# **Project Two - Grazioso Salvare Dashboard**

## **Overview**

This project is a **web-based dashboard** designed for **Grazioso Salvare**, a company that identifies dogs suitable for **search-and-rescue training**. The dashboard utilizes **MongoDB** to retrieve shelter data and enables users to filter dogs based on their suitability for different rescue operations.

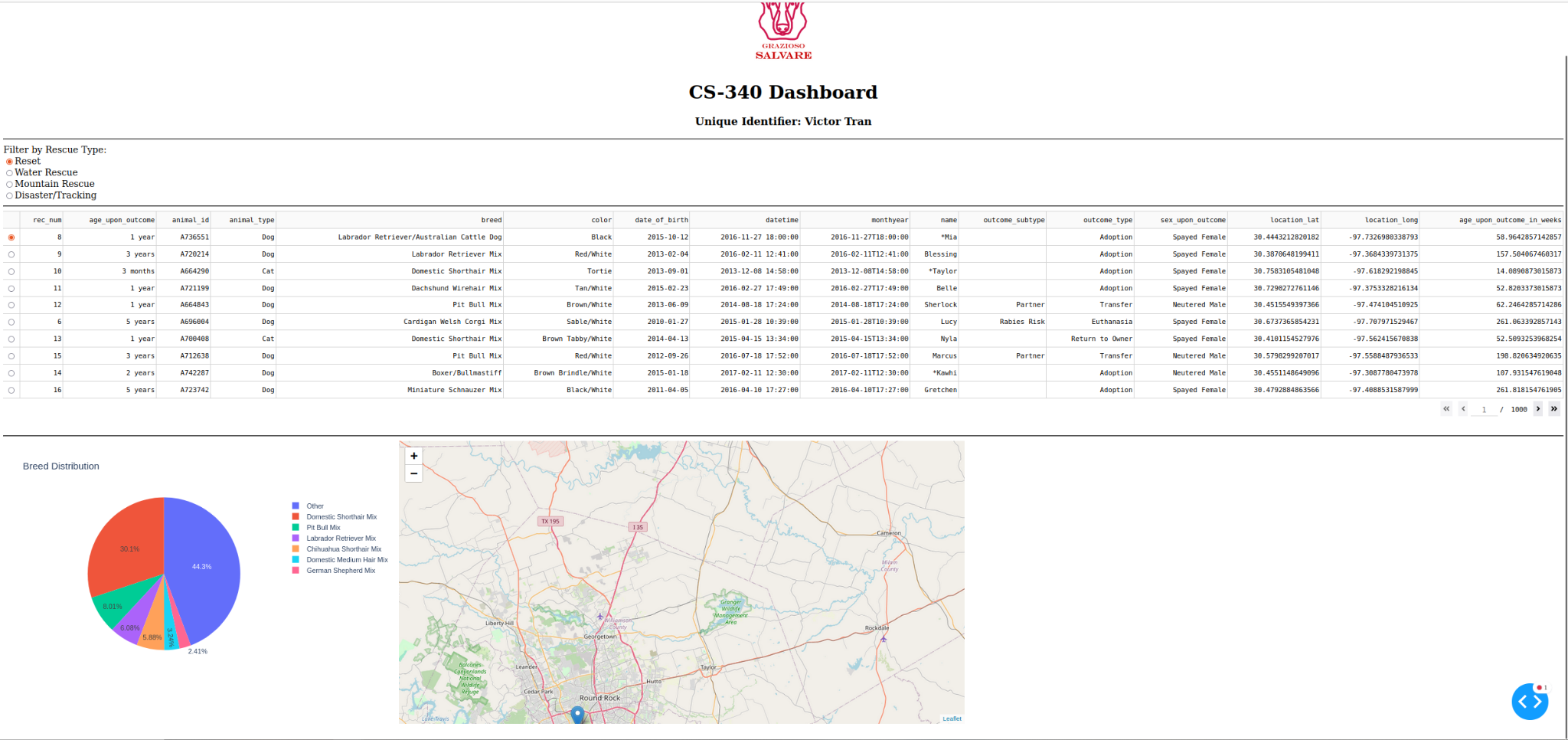
## **Required Functionality**

This dashboard allows users to:

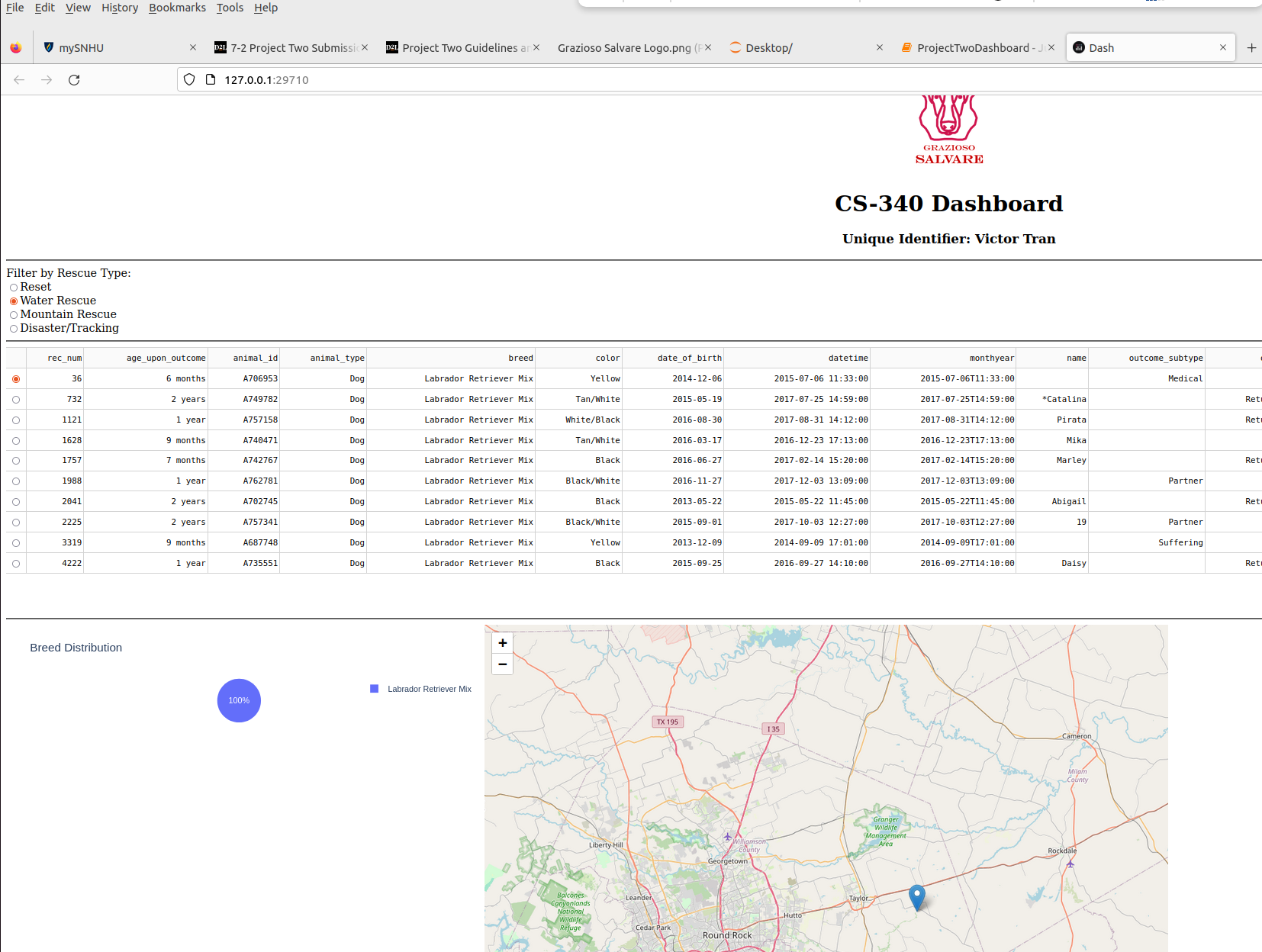
* **Filter Dogs** by:
  + **Water Rescue**
  + **Mountain or Wilderness Rescue**
  + **Disaster/Tracking**
  + **Reset (removes all filters and returns full dataset)**
* **View a Dynamic Data Table** that updates based on filter selection.
* **Analyze Breed Distribution** through a pie chart.
  + If too many breeds are present (when "Reset" is selected), small breeds are grouped into **"Other"** to keep the chart readable.
* **View Animal Locations on a Geolocation Map** that updates dynamically.

### **Screenshots and Screencast**

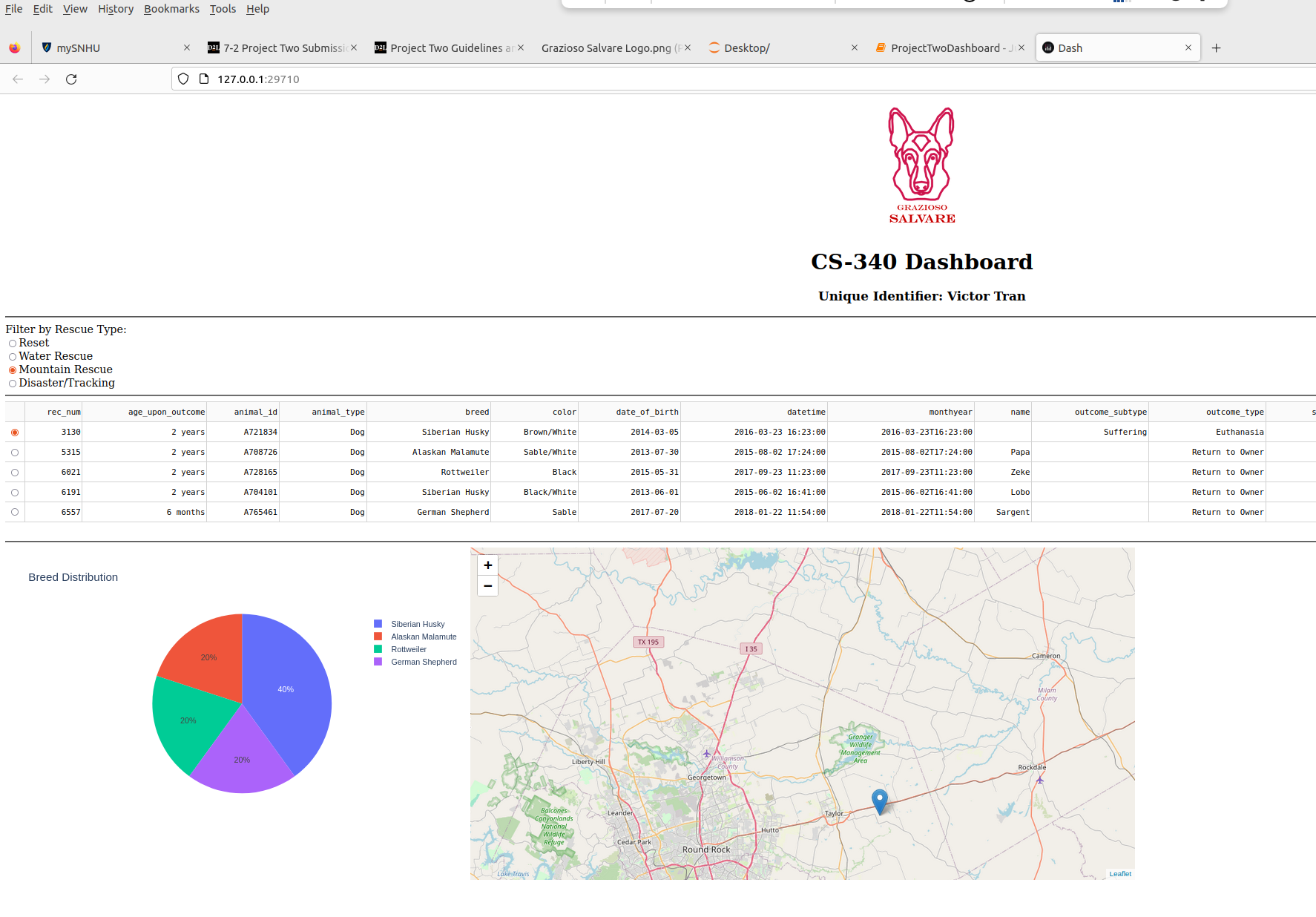
**Reset screenshot**



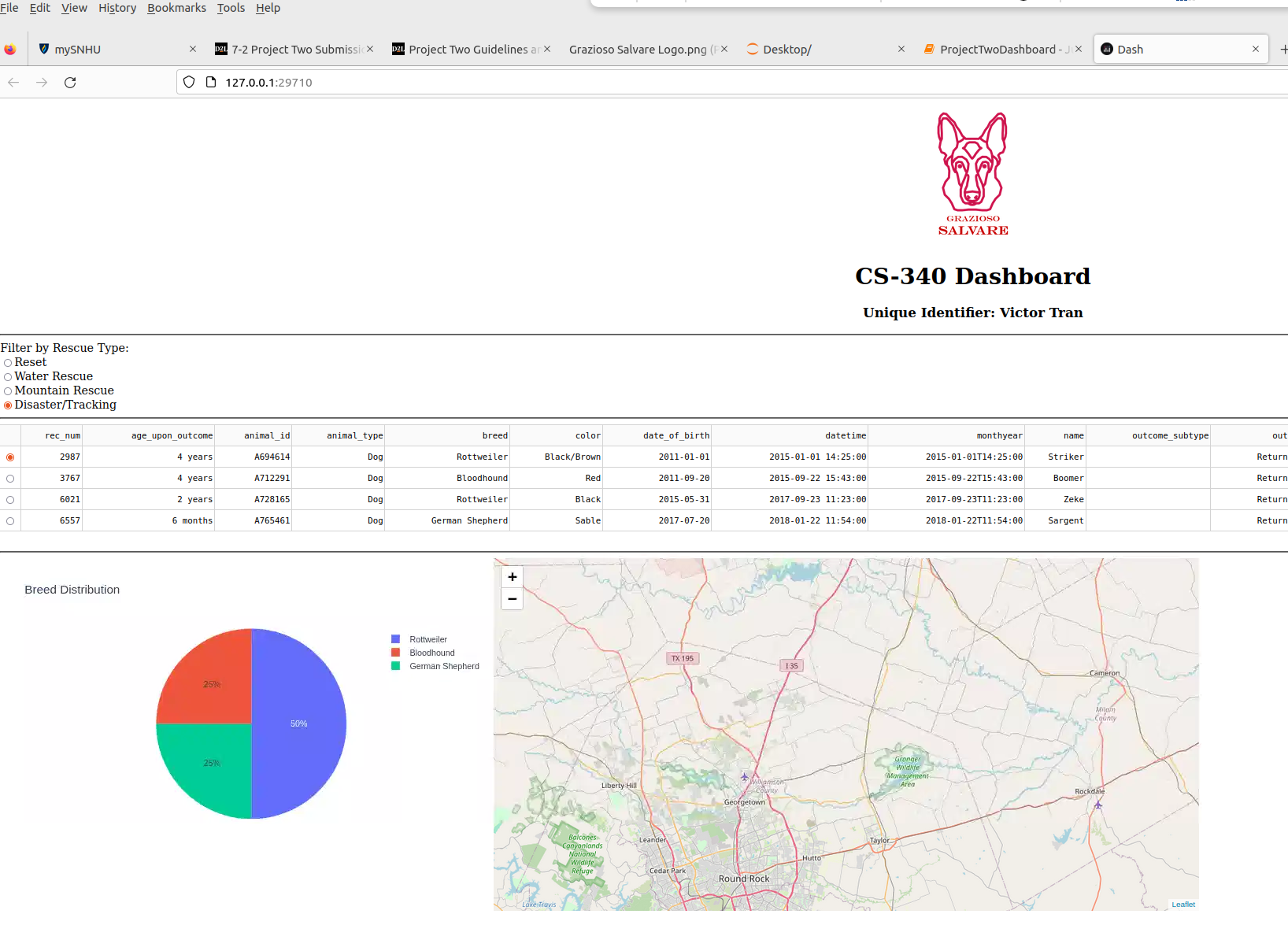
**Water Rescue Screenshot**



**Mountain Rescue**



**Disaster/Tracking**

****

## **Tools Used & Rationale**

### **MongoDB (Model Layer)**

MongoDB was chosen as the **database model** for its flexibility in handling large datasets of unstructured and semi-structured animal shelter data. It provides:

* **Easy querying with Python** via the **pymongo** library.
* **Scalability** for handling continuous updates to shelter data.
* **Fast lookups** with indexed search capabilities.

**Dash (View & Controller Layer)**

The **Dash framework** by Plotly was used to build the interactive web dashboard because:

* It provides **built-in interactivity** for data filtering and visualization.
* It allows **real-time updates** based on user inputs.
* It supports multiple **visual components** (tables, charts, and maps) in a single UI.

### **Other Tools Used**

* **Pandas**: For **data manipulation** and converting MongoDB query results into a DataFrame.
* **Plotly Express**: For creating **interactive pie charts** and **bar charts**.
* **Dash Leaflet**: For rendering **interactive maps** that display animal locations.
* **Jupyter Notebook**: For **running the dashboard** in a development environment.

## **Steps Taken to Complete the Project**

1. **Setup MongoDB Connection**
   * Connected to the **AAC shelter database** using **pymongo**.
   * Retrieved all records and structured them into a **Pandas DataFrame**.
2. **Implemented Filtering Mechanism**
   * Applied strict filters for **breed, sex, and age range** based on Grazioso Salvare’s **rescue type criteria**.
   * Ensured **only exact breeds listed** were included (e.g., allowing **"Labrador Retriever Mix"** but not **"Labrador Retriever / Another Breed"**).
3. **Developed Interactive Dashboard with Dash**
   * Created a **radio button selector** for filtering rescue types.
   * Implemented a **data table** that dynamically updates.
   * Built a **pie chart** for breed distribution.
   * Integrated a **map** to show dog locations.
4. **Handled Overcrowding in Pie Chart**
   * When "Reset" is selected, breeds with less than **2% frequency** are grouped into **"Other"**.
   * This ensures the pie chart remains readable.
5. **Ensured Proper Functionality & Deployment**
   * Verified that all filters correctly update the **table, chart, and map**.
   * Ran tests to ensure data was retrieved correctly from MongoDB.
   * Prepared documentation and screenshots for submission.

## **Challenges & Solutions**

### **1. Implementing Strict Breed Filtering**

* Challenge: The dataset contained **various breed combinations** such as **"Labrador Retriever / Other Breed"**, which made filtering difficult.
* Solution: Applied **strict filtering** to allow only **exact matches** based on Grazioso Salvare’s list.

### **2. Overcrowding in Pie Chart When Resetting**

* Challenge: When "Reset" was selected, the **pie chart displayed too many breeds**, making it unreadable.
* Solution: **Grouped breeds with less than 2% representation** into an **"Other"** category.

## **References**

* MongoDB. (n.d.). MongoDB Documentation. Retrieved from<https://www.mongodb.com/docs/>
* Plotly. (n.d.). Dash by Plotly Documentation. Retrieved from https://dash.plotly.com/
* The Pandas Development Team. (n.d.). Pandas Documentation. Retrieved from https://pandas.pydata.org/