**README File**

**Project Overview**

Grazioso Salvare has requested documentation to accompany the code for your dashboard. This document ensures that they can understand the work that was completed and easily maintain the code for this project.

This README file documents the project and provides instructions for reproducing the project.

**Required Functionality**

This project implements a dashboard that allows users to filter and analyze animal data based on different rescue types. The dashboard includes the following features:

* A user interface for selecting different rescue types.
* A dynamically updated table displaying animal records.
* A graphical representation of the data through a pie chart.
* A map that plots the location of selected animals.

**Screenshots**

Below are placeholders for screenshots demonstrating the functionality of the dashboard under different filter selections:

1. **Water Rescue**

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a map

AI-generated content may be incorrect.

1. **Mountain or Wilderness Rescue**
2. **A screenshot of a computer

   AI-generated content may be incorrect.A map of mexico with a location pin

   AI-generated content may be incorrect.Disaster Rescue or Individual Tracking**

A close up of a document

AI-generated content may be incorrect.A screenshot of a map

AI-generated content may be incorrect.

1. **Reset (Unfiltered Data)**

**A screenshot of a computer

AI-generated content may be incorrect.Tools Used & Rationale**

**MongoDB**

MongoDB was used as the database component to store and retrieve animal records. It was chosen due to its flexibility in handling semi-structured data and its seamless integration with Python.

**Dash Framework**

Dash was used to create the web-based dashboard. It provides an intuitive way to build interactive visualizations and data tables using Python.

**Additional Resources**

* MongoDB Documentation: <https://www.mongodb.com/docs/>
* Dash Framework: <https://dash.plotly.com/>

**Steps Taken to Complete the Project**

1. Established a connection to the MongoDB database and retrieved the animal data.
2. Processed the data into a Pandas DataFrame for easy manipulation and filtering.
3. Created an interactive Dash application with:
   * A table displaying animal data.
   * A pie chart representing the distribution of breeds.
   * A map displaying the location of selected animals.
4. Implemented filtering functionality based on different rescue types.
5. Debugged and ensured that all functionality met the project requirements.

**Challenges & Solutions**

1. **Filtering Issues:** Initially, filtering by rescue type did not return expected results. The issue was resolved by ensuring that the correct field in the database was queried.
2. **No Data Displayed in the Table:** The table did not populate correctly at first. This was fixed by checking the connection to MongoDB and ensuring data was successfully retrieved.
3. **Graph and Map Not Updating Properly:** The map and pie chart were not updating dynamically. The issue was resolved by correctly linking callbacks in Dash.

**Instructions for Running the Project**

1. Ensure you have Python and the required libraries installed:
2. pip install pymongo pandas dash plotly dash-leaflet
3. Ensure you have access to the MongoDB database with the correct credentials.
4. Run the Dash application using:
5. python dashboard.py
6. Open the provided local URL in your web browser to interact with the dashboard.