# CS 340 README Project Two

## About the Project/Project Title

The CS-340 Dashboard project is a web application designed to provide insights into animal rescue operations at a shelter. It interfaces with a MongoDB database to retrieve and display information about various rescue types, allowing users to filter data and make informed decisions.

## Motivation

The motivation behind this project is to enhance the operational efficiency of animal shelters by providing a user-friendly interface for accessing critical rescue data. By using this dashboard, shelter staff can quickly analyze and respond to the needs of animals in various rescue situations.

## Getting Started

To get a local copy up and running, follow these simple steps:

1. Clone the repository to your local machine.
2. Ensure you have Python and MongoDB installed.
3. Install the required Python libraries.
4. Configure your MongoDB connection settings in the code.
5. Run the application to access the dashboard.

## Installation

The tools used:

* **MongoDB:** The database used to store and retrieve data.
* **Python:** The programming language used for this project.
* **Jupyter Notebook:** Used for development and testing in an interactive format.

## 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

app.layout = html.Div([

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

html.Center(html.B(html.H1('Lauren-Ann Javier: CS-340 Dashboard'))),

html.Center(html.Img(src=f'data:image/png;base64,{encoded\_image}', style={'width': '50%'})),

html.Hr(),

html.Div([

html.Label("Rescue Type Filter:"),

dcc.RadioItems(

id='filter-type',

options=[

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

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

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

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

],

value='All',

inline=True

)

], style={'padding': '10px'}),

html.Hr(),

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'),

page\_size=10,

row\_selectable='single',

sort\_action="native",

style\_table={'overflowX': 'auto'},

style\_data\_conditional=[{

'if': {'column\_id': 'column\_name'},

'background\_color': '#D2F3FF'

}]

),

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'}, children=[

html.Div(id='graph-id', className='col s12 m6'),

html.Div(id='map-id', className='col s12 m6')

])

])

### Tests

import unittest

from animalShelter import AnimalShelter

class TestAnimalShelter(unittest.TestCase):

def setUp(self):

username = "aacuser"

password = "SNHU1234"

self.db = AnimalShelter(username, password)

def test\_create\_animal(self):

animal\_data = {'name': 'Buddy', 'breed': 'Golden Retriever', 'rescueType': 'Water Rescue'}

result = self.db.create(animal\_data)

self.assertTrue(result) # Check if the animal was created successfully

def test\_read\_animal(self):

animal = self.db.read({'name': 'Buddy'})

self.assertEqual(animal[0]['breed'], 'Golden Retriever') # Verify the breed

def test\_update\_animal(self):

self.db.update({'name': 'Buddy'}, {'$set': {'breed': 'Labrador'}})

animal = self.db.read({'name': 'Buddy'})

self.assertEqual(animal[0]['breed'], 'Labrador') # Verify the breed was updated

def test\_delete\_animal(self):

self.db.delete({'name': 'Buddy'})

animal = self.db.read({'name': 'Buddy'})

self.assertEqual(len(animal), 0) # Ensure the animal was deleted

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()

### Screenshots

A screenshot of a computer

Description automatically generatedA computer screen shot of a computer screen

Description automatically generatedA screenshot of a computer

Description automatically generatedA computer screen shot of a computer screen

Description automatically generatedA computer screen shot of a program

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

## Contact

Your name: Lauren-Ann Javier