CS 340 Project Two README

Southern New Hampshire University

October 22nd, 2024

Ryan A. Manchanthasouk

**Grazioso Salvare Animal Shelter Dashboard**

**Project Description**

This project has been developed for Grazioso Salvare to help identify potential rescue dogs from local animal shelters. I have created a web-based dashboard that connects to a MongoDB database containing animal shelter data from Austin, Texas. The dashboard allows users to filter and visualize data to find dogs suited for different types of rescue work.

**Functionality**

The dashboard provides several key features to help Grazioso Salvare staff efficiently identify suitable rescue dogs. Users can filter the data using radio buttons that correspond to different rescue types: Water Rescue, Mountain or Wilderness Rescue, and Disaster or Individual Tracking. A reset option is also available to return to the full dataset view.

The main component is an interactive data table that displays detailed information about each animal. I implemented sorting and filtering capabilities for each column to help users quickly find specific information. The table uses pagination to manage the large dataset effectively, showing 10 entries per page for optimal viewing.

To help visualize the data, I included two charts:

1. A geolocation chart that shows where each animal is located in the Austin area
2. A pie chart that displays the distribution of dog breeds in the current selection

**Screenshots**

I have included screenshots demonstrating the dashboard's functionality in different states:

1. The initial dashboard view showing all data
2. Results filtered for water rescue dogs
3. Results filtered for mountain rescue dogs
4. Results filtered for disaster rescue dogs
5. The reset state displaying all records
6. A screenshot of a computer

   Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**Tools and Technologies Used**

For this project, I carefully selected tools that would best serve Grazioso Salvare's needs. MongoDB was my choice for the database solution because of its flexible schema design and powerful query capabilities. I found its ability to handle complex filtering requirements particularly useful for this application. The native support for geospatial data made it perfect for implementing location-based features, while the PyMongo driver provided seamless Python integration.

I built the web application using the Dash framework, which allowed me to create an interactive dashboard using Python. Dash's integration with Plotly made it easy to create dynamic visualizations, and its reactive programming model helped me build a responsive user interface. The framework's pre-built components saved development time while ensuring a professional look and feel.

For data manipulation and analysis, I utilized several Python libraries:

* PyMongo for database interactions
* Pandas for data processing
* Plotly for interactive visualizations
* dash\_leaflet for the geolocation features

**Implementation Process**

My development process followed several key stages. I began by setting up the MongoDB database and implementing the basic CRUD operations. This involved creating the AnimalShelter class, which handles all database interactions. I paid special attention to error handling and input validation to ensure reliable operation.

Next, I developed the dashboard interface. I started with the basic layout and integrated the Grazioso Salvare branding. The interactive data table came next, followed by the filtering functionality. The final step was adding the visualization components and ensuring they updated properly based on user interactions.

Throughout the development, I maintained clean, well-documented code and followed Python best practices to ensure maintainability.

**Project Setup Instructions**

To set up this project, follow these steps:

1. First, install the required Python packages:

bash

Copy

pip install pymongo pandas dash plotly dash-leaflet

1. Ensure MongoDB is installed and running on your system. Create a user with appropriate permissions and import the animal shelter data.
2. Update the MongoDB connection details in the code:

* Replace username and password with your credentials
* Verify the correct database name and collection
* Check that the MongoDB port matches your setup

1. Place the Grazioso Salvare logo (logo.png) in the project directory.
2. Launch the Jupyter notebook and run the dashboard code.

**Challenges and Solutions**

During development, I encountered several challenges. One significant issue was handling inconsistent data formats in the shelter records. I solved this by implementing robust data cleaning functions in the CRUD operations, ensuring that all data was properly formatted before being displayed.

Query performance was another challenge, particularly with large datasets. I addressed this by adding appropriate MongoDB indexes and optimizing my database queries. The pagination in the data table also helped manage large result sets efficiently.

The geolocation feature presented its own challenges, particularly with missing or invalid coordinate data. I implemented thorough error checking and added fallback values to ensure the map would display properly even with incomplete data.

**Resources Used**

Throughout this project, I consulted several resources:

* MongoDB's official documentation (mongodb.com/docs)
* Dash framework documentation (dash.plotly.com)
* PyMongo documentation (pymongo.readthedocs.io)
* Various Stack Overflow discussions for specific implementation challenges

**Technical Requirements**

The dashboard requires:

* Python 3.7 or higher
* MongoDB 4.0 or higher
* A modern web browser
* Stable internet connection for map features