Darius Stewart

Southern New Hampshire University

CS-340 Client/Server Development

Professor Paruchuri

June 23, 2024

**Grazioso Salvare Dashboard README**

**Project Overview**

Grazioso Salvare, an international rescue-animal training company, uses this dashboard to identify and categorize dogs suitable for search-and-rescue training. The dashboard allows users to filter data, visualize results, and interact dynamically with the dataset from local animal shelters.

**Required Functionality**

The dashboard provides the following functionality:

1. **Interactive Data Table:** Displays animal data with filtering options for rescue types.
2. **Dynamic Charts:** Visualizes the filtered data in charts, including a pie chart for breed distribution and a map for geolocation.
3. **Filter Options:** Allows users to filter the data by rescue type (Water Rescue, Mountain or Wilderness Rescue, Disaster or Individual Tracking).
4. **Real-Time Updates**: Components update dynamically based on user input.

**Screenshots**

1. **Water Rescue Filter**

This screenshot demonstrates the dashboard with the "Water Rescue" filter applied. It highlights how the data table and visualizations update to show only dogs suitable for water rescue missions.

A screenshot of a computer

Description automatically generated

1. **Mountain or Wilderness Rescue Filter**

This screenshot illustrates the dashboard with the "Mountain or Wilderness Rescue" filter selected. It showcases the dynamic filtering capability, displaying data specific to dogs trained for mountain or wilderness rescue operations.

A screenshot of a computer screen

Description automatically generated

1. **Disaster or Individual Tracking Filter**

This screenshot shows the dashboard filtered by "Disaster or Individual Tracking." It emphasizes how the dashboard adapts to show relevant data for dogs trained in disaster response or individual tracking scenarios.

A screenshot of a map

Description automatically generated

1. **Reset Filter**

This screenshot depicts the dashboard after resetting the filters. It returns the view to the initial state, displaying the full dataset and demonstrating the ease of resetting and reapplying filters.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Installation**

**Prerequisites**

* Python 3.x
* MongoDB

**Usage**

After setting up, you can interact with the dashboard by:

* Selecting rescue type filters to view specific data subsets.
* Viewing the updated data table and charts.
* Observing geolocation data on the map.

**Technologies Used**

* **MongoDB:** For data storage and retrieval.
  + *Flexibility:* MongoDB's schema-less nature allows for easy storage and retrieval of JSON-like documents, which is ideal for the varying data structures in animal records.
  + *Scalability:* It can handle large volumes of data and scale horizontally, ensuring performance remains consistent.
  + *Ease of Integration:* The PyMongo library provides a seamless interface for interacting with MongoDB directly from Python, simplifying CRUD operations.
* **Dash:** For building interactive web applications.
  + *Simplicity and Power:* Dash simplifies the creation of interactive web applications using Python alone, leveraging the power of Plotly for visualizations.
  + *Component-based Architecture:* It allows the creation of reusable components, making the application modular and easy to maintain.
  + *Interactivity:* Dash supports real-time updates and interactivity, essential for creating dynamic data visualizations.
* **Pandas:** For data manipulation.
* **Plotly Express:** For creating the pie chart, offering high-quality and easy-to-use plotting functions.
* **Dash Leaflet:** For mapping components, providing robust mapping capabilities.

**Steps to Complete the Project**

1. **Set Up MongoDB and CRUD Module:** Established a MongoDB database and created a Python module for CRUD operations.
2. **Develop Data Table:** Created an interactive data table using Dash DataTable, displaying data from MongoDB.
3. **Implement Filter Options:** Added radio buttons for filtering data by rescue type and implemented the corresponding queries.
4. **Create Dynamic Charts:** Developed a pie chart to show breed distribution and a map to display the geolocation of animals.
5. **Integrate Components:** Ensured the interactive components work together to dynamically update the data table and charts.
6. **Test and Deploy:** Tested the dashboard for functionality and took screenshots to demonstrate its features.

**Challenges and Solutions**

**Challenge : Dynamic Updates**

* **Issue:** Ensuring charts and data table update dynamically based on filter selections.
* **Solution:** Utilized Dash's callback functions to link the filter options with data retrieval and chart updates, ensuring real-time interactivity.