Grazioso Salvare MongoDB Dashboard

Project Overview

This project involves the development of a MongoDB-based dashboard for Grazioso Salvare, an international rescue-animal training company. The dashboard allows users to interact with and visualize data from animal shelters in the Austin, Texas region. The goal is to help identify suitable dogs for search-and-rescue training based on specific criteria.

Features

Interactive Data Table: Displays unfiltered shelter animal data with pagination and sorting.

Filtering Options: Users can filter animals by rescue type (Water Rescue, Mountain/Wilderness Rescue, Disaster/Individual Tracking).

Geolocation Visualization: A map visualization displaying the location of rescue animals.

Additional Data Visualization: A second dynamic chart based on selected filtering criteria.

User-Friendly Interface: Includes the Grazioso Salvare logo and a unique identifier for tracking.

Tools and Technologies Used

MongoDB: Used as the database for storing animal shelter data due to its flexible and scalable schema.

Python: Core programming language for development.

Dash by Plotly: Used to build the interactive web-based dashboard.

Pandas: For data manipulation and preprocessing.

Leaflet.js via Dash: For geolocation mapping.

Installation Instructions

Clone the Repository:

git clone <repository-url>

cd <repository-folder>

Install Dependencies:

pip install -r requirements.txt

Set Up MongoDB Connection:

Ensure MongoDB is running locally or provide a remote MongoDB URI.

Modify the CRUD Python module to include the appropriate MongoDB credentials.

Run the Dashboard:

jupyter notebook

Open ProjectTwoDashboard.ipynb in Jupyter Notebook and run the cells.

Functionality Demonstration

Below are screenshots showcasing the working dashboard:

Dashboard Initial State

Water Rescue Filter Applied

Mountain/Wilderness Rescue Filter Applied

Disaster/Individual Tracking Filter Applied

Reset State

Challenges and Solutions

Challenges:

Connecting Dash to MongoDB: Had issues retrieving and displaying real-time data.

Geolocation Mapping: Properly formatting data for Leaflet.js visualization.

Solutions:

Ensured MongoDB credentials were correctly set up and optimized queries.

Reformatted latitude/longitude values for compatibility with Dash Leaflet components.

Contributors

Susan Chappelone

License

This project is open source and available under the MIT License.