**Grazioso Salvare - Dashboard Project**

**Project Overview**

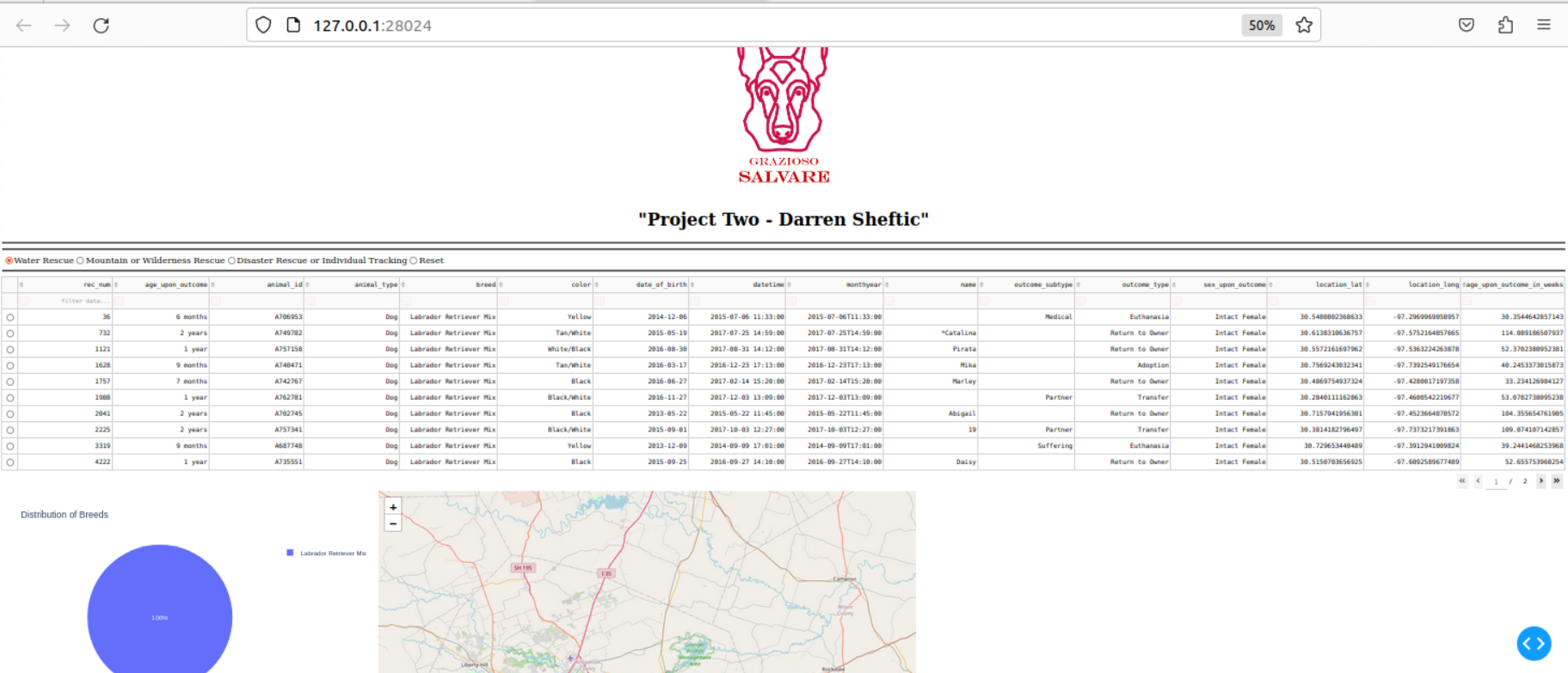
The Grazioso Salvare Dashboard is designed to enable the visualization and interaction with animal data from the Austin Animal Center. This project provides rescue staff with a user-friendly web application to filter, view, and analyze data regarding animals in the shelter, which is crucial for identifying suitable candidates for various types of rescue operations.

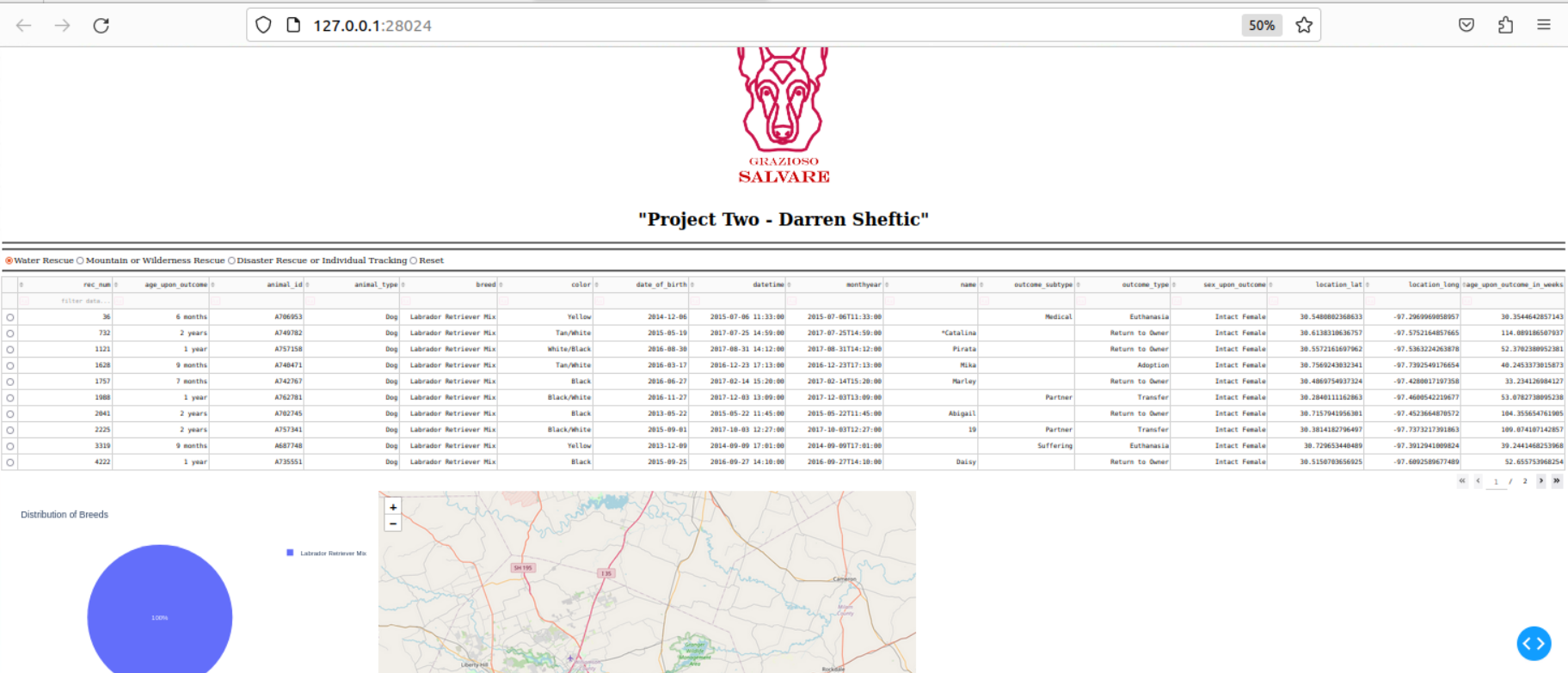
**Required Functionality**

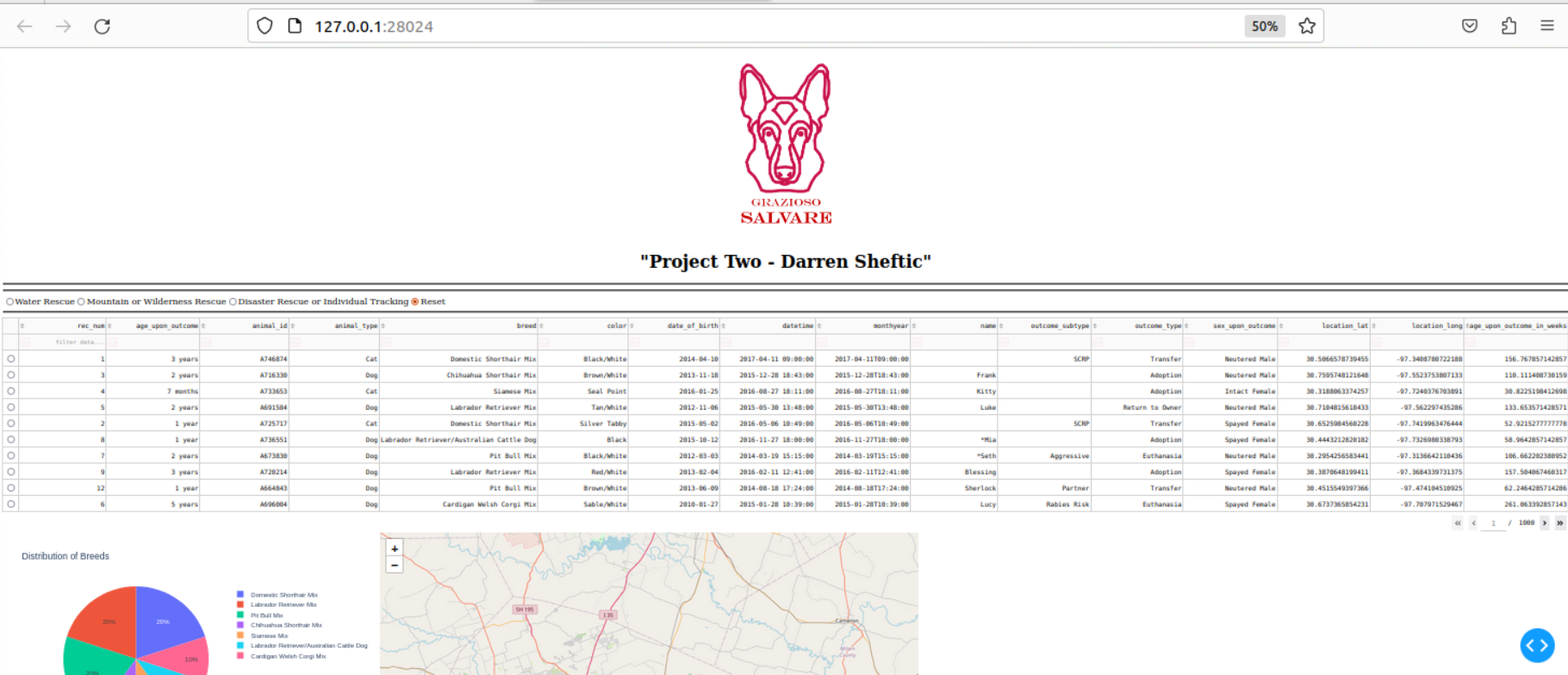
The dashboard includes the following functionalities:

* Interactive filters to select animals based on rescue type (Water Rescue, Mountain or Wilderness Rescue, Disaster or Individual Tracking).
* A dynamic data table that updates based on the selected filters.
* A geolocation map that displays the location of animals based on the filter applied.
* A pie chart that provides a visual distribution of animal breeds within the filtered results.

**Screenshots/Screencast:**



****

****

**Tools and Technologies Used**

* **Python**: The primary programming language used to build the module for database logic.
* **Dash by Plotly**: A Python web application framework used to build the client-side of the dashboard.
* **MongoDB**: Chosen for its flexibility with large volumes of data and its powerful querying capabilities.
* **Plotly**: Used for creating interactive charts. Plotly integrates with Dash, providing dynamic data visualization that updates based on user interactions.

**Rationale for Tool Choices**

* **MongoDB** was selected due to its flexible and scalable nature, which allows for storing large amounts of animal data efficiently. Its ability to handle large datasets and perform fast retrievals is key in a responsive dashboard environment.
* **Dash** provides a straightforward way to build a user interface with Python code, making it an excellent choice for projects requiring rapid development and prototyping of data-based applications.

**Project Reproduction Steps**

1. **Environment Setup**: Ensure Python and MongoDB are installed on your system. You can download them from:
   * Python: <https://www.python.org/downloads/>
   * MongoDB: <https://www.mongodb.com/try/download/community>
2. **Python Dependencies Used:**

dash pandas plotly dash-leaflet jupyter-dash pymongo

1. **Run The Application:**

ProjectTwoDashboard.ipynb

1. **Open the Dashboard**: Access the dashboard via a web browser at **http://127.0.0.1:8050**.
2. **Challenges and Solutions**
   * **Data Integration**: One challenge was integrating real-time data updates from MongoDB into the Dash application. This was addressed by using PyMongo to establish a live data connection between MongoDB and the Dash application.
   * **Responsive Design**: Ensuring the dashboard was responsive and performed well with large datasets involved optimizing MongoDB queries and utilizing efficient data modules in Python.
3. **Links and Resources**

Dash Documentation: https://dash.plotly.com/

MongoDB Python Library: <https://pymongo.readthedocs.io/en/stable/>