Grazioso Salvare Animal Shelter Dashboard

Project Overview

This project involves developing an interactive dashboard for **Grazioso Salvare**, a company that specializes in training rescue dogs for specific purposes such as disaster response, search and rescue, and law enforcement. The dashboard connects to a MongoDB database that contains animal shelter records, allowing users to search, filter, and visualize data related to the organization's mission.

The dashboard supports **CRUD operations** (Create, Read, Update, and Delete) and integrates with MongoDB using Python. It is designed to be both functional and user-friendly, offering a responsive interface through the **Dash framework**. The project demonstrates a complete model-view-controller (MVC) implementation, with MongoDB serving as the model, Dash as the view and controller, and Python providing the underlying logic and connectivity.

Screenshots and screencasts were captured during testing and deployment to verify that the dashboard achieves all required functionalities.

Required Functionality

The primary functionality of the dashboard includes:

- **Connecting to MongoDB:** Securely connecting to the database using the AnimalShelter class defined in `animal_shelter.py`.
- **CRUD Operations:** Enabling the creation, reading, updating, and deletion of animal records from the MongoDB database.
- **Filtering and Visualization:** Allowing users to apply filters based on animal attributes such as breed, age, and outcome type.
- **Data Visualization:** Generating graphs and tables to represent filtered datasets visually.
- **User Interface:** Providing an interactive interface through Dash components that supports responsive updates when users apply filters or modify data.

razioso Salvare Dashboard

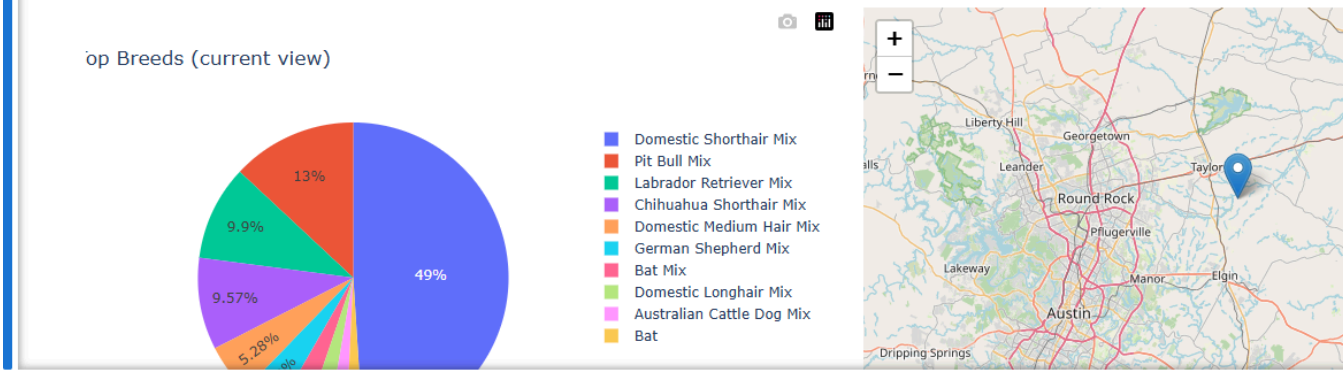
eloper: Donovan Taylor

Type Filter:

ter Rescue ○ Mountain or Wilderness Rescue ○ Disaster or Individual Tracking ◉ Reset (All Data)

rec_num	age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth
ilter da						
1	3 years	A746874	Cat	Domestic Shorthair Mix	Black/White	2014-04-10
9	3 years	A720214	Dog	Labrador Retriever Mix	Red/White	2013-02-04
10	3 months	A664290	Cat	Domestic Shorthair Mix	Tortie	2013-09-01

[ ]:



[ ]:

Tools and Technologies Used

1. Python

Used as the primary programming language due to its versatility and the availability of robust libraries for database interaction and web application development.

2. MongoDB

Serves as the **model** component in the MVC structure. MongoDB was chosen for its:

- Schema flexibility, which accommodates varying data attributes across animal records.
- Native support for **JSON-like documents**, enabling seamless integration with Python.
- High performance for **querying and filtering** operations.
- Ability to store large volumes of semi-structured data efficiently.

3. Dash Framework

Dash provides both the **view** and **controller** functionality for the web application. It allows for:

- The creation of interactive, data-driven web dashboards.
- Integration of HTML and CSS with Python back-end logic.
- Dynamic updates to visual components (tables, graphs, etc.) without page reloads.

4. Pandas

Utilized for managing and transforming data retrieved from MongoDB before visualizing it through Dash components.

5. Plotly

Used to create the interactive graphs and charts displayed in the dashboard.

Project Reproduction Steps

1. **Set Up MongoDB:**
    - Install MongoDB locally or connect to a MongoDB Atlas cluster.
    - Import the provided animal shelter dataset into a MongoDB database.

Install Required Dependencies:

```
pip install pymongo dash pandas plotly
```

2.
3. **Configure the Connection:**

Update the credentials in the `AnimalShelter` class within `animal_shelter.py`:

```
USER = "aacuser"

PASS = "StrongPassword123"
```

    -
4. **Run the Dashboard:**

Execute the `ProjectTwoDashboard.py` file:

```
python ProjectTwoDashboard.py
```

    -
    - Open the displayed URL in a web browser to interact with the dashboard.

Challenges and Solutions

Challenge	Solution

Establishing a stable MongoDB connection	Configured proper authentication and verified network access for MongoDB Atlas.
Handling data inconsistencies within the dataset	Utilized Pandas for data cleaning and filtering prior to visualization.
Implementing responsive Dash callbacks	Used Dash's callback decorators to synchronize filters, data tables, and graphs dynamically.
Managing CRUD operations without disrupting active sessions	Created separate database access functions in the AnimalShelter class to isolate transactions safely.

References and Resources

- MongoDB Documentation
- Dash by Plotly
- Pandas Documentation
- Plotly Documentation
- Python 3.10+ Official Documentation: https://docs.python.org/3/