

snhu cs-340 Project 2

Shari Storlie



December 15, 2024

**Project Two README**

**Overview**

This project involves the development of an interactive dashboard for Grazioso Salvare, a rescue-animal training company. The dashboard dynamically interacts with a MongoDB database to retrieve and visualize data about animals based on specified rescue types. The dashboard includes interactive filter options, an updated data table, a geolocation chart, and a pie chart. These features enable the user to filter, view, and analyze animal data intuitively.

---

**Required Functionality**

The dashboard fulfills the following functionalities:

1. Interactive Filtering Options:

- Filters animal data based on rescue types:

- Water Rescue

- Mountain or Wilderness Rescue

- Disaster or Individual Tracking

- Reset (default state displaying all data)

2. Data Table:

- Displays filtered data dynamically in a user-friendly table.

3. Charts:

Geolocation Chart: Displays the location of the selected animal.

Pie Chart: Shows the distribution of breeds based on the selected data.

4. Grazioso Salvare Branding:

- Includes the Grazioso Salvare logo and the unique identifier of the developer.

**Screenshots**

1. Starting State:

(Include screenshot showing default dashboard layout with logo, filters, table, and charts.)

A screenshot of a computer

Description automatically generated

A screenshot of a map

Description automatically generated

2. Water Rescue Filter:

A screenshot of a computer

Description automatically generated

3. Mountain or Wilderness Rescue Filter:

A screenshot of a computer

Description automatically generated

4. Disaster or Individual Tracking Filter:

A screenshot of a computer

Description automatically generated

5. Reset Filter:

A screenshot of a computer

Description automatically generated

**Tools and Rationale**

1. MongoDB

Why MongoDB?

- Flexible schema design accommodates varied data types.

- Efficient querying capabilities for filtering data based on attributes such as ‘breed’ and `location`.

- Seamless integration with Python using the pymongo library.

2. Dash Framework

Why Dash?

- Simplifies the development of interactive web applications.

- Combines Python backend with HTML and JavaScript for front-end interactivity.

- Provides rich library components such as DataTable, charts, and maps for dynamic updates.

3. Tools Used

Python Libraries:

- dash for web application framework.

- pandas for data manipulation.

- plotly for creating charts.

- dash-leaflet for geolocation mapping.

- MongoDB Atlas: Hosted MongoDB database for reliable and scalable storage.

---

**Steps to Reproduce**

1. Set Up MongoDB:

- Create a MongoDB database with the `animals` collection.

- Insert sample data into the collection based on project requirements.

2. Set Up Python Environment:

- Install required libraries:

```bash

pip install dash pandas plotly pymongo dash-leaflet

```

3. Run the Application:

- Execute the ProjectTwoDashboard.ipynb file.

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

4. Test Functionality:

- Use filter options to verify data table and chart updates.

- Ensure the geolocation chart accurately displays the location of selected animals.

**Challenges and Solutions**

1. Missing Data Columns:

Challenge: The ‘\_id’ field in MongoDB caused errors in the DataTable.

Solution: Dropped the ‘\_id’ column during data transformation in Python.

2. File Not Found Errors:

Challenge: Grazioso Salvare’s logo was not found in the expected directory.

Solution: Uploaded the logo image to Jupiter and fixed the file path name.

3. Dynamic Updates:

Challenge: Ensuring smooth interaction between filters and charts.

Solution: Leveraged Dash callbacks to dynamically query MongoDB and update components.

By completing this project, Grazioso Salvare now has an intuitive tool to identify rescue dogs effectively, ensuring operational efficiency and success in their rescue missions.

**Resources:**

<https://www.mongodb.com/docs/>

<https://dash.plotly.com/>

<https://www.mongodb.com/docs/>