Grazioso Salvare Dashboard - SNHU CS-340

This project is a full-stack application for Grazioso Salvare, a rescue-animal training company, built using MongoDB, Dash, Plotly, and other Python libraries. The dashboard allows users to interact with a dataset of animals available for rescue, providing various filtering options and visualizations.

Project Description

This dashboard allows users to:

View and filter animal records from the MongoDB database based on rescue categories like Water Rescue, Mountain/Wilderness Rescue, and Disaster Rescue.

Display a pie chart showing the distribution of animal breeds.

Map selected animals on an interactive map, showing their breed, age, and other details.

Requirements

Python Libraries

To run this project, you need to install the following Python libraries:

dash

dash\_core\_components

dash\_html\_components

dash\_table

plotly

dash\_leaflet

pandas

pymongo

bson

You can install all the required libraries with the following command:

pip install -r requirements.txt

Create a requirements.txt file containing the following dependencies:

dash==2.0.0  
dash-core-components==2.0.0  
dash-html-components==2.0.0  
dash-table==5.0.0  
plotly==5.0.0  
dash-leaflet==0.1.0  
pandas==1.3.0  
pymongo==3.11.0  
bson==0.5.10

MongoDB

You will also need a running instance of MongoDB. You can set up MongoDB locally or use a cloud service like MongoDB Atlas.

Project Setup

Set Up MongoDB:

Ensure MongoDB is running on your machine or connect to a MongoDB Atlas database.

Replace the MongoDB credentials (username, password) in the app.py file with your actual database details.

Logo:

The dashboard uses a logo image (Grazioso\_Salvare\_Logo.png). Replace the image path in the app.py file with the path to your own logo file.

Running the Dashboard:

Once you have installed the required libraries and set up MongoDB, run the following command to start the Dash application.

python app.py

Accessing the Dashboard:

Once the server is running, access the dashboard by opening a web browser and navigating to http://127.0.0.1:8050.

Features

Animal Data Table: The table displays animal records, including breed, age, sex, and other attributes. You can filter the records based on specific rescue categories.

Filter Options: Users can filter animal records by selecting different rescue categories such as:

Water Rescue (WR)

Mountain/Wilderness Rescue (MWR)

Disaster Rescue/Individual Tracking (DRIT)

Reset (RESET), which clears any filters applied

Pie Chart: A dynamic pie chart visualizes the distribution of animal breeds in the dataset based on selected records.

Interactive Map: The map displays markers for selected animals, showing their breed, name, sex, and age, as well as their geographic location (latitude/longitude).

Code Overview

app.py: The main entry point of the application. This file contains the layout, callbacks, and logic for rendering the dashboard and handling user interactions.

animal\_shelter.py: This module contains the AnimalShelter class, which handles interactions with the MongoDB database (CRUD operations). Make sure to update your database credentials here.

MongoDB: Data is fetched from a MongoDB collection containing animal records. The data includes fields like breed, sex, age, and geographic location.

Running the Application

Once you have the dependencies installed and MongoDB set up:

Clone the repository to your local machine.

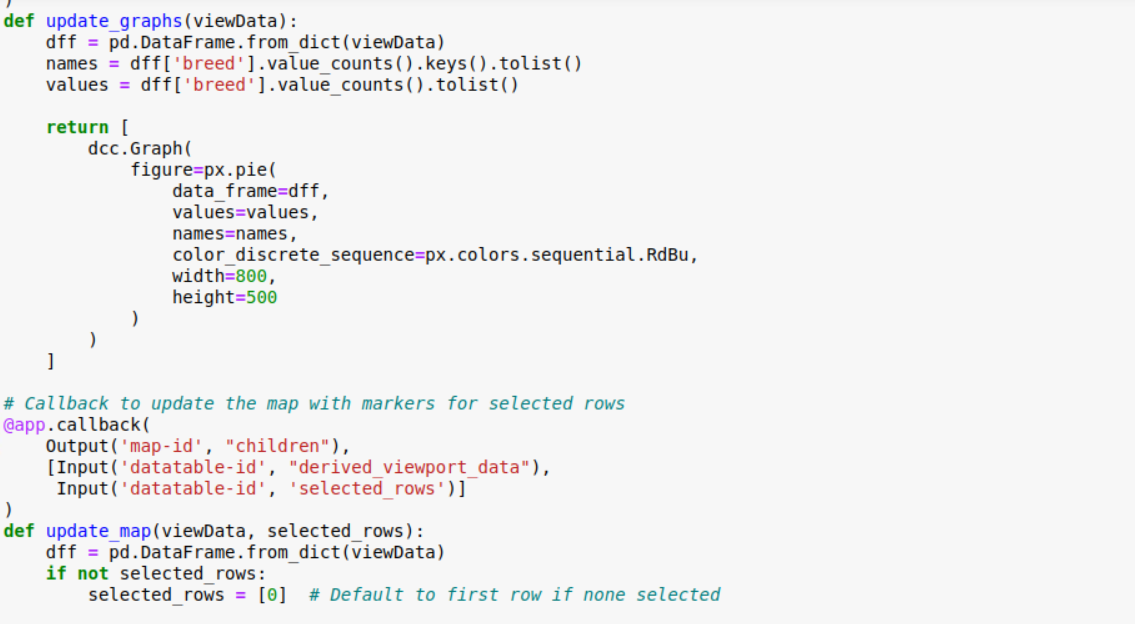
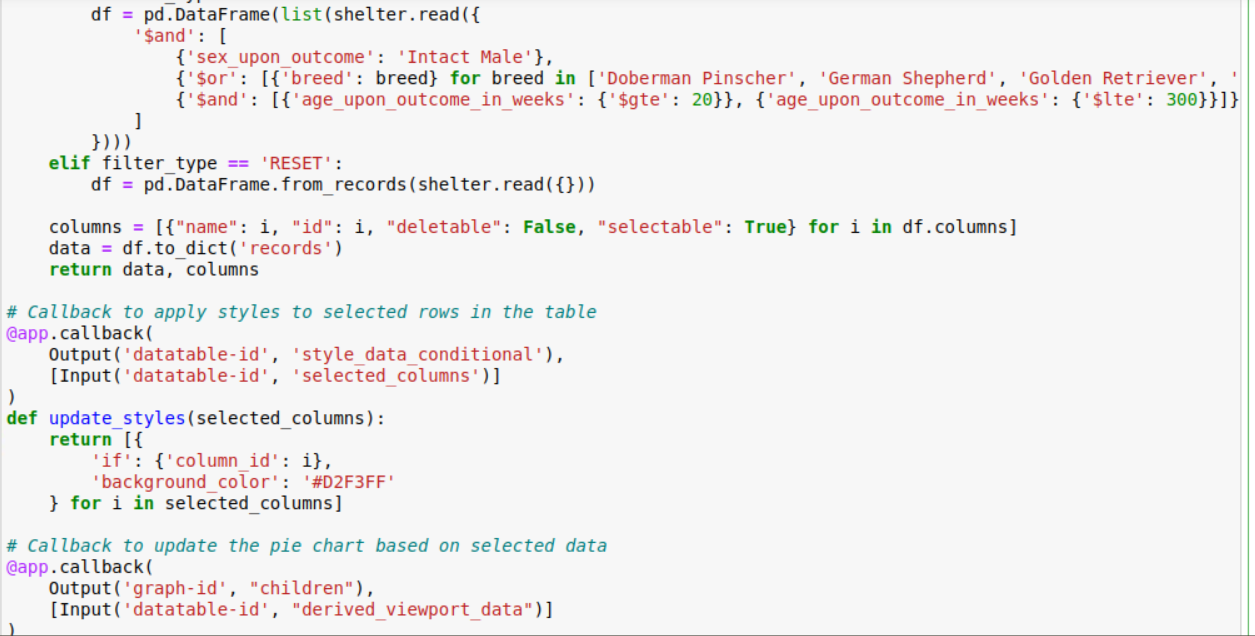
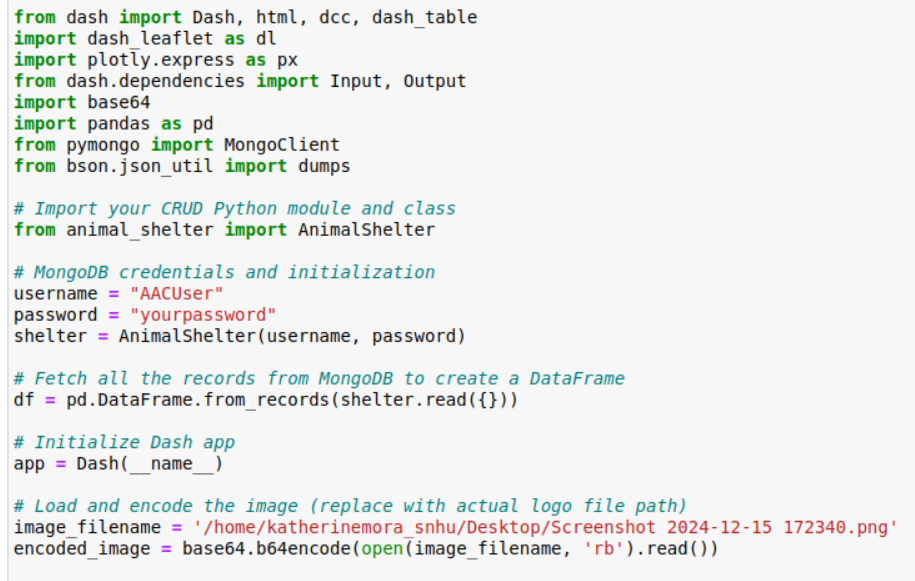
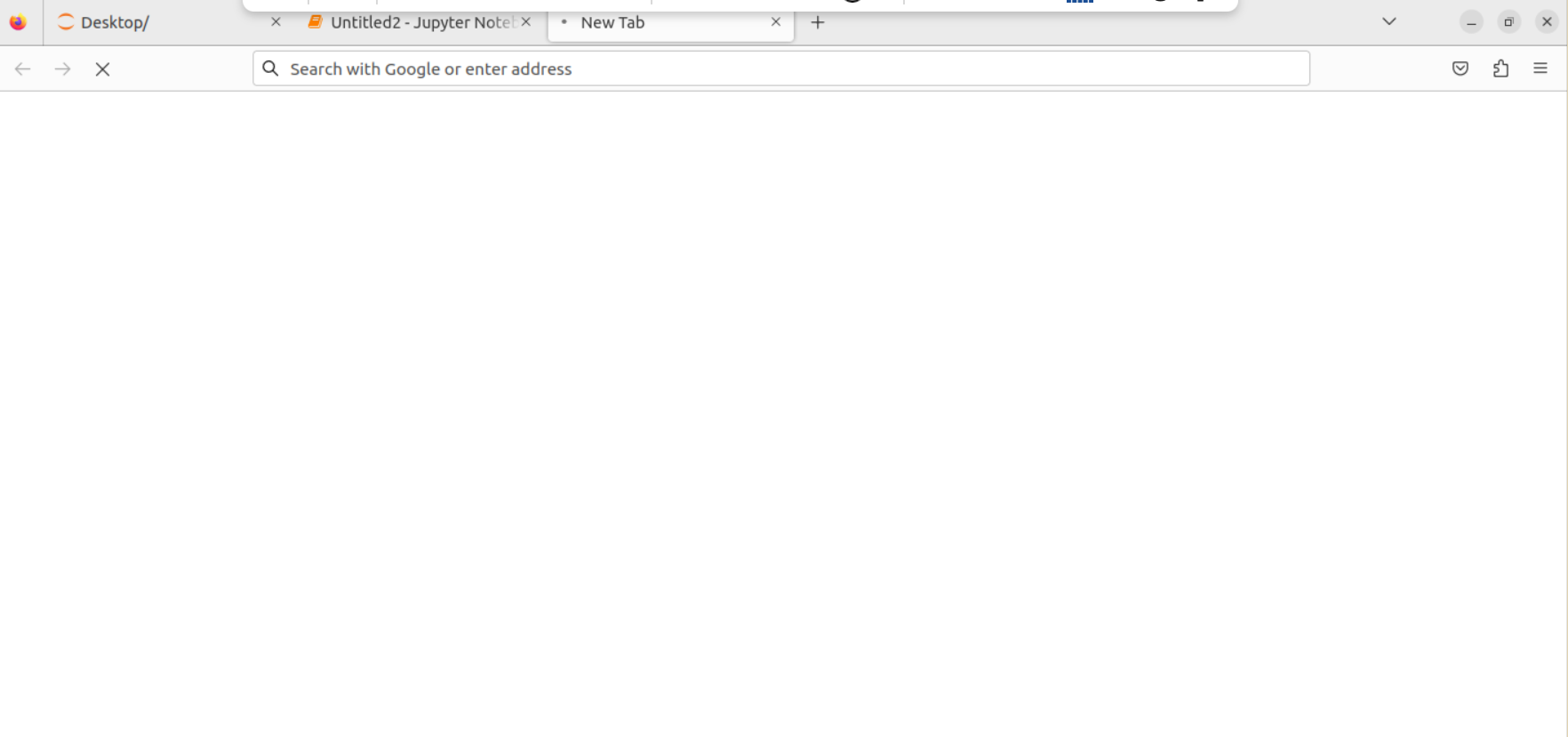
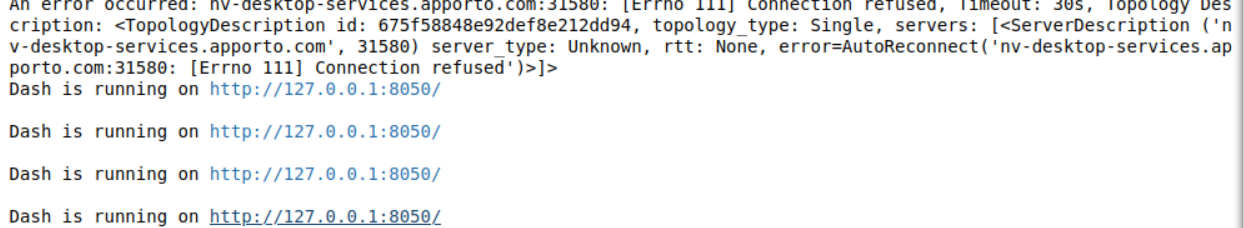
Install the required dependencies using pip install -r requirements.txt.

Update the MongoDB credentials in app.py.

Run the application using python app.py.

Visit http://127.0.0.1:8050 in your browser to view the dashboard.

Screenshots



Future Improvements

Add authentication and user roles to control access to the dashboard.

Implement additional charts and data visualizations to help with analysis.

Improve performance with data pagination or aggregation in MongoDB.