## **Project Overview**

The Grazioso Salvare Rescue Dog Dashboard is a web-based application designed to help Global Rain identify dogs for specialized rescue training. The dashboard provides interactive filters, a data table, a geolocation chart, and a breed distribution chart to assist in selecting suitable rescue dogs based on age, breed, and rescue category.

This project follows the **Model-View-Controller (MVC)** design pattern:

* **Model:** MongoDB stores and retrieves dog data.
* **View & Controller:** Dash framework handles UI interactions and data processing.

## **Interactive Dashboard Features**

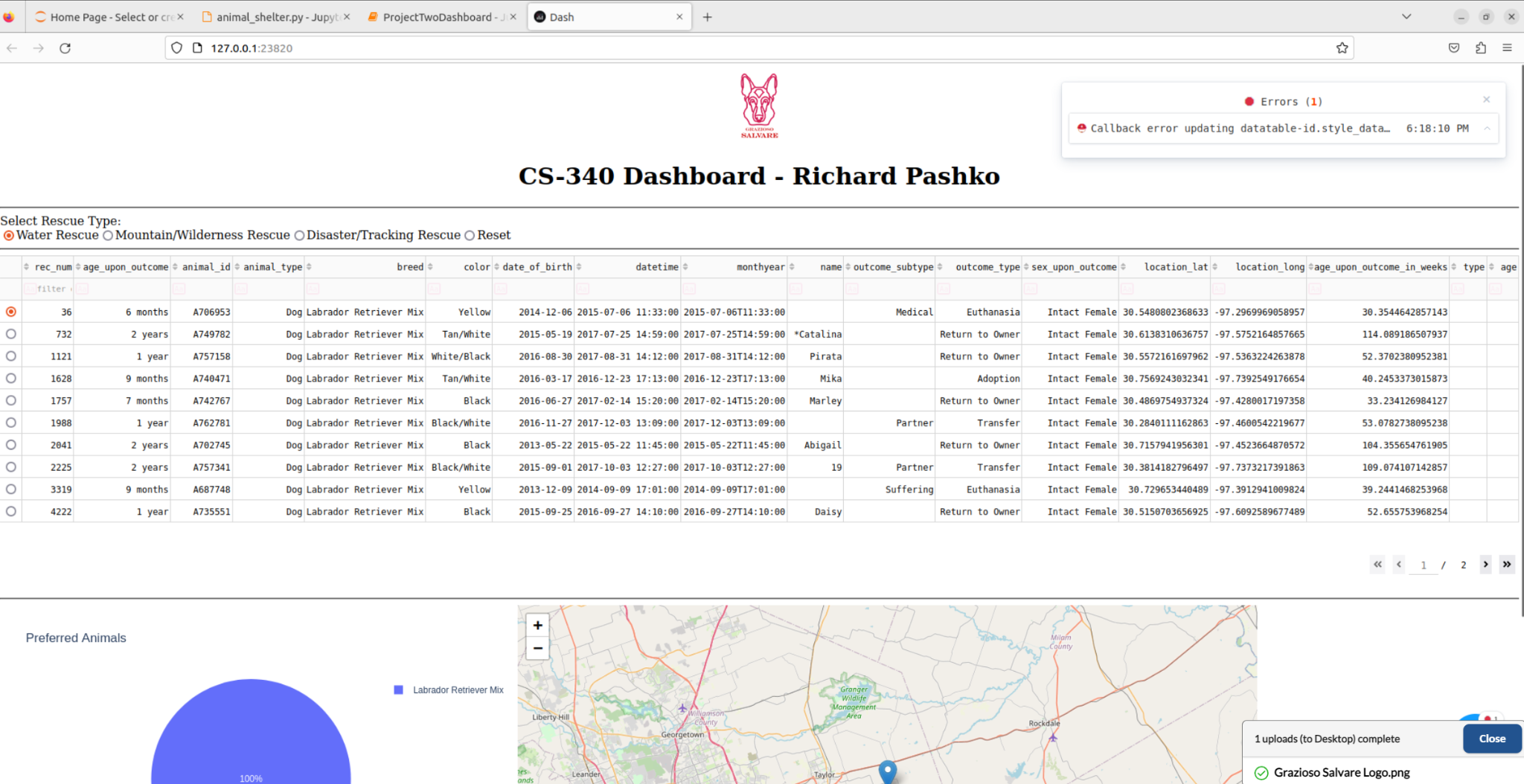
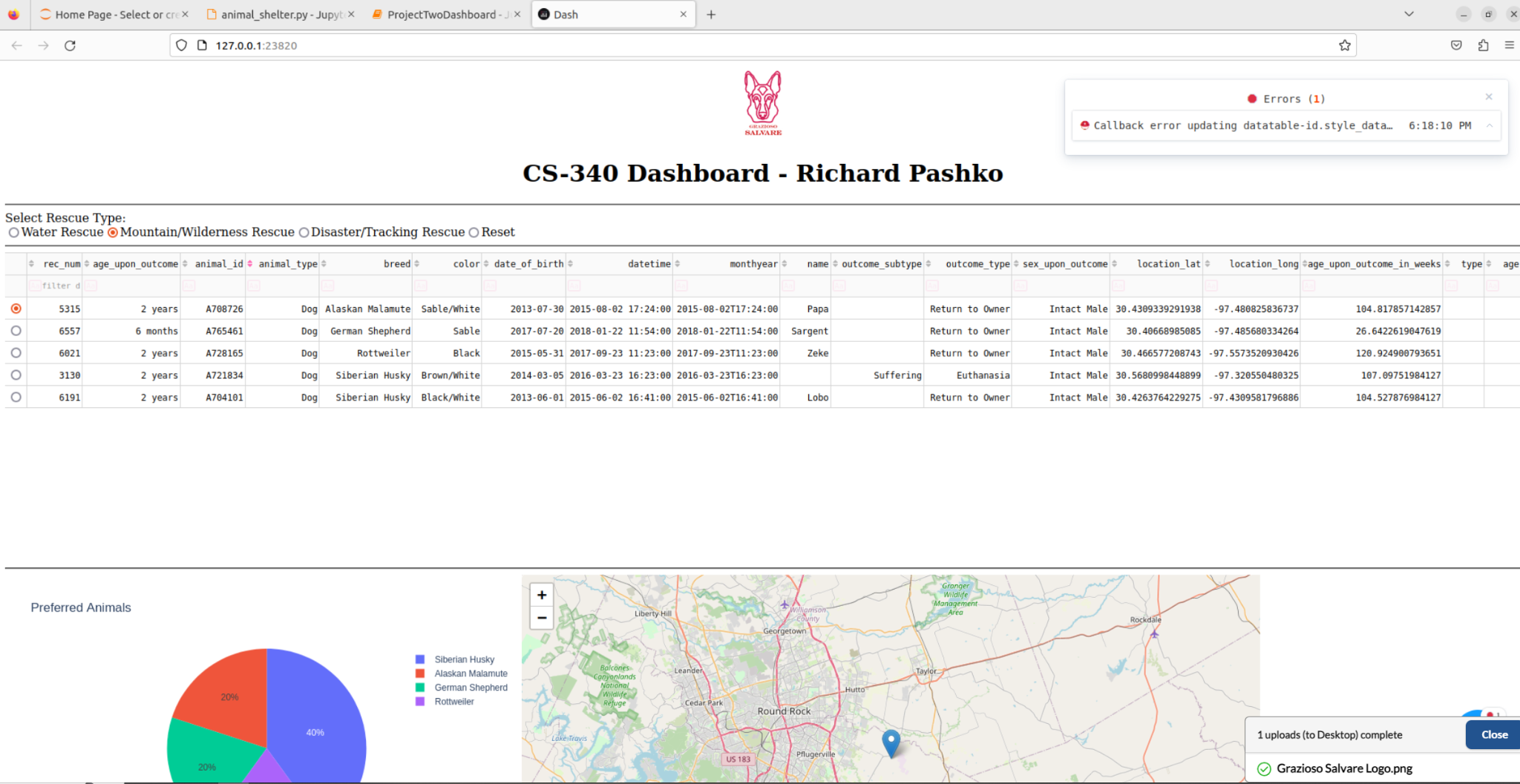
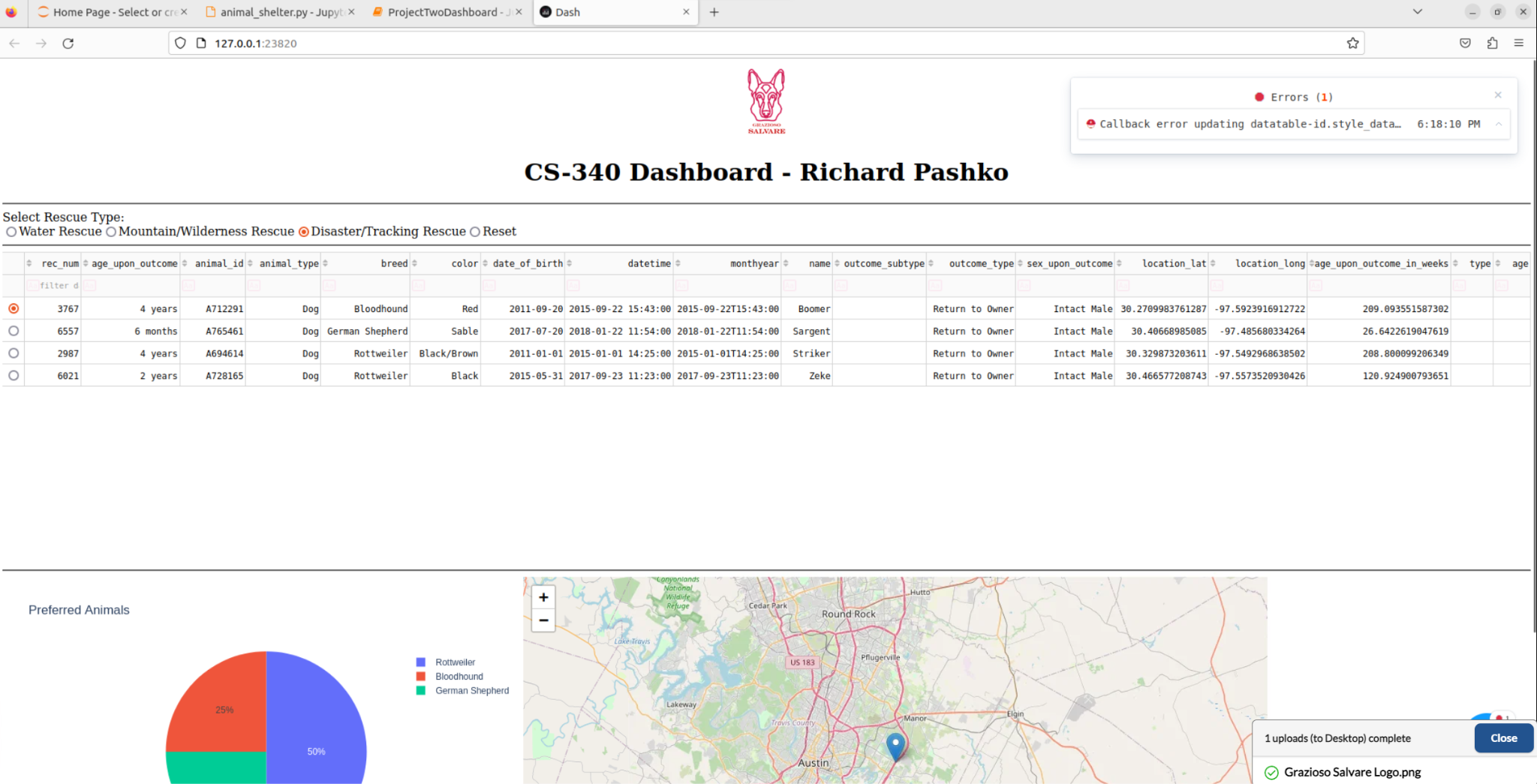
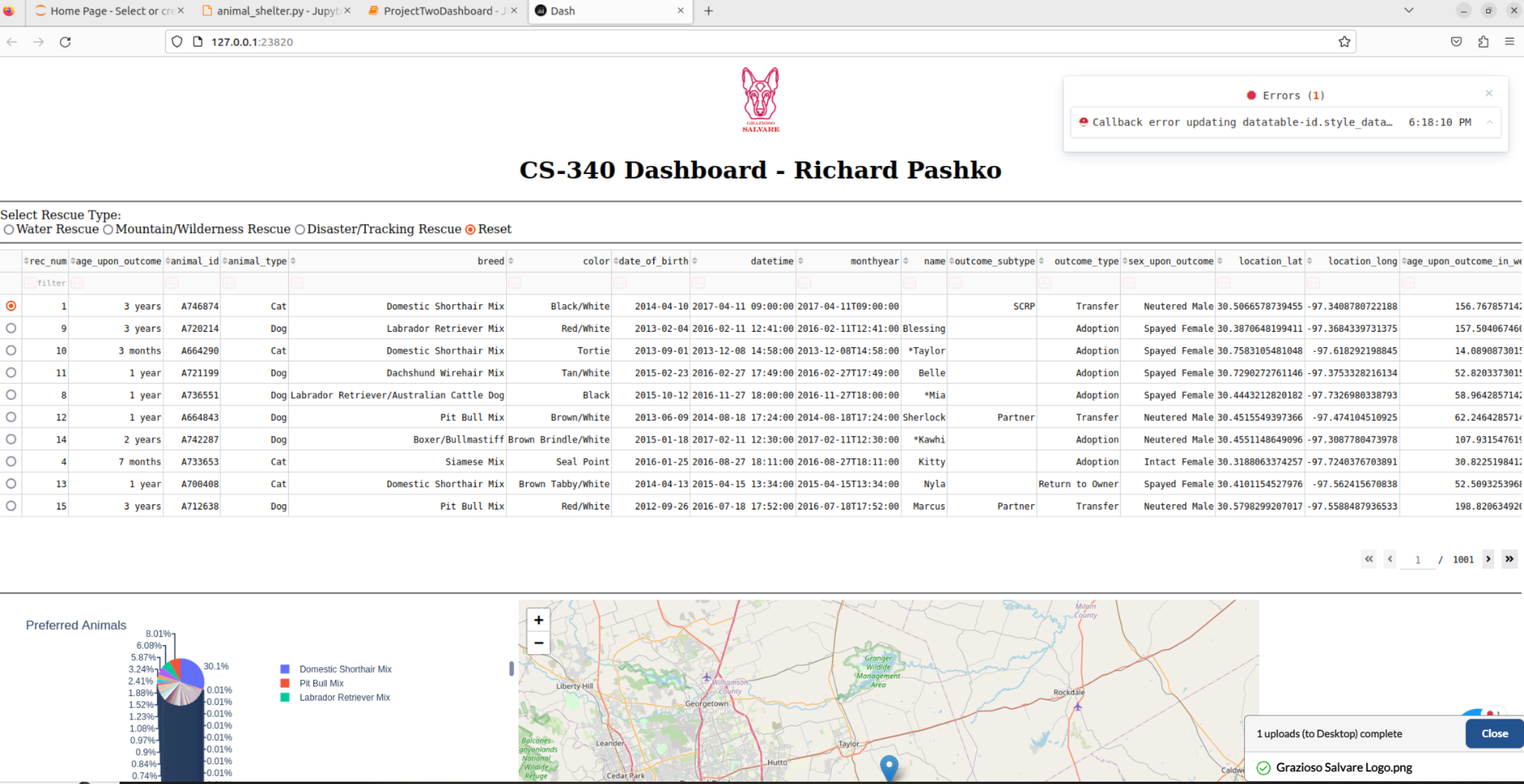
**Filtering Options:** Users can select Water Rescue, Wilderness Rescue, Disaster Rescue, or Reset to update results dynamically.

**Dynamic Data Table:** Displays shelter dog records and updates based on selected filters.

**Geolocation Map:** Highlights the location of selected dogs.

**Breed Distribution Chart:** Shows a Pie Chart summarizing available breeds.

## **Screenshots / Proof of Functionality**



## **Tools Used**

1. **MongoDB  
   A.** NoSQL database ideal for handling semi-structured animal shelter data.  
   **B.** Supports fast queries with flexible document storage.  
   **C.** Seamless integration with Python via pymongo.
2. **Dash Framework  
   A.** Allows building interactive web applications directly in Python.
3. **Python Libraries  
   A. pymongo** was used to connect to MongoDB and build CRUD functionality. **B. dash** used to build the web dashboard. **C. dash\_leaflet** displays the geological map.  
   **D. plotly.express** generates the interactive chart. **E. pandas** converted MongoDB documents into a structured DataFrame.

## **Installation and Setup**

1. Install the above python libraries.
2. Ensure MongoDB is running and contains the animal shelter dataset.
3. Run the python dashboard.
4. Open the Notebook & Execute Cells to start the web server.
5. Access the dashboard through the link provided under the cell.

## **Project Steps & Development**

1. Connected to MongoDB using] pymongo library.
2. Created a CRUD Python module to query shelter records.
3. Designed the dashboard UI with Dash and Leaflet maps.
4. Implemented filters for dog selection.
5. Developed data visualization for breed distribution.
6. Tested & debugged for errors in MongoDB.

## **Project Steps & Development**

| **Challenges** | **Solution** |
| --- | --- |
| Auth error with MongoDB | Verified user credentials, granted readwrite access. |
| No \_id column found in DataFrame | Handled empty queries with exception handling |

## **Resources & References**

* **MongoDB**: <https://www.mongodb.com/docs/v6.0/tutorial/enable-authentication/>
* **Dash Docs**: <https://dash.plotly.com>
* **Plotly Express Docs**: <https://plotly.com/python/plotly-express/>