**CS 340 Project Two READ ME**

**Project 2 Overview**

This project was to develop a fully functional dashboard for Grazioso Salvare, an organization that identifies and trains dogs for search-and-rescue operations. The dashboard is designed to interact with data from a MongoDB database, allowing users to filter, visualize, and manage information about available dogs in a user-friendly interface.

**The dashboard provides the following functionalities:**

* Interactive Data Filtering: Users can filter data based on specific rescue types (Water Rescue, Mountain/Wilderness Rescue, Disaster/Individual Tracking) using radio buttons. The data table and charts update dynamically based on the selected filter.
* Dynamic Data Table: The dashboard features a dynamic table that displays information about the dogs, such as breed, age, and outcome. The table supports sorting, pagination, and filtering by column headers.
* Data Visualization: A pie chart visualizes the distribution of dog breeds based on the applied filters. This chart is hidden when no filter is selected to keep the screen clean.
* Geolocation Mapping: The dashboard includes an interactive map that shows the location of the selected dog.

**Here are Some examples:**

A screenshot of a map

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a map

Description automatically generated

A screenshot of a map

Description automatically generated

**Tools and Libraries Used**

**1. MongoDB**

* MongoDB was chosen as the database for its flexibility in handling data and its speed. It allows for easy integration with Python using the pymongo library, which adds CRUD operations on the database.

**2. Dash by Plotly**

* Dash is a Python framework that is great for building interactive web applications with minimal lines of code. It integrates with Plotly for data visualization and supports various components for building UIs.

**Resources and References**

* MongoDB Documentation: https://docs.mongodb.com/
* Dash by Plotly Documentation: https://dash.plotly.com/

**Project Steps**

1. Setting Up the MongoDB Database

* The database connection was set up using the AnimalShelter class, which handles CRUD operations. An instance of this class was created to interact with the animals collection in the MongoDB database.

2. Building the Dashboard Layout

* The dashboard layout was designed using Dash components. Radio buttons were added to filter data, and a DataTable was set up to display the results. A pie chart and a map were also added to provide visuals.

3. Implementing Data Filtering

* The update\_dashboard callback function was created to filter the data based on the selected rescue type. The filtered data is then displayed in the DataTable and used to generate a pie chart.

4. Adding Data Visualization

* The update\_graphs function generates a pie chart that shows the distribution of dog breeds. The chart is displayed only when a filter is applied.

**Challenges and Solutions**

1. Pie chart showing to much data

Challenge: The pie chart would try and show data for every dog inside the database Making the interface very cluttered.

Solution: Simply only show the chart when a filter is selected.

-Project by Devin Wheeler