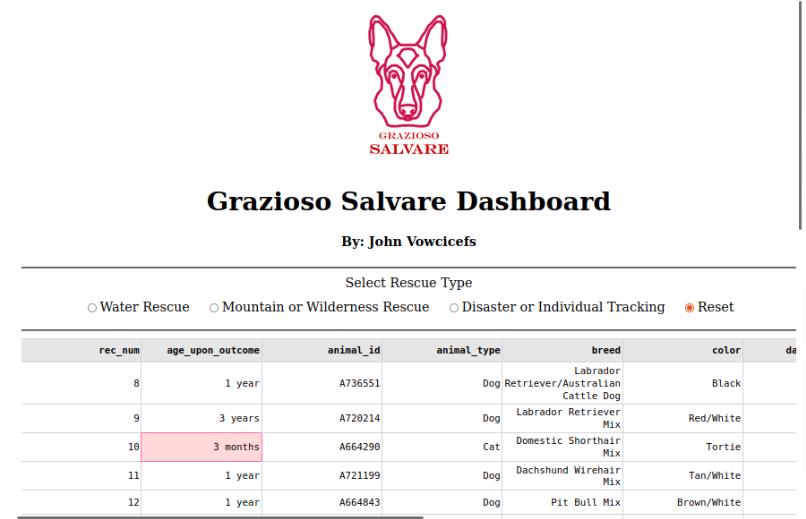
**Grazioso Salvare Dashboard Project – README File**

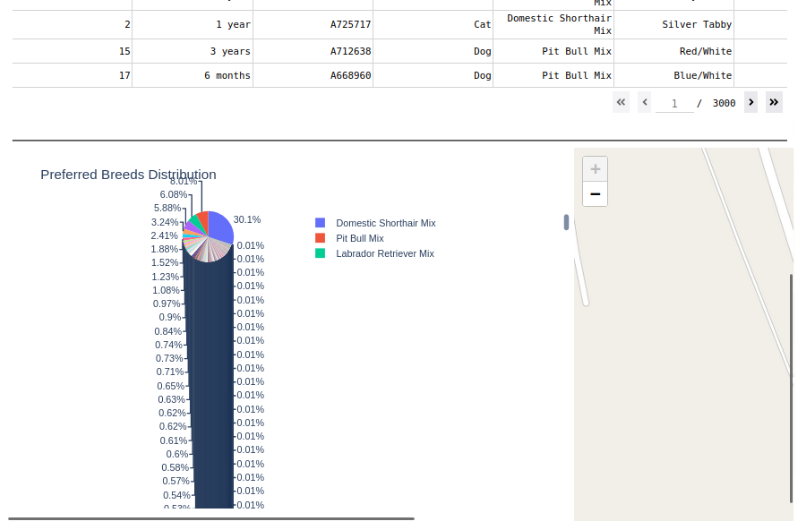
**Author**: John Vowcicefski  
**Project**: CS 340 - Client/Server Development  
**Institution**: Southern New Hampshire University  
**Submission Date**: 8/18/2024

**Project Overview**

This project involves the development of an interactive web application dashboard for Grazioso Salvare, an innovative international rescue-animal training company. The dashboard is designed to help the company identify dogs that are good candidates for different types of rescue training by interacting with data from local animal shelters in Austin, Texas. The application was built using Python's Dash framework and MongoDB for database operations.

**Screenshots:**

****

****

**Functionality**

The dashboard provides the following functionalities:

* **Data Table**:
  + Displays data from the Austin Animal Center Outcomes dataset.
  + The table is interactive, allowing users to sort and paginate through the data.
  + The table dynamically updates based on the filters applied by the user.
* **Filtering Options**:
  + Users can filter the dataset by selecting one of the following rescue types:
    - Water Rescue
    - Mountain or Wilderness Rescue
    - Disaster or Individual Tracking
    - Reset (to display all data)
  + These filters are applied using radio buttons that interact with the database to retrieve relevant records.
* **Charts**:
  + A pie chart visualizes the distribution of preferred breeds based on the selected rescue type filters.
  + The chart updates dynamically to reflect changes made by the user to the filter options.
* **Map**:
  + A geolocation map displays the locations of animals based on the selected data from the table.
  + The map updates based on the user's selection in the data table.

**Tools and Technologies Used**

* **Python**: The primary programming language used to develop the dashboard.
* **Dash**: A Python framework that is particularly suited for creating interactive web applications and data visualization dashboards.
* **JupyterDash**: An extension of Dash that allows the dashboard to be run directly within a Jupyter notebook.
* **MongoDB**: A NoSQL database used to store and query the Austin Animal Center Outcomes data.
* **Dash Leaflet**: A library used for creating the interactive map within the dashboard.
* **Plotly**: A graphing library used to create the pie chart for data visualization.
* **Pandas**: A data manipulation library that was used to process and analyze the dataset.

**Project Structure**

* **ProjectTwoDashboard.ipynb**: The Jupyter notebook containing the complete code for the dashboard.
* **animal\_shelter.py**: A Python module that includes CRUD (Create, Read, Update, Delete) operations for interacting with the MongoDB database.
* **Grazioso Salvare Logo.png**: The logo used in the dashboard for branding purposes.
* **README.md**: This documentation file.

**Installation and Setup**

To run this project locally, follow these steps:

1. **Clone the Repository**: Clone the project repository to your local machine.
2. **Install Required Libraries**: Ensure that the following Python libraries are installed:
   * dash
   * jupyter-dash
   * dash-leaflet
   * plotly
   * pandas
   * pymongo
   * numpy
3. **Run Jupyter Notebook**: Open the ProjectTwoDashboard.ipynb file in Jupyter Notebook.
4. **Execute the Notebook**: Run all cells in the notebook to launch the dashboard within the notebook environment.

**Usage Instructions**

1. **Start the Dashboard**: The dashboard will load within the Jupyter notebook interface.
2. **Apply Filters**: Use the radio buttons to filter the dataset based on the rescue type.
3. **View Data**: The data table will update based on the selected filter, displaying relevant animal data.
4. **Analyze Data**: Review the pie chart and map for visual insights into the preferred breeds and geolocation of animals.
5. **Reset Filters**: Click "Reset" to clear filters and return to the full dataset view.

**Challenges and Solutions**

**Challenge 1: MongoDB Connection**

Initially, connecting securely to the MongoDB database and fetching data using the AnimalShelter class was challenging. This was resolved by ensuring the correct parameters (host, port, database name, collection name) were passed when initializing the class.

**Challenge 2: Dynamic Updates**

Ensuring that the data table, pie chart, and map components dynamically updated based on user interactions required careful implementation of Dash callbacks. These callbacks successfully linked the filter options with the data display components.

**Challenge 3: Dashboard Layout**

Creating a user-friendly and visually appealing dashboard layout involved multiple iterations. The use of Dash components and styling options helped in achieving a clean and functional design that meets the project requirements.

**References**

1. Austin Animal Center. (2020). Austin Animal Center Outcomes [Data set]. City of Austin, Texas Open Data Portal. <https://doi.org/10.26000/025.000001>
2. MongoDB Documentation. (n.d.). Introduction to MongoDB. MongoDB, Inc. <https://www.mongodb.com/docs/>
3. Dash by Plotly. (n.d.). Dash User Guide and Documentation. <https://dash.plotly.com/>
4. Leaflet Documentation. (n.d.). Leaflet - JavaScript Library for Interactive Maps. <https://leafletjs.com/>