2024

Project 2- Gonzalo Patino- CS340- Client and Server Development- Computer Science Department

FINAL FULL STACK PROJECT GRAZIOSO SALVARE

DEVELOPER GONZALO PATINO

Contents

Introduction

Grazioso Salvare, a leading dog rescue organization, required the development of an interactive and data-driven dashboard to improve decision-making and operational efficiency. This project addresses their need for a centralized, easy-to-use system that enables the staff to manage, visualize, and filter animal shelter data. By aligning the design with business requirements, this dashboard enhances their ability to deploy rescue dogs effectively for specific operations, such as water, mountain, or disaster rescues.

Business Requirements Documentation (BRD)

The system is designed based on the following **business requirements**:

1. **Data Management**: Enable easy access to animal shelter data, including breed, sex, age, and geolocation.
2. **Operational Efficiency**: Provide tools for filtering and selecting animals that meet specific rescue requirements (e.g., water rescue, mountain rescue).
3. **Usability and Maintainability**: Ensure that the dashboard is simple enough for non-technical staff to use and maintain.
4. **Real-Time Insights**: Dynamically visualize data updates across maps and charts as filters are applied.

Functional Requirements

1. **Interactive Filters**: Implement radio buttons for rescue types (Water, Mountain, Disaster, or Reset) to dynamically update the table, pie chart, and map.
2. **Dynamic Data Table**: Display animal shelter data with sortable and filterable columns, allowing single-row selection to update the map and chart.
3. **Pie Chart Visualization**: Generate a real-time pie chart visualizing breed distribution based on the selected filter.
4. **Geolocation Mapping**: Display the location of the selected animal record on an interactive map and dynamically update based on user input.

5. **Fallback Handling**: Handle missing or incomplete data gracefully (e.g., blank names display "No name found").
6. **Database Integration**: Use MongoDB as the backend database to store and retrieve animal data efficiently.

Non-Functional Requirements

1. **Performance**: Handle large datasets (10,000+ records) without noticeable delays in filtering or visualization.
2. **Scalability**: Support future expansion with additional records or features.
3. **Ease of Use**: Ensure the dashboard is intuitive and requires minimal training for staff to operate effectively.
4. **Maintainability**: Ensure the codebase is well-documented for future updates or extensions.
5. **Cross-Platform Accessibility**: Ensure the dashboard runs on modern web browsers.

Technologies Used

Architecture: Model-View-Controller (MVC)

The dashboard is built using the **MVC architecture**, where each component serves a distinct purpose:

- **Model**: MongoDB serves as the backend for storing and querying animal shelter data.
- **View**: Dash framework provides the interface for users to interact with the data.
- **Controller**: Dash callbacks handle user input, retrieve data, and update the view dynamically.

Tech Stack

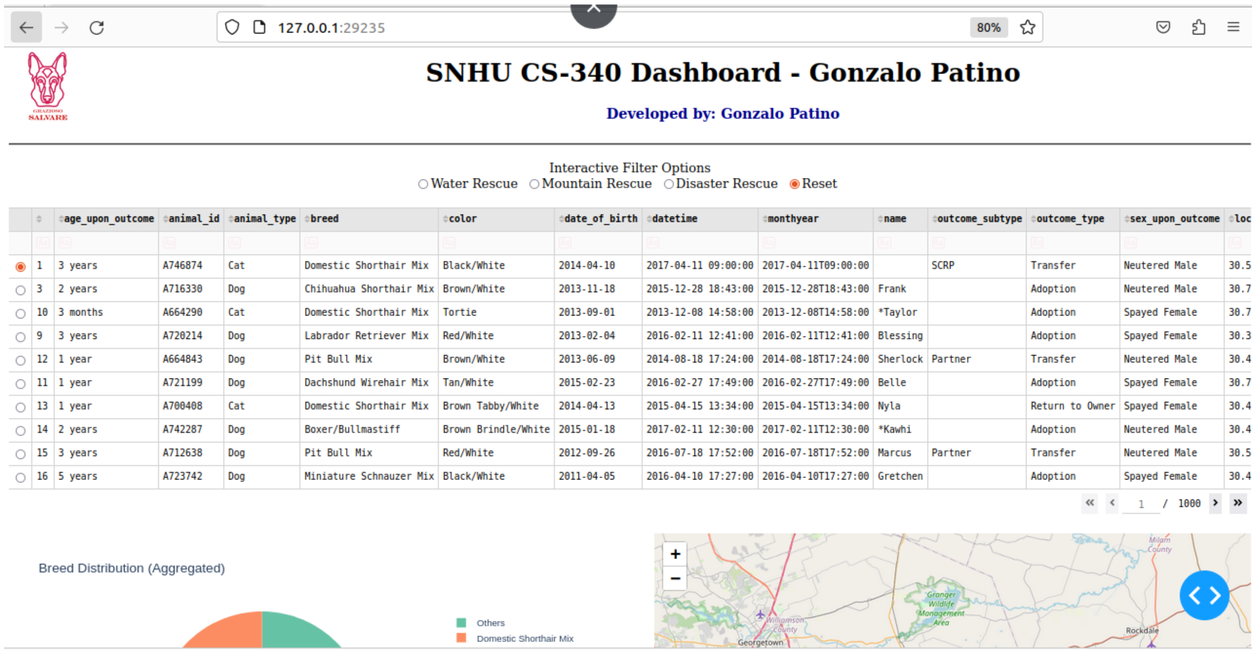
1. **Backend**:
   - **Python**: Handles backend logic and integration with MongoDB.
   - **MongoDB**: A NoSQL database for storing animal data with varied attributes.
   - **PyMongo**: Python library for interacting with MongoDB.
2. **Frontend**:
   - **Dash**: A Python framework for building interactive dashboards.
   - **Dash Leaflet:** Integrates geolocation mapping functionality into the dashboard.
   - **Plotly**: Used for creating dynamic pie chart visualizations.
3. **Middleware**:
   - **Dash Callbacks**: Implements the logic for updating the dashboard based on user interaction.
4. **Development Environment**:
   - **Jupyter Dash**: Used for developing and testing the application.

Why These Technologies Were Used

1. **MongoDB**:
   - Flexible schema for handling diverse animal attributes.
   - Efficient querying capabilities for large datasets.
   - Seamless integration with Python via PyMongo.
2. **Dash Framework**:
   - Combines the controller and view components, making it easy to create a fully functional web app.
   - Built-in support for dynamic updates to charts and tables through callbacks.
   - Extensibility for incorporating advanced visualizations like maps and charts.

3. **Dash Leaflet and Plotly**:
   - Dash Leaflet enables embedding geolocation data in the application.
   - Plotly simplifies creating interactive pie charts and visualizations.

Part 1: Dashboard Screenshots (Project 2 deliverables)



SNHU CS-340 Dashboard - Gonzalo Patino

Developed by: Gonzalo Patino

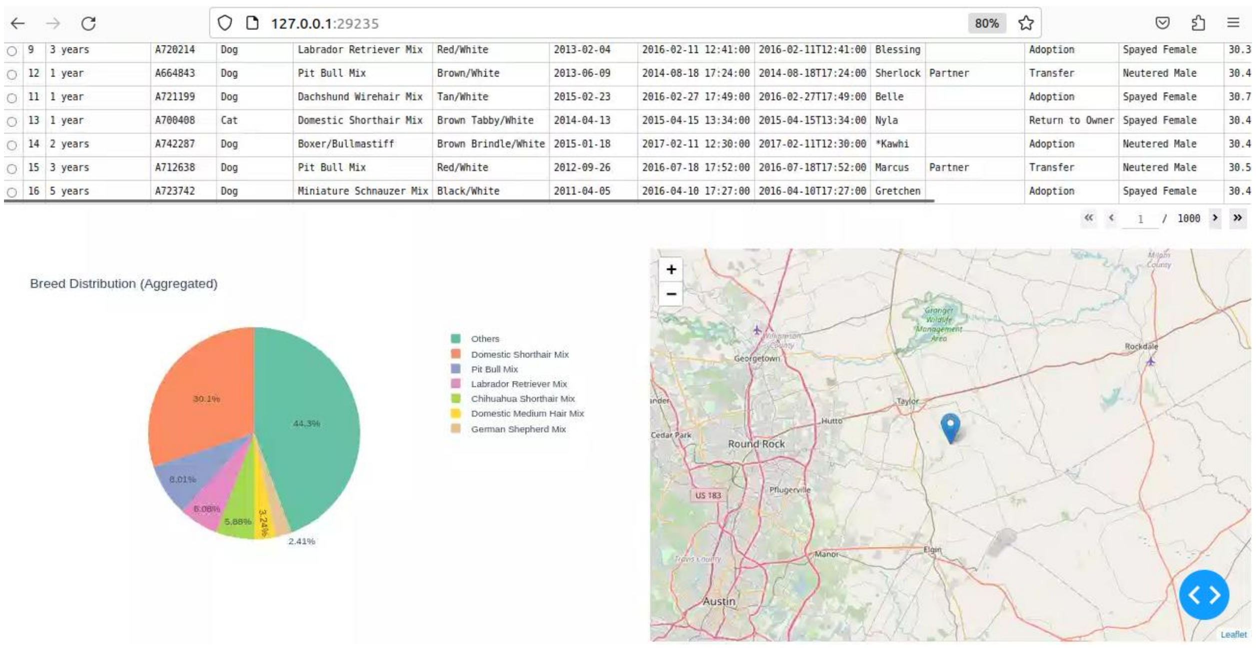
Interactive Filter Options

Water Rescue Mountain Rescue Disaster Rescue [✓ Reset]

		age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth	datetime	monthyear	name	outcome_subtype	outcome_type	sex_upon_outcome	loc
[✓]	1	3 years	A746874	Cat	Domestic Shorthair Mix	Black/White	2014-04-10	2017-04-11 09:00:00	2017-04-11T09:00:00		SCRP	Transfer	Neutered Male	30.5
	3	2 years	A716330	Dog	Chihuahua Shorthair Mix	Brown/White	2013-11-18	2015-12-28 18:43:00	2015-12-28T18:43:00	Frank		Adoption	Neutered Male	30.7
	10	3 months	A664290	Cat	Domestic Shorthair Mix	Tortie	2013-09-01	2013-12-08 14:58:00	2013-12-08T14:58:00	*Taylor		Adoption	Spayed Female	30.7
	9	3 years	A720214	Dog	Labrador Retriever Mix	Red/White	2013-02-04	2016-02-11 12:41:00	2016-02-11T12:41:00	Blessing		Adoption	Spayed Female	30.3
	12	1 year	A664843	Dog	Pit Bull Mix	Brown/White	2013-06-09	2014-08-18 17:24:00	2014-08-18T17:24:00	Sherlock	Partner	Transfer	Neutered Male	30.4
	11	1 year	A721199	Dog	Dachshund Wirehair Mix	Tan/White	2015-02-23	2016-02-27 17:49:00	2016-02-27T17:49:00	Belle		Adoption	Spayed Female	30.7
	13	1 year	A700408	Cat	Domestic Shorthair Mix	Brown Tabby/White	2014-04-13	2015-04-15 13:34:00	2015-04-15T13:34:00	Nyla		Return to Owner	Spayed Female	30.4
	14	2 years	A742287	Dog	Boxer/Bullmastiff	Brown Brindle/White	2015-01-18	2017-02-11 12:30:00	2017-02-11T12:30:00	*Kawhi		Adoption	Neutered Male	30.4
	15	3 years	A712638	Dog	Pit Bull Mix	Red/White	2012-09-26	2016-07-18 17:52:00	2016-07-18T17:52:00	Marcus	Partner	Transfer	Neutered Male	30.5
	16	5 years	A723742	Dog	Miniature Schnauzer Mix	Black/White	2011-04-05	2016-04-10 17:27:00	2016-04-10T17:27:00	Gretchen		Adoption	Spayed Female	30.4

« ‹ 1 / 1000 › »

Figure 1. Initial state part 1 out of 2. Please note there are left and right arrows for more records.

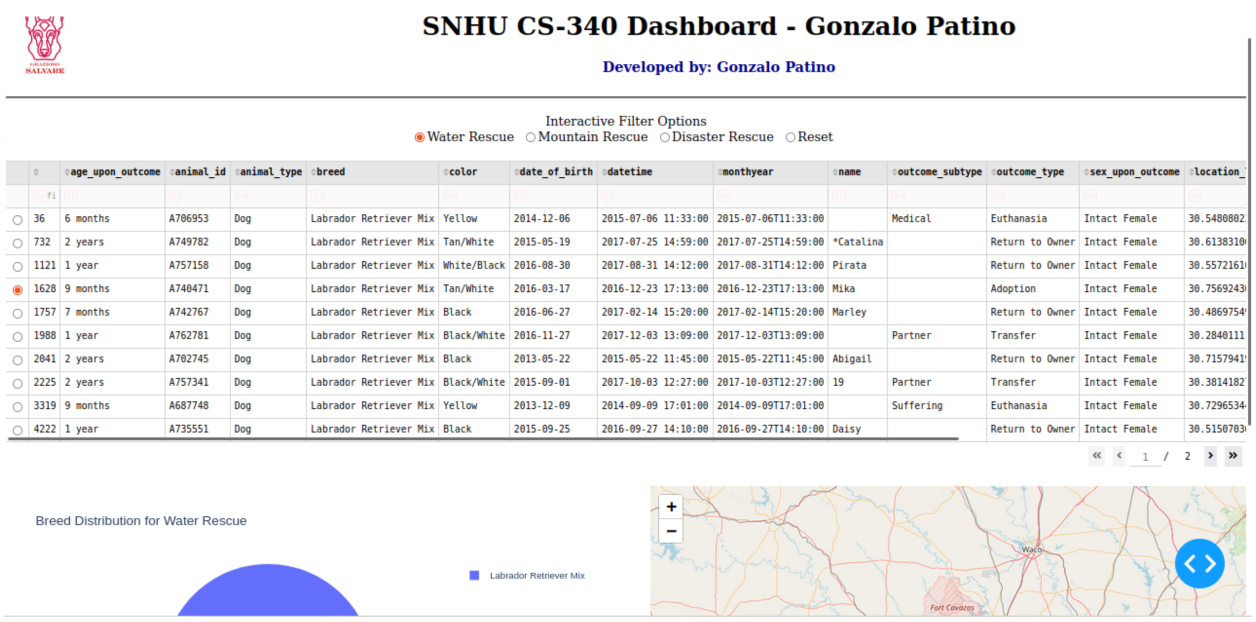


	9	3 years	A720214	Dog	Labrador Retriever Mix	Red/White	2013-02-04	2016-02-11 12:41:00	2016-02-11T12:41:00	Blessing		Adoption	Spayed Female	30.3
	12	1 year	A664843	Dog	Pit Bull Mix	Brown/White	2013-06-09	2014-08-18 17:24:00	2014-08-18T17:24:00	Sherlock	Partner	Transfer	Neutered Male	30.4
	11	1 year	A721199	Dog	Dachshund Wirehair Mix	Tan/White	2015-02-23	2016-02-27 17:49:00	2016-02-27T17:49:00	Belle		Adoption	Spayed Female	30.7
	13	1 year	A700408	Cat	Domestic Shorthair Mix	Brown Tabby/White	2014-04-13	2015-04-15 13:34:00	2015-04-15T13:34:00	Nyla		Return to Owner	Spayed Female	30.4
	14	2 years	A742287	Dog	Boxer/Bullmastiff	Brown Brindle/White	2015-01-18	2017-02-11 12:30:00	2017-02-11T12:30:00	*Kawhi		Adoption	Neutered Male	30.4
	15	3 years	A712638	Dog	Pit Bull Mix	Red/White	2012-09-26	2016-07-18 17:52:00	2016-07-18T17:52:00	Marcus	Partner	Transfer	Neutered Male	30.5
	16	5 years	A723742	Dog	Miniature Schnauzer Mix	Black/White	2011-04-05	2016-04-10 17:27:00	2016-04-10T17:27:00	Gretchen		Adoption	Spayed Female	30.4

« ‹ 1 / 1000 › »

Figure 2.Initial state part 2 out of 2. Please note there are left and right arrows for more records.

Key considerations: The table has a horizontal bar to scroll for more columns



SNHU CS-340 Dashboard - Gonzalo Patino

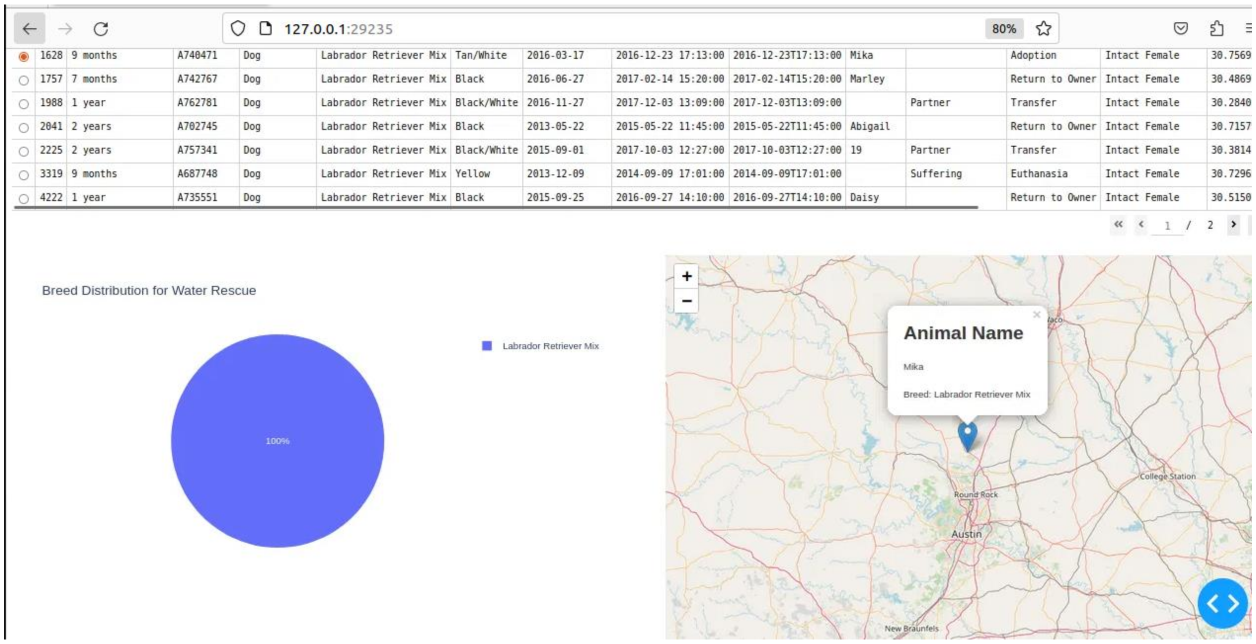
Developed by: Gonzalo Patino

Interactive Filter Options
[✓ Water Rescue] Mountain Rescue Disaster Rescue Reset

		age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth	datetime	monthyear	name	outcome_subtype	outcome_type	sex_upon_outcome	location_
○	36	6 months	A706953	Dog	Labrador Retriever Mix	Yellow	2014-12-06	2015-07-06 11:33:00	2015-07-06T11:33:00		Medical	Euthanasia	Intact Female	30.5480802
○	732	2 years	A749782	Dog	Labrador Retriever Mix	Tan/White	2015-05-19	2017-07-25 14:59:00	2017-07-25T14:59:00	*Catalina		Return to Owner	Intact Female	30.6138310
○	1121	1 year	A757158	Dog	Labrador Retriever Mix	White/Black	2016-08-30	2017-08-31 14:12:00	2017-08-31T14:12:00	Pirata		Return to Owner	Intact Female	30.5572161
●	1628	9 months	A740471	Dog	Labrador Retriever Mix	Tan/White	2016-03-17	2016-12-23 17:13:00	2016-12-23T17:13:00	Mika		Adoption	Intact Female	30.7569243
○	1757	7 months	A742767	Dog	Labrador Retriever Mix	Black	2016-06-27	2017-02-14 15:20:00	2017-02-14T15:20:00	Marley		Return to Owner	Intact Female	30.4869754
○	1988	1 year	A762781	Dog	Labrador Retriever Mix	Black/White	2016-11-27	2017-12-03 13:09:00	2017-12-03T13:09:00		Partner	Transfer	Intact Female	30.2840111
○	2041	2 years	A702745	Dog	Labrador Retriever Mix	Black	2013-05-22	2015-05-22 11:45:00	2015-05-22T11:45:00	Abigail		Return to Owner	Intact Female	30.7157941
○	2225	2 years	A757341	Dog	Labrador Retriever Mix	Black/White	2015-09-01	2017-10-03 12:27:00	2017-10-03T12:27:00	19	Partner	Transfer	Intact Female	30.3814182
○	3319	9 months	A687748	Dog	Labrador Retriever Mix	Yellow	2013-12-09	2014-09-09 17:01:00	2014-09-09T17:01:00		Suffering	Euthanasia	Intact Female	30.7296534
○	4222	1 year	A735551	Dog	Labrador Retriever Mix	Black	2015-09-25	2016-09-27 14:10:00	2016-09-27T14:10:00	Daisy		Return to Owner	Intact Female	30.5150703

« ‹ 1 / 2 › »

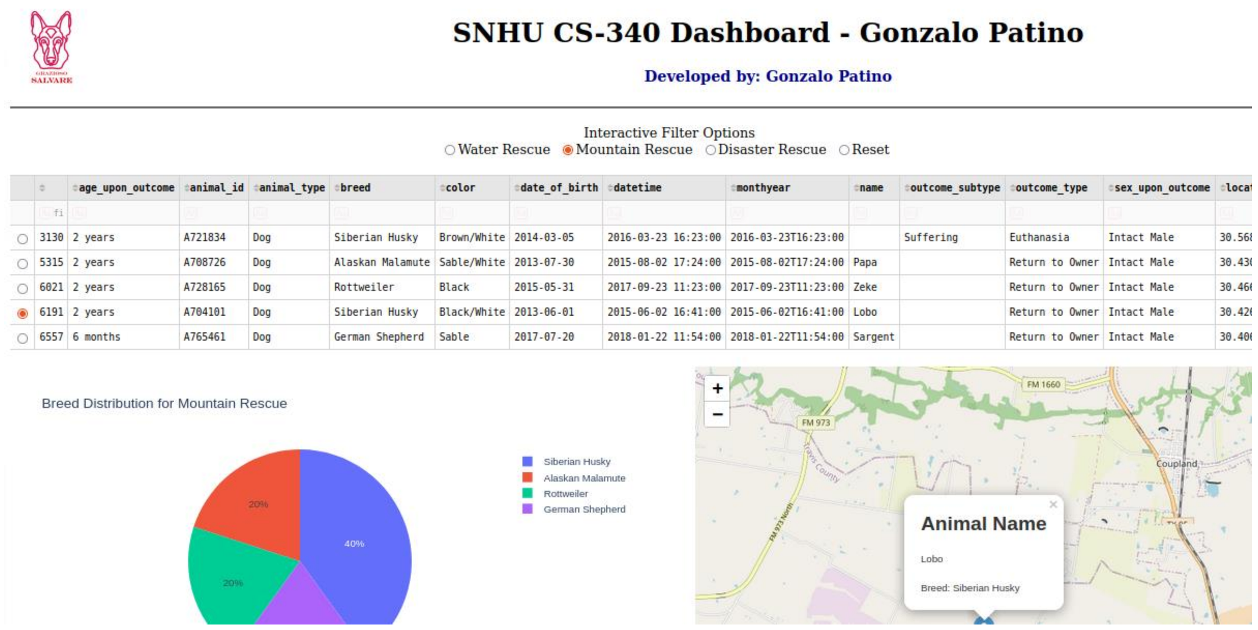
Figure 3. Water Rescue Filter and selecting record ID 1628- 1 out of 2. Please note there are left and right arrows for more records.



●	1628	9 months	A740471	Dog	Labrador Retriever Mix	Tan/White	2016-03-17	2016-12-23 17:13:00	2016-12-23T17:13:00	Mika		Adoption	Intact Female	30.7569
○	1757	7 months	A742767	Dog	Labrador Retriever Mix	Black	2016-06-27	2017-02-14 15:20:00	2017-02-14T15:20:00	Marley		Return to Owner	Intact Female	30.4869
○	1988	1 year	A762781	Dog	Labrador Retriever Mix	Black/White	2016-11-27	2017-12-03 13:09:00	2017-12-03T13:09:00		Partner	Transfer	Intact Female	30.2840
○	2041	2 years	A702745	Dog	Labrador Retriever Mix	Black	2013-05-22	2015-05-22 11:45:00	2015-05-22T11:45:00	Abigail		Return to Owner	Intact Female	30.7157
○	2225	2 years	A757341	Dog	Labrador Retriever Mix	Black/White	2015-09-01	2017-10-03 12:27:00	2017-10-03T12:27:00	19	Partner	Transfer	Intact Female	30.3814
○	3319	9 months	A687748	Dog	Labrador Retriever Mix	Yellow	2013-12-09	2014-09-09 17:01:00	2014-09-09T17:01:00		Suffering	Euthanasia	Intact Female	30.7296
○	4222	1 year	A735551	Dog	Labrador Retriever Mix	Black	2015-09-25	2016-09-27 14:10:00	2016-09-27T14:10:00	Daisy		Return to Owner	Intact Female	30.5150

« ‹ 1 / 2 ›

Figure 4. Water Rescue Filter and selecting record ID 1628- 2 out of 2. Please note there are left and right arrows for more records.



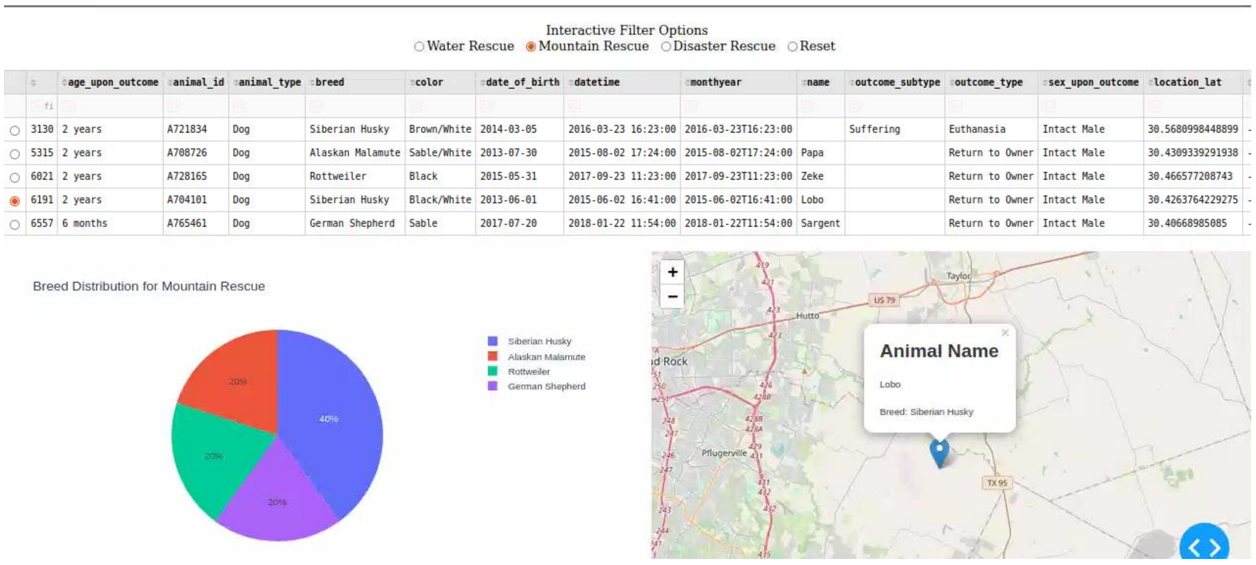
SNHU CS-340 Dashboard - Gonzalo Patino

Developed by: Gonzalo Patino

Interactive Filter Options
( ) Water Rescue [✓ Mountain Rescue] ( ) Disaster Rescue ( ) Reset

		age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth	datetime	monthyear	name	outcome_subtype	outcome_type	sex_upon_outcome	loca
○	3130	2 years	A721834	Dog	Siberian Husky	Brown/White	2014-03-05	2016-03-23 16:23:00	2016-03-23T16:23:00		Suffering	Euthanasia	Intact Male	30.568
○	5315	2 years	A708726	Dog	Alaskan Malamute	Sable/White	2013-07-30	2015-08-02 17:24:00	2015-08-02T17:24:00	Papa		Return to Owner	Intact Male	30.43
○	6021	2 years	A728165	Dog	Rottweiler	Black	2015-05-31	2017-09-23 11:23:00	2017-09-23T11:23:00	Zeke		Return to Owner	Intact Male	30.46
◉	6191	2 years	A704101	Dog	Siberian Husky	Black/White	2013-06-01	2015-06-02 16:41:00	2015-06-02T16:41:00	Lobo		Return to Owner	Intact Male	30.42
○	6557	6 months	A765461	Dog	German Shepherd	Sable	2017-07-20	2018-01-22 11:54:00	2018-01-22T11:54:00	Sargent		Return to Owner	Intact Male	30.40

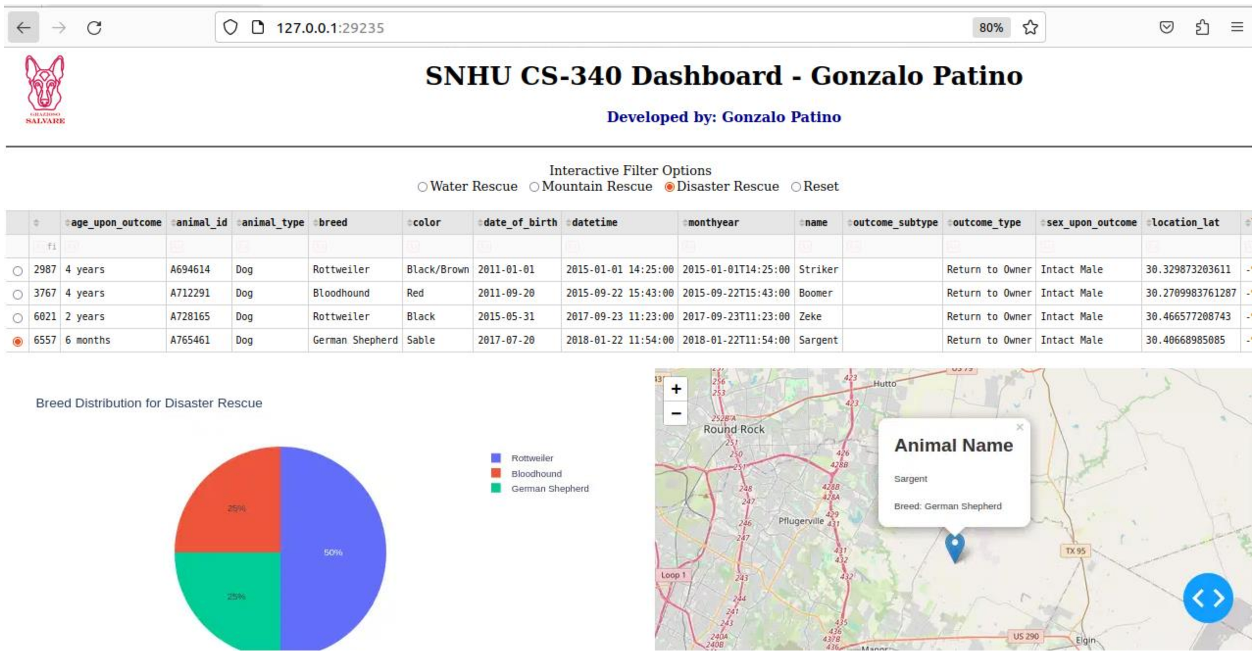
Figure 5. Montain Rescue Filter and selecting record ID 6191- 1 out of 2.



Interactive Filter Options
( ) Water Rescue [✓ Mountain Rescue] ( ) Disaster Rescue ( ) Reset

		age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth	datetime	monthyear	name	outcome_subtype	outcome_type	sex_upon_outcome	location_lat
○	3130	2 years	A721834	Dog	Siberian Husky	Brown/White	2014-03-05	2016-03-23 16:23:00	2016-03-23T16:23:00		Suffering	Euthanasia	Intact Male	30.5680998448899
○	5315	2 years	A708726	Dog	Alaskan Malamute	Sable/White	2013-07-30	2015-08-02 17:24:00	2015-08-02T17:24:00	Papa		Return to Owner	Intact Male	30.4309339291938
○	6021	2 years	A728165	Dog	Rottweiler	Black	2015-05-31	2017-09-23 11:23:00	2017-09-23T11:23:00	Zeke		Return to Owner	Intact Male	30.466577208743
◉	6191	2 years	A704101	Dog	Siberian Husky	Black/White	2013-06-01	2015-06-02 16:41:00	2015-06-02T16:41:00	Lobo		Return to Owner	Intact Male	30.4263764229275
○	6557	6 months	A765461	Dog	German Shepherd	Sable	2017-07-20	2018-01-22 11:54:00	2018-01-22T11:54:00	Sargent		Return to Owner	Intact Male	30.40668985085

Figure 6. Montain Rescue Filter and selecting record ID 6191- 2 out of 2.



127.0.0.1:29235

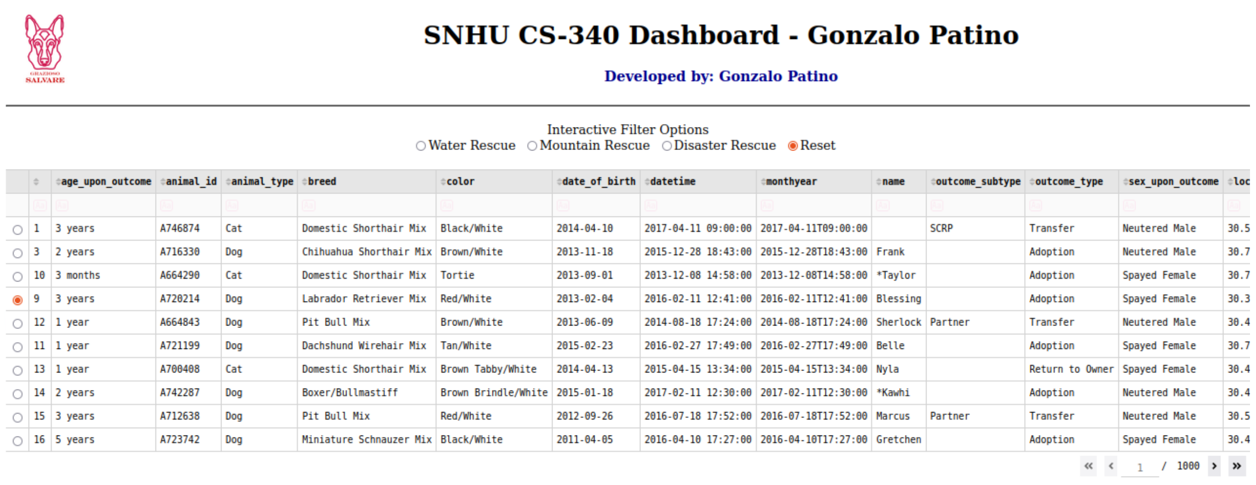
SNHU CS-340 Dashboard - Gonzalo Patino

Developed by: Gonzalo Patino

Interactive Filter Options
Water Rescue Mountain Rescue [✓ Disaster Rescue] Reset

		age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth	datetime	monthyear	name	outcome_subtype	outcome_type	sex_upon_outcome	location_lat
	fi													
○	2987	4 years	A694614	Dog	Rottweiler	Black/Brown	2011-01-01	2015-01-01 14:25:00	2015-01-01T14:25:00	Striker		Return to Owner	Intact Male	30.329873203611
○	3767	4 years	A712291	Dog	Bloodhound	Red	2011-09-20	2015-09-22 15:43:00	2015-09-22T15:43:00	Boomer		Return to Owner	Intact Male	30.2709983761287
○	6021	2 years	A728165	Dog	Rottweiler	Black	2015-05-31	2017-09-23 11:23:00	2017-09-23T11:23:00	Zeke		Return to Owner	Intact Male	30.466577208743
●	6557	6 months	A765461	Dog	German Shepherd	Sable	2017-07-20	2018-01-22 11:54:00	2018-01-22T11:54:00	Sargent		Return to Owner	Intact Male	30.40668985085

Figure 7. Disaster Rescue Filter and selecting record ID 6557



SNHU CS-340 Dashboard - Gonzalo Patino

Developed by: Gonzalo Patino

Interactive Filter Options
Water Rescue Mountain Rescue Disaster Rescue [✓ Reset]

		age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth	datetime	monthyear	name	outcome_subtype	outcome_type	sex_upon_outcome	loc
○	1	3 years	A746874	Cat	Domestic Shorthair Mix	Black/White	2014-04-10	2017-04-11 09:00:00	2017-04-11T09:00:00		SCRP	Transfer	Neutered Male	30.5
○	3	2 years	A716330	Dog	Chihuahua Shorthair Mix	Brown/White	2013-11-18	2015-12-28 18:43:00	2015-12-28T18:43:00	Frank		Adoption	Neutered Male	30.7
○	10	3 months	A664290	Cat	Domestic Shorthair Mix	Tortie	2013-09-01	2013-12-08 14:58:00	2013-12-08T14:58:00	*Taylor		Adoption	Spayed Female	30.7
●	9	3 years	A720214	Dog	Labrador Retriever Mix	Red/White	2013-02-04	2016-02-11 12:41:00	2016-02-11T12:41:00	Blessing		Adoption	Spayed Female	30.3
○	12	1 year	A664843	Dog	Pit Bull Mix	Brown/White	2013-06-09	2014-08-18 17:24:00	2014-08-18T17:24:00	Sherlock	Partner	Transfer	Neutered Male	30.4
○	11	1 year	A721199	Dog	Dachshund Wirehair Mix	Tan/White	2015-02-23	2016-02-27 17:49:00	2016-02-27T17:49:00	Belle		Adoption	Spayed Female	30.7
○	13	1 year	A700408	Cat	Domestic Shorthair Mix	Brown Tabby/White	2014-04-13	2015-04-15 13:34:00	2015-04-15T13:34:00	Nyla		Return to Owner	Spayed Female	30.4
○	14	2 years	A742287	Dog	Boxer/Bullmastiff	Brown Brindle/White	2015-01-18	2017-02-11 12:30:00	2017-02-11T12:30:00	*Kawhi		Adoption	Neutered Male	30.4
○	15	3 years	A712638	Dog	Pit Bull Mix	Red/White	2012-09-26	2016-07-18 17:52:00	2016-07-18T17:52:00	Marcus	Partner	Transfer	Neutered Male	30.5
○	16	5 years	A723742	Dog	Miniature Schnauzer Mix	Black/White	2011-04-05	2016-04-10 17:27:00	2016-04-10T17:27:00	Gretchen		Adoption	Spayed Female	30.4

« ‹ 1 / 1000 › »

Figure 8. Reset filter part 1 out of 2 with record ID9. Please note there are left and right arrows for more records.

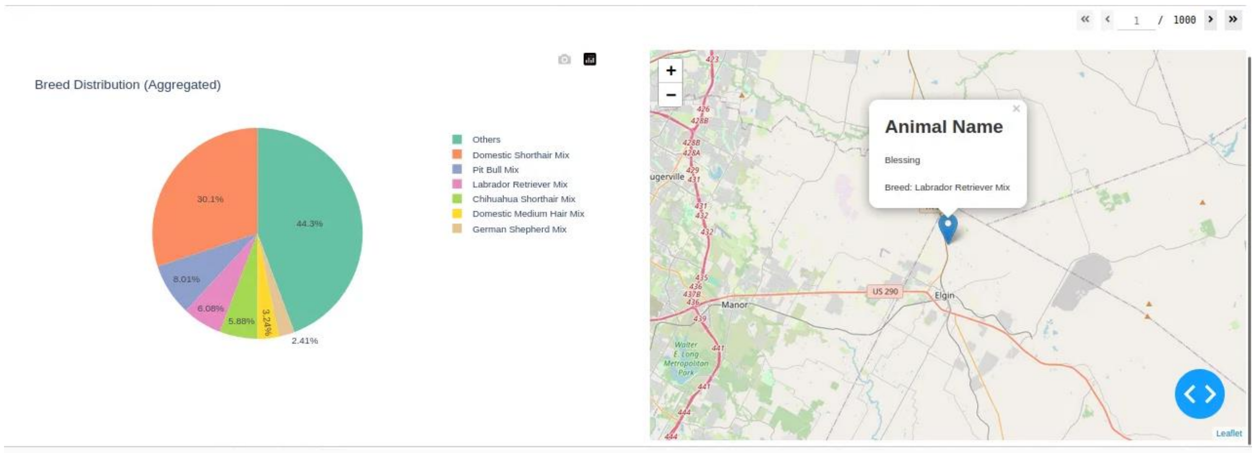


Figure 9. Reset filter part 2 out of 2 with record ID9. Please note there are left and right arrows for more records.

Part II: README (Project 2 deliverables)

Project Overview

This project involves the development of a dashboard for Grazioso Salvare to manage and visualize animal shelter data. The dashboard enables filtering, geolocation mapping, and breed distribution analysis for rescued animals. Built using Python, MongoDB, and the Dash framework, this project ensures a user-friendly interface and robust backend for maintaining the data.

Required Functionality

The dashboard includes:

- **Interactive Filters:** Users can filter animals based on Water, Mountain, or Disaster Rescue types or reset the filters. Filters dynamically update the data table, pie chart, and geolocation map.
- **Dynamic Map:** Displays the geolocation of selected animal records.
- **Pie Chart Visualization:** Displays breed distribution for selected filters.
- **Data Table:** A scrollable table presenting detailed animal records, allowing single-row selection to update the map and pie chart.

Screenshots

Include screenshots of the following:

1. **Initial State:** Showing the full dataset loaded into the table (refer to Figure 1 and Figure 2 in the Word file).
2. **Filter Examples:** Demonstrating functionality for Water (Figures 3 and 4), Mountain (Figures 5 and 6), and Disaster Rescue Filters (Figure 7).
3. **Reset Filter:** Showing the default dataset view (Figures 8 and 9).

Tools Used

- **Python**: Programming language for implementing backend logic and integrating MongoDB with Dash.
- **MongoDB**: Used as the model for storing and querying animal shelter data.

- **Dash Framework**: Provides the controller (application logic) and view (front-end visualization) components for building an interactive dashboard.
- **Plotly**: Used for generating pie charts to visualize breed distribution.
- **Dash Leaflet**: Enables integration of geolocation mapping features.

Rationale for Tool Selection

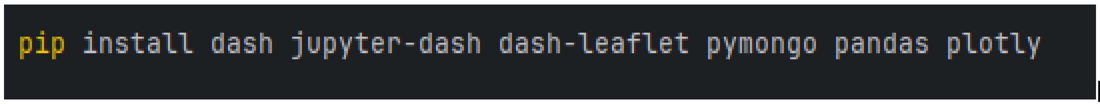
1. **MongoDB**:
    - **Flexible Schema**: Allows storing unstructured data such as animal information with varied attributes.
    - **Python Integration**: Easy interfacing using the PyMongo library.
    - **Scalability**: Handles large datasets efficiently.
2. **Dash Framework**:
    - **Ease of Use**: Allows seamless integration of Python back-end logic with front-end visualizations.
    - **Interactivity**: Built-in support for callbacks enables dynamic updates of filters, charts, and maps.
3. **Dash Leaflet**:
    - **Geolocation Features**: Simplifies adding map functionality to Dash applications.

Resources Used

- Dash Documentation
- MongoDB Documentation
- Plotly Documentation
- Dash Leaflet Documentation

Steps to Reproduce the Project

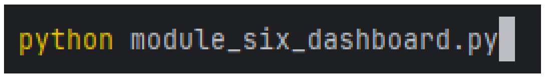
1. **Set Up MongoDB**:
   - Install MongoDB and create a database (AAC) with a collection (animals).
   - Populate the collection with the provided animal shelter data.
2. **Install Required Libraries**:



```
pip install dash jupyter-dash dash-leaflet pymongo pandas plotly
```

Run the Dashboard:

   - Save the Python script and the crud_operations.py file in the same directory.
   - Run the script:



```
python module_six_dashboard.py
```

   - Access the dashboard at http://127.0.0.1:<port>.

3. **Interact with the Dashboard**:
   - Test filters, map, and pie chart updates using the dataset.

Challenges and Solutions

Challenge 1: Filter Implementation

- **Problem**: Querying MongoDB for specific rescue types required translating filter logic to Python.
- **Solution**: Developed specific queries in the callback function to filter the dataset dynamically.

Challenge 2: Blank Names in Records

- **Problem**: Some records had missing names.
- **Solution**: Used a fallback value ("No name found") in the map popup for empty names.

Challenge 3: Map Alignment Issues

- **Problem**: The map pointer sometimes failed to update correctly with selected records.
- **Solution**: Debugged callback logic to validate geolocation data and handle missing or incorrect coordinates.

Reflection and Insights

1. How do you write programs that are maintainable, readable, and adaptable?

Programs are made maintainable by following coding standards, implementing modular design, and providing clear documentation. For instance, the CRUD Python module created in Project One was reusable and well-documented, which enabled seamless integration with the dashboard widgets in Project Two. This modular approach reduced complexity, simplified debugging, and enhanced adaptability when adding features.

2. How else could you use this CRUD Python module in the future?

The CRUD module could be used in various applications where database interactions are required, such as inventory management, customer relationship systems, or any application involving dynamic data manipulation. Its flexibility makes it suitable for any project requiring structured data operations.

3. How do you approach a problem as a computer scientist?

As a computer scientist, I break down problems into manageable parts. For example, in this project, I started by understanding Grazioso Salvare's database and dashboard requirements. Then, I planned the architecture, focusing on the separation of concerns between the model (MongoDB), view (Dash), and controller (callbacks). This methodical approach ensured all requirements were met efficiently.

4. How did your approach to this project differ from previous assignments in other courses?

This project required integrating a front-end dashboard with a back-end database, which was a more holistic task compared to the isolated back-end or front-end assignments in previous courses. It also demanded attention to both technical performance (e.g., handling large datasets) and user experience (e.g., intuitive filters and dynamic visualizations).

5. What techniques or strategies would you use in the future to create databases to meet other client requests?

In future projects, I would:

- Conduct thorough requirement analysis to design a schema that aligns with client needs.
- Maintain a modular and well-documented codebase for easy updates and adaptability.

6. What do computer scientists do, and why does it matter?

Computer scientists design, develop, and optimize systems to solve complex problems and improve efficiency. Their work underpins technological progress across industries. For example, my work on this project helps Grazioso Salvare optimize rescue operations by providing actionable insights, ultimately enhancing their mission's effectiveness.

Features and Functionality

- **Interactive Filters**: Filters animals by rescue type (Water, Mountain, Disaster, or Reset), dynamically updating the data table, pie chart, and geolocation map.
- **Dynamic Map**: Displays geolocations of selected animal records.
- **Pie Chart Visualization**: Shows breed distribution for selected filters.
- **Data Table**: A scrollable, sortable table for viewing detailed records.

Technologies Used

- **Backend**: Python, MongoDB, PyMongo
- **Frontend**: Dash, Dash Leaflet, Plotly
- **Architecture**: Model-View-Controller (MVC)

Challenges and Solutions

1. **Filter Logic**: Developed specific MongoDB queries to implement dynamic filters.
2. **Data Quality Issues**: Handled missing names by displaying "No name found" in the map popups.
3. **Map Updates**: Debugged callback functions to ensure correct pointer alignment with selected records.

Summary

The Grazioso Salvare Dashboard project successfully meets the business and technical requirements outlined in the initial specifications. By leveraging modern technologies such as MongoDB, Dash, and Plotly, the system provides an interactive and user-friendly interface for managing animal shelter data.

The Grazioso Salvare Dashboard demonstrates the power of combining advanced data management (MongoDB) with interactive visualization tools (Dash and Plotly). This project not only fulfills its functional requirements but also lays the foundation for future scalability, such as adding new rescue types, integrating advanced analytics, or expanding database records.