Braydon Woodward

CS-340

August 17, 2024

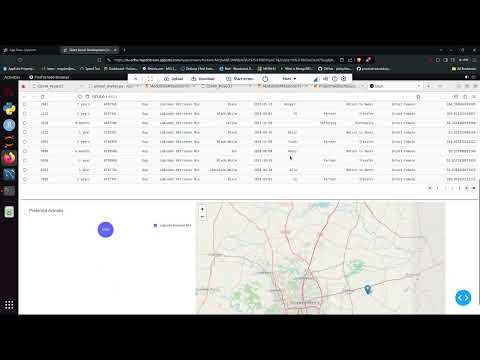
**Grazioso Salvare Dashboard README**

**Project Overview**

The Grazioso Salvare dashboard is a web-based application designed to assist in identifying and categorizing available dogs from animal shelters for search-and-rescue training. The dashboard provides a user-friendly interface to filter and visualize data on dogs based on various criteria.

**Required Functionality**

* **Interactive Filters**: Users can filter the data by categories such as Water Rescue, Mountain or Wilderness Rescue, Disaster or Individual Tracking, and Reset.
* **Data Table**: Displays detailed information about the filtered dogs.
* **Graph**: A pie chart showing the distribution of breeds among the filtered data. It does not appear when all animals are selected.
* **Map**: Displays geolocation of selected dogs on a map.

**Screencast**: [](https://www.youtube.com/embed/8aVKOZVJObE?feature=oembed)

**Tools Used**

* **Python**: The main programming language used for backend logic and data processing for the CRUD Module.
* **JupyterDash Framework**: Provides the structure for the web application, handling the view and controller aspects. It enables interactive components and callbacks within Jupyter Notebooks.
* **Plotly**: Utilized for creating visualizations and graphs, including the interactive charts within the dashboard.
* **Pandas**: Used for data manipulation and analysis, enabling efficient handling of the dataset.
* **Dash Leaflet**: Implemented for rendering the map component, allowing for interactive geographical data visualization.

**Rationale for Using MongoDB**

MongoDB was chosen as the database for this project due to its flexibility and scalability. It allows for easy storage and retrieval of unstructured data, which is ideal for handling the diverse and dynamic nature of animal data. MongoDB's ability to interface with Python via libraries like pymongo makes it a suitable choice for this dashboard.

**Explanation of Dash Framework**

Dash is a Python web application framework that simplifies the process of building interactive web applications. It integrates with Plotly for data visualization and provides a straightforward way to create web interfaces using Python. Dash manages the view and controller structure, allowing for the creation of interactive components like dropdowns, radio buttons, and data tables.

**Resources and Software**

* **Dash Documentation**: https://dash.plotly.com/
* **Plotly Express Documentation**: https://plotly.com/python/pie-charts/
* **Dash Leaflet Documentation**: https://dash-leaflet.plotly.com/
* **MongoDB Documentation**: <https://docs.mongodb.com/>
* **Python Documentation**: <https://docs.python.org/3/>

**Steps to Complete the Project**

1. **Project Setup**:
   * Install necessary libraries: Dash, Plotly, Dash Leaflet, pandas.
   * Set up the database structure and configure the development environment to match your username, password, host, port, database and collection.
2. **Data Preparation**:
   * Load the data using the Mongo shell.
   * Set up PyMongo module to interact with MongoDB to store and retrieve data as needed.
3. **Dashboard Development**:
   * Create the layout of the dashboard using Dash components.
   * Implement interactive filters using Dash callbacks.
   * Integrate Plotly Express for data visualization and Dash Leaflet for map functionality.
4. **Testing**:
   * Test the functionality of the dashboard, including interactive filters and data displays.

**Challenges and Solutions**

* **Challenge**: Ensuring the data table fits within the visible screen area without overlapping.
  + **Solution**: Adjusted column width, table styling, and left irrelevant columns off the table to ensure proper display of all columns.
* **Challenge**: Handling NoneType errors in callbacks when first deploying the dashboard.
  + **Solution**: Implemented checks to handle None values and provided default responses for empty data.
* **Challenge**: Aligning components and ensuring proper layout.
  + **Solution**: Utilized CSS styling and Dash layout components to achieve the desired appearance and spacing for the logo in the top left.

For any further questions or issues, please contact Braydon Woodward - <https://github.com/BraydonWoodward> .