# CS 340 README

**Project 2**

## About the Project:

*The application allows a user to access a database of animals in the Austin Animal Center(AAC) for the purposes of searching and filtering animals. The user will be able to search for animals using the dash and sort based on criteria from the client Grazioso Salvarethe. The dash also includes an interactable data table, a geolocation map, and a histogram to better display animals best suited for rescue.*

## Motivation

*This program was designed to be a test of my capabilities as a full-stack developer. First, working with MongoDB for the backend, then connecting that with Python functions to the shell in Jupyter Notebook.*

## Getting Started

*To get started using the program you will:*

1. *Import the Austin Animal Center csv file aac\_shelter\_outcomes.csv.*
2. *We must parse the data in the file using MongoDB and add it to the specific table.*
3. *Create both admin and aacuser accounts for managing access to the database.*
4. *The user would need to run the Python file supporting the backend through Jupyter Notebook.*
5. *Once successfully compiled, the user and enter the address of dash program and connect to the app.*

## Installation

*Installed from a web browser:*

* *Spyder for editing python code*
  + *Imported MongoClient from pymongo*
  + *Imported ObjectId from bson.objectid*
* *Jupyter Notebooks for executing python scripts*
* *MongoDB for managing the noSQL database*
* *Latest version of Python for managing functionality and queries(CRUD)*

## Usage

### Code Example

First we import our python file and instantiate it with credentials:

From CRUD import AnimalShelter

test = AnimalShelter(“username”, “password”)

*Create a file:*

test.create({“key” : “value”, “key2” : “value2”) # Should be a dictionary

*Read a file:*

test.read({“key”: “value”} # Should return true

*Update a file:*

test.update({“key” : ”value”}, {“$set” : {"key" : "new\_value"}})

*Delete a file:*

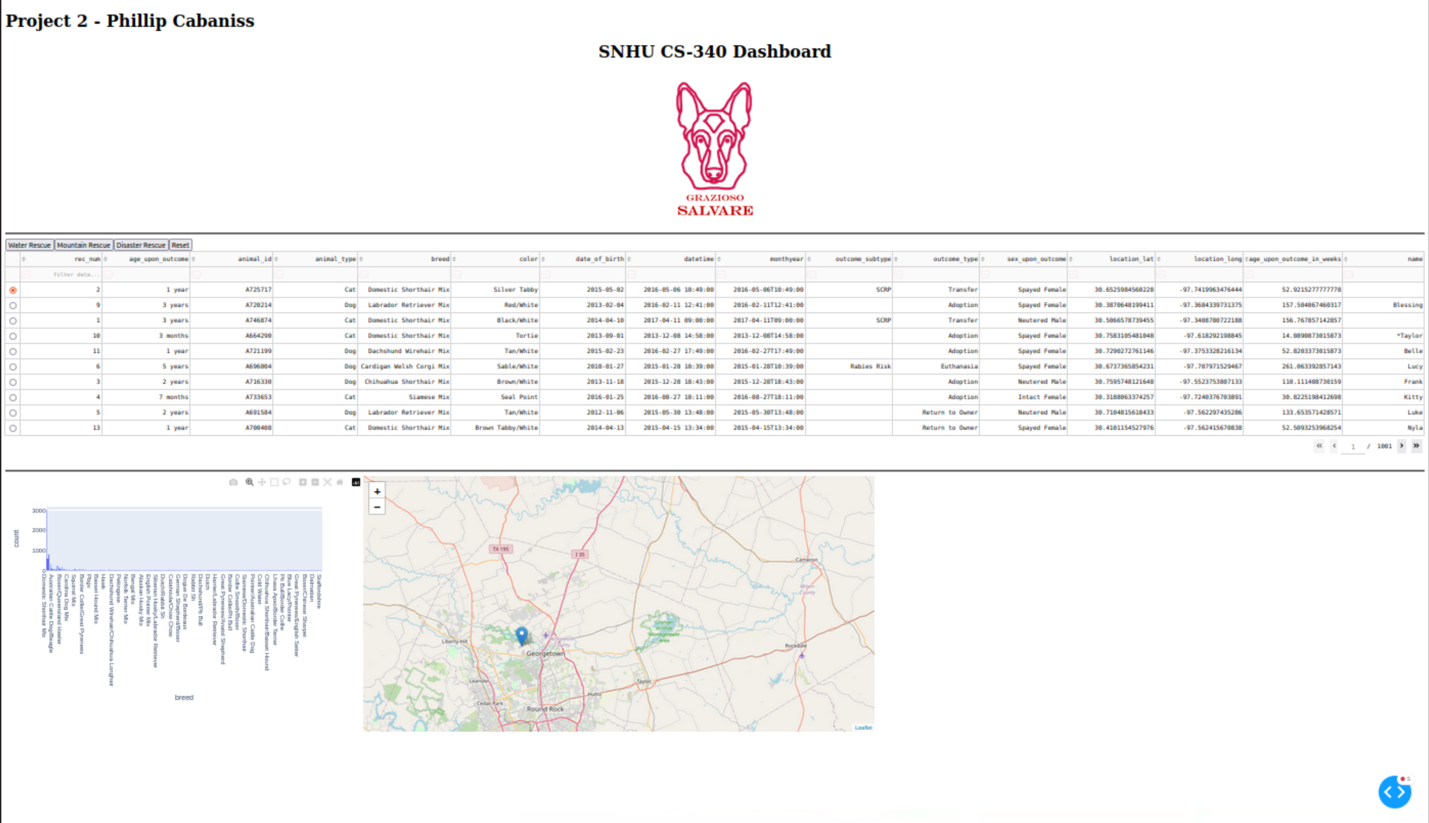
test.delete({“key” : “value”}) # Should return deleted query.

### Tests

*To run tests on the Python module, pay attention to the log. Each function returns a confirmation or result query for verification.*

### Screenshots

*Main Screen:*

**

*Results of pressing Water Rescue button:*

*A screenshot of a computer

Description automatically generated*

*Results of pressing Mountain Rescue button:*

*A screenshot of a computer

Description automatically generated*

*Results of pressing Disaster Rescue button:*

*A screenshot of a computer

Description automatically generated*

*Results of pressing Reset button:*

*A screenshot of a computer

Description automatically generated*

## Issues/Errors:

*The main problems I had were with connecting the jupyter\_dash to the web app successfully. To fix this I found a workaround that is added at the end of my .ipynb file. I also had issues incorporating the chart and implementing my own function for it. After doing research on the dash component library and using the other function as a template, I was eventually successful.*

## Contact

Name : Phillip Cabaniss

Email: Phillip.cabaniss@snhu.edu