Abdulrahman Al-Nachar

CS340 / Project 2 / README — Grazioso Salvare Dashboard Project

This project was made to help Grazioso Salvare identify dogs that are good candidates for different types of search-and-rescue training. The dashboard works by connecting to the MongoDB database to pull data about shelter animals.

It has the following components:

1. A data table that shows an unfiltered view of the Austin Animal Center Outcomes data set.
2. Interactive filter options (radio buttons) that let the user filter for Water Rescue, Mountain Rescue, Disaster Rescue, or Reset to show all.
3. A geolocation chart that updates to show the location of the selected dog on the map.
4. A second chart shows the distribution of breeds in the filtered data.
5. Grazioso Salvare logo and my unique identifier are displayed to match branding requirements.
6. Screenshots are included in the submission to show the starting state and each filter applied (Water Rescue, Mountain Rescue, Disaster Rescue, and Reset). These screenshots prove the dashboard works as intended.

These are the tools that were needed and used to create this project:

1. MongoDB: Used as the database to store and retrieve all the shelter animal data. It’s flexible for storing JSON-like documents, which makes it simple to run dynamic queries with Python.
2. Python: The main programming language used to connect the backend CRUD operations to the dashboard.
3. Dash / Plotly / Dash Leaflet: Dash was used for building the web application. Dash works well for building dashboards because it ties the view and controller together. Plotly was used for the charts, and Dash Leaflet was used for the interactive map.
4. Pandas: Used to handle the data returned from MongoDB and to feed it into the Dash components.

Resources used:

1. MongoDB official docs
2. Dash and Plotly official docs
3. Course-provided sample data set (Austin Animal Center Outcomes)
4. Grazioso Salvare branding assets (logo)

Steps taken:

1. Set up the CRUD Python module to connect to MongoDB.
2. Built the base dashboard with an unfiltered data table that pulls all animals from the database.
3. Created the queries for each rescue type filter based on the Dashboard Specifications Document.
4. Added radio items to let the user select the rescue type filter.
5. Linked the filter to the data table, map, and pie chart so that all widgets update when the filter changes.
6. Added the Grazioso Salvare logo and my name for branding.

**Challenges and how I overcame them**

One thing that gave me some trouble was getting the map to update when the data table changed. I fixed that by tweaking the callbacks and double-checking that the column indexes lined up with what the Leaflet map needed.

Another issue was making the pie chart look right. At first, it didn’t match the style I wanted, and the colors didn’t fit the branding. I fixed that by setting the colors manually and making sure the chart showed clear and useful info.

The last challenge was the layout. I played around with the styling and spacing until it looked neat and everything lined up in a way that’s easy to read and use.

**Screenshots:**

A screenshot of a computer

AI-generated content may be incorrect.A map of the united states

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A map with a location pin

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A map with a location pin

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a map

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a map

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.