# CS 340 PROJECT TWO README

## PROJECT TWO: ANIMAL SHELTER DASHBOARD

The Animal Shelter Dashboard project aims to create an interactive and user-friendly web application for managing and visualizing data related to an animal shelter. This project is designed to assist shelter administrators and staff in efficiently tracking and managing the shelter's operations, including information about the animals, their outcomes, and their geographic locations. The dashboard provides a comprehensive view of the shelter's data, allowing users to filter and explore the information based on various criteria such as animal type, breed, age, outcome type, and more. It also includes a geospatial component that displays the locations of the animals on an interactive map, enabling users to visualize the distribution of animals and their outcomes across different areas. The application's goal is to streamline shelter management, facilitate data-driven decision-making, and enhance the overall experience of caring for and finding homes for animals. Through this project, we aim to contribute to the welfare of animals and improve the efficiency of animal shelter operations.

## MOTIVATION

The motivation behind the creation of the Animal Shelter Dashboard project is driven by the need to address the challenges and complexities faced by animal shelters in managing their operations and data effectively. Animal shelters play a crucial role in the welfare of animals, and the efficient management of shelter resources, animal records, and outcomes is paramount. Many shelters still rely on manual or outdated systems for record-keeping, which can be time-consuming and error prone. By developing a modern, user-friendly dashboard, we aim to simplify the process of managing shelter data, improve transparency, and enable shelters to make informed decisions regarding animal care and adoption strategies. Ultimately, our motivation is to enhance the lives of animals in shelters and increase the chances of finding them loving homes by providing shelter staff with a powerful tool for data analysis and visualization.

## GETTING STARTED:

* Create a Mongo Database and name it "AAC."
* Create a user with read/write privileges for the AAC database.
* Import the data from the 'aac\_shelter\_outcomes.csv' file into the AAC database.
* Update the port number (Mongo provides the port number when you start the service) during the file import process.

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

* Update the port number on localhost in the 'animal\_shelter.py' Python code.
  + *This is essential for connecting to the MongoDB database.*
* Update the "aacuser" and "password" with the username and password you created.

A screenshot of a computer

Description automatically generated

* Testing:
  + - To run tests, add test code to a Jupyter notebook. Ensure that the test data for the create function is different each time or delete the added record between tests.

A screenshot of a computer program

Description automatically generated

* Dash Web Application:
  + - Create a new Dash web application dashboard and configure it with the desired HTML/CSS layout. Assign appropriate IDs for the data frame, map, and chart components.
* Data Population:
  + - Create an app callback to populate the initial data frame with all available data.
* Filtering Options:
  + - Implement radial options for filtering data based on the desired breed specifications for the client. Be cautious when dealing with complex queries that combine multiple searches, as this may lead to syntax problems if not handled carefully.
* Map Integration:
  + - Create an application callback to update the map with the first item of a given category until the user selects an item. Then, develop functionality to determine the user's selection and display it on the map.
* Pie Chart Creation:
  + - Create a pie chart from the displayed data on the screen at any given moment. To populate the pie chart, use the "values" parameter with an array of numbers to represent occurrences. Additionally, define an index of names under "names" for the chart.
    - Implement the desired functionality for the pie chart and ensure that it correctly targets the data (e.g., 'breed').

## INSTALLATION:

* To run this project, I used the following:
  + - Linux Terminal
    - Python
    - PyMongo Library
    - Jupiter Notebook

## USAGE:

### RELEVANT CODE:

A screen shot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

### SCREENSHOTS:

* Water Rescue

A screenshot of a computer

Description automatically generated

* Mountain/Wilderness Rescue:

A screenshot of a computer

Description automatically generated

* Disaster Rescue/Individual Tracking

A screenshot of a computer

Description automatically generated

A screenshot of a graph

Description automatically generated

A map with a location pin

Description automatically generated

## ROADMAP/FEATURES:

## User Authentication Enhancements:

## Implement role-based access control to differentiate between regular users, administrators, and other user types.

## Allow administrators to manage user accounts and permissions directly through the dashboard.

## Data Visualization Improvements:

## Expand the range of available charts and graphs for better data representation, such as bar charts, histograms, and trend analyses.

## Enable users to customize chart settings and save their preferred chart configurations.

## Advanced Filtering Options:

* + - Provide more advanced filtering options, including multi-select filters, date range filters, and the ability to filter by multiple criteria simultaneously.
    - Implement an intuitive and user-friendly interface for creating and saving custom filters.
  + Data Export and Reporting:
    - Enable users to export filtered data to common formats (e.g., CSV, Excel) for further analysis or reporting purposes.
    - Create customizable report templates that allow users to generate detailed reports on shelter statistics.
  + Geospatial Features:
    - Enhance the geospatial component to display additional location-related information, such as nearby veterinary clinics, adoption centers, or shelters.
    - Implement geofencing and location-based alerts for tracking animals' movements or escapes.
  + Integration with External Services:
    - Integrate with external services or APIs to fetch real-time weather data, local adoption events, or animal-related news to provide users with valuable information.
  + Mobile-Friendly Interface:
    - Develop a responsive design to ensure that the dashboard is accessible and usable on various devices, including smartphones and tablets.
  + Data Synchronization:
    - Explore options for synchronizing data with external databases, shelters, or animal welfare organizations to keep information up to date.

**WHAT SETS OUR PROJECT APART:**

The Animal Shelter Dashboard project stands out by addressing the specific needs of animal shelters and rescue organizations. It offers a comprehensive solution for managing shelter operations, providing insights into animal data, and visualizing geographic information. The project's commitment to user-friendliness and continuous improvement ensures that it remains a valuable tool for animal welfare professionals. Its open roadmap reflects our dedication to enhancing the project's capabilities and making a positive impact on the lives of shelter animals and the dedicated individuals caring for them.

## CONTACT:

Your name: Pravishna Nand

Email: Pravishna.nand@snhu.edu