**CS 340 README 7-2 Project Two**

**About the Project/Project Title**

This project is a dashboard application developed for Grazioso Salvare, an international rescue-animal training company. The application interfaces with data from the Austin Animal Center Outcomes dataset and MongoDB to assist Grazioso Salvare in identifying and categorizing dogs available for training in search-and-rescue operations.

**Motivation**

Grazioso Salvare needed a software solution to simplify their process of identifying suitable dogs for training. The dashboard provides user-friendly interfaces and visualizations, enabling the company to explore available data and select dogs based on their training needs.

**Getting Started**

To get a local copy up and running, follow these steps:

1. **Install Python and necessary libraries:** Ensure Python is installed on your system. Install the required libraries by running the following command: pip install -r requirements.txt
2. Set up MongoDB: Ensure MongoDB is installed and running on your system or hosted elsewhere. The database should contain data from the Austin Animal Center Outcomes dataset.
3. Run the Dashboard: Open and execute the ProjectTwoDashboard.ipynb file using Jupyter Notebook or JupyterLab.

**Installation**

Ensure the following tools are installed:

* Python (version 3.9 or later)
* Libraries: **pandas**, **dash**, **plotly**, **dash-leaflet**, and **pymongo**
* MongoDB server running locally or accessible via credentials.

**Usage**

The dashboard provides interactive options to explore and filter data from the Austin Animal Center. It includes:

1. **Data Table:** Displays an unfiltered view of the dataset with sorting and filtering capabilities.
2. **Filters:** Dropdown menu to filter data by animal type.
3. **Charts:** Displays dynamic charts, including a pie chart showing breed distribution, and a geolocation chart visualizing animal data on a map

**Code Example**

*@app.callback(*

*Output('datatable-id', 'data'),*

*[Input('filter-type', 'value')]*

*)*

*def update\_dashboard(filter\_type):*

*"""Filters the data table with MongoDB queries."""*

*df\_filtered = pd.DataFrame.from\_records(db.read({'animal\_type': filter\_type})) if filter\_type else pd.DataFrame.from\_records(db.read({}))*

*return df\_filtered.to\_dict('records')*

**Tests**

To test the dashboard, run the Jupyter Notebook file and interact with its components. Ensure all widgets and interactions function as expected, especially filtering options and visualizations.

**Screenshots**

A screenshot of a map

Description automatically generated

**Contact**

Your name: Corvinna Curtis  
Email: Corvinna.curtis@snhu.edu