# README: Grazioso Salvare Dashboard Project

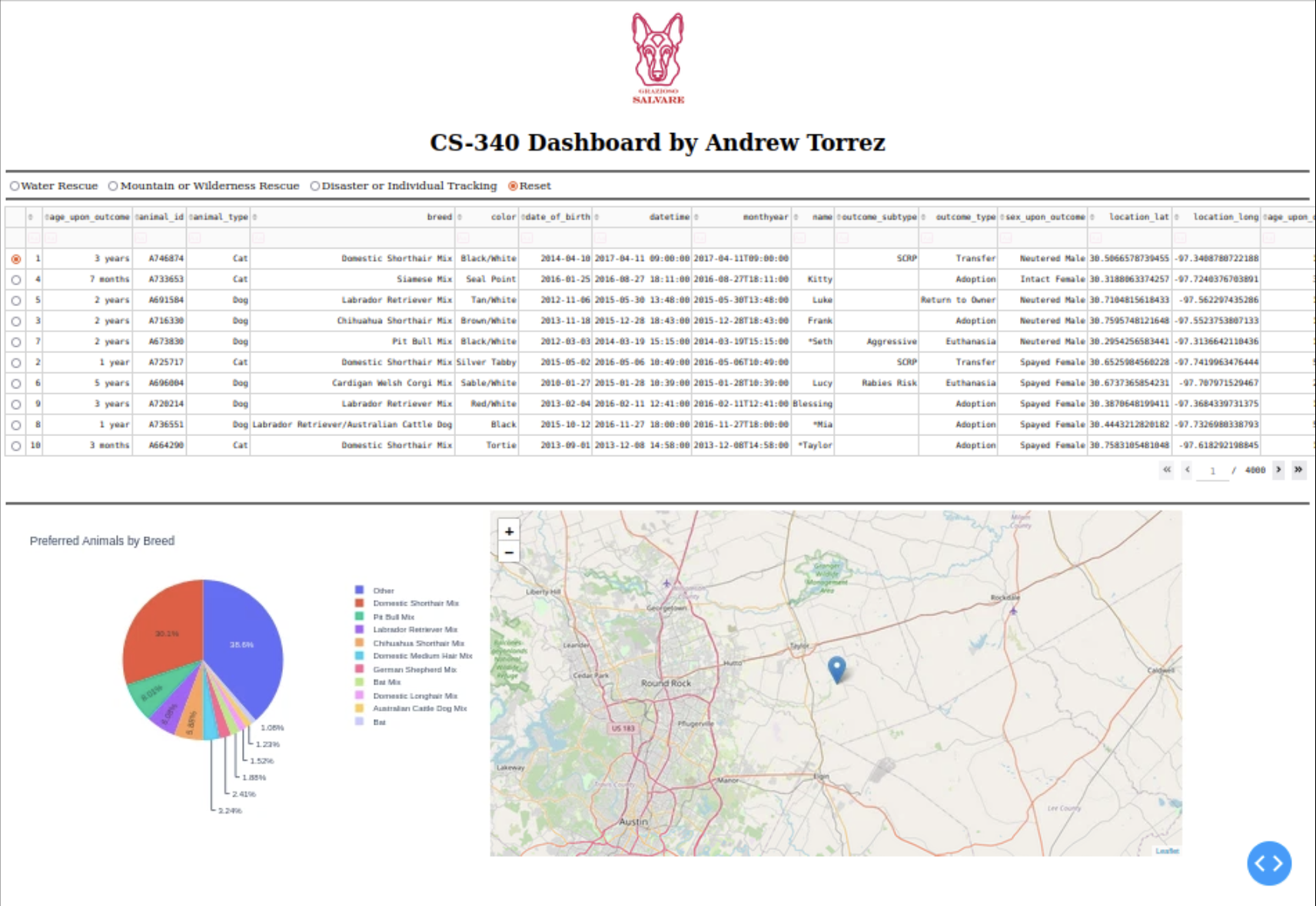
The Grazioso Salvare dashboard is a web application created to identify animals suitable for search-and-rescue training by allowing users to interactively filter data, visualize results, and gain insights. The dashboard incorporates several functionalities and tools to achieve the desired outcomes.

The project required the following features to meet its objectives:

The dashboard provides interactive filtering options for rescue types such as Water Rescue, Mountain or Wilderness Rescue, and Disaster or Individual Tracking. A reset option allows users to clear filters and view all data again. The data is displayed dynamically in a table that updates based on the selected filter. The geolocation map allows users to pinpoint animal locations on a visual interface. A pie chart further enhances data visualization by showing the distribution of animals by breed.

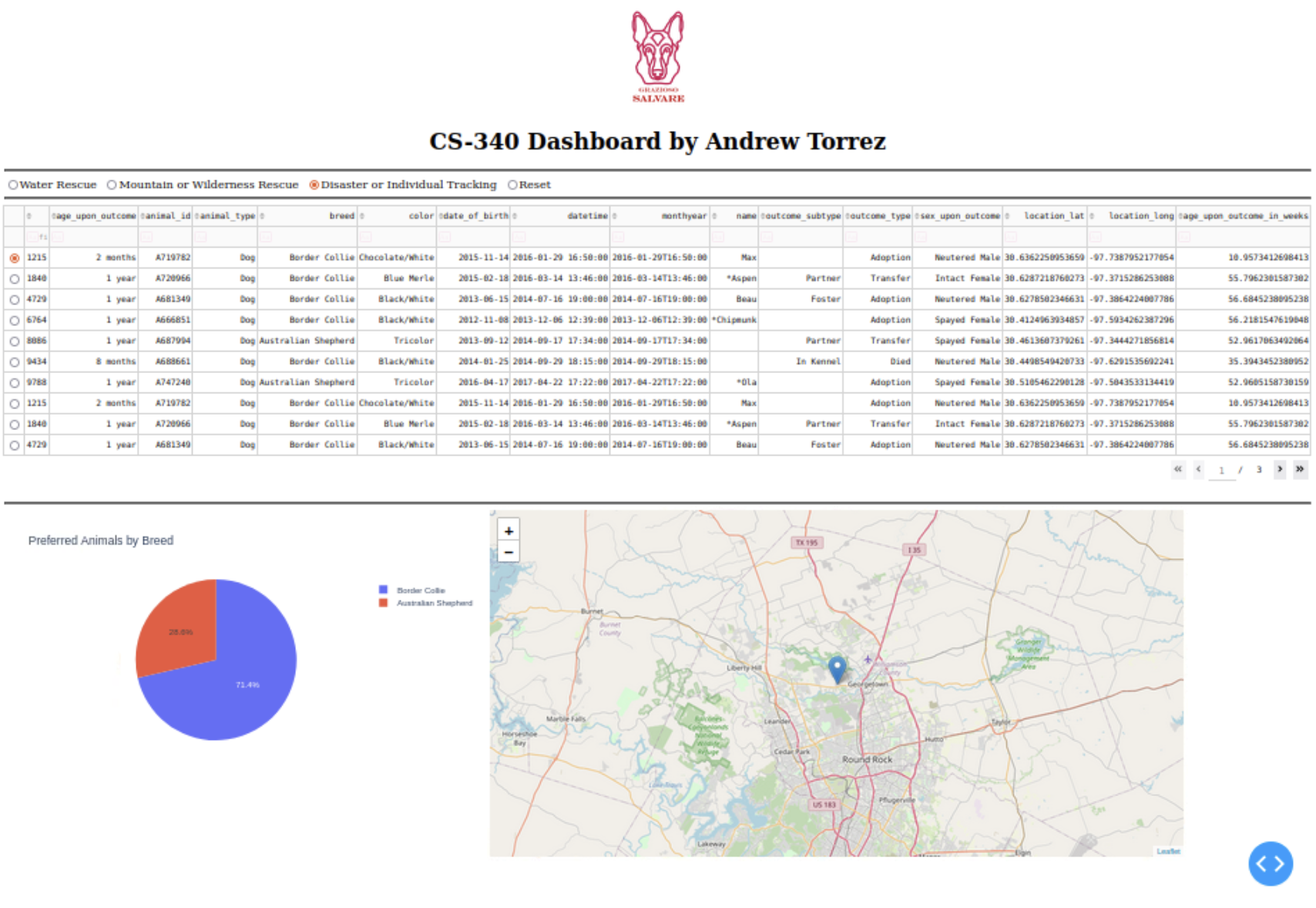
Screenshots of the dashboard demonstrating functionality:

### **Screenshot 1: Dashboard in Reset State (Unfiltered View)**



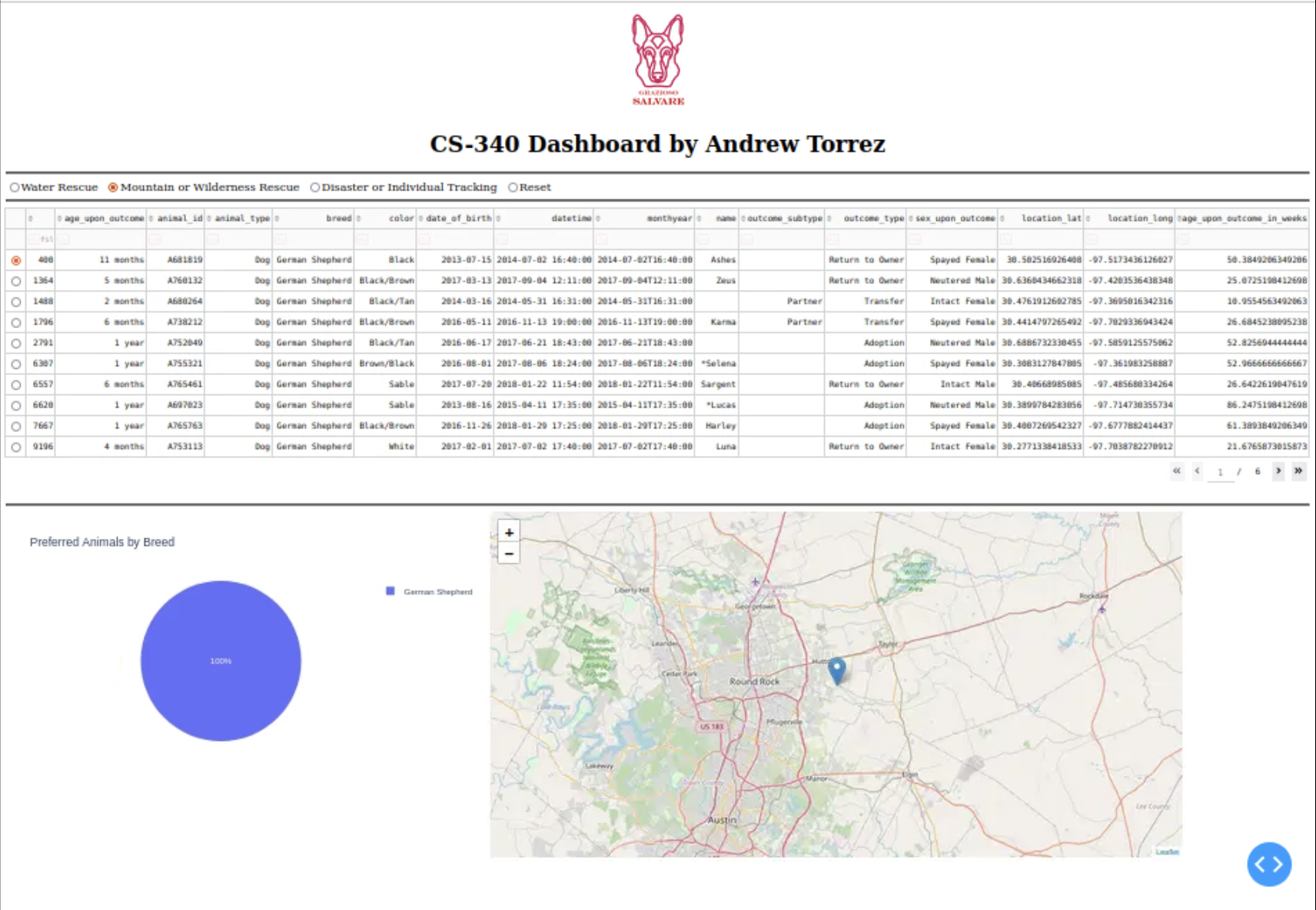
This screenshot shows the dashboard in its default, unfiltered state. All animal records from the dataset are displayed in the data table, offering a comprehensive view of the available data. The pie chart reflects the overall distribution of breeds across all rescue types, providing a high-level overview. The map visualizes the geolocations of all animals, allowing users to see the spread of available candidates.

### **Screenshot 2: Dashboard with "Disaster or Individual Tracking" Filter Applied**



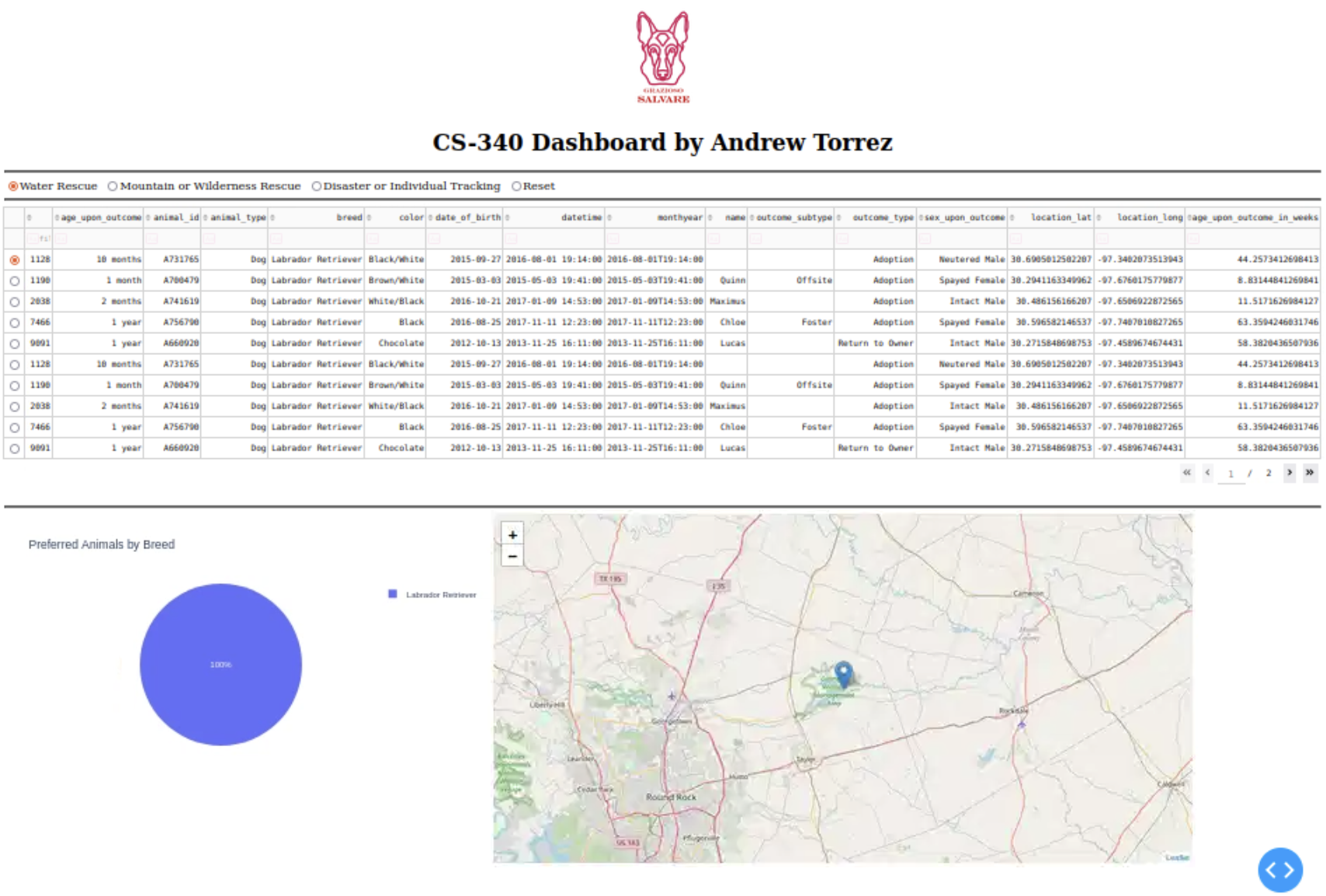
This screenshot demonstrates the functionality of filtering the data using the "Disaster or Individual Tracking" radio button. The table dynamically updates to show only the animals best suited for disaster or individual tracking rescue missions based on the criteria provided. The pie chart highlights the breeds most commonly selected for these missions, while the map pinpoints their geolocations.

### **Screenshot 3: Dashboard with "Mountain or Wilderness Rescue" Filter Applied**



Here, the dashboard showcases animals that are ideal candidates for mountain or wilderness rescue operations. The table, chart, and map reflect the filtered data. The pie chart emphasizes the distribution of breeds suited for this rescue type, while the map indicates their locations around the Austin area.

### **Screenshot 4: Dashboard with "Water Rescue" Filter Applied**



This view highlights the dashboard's functionality when the "Water Rescue" filter is selected. The data table is updated to show only animals suitable for water rescue missions. The pie chart reflects the percentage distribution of breeds within this category, while the map shows their corresponding geolocations.

# The tools used to build this dashboard include:

MongoDB serves as the backend database due to its flexible document-oriented structure, making it ideal for storing JSON-like data. It integrates seamlessly with Python via the PyMongo library, providing robust data management capabilities for filtering and retrieval.

The Dash framework was used to create the web application because of its simplicity and powerful interactive capabilities. Dash allows for seamless integration with Python while offering an extensive ecosystem, including Plotly for visualization and Dash Leaflet for mapping functionalities.

Additional libraries and tools were used, such as Dash Leaflet for map visualizations, Plotly Express for creating responsive pie charts, and Dash DataTable for interactive table rendering. Base64 encoding was employed to display the Grazioso Salvare logo.

# To complete the project, the following steps were taken:

First, the environment was set up by installing all the necessary Python libraries. This included configuring the MongoDB connection for efficient data handling. Next, a CRUD module was developed in Python to perform Create, Read, Update, and Delete operations on the MongoDB database. The dashboard layout was designed using Dash components. Key elements such as the logo, radio buttons for filtering, and a dynamically updating data table were included in the design.

Visualization components were added next. A pie chart was created to show breed distributions, while a geolocation map displayed animal locations dynamically based on selected filters. Callback functions were implemented to enable interactivity, ensuring that the filters, data table, and visualizations worked seamlessly together.

Extensive testing ensured the accuracy and reliability of each feature. Screenshots were captured to demonstrate the dashboard’s functionality across different filter states.

Several challenges were encountered during development. Initial filtering issues led to incomplete or incorrect results. This was resolved by refining MongoDB queries and validating data. The pie chart initially displayed incorrect percentages, which was fixed through data preprocessing. Geolocation maps occasionally failed to render correctly; adding checks for valid latitude and longitude values addressed this issue. Lastly, styling adjustments were made to resize the logo and improve the overall visual layout.