Project Two ReadMe

Nicholas Vito  
CS-340

10/20/2024

The required functionality of the project includes creating a dynamic, interactive dashboard that allows Grazioso Salvare to filter and visualize animal shelter data to identify dogs suitable for search-and-rescue training. The dashboard features interactive filters for selecting animals based on rescue types (Water Rescue, Mountain or Wilderness Rescue, Disaster or Individual Tracking), a data table that dynamically updates based on the filters, a pie chart showing breed distribution with percentages displayed inside the slices, and a geolocation map showing the animals' locations. The successful deployment of the dashboard can be demonstrated through screenshots or screencasts that show the working filters, updated data table, pie chart, and the map component.

The project utilized MongoDB as the model component due to its flexibility in managing unstructured data, making it ideal for the animal shelter dataset with its varied fields. MongoDB's seamless integration with Python through the PyMongo library enabled efficient data querying and manipulation. For the view and controller structure, the Dash framework was chosen as it allows for building interactive web applications in Python, combining user interface and data management in one cohesive structure. Dash's integration with Plotly for visualizations, such as pie charts and data tables, made it suitable for creating dynamic and responsive data-driven applications. Python's simplicity and readability made it an ideal programming language for this project, allowing for rapid development and ease of maintenance while leveraging powerful libraries like Dash and PyMongo.

The project began by setting up a MongoDB database and importing the animal shelter dataset into a collection. Next, a Python CRUD class was developed using PyMongo to interface with the database and retrieve the data for the dashboard. The Dash framework was then used to create the web application, incorporating interactive filters, a data table, a pie chart, and a map for visualizing animal location. The data filters were configured to dynamically update the charts and table based on user inputs. Finally, the application was tested and debugged to ensure proper functionality and responsiveness.

A major challenge I faced during the development of the dashboard was getting all the components, particularly the map, to function correctly when I integrated the data table. Initially, the map wouldn’t display at all once the table was added, and I struggled to figure out why it wasn’t rendering properly. I also encountered issues with the Apporto server not connecting, even though it indicated a successful connection, which delayed progress. Resolving these problems required a combination of debugging the callbacks in the Dash framework and troubleshooting the server connection to ensure all elements of the dashboard worked together seamlessly. I realized that if I restarted apporto that the dashboard would work properly if it ever ran into that error.

This shows unique signature “created by N Vito” Also has the Companies Logo which if clicked brings us to [www.snhu.edu](http://www.snhu.edu) the companies preferred website.

A white paper with black text and black text

Description automatically generated

As you select different filters for the data the table itself updates and the pie chart shown changes

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

As you select different animals the location is updated on the GeoLocation map

A map with a blue point on it

Description automatically generatedA map with a location pin

Description automatically generated