**Merrik Wright**

**CS340**

**README File**

**About the Project/Project Title – Project 2**

This application provides users with access to the **CS 340 Beaver Utah, (Really Austin Texas) Animal Center (AAC)** database, enabling efficient searching and filtering of animal data. Designed specifically for Grazioso Salvare—the organization requesting this project—the dashboard helps users identify animals that meet search-and-rescue training criteria. Key features include interactive filters, geolocation mapping, and dynamic visual charts to help locate and utilize animals for rescue operations.

Motivation

The purpose of this project was to demonstrate my skills in working with databases and manipulating data effectively. I selected **Python** as the primary programming language because of its flexibility, ease of use, and strong compatibility with MongoDB. Python's built-in support for Jupyter Notebook also allows for quick testing, development, and compilation.

**Getting Started**

Please follow the instructions below to start.

* Set Up MongoDB:
  + Start MongoDB and load the provided CSV file aac\_shelter\_outcomes.csv into the database.
* Create Indexes:
  + Create both basic and complex indexes to efficiently parse and query the data.
* Set Up User Authentication:
  + Create two accounts: an Admin account for administrative tasks and an aacuser account to interact with the database.
* Install Python and Jupyter Notebook:
  + Ensure Python is installed on your system and set up Jupyter Notebook for running the .py and .ipynb files.
* Run the Dashboard:
  + Start the program and navigate to the Dash web address provided upon execution to access the dashboard.

Installation

**The required files needed to do this project:**

* **Python 3.10 or higher**: The programming language used for this module.
* **PyMongo**: A Python library for interacting with MongoDB.
  + Install PyMongo using pip: pip install pymongo
* **MongoDB**: The database used for storing and retrieving data.
* Required Libraries:
  + PyMongo: For connecting and interacting with MongoDB.
  + Dash: For building the interactive dashboard.
  + Plotly: For dynamic data visualizations.
  + Pandas: For data manipulation and analysis.
  + Leaflet: For geolocation mapping.

**Usage**

**Code Example. Please do not follow the code exactly, as your own touch may avoid errors for your specific examples. The examples are of the layout. This is to create an easy to navigate area.**

app.layout = html.Div([

#create an anchor for the image/logo

#make the image an href to the website, www.snhu.edu

#open the link in a new tab by setting a blank target

html.A([

html.Center(html.Img(src='data:image/png;base64,{}'.format(encoded\_image.decode()),

height = 250, width = 251))], href = '<https://www.snhu.edu>', target = "\_blank"),

html.Center(html.B(html.H1('Merrik Wright CS340 Project2))),

html.Hr(),

#create the radio buttons to act as a filter

#set the default on initial load to to 'All'

dcc.RadioItems(

id='filter-type',

options=[

{'label': 'All', 'value': 'All'},

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

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

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

],

value='All'

),

html.Hr(),

dt.DataTable(

id='datatable-id',

columns=[

{"name": i, "id": i, "deletable": False, "selectable": True} for i in df.columns

],

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

editable=True,

row\_selectable="single", #allow a row to be selected

selected\_rows=[],

filter\_action="native", #allow a filter

sort\_action="native", #allow sorting

page\_action="native", #enable pagination

page\_current=0, #set start page

page\_size=10, #set rows per page

),

html.Br(),

html.Hr(),

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

html.Div(className='row',

style={'display' : 'flex', 'justify-content':'center'},

children=[

html.Div(

id='graph-id',

className='col s12 m6',

),

html.Div(

id='map-id',

className='col s12 m6',

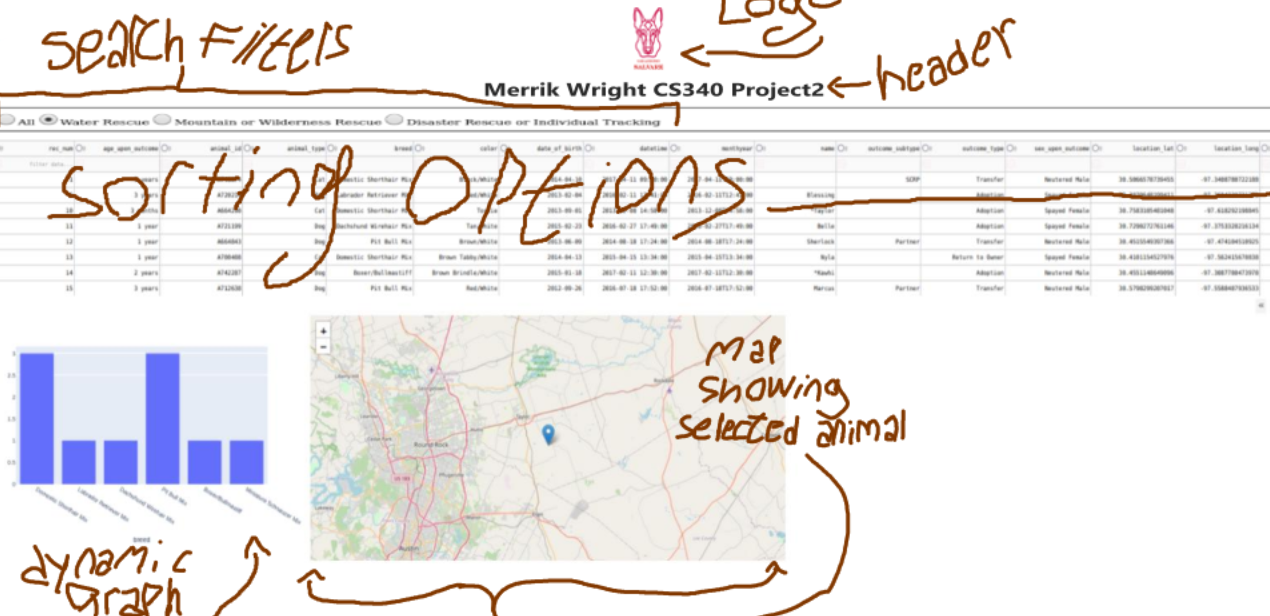
)

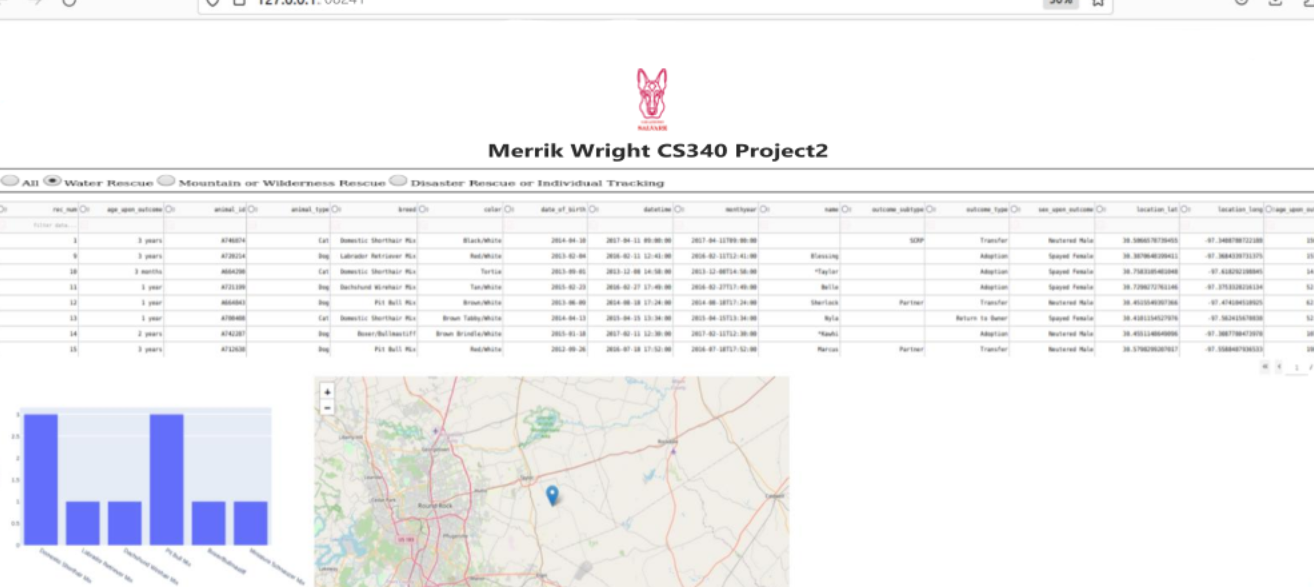
])

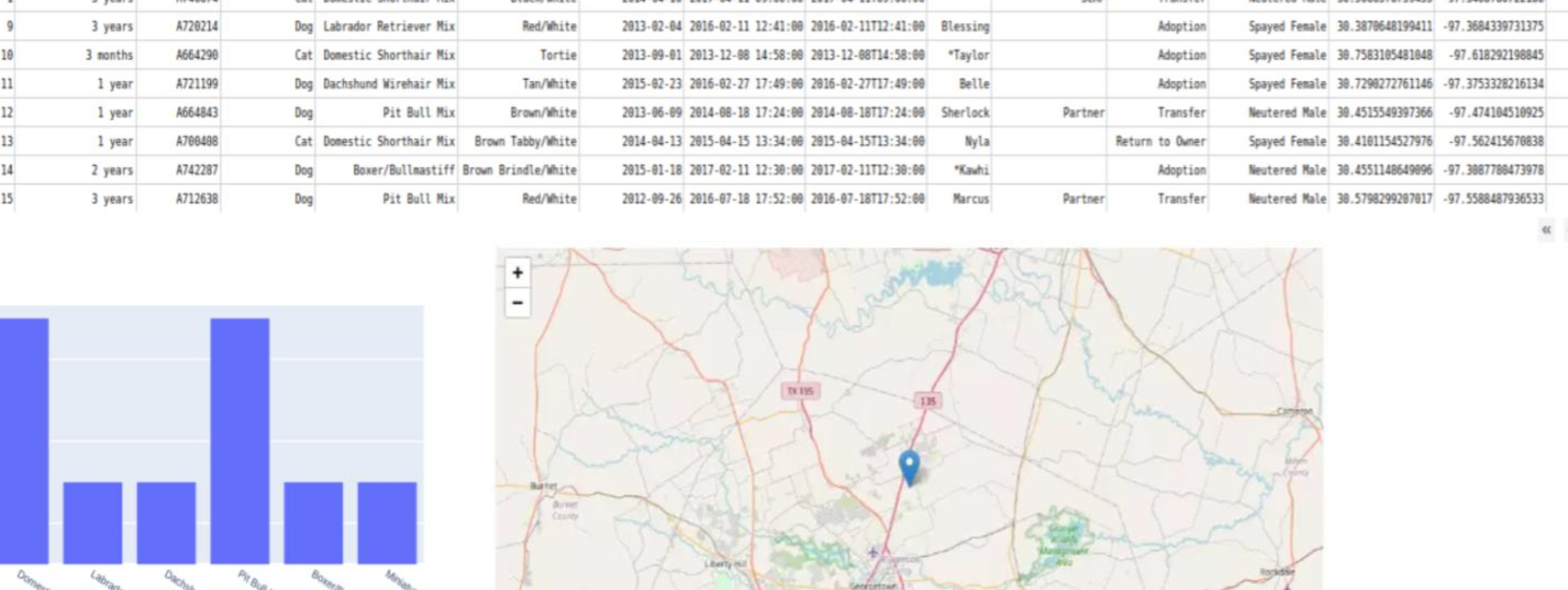
])

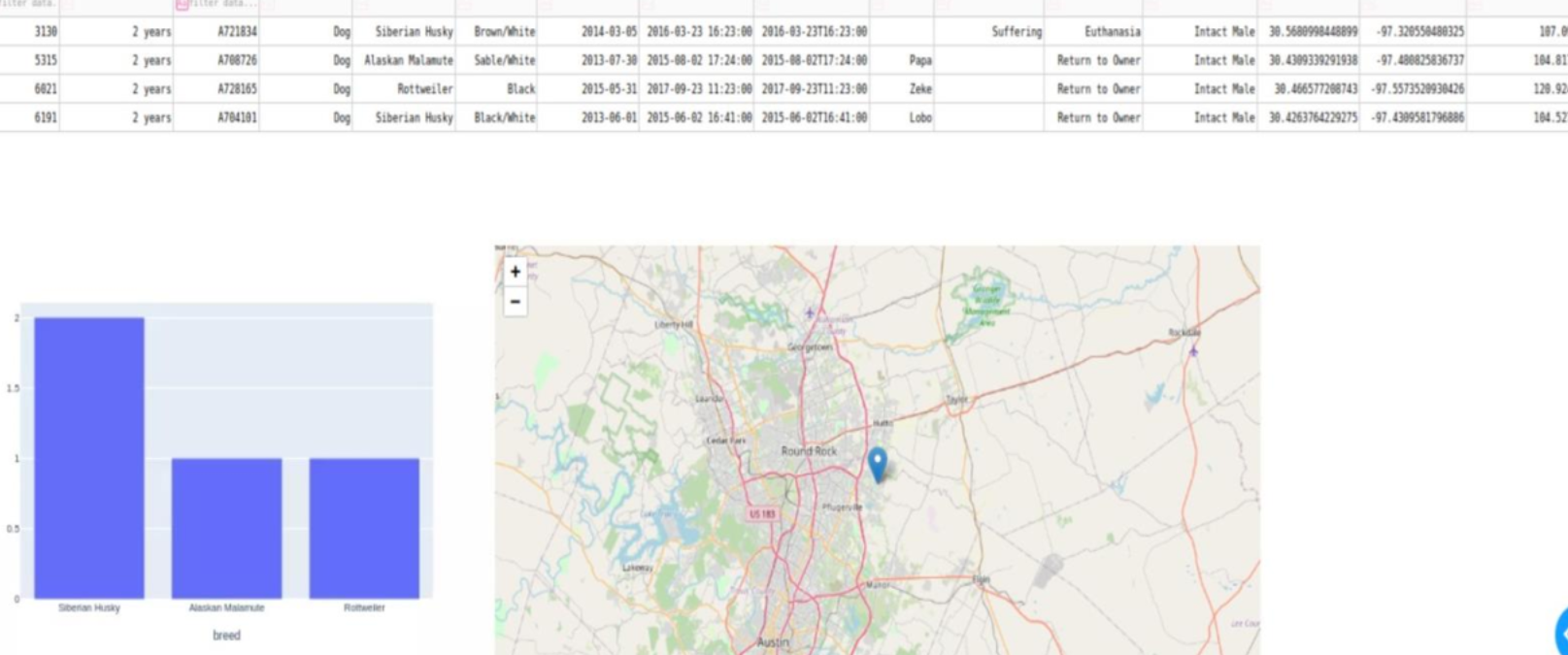
Tests

**Screenshots**









**Contact**

Merrik Wright