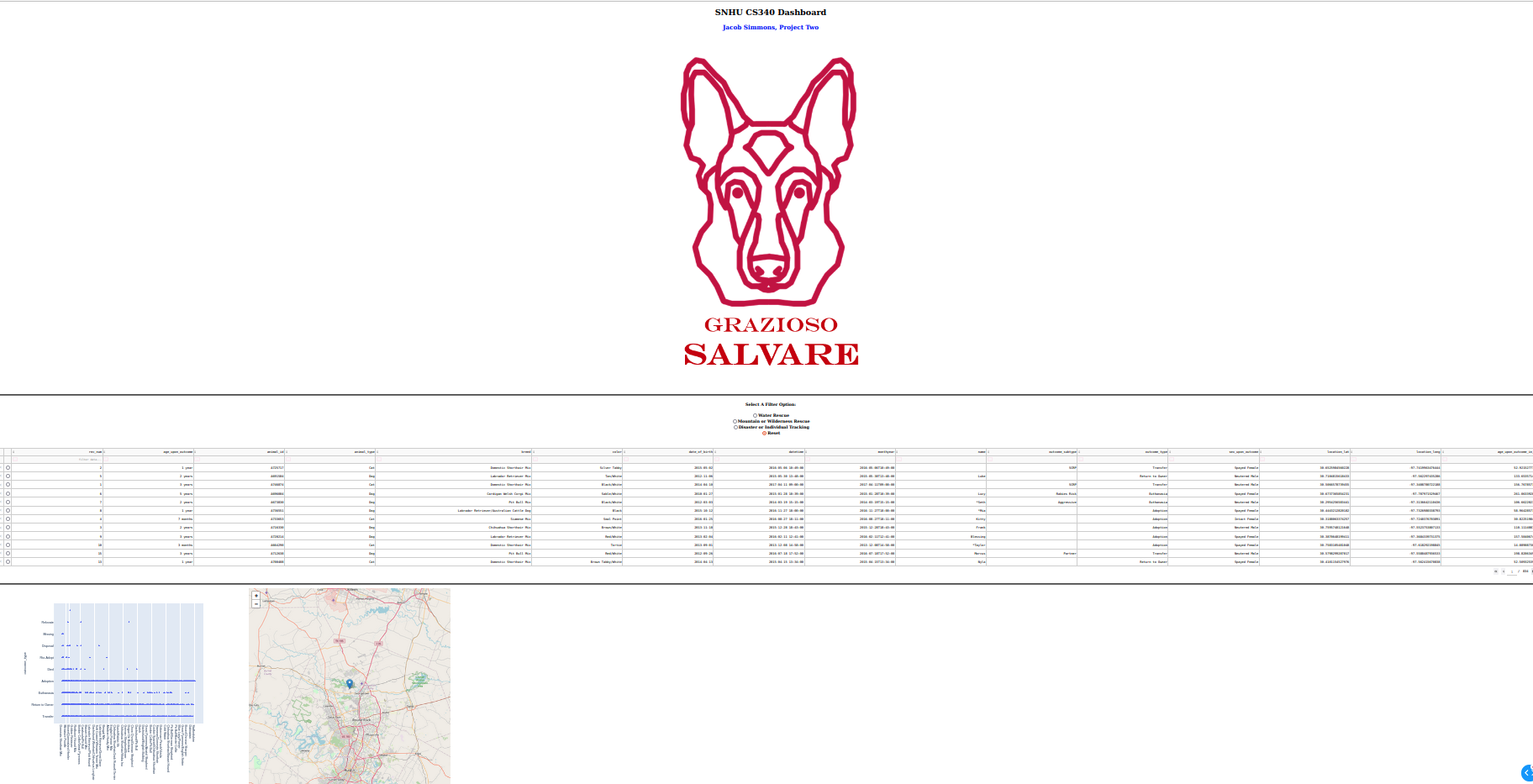
# CS 340 README

Jacob Simmons, 17 Jun 2023



## About the Project/Project Title/Purpose

This project is designed to allow the user to search a large database of dogs with differing traits such as breed, age, location, sex for rescue. This databased is ran using MongoDB with python generated APIs.

## Motivation

The motivation for using MongoDB is due to the No-SQL abilities of MongoDB and its ease of communication with files specifically when it comes to integrating with Python. The creating the basic CRUD functionalities in Python as simple and integrate very well with MongoDB.

## Installation

All the coding and testing was performed within the SNHU virtual lab using Linux with the Apporto server system. MongoDB was ran using a Linux system where the Linux and Mongo shells were established for unique purposes.

Essentially, the .csv file was uploading into the MongoDB and a unique user was established with admin read and write credentials. A python file was developed and the pymongo libray was used to create a link to the mongo AAC database using the credentials of the unique user – this was executed in the class constructor. Within this python file, CRUD functions were developed to modify the database as requested. Finally, a Juptyer notebook was used to develop act as a "driver" to explore and validate the CRUD functionalities – this serves as evidence of the development.

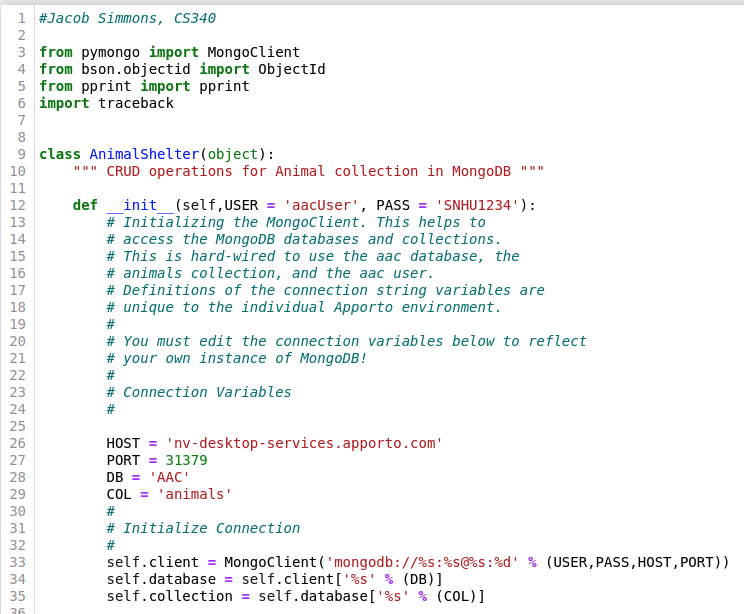
## Usage

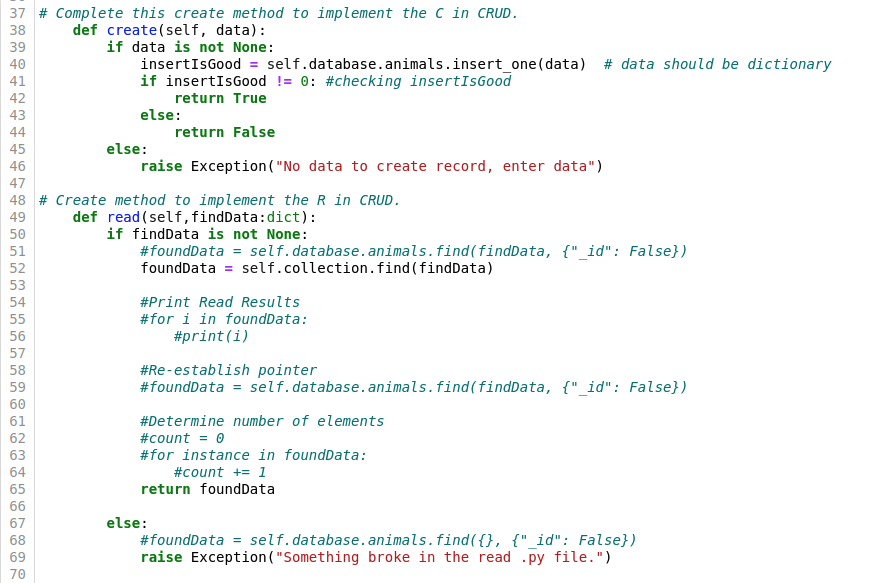
*Use this space to show useful examples of how your project works and how it can be used. Be sure to include examples of your code, tests, and screenshots.*

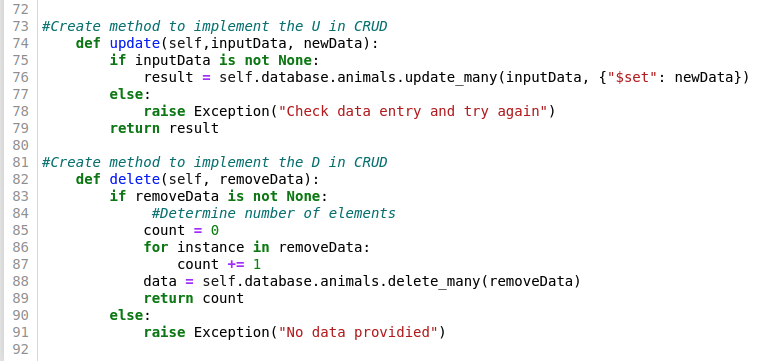
### Code Example

*Show what the library does as concisely as possible. Developers should be able to figure out how your project solves their problem by looking at the code example. Make sure that your code is short and concise.*

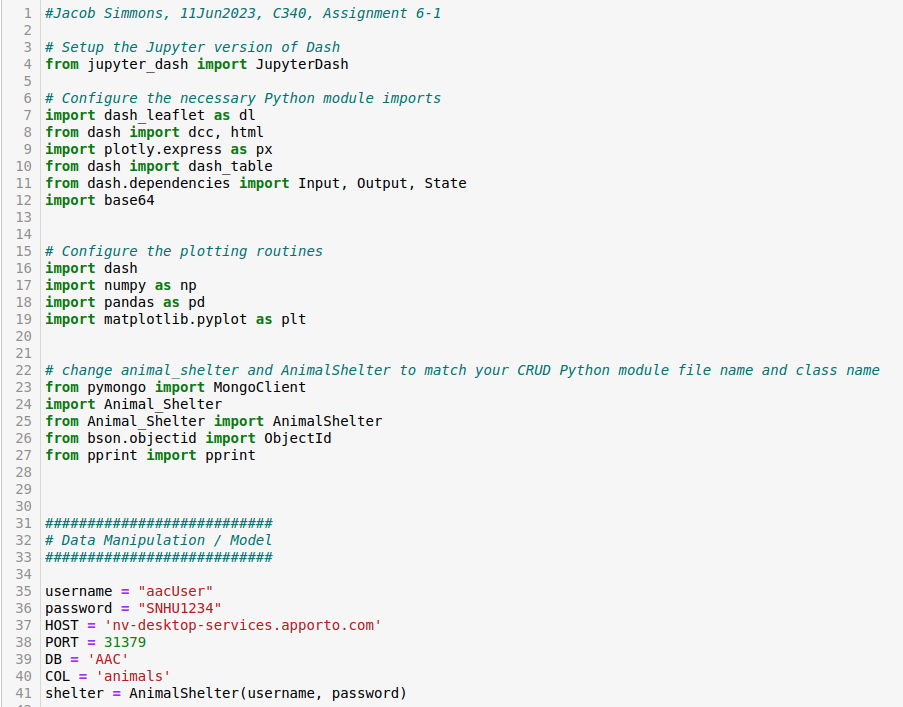
Below is an example of the CRUD python code that was generated.

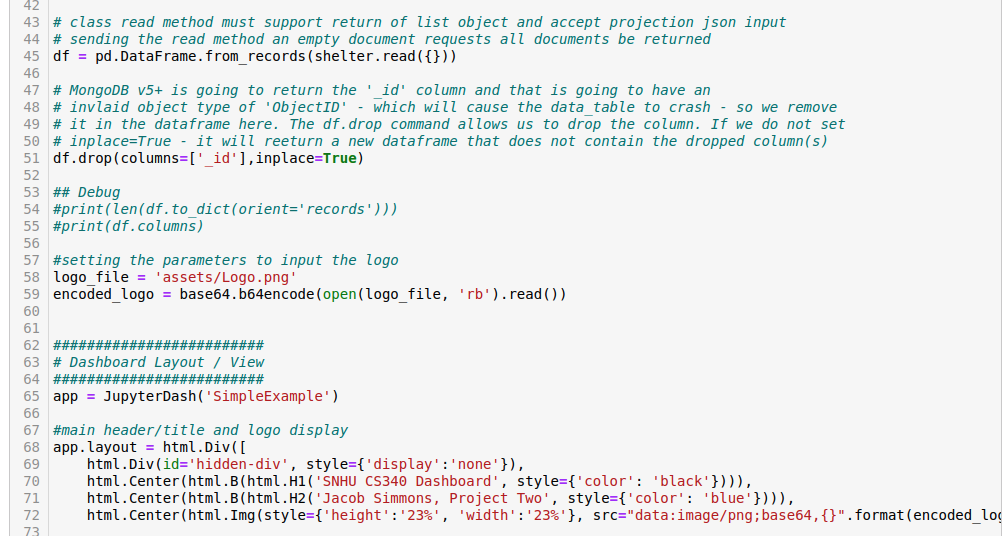
****

****

****

* **This is the code for the Jupyter Notebook for the Dash, MongoDB, and Pandas Interface**

****

****

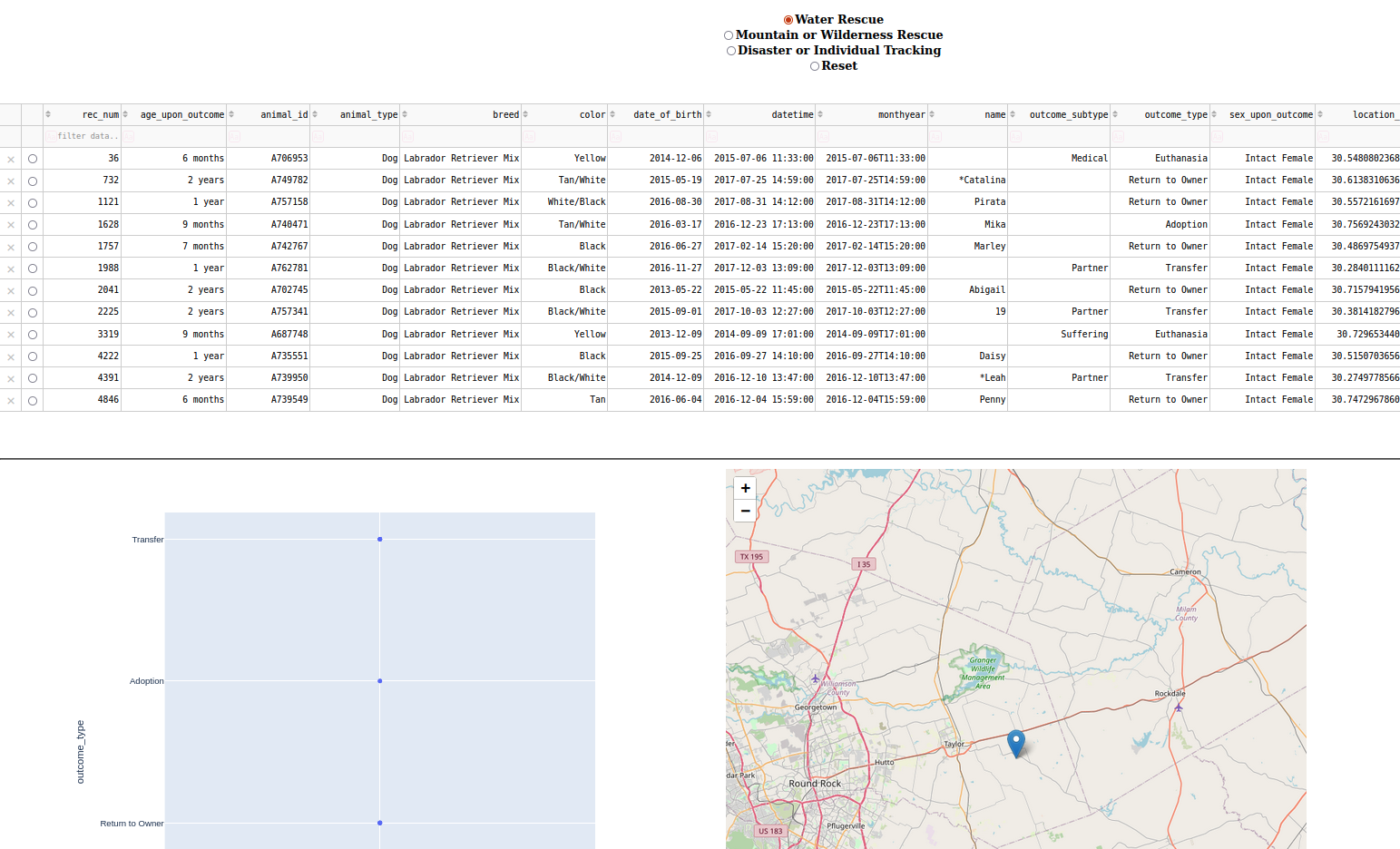
****

****

### Tests

*Describe and show how to run the tests with code examples.*

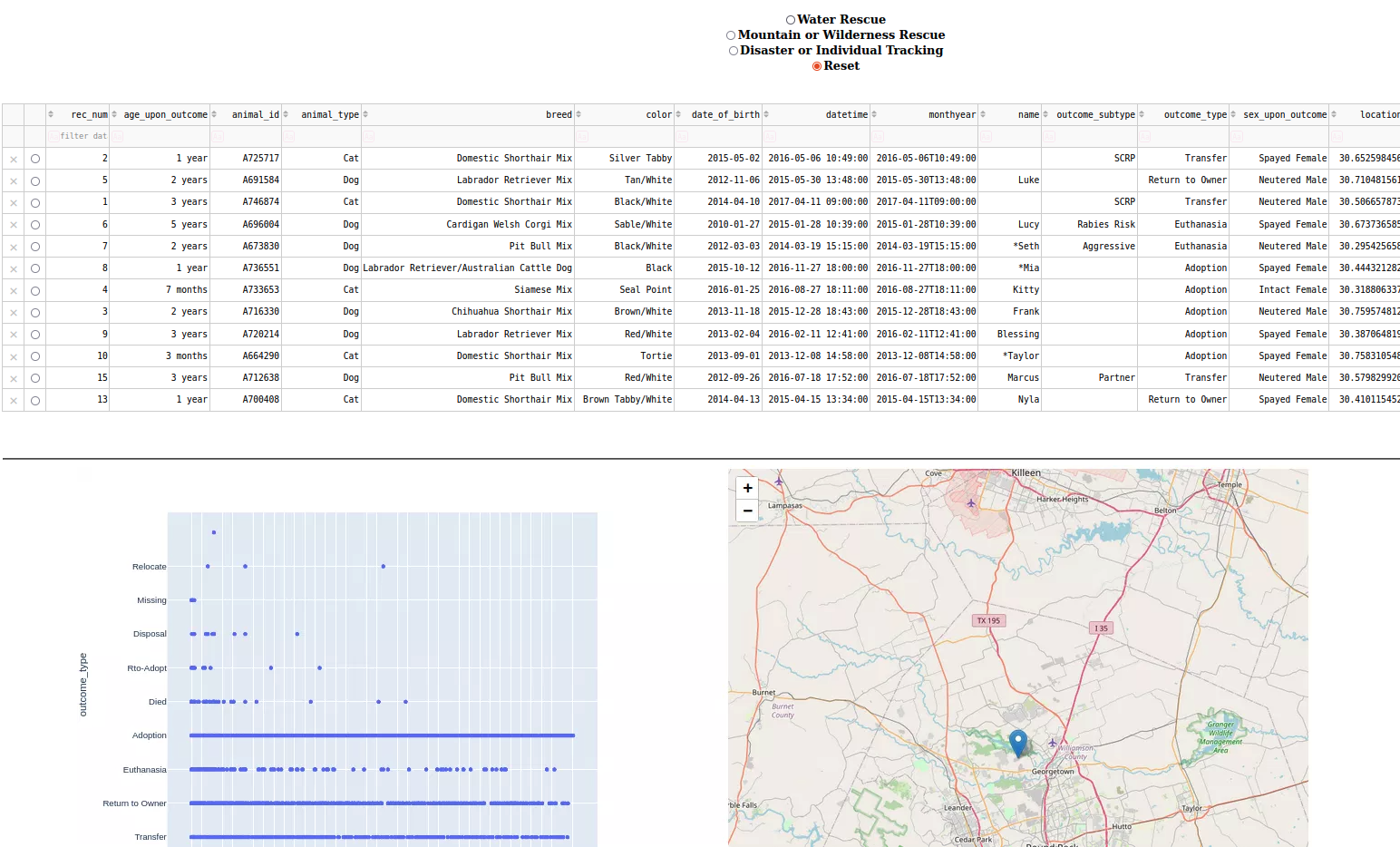
**This shows the water rescue filter working**

****

**This is showing the Disaster Tracking filter work**

****

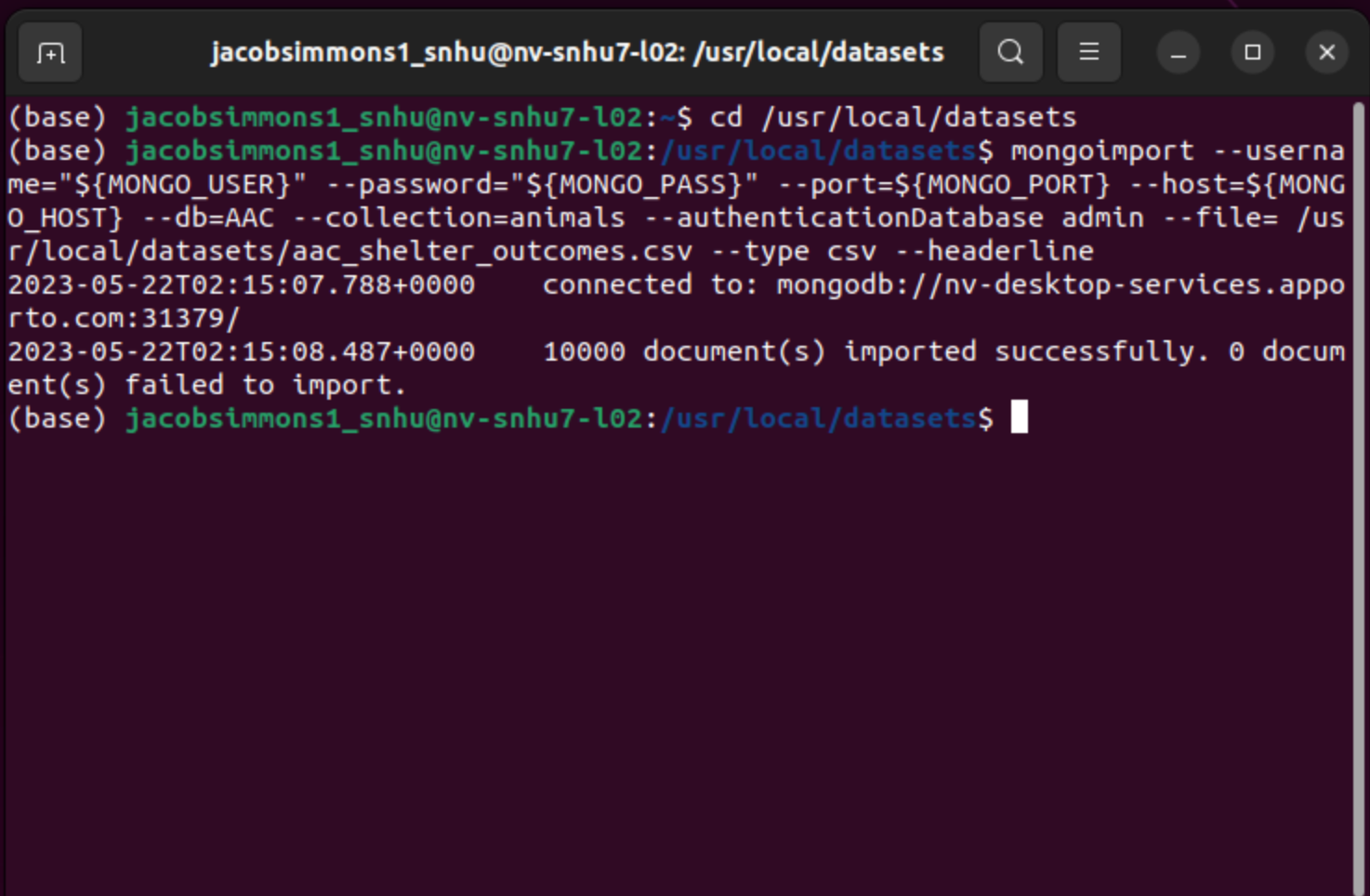
**This is showing the reset filter button working**

****

### Screenshots

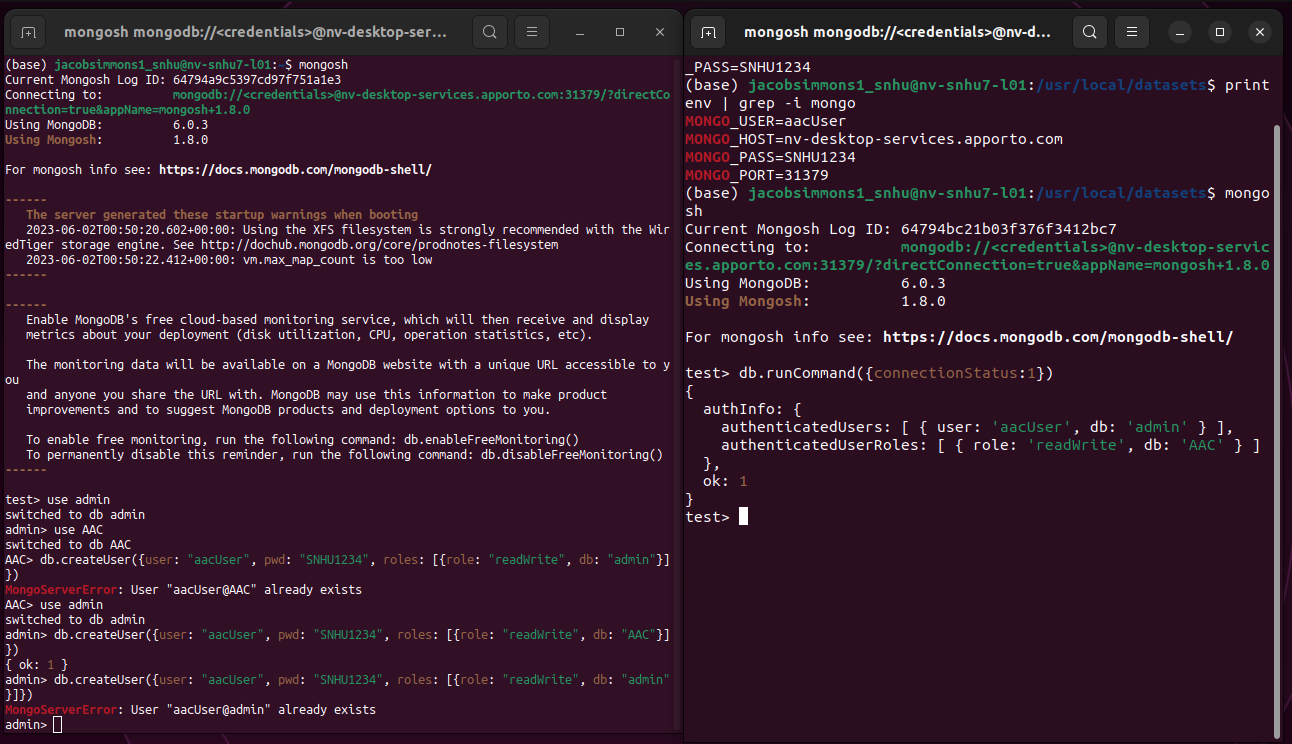
*Provide screenshots that demonstrate your work.*

***Database Commands (Mongo Shell and Screenshots): MongoImport***

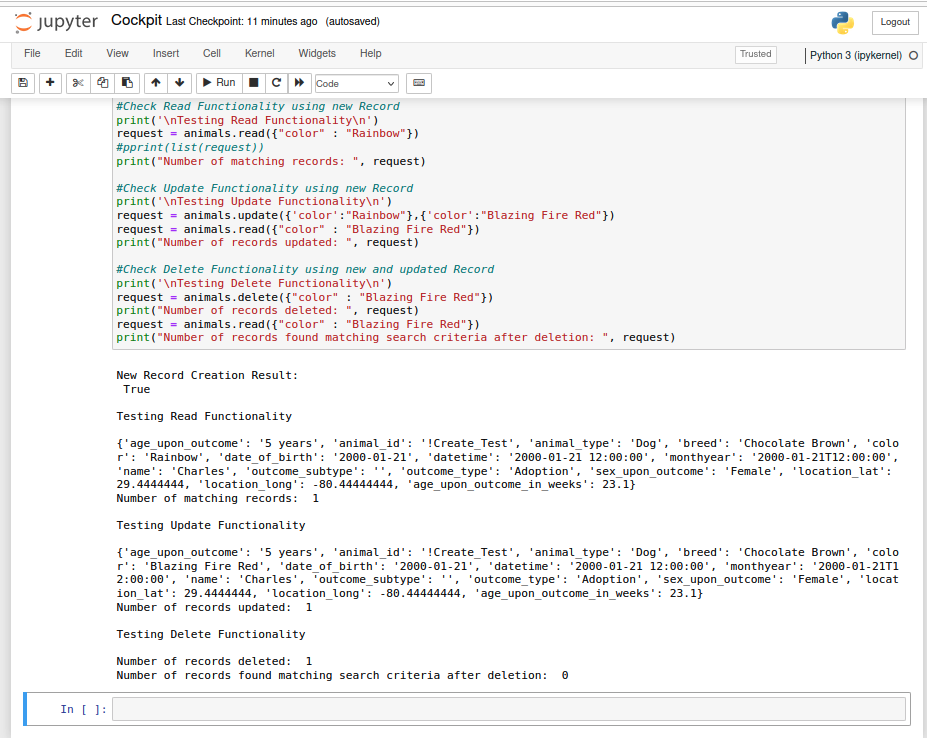
****

***Database Commands (Mongo Shell and Screenshots): Authentication***

* ***Create user "aacUser" (left terminal window)***
* ***Validate Login for "aacUser"(right terminal window)***

******

***CRUD Python Module (Python Code and Screenshots): Script to Test CRUD Functionality***

**

## Roadmap/Features (Optional)

*Provide an open issues list of proposed features (and known issues). If you have ideas for releases in the future, it is a good idea to list them in the README. What makes your project stand out?*  
Reviewing the code, additional improvements should be made for data validation and input security. This will allow the program to be much more robust and ensure long term CRUD functionality.

Further improvements could be made in determining the number of matching records, number of records updated and number of records deleted using the built in pyMongo functionalities instead of "for loops".

*Note: This section is optional for the purposes of this assignment. If you choose not to fill out this section, remove it from your final README file.*

## Contact

Jacob Simmons