# CS 340 README

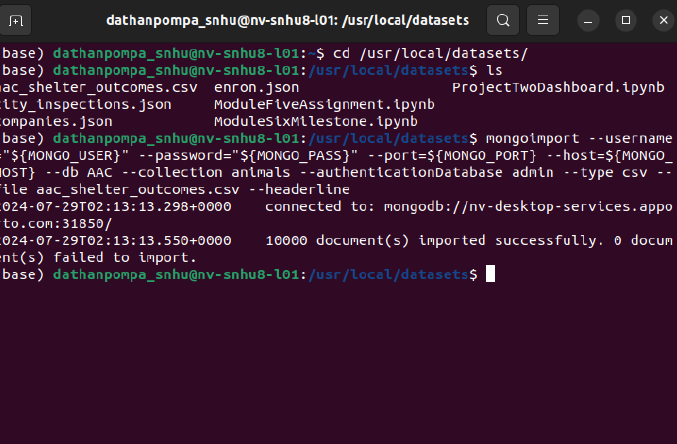
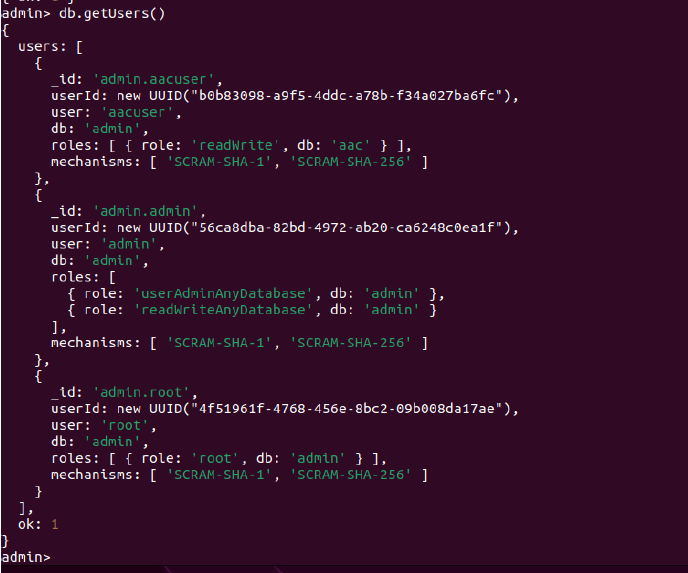
## About the Project/Project Title

*Grazioso Salvare Dashboard displays a database of animals, a pie chart, and a map of its location. This dashboard allows you to filter specific rescue dogs.*

## Motivation

*This dashboard is intended to help others find dogs that can help with certain types of rescues as well as display data on the animal and show their location.*

## Getting Started

1. *To start you must import the csv acc\_shelter\_outcomes.csv into your mongo client.*   
   
2. *Download the provided files for Jupyter and Python scripts.*
3. *Once completed, verify the connection of your database to your mongo client for access to the user, host, and port within the python program.*
4. *If needed, create a user to match the credentials within the python program. Change the User string, Pass string, Host string, and Port numbers to match your mongo client.*   
     
   *-* ***Error I encountered****: When I made the user in my db, I did not have case specific letters in my db. I needed to rename the db to match the db I was trying to access. The db was AAC and I had my aacuser with access to aac. Once I renamed the db to AAC, then it worked as intended.*  
   
5. *Ensure that your Mongo and Python clients are connecting by running the JupyterTest file to see if it can create an animal in the database.*
6. *Run each line to ensure connections are made. If you can Create an animal within the database, the read function works as intended, you are able to update an object, and you can delete an object.*
7. Download the ProjectTwoDashboard.ipynb file and launch the file in Jupyter.
8. Change from CRdatabase import AnimalShelter to match your desired database and object in the database.
9. Change the username, password, host, port, database, and collection to your desired information.
10. Once you run the ProjectTwoDashboard file your Dashboard should display on your local host.  
    ***You have set up the Dashboard program successfully!***

## Installation

*You need to install these* ***tools****:*

*- MongoDB (NoSQL database)*

*- Python + (IDE of your choice)*

*- Jupyter (Used to write and test python script)*

*- Linux (Your choice of OS is fine too)*

*- pymongo (python library that provides tools to work with MongoDB from python)*

*- bson (python library used for working with BSON data)*

- *Dash and Plotly*

*- Pandas*

*- Jupyter Dash*

*- Base 64*

*For installing these tools on Linux*

*1. Installing MongoDB: please refer to their website for instructions as your system may vary.*  
 *-* [*https://www.mongodb.com/docs/manual/installation/*](https://www.mongodb.com/docs/manual/installation/)

*2. Install Python: please refer to python’s setup and usage page.*

*-* [*https://docs.python.org/3/using/index.html*](https://docs.python.org/3/using/index.html)

*3. Install Jupyter: please refer to Jupyters installation guide.*

*-* [*https://jupyter.org/install*](https://jupyter.org/install)

*4. Python libraries: I will be using pip package installer for pymongo and bson.*

*- Pip (*[*https://pip.pypa.io/en/stable/*](https://pip.pypa.io/en/stable/)*)*

*- pymongo: in your command line type “pip install pymongo”*

*- bson: in your command line type “pip install bson”*

*5. Dash and Plotly:* run command “pip install dash plotly dash-leaflet”

*6. Pandas:* run command “pip install pandas”

*7. Jupyter Dash:* run command “pip install jupyter-dash”

## Usage

*Use this space to show useful examples of how your project works and how it can be used. Be sure to include examples of your code, tests, and screenshots.*

### Code Example

# Setup the Jupyter version of Dash

from jupyter\_dash import JupyterDash

# Configure the necessary Python module imports for dashboard components

import dash\_leaflet as dl

from dash import dcc

from dash import html

import plotly.express as px

from dash import dash\_table

from dash.dependencies import Input, Output, State

import base64

# Configure OS routines

import os

# Configure the plotting routines

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

#### FIX ME #####

# change animal\_shelter and AnimalShelter to match your CRUD Python module file name and class name

from CRdatabase import AnimalShelter

These lines of code are going to import the necessary library files that is needed to run the website.

#This sets up the dashboard so that your chart and your geolocation chart are side-by-side

html.Div(className='row',

style={'display' : 'flex'},

children=[

html.Div(

id='graph-id',

className='col s12 m6',

),

html.Div(

id='map-id',

className='col s12 m6',

)

])

])

This is how the map and graph are displayed.

#Radio buttons

html.Div([

dcc.RadioItems(

id='filter-type',

options=[

{'label': 'Water Rescue', 'value': 'Water Rescue'},

{'label': 'Mountain or Wilderness Rescue', 'value': 'Mountain or Wilderness Rescue'},

{'label': 'Disaster Rescue or Individual Tracking', 'value': 'Disaster Rescue or Individual Tracking'},

{'label': 'Reset', 'value': 'Reset'}

],

value='Reset')

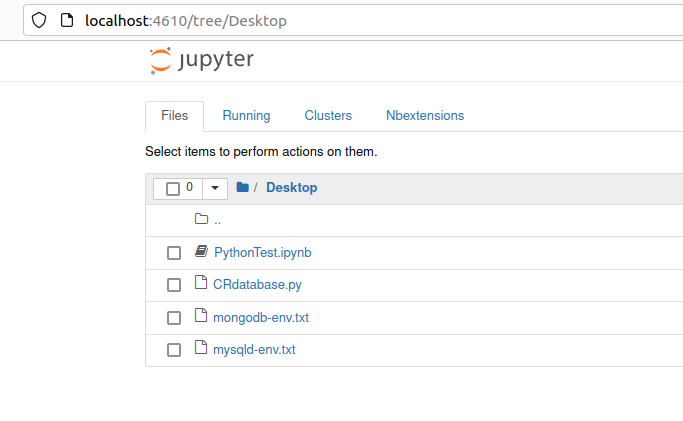
]),

Above is the code for creating the radio buttons on the website.

### Tests

*Describe and show how to run the tests with code examples.*

1. *To test, launch the Jupyter program and verify that the test script and python script are in the same folder.*



1. *Select the .ipynb file*
2. *From here, select each cell and click on the run button above in order from cell 1 to cell 4.*
3. *When selecting the cell it will turn green, when you select run, if it works then it hover over the next cell with a blue color, select the next cell and click run again.*
4. *If there is an error, then it will display an error message below the cell.*

*Within my update\_dashboard function*   
*df = pd.DataFrame.from\_records(db.read(query))*

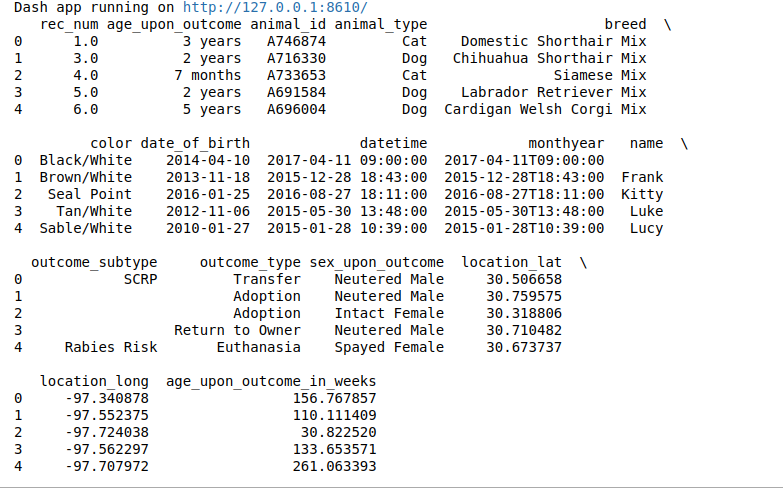
*df.drop(columns=['\_id'],inplace=True)*

*print(df.head())*

*data=df.to\_dict('records')*

*return data*

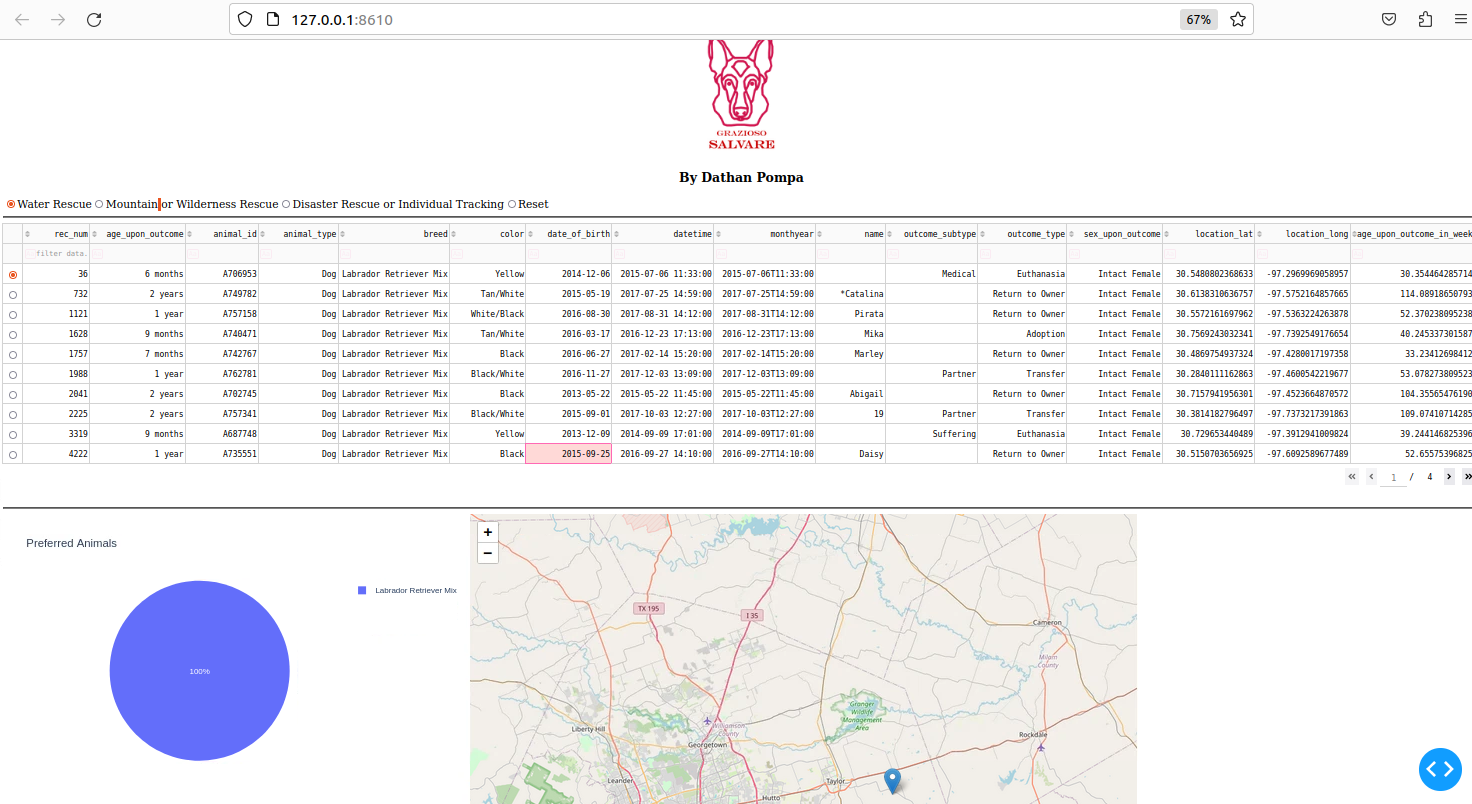
*I have a print df.head code that will display what is being read from my query onto Jupyter dashboard itself incase there are errors on the website.*

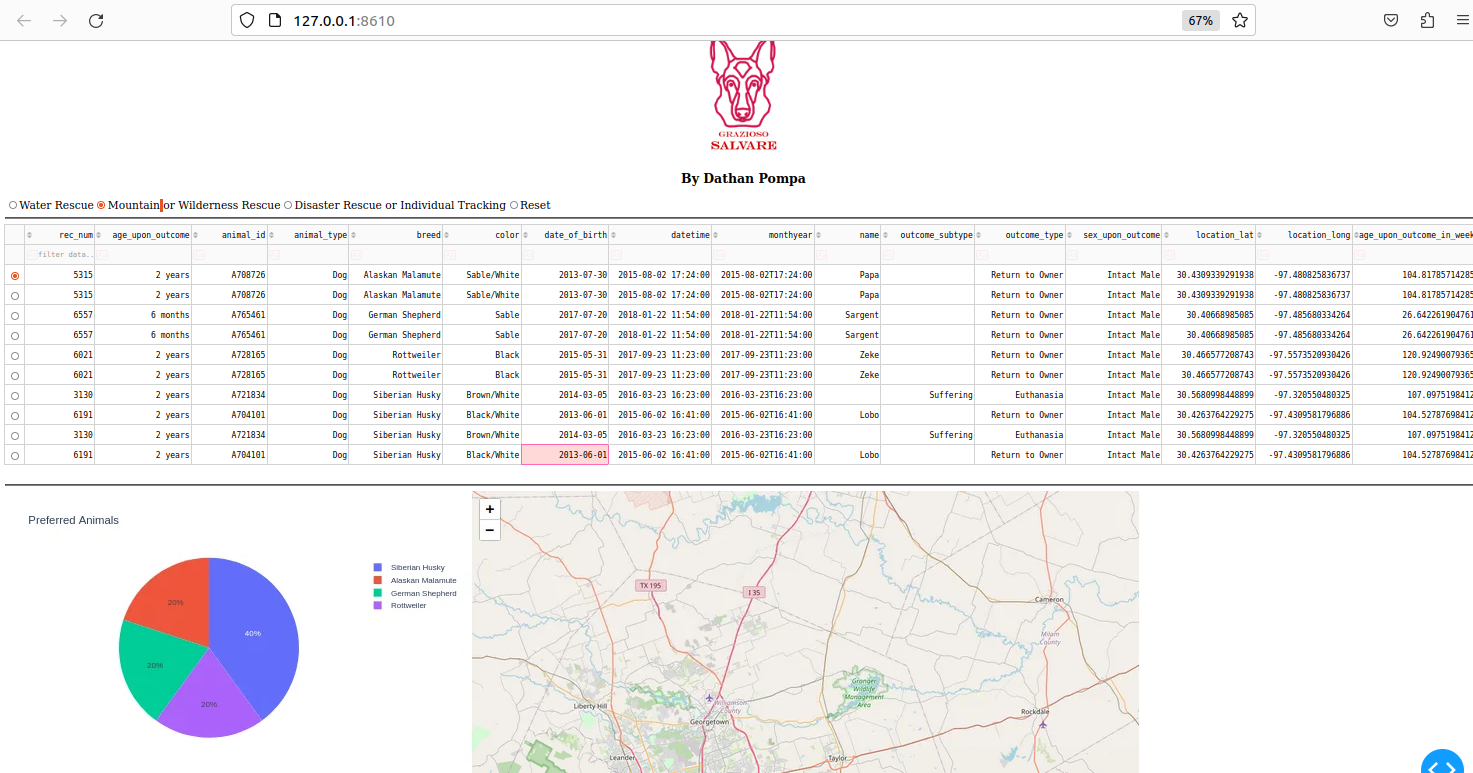


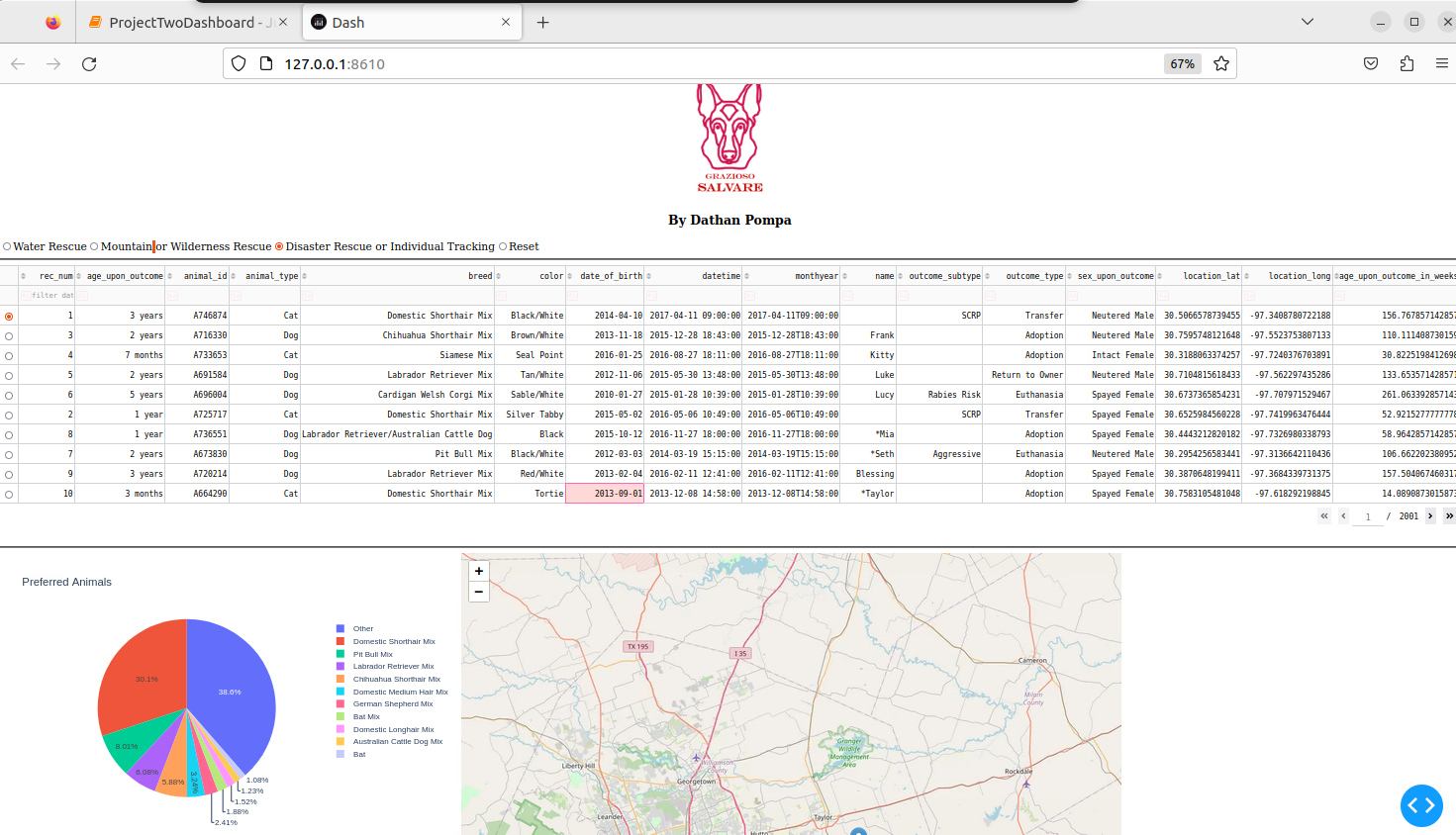
These are the results of the test.

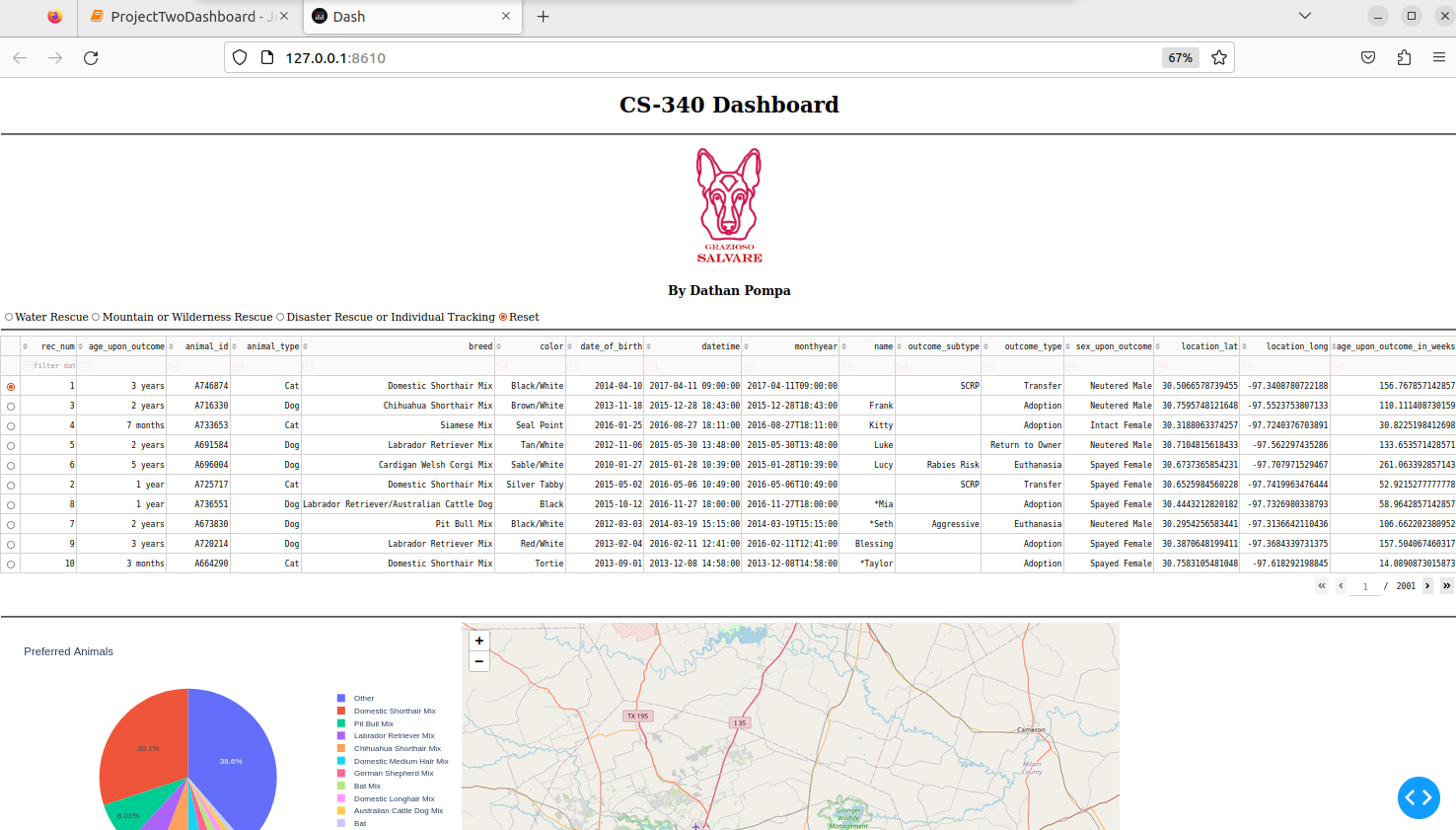
### Screenshots

*Provide screenshots that demonstrate your work.*









## Contact

Your name: Dathan Pompa