Serenity Rogers

Project Two

CS340-16764

10/19/2024

# **Grazioso Salvare Rescue Animal Dashboard**

## **Project Description**

This project involves the development of a dynamic dashboard for **Grazioso Salvare**, a global rescue animal training organization. The dashboard provides an interactive interface for filtering, visualizing, and mapping rescue animals based on different types of rescue missions. It integrates with a **MongoDB** database to fetch live data on rescue animals, including their breed, outcome types, and rescue locations.

The dashboard also features geolocation data visualization, showing where specific rescue missions occurred on a map. Additionally, it allows users to filter data based on four rescue types: Water Rescue, Mountain/Wilderness Rescue, Disaster Rescue, and Individual Tracking.

## **Functionality**

### **Key Features:**

1. **Interactive Data Filtering**: Users can filter the rescue data based on different rescue types (e.g., Water Rescue, Disaster Rescue).
2. **Data Table**: A table view that allows filtering and sorting of rescue animal details.
3. **Pie Chart**: An automatically updated pie chart showing the distribution of breeds in the dataset.
4. **Map Visualization**: A map showing rescue locations with tooltips and popups displaying the animal's name and details.
5. **Rescue Type Filtering**: Filtered view of the map and data table based on rescue type selection.
6. **Grazioso Salvare Logo and Personal Branding**: The logo of Grazioso Salvare is displayed along with a unique identifier for the developer (Serenity Rogers).

### **Screenshots of Working Dashboard**

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

## **Tools and Technologies Used**

### **1. MongoDB:**

* **Reason for Usage**: MongoDB was chosen as the database because of its flexibility with unstructured data, which is ideal for rescue records that may vary in attributes. It allows the rapid insertion and querying of JSON-like documents. MongoDB’s schema-less nature ensures that it can easily store the diverse attributes of different rescue animals.
* **Key Qualities**:
  + Stores data in a JSON-like format, making it easy to interface with Python and Pandas for data analysis.
  + Provides fast access to data and supports filtering via queries.
  + Can handle large datasets with ease, ideal for large-scale animal rescue operations.

### **2. Dash Framework:**

* **Reason for Usage**: Dash is a Python-based web application framework that provides both the controller and view components of the project. It was chosen for its simplicity in creating web interfaces with Python alone.
* **Key Qualities**:
  + Dash provides a built-in system for linking user interface elements (radio buttons, checkboxes) to backend data and visualizations.
  + Dash integrates seamlessly with Plotly for generating visualizations like charts and maps.
  + The framework is efficient for building dashboards and web applications without needing to manage complex frontend technologies (e.g., JavaScript).

### **3. Plotly:**

* Plotly was used to generate dynamic visualizations such as pie charts and scatter plots. It provides interactive graphs that can easily be integrated into Dash applications.

### **4. Dash Leaflet:**

* This tool was used to integrate map functionality, allowing rescue locations to be visualized dynamically based on filtered data.

### **5. Pandas:**

* Pandas was used for efficient data handling, filtering, and transforming the MongoDB data into tabular format for further processing and visualization.

## **Steps to Complete the Project**

### **1. Data Fetching from MongoDB:**

* Established a connection to a MongoDB collection using pymongo.
* Fetched the data from the dogs collection and transformed it into a DataFrame using Pandas for easier manipulation and analysis.

### **2. Dashboard Development:**

* Built the dashboard layout using Dash, including:
  + Radio buttons for rescue type selection.
  + A data table using dash\_table.DataTable to display filtered rescue animal records.
  + A pie chart using Plotly to visualize the most common animal breeds.
  + An interactive map using Dash Leaflet for geospatial visualization of rescue locations.

### **3. Callback Functions:**

* Created callback functions that:
  + Filter the data based on rescue type.
  + Update the pie chart and map dynamically as users interact with the dashboard.
  + Display tooltips and popups on the map to show details about each rescue.

### **4. Incorporation of Logo and Branding:**

* Added the **Grazioso Salvare** logo to the dashboard using base64 encoding to embed it directly into the app.
* Included the developer’s name (Serenity Rogers) as part of the branding in the app layout.

### **5. Deployment and Testing:**

* Tested the dashboard to ensure all components worked correctly, including filtering by rescue type, updating graphs, and interactive maps.
* Ensured that MongoDB interactions performed efficiently and returned the correct data for visualization.

## **Challenges Encountered**

### **1. Handling Missing Data:**

* Certain fields in the dataset (like latitude and longitude) were missing or incorrectly formatted. This required data cleaning to handle missing or erroneous values using Pandas.

### **2. Geospatial Mapping:**

* Aligning map markers with correct rescue locations was initially tricky. Dash Leaflet required specific formatting of the lat and long fields, and ensuring that out-of-range values didn’t cause errors was critical.

### **3. Callback Errors:**

* There were initial issues with callbacks, especially when no rows were selected in the data table. This was handled by adding conditions to check for empty selections and defaulting to showing all data.

## **Conclusion**

This project successfully achieved its goal of building a dynamic and interactive dashboard for **Grazioso Salvare**. The use of MongoDB allowed efficient storage and querying of rescue data, while Dash and Dash Leaflet provided a user-friendly interface for visualizing and interacting with the data.

Resources

Dash. (n.d.). *Dash documentation*. Plotly. Retrieved from <https://dash.plotly.com/>

MongoDB. (n.d.). *MongoDB documentation*. MongoDB, Inc. Retrieved from<https://docs.mongodb.com/>

Plotly. (n.d.). *Plotly Python graphing library*. Plotly. Retrieved from <https://plotly.com/python/>

The Pandas Development Team. (n.d.). *Pandas documentation*. Pandas. Retrieved from https://pandas.pydata.org/