# **Grazioso Salvare Dashboard**

## About the Project / Project Title

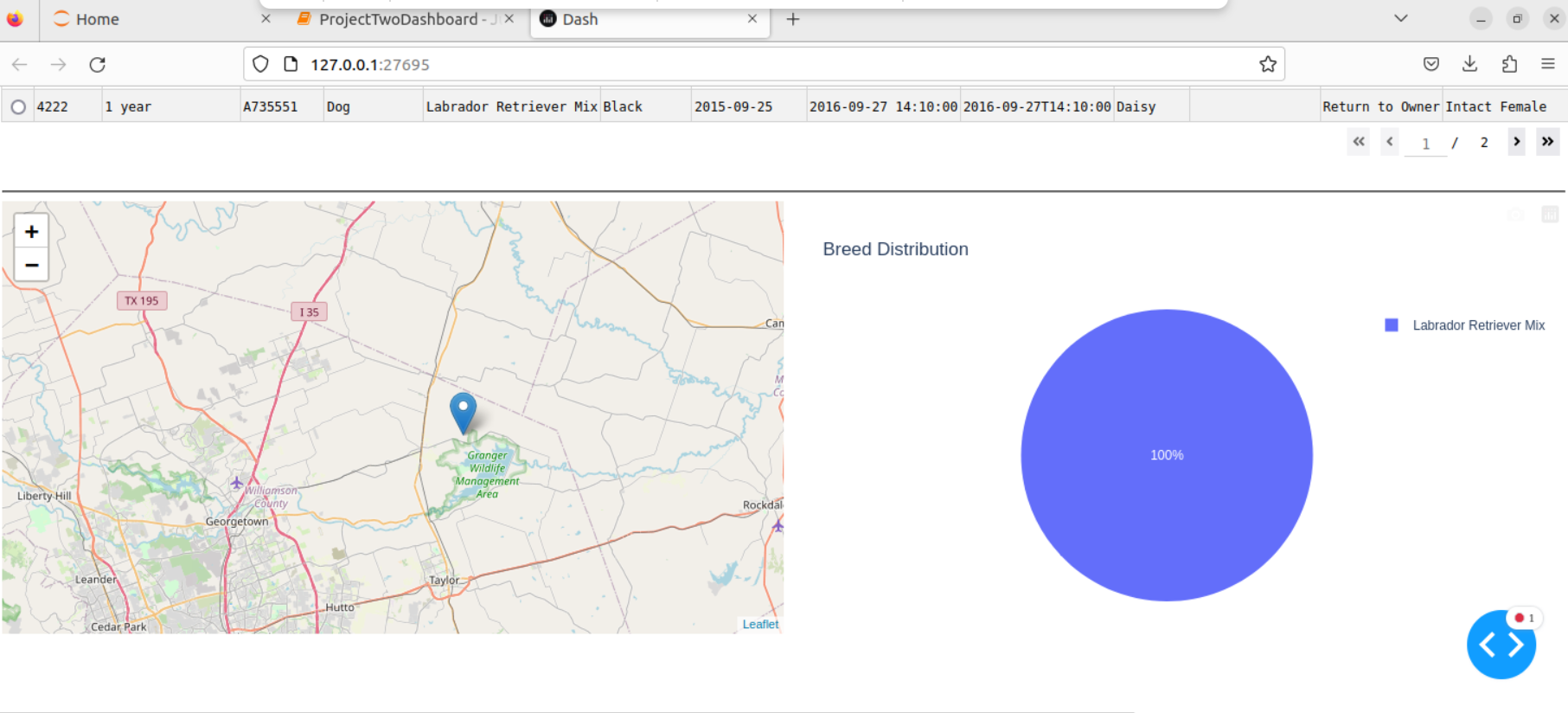
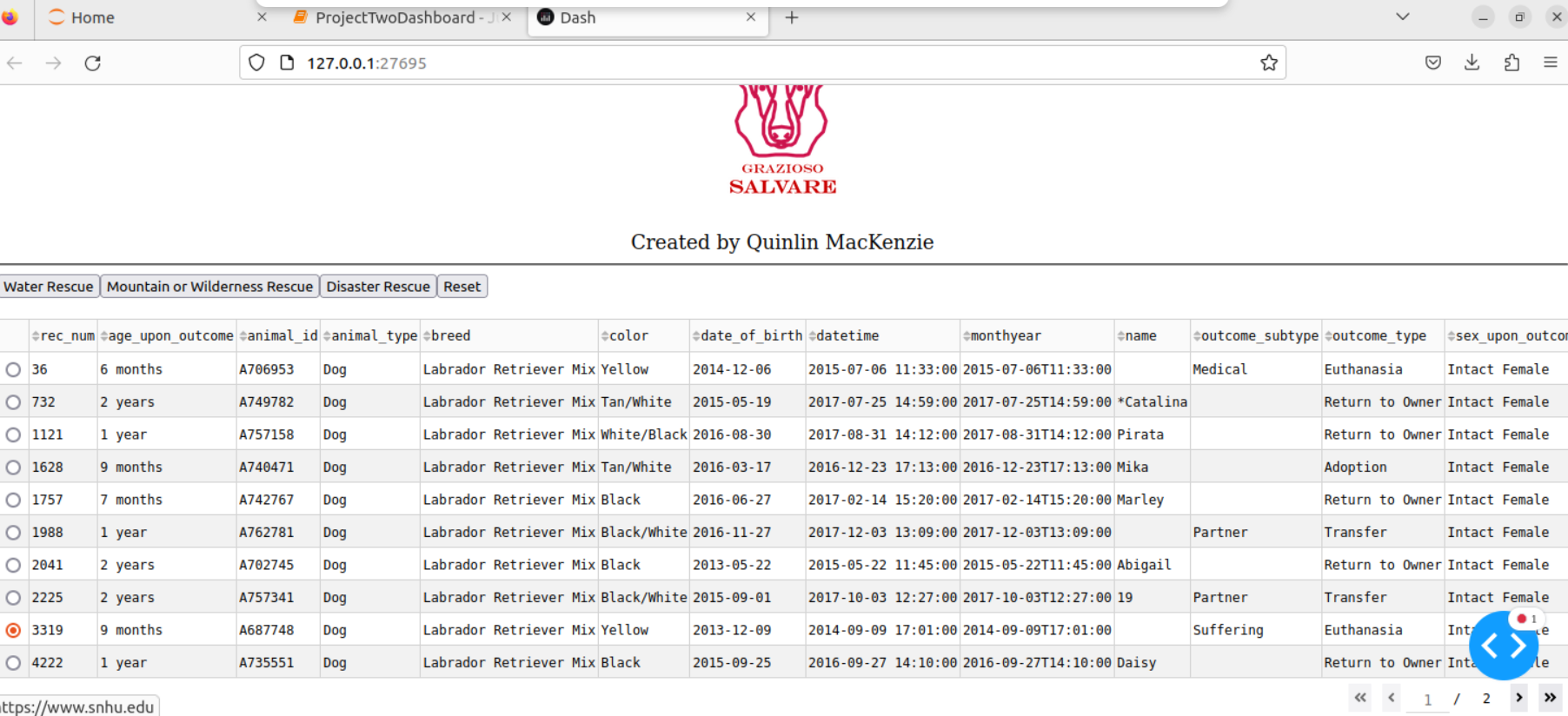
This project is a Python-based dashboard designed for Grazioso Salvare, a nonprofit organization that identifies dogs suitable for search-and-rescue training. The dashboard connects to a MongoDB database containing data from multiple animal shelters, allowing users to filter, visualize, and interact with this data effectively.

## Required Functionality

The dashboard must include the following features:

* **Branding Elements**:
  + Display the Grazioso Salvare logo with a link to the client’s homepage: [www.snhu.edu](http://www.snhu.edu/).
  + A unique identifier crediting Quinlin MacKenzie as the creator of the dashboard.
* **Interactive Filtering**:
  + Buttons to filter the Austin Animal Center Outcomes dataset by:
    - Water Rescue
    - Mountain or Wilderness Rescue
    - Disaster Rescue or Individual Tracking
    - Reset (returns all widgets to their original, unfiltered state)
* **Data Table**:
  + A dynamic table that updates based on the selected filter options.
* **Geolocation Chart**:
  + A map visualizing the locations of the selected animals.
* **Secondary Chart**:
  + A pie chart showing the distribution of breeds among the filtered data.

### Screenshots

Final Dashboard layout, logo redirects to website, buttons above table update filters and charts

## Tools Used

The following tools and libraries were employed in the development of this project:

* **Python**: The primary programming language used for backend logic and data manipulation.
* **Dash**: A web application framework that provides an easy way to build interactive web applications. It was chosen for its ability to integrate seamlessly with Plotly for data visualization.
* **Plotly**: Used for creating dynamic visualizations, including the pie chart and geolocation map.
* **Pandas**: Essential for data manipulation and analysis, especially when handling data retrieved from MongoDB.
* **MongoDB**: The database chosen for its flexibility in storing unstructured data and its ability to interface smoothly with Python via PyMongo.

### Rationale for Tool Selection

* **MongoDB** was selected for its document-oriented structure, allowing for easy storage of various animal records with different attributes. This flexibility supports the diverse data types in the Austin Animal Center Outcomes dataset.
* **Dash** provides a structured approach to building web applications, separating the data, view, and controller layers effectively, making it easier to maintain and update.

## Steps to Complete the Project

1. **Project Initialization**: Set up a MongoDB instance and create an account (aacuser) with appropriate permissions.
2. **Data Retrieval**: Use the shelterCRUD.py module to connect to MongoDB and retrieve the Austin Animal Center Outcomes dataset.
3. **Dashboard Design**: Implement the dashboard layout using Dash, including the necessary components such as the data table, filters, and charts.
4. **Filtering Logic**: Develop callbacks to update the data table and charts based on the filter selections made by users.
5. **Testing and Debugging**: Conduct thorough testing to ensure all components work as intended, addressing any issues that arise.

## Challenges Encountered

* **Filtering Functionality**: Initially, the interactive filter buttons did not update the data table as expected. This was resolved by correctly linking the buttons to the appropriate callbacks in Dash, ensuring that the filtering logic was correctly implemented.
* **Data Integration**: Some columns were missing from the DataFrame during initial data retrieval. This was fixed by ensuring that the DataFrame accurately reflected the structure of the MongoDB documents.