CS 340 Project Two - Grazioso Salvare Dashboard

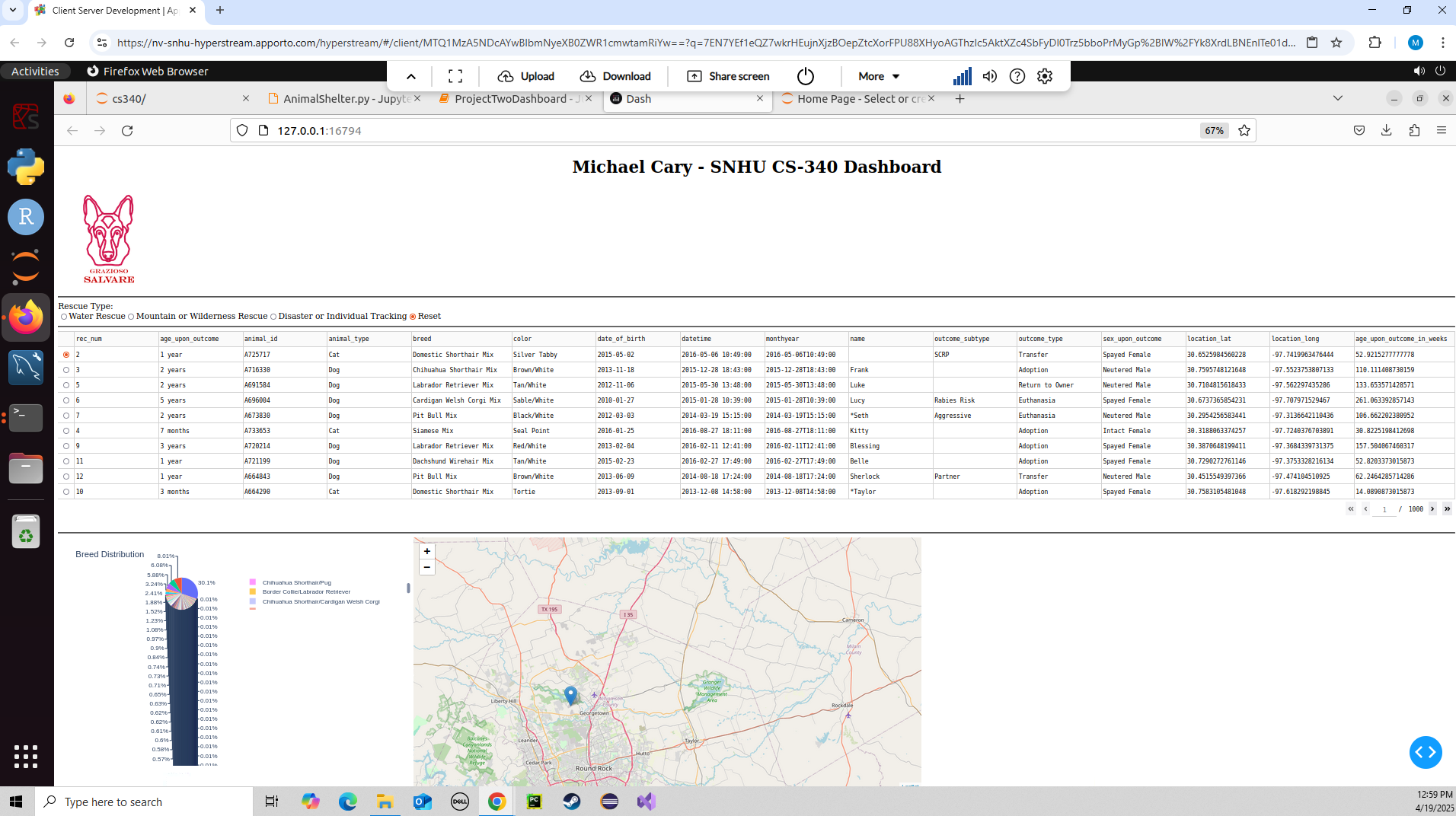
# Project Functionality

This dashboard application enables Grazioso Salvare to filter and visualize rescue animal data from the Austin Animal Center. Users can filter animals based on the type of rescue operation they are suited for, view breed distributions in a pie chart, and see their geolocations on a map. The dashboard dynamically updates based on user selections.

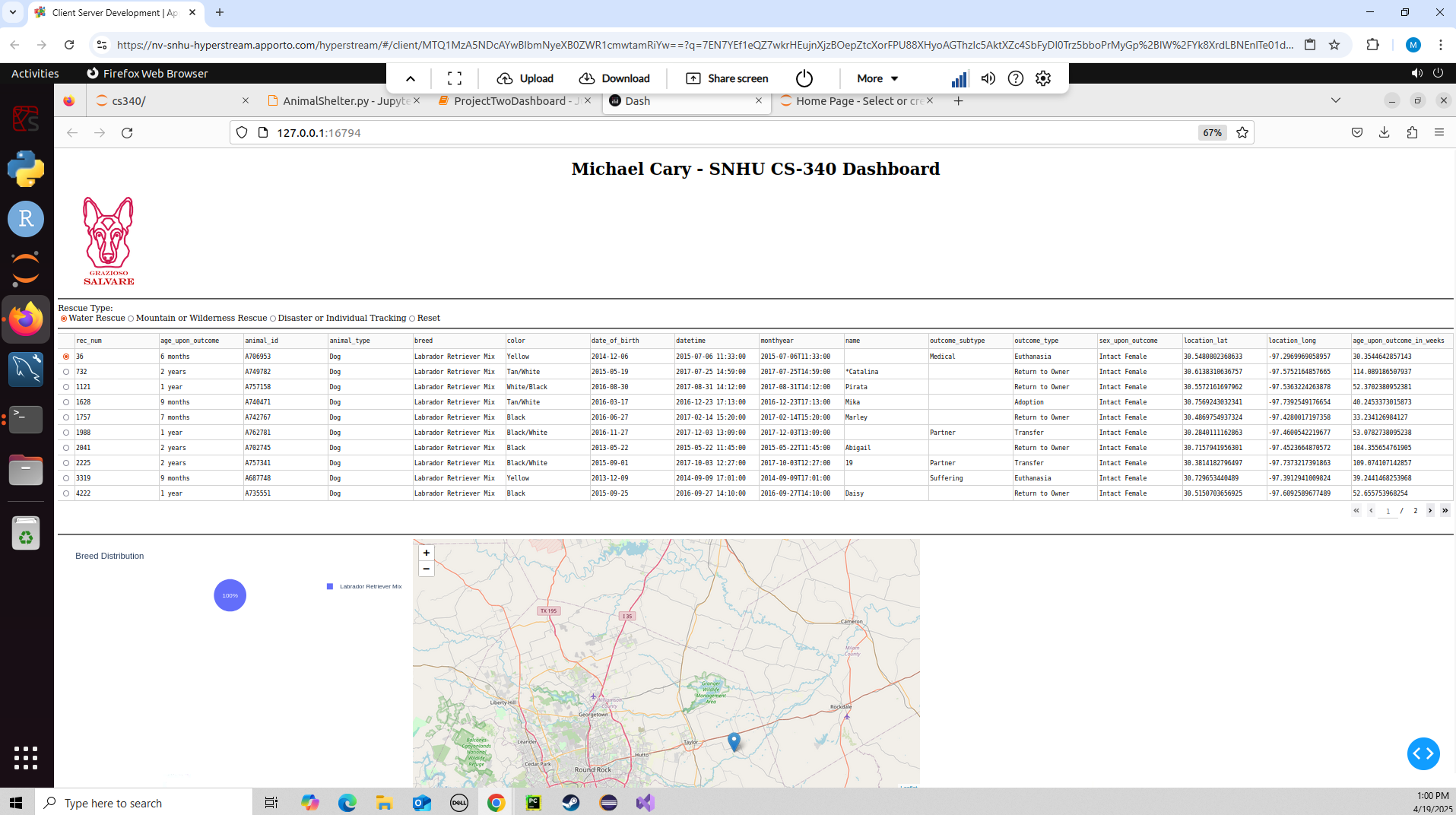
# Dashboard Screenshots

Below are screenshots that demonstrate the required dashboard functionality:

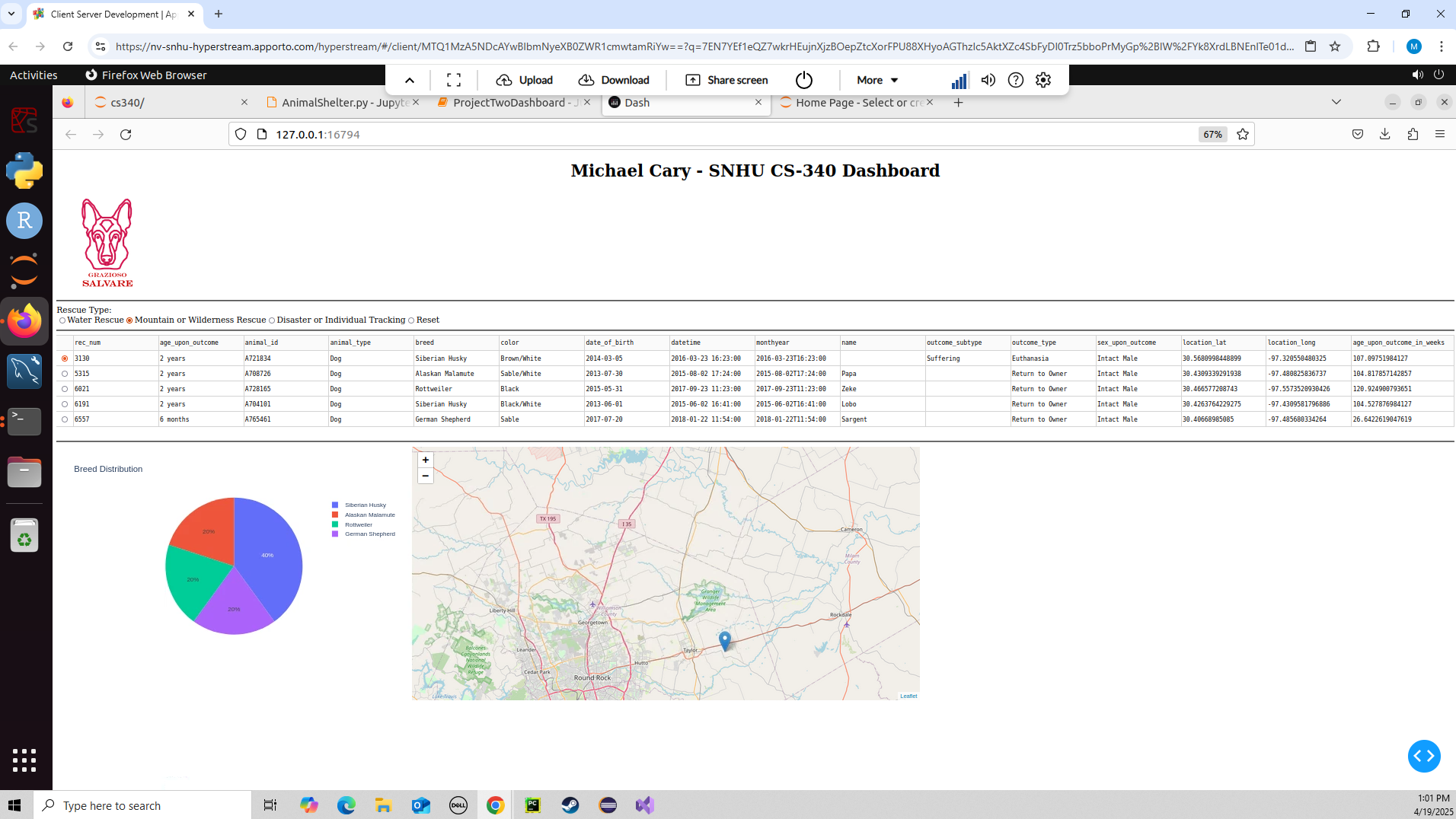
Initial dashboard view with all widgets



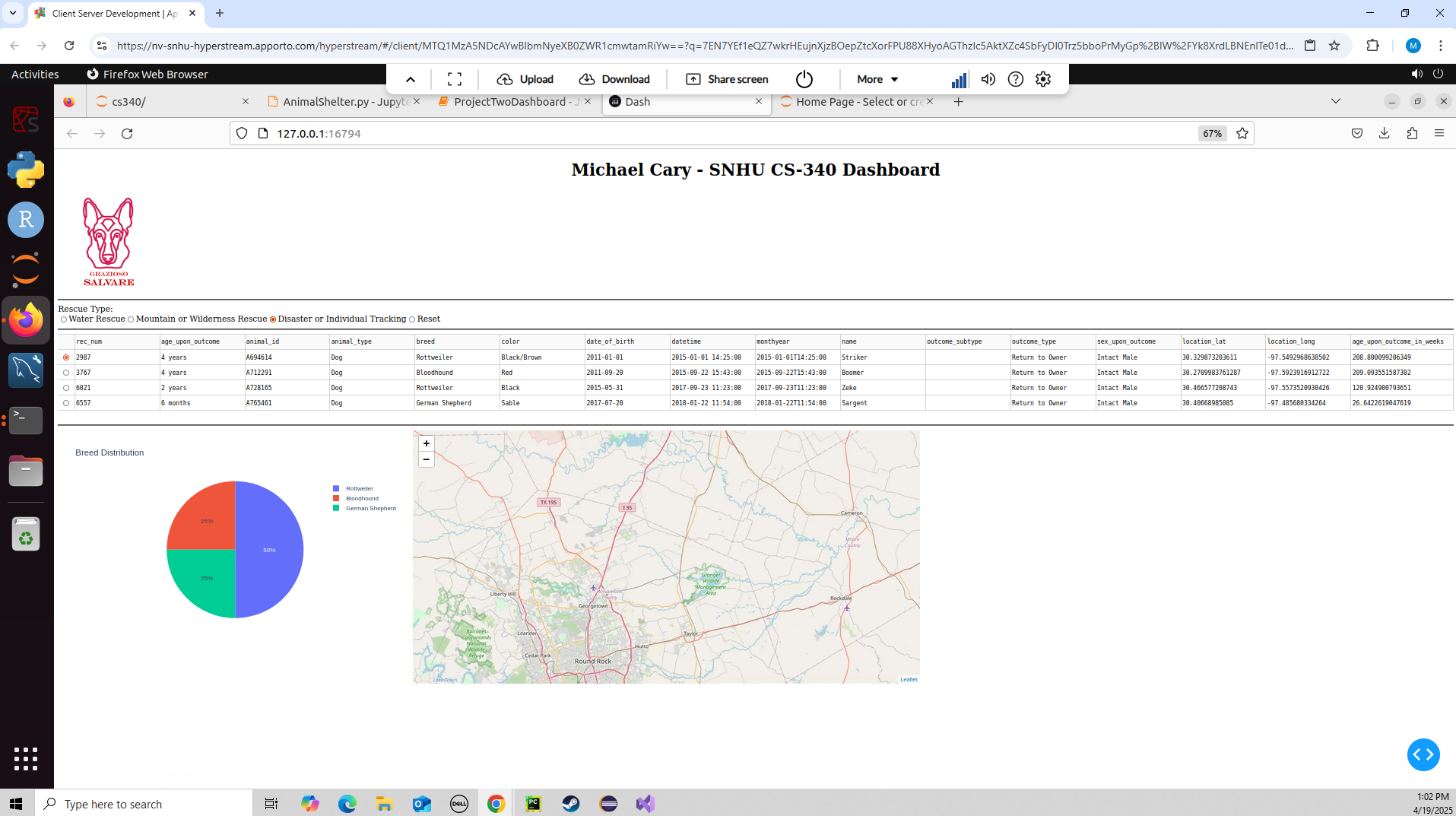
Dashboard view after filtering for Water Rescue



Dashboard view after filtering for Mountain or Wilderness Rescue



Dashboard view after filtering for Disaster or Individual Tracking



# Tools and Technologies Used

• MongoDB: Used as the backend database for its schema-less flexibility and integration with PyMongo.  
• Dash (Plotly): Used to build the web-based dashboard for its ease of use in creating interactive visualizations.  
• Python: Programming language used for data manipulation and application logic.  
• JupyterDash: Used to render the dashboard within a Jupyter Notebook environment.

# Implementation Steps

1. Loaded the Austin Animal Center data into MongoDB.  
2. Implemented CRUD operations using a custom Python module.  
3. Designed a dashboard layout in Dash using MVC architecture.  
4. Connected Dash components with callbacks to allow dynamic filtering and visualizations.  
5. Conducted functional testing and captured proof of functionality via screenshots.

# Challenges and Solutions

One challenge was formatting and displaying MongoDB’s ObjectId field in the Dash DataTable, which was resolved by removing the '\_id' column. Another challenge was ensuring that the dashboard updated in real-time with filtering options. This was overcome by carefully crafting callbacks that updated each visualization and data view synchronously.