# CS 340 README Template

## About the Project/CS340 Animal Shelter

*Read,create,delete and update information in database using the web dashboard interface*

## Motivation

*This project was created to update the database for the animal shelter. This was created to be able to create entries, read entries, update entries and finally delete entries. I created the crud python module to handle these tasks. The tasks here are critical to what the animal shelter is trying to do. I used mongo db for the database because of how seamlessly it works with python. I then created a web dashboard to allow the users to explore the data base and search through the data by breed,sex and age .*

## Getting Started

*Use the source file provided to create this program. Open in your ide of choice. Change the main file. Change “username” and “Password” You will also need access to the python programming language as this is what was used. You will also need Jupyter Notebooks to verify tests.*

* *log in using aacuser or admin account*
* *Insert your mongo db port number*
* *Create custom test by added test to notebook*
* *access the web dashboard*
* *search database and use dashboard for your needs*

## Installation

*Mongodb*

*Jupyter notebook*

*python*

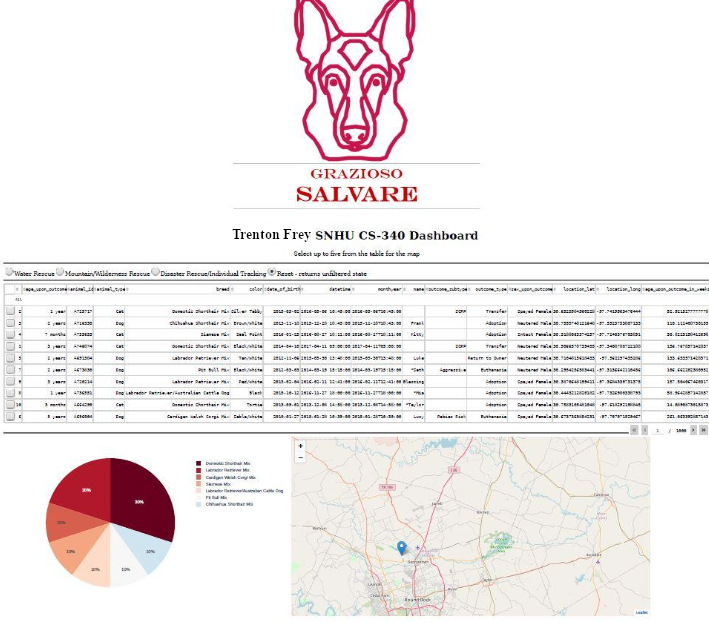
*dash*

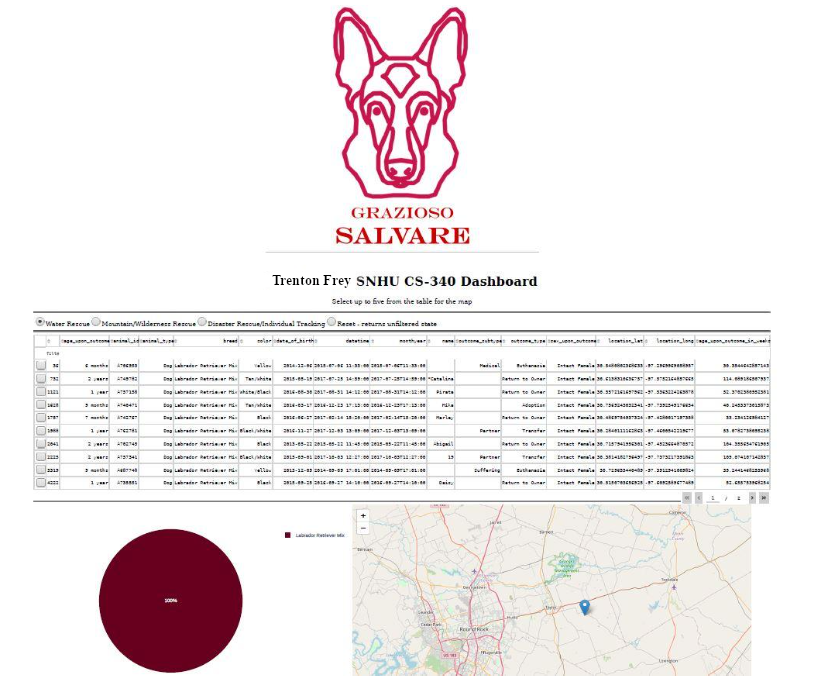
*pandas*

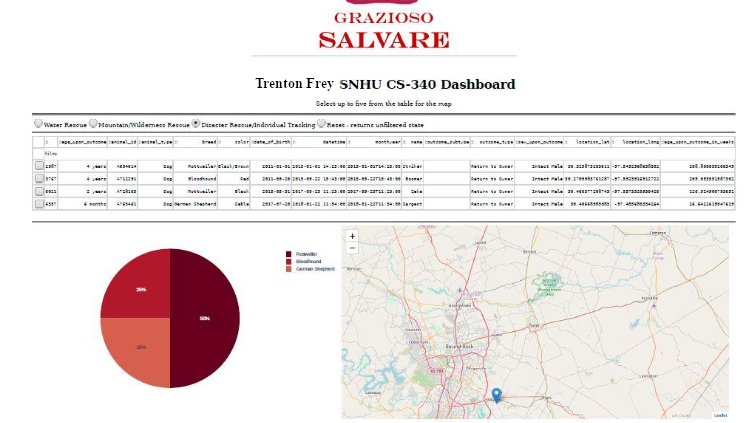
*plotly*

## Usage

*The use of this application comes down to three main uses. Shorting the database by using different buttons to short differently. The other use is to plot animal lotions on the map and update them in real time. This allows for a good tracking ability. The third and final use is to create pie charts to show the data visualization.*

**

**

**

### Code Example

def deleteData(self, target, count):

if target is not None:

if count == 1:

try:

delete\_result = self.database.animals.delete\_one(target)

pprint("Count- " + str(delete\_result.deleted\_count))

if delete\_result.deleted\_count == 0:

print(delete\_result)

return True

else:

print(delete\_result)

return True

except Exception as e:

return False

*animal\_data = [*

*{*

*"name":"stone",*

*"type":"dog"*

*},*

*{*

*"name":"hanki",*

*"type":"dog"*

*},*

*{*

*"name":"walker",*

*"type":"dog"*

*}*

*]*

*@app****.****callback(*

*Output('datatable-id', 'data'),*

*[dash****.****dependencies****.****Input('demo-dropdown', 'value')])*

***def*** *update\_output(value):*

***if*** *(value* ***==****'water'):*

*df* ***=*** *pd****.****DataFrame(list(shelter****.****read\_all(*

*{*

*"breed": {*

*"$in": [*

*"Chesapeake Bay Retriever",*

*"Labrador Retriever Mix",*

*"Newfoundland"*

*]*

*},*

*"sex\_upon\_outcome": "Intact Female",*

*"age\_upon\_outcome\_in\_weeks": {*

*"$gte": 26.0*

*},*

*"$and": [*

*{*

*"age\_upon\_outcome\_in\_weeks": {*

*"$lte": 156.0*

*}*

*}*

*]*

*}*

*)))*

***elif*** *(value* ***==*** *'mountain'):*

*df* ***=*** *pd****.****DataFrame(list(shelter****.****read\_all(*

*{*

*"breed": {*

*"$in": [*

*"German Shepherd",*

*"Alaskan Malamute",*

*"Old English Sheepdog",*

*"Siberian Husky",*

*"Rottweiler"*

*]*

*},*

*"sex\_upon\_outcome": "Intact Male",*

*"age\_upon\_outcome\_in\_weeks": {*

*"$gte": 26.0*

*},*

*"$and": [*

*{*

*"age\_upon\_outcome\_in\_weeks": {*

*"$lte": 156.0*

*}*

*}*

*]*

*}*

*)))*

***elif*** *(value* ***==*** *'disaster'):*

*df* ***=*** *pd****.****DataFrame(list(shelter****.****read\_all(*

*{*

*"breed": {*

*"$in": [*

*"Doberman Pinscher",*

*"German Shepherd",*

*"Golden Retriever",*

*"Bloodhound",*

*"Rottweiler"*

*]*

*},*

*"sex\_upon\_outcome": "Intact Male",*

*"age\_upon\_outcome\_in\_weeks": {*

*"$gte": 20.0*

*},*

*"$and": [*

*{*

*"age\_upon\_outcome\_in\_weeks": {*

*"$lte": 300.0*

*}*

*}*

*]*

*}*

*)))*

***else****:*

*df* ***=*** *pd****.****DataFrame****.****from\_records(shelter****.****read\_all({}))*

***return*** *df****.****to\_dict('records')*

### Tests

def testUpdate(self):

with self.assertRaises(Exception):

testShleter=AnimalShelter(user,password)

testShelter.update(None, sampleUpdate,1)

*from AnimalShelter import AnimalShelter*

*import urllib.parse*

*animal\_data = [*

*{*

*"name":"stone",*

*"type":"dog"*

*},*

*{*

*"name":"hanki",*

*"type":"dog"*

*},*

*{*

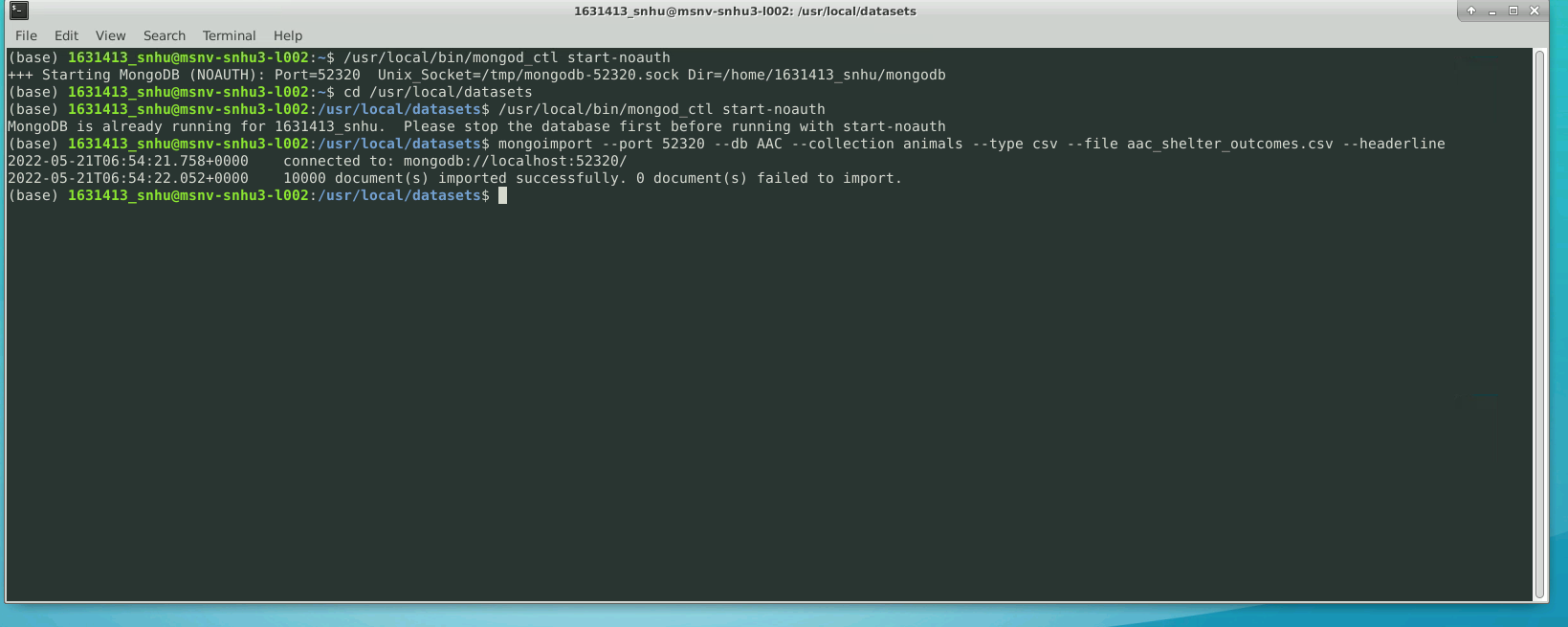
*"name":"walker",*

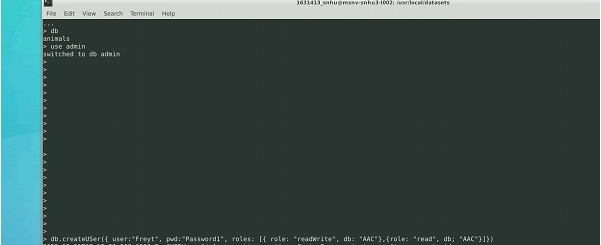
*"type":"dog"*

*}*

*]*

### Screenshots

**

**

**

**Known issues:**

*Dashboard application is seemingly slower than I would like for end user use. This is not a big concern of mine currently as I had a week to really iron out this for the project. Given this was for a paying customer I would spend more time making it run faster.*

## Contact

Your name:Trenton Frey