**Grazioso Salvare Dashboard**

Project Overview

The Grazioso Salvare Dashboard is a web application developed for Grazioso Salvare, an innovative international rescue-animal training company. This dashboard allows users to interact with and visualize data from animal shelters in the Austin, Texas region. The data includes information on dogs that are candidates for search-and-rescue training.

The application enables users to filter data based on rescue type, view details in a table format, visualize breed distribution in a pie chart, and see the geolocation of selected animals on a map.

Features

Interactive Data Table: View and interact with the animal data in a sortable and filterable table.

Filter Options: Filter the data based on rescue type (Water Rescue, Mountain Rescue, Disaster Rescue).

Pie Chart: Visualize the breed distribution of the animals based on the selected filter.

Geolocation Map: View the location of the animals on an interactive map.

User-Friendly Interface: The dashboard is designed to be intuitive and easy to use.

Screenshots

A screenshot of a computer

Description automatically generated

Water Rescue Filter

A screenshot of a computer

Description automatically generated

Mountain Rescure filter (different dogs selected)

A screenshot of a computer

Description automatically generatedDisaster Rescue (Golden retrieve instead of Rottys select)

A screenshot of a computer

Description automatically generated

Reset (See all dog types)

Tools and Technologies

* MongoDB: Used as the database to store and retrieve animal data. MongoDB provides a flexible and scalable NoSQL database solution that integrates well with Python.
* Dash and Plotly: Used to create the web application dashboard. Dash is a Python framework for building analytical web applications, and Plotly is a graphing library that provides interactive plots.
* Dash Leaflet: Used for creating interactive maps in the dashboard.
* Pandas: Used for data manipulation and transformation. Pandas makes it easy to work with structured data and convert it into formats suitable for visualization.

**Implementation Details**

MongoDB

MongoDB was chosen as the model component for this project due to its flexibility in handling unstructured data. The AnimalShelter class in the animal\_shelter\_crud module handles the connection to MongoDB and provides CRUD functionality.

Dash Framework

The Dash framework was used to build the web application. It allows for the creation of interactive, web-based dashboards with minimal code. Dash integrates well with Plotly for data visualization and provides a seamless way to build and deploy web applications.

Callback Functions

Several callback functions were implemented to handle the interactivity of the dashboard:

* update\_dashboard: This callback updates the data table, pie chart, and map based on the selected filter.
* update\_styles: This callback updates the styling of the data table based on selected columns.
* update\_graphs: This callback updates the pie chart to reflect the filtered data.
* update\_map: This callback updates the map to show the locations of the filtered animals.

Steps to Reproduce

To reproduce this project, follow these steps:

1. Set Up MongoDB: Ensure you have MongoDB installed and running. Import the animal data into a MongoDB collection named animals in a database named AAC.
2. Clone the Repository: Clone the project repository from GitHub (link to your repository).
3. Install Dependencies: Install the required Python packages using pip install -r requirements.txt.
4. Update Credentials: Update the MongoDB credentials in the AnimalShelter class.
5. Run the Application: Run the application using the Jupyter notebook or a standalone script.

**Challenges and Solutions**

Data Handling

Challenge: Handling large volumes of data from MongoDB and transforming it into formats suitable for visualization.

Solution: Used Pandas for efficient data manipulation and transformation. Added error handling to manage any issues during data conversion.

Interactive Visualizations

Challenge: Ensuring that the interactive components (data table, pie chart, and map) update correctly based on user input.

Solution: Implemented and debugged multiple callback functions to ensure that the components update seamlessly. Added detailed debugging statements to trace the flow of data.

Conclusion

The Grazioso Salvare Dashboard provides a powerful tool for visualizing and interacting with animal shelter data. By leveraging MongoDB, Dash, and Plotly, we created an intuitive and user-friendly interface that allows users to filter, view, and analyze data effectively.