**Grazioso Salvare Dashboard Project Two - README**

**Created by:** Cassim Mabirizi  
**Client:** Grazioso Salvare  
**Course:** CS 340 Client/Server Development

**1. Project Functionality Overview**

This project is a full-stack web application built to help Grazioso Salvare identify suitable dogs for search-and-rescue training. The dashboard allows users to filter and view animal data from Austin-area shelters stored in a MongoDB database.

**Key Features:**

* Interactive filtering options:
  + Water Rescue
  + Mountain/Wilderness Rescue
  + Disaster/Tracking
  + Reset button
* Responsive data table
* Geolocation chart (Map view)
* Secondary chart (Breed Distribution - Pie Chart)
* Branding with Grazioso Salvare logo and developer name

**2. Tools and Rationale**

* **MongoDB**: Chosen for its flexibility in handling varied animal data and robust integration with Python.
* **Dash (by Plotly)**: Used to build an interactive web dashboard within a Python environment.
* **PyMongo**: Provides direct access to MongoDB from Python.
* **Pandas & Plotly Express**: For data handling and visualization.
* **Jupyter Notebook**: For organizing the project in an accessible, modular format.

**3. Steps Taken to Complete the Project**

1. Connected MongoDB and implemented full CRUD functionality.
2. Loaded and prepared the Austin Animal Center Outcomes dataset.
3. Built a responsive dashboard using Dash components.
4. Implemented filtering functionality for rescue type using query logic aligned with the client’s breed/age/sex criteria.
5. Designed interactive charts (Map + Pie Chart).
6. Branded the dashboard with logo and developer name.
7. Tested all features and captured screenshots for each filter state.

**4. Challenges and How They Were Overcome**

* **Challenge:** Dynamic filtering with MongoDB queries.
  + **Solution:** Used structured filter logic based on provided breed and age data, ensuring query accuracy with age\_upon\_outcome\_in\_weeks, breed, and sex\_upon\_outcome.
* **Challenge:** Map chart visualization performance with large datasets.
  + **Solution:** Optimized dataset filtering before map rendering.

**5. How to Reproduce the Project**

1. Install dependencies:

**pip install dash pandas pymongo plotly**

1. Ensure MongoDB is running locally and the aac database is populated.
2. Clone or unzip tearliehe project folder.
3. Open and run ProjectTwoDashboard.ipynb in Jupyter Notebook.
4. Adjust MongoDB login credentials in the AnimalShelter class if needed.
5. Use the dashboard’s radio/dropdown to explore rescue types.

**6. Dashboard Screenshot**

