as well as bash. From there, you can download the files you need to perform the CRUD operations from my Github here. Lastly, you will need Jupyter installed to run the CRUD script.

## Usage

To use the program, simply run the ipynb script and it should display the table with all animals for you. It should also display a pie chart that breaks down the animal info for you. There’s also a map that shows the location of each animal. You can use a dropdown menu to filter which animals are shown on the display.

### Code Example

The ipynb uses the Jupyter dash framework to display all of the info contained in the mongoDB in a nicely formatted table. It also allows us to display a pie chart offering a different view of your info. The AnimalShelter class uses a library called Pymongo to connect and interact with MongoDB. It also uses the library bson.objectid which allows python to create MongoDB objects that will be accepted by the database. It defines the database being connected to and also which user on the database. Then, it utilizes commands like instert() and find() to perform CRUD operations.

### Tests

To run a test on the python script, you will need to run the CRUD script that was included separately. Before you run it, you’ll need to edit it and add the username and password of your MongoDB user. From there, you can add animals like in the screenshot below to your database. When running the script, it will display whether it was successful.

### Screenshots

Default Dashboard State

Graphical user interface

Description automatically generated with medium confidence

Water Rescue Filter

Graphical user interface

Description automatically generated with medium confidence

Mountain Rescue

Graphical user interface

Description automatically generated with low confidence

Disaster Rescue  
Graphical user interface

Description automatically generated

Back to all  
Graphical user interface

Description automatically generated

## Challenges

One of the biggest challenges I faced was figuring out that I needed to import additional framework that was not included in the example file I was given. This was needed to add the proper logo to the web page.

## Contact

Your name: Ryan Mackenzie

Email: ryan.mackenzie@SNHU.edu