William Fishburn

CS 340 Project Two

**README for Grazioso Salvare Dashboard**

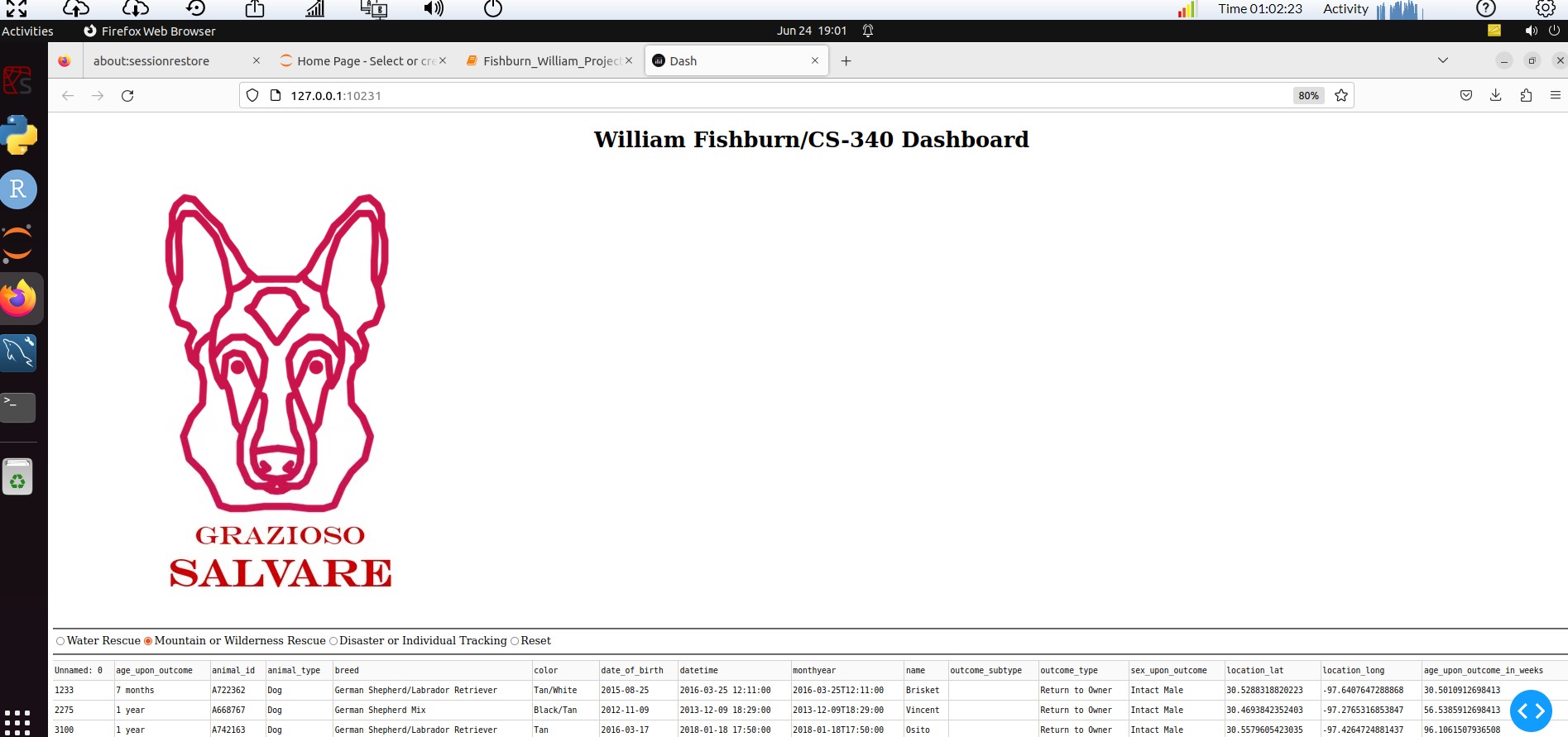
**Project Overview**

The goal of this Grazioso Salvare Dashboard project was to provide a comprehensive and interactive dashboard for managing and analyzing data from an animal shelter. The dashboard enables users to filter and visualize data, specifically focusing on animal breeds suitable for water rescue, mountain or wilderness rescue, and disaster or individual tracking. The project leverages MongoDB for data storage and Dash for the web application's frontend.

**Required Functionality**

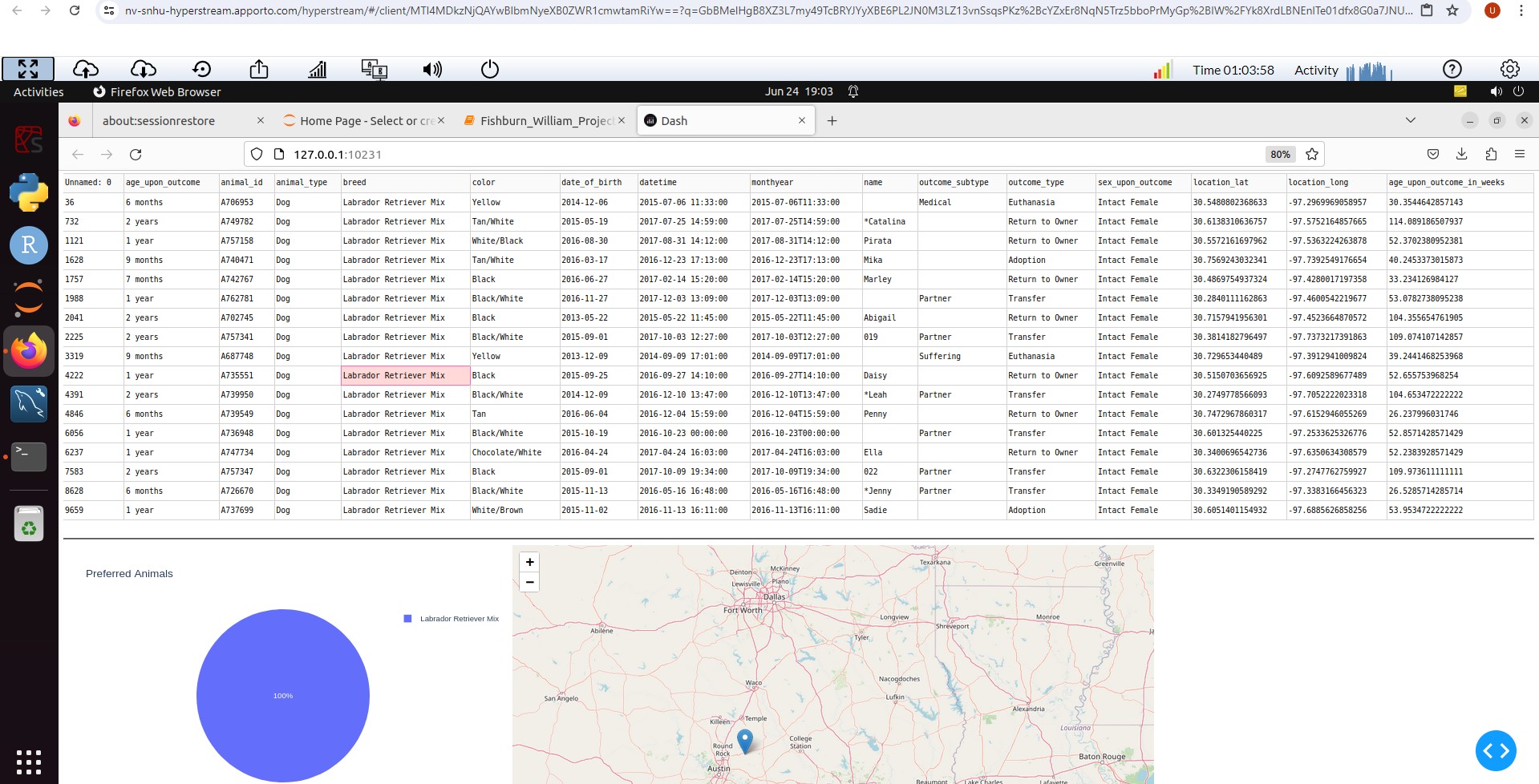
The dashboard fulfills the following functionalities:

1. **Interactive Filtering**: Users can filter data based on rescue categories using radio buttons.
2. **Data Table**: Displays filtered data in an interactive table.
3. **Pie Chart**: Visualizes the distribution of animal breeds using a pie chart.
4. **Geolocation Map**: Displays the location of selected animals on a map with detailed information.  
     
   **Figure A: My name shown, along with the required logo.**

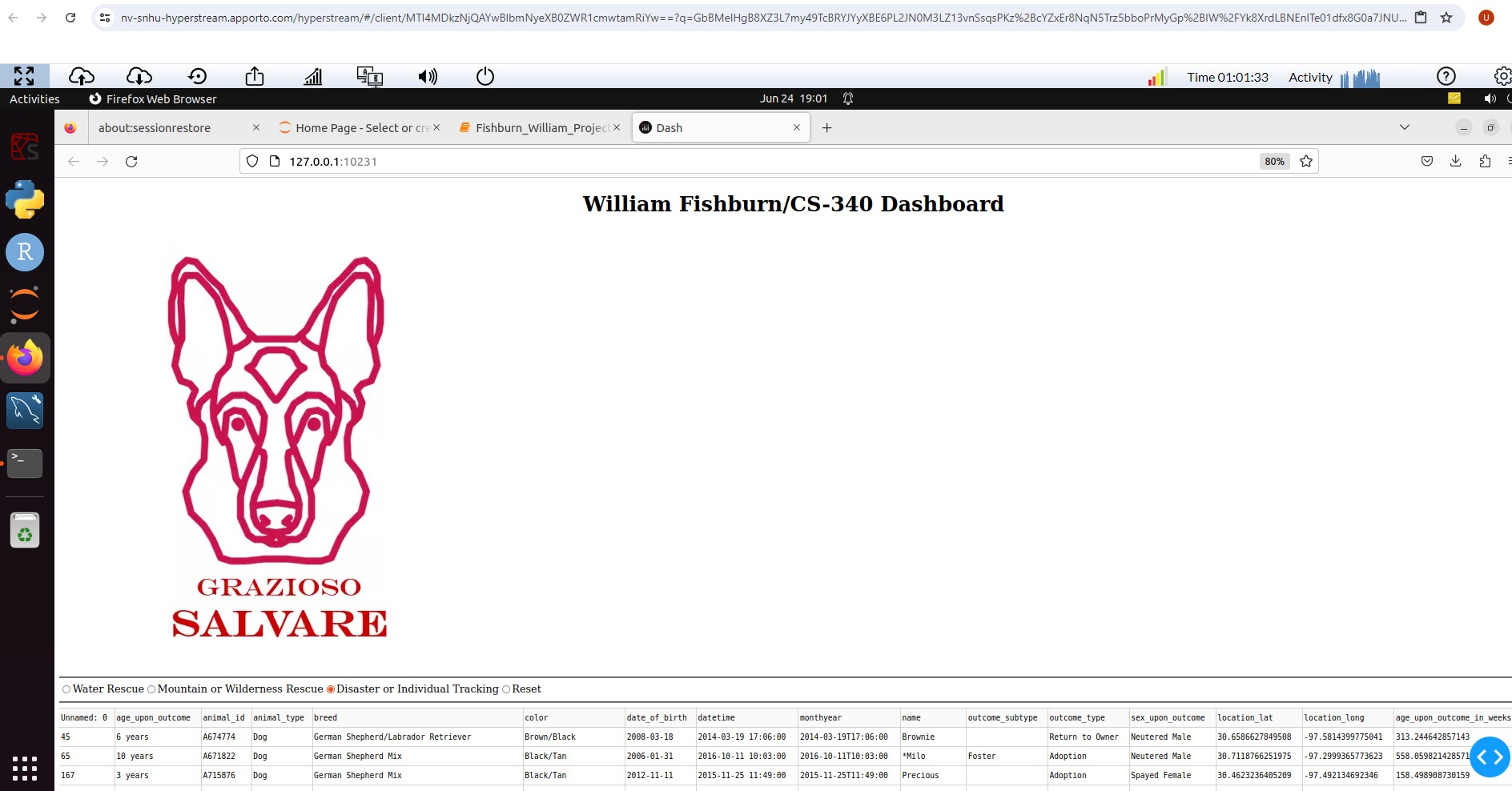
****

**Figure Ba, Bb, Bc: Each filtering option shown**

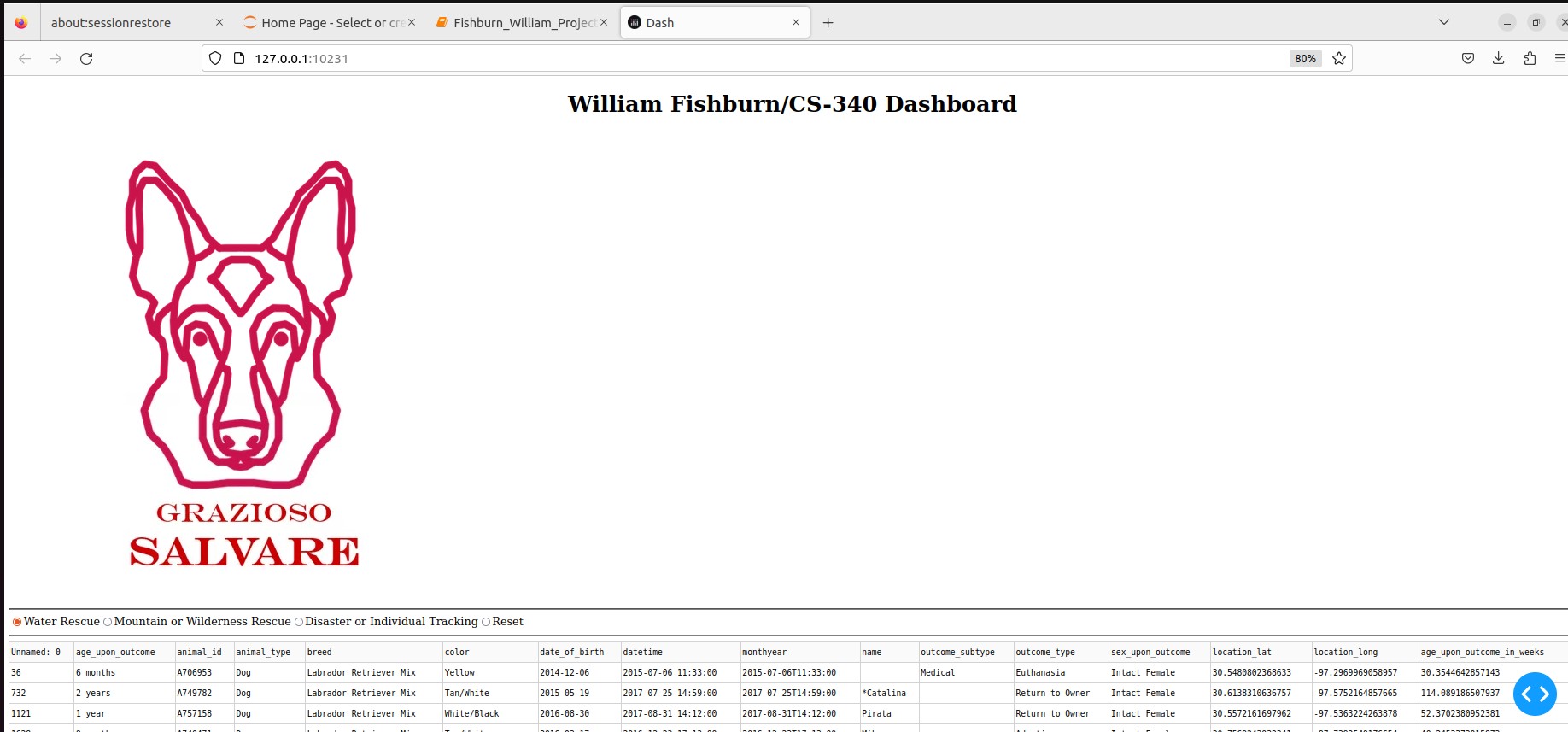
**Mountain/Wilderness(Ba):**

****

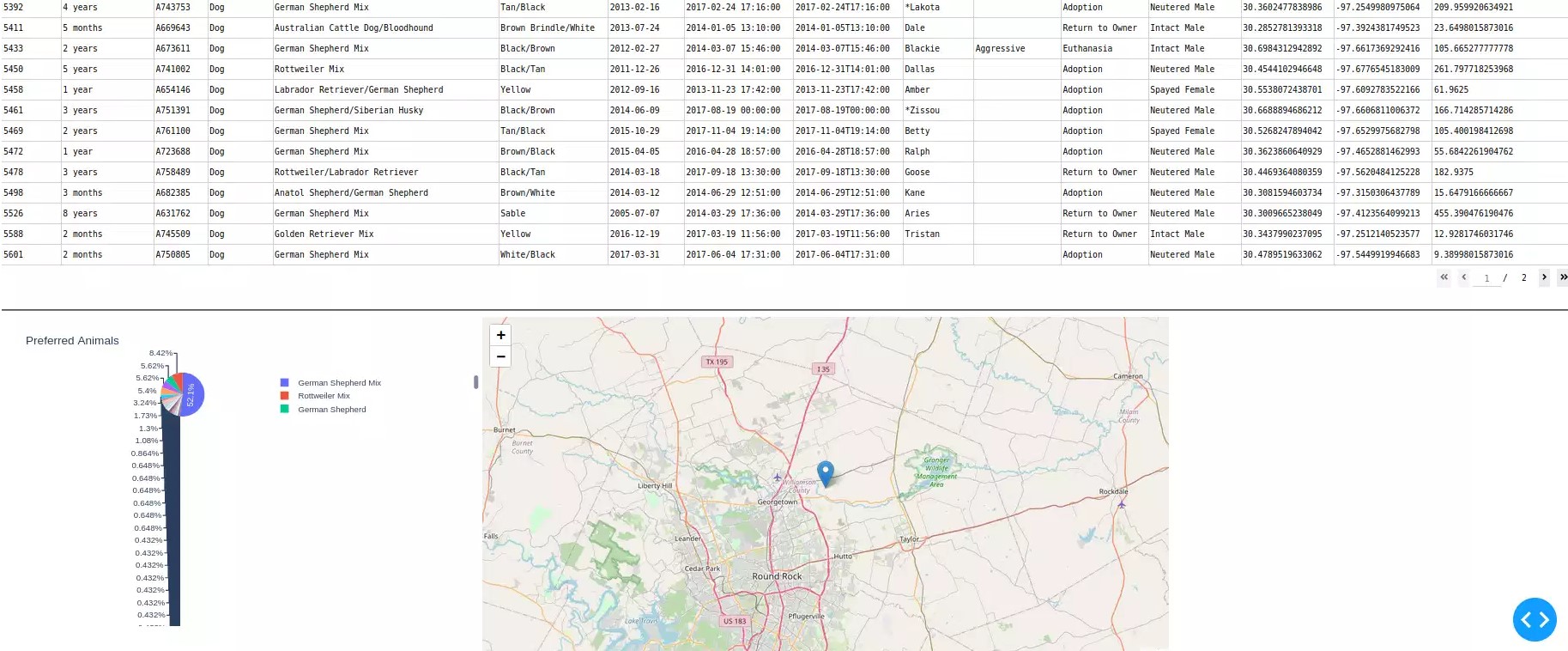
**Dizaster or Individual(Bb):**

****

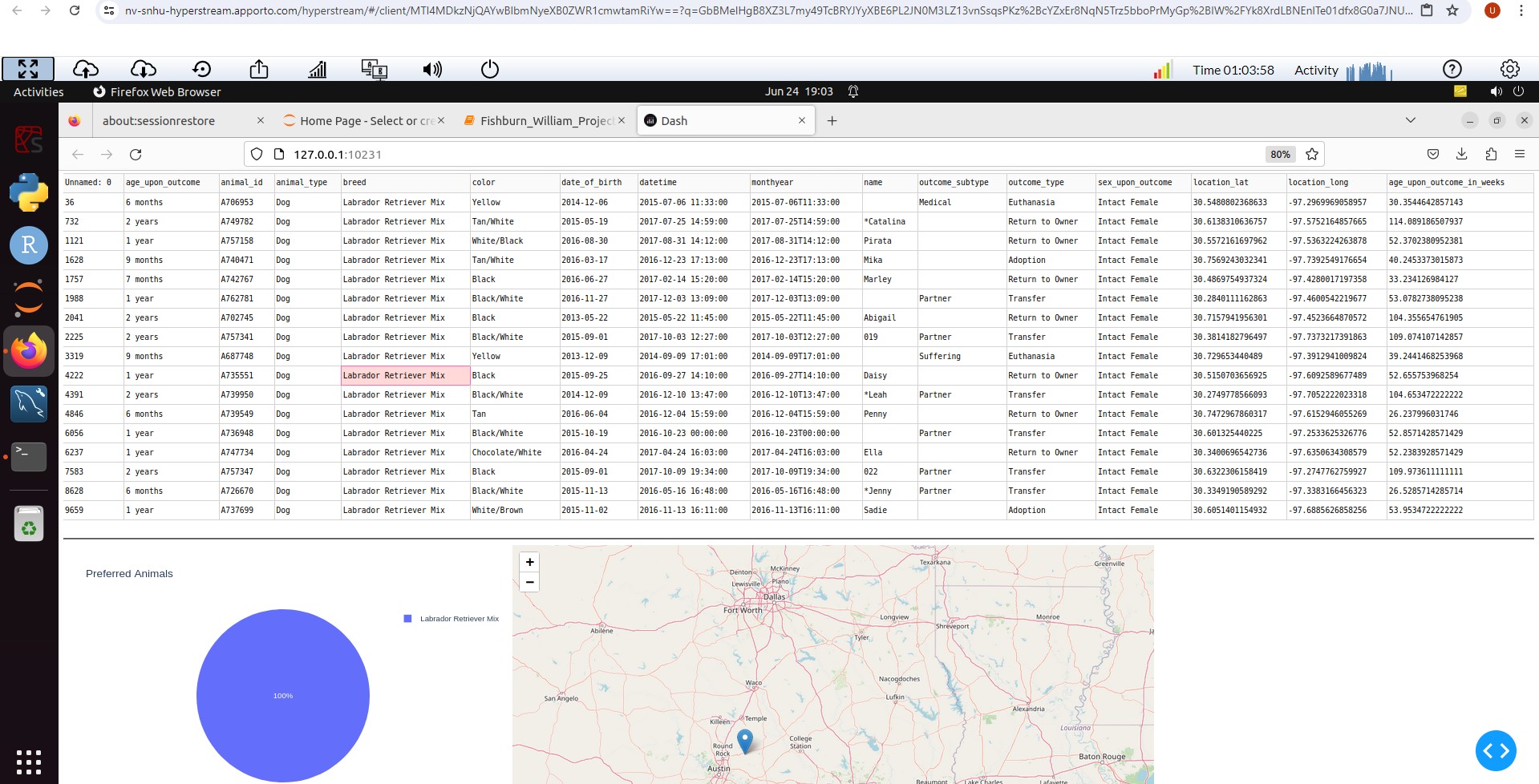
**Water(Bc):**

****

**Figure C: Pagination**(currently, only the Dizaster filtered records have enough rows to activate pagination)

****

**Figure D: Geolocation chart and Pie Chart**



**Tools and Technologies**

* **MongoDB**: A NoSQL database used for storing the animal shelter data. MongoDB is chosen for its flexibility in handling unstructured data, scalability, and seamless integration with Python using the pymongo library.
* **Dash by Plotly**: A Python framework used for building analytical web applications. Dash is chosen for its ability to create interactive, web-based dashboards with minimal coding.
* **Pandas**: A data manipulation library used for reading and processing data.
* **Dash Leaflet**: A library used for rendering maps in Dash applications.

**The importance of MongoDB for component modeling**

* **Flexibility**: Handles various data types without requiring a predefined schema.
* **Scalability**: Easily scales horizontally, making it suitable for large datasets.
* **Integration**: Provides native support for Python through the pymongo library, enabling seamless data operations.

**The importance of Dash Framework**

* **Simplicity**: Allows rapid development of web applications with Python code.
* **Interactivity**: Provides a wide range of components for building interactive dashboards.
* **Integration**: Supports integration with Plotly for advanced data visualization.

Resources and Software

* [MongoDB](https://www.mongodb.com/)
* [Dash by Plotly](https://dash.plotly.com/)
* [Pandas Documentation](https://pandas.pydata.org/docs/)
* [Dash Leaflet Documentation](https://www.dash-leaflet.com/)
* [Plotly](https://plotly.com/)

**Steps to Complete the Project**

1. **Setup MongoDB**: Configure MongoDB with the animal shelter data.
2. **Develop CRUD Operations**: Implement CRUD operations for interacting with MongoDB.
3. **Read and Process Data**: Use Pandas to read and preprocess data from CSV files.
4. **Create Dash Layout**: Develop the layout of the Dash application with filtering options, data table, pie chart, and map.
5. **Implement Callbacks**: Create callbacks to handle user interactions and update the dashboard components dynamically.
6. **Test and Deploy**: Test the application thoroughly and deploy it for use.

**Challenges and Solutions**

1. **Data Processing**: Handling missing or invalid data in the dataset was a challenge. This was addressed by adding checks and default values for missing data, as well as making sure .csv and image files are referenced properly.
2. **Map Rendering**: Initial issues with geolocation data rendering were resolved by ensuring the correct data format and handling edge cases.
3. **Interactive Filtering**: Ensuring the filters worked correctly with the data required careful testing using an iterative approach.