# CS 340 README Template

## About the Project/Project Title

*Crud is Creation, Reading, Updating, and Deleting. This repository includes methods for creating and reading (and will soon include updating and deleting), within mongo, using Python to simplify the process. It allows new entries to be added to the AAC database quickly and conveniently.*

## Motivation

*Global rain is a software engineering company dedicated to innovative code. Grazioso Salvare has enlisted Global Rain to help decipher data from animal shelters so as to help look for valid search and rescue dogs. Working with mongo databases is challenging work. The smallest syntax error can be catastrophic, and unlike a lot of other things, the results are often really difficult to undo or mitigate the damage of. The integration of python and methods for reading and writing to databases helps to ensure that the headache of working with mongo, as well as the safety and security against mistakes, is always on your side. With the addition of a dash, it’s easier than ever to view and update information with CRUD.*

## Getting Started

*Open the animal\_shelter.py file, and navigate to the section that includes MONGO\_PASS, MONGO\_USER, and MONGO\_PORT. To ensure that the animal\_shelter.py file points to the correct connections within your mongo terminal, you will need to change the values on this file to match the ones in your own terminal / workspace that you wish to use. Ensure that the capitalization is correct- case matters here. Once you have set up the py file, you can use the CRTest python file or create your own by adding “from animal\_shelter import AnimalShelter”. Once this is added, you can read, write, update, and delete files from your database from python instead of mongo. Also included is the Dashboard ipynb file. You can run this to read the data and display a dashboard of the information. You will need to update the relevant values for this file for your own use as well.*

## Installation

*These applications are designed to be used for mongo databases, and running the scripts through ipynb. Ensure that you are running the most up to date versions of both mongo and Jupyter notebook on your machine. Drag and drop these files into a folder, and run the CRTest script to begin adding or reading entries.*

## Usage

### Code Example, create / read methods (truncated)

*def create(self, data):*

*if data is not None: #create method adds database entries based on dictionary input*

*successfulInsert = self.database.animals.insert\_one(data)*

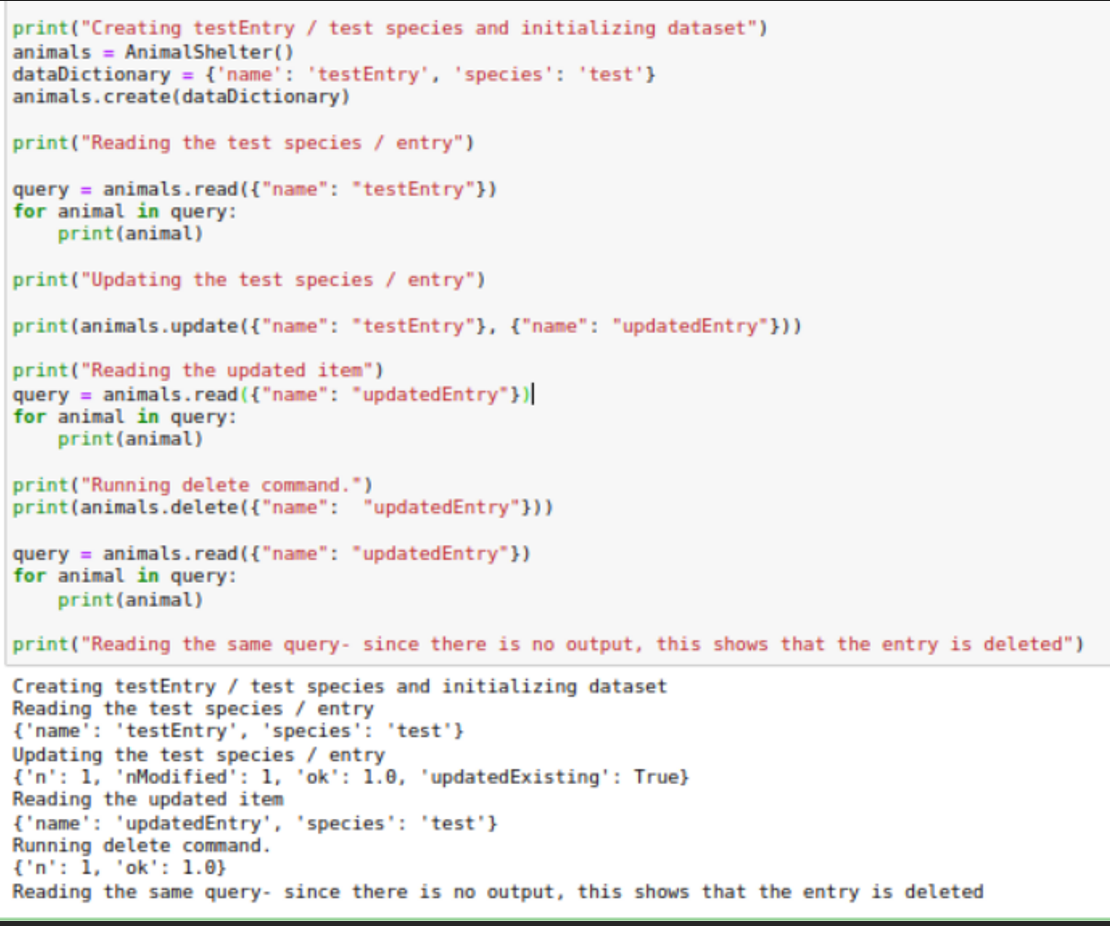
*def read(self, findData): #read by the 'findData' #Read method finds entries within database by key / value pair.*

*if findData:*

*returnData = self.database.animals.find(findData, {"\_id": False})*

### Tests

*Here is an example of the create, read, update, and deletion methods being used.*

**

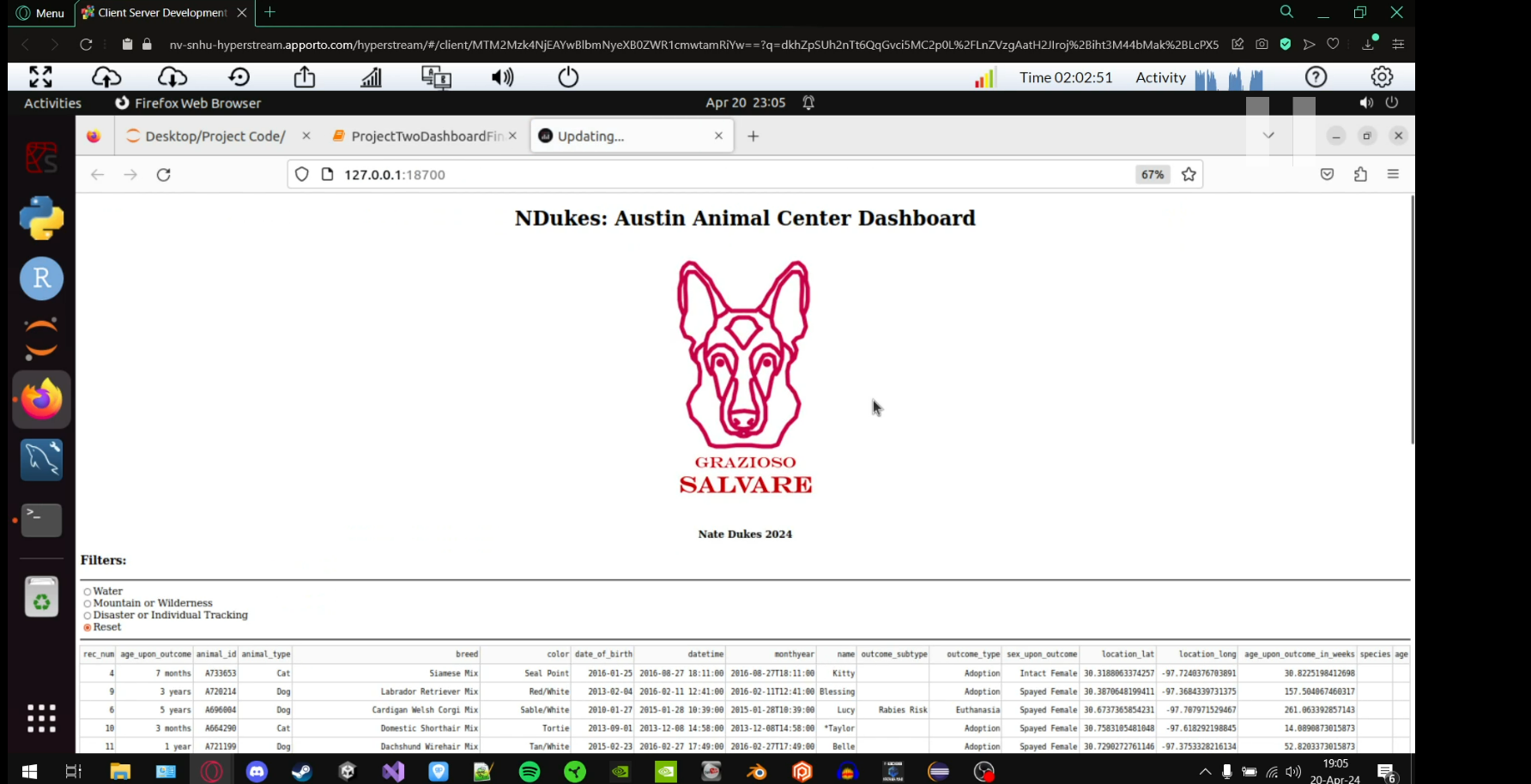
*These methods are used by the dashboard file in order to retrieve and display data onto charts and tables. Below, you can see examples of their use cases on the next page.*

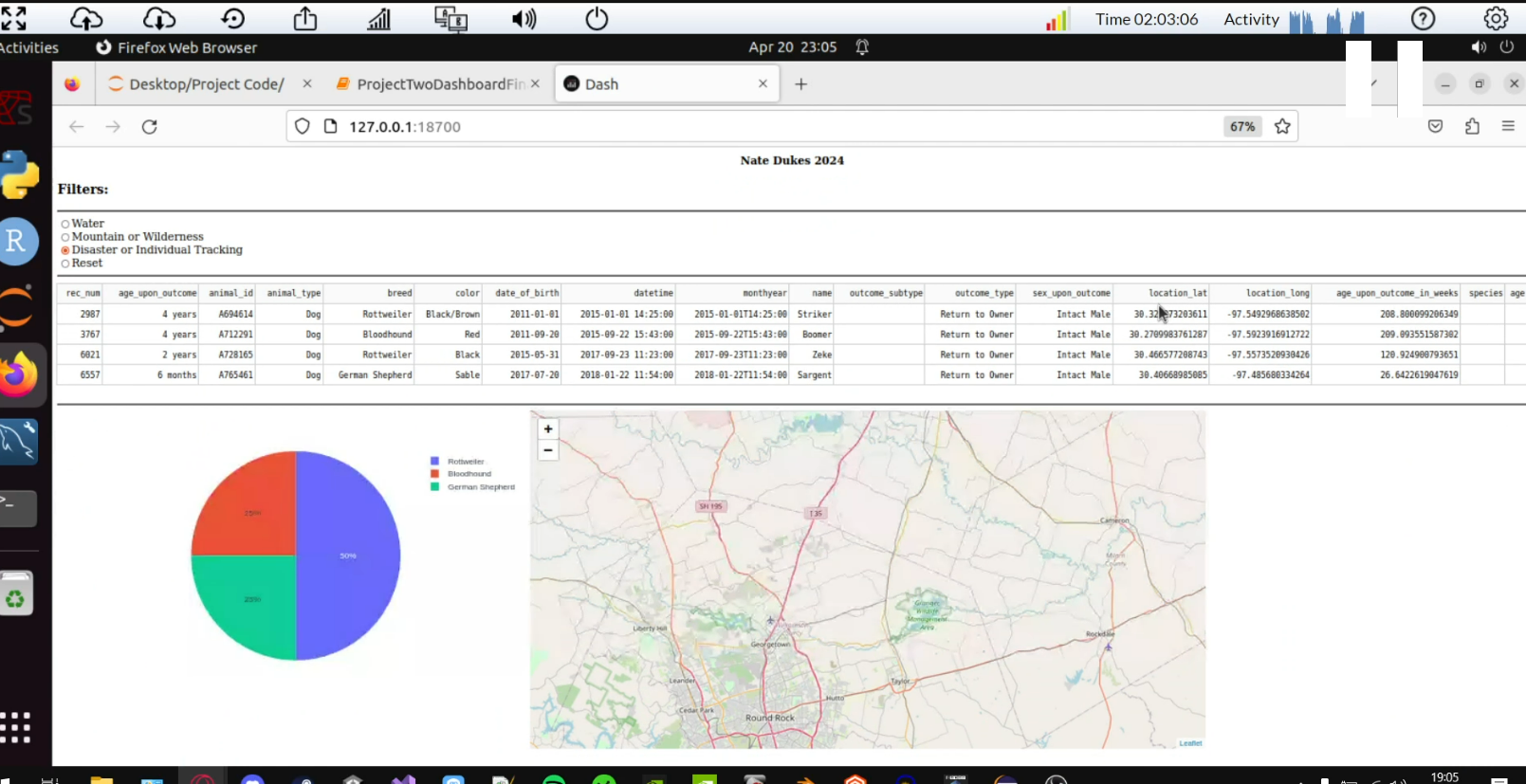
[*https://youtu.be/azleZkOFJcM*](https://youtu.be/azleZkOFJcM)

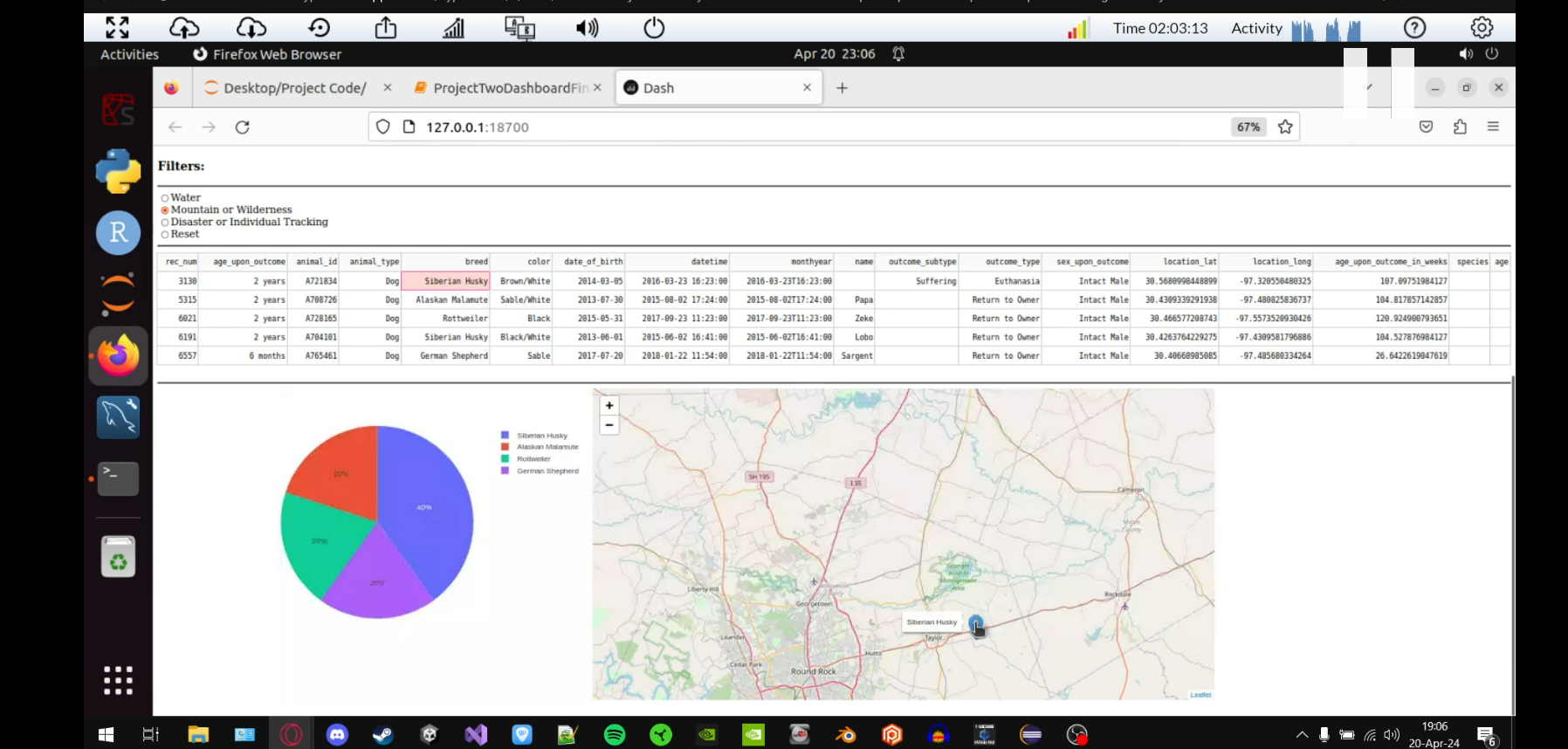
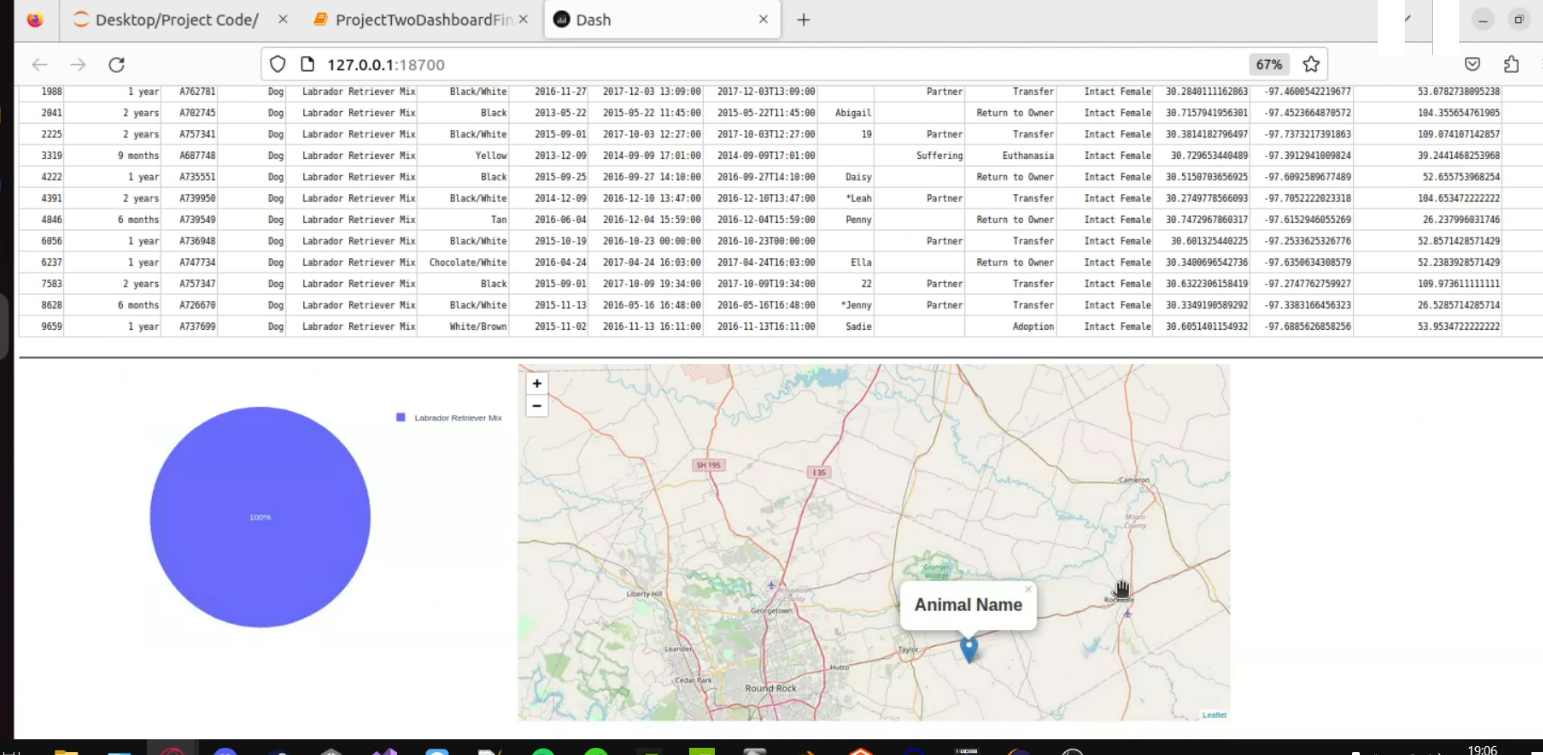
*(Going through a checklist of things that are required for the software)*

*Screenshots taken from the screencast in case the link doesn’t work, demonstrating the filters:*

*Starting state / Reset filters state:*

**

*Filter 1: *

*Filter 2: Filter 3: *

## Roadmap/Features (Optional)

*Features to add:   
Improve dash functionality*

## Contact

Nate Dukes