Vudeh Grazioso Salvare Dashboard Project

SNHU CS340

Project Overview

The Grazioso Salvare Dashboard project involves creating a fully functional MongoDB dashboard for Grazioso Salvare. The dashboard allows interaction with and visualization of the database containing animal shelter data. The project leverages the MVC design pattern, where MongoDB serves as the model, Dash components act as the views, and the CRUD Python module handles the controller logic.

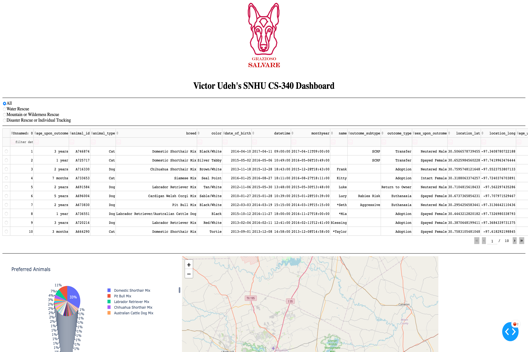
Required Functionality

The dashboard provides the following functionalities:

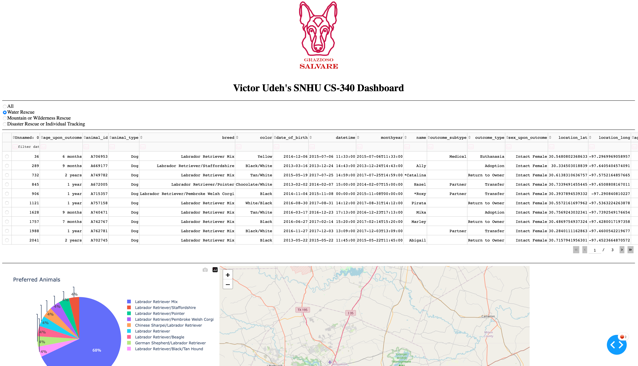
1. Unfiltered Data Table: Displays an unfiltered view of the Austin Animal Center Outcomes data set.
2. Filtering Options: Allows users to filter the data based on rescue types such as Water Rescue, Mountain or Wilderness Rescue, and Disaster or Individual Tracking.
3. Interactive Data Table: Responds dynamically to the filtering options.
4. Charts: Includes a geolocation chart and another chart of choice that respond to updates from the data table.

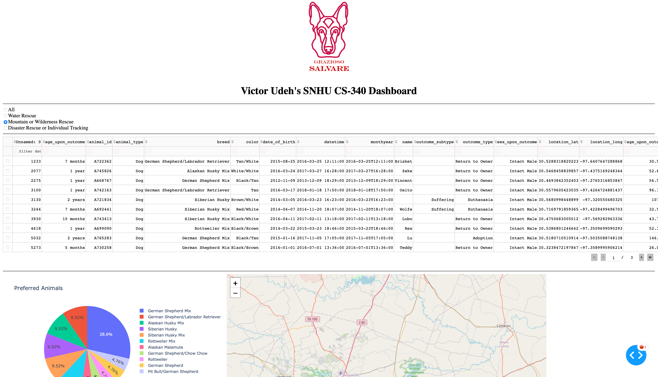
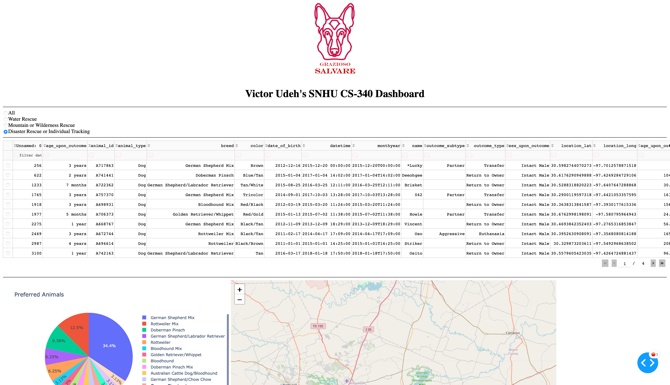
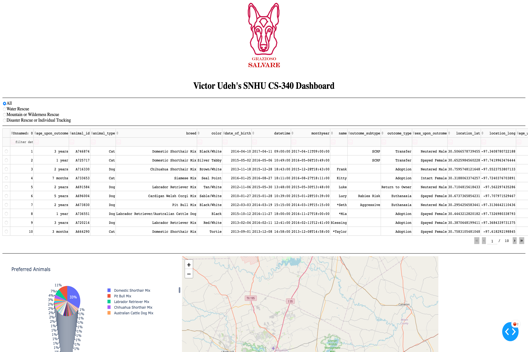
Screenshots of Functionality

1. Starting State of the Dashboard:



1. Filter: Water Rescue:



1. Filter: Mountain or Wilderness Rescue:  
     
   
2. Filter: Disaster or Individual Tracking:  
     
   
3. Reset State:  
   

Tools Used

1. MongoDB: Used as the model component for its flexibility, scalability, and ease of integration with Python.
2. Dash Framework: Provides the structure for building the web application, allowing for the creation of interactive web-based data visualizations.
3. Plotly: Utilized for creating dynamic charts.
4. Python: The main programming language for developing the dashboard and the CRUD operations.
5. Pandas: Used for data manipulation and analysis.
6. Matplotlib: Used for additional plotting capabilities.

Rationale for Tool Selection

* MongoDB: Chosen for its document-based storage, which is ideal for handling JSON-like documents and provides powerful querying capabilities. It allows seamless integration with Python using the pymongo library, making it easier to perform CRUD operations.
* Dash Framework: Selected for its simplicity and effectiveness in creating interactive web applications without requiring extensive front-end development knowledge. Dash abstracts away the complexities of web development, allowing developers to focus on the data and visualization aspects.
* Plotly: Preferred for its high-quality interactive visualizations that integrate seamlessly with Dash. Plotly provides a wide range of chart types and customization options, making it a versatile choice for creating dynamic visualizations.
* Python: The preferred programming language due to its extensive libraries and support for data science and web development. Python's ecosystem, including libraries like Pandas and Matplotlib, provides powerful tools for data manipulation and visualization.

Steps Taken to Complete the Project

1. Database Connection and CRUD Operations:
   * Established a connection to the MongoDB database using the CRUD Python module.
   * Verified the connection and retrieved data from the database.
2. Dashboard Layout and Components:
   * Designed the layout of the dashboard using Dash components.
   * Created the data table to display the unfiltered data.
   * Added interactive options (radio items) to filter the data.
3. Dynamic Data Table:
   * Implemented callbacks to update the data table based on the selected filters.
4. Charts and Visualizations:
   * Developed a geolocation chart and another chart to dynamically display data based on the filtered results.
5. Testing and Deployment:
   * Tested the dashboard to ensure all components worked as expected.
   * Captured screenshots to document the functionality.

Challenges Encountered and Solutions

1. Connection Issues with MongoDB:
   * Challenge: Encountered SSL handshake errors when connecting to MongoDB.
   * Solution: Ensured the correct MongoDB URI and required packages were installed.
2. Data Table Not Updating:
   * Challenge: The data table was not responding to filter changes.
   * Solution: Debugged the callback functions to ensure proper data retrieval and updates.
3. Interactive Map Not Displaying:
   * Challenge: The geolocation chart was not displaying the markers correctly.
   * Solution: Adjusted the data format and ensured the correct coordinates were passed to the map component.

Resources and References

* MongoDB Documentation: [MongoDB Documentation](https://docs.mongodb.com/)
* Dash by Plotly Documentation: [Dash Documentation](https://dash.plotly.com/)
* Pandas Documentation: Pandas Documentation
* Plotly Documentation: Plotly Documentation

Conclusion

The Grazioso Salvare Dashboard project demonstrates the successful creation of an interactive web application using MongoDB, Dash, and Plotly. The dashboard provides dynamic data visualization and filtering capabilities, enabling the client to effectively interact with their animal shelter data. The project showcases the integration of various tools and technologies to deliver a comprehensive solution tailored to the client's needs.