**Grazioso Salvare Rescue Animal Dashboard**

**Description**

The Grazioso Salvare Rescue Animal Dashboard is a web application that provides an interactive interface for managing and visualizing data related to rescue animals. It includes features such as filtering data, displaying an interactive data table, and showing a pie chart depicting preferred breeds based on rescue type.

**Screenshots**

With Logo of GraziosoA close-up of a computer screen

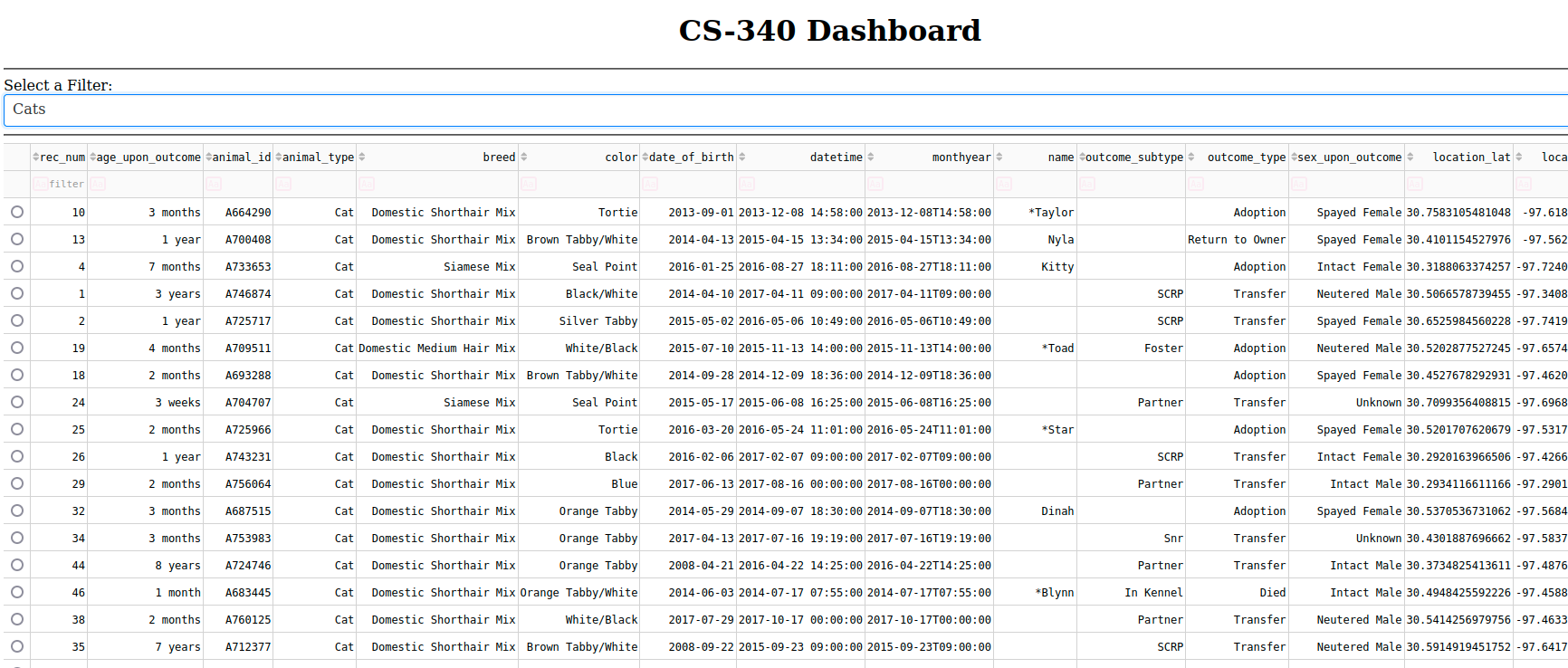
Description automatically generated

Filter Option with all animals, dogs & cats

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated



When clicked on the data it shows the location of the breed on map and graph provides info on all breedsA screenshot of a map

Description automatically generated

**Tools Used**

**Programming Languages and Libraries**

* Python
* Dash (a Python web application framework)
* Dash Leaflet (for interactive maps)
* Plotly Express (for data visualization)
* Pandas (for data manipulation)
* MongoDB (as the database)

**Rationale**

* **Python**: Chosen as the primary programming language for its simplicity, data analysis capabilities, and extensive libraries for web development.
* **Dash Framework**: Used to create the web application's user interface, allowing for easy integration of data visualization and interaction components.
* **Dash Leaflet**: Integrated for displaying interactive maps.
* **Plotly Express**: Utilized for creating dynamic charts and visualizations.
* **Pandas**: Empowered data manipulation and preprocessing before visualization.
* **MongoDB**: Selected as the database due to its flexibility, scalability, and compatibility with Python.

**MongoDB**

MongoDB was used as the model component of the development for several reasons:

* **Flexibility**: MongoDB is a NoSQL database that allows for flexible data storage, which is crucial when dealing with diverse data structures.
* **Scalability**: MongoDB scales horizontally, making it suitable for handling a growing dataset.
* **Compatibility with Python**: MongoDB can be easily interfaced with Python using libraries such as PyMongo, allowing for seamless integration into a Python-based web application.

**Dash Framework**

Dash is a Python web application framework that provides a clean separation between the view and controller components of a web application. It simplifies web development and offers interactive components, making it an ideal choice for building data-driven web applications like this dashboard.

**Resources and Dependencies**

* [JupyterDash](https://github.com/plotly/jupyter-dash): Used for creating a Jupyter Notebook-compatible Dash application.
* [Dash Leaflet](https://dash-leaflet.herokuapp.com/): Integrated for interactive mapping features.
* [Plotly Express](https://plotly.com/python/plotly-express/): Utilized for creating interactive charts and visualizations.
* [PyMongo](https://pymongo.readthedocs.io/en/stable/): A Python driver for MongoDB used to connect and interact with the MongoDB database.

**Steps to Reproduce**

Follow these steps to reproduce the Grazioso Salvare Rescue Animal Dashboard:

1. Clone the repository to your local machine:

bashCopy code

git clone https://github.com/your-username/rescue-animal-dashboard.git cd rescue-animal-dashboard

1. Create a virtual environment (optional but recommended):

bashCopy code

python -m venv venv source venv/bin/activate # On Windows, use venv\Scripts\activate

1. Install the required Python packages:

bashCopy code

pip install -r requirements.txt

1. Ensure you have MongoDB installed and running locally or provide connection details to a remote MongoDB server in the code.
2. Modify the code to connect to your MongoDB database and adjust any other configuration settings as needed.
3. Run the application:

bashCopy code

python app.py

1. Access the dashboard in your web browser at **http://localhost:8050**.
2. Interact with the dashboard, filter data, and explore the features.

**Challenges Faced**

Document any challenges you encountered during the development of the dashboard and explain how you overcame them. This can include technical issues, design decisions, or data-related challenges.

* **Challenge 1**: Connecting to MongoDB and managing data retrieval.
  + PyMongo established a connection to the MongoDB database and created functions to manage data retrieval and manipulation.
* **Challenge 2**: Designing an interactive user interface with Dash.
  + **Solution**: Dash documentation and examples helped in creating interactive components like data tables, dropdowns, and charts.
* **Challenge 3**: Data preprocessing and cleaning.
  + **P**andas for data manipulation, allowing us to clean and structure the data for presentation.