Katelynn Thompson

Satish Penmatsa

CS-340

3/3/24

README for Project Two

**Overview/Motivation**

The animal shelter project for Graziso Salvare is an interactive web application designed to assist in animal rescue operations by utilizing the Austin Animal Center Outcomes dataset. The dashboard enables users to filter data according to specific rescue operation types and visualize this data through interactive components such as maps, charts, and tables.

**Required Functionality**

The dashboard incorporates the following key functionalities:

1. **Interactive Filtering Options**: Users can filter the dataset by rescue operation types:
   * Water Rescue
   * Mountain/Wilderness Rescue
   * Disaster Rescue and Individual Tracking
   * Reset to display all data
2. **Data Table**: Displays detailed animal information, dynamically updating based on the applied filters.
3. **Geolocation Chart**: Visualizes the location of animal rescues on a map, aiding in the planning and analysis of rescue operations.
4. **Bar Chart**: A bar chart that provides additional insights into the data, such as the breed quantity of rescued animals, dynamically responding to applied filters.

**Tools and Technologies**

**MongoDB:** I chose MongoDB, a NoSQL database, for its flexibility and scalability in handling unstructured data. Its document-oriented model is particularly well-suited for storing the varied data of animal rescue operations. Additionally, MongoDB's compatibility with Python through the Pymongo library simplifies data manipulation and retrieval, enabling efficient data processing for the dashboard.

**Dash Framework**: Dash allows for developing highly interactive web applications purely in Python without needing direct use of HTML or JavaScript. Dash's components and callback system serve as the view and controller layers, respectively, managing user interactions and dynamic content rendering. I selected Dash for its ease of use in creating data-driven applications and its extensive support for interactive components, making it ideal for the Grazioso Salvare Dashboard.

**Project Reproduction Steps**

* + 1. Setup MongoDB: Install MongoDB and import the Austin Animal Center Outcomes dataset into a collection.
    2. Install Dependencies: Use pip to install required Python libraries: dash, pandas, plotly, and pymongo.
    3. Run the Dashboard: Execute the Python script containing the dashboard code. Ensure MongoDB is running and accessible.
    4. Access the Dashboard: Open a web browser and navigate to the URL provided by the script.

**Challenges**

One of the major challenges I encountered during the development process was making sure that the filters, maps, and charts were working together seamlessly. To address this issue, I utilized the callback system in Dash to dynamically update the various components based on user input. With this approach, I was able to solve the problem and ensure that the filters were functioning properly with the maps and charts.

A screenshot of a computer

Description automatically generatedA screenshot of a computer screen

Description automatically generated

A screenshot of a map

Description automatically generatedA screenshot of a map

Description automatically generated

A screenshot of a map

Description automatically generated