# **Grazioso Salvare Animal Rescue Dashboard**

## **Overview**

The **Grazioso Salvare Animal Rescue Dashboard** is an interactive web application designed to help the Grazioso Salvare company analyze data from the Austin Animal Center. The dashboard provides insights into animal rescues based on breed types, rescue categories, and geo-location. The dashboard was developed using Python's Dash framework and MongoDB for data storage and management.

This project is part of the CS-340 course at SNHU, and it demonstrates the use of CRUD operations on a MongoDB database, data visualizations using Plotly, and geospatial data mapping with Dash Leaflet.

## **Features**

1. **Data Filtering**: Users can filter animal data based on the type of rescue (Water Rescue, Mountain or Wilderness Rescue, Disaster or Individual Tracking).
2. **Interactive Data Table**: The dashboard displays a dynamic table of animal outcomes, allowing users to explore the available data interactively.
3. **Pie Chart Visualization**: A pie chart updates dynamically based on the filtered data, showing the distribution of preferred rescue dog breeds.
4. **Geo-location Map**: A map centered on Austin, Texas, shows the location of selected animal breeds with tooltips for additional information.
5. **CRUD Functionality**: The project uses the AnimalShelter class to perform Create, Read, Update, and Delete operations on the MongoDB dataset.

## **Project Files**

* **ProjectTwoDashboard.ipynb**: The Jupyter Notebook file containing the main code for the Dash web application.
* **animal\_shelter.py**: The Python module for CRUD operations, connecting to the MongoDB database.
* **Grazioso\_Salvare\_Logo.png**: The company logo used in the dashboard layout.
* **README.md**: This file, which provides instructions and details about the project.

## **Requirements**

To run this project, the following libraries are required:

* dash
* dash-leaflet
* dash-bootstrap-components
* pandas
* pymongo
* plotly
* jupyter-dash

Ensure MongoDB is running locally or remotely, and you have the correct connection details (username, password, database name) for the CRUD module to interact with the database.

## **Setup Instructions**

**Install Dependencies**: Install the required libraries using pip.  
  
pip install dash pandas pymongo dash-leaflet dash-bootstrap-components plotly jupyter-dash

1. **Set Up MongoDB**: Ensure that MongoDB is installed and running on your machine. The default configuration uses the following credentials:
   * **Username**: aacuser
   * **Password**: SNHU1234
   * **Database**: Austin Animal Center
2. **Run the Dashboard**:
   * Open ProjectTwoDashboard.ipynb in Jupyter Notebook.
   * Run all the cells to start the Dash web server and launch the dashboard in your browser.
3. **CRUD Operations**: The animal\_shelter.py file contains the AnimalShelter class, which is responsible for the following CRUD operations:
   * create(): Add new data to MongoDB.
   * read(): Query data from MongoDB.
   * update(): Modify existing records in MongoDB.
   * delete(): Remove data from MongoDB.
4. **Explore the Dashboard**: After running the dashboard, interact with the filters, data tables, and visualizations. Use the dropdown to filter data by rescue type, and observe the changes in the pie chart and map.

## **Example Queries**

The following are example filters used to retrieve specific animal data for different rescue scenarios:

* **Water Rescue**: Retrieves Labrador Retriever and Golden Retriever breeds.
* **Mountain or Wilderness Rescue**: Retrieves German Shepherd and Alaskan Malamute breeds.
* **Disaster or Individual Tracking**: Retrieves Bloodhound and Coonhound breeds.

These filters are dynamically applied using MongoDB queries in the background.

## **Dashboard Layout**

* **Grazioso Salvare Logo**: Displayed at the top of the dashboard.
* **Rescue Type Filter**: Dropdown menu allowing users to select specific rescue types.
* **Animal Data Table**: Interactive table displaying data such as breed, age, and outcome type.
* **Pie Chart**: Visualizes the distribution of rescue dog breeds.
* **Map**: Displays the location of the selected breed on a map centered in Austin, Texas.

## **Future Enhancements**

* Add more complex filtering options such as age groups, outcome types, and animal colors.
* Implement additional visualizations (e.g., bar charts, line graphs) to provide deeper insights into the data.
* Extend the geolocation feature to dynamically center on various shelters or cities based on user input.

## **Authors**

* Kholood (Project Developer)

## **License**

This project is licensed under the MIT License.