# CS 340 README

*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

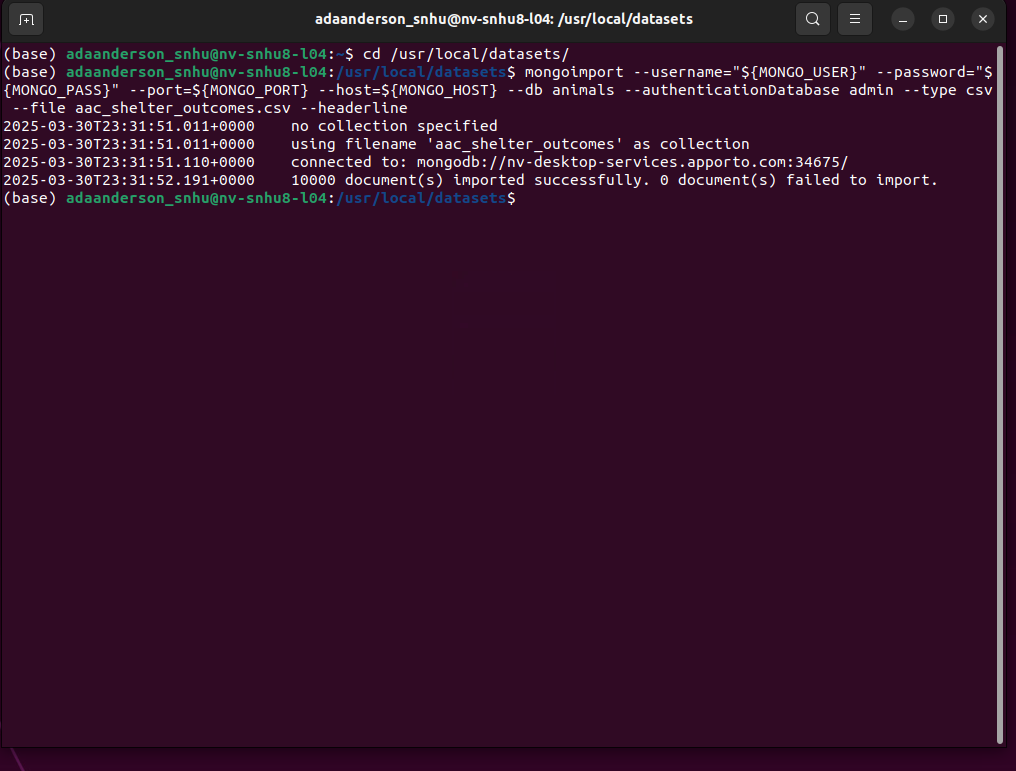
*This program is designed to access the database “animals’ to create, read, update, and delete (CRUD) queries through MongoDB, and was created in Python. Alongside this, the user can search and filter through the database within the dashboard. The dashboard also contains info about geolocation.*

## Motivation

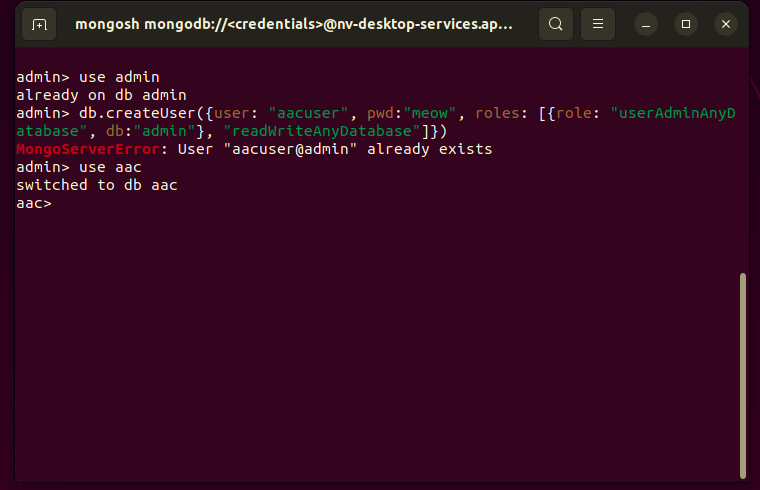
*This program was made to allow users to create, read, update, and delete animal data in an easy, manageable way by using the various implemented Python functions in MongoDB.*

## Getting Started

1. *Import the aac\_shelter\_outcome.csv to MongoDB:*



1. *Then, you must have a created aacuser account and password with the following permissions in the admin database:*

**

1. *Finally, switch to the aac database and use the desired CRUD functions.*

## Installation

*-A current version of Python*

*-MongoDB*

*-The animal shelter outcomes csv file.*

## 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.*

*Create data, read data, update data, and delete data:*

### Code Example

Example of create code:

*Def create(self, data)*

*try:  
   
 result = self.database.animals.insert\_one(data) # data should be dictionary  
 return True if result.inserted\_id != 0 else False  
except Exception as e:  
 print(f"Error inserting the document {e}")  
 return False*

*To use this “create” function, you would follow the following example:*

*animals.create({“breed”:”Russian Blue”})*

### Tests

*This test is a simple test to add an animal breed and read the animal breed:*

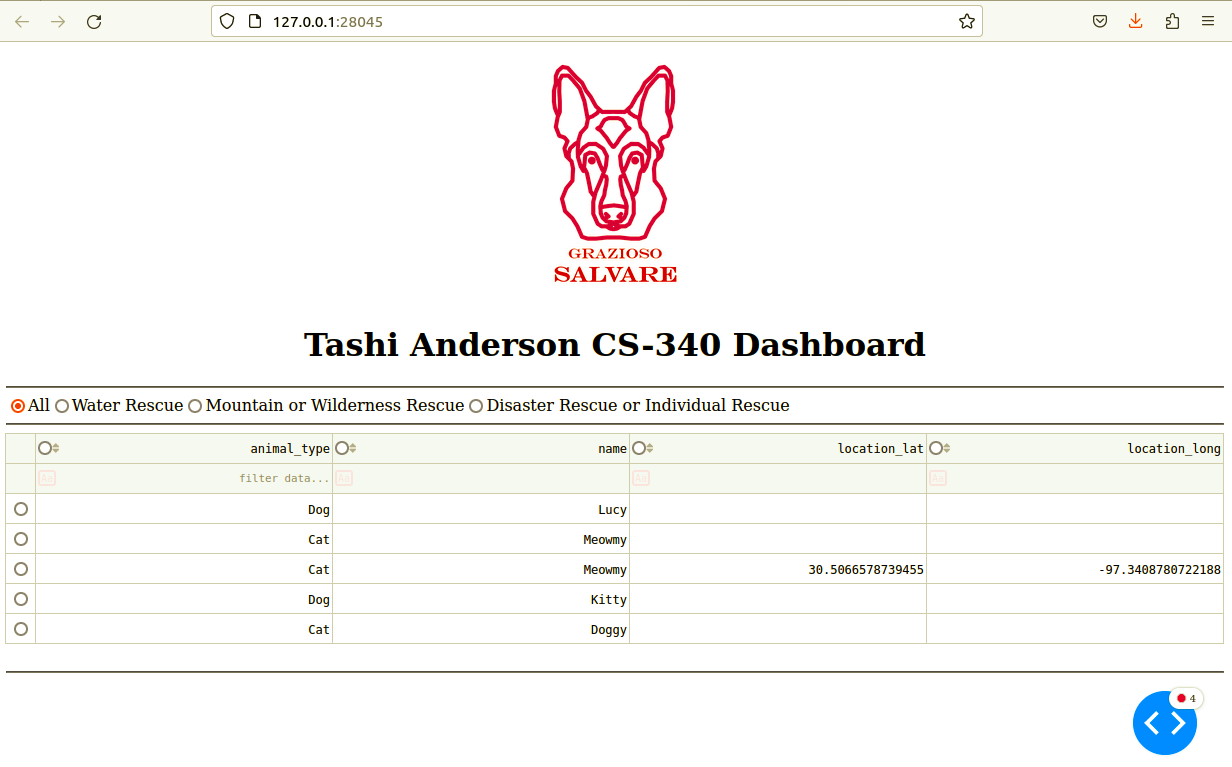
*#CRUD testing****import****milestone****from****milestone****import****AnimalShelter  
animals = AnimalShelter("aacuser","meow")  
#simple create and read test  
print(animals.create({"breed":"Russian Blue"}))  
  
query = animals.read({"breed":"Russian Blue"})*

*# simple update and delete test*

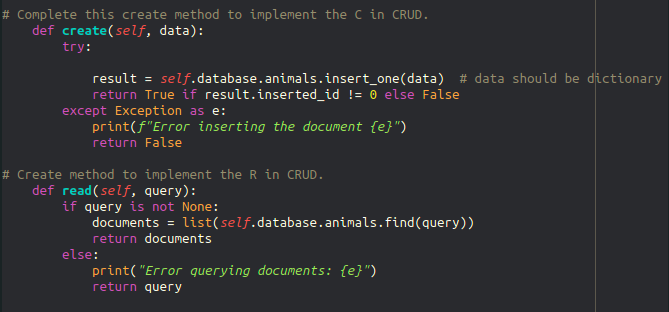
*print(animals.update({"breed":"Russian Blue"}, {"outcome\_type":"Adopted"}))*

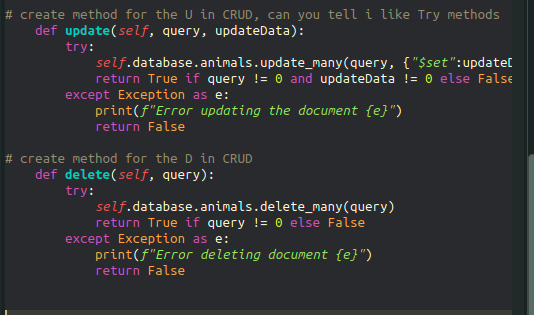
*print(animals.delete({"breed":"Russian Blue"}))*

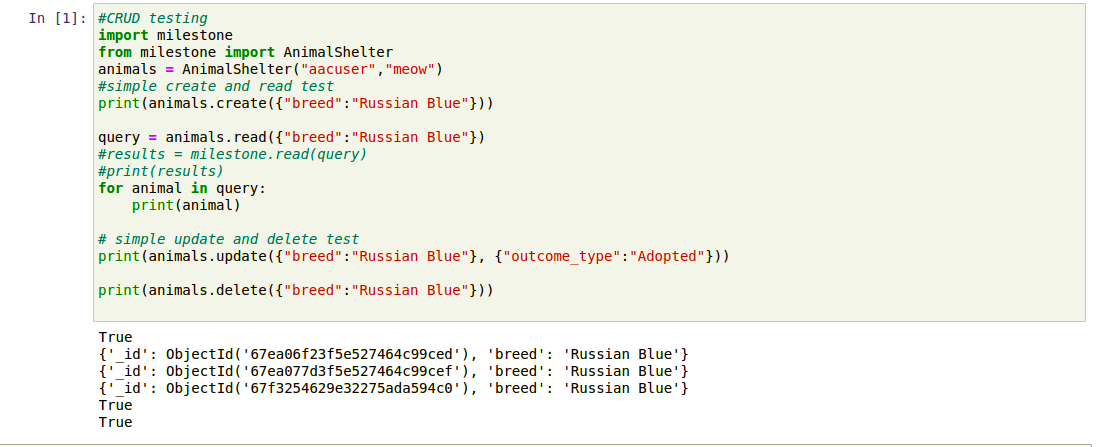
***Running the dashboard***

once the file has been ran, the dashboard should run. You can now select from the various filters to support your search, and even view geolocation data from a map.

### Screenshots of the code itself



**



**Errors**

one major error I encountered was the fact that my animals database refused to update with the proper values. This has continuously affected how I created my code and troubleshooted it. One work around I implemented was by directly importing the aac file into the milestone file, which was dodgy at best. In a proper set up, this code should properly work as intended.

## Contact

Tashi Anderson

SNHU