**README File**

**Project Description:**

The Grazioso Salvare project aims to develop a web-based dashboard for an animal shelter, providing functionalities for filtering and displaying animal data, visualizing data through charts, and displaying geographical locations of animals on a map.

**Required Functionality:**

The project requires the following functionalities:

* Display a dashboard interface with filtering options for animal data.
* Visualize animal data through interactive charts.
* Display geographical locations of animals on a map.
* Provide documentation detailing the project and instructions for reproducing it.

**Steps Taken to Complete the Project:**

* Set up the project environment, including installing necessary dependencies.
* Developed the dashboard layout using Dash components.
* Integrated data retrieval from MongoDB using a custom CRUD class.
* Implemented filtering options for animal data based on predefined criteria.
* Created interactive charts to visualize animal data using Plotly Express.
* Integrated Dash Leaflet to display geographical locations of animals on a map.
* Tested the dashboard for functionality and deployed it.

**Screenshots**

A screenshot of a computer

Description automatically generatedA pie chart with different colored circles

Description automatically generatedA white background with many lines

Description automatically generated with medium confidence

**Tools Used:**

**Dash Framework:** Used for building the web application's front-end interface and handling user interactions.

**Dash Leaflet**: Utilized for integrating interactive maps into the dashboard.

**Plotly Express:** Employed in creating interactive data visualizations and charts.

**Pandas:** Used for data manipulation and analysis.

**MongoDB:** Chosen as the model component for interfacing with Python due to its flexibility, scalability, and ease of integration with Python applications.

Rationale for Tool Selection:

**Dash Framework:** Chosen for its simplicity and ease of use in creating interactive web applications with Python.

**Dash Leaflet:** Selected for its seamless integration with Dash and ability to create interactive maps with minimal code.

**Plotly Express:** Chosen for its capability to generate interactive charts with just a few lines of code, enhancing data visualization in the dashboard.

**Pandas:** Leveraged for its powerful data manipulation capabilities, making it easier to work with and analyze tabular data.

**MongoDB:** Selected for its document-oriented data model, which aligns well with the JSON-like data structures commonly used in Python applications. Additionally, MongoDB's scalability and flexibility make it suitable for handling diverse data requirements in the animal shelter dashboard.

**Challenges Faced:**

Data Incorporation into Filtering Section: The greatest challenge was fully integrating the data retrieval from MongoDB into the filtering section. This involved mapping user-selected filter options to corresponding MongoDB queries and ensuring seamless data updates in the dashboard. I was unable to fully incorporate the full filtering on the grid display, but was able to have it appear on the pie chart. The selection was made possible, but encountered an issue with type error for the ‘animals’ collection, but I was unable to progress it.

**Resources:**

Dash Documentation

Dash Leaflet Documentation

Plotly Express Documentation

Pandas Documentation

MongoDB Documentation

Python Documentation