# **CS 340 Project Two – Grazioso Salvare Dashboard**

## **Project Overview**

This project was developed for Grazioso Salvare, an international rescue animal training company. The dashboard application allows users to identify and filter dogs using regional animal shelter data that could be ideal candidates for various search-and-rescue training. The dashboard provides an interactive web interface through which users can view, filter, sort, and visualize data from the Austin Animal Center dataset.

## **Required Functionality**

The Grazioso Salvare dashboard meets all the required functionalities:

• An interactive data table that displays records from the MongoDB database.  
• A filter system using radio buttons that allows users to select rescue categories such as: Water Rescue, Mountain or Wilderness Rescue, Disaster or Individual Tracking, and Reset.  
• Two charts that update in real time: a pie chart that shows the top breeds by count and a geolocation map that pinpoints animal locations.  
• All components update using Dash callbacks.  
• The Grazioso Salvare logo and a unique identifier appear at the top of the dashboard.

## **Tools and Technologies**

• MongoDB – Database that stores and retrieves animal records efficiently.

• Dash by Plotly – Framework for the web application interface that allows real time updates, interactivity, and visualization.

• CRUD Python Module – Connects the dashboard to the MongoDB database and allows Create, Read, Update, and Delete operations.

• Dash Leaflet – Generates the interactive geolocation map where users can see animal locations in the Austin, Texas region.

## **Setup and Reproduction Instructions**

To reproduce this project in Codio (or similar environment), follow these steps:

1. Open the Project workspace in Codio. Locate ProjectTwoDashboard.ipynb and CRUD\_Python\_Module.py in the file tree.
2. Import the logo file.
3. In the .ipnyb file, set the credentials to username to aacuser and password to what you had created earlier.
4. In the layout cell, show the logo using base64 and center it with html.Center. Set the style ={‘height’: ‘40px’} to shrink the logo size.
5. Add the Rescue Filter radio group with id=’filter-type’ and the options: Water Rescue, Mountain or Wilderness Rescue, Disaster or Individual Tracking, and Reset.
6. Enable features in the provided DataTable by adding: filter\_action, sort\_action, sort\_mode, row\_selectable, and selected\_rows.
7. In the update\_dashboard callback, build the Mongo query for each filter. You need to include that the max age needs to be equal to or less than 104 weeks and the case-insensitive breed regex.
8. In update\_graphs, generate a pie chart of the top 10 breeds.
9. Leave the provided map callback unchanged.
10. Verify functionability by selecting each filter option and confirming the table, chart, and map update.

## **Testing and Verification**

The following screenshots demonstrate the functionality and interactivity of the dashboard:

**Starting Dashboard**  
A screenshot of a computer screen

AI-generated content may be incorrect.

**Water Rescue Dashboard**  
A close-up of a map

AI-generated content may be incorrect.

**Mountain or Wilderness Rescue Dashboard  
A screenshot of a computer

AI-generated content may be incorrect.**

**Disaster or Individual Tracking Dashboard  
A screenshot of a computer

AI-generated content may be incorrect.**

**Reset Dashboard  
A screenshot of a computer

AI-generated content may be incorrect.**