7-2 Project Two

Gavin M. Bish

Southern New Hampshire University

CS340: Client/Server Development

Professor Reuben Wilson

December 10th, 2024

7-2 Project Two  
**Grazioso Salvare Dashboard**

## **Project Overview**

The Grazioso Salvare Dashboard is a dynamic web application designed to support the identification of rescue dogs for specialized training programs. It provides an intuitive interface to interact with shelter data, enabling users to filter, analyze, and visualize rescue dog information. This project integrates MongoDB as the database backend and uses Dash for the frontend to create an interactive and user-friendly experience.

## **Features**

The dashboard includes the following key features:

1. **Interactive Data Table**:
   1. Displays shelter data with options for filtering, sorting, and selecting rows.
   2. Responds dynamically to user-selected filters for Water Rescue, Mountain/Wilderness Rescue, Disaster/Tracking, and Reset.
   3. Highlights selected columns for easier user navigation.
2. **Geolocation Map**:
   1. Visualizes the location of animals dynamically based on user-selected filters.
   2. Displays tooltips and popups with relevant animal details (e.g., name and breed).
3. **Additional Chart**:
   1. Provides a visual representation of aggregated data (e.g., breed distribution).
   2. Dynamically updates based on filtered results.
4. **Interactive Filtering Options**:
   1. Intuitive filter widgets (radio buttons) to select specific rescue types or reset to show all data.
5. **Branding and Identification**:
   1. Includes the Grazioso Salvare logo and a unique developer identifier.

## **How the Project Was Created**

### **1. Database Integration**

* **MongoDB** was used to store the Austin Animal Center Outcomes dataset.
* A Python CRUD module facilitated database operations, allowing queries to filter and retrieve specific data based on user selections.
* The dataset was processed into a Pandas DataFrame for compatibility with Dash components.

### **2. Dashboard Design and Layout**

* **Dash Framework** was used to build the frontend:
  + The layout was structured with HTML and Dash Core Components for the data table, charts, and map.
  + The Grazioso Salvare logo and developer identifier were embedded in the header.
* The layout dynamically updates in response to user actions.

### **3. Interactivity with Callbacks**

* Dash callbacks were implemented to:
  + Update the DataTable with filtered results.
  + Apply styles to highlight selected columns.
  + Dynamically update the geolocation map and charts based on filtered data.

### **4. Charts and Visualization**

* A **geolocation map** was created using Dash Leaflet to display animal locations.
* An **additional chart** (e.g., a bar chart) visualized aggregated data like breed distribution.
* Plotly Express was used to generate interactive and visually appealing charts.

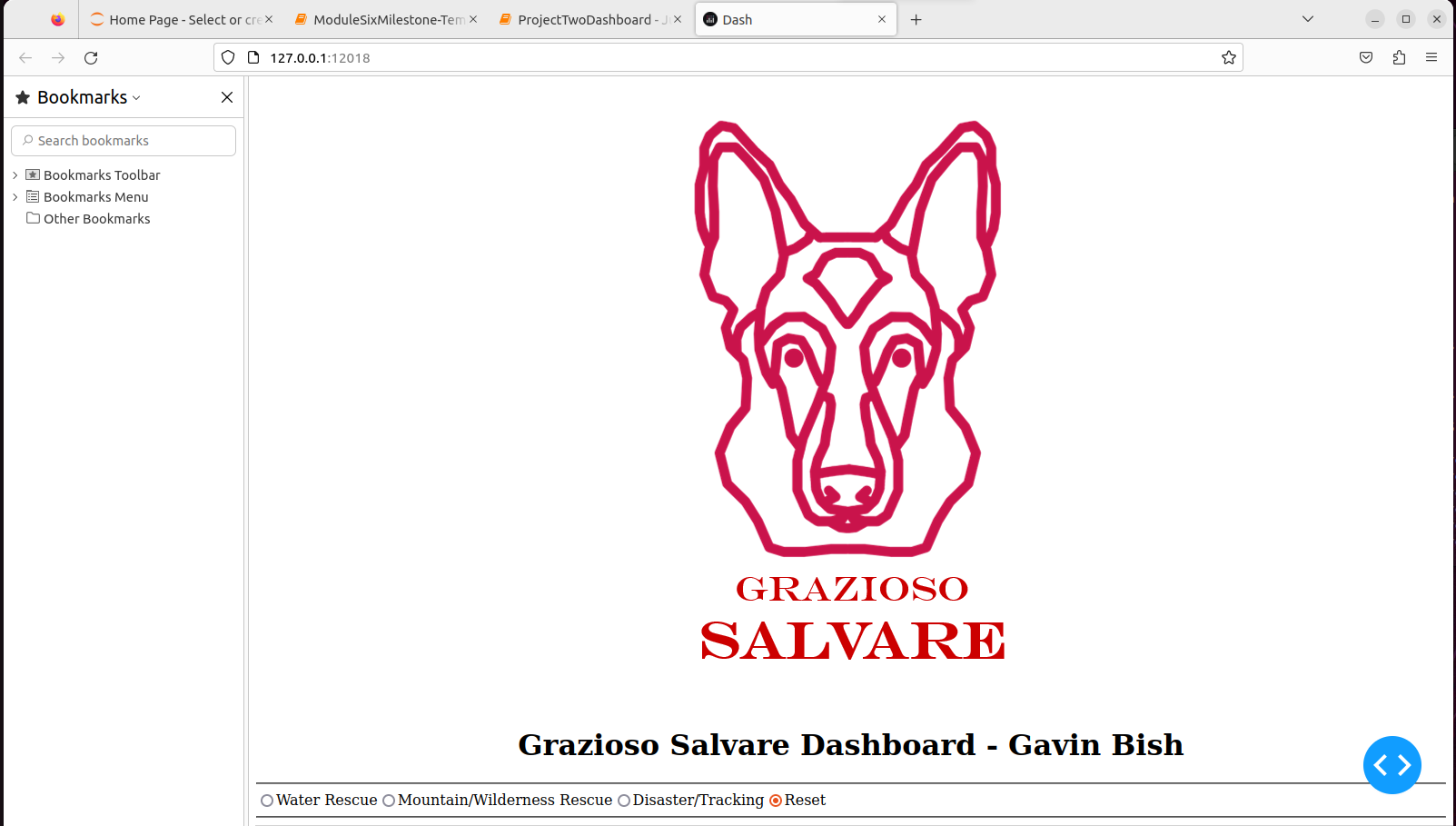
### **5. Testing and Debugging**

* The dashboard was rigorously tested to ensure:
  + Filters dynamically updated all components (DataTable, map, and chart).
  + Edge cases like empty or invalid inputs were handled gracefully.
  + No errors occurred during user interactions.

## **Screenshots and Proof of Functionality**

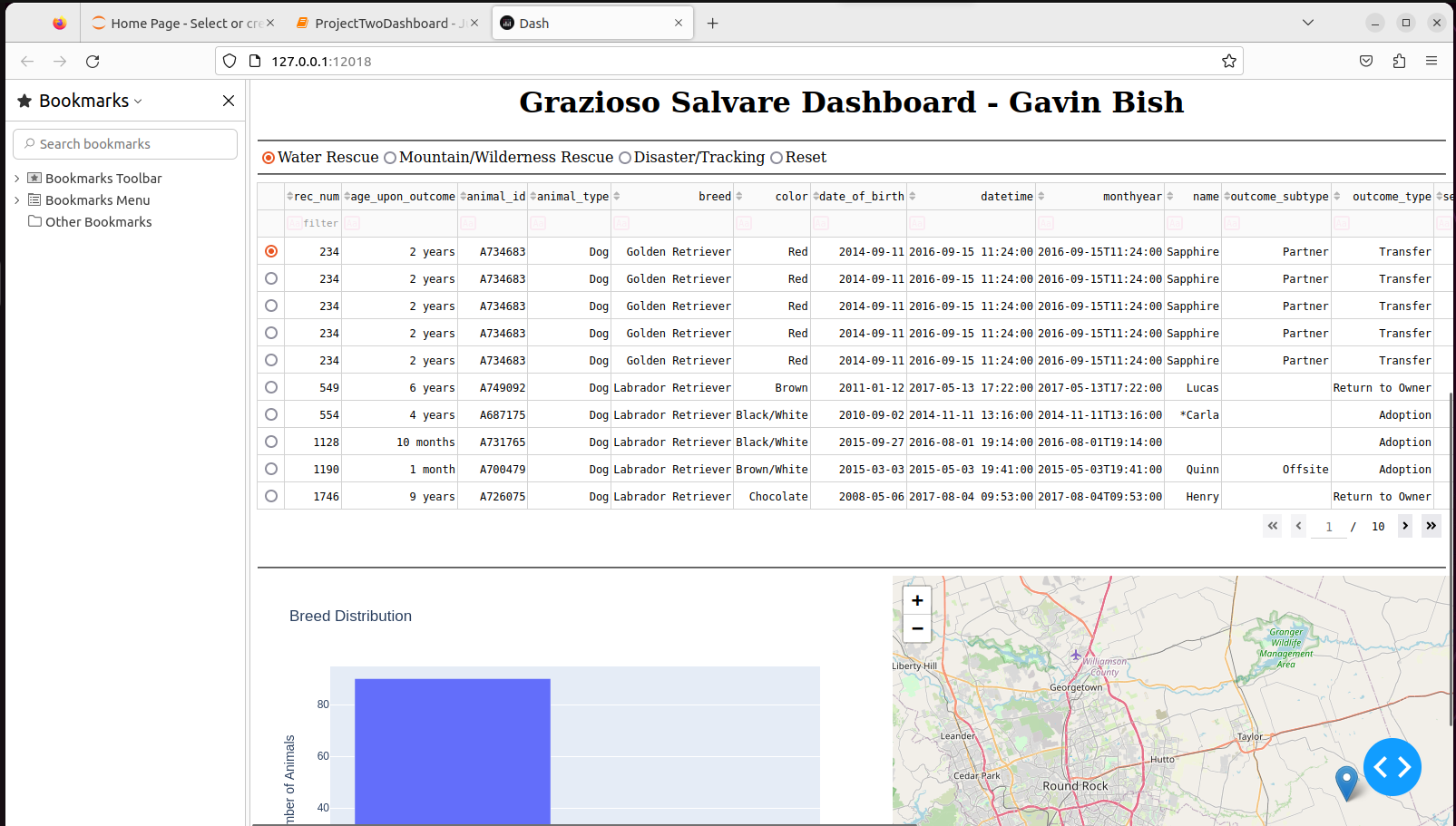
### **Screenshot 1: Initial Dashboard State**

* Includes the unfiltered data table, geolocation map, and additional chart.



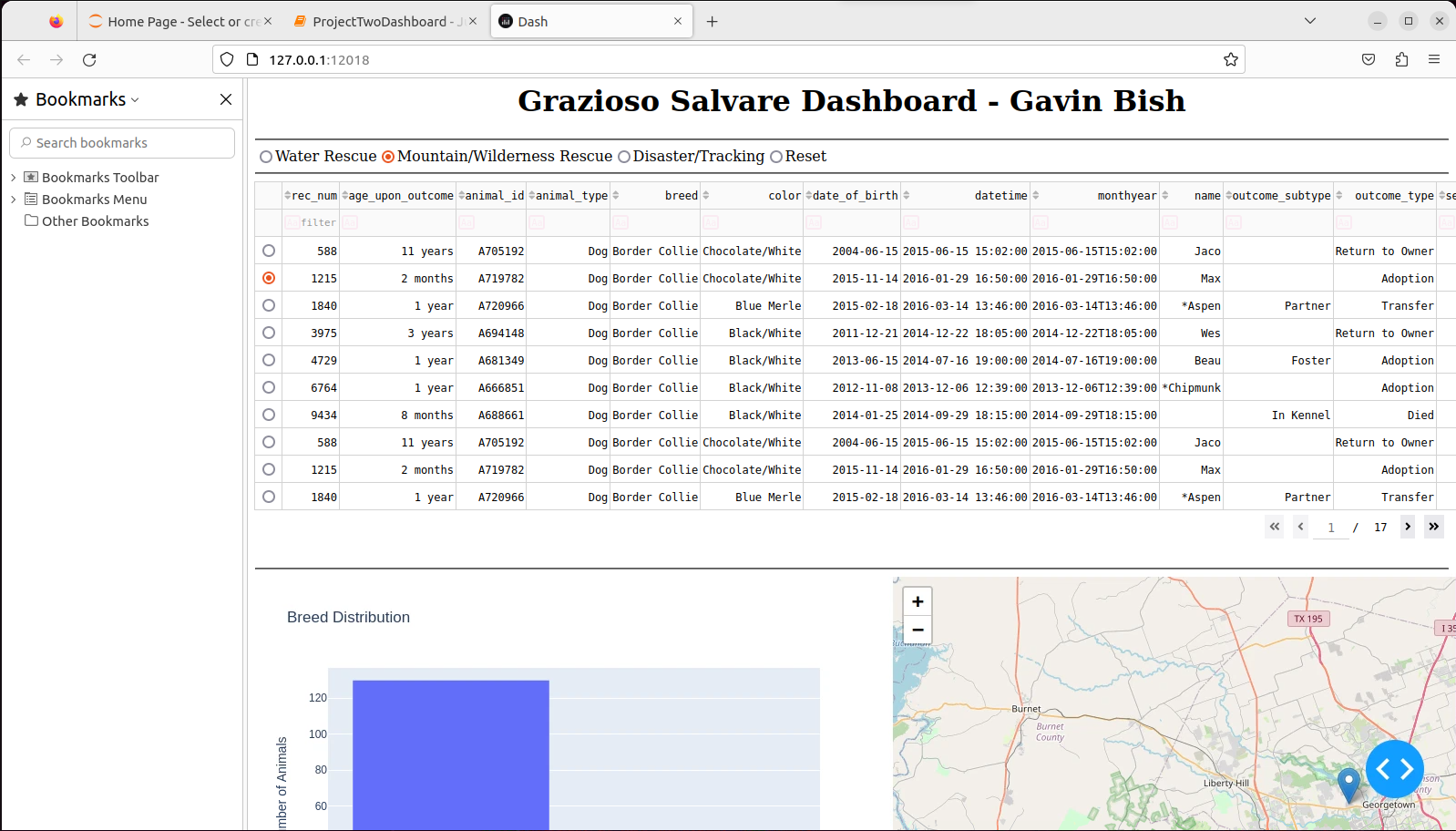
### **Screenshot 2: Filtered View - Water Rescue**

* Shows the DataTable, map, and chart updated with Water Rescue data.



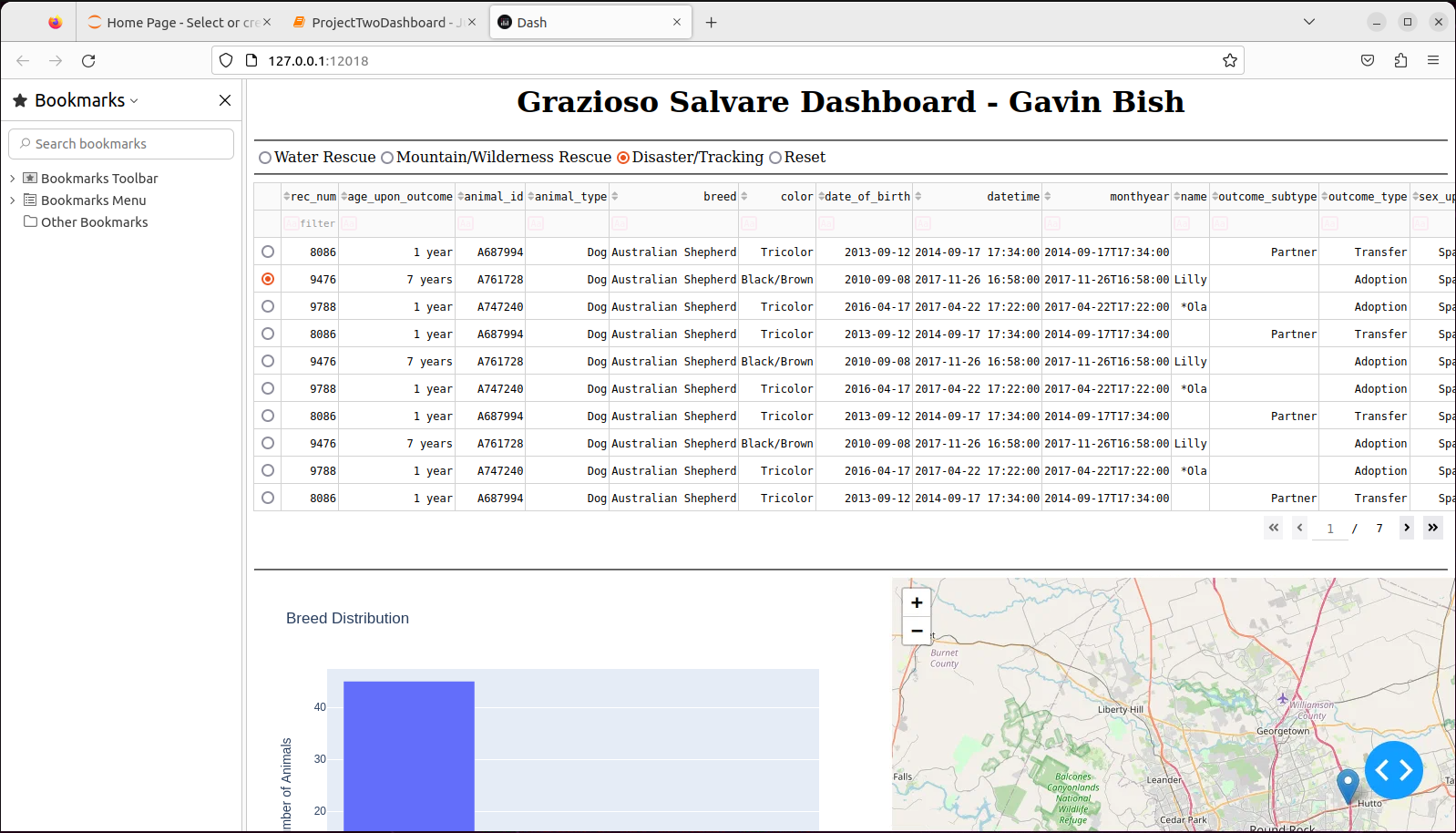
### **Screenshot 3: Filtered View - Mountain/Wilderness Rescue**

* Displays components updated with Mountain/Wilderness Rescue data.



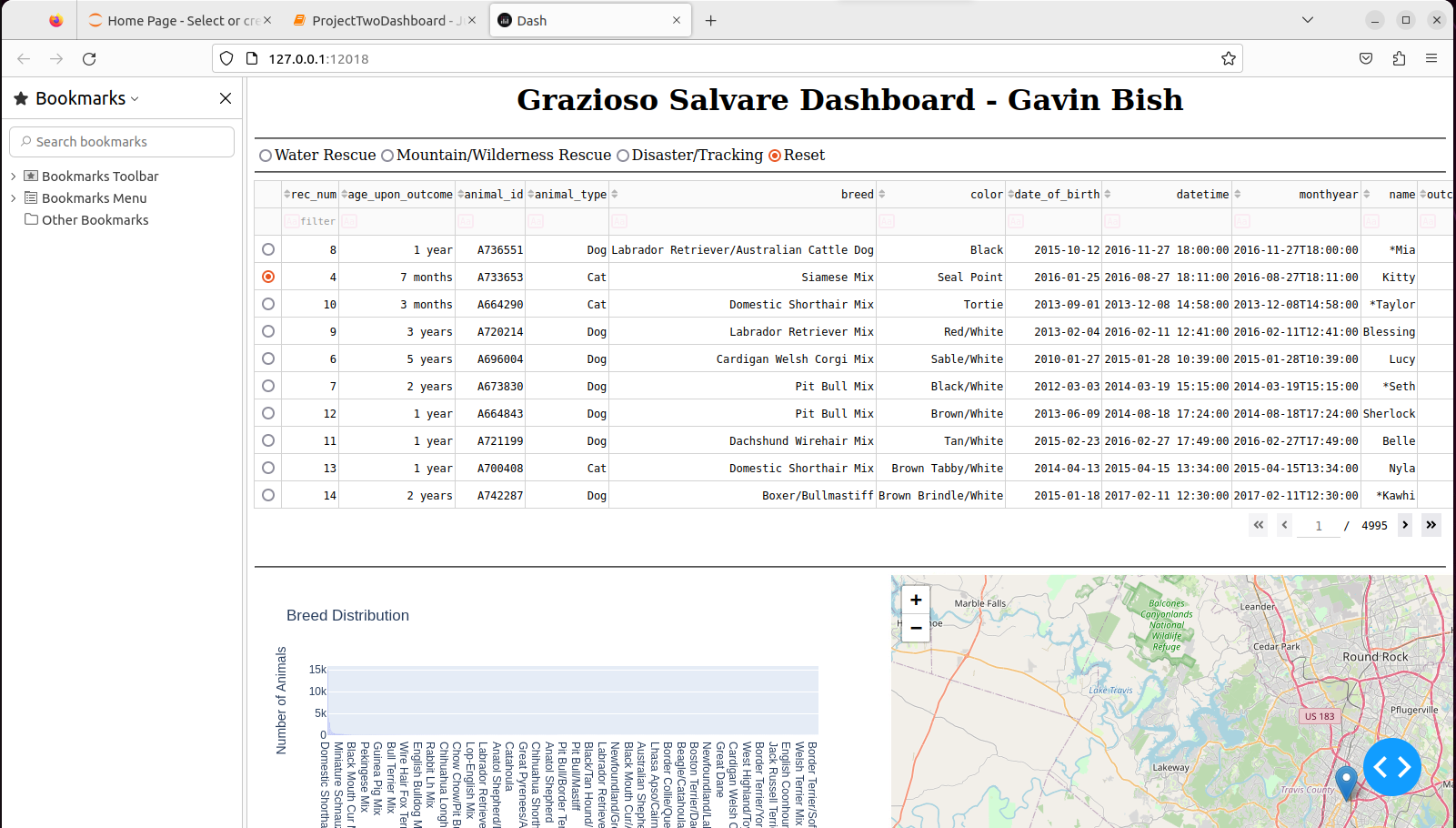
### **Screenshot 4: Filtered View - Disaster/Tracking**

* Displays components updated with Disaster/Tracking data.



### **Screenshot 5: Reset View**

* Shows the dashboard returned to its unfiltered state.



## **Challenges and Solutions**

### **1. Dynamic Data Updates**

* **Challenge**: Ensuring all components updated seamlessly in response to filters.
* **Solution**: Implemented optimized Dash callbacks and leveraged MongoDB queries for real-time data retrieval.

### **2. Geolocation Mapping**

* **Challenge**: Displaying accurate animal locations on the map.
* **Solution**: Verified latitude and longitude fields in the dataset and used Dash Leaflet for dynamic updates.

### **3. Error Handling**

* **Challenge**: Preventing crashes when no data or invalid selections were made.
* **Solution**: Added validation checks for empty datasets and missing columns in callbacks.

### **4. Logo Integration**

* **Challenge**: Handling file path errors for the Grazioso Salvare logo.
* **Solution**: Used Base64 encoding to embed the logo directly and implemented fallback logic for missing files.

## **Tools and Technologies**

1. **MongoDB**:
   1. Used for dynamic data storage and retrieval.
   2. Integrated with Python via the pymongo library.
2. **Dash Framework**:
   1. Provided a robust structure for building the dashboard.
   2. Enabled dynamic interactivity between components.
3. **Pandas**:
   1. Processed MongoDB data for compatibility with Dash.
4. **Plotly**:
   1. Created interactive charts for visualizing aggregated data.
5. **Dash Leaflet**:
   1. Implemented geolocation mapping.

## **Additional Notes**

* The dashboard was developed with scalability and ease of use in mind, ensuring it meets Grazioso Salvare’s needs.
* Placeholder screenshots have been included. Replace these with actual screenshots demonstrating the functionality of your dashboard.