John Munguia

CS 340

December 22, 2024

Project Two: Grazioso Salvare Dashboard

This project is part of the CS 340 course and involves the development of a dashboard for Grazioso Salvare, an international rescue-animal training company. The dashboard allows users to filter and visualize data on animals suitable for various rescue operations. It integrates MongoDB as the database (model), Python as the controller, and Dash for the frontend (view).

Tools and Technologies

1. MongoDB: Used as the backend database for storing animal data.  
2. Python: Used for building the CRUD operations and integrating the database with the dashboard.  
3. Dash Framework: Provides an intuitive way to build interactive dashboards.  
4. Jupyter Dash: A version of Dash tailored for Jupyter Notebook environments.  
5. Dash Leaflet: Used for creating geolocation maps.  
6. Plotly: Used for data visualization (charts).

Installation Instructions

1. Install Python 3.9 or later.  
2. Install required libraries using the following command:  
 pip install pymongo dash jupyter-dash pandas plotly dash-leaflet matplotlib  
3. Clone the repository and ensure the following files are available:  
 - ProjectTwoDashboard.ipynb  
 - animal\_shelter.py  
 - assets/logo.png (Grazioso Salvare's logo)

Getting Started

1. Open the ProjectTwoDashboard.ipynb file in Jupyter Notebook.  
2. Ensure MongoDB is running and accessible with the correct credentials.  
3. Run the notebook to start the dashboard server.  
4. Open the dashboard URL (http://127.0.0.1:8050/) in your web browser.

Features

1. Interactive data table: Allows filtering, sorting, and pagination of animal data.  
2. Rescue filters: Enables users to filter animals based on rescue type (Water, Wilderness, Disaster).  
3. Geolocation map: Displays the location of selected animals.  
4. Dynamic charts: Provides insights into animal breeds and their distribution.

Challenges

Some challenges faced during the development included handling MongoDB ObjectIDs in the DataFrame and ensuring seamless integration between the backend and frontend.