# CS 340 README

## About the Project/Project Title

The CRUD Python module allows for someone to utilize the python module to interact with MongoDB to create, read, update, and delete documents within the database. Utilizing Dash & MongoDB we are able to create a dashboard that would a user could interact with and see the information visually displayed with the information with the specific database information along with interactive filters to see specific accurate information within the database.

## Motivation

Through educational experience with MongoDB the reason I’ve decided to develop this module has to do with just learning how to interact with MongoDB via Python.

## Getting Started

1. Download the following CRUD.py file
2. Place inside your project directory
3. Import within your Python source
4. Setup MongoDB connection
5. Setup Dash Initialization
6. Enjoy

## Installation

1. Some way of utilizing the Python software language: visit python.com
   1. Or have some other IDE like Visual Studio, PyCharm, VS Code..
2. Acquire MongoDB - https://www.mongodb.com/
3. Acquire Dash - https://dash.plotly.com/

## Usage

### Code Example

def create(self, data):

if data is not None:

self.database.animals.insert\_one(data) # data should be dictionary

return True

else:

raise Exception("Nothing to save, because data parameter is empty")

return False

The following method allows for the ability to create a document within the database.

The parameter data must be a dictionary of the document information.

Ex. { "name": "Fluffy", "type": "Cat", "age": 3}

def read(self, query):

result = self.database.animals.find(query)

return list(result) if result else []

The following method allows for the ability to read from the database based upon a search query.

The parameter query must be a dictionary of the query information.

Ex. {"type": "Cat"}

def update(self, query, update\_data):

result = self.database.animals.update\_many(query, {‘$set’: update\_data})

return {‘modified\_count’: result.modified\_count}

The following method allows for the ability to update the queried document within the database.

The parameter query & update\_data both need to be a dictionary of the document information.Ex. {"type": "Cat"}

def delete(self, query)

result = self.database.animals.delete\_many(query)

return {‘deleted\_count’: result.deleted\_count}

The following method allows for the ability to delete a document within the database, but deletes all of them with the query.

### Tests

Here are some code snippets able to be copy and pasted once your CRUD object has been instanced.

Create method snippet

document\_to\_create = {“name”: “Fluffy”, “type”: “Cat”, “age”: 3}

created\_result = <insert CRUD object variable name>.create(document\_to\_create)

print(“Creation result:”, created\_result)

Read method snippet

query = {“type”: “Cat”}

read\_result = < insert CRUD object variable name >.read(query)

print(“Read result:”, read\_result)

Update method snippet

update\_query = {“name”: “Fluffy”}

update\_date = {“age”: 30}

update\_result = < insert CRUD object variable name >.update(update\_query, update\_data)

print(“Update result: “. update\_result)

Delete method snippet

delete\_query = {“name”: “Fluffy”}

delete\_result = < insert CRUD object variable name >.delete(delete\_query)

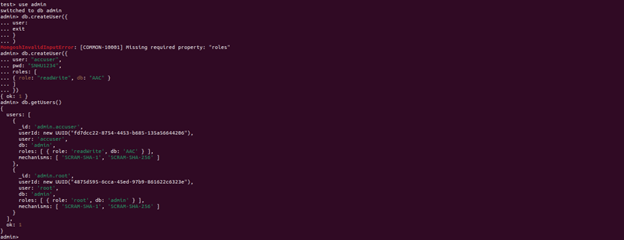
print(“Delete result: ”, delete\_result)

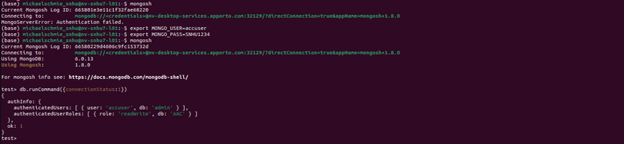
### Screenshots

Importing of the data to database

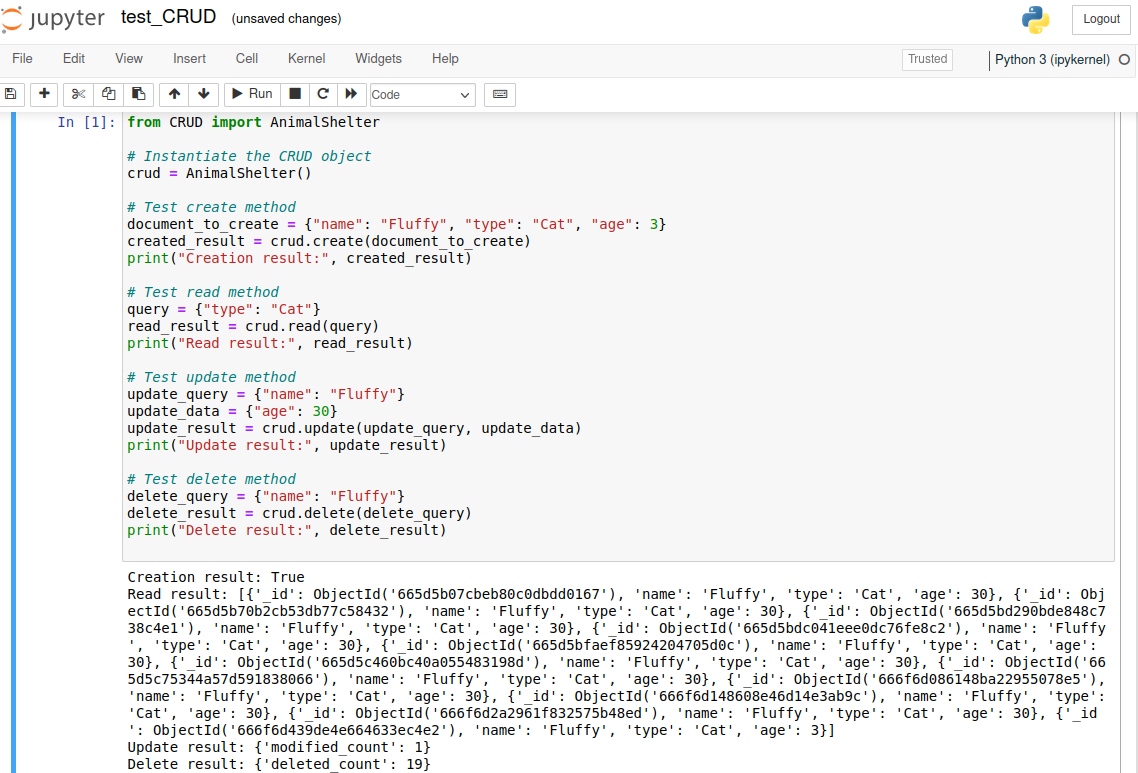


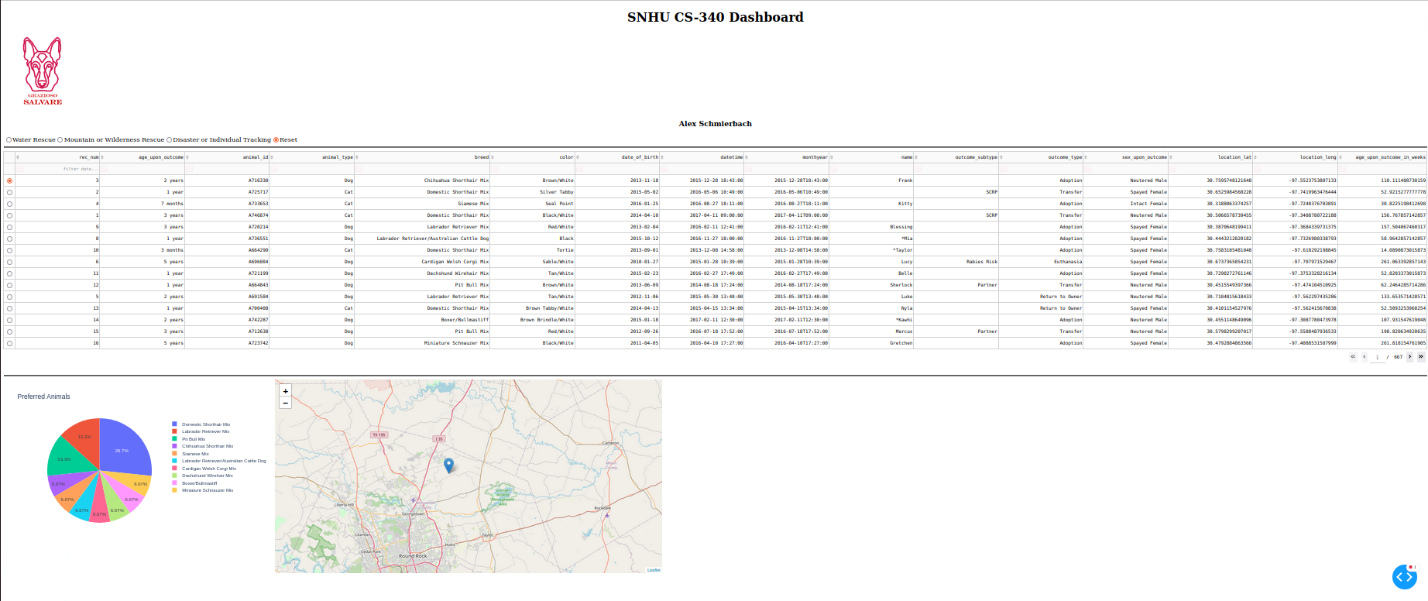
User Authentication

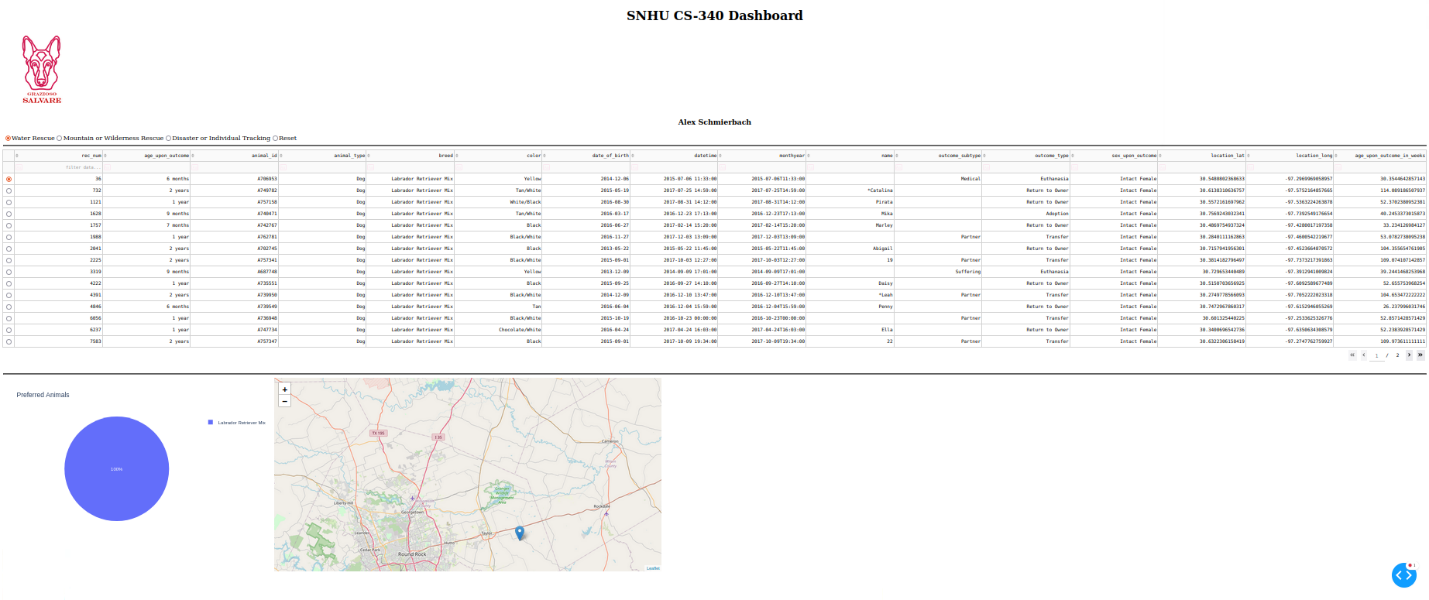




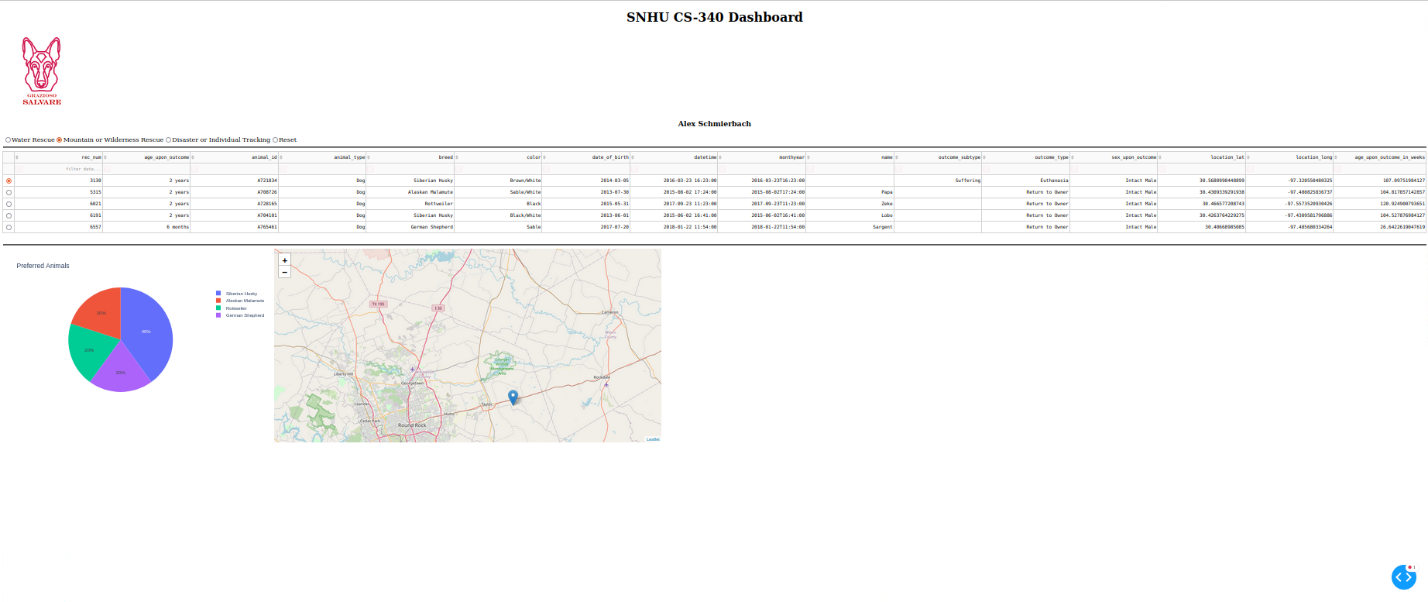
Test Functionality



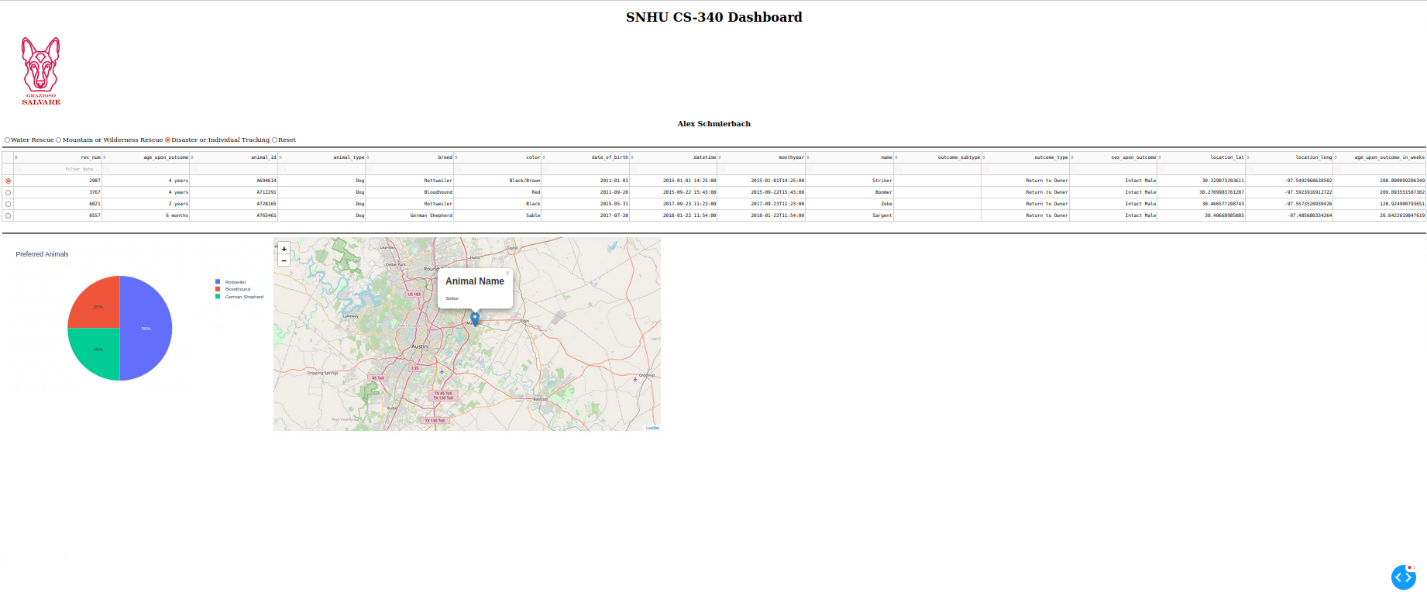
Dashboard – Everything/Reset Filter

Dashboard – Water Rescue Filter 

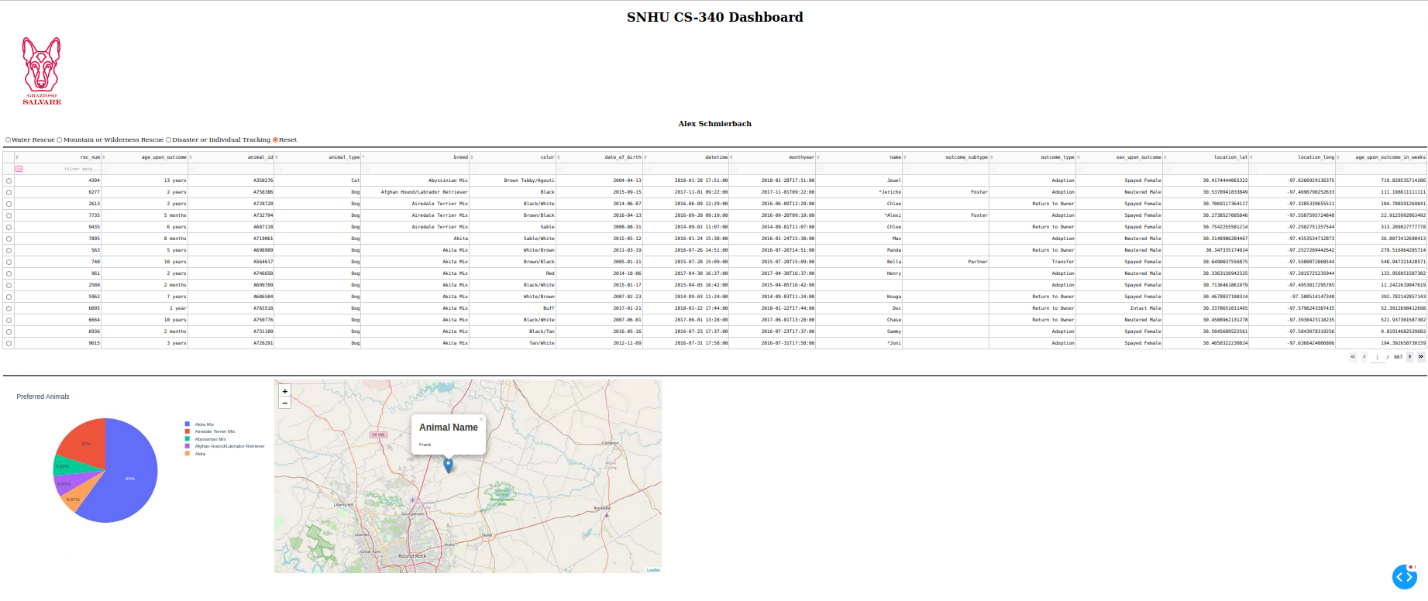
Dashboard – Mountain or Wilderness Rescue Filter



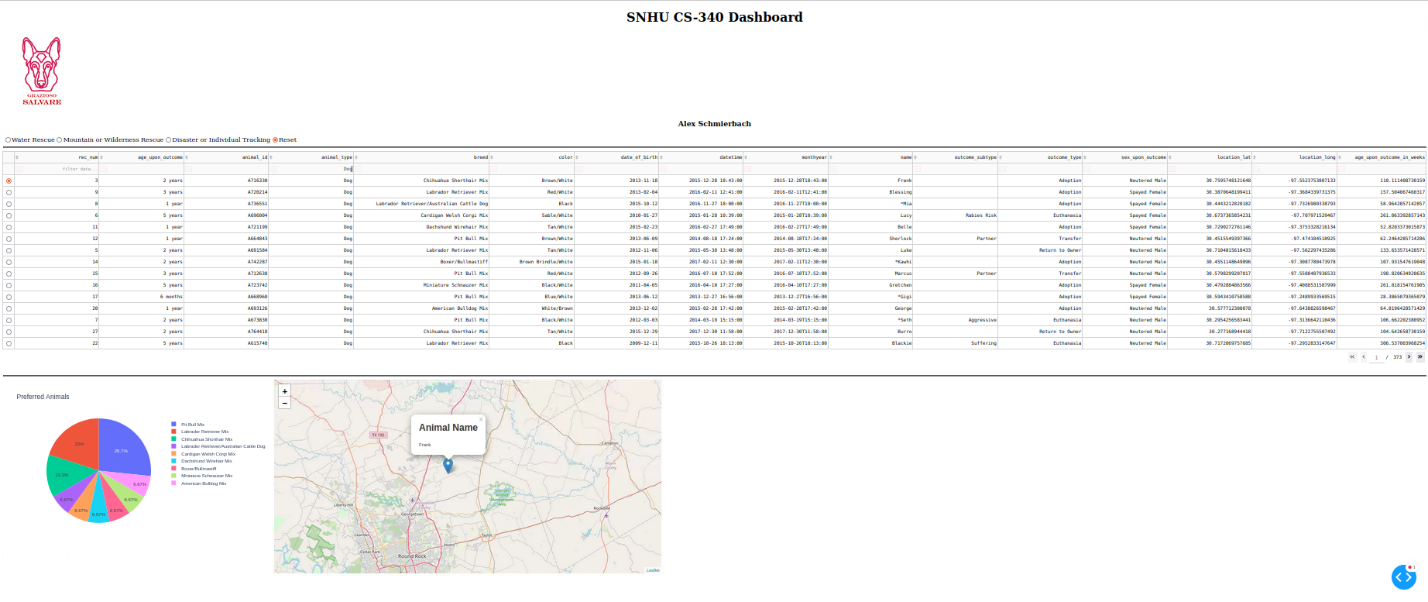
Dashboard – Disaster or Individual Tracking



Dashboard – Specific Column Filter via Click Column Breed



Dashboard – Specific Column Filter via Typed Box Below Column Header animal\_type



## Roadmap/Features

Here is a list of future features that expected to be implemented upon future updates:

1. Create (Completed)
2. Read {Completed)
3. Update {Completed)
4. Delete {Completed)
5. Create Interactive Dashboard (Completed)

## Contact

Your name: SpoonsTheUtensil

How do you write programs that are maintainable, readable, and adaptable?

I wrote programs that are maintainable, readable, and adaptable in a way that takes the common practices required within the language that I am utilizing along with using best practices when it comes to programming to stay organized so to speak within the codebase. Keeping things simple, clear and concise allows a programmer to easily check off these boxes.

How do you approach a problem as a computer scientist? How did your approach to this project differ from previous assignments in other courses? What techniques or strategies would you use in the future to create databases to meet other client requests?

Personally, I consider the client’s needs first then through testing think of other quality of life features that could enhance the project, but only after the client’s needs have been met. I read through the requirements documentation and go from there as it is kind of like a checklist for yourself and it is easy to follow. The approach I utilized this time around required more research into Dash and MongoDB as I haven’t been familiar with or used them at all. Honestly, just talking with the client is all that I would need to determine further databases with clients.

What do computer scientists do, and why does it matter? How would your work on this type of project help a company to do their work better?  
A computer scientist job is to solve a real world problem through the development and analysis of programming and it matters because sometimes this could be a simple solution that could potentially benefit a plethora of people and an argument that the solution could be of low overheard meaning low cost thus resulting in a win-win. The work done within this project might help a company that requires them to do analysis within a database and this dashboard would help them be able to filter through the plethora of data to find the exact information regarding the data set that is needed for whatever purpose they impose.