# **Grazioso Salvare Animal Rescue Dashboard**

## **Project Overview**

This project is a full-stack dashboard application developed as part of the CS-340 course to assist Grazioso Salvare, an international rescue-animal training company. The dashboard allows users to interact with a MongoDB database of dogs from animal shelters around Austin, Texas, filtering and visualizing dogs that are good candidates for various types of rescue training.

## **Features and Functionality**

* **Data Table:** Displays a dynamic, paginated table of dogs with detailed attributes.
* **Filtering Options:** Users can filter the data by rescue types, including Water Rescue, Mountain/Wilderness Rescue, Disaster Rescue & Tracking, or reset to view all dogs.
* **Charts:** A histogram visualization shows the distribution of dog breeds based on the current filter.
* **Map:** An interactive map displays the selected dog's location with tooltips and popups.
* **Branding:** Includes the Grazioso Salvare logo and developer identification on the dashboard.
* **Interactivity:** Selecting filters updates both the data table and visual components dynamically.

## **Technologies Used**

* **MongoDB:** Used as the backend database to store and manage animal shelter data. MongoDB was chosen for its flexible document model, scalability, and ease of integration with Python.
* **Python:** Serves as the main programming language for backend logic, including database CRUD operations and dashboard integration.
* **Dash Framework:** Provides the MVC structure for the dashboard — Dash’s reactive components handle the view and controller, while MongoDB acts as the model.
* **Pandas:** Used to manipulate and transform data retrieved from MongoDB into tabular formats for visualization and display.
* **Plotly Express:** Enables fast and simple generation of interactive charts for breed distribution.
* **Dash Leaflet:** Offers an easy way to embed interactive maps into the dashboard with markers and popups.

## **Setup Instructions**

1. **Clone the Repository**

bash

CopyEdit

git clone https://github.com/yourusername/grazioso-salvare-dashboard.git

cd grazioso-salvare-dashboard

1. **Install Required Packages**

Make sure you have Python 3.7 or later installed. Then install dependencies:

bash

CopyEdit

pip install -r requirements.txt

1. **Configure MongoDB**

* Ensure MongoDB is running locally or remotely.
* The database AAC should be accessible with username and password: aacuser.
* Import or insert the animal shelter dataset into the animals collection in the AAC database.

1. **Run the Dashboard**

bash

CopyEdit

python ProjectTwoDashboard.py

1. **Access the Dashboard**

Open your browser and go to http://127.0.0.1:8050

## **Project Development Process**

* Developed a MongoDB CRUD Python module (animalShelter.py) to interact with the animals collection.
* Created the initial unfiltered data table to display all shelter dogs.
* Implemented filters based on rescue types, mapping to preferred dog breeds, sex, and age ranges.
* Connected filters to update the data table and linked visual widgets (histogram and map).
* Ensured responsive UI components and error handling for empty or missing data.
* Added branding elements and a clear dashboard title including the developer’s unique identifier.
* Tested functionality through multiple scenarios and captured screenshots as proof.

## **Challenges and Solutions**

* **MongoDB Query Construction:** Designing accurate queries to filter by multiple criteria (breed, sex, age) required careful use of MongoDB query operators.
* **Dynamic Dashboard Updates:** Ensuring all components (table, chart, map) react synchronously to user input was handled by Dash’s callback system.
* **Data Consistency:** Some documents had missing or inconsistent location data, which required conditional handling to avoid errors.
* **Environment Setup:** Running the full dashboard locally required installing several Python packages and a working MongoDB instance, which was addressed with clear setup instructions.

## **Resources and References**

* [MongoDB Official Documentation](https://docs.mongodb.com/)
* Dash by Plotly Documentation
* Plotly Express Documentation
* Dash Leaflet Documentation
* Python Pandas Documentation

## **Contact**

For questions or contributions, please reach out to:

**Your Name** Email: brandon.parkerson@snhu.edu  
 GitHub: [github.com/](https://github.com/yourusername)brandon-parkerson