Grazioso Salvare Dashboard - README

**Developer:** Valerie Dawson  
**Course:** CS-340 Client/Server Development  
**Project:** Dashboard for Animal Rescue Analysis  
**Technology Stack:** Dash, Python, MongoDB, Plotly, Dash Leaflet

This dashboard was built for my CS-340 Client/Server Development course as a solution for Grazioso Salvare, a company that specializes in training rescue animals. The goal was to create a user-friendly dashboard that helps the client explore animal shelter data and identify dogs suitable for specific types of training like Water Rescue, Mountain/Wilderness Rescue, and Disaster or Individual Tracking. I designed the dashboard to provide fast access to meaningful information that would support their decision-making process when evaluating potential rescue animals.

I used Dash (a Python framework) to build the dashboard and integrated several components that make the data interactive and visual. One of the main features is a radio button filter that lets the user choose a training type. Once selected, the rest of the dashboard updates automatically to show only the animals that match the selected category based on breed, age, and sex.

For the visuals, I included a geolocation map using Dash Leaflet, which shows each animal’s shelter location on an interactive map. I also added a pie chart using Plotly Express to show the distribution of animal breeds, so the client can get a quick look at the breed breakdown for the filtered animals. To make the data more accessible, I added a Dash DataTable that shows the full details for each animal. It’s sortable, paginated, and allows row selection to help the client dig deeper into individual records.

The data comes from a MongoDB database, which I accessed using a custom CRUD module I wrote in Python. I chose MongoDB because its flexible structure works really well with shelter data, which can vary by record. It also supports geolocation, which made mapping the animal locations easier. I used Pandas to clean and format the data before feeding it into the dashboard, and Base64 encoding to embed the Grazioso Salvare logo directly into the layout without relying on external image files.

This project taught me how to combine several different technologies into a single, interactive client/server application. I was able to apply everything from CRUD database access to dynamic visual rendering in Dash, and it gave me a real sense of how data tools can support real-world needs.

Challenges and How I Solved Them

Like with most projects, there were a few challenges I had to work through. One of the bigger ones was getting Dash Leaflet to work properly. I initially tried to return the map like a regular Plotly figure, which doesn’t work since Leaflet components have to be placed directly into the layout using the 'children' property. Once I realized that, I restructured my callback and layout to handle the map correctly and it worked!

I also ran into some issues with the data from MongoDB. Some records were missing location data, and others had latitude and longitude as strings instead of numbers. This caused crashes when trying to map the data. I fixed this by using pandas.to\_numeric() with error handling, and dropping any rows that didn’t have proper location data.

Another tricky part was embedding the Grazioso Salvare logo with Base64. I had to make sure it encoded correctly and displayed nicely at the top of the dashboard. Some trial and error with the styling got it looking just right.

Lastly, I had to make sure everything updated together when a filter was selected. It took a few tries to get the callback working the way I wanted, but once I figured out how to return the right data and visuals at the same time, it came together smoothly.

ScreenShots/ScreenCast

[**https://youtu.be/Ks\_hFTRP4bc**](https://youtu.be/Ks_hFTRP4bc)