# CS 340 README Template

## About the Project/Project Title

Austin Animal Center

This project manipulates and uses databases. Using the Mongo Database, CRUD architecture and Python, this project creates a program for the Animal Shelter to edit and review their database with ease.

## Motivation

The project has been created because the Animal Center has a large number of animals that they take care of. They need to be able to access that data fast and they currently are having a hard time doing so. This project solves their issues and allows them not to spend too much time working on using the database and focus on the animals.

## Getting Started

*This is an example of how you may give instructions on setting up your project locally: “To get a local copy up and running, follow these simple example steps.”*

To get the project locally, access the Mongo Database. Using the code below will help access the database.

Gain access by entering:

cd /usr/local/datasets

/usr/local/bin/monod\_ctl start noauth

To bring up the database:

Mongoimport –port (port #) –db (databasename) –collection (collection)./filename

mongo

show dbs

use (databasename)

This will bring up your database and allow you to use it.

**Tools Needed**

Linux OS - Before using any part of the system you will need access to a machine with Linux. This is necessary for using the mongoDB

MongoDB - An API created to use and create databases with ease.

Python - To edit and using the database in other ways, the use of python will be necessary.

Dash framework - the Dash framework gives a lot of functionality for the web app. It provides the coder with a ton of tools that present analytics and is easy to use. It is what made this database visible and easily readable.

Pandas - Another great tool for python that is a library to help do data analysis

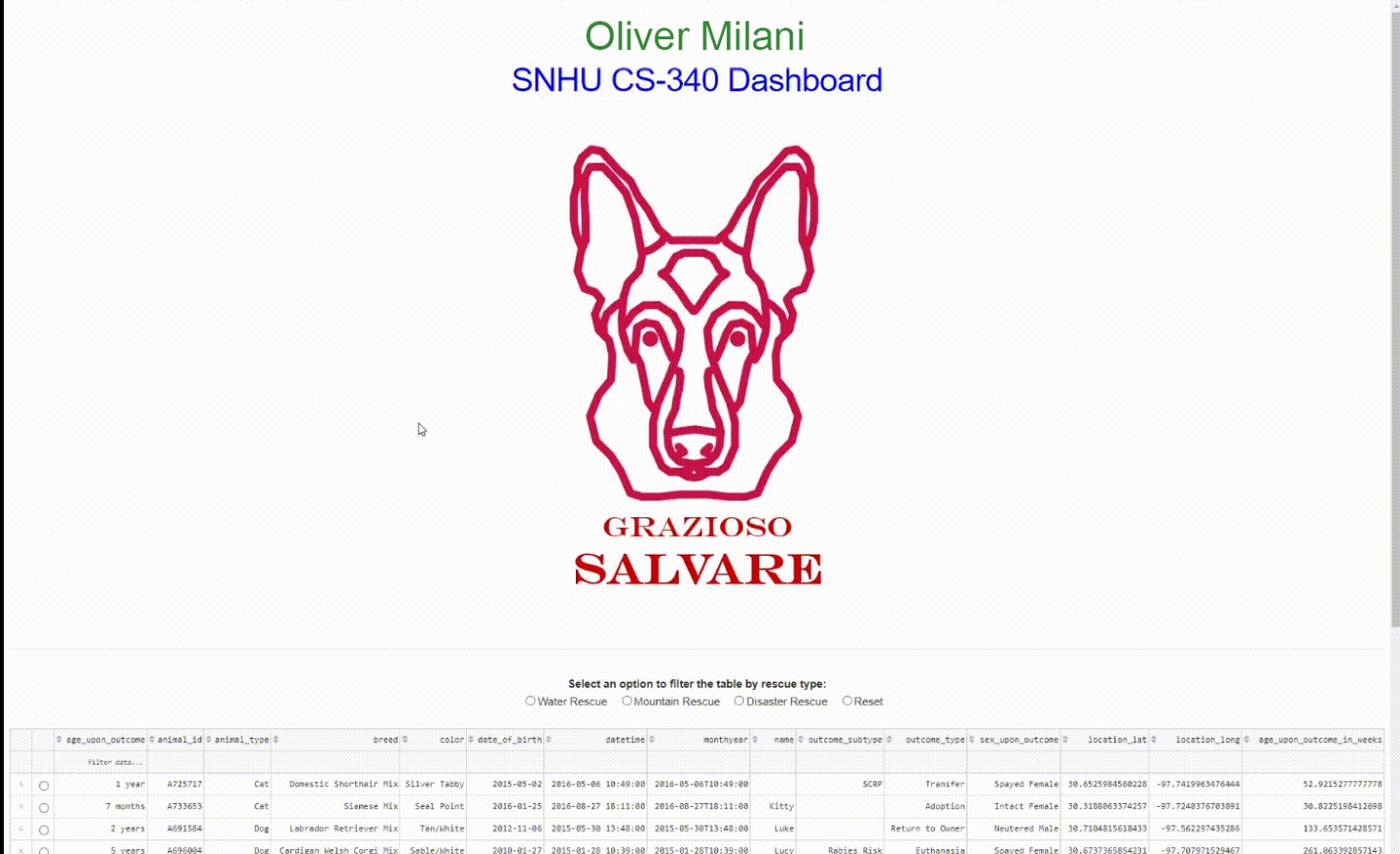
**Steps taken to complete the Project**

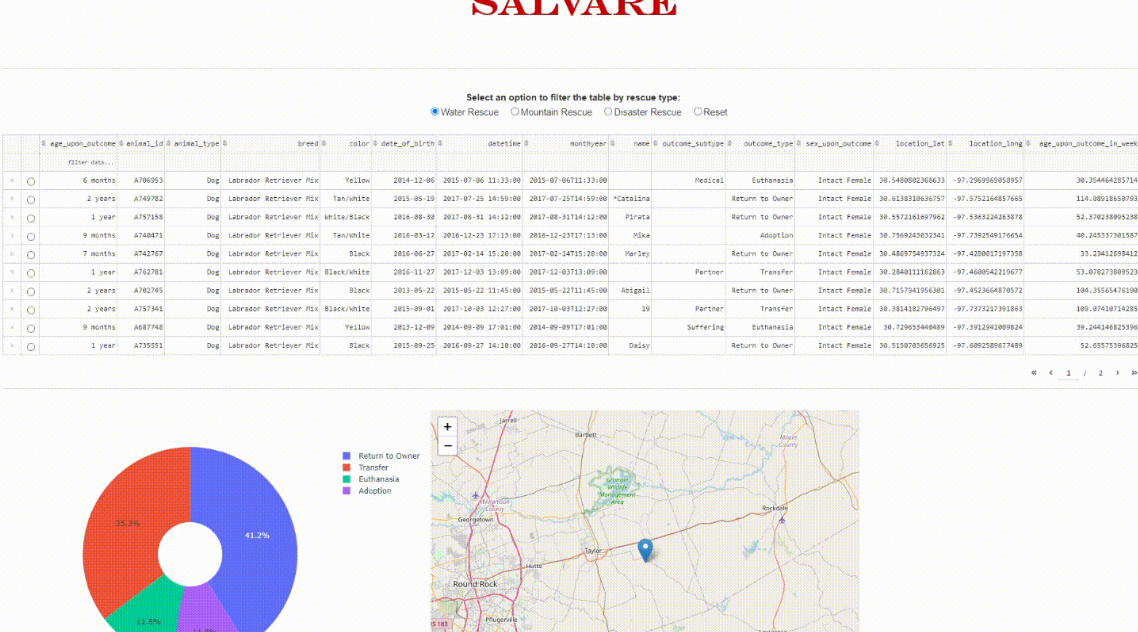
The steps taken to complete the project were first to understand the Mongo Database and how to access it on the shell level. From there, I had to be able to access it using python code by creating a CRUD file. This CRUD file could take the data from the database and create a file with it, read the file, update it if needed, or delete it if needed. We then needed to create an ipynb file that could use that CRUD file. The ipynb file takes the data that the CRUD file receives, and displays that data using the dash framework.

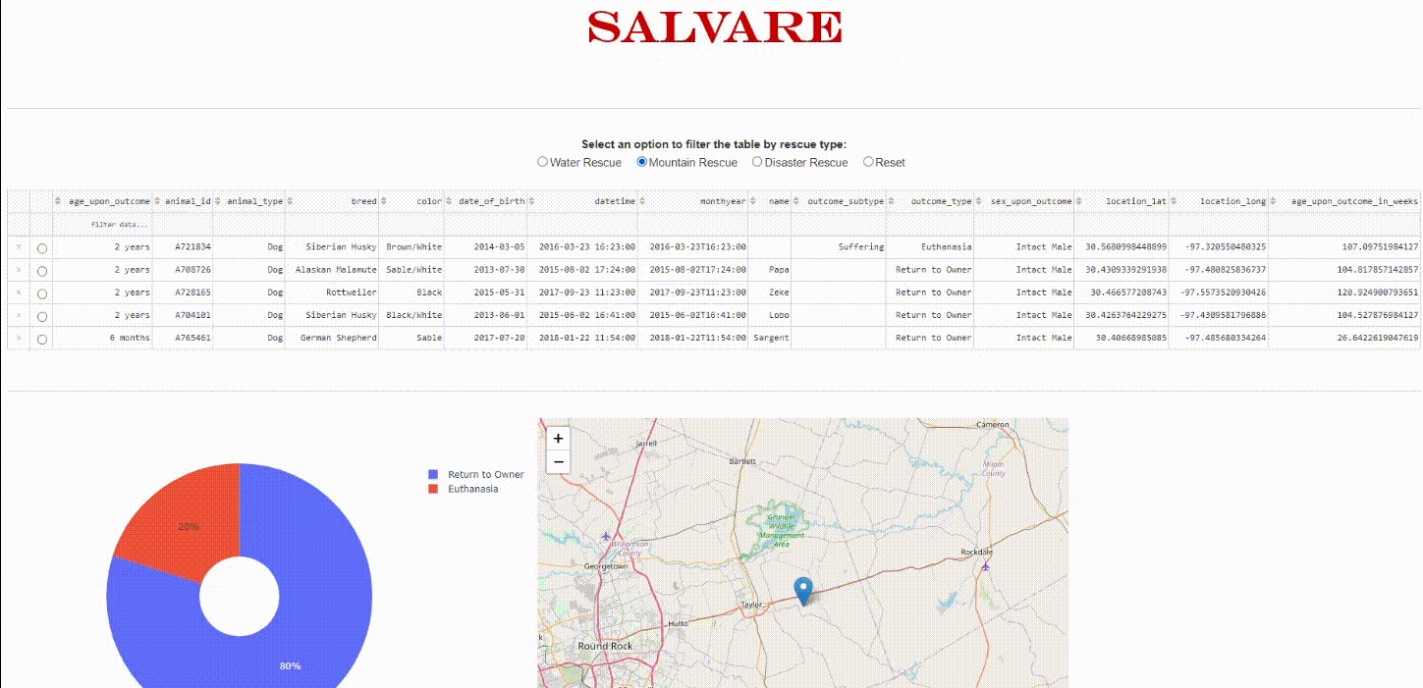
**Challenges**

The majority of the challenges I got was when accessing the database from the shell. I don't know if these issues were because of issues connecting to the server or just because I was going about the process wrong, but eventually it worked. From there, it was easier to approach the python coding when accessing the database. Creating the ipynb file took a bit of time and I think having a lot of the code already done was very helpful.

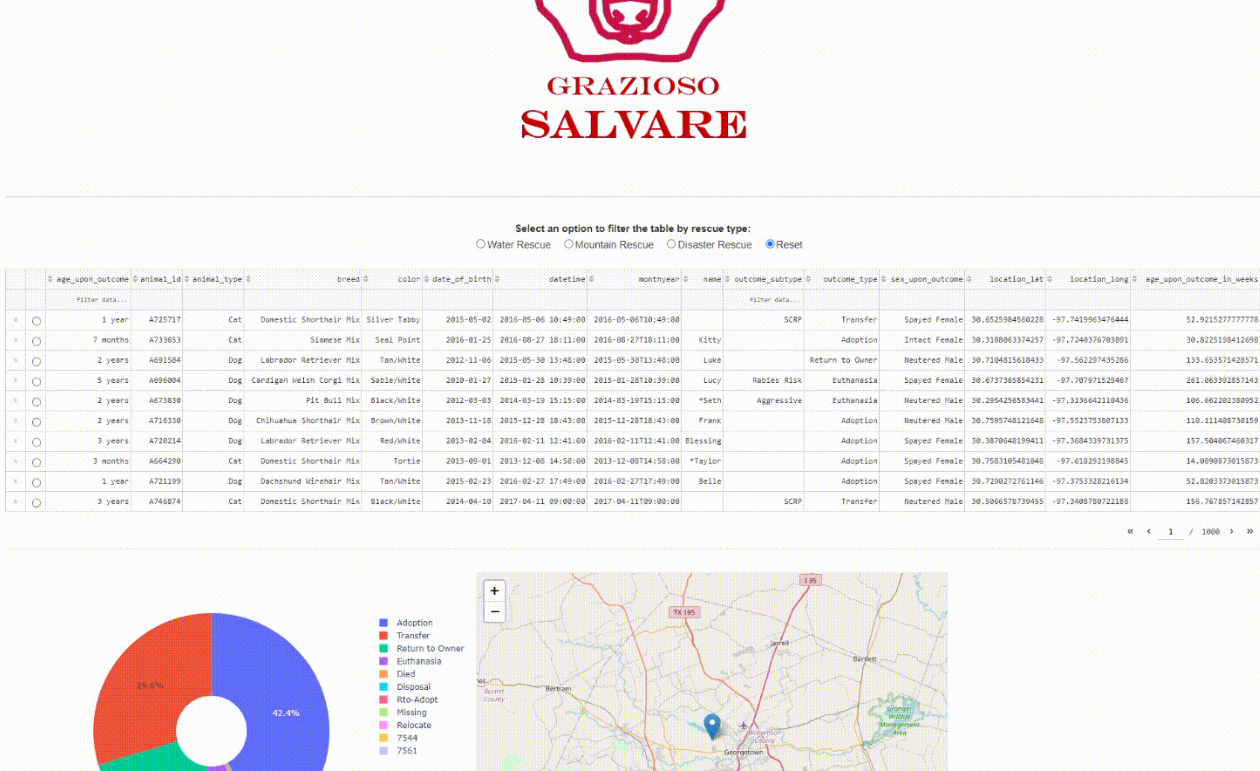
**Screenshots of Starting State of Dashboard, Water Rescue, Mountain Rescue, Disaster Rescue, and Reset below**



**

**

## Contact



Your name: Oliver Milani