**Project Overview:**

The Grazioso Salvare Dashboard is an interactive web application designed to analyze and visualize animal shelter data from the Austin Animal Center Outcomes dataset. Built using the Model-View-Controller architecture, the dashboard integrates MongoDB as the database, Dash for front-end visualization and Python CRUD operations to manage data interactions.

**Functionality:**

The dashboard enables end users to perform the following actions.

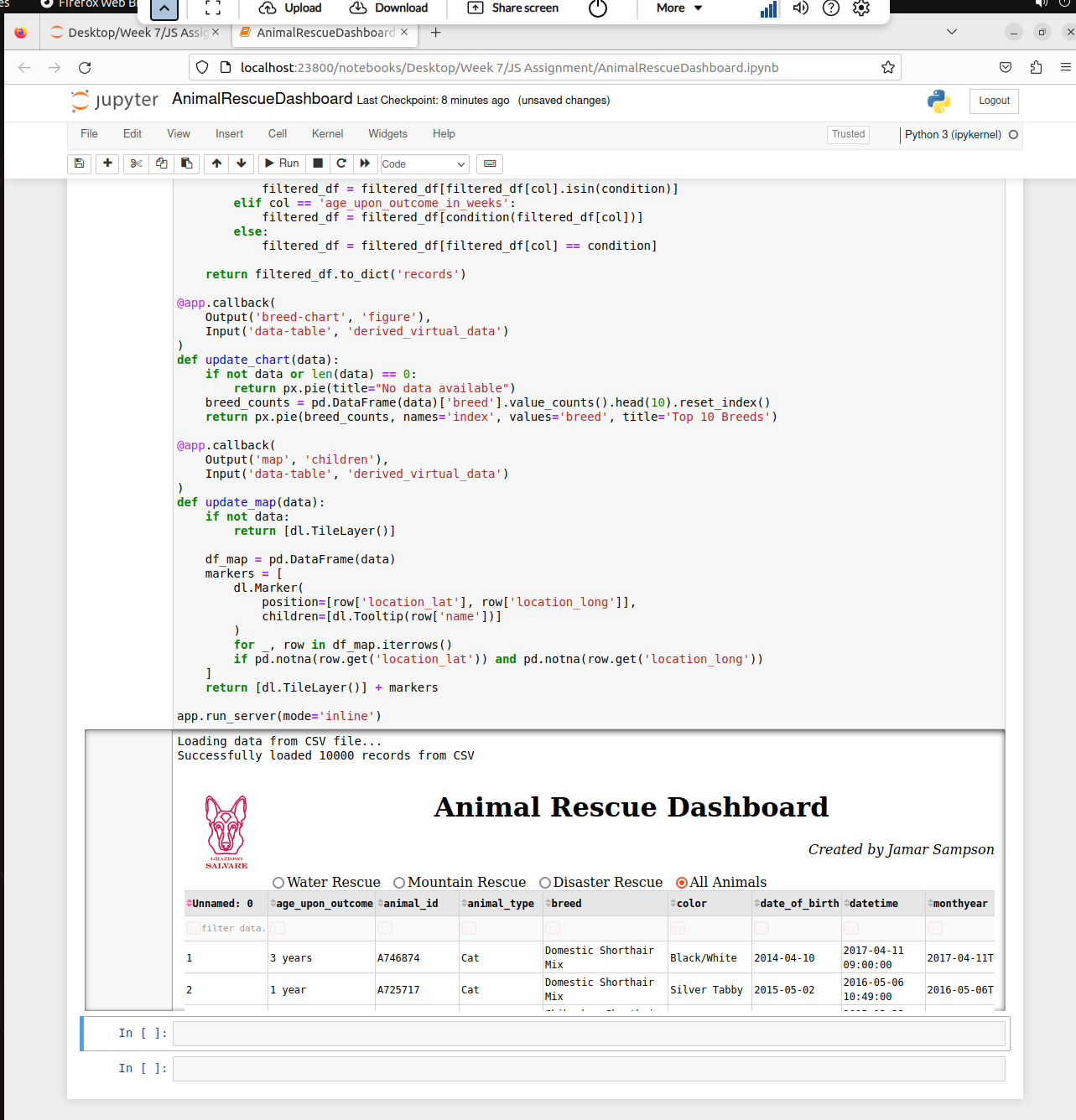
* View unfiltered animal shelter data in a dynamic table.
* Apply different specialized filters for different rescue scenarios to meet project goals.
* Visualize data through a learning interactive chart with geolocation mapping.

**Required Functionality:**

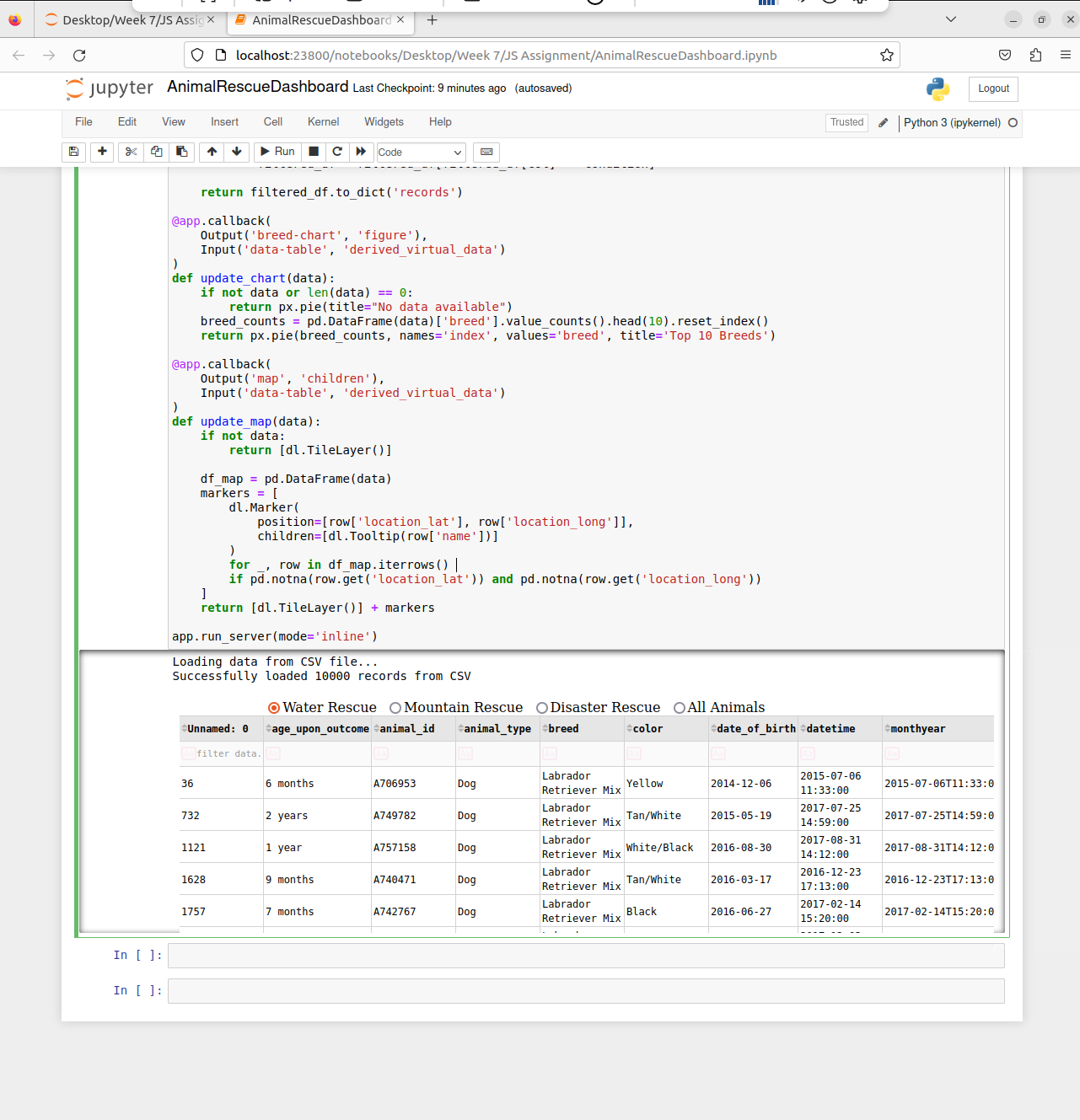
The dashboard fulfills the following key requirements:

1. **Unfiltered Data Table** - Displays the complete dataset from the Austin Animal Center through sorting & search functionality.
2. **Rescue-Specific Filtering** - **Water Rescue**, Mountain & Wilderness Rescue, Disaster & Individual tracking.
3. **Interactive Visualizations** - **Geolocation Map**: Displays animal locations with clickable markers with selectable breeding distribution chart in a pie graph.
4. **Dynamic Updates** - All charts & graphs update in real-time when filters are applied.

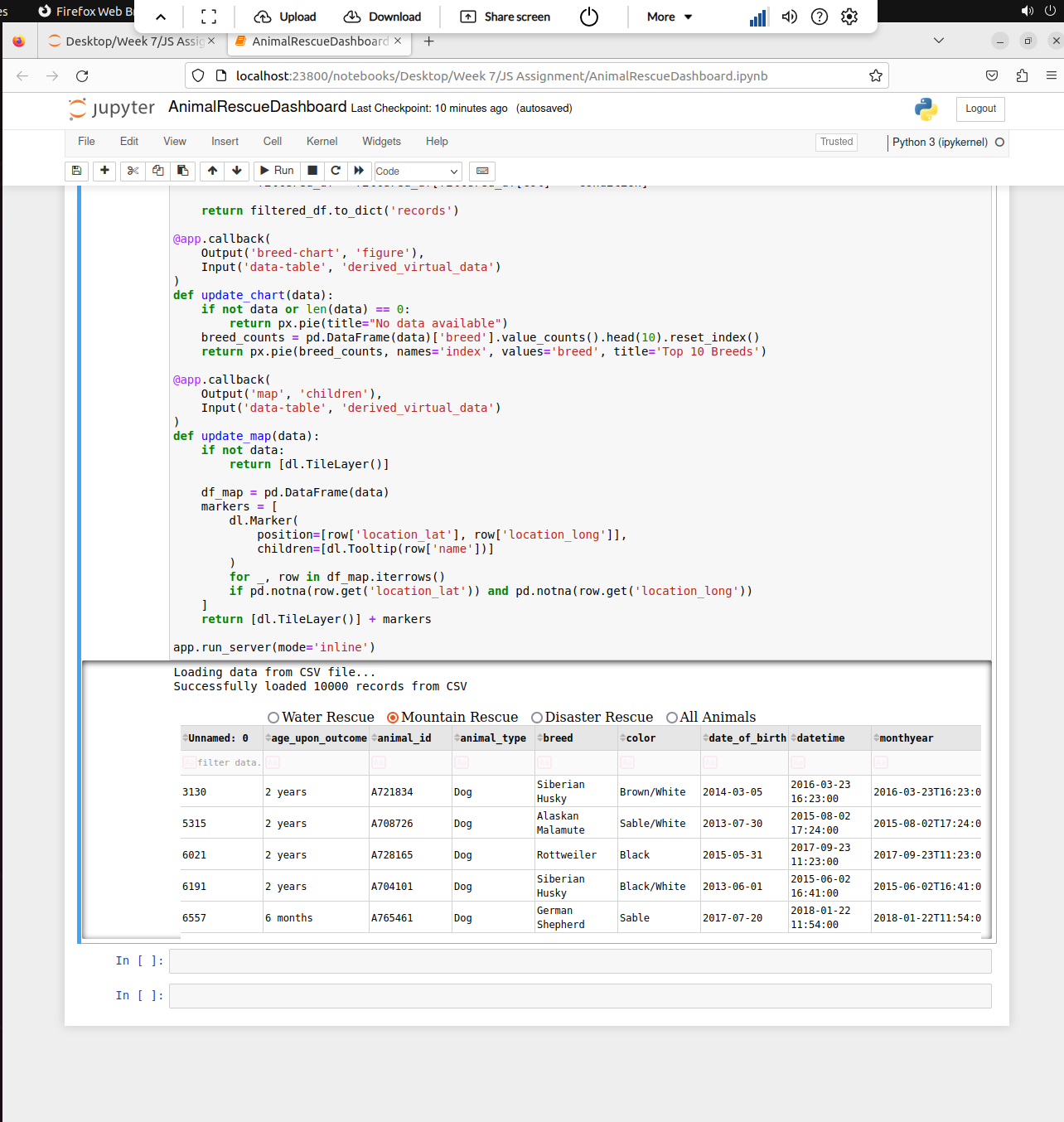
Screenshots of Functionality Below.



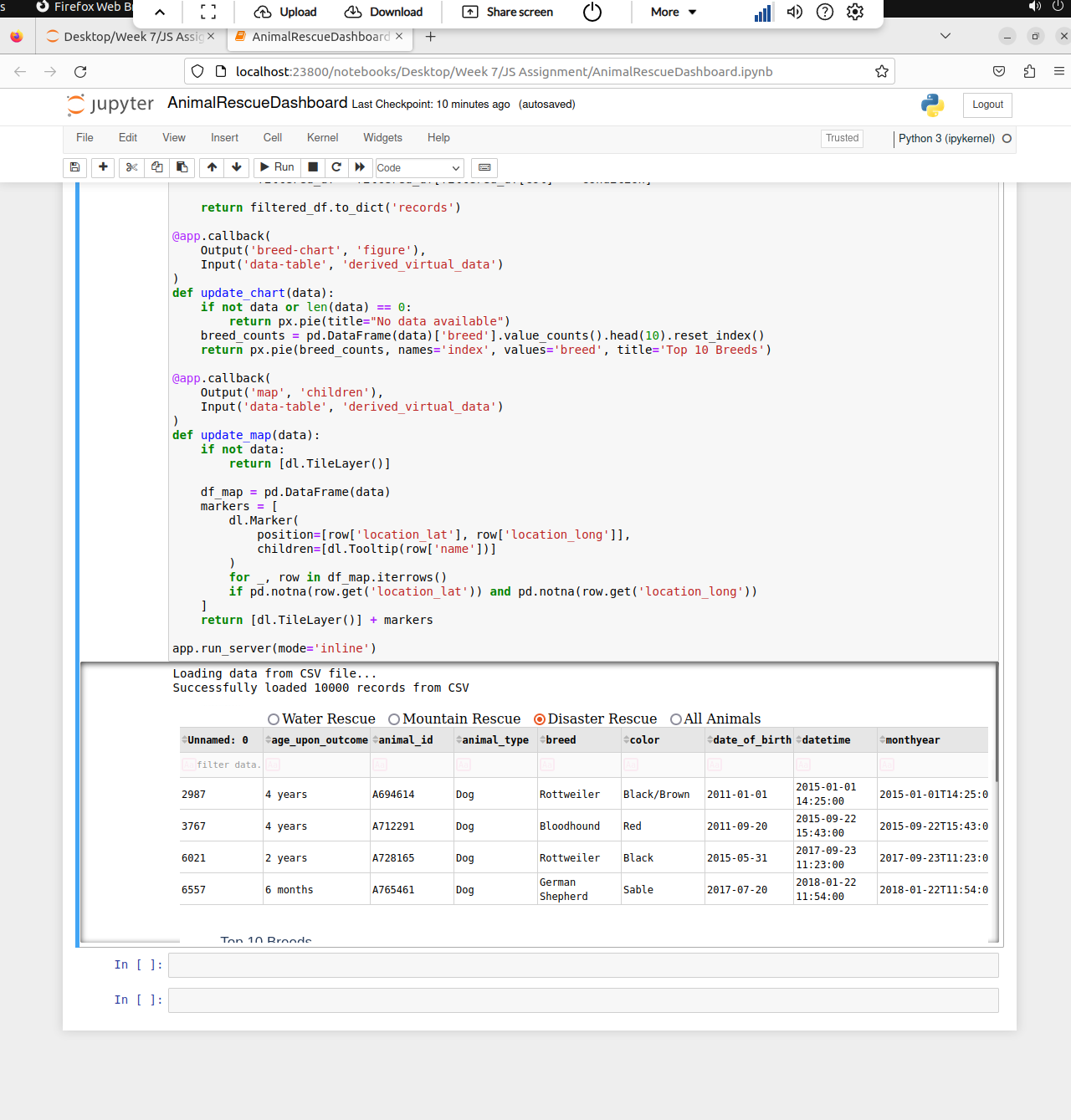
Initial Rescue Build.



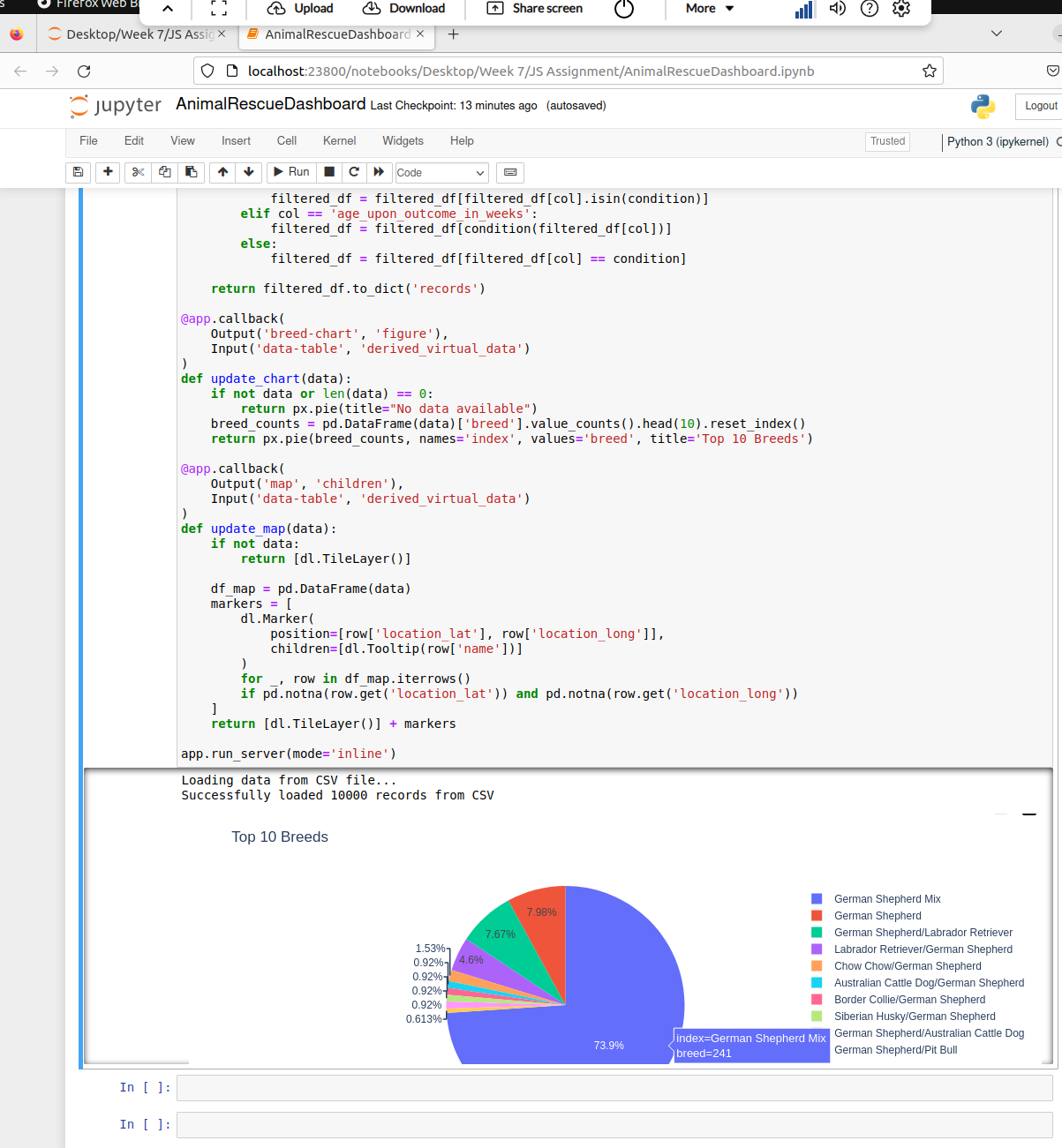
Water Rescue.



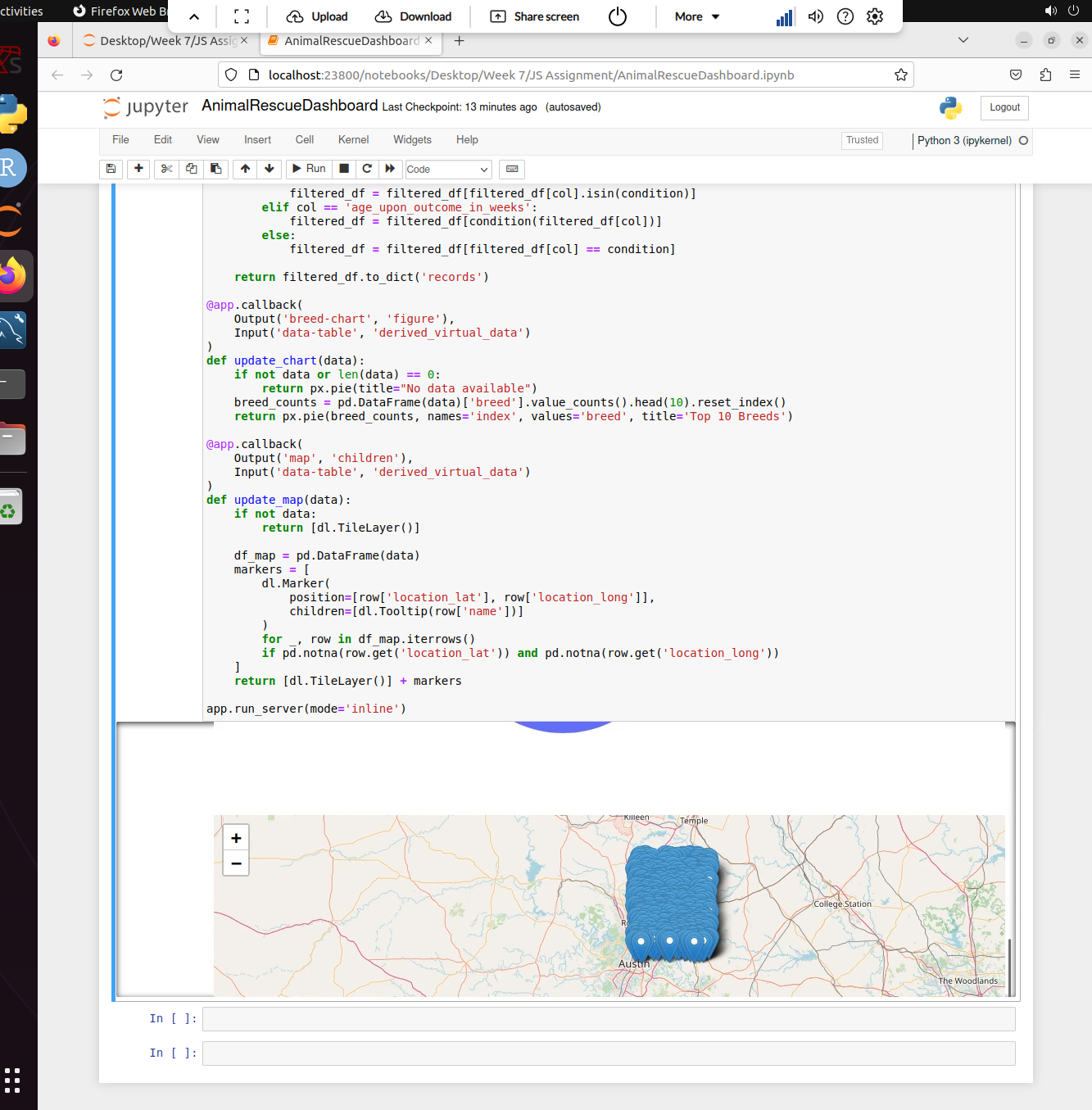
Mountain & Wilderness Rescue.



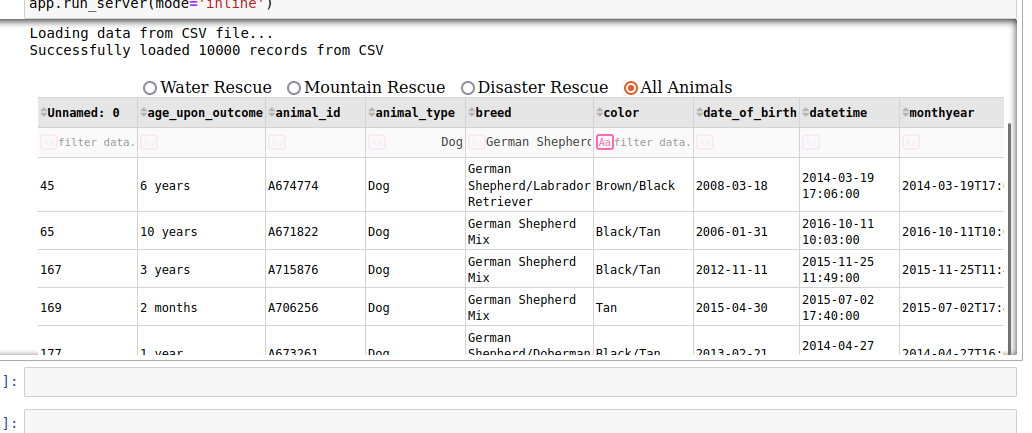
Disaster & individual Tracking.



Updating Pie Chart.



Engageable Map.



Working Rescue Filter.

**Tools Used:**

1. **Dash & JupyterDash –** Used for building the interactive web application.
2. **Pandas –** Used for data loading & cleaning from the csv. Filters & processes for the table & charts.
3. **Plotly Express –** Generates the pie chart.
4. **Dash Leaflet –** Renders the map with the animal location markers.
5. **Base64 –** Used to encode the logo image for the display.
6. **MongoDB** – Used as the basis for the model component due to its flexibility, scalability, & ease of use when integration with Python.

**Reasoning for the Tool Selection:**

I choose these tools for the database primarily for the ease of development, interactivity, & compatibility with Python-based data workflows. Dash (JupyterDash) was selected as the core framework because it allows the creation of interactive web applications directly from Python, eliminating the need for complex JavaScript front-end development. Its seamless integration with Plotly Express enables dynamic, publication-quality visualizations with minimal code, while Dash Leaflet provides an intuitive way to render geospatial data without external mapping APIs.

For data handling, Pandas was the natural choice due to their powerful CSV processing capabilities, enabling efficient filtering, sorting, and transformation of the Austin Animal Center dataset. While the initial design included MongoDB for scalable document storage, the final implementation opted for a CSV-based approach to simplify deployment and reduce dependencies. This shift kept my functionality while streamlining the architecture, ensuring the dashboard remains lightweight and easy to reproduce. The combination of these tools delivered a responsive, user-friendly interface that met all client requirements without unnecessary complexity.

**Steps Taken for Project Completion:**

1. Load the animal shelter data from a CSV file using Pandas.
2. Check that all important columns (like names, breeds, and locations) were present.
3. Convert age formats (like "3 years") into weeks to make filtering easier.
4. Used Dash to create a web framework.
5. Add the company logo and my name as developer.
6. Made sure to include the simple radio buttons to filter by rescue type.
7. Created an interactive table with sorting and pagination.
8. Displayed the pie chart showing the most common breeds.
9. Displayed the map to display animal locations.
10. Programmed the filters to update the table and charts automatically.
11. Made sure that all the rescue type filters worked correctly.

The final result was a working dashboard that helps Grazioso Salvare quickly find animals suited for different rescue missions.