# CS 340 README Template

## About the Project/Project Title

SNHU CS-340 Dashboard is designed to manage and display animal data from a MongoDB database. This dashboard allows users to filter data, visualize it on a geolocation map and in a pie chart. and view details about individual animals in the database

## Motivation

The motivation behind this project was to create a data management system for animal rescue operations. The project aims to assist rescue organizations in tracking and analyzing animal data effectively and making informed decisions. By integrating data visualization with real-time database interaction, the project provides a powerful tool for rescue teams to optimize their operations.

## Getting Started

First, to get this project up and running locally, I will provide a repository on GitHub where it will be stored and cloned to use whenever and wherever. ( Will provide a link later)

After that ensure that MongoDB is installed correctly on your system that would be using it and Python3.8+ is installed, Python package manager to install required libraries, and Dash and Plotly.

## Installation

MongoDB – Download and Install https://www.mongodb.com/try/download/community

Python 3.8 or latest - Follow this site to install Python to MongoDB to utilize PyMongo https://pypi.org/project/pymongo/3.8.0/

Python IDE (optional) – I used Pycharm to create the coding script, it’s not necessary to have but if you’d like to have an IDE installed then follow this link. https://www.jetbrains.com/pycharm/

## 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

*Show what the library does as concisely as possible. Developers should be able to figure out how your project solves their problem by looking at the code example. Make sure that your code is short and concise.*

# Import the AnimalShelter class from your CRUD module

from AnimalShelter import AnimalShelter

###########################

# Data Manipulation / Model

###########################

# Connection Variables

username = "aacuser"

password = "SNHU1234"

host = 'nv-desktop-services.apporto.com'

port = 32450

DB = 'AAC'

COL = 'animals'

# Instantiate AnimalShelter

shelter = AnimalShelter(user=username, password=password, host=host, port=port)

# Fetch initial data from the database (all records)

df = pd.DataFrame.from\_records(shelter.read({}))

# Drop the '\_id' column if it exists to prevent issues with the DataTable

if '\_id' in df.columns:

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

#########################

# Dashboard Layout / View

#########################

app = JupyterDash(\_\_name\_\_)

image\_filename = '/home/rikkixaysanas\_snhu/Desktop/AnimalShelter/Grazioso Salvare Logo.png' # Image FilePath

# Encode the image

encoded\_image = base64.b64encode(open(image\_filename, 'rb').read()).decode('ascii')

app.layout = html.Div([

# Hidden div for potential future use

html.Div(id='hidden-div', style={'display': 'none'}),

# Displays the logo image with a URL anchor

html.A(

href="https://www.snhu.edu", # SNHU webpage

children=html.Img(

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

style={'width': '200px', 'display': 'block', 'margin-left': 'auto', 'margin-right': 'auto'}

)

),

# My Unique identifier with navy blue color and cursive font

html.Div(

"Created by: Rikki Xaysanasy",

style={

'textAlign': 'center',

'padding': '10px',

'fontSize': '18px',

'fontFamily': 'cursive',

'color': 'navy'

}

),

# Main Title

html.Center(html.B(html.H1('SNHU CS-340 Dashboard'))),

html.Hr(),

# Buttons for filtering data

html.Div(className='buttonRow',

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

children=[

html.Button(id='submit-button-all', n\_clicks=0, children='Reset'),

html.Button(id='submit-button-one', n\_clicks=0, children='Water Rescue'),

html.Button(id='submit-button-two', n\_clicks=0, children='Mountain or Wilderness Rescue'),

html.Button(id='submit-button-three', n\_clicks=0, children='Disaster Rescue or Individual Tracking'),

]),

# DataTable to display the data

dash\_table.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,

filter\_action="native",

sort\_action="native",

sort\_mode='multi',

row\_selectable='multi',

row\_deletable=True,

selected\_rows=[],

page\_action='native',

page\_current= 0,

page\_size= 10,

),

html.Br(),

html.Hr(),

# This div contains both the map and the pie chart side by side

html.Div(className='row', style={'display': 'flex'}, children=[

html.Div(id='map-id', className='col s12 m6', style={'width': '50%'}),

html.Div(id='graph-id', className='col s12 m6', style={'width': '50%'})

]),

html.Hr(),

])

### Tests

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

## Debug

# print(len(df.to\_dict(orient='records')))

# print(df.columns)

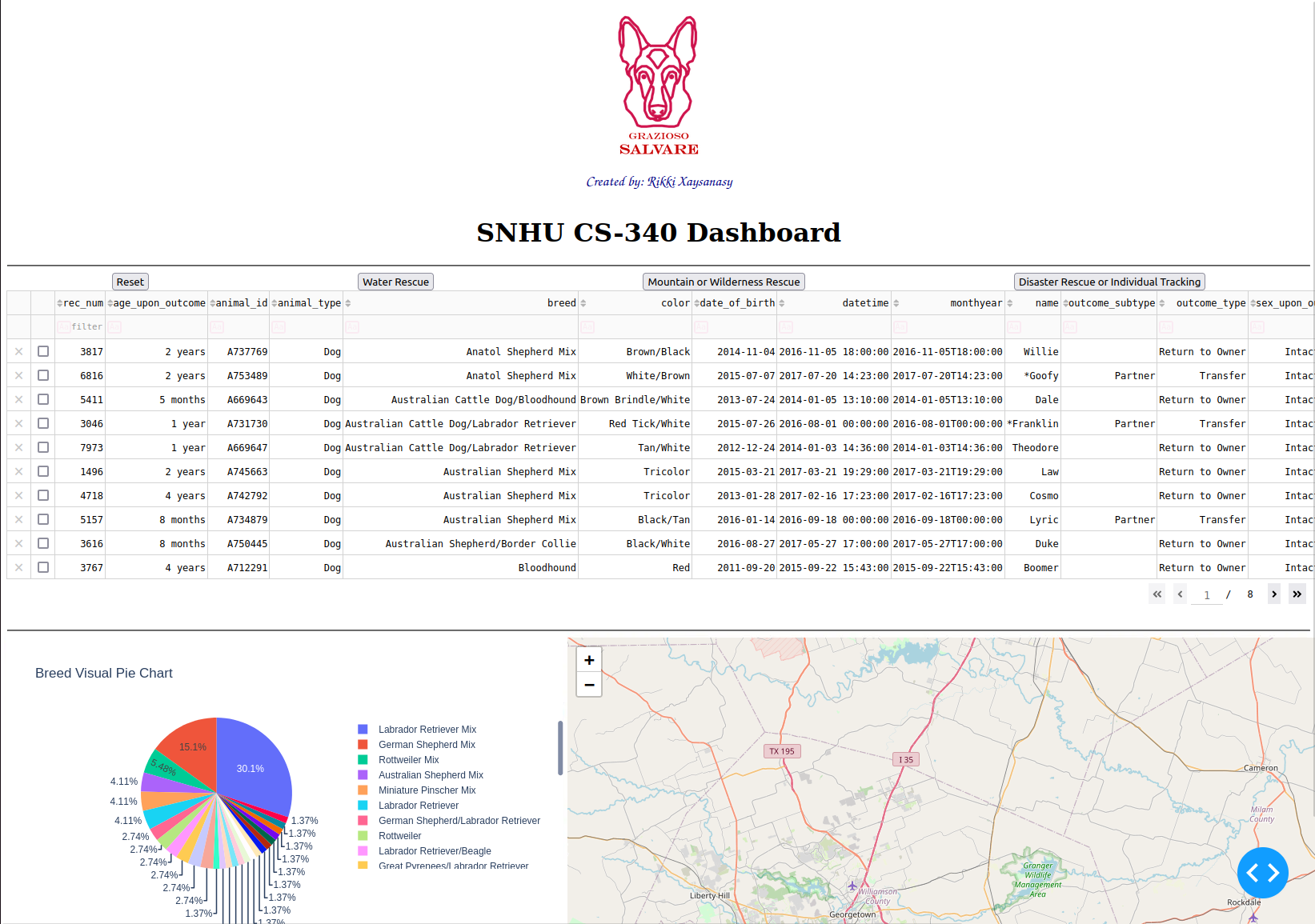
### Screenshots

*Provide screenshots that demonstrate your work.*

* Reset Data Button Pressed
  + A screenshot of a computer

    Description automatically generated
* Water Rescue Button Pressed
  + A screenshot of a computer

    Description automatically generated
* Mountain or Wilderness Rescue Button Pressed
  + A screenshot of a computer

    Description automatically generated
* Disaster Rescue or Individual Tracking Button Pressed
  + 

## Contact

Rikki.Xaysanasy@snhu.edu