Grazioso Salvare Animal Shelter Dashboard - README

# Overview

This dashboard project was developed for Grazioso Salvare to improve their animal rescue and placement operations. It provides an interactive web-based interface for exploring and analyzing animal data stored in a MongoDB database. The dashboard supports visual data analysis, filtering, and geolocation mapping, allowing users to make informed decisions regarding animal rescue and adoption.

A screenshot of a computer

AI-generated content may be incorrect.

# Required Functionality

The dashboard includes the following core features:  
- Interactive Filtering: Users can filter animals by type, breed, or other criteria.  
- Dynamic Data Table: Displays MongoDB data in a searchable, sortable table.  
- Bar Chart Visualization: Graphical representation of animal breeds.  
- Geolocation Mapping: An interactive map using dash\_leaflet.  
- Logo and Branding: Grazioso Salvare’s logo is included in the layout.  
- Responsive Layout: Ensures compatibility across devices.

# Tools and Technologies Used

MongoDB (Model Layer):  
MongoDB was chosen for its flexibility in handling semi-structured data. It integrates well with Python and Dash, allowing for efficient read/write operations.  
  
Dash Framework by Plotly (View + Controller):  
Dash is a Python framework for building interactive web applications. Components used include:  
- dash\_table.DataTable  
- dcc.Graph  
- dash\_leaflet  
- html.Div, dcc.Dropdown, dcc.RadioItems  
  
Python Libraries:  
- pandas  
- plotly.express  
- dash\_leaflet  
- pymongo  
- base64, os

# Steps Taken to Complete the Project

1. Setup MongoDB & CRUD Module  
2. Built Initial Dashboard Layout  
3. Integrated MongoDB Data  
4. Added Charting and Mapping  
5. Tested Interactions  
6. Polished UI

# Challenges and Solutions

- MongoDB Read Failures: Fixed using try/except and None handling.  
- TypeError with NoneType: Prevented with validation before DataFrame creation.  
- ObjectID in Dash Table: Dropped problematic \_id column.  
- Linking Map & Table: Used Dash callbacks to connect components.

# Reproducing This Project

1. Clone or download the project files.  
2. Install dependencies:  
 pip install dash dash-bootstrap-components pandas plotly pymongo dash-leaflet  
3. Ensure MongoDB is running and populated with animal data.  
4. Launch the Jupyter Notebook and run all cells.  
5. Open the dashboard in your browser using the URL provided by JupyterDash.