**[Project Two] README: Grazioso Salvare Rescue Dog Dashboard**

Mario Frederick

Southern New Hampshire University

CS-340

10/25/25

About the Project

This project entails the development of a dynamic, interactive dashboard utilizing Python, Dash, Plotly, Dash Leaflet, and MongoDB. The primary objective of the dashboard is to facilitate Grazioso Salvare in the identification and monitoring of shelter dogs that are most suitable for search-and-rescue training missions. By establishing a direct connection to a MongoDB database, the dashboard enables real-time data filtering and visualization, encompassing variables such as breed, age, outcome, and geographic location.

Project Purpose

Grazioso Salvare, a dog rescue organization, needed a way to efficiently search and evaluate animals that are ideal candidates for rescue missions such as Water Rescue, Mountain or Wilderness Rescue, and Disaster or Individual Tracking. This project represents the culmination of efforts to develop a complete data dashboard that helps support decision-making through:

- Interactive breed filtering via radio buttons.

- Real-time animal data viewing in an interactive data table.

- Pie chart visualizing preferred breeds.

- Map plotting the selected animal’s last known location.

Tools and Libraries

The following tools and libraries were used in this project:

|  |  |
| --- | --- |
| Tool/Library | Purpose |
| Python | Backend development and data handling |
| Dash | Building the web application interface |
| Leaflet | Embedding interactive geolocation maps |
| Plotly Express | Chart and data visualization |
| Pandas | Data manipulation and table formatting |
| MongoDB | NoSQL database holding animal records |
| PyMongo | MongoDB driver to connect Python to the database |

Overview of MongoDB Selection and Application

MongoDB was selected as the database solution due to its capacity to store flexible, schema-less documents in a JSON-like format. This characteristic facilitates efficient integration with Python dictionaries and allows for straightforward modifications throughout the development process. Furthermore, MongoDB's robust querying capabilities position it as an ideal choice for managing the application’s filtering logic, which necessitates the return of subsets of animals based on age and breed.

For the front end, Dash was chosen because of its ability to create sophisticated web applications using Python. It supports user interactivity through callback functions that respond to input changes, such as filtering options and table row selections. The Dash Leaflet component was implemented to visualize the geographic locations of animals, enabling users to view a pin on the map when selecting a record from the data table.

The application interface includes the following features:

* A radio button filter for toggling between various rescue types (Water Rescue, Mountain or Wilderness Rescue, Disaster or Individual Tracking, and Reset).
* An interactive DataTable that retrieves live data from MongoDB.
* A pie chart that visualizes breed distribution.
* A dynamic map that updates in response to selected animal records.

Callback functions are employed to connect these components, which ensures real-time updates based on user-actions.

Project Steps and Development Process

The project development adhered to the following steps:

1. Imported and processed the Austin Animal Center CSV file.
2. Established the MongoDB database and configured user access authentication.
3. Developed a Python module to facilitate basic CRUD (Create, Read, Update, Delete) operations within the database.
4. Constructed the initial Dash layout, which integrated a DataTable, radio items, a pie chart, and the mapping component.
5. Connected all components via callback functions to ensure interactivity.
6. Conducted functionality testing using JupyterDash to verify that all components updated correctly in response to user selections.
7. Resolved various issues, including an obscured map resulting from missing or invalid coordinates and width restrictions.
8. Finalized the layout by incorporating the Grazioso Salvare logo and author information.
9. Captured screenshots of the operational dashboard for documentation purposes.

Resources Utilized During Development

* [Using Jupyter Labs in Codio (Virtual Lab) Tutorial](https://learn.snhu.edu/content/enforced/2019710-CS-340-10388.202581-1/course_documents/CS%20340%20Using%20Jupyter%20Labs%20in%20Codio%20(Virtual%20Lab)%20Tutorial.pdf?isCourseFile=true&ou=2019710)
* [Mongo in Codio (Virtual Lab) Tutorial](https://learn.snhu.edu/content/enforced/2019710-CS-340-10388.202581-1/course_documents/CS%20340%20Mongo%20in%20Codio%20(Virtual%20Lab)%20Tutorial.pdf?isCourseFile=true&ou=2019710)
* [Head First Python](https://go.oreilly.com/SNHU/library/view/head-first-python/9781491919521/)
* [Style Guide for Python Code](https://www.python.org/dev/peps/pep-0008/)
* [Make a README](https://www.makeareadme.com/)

Project Functionality

The primary functionalities of this dashboard encompass:

* A fully interactive Data Table that displays all pertinent animal data retrieved from the MongoDB database.
* A filtering system utilizing radio buttons, allowing users to refine their search for dogs based on rescue type (e.g., Water Rescue, Disaster Tracking).
* A pie chart that dynamically updates to illustrate breed distribution based on selected criteria.
* An interactive Dash Leaflet map that displays the geolocation of selected animals.
* Backend CRUD operations facilitated through a custom Python module that interfaces with MongoDB utilizing PyMongo.

Successful Connection to Data Table and Filter Controls MongoDB

A screenshot of a computer

AI-generated content may be incorrect.

DataTable ScreenshotA screenshot of a computer

AI-generated content may be incorrect.

Figure 1: Starting State of Dashboard

A screenshot of a computer

AI-generated content may be incorrect.

Figure 2: Default Dashboard View With Reset Filter Selected

A screenshot of a computer

AI-generated content may be incorrect.

Figure 3: Dashboard With Water Rescue Filter Applied

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4: Dashboard With Mountain or Wilderness Rescue Filter Applied

A screenshot of a computer

AI-generated content may be incorrect.

Figure 5: Dashboard With Disaster or Individual Tracking Filter Applied

A screenshot of a computer

AI-generated content may be incorrect.

Interactive Map with Popups

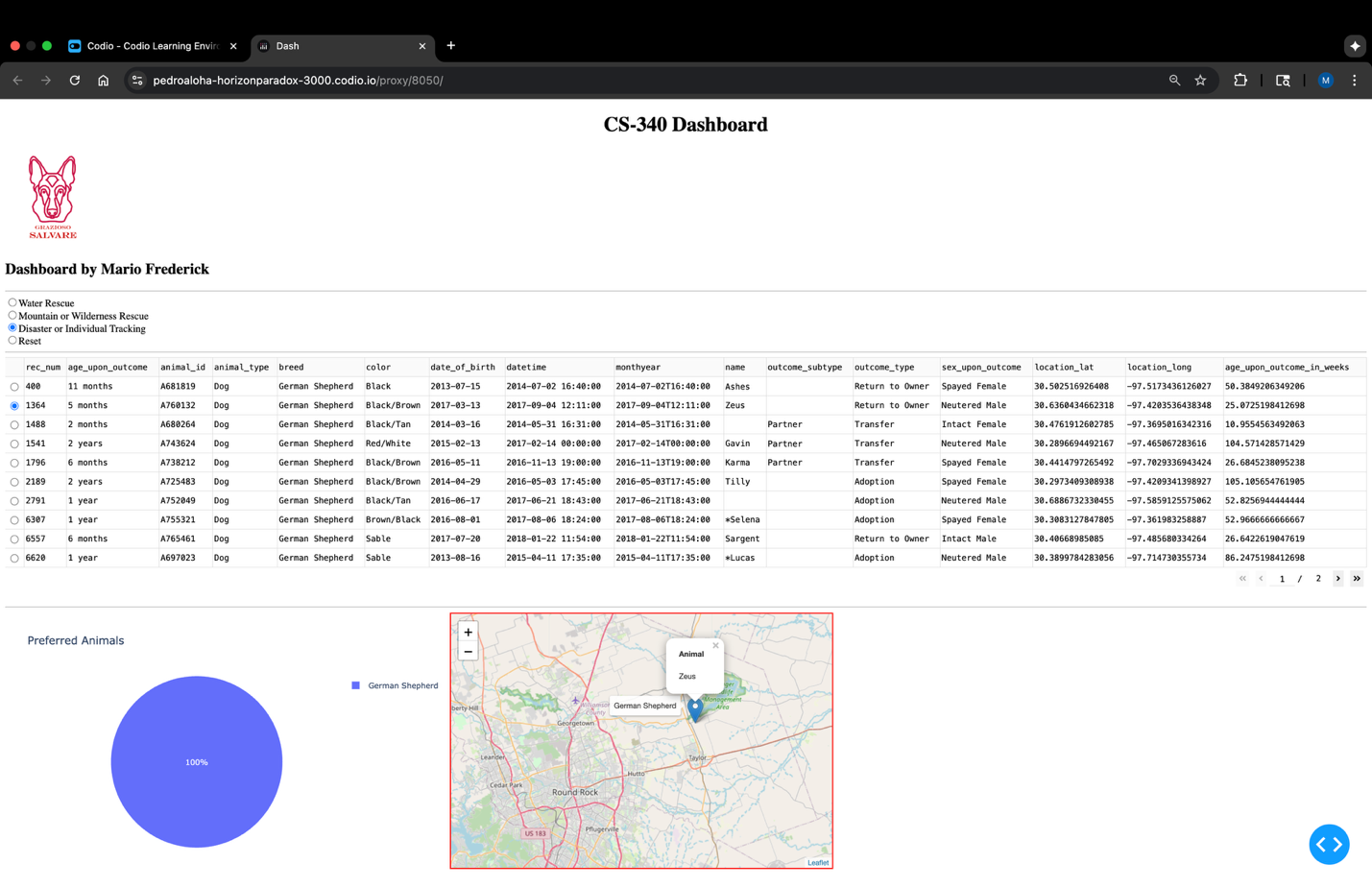
When a user selects an animal record from the Data Table, the map displays:

• A marker pinned to the animal’s location\_lat and location\_long.

• A Tooltip showing the animal’s breed.

• A Popup showing:

* Animal name
* A header labeled "Animal"



The dashboard has been fully optimized to meet the operational requirements of Grazioso Salvare by facilitating data visualization, tracking animal outcomes, and guiding informed decisions regarding the suitability of dogs for various types of training. Employing Dash, MongoDB, and interactive visualizations, this application provides a real-time, user-centric solution to a critical operational task.