**CS-340 Client/Server Development**

**Dashboard Project for Grazioso Salvare**

**SNHU, Justin D. Perez  
Professor Hari Paruchuri  
June 21, 2024**

**README file:**

# CS-340 Client/Server Development

## Dashboard Project for Grazioso Salvare

\*\*SNHU, Justin D. Perez\*\*

\*\*Professor Hari Paruchuri\*\*

\*\*June 21, 2024\*\*

---

## Project Overview

This project is an interactive dashboard application developed for Grazioso Salvare as part of the CS-340 course at Southern New Hampshire University. The dashboard provides a user-friendly interface to view and analyze data from an animal shelter database. The application is built using MongoDB for data storage and Dash for the interactive web interface.

---

## Required Functionality

The dashboard achieves the following functionalities:

- \*\*Interactive Data Table:\*\* Display data from the MongoDB collection with options for filtering and selection.

- \*\*Map Visualization:\*\* Show the location of animals on a map based on their geographical coordinates.

- \*\*Graphs and Charts:\*\* Provide visual representations of animal data, such as distribution by breed or age.

---

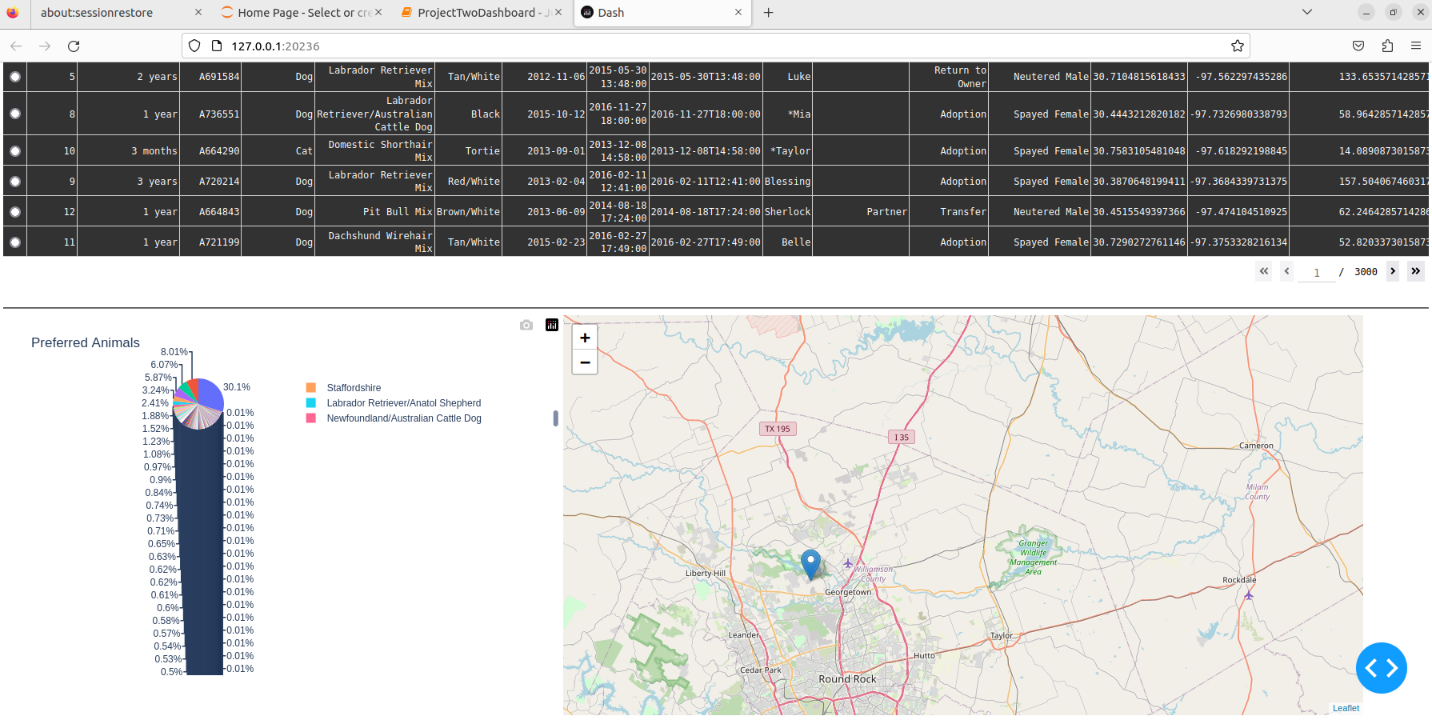
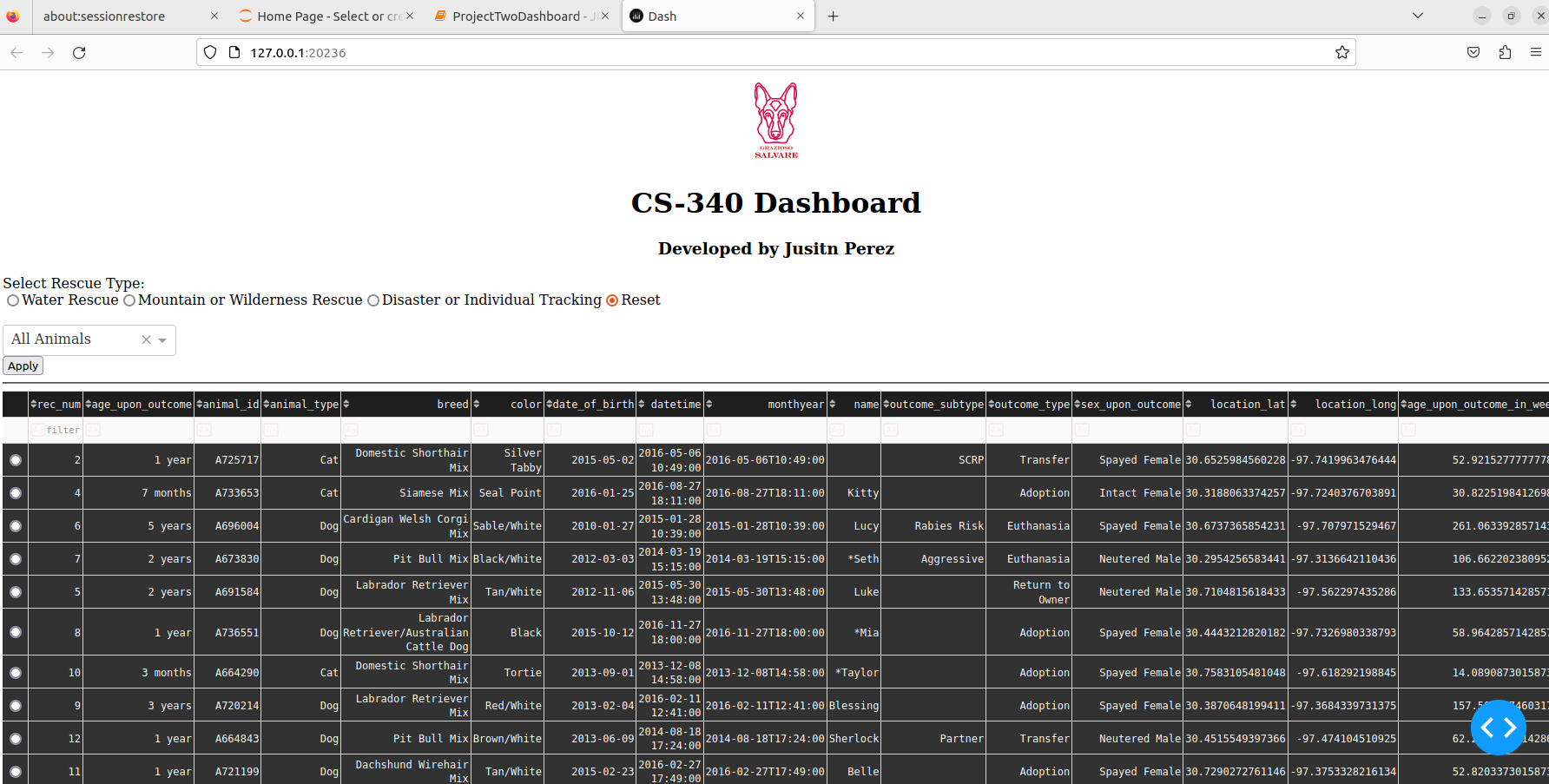
## Proof of Functionality

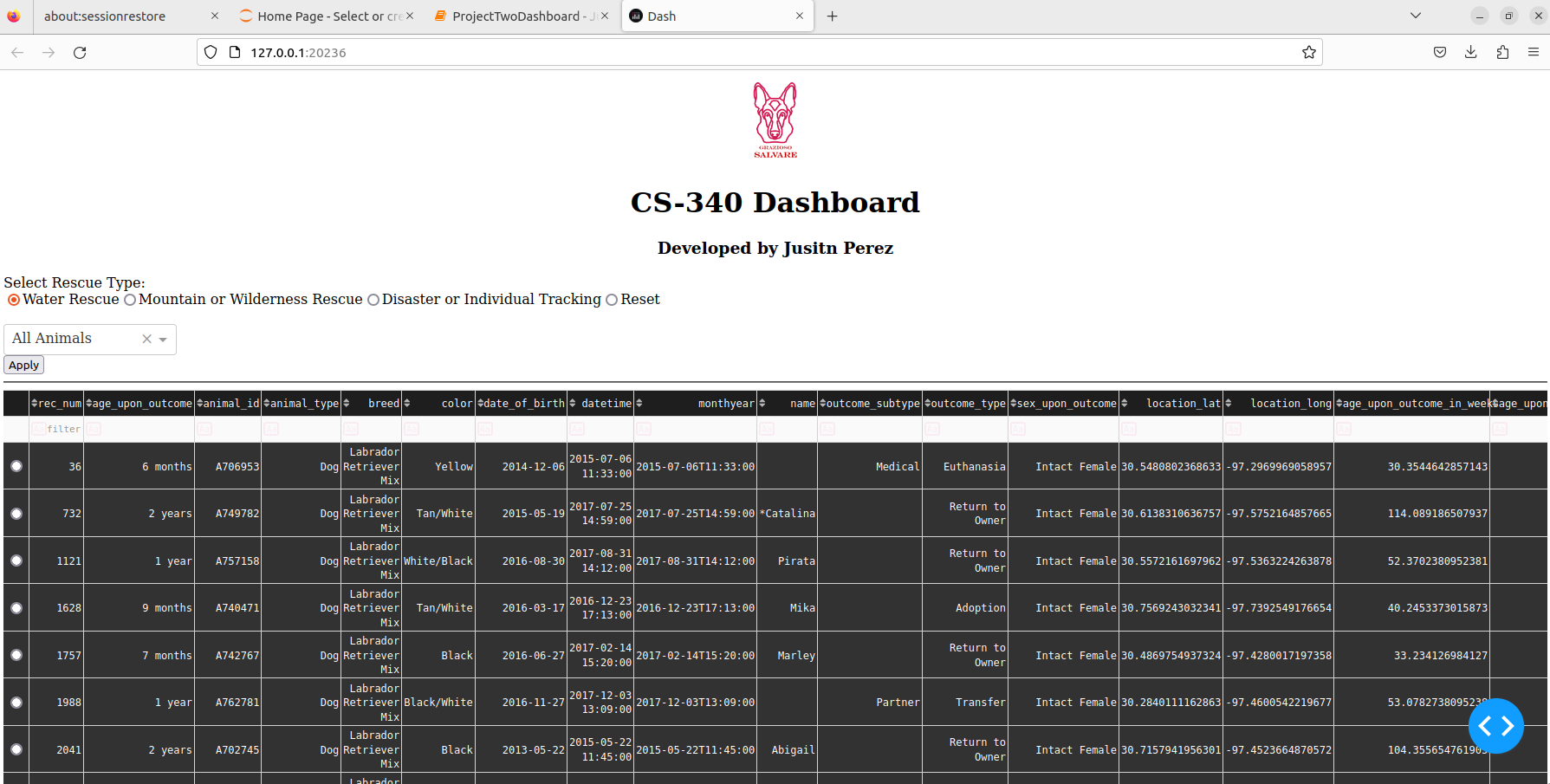
### Screenshots

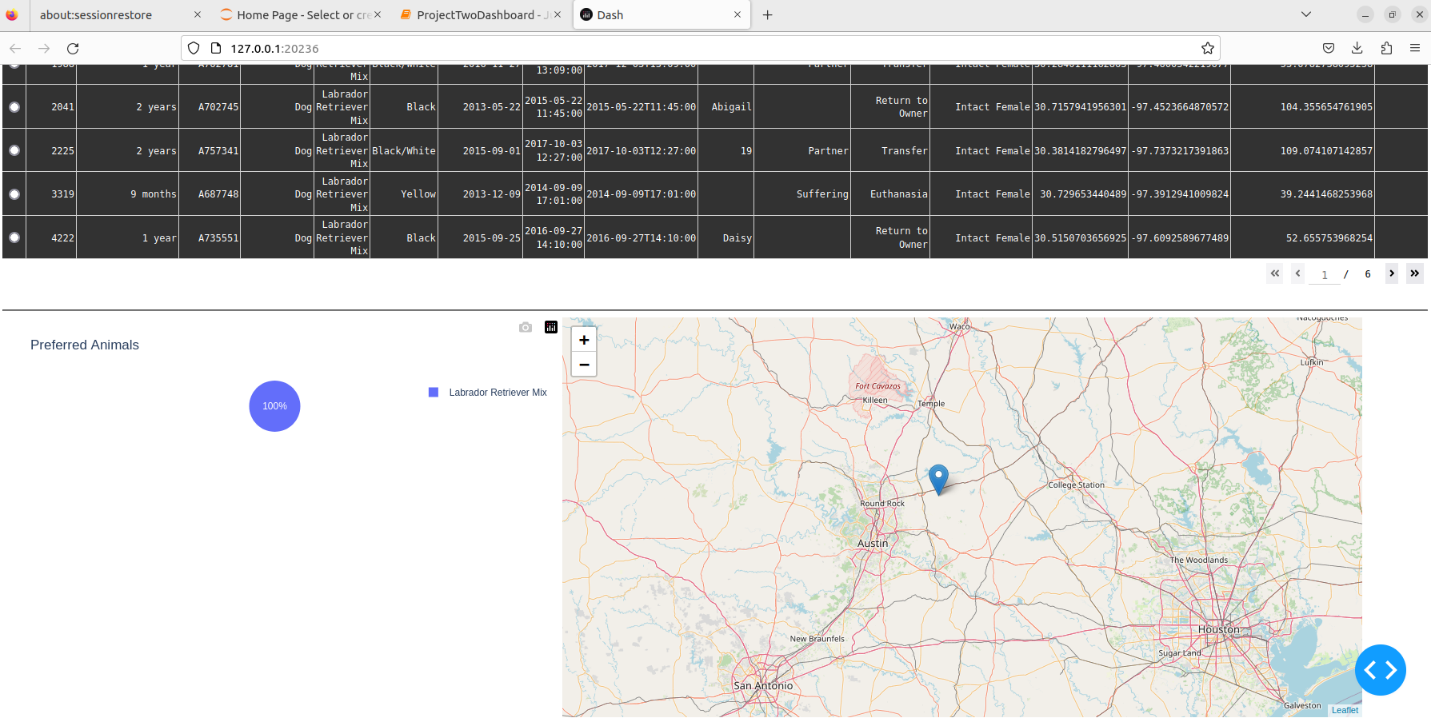
- Screenshot of the data table displaying animal records.

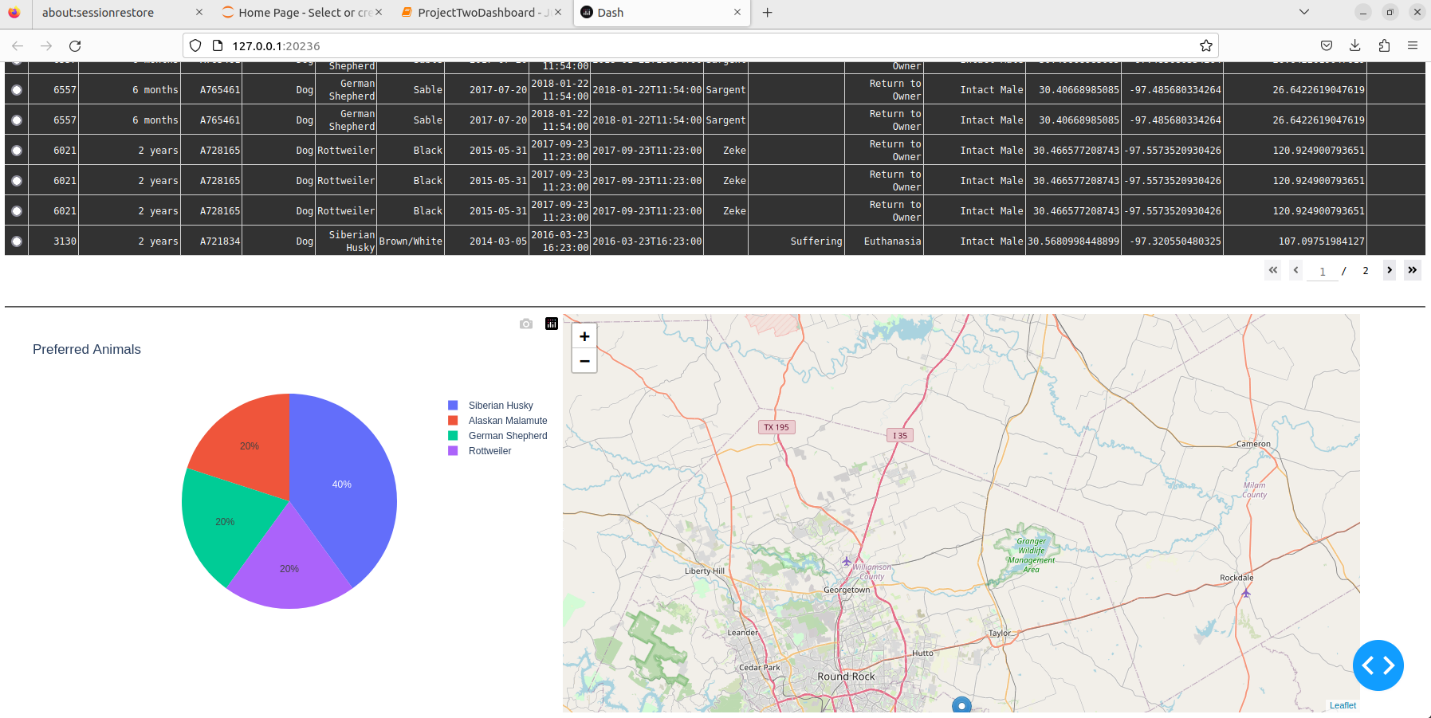
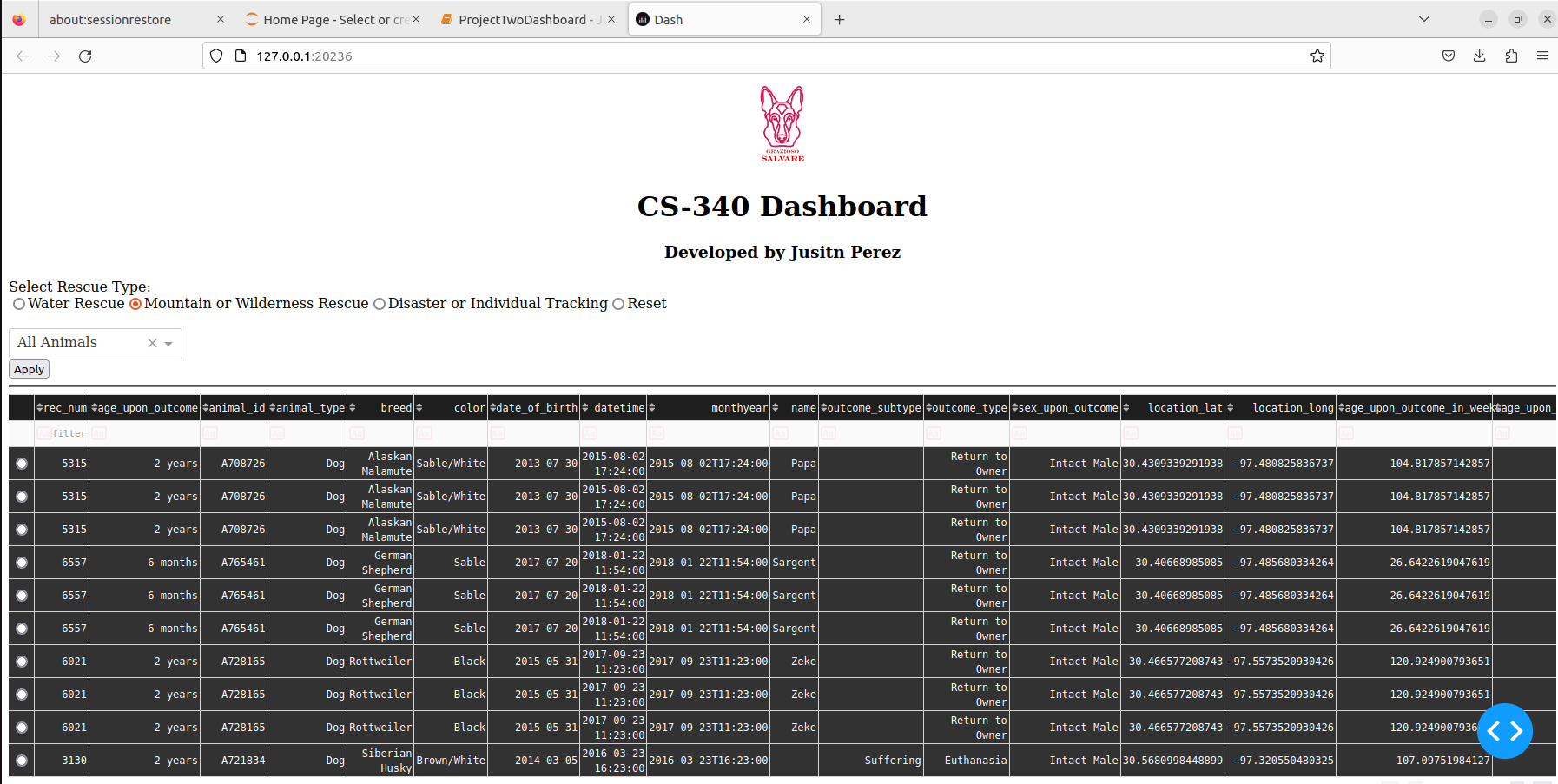
- Screenshot of the map displaying animal locations.

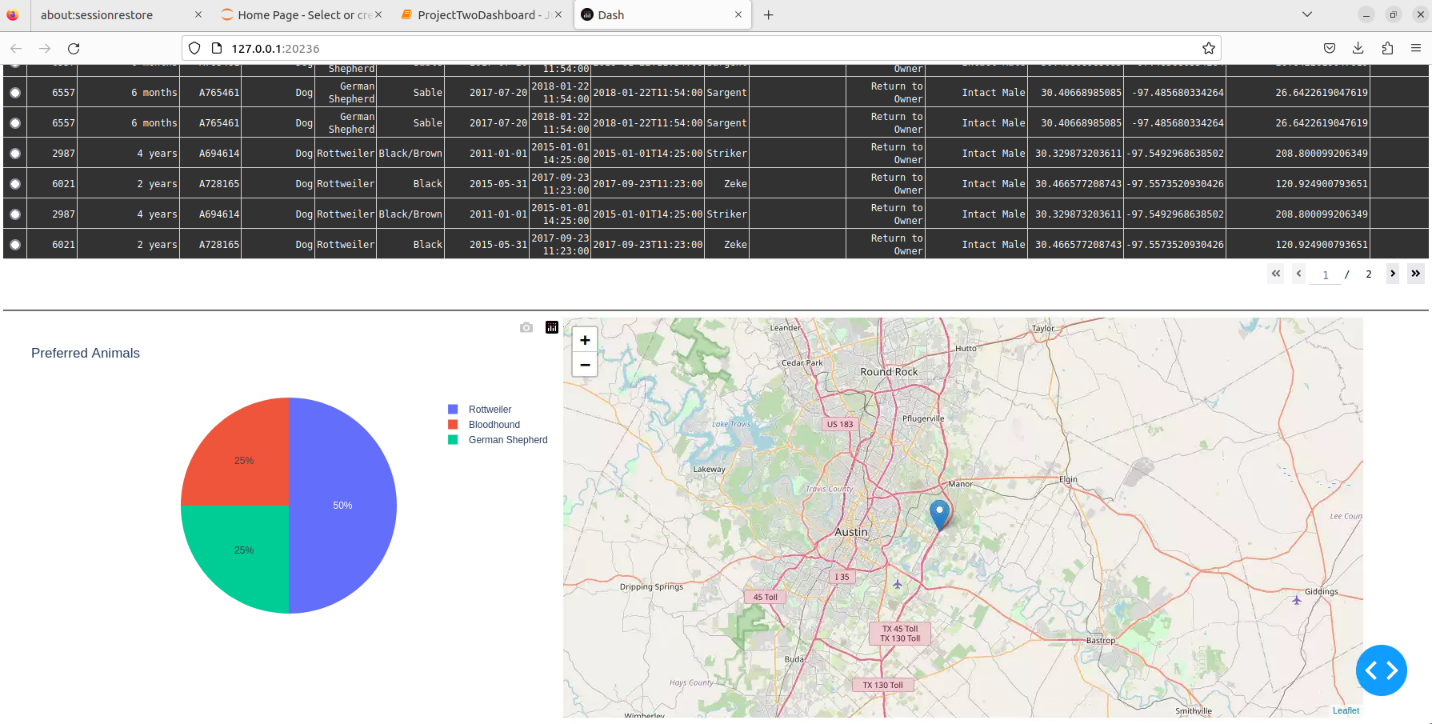
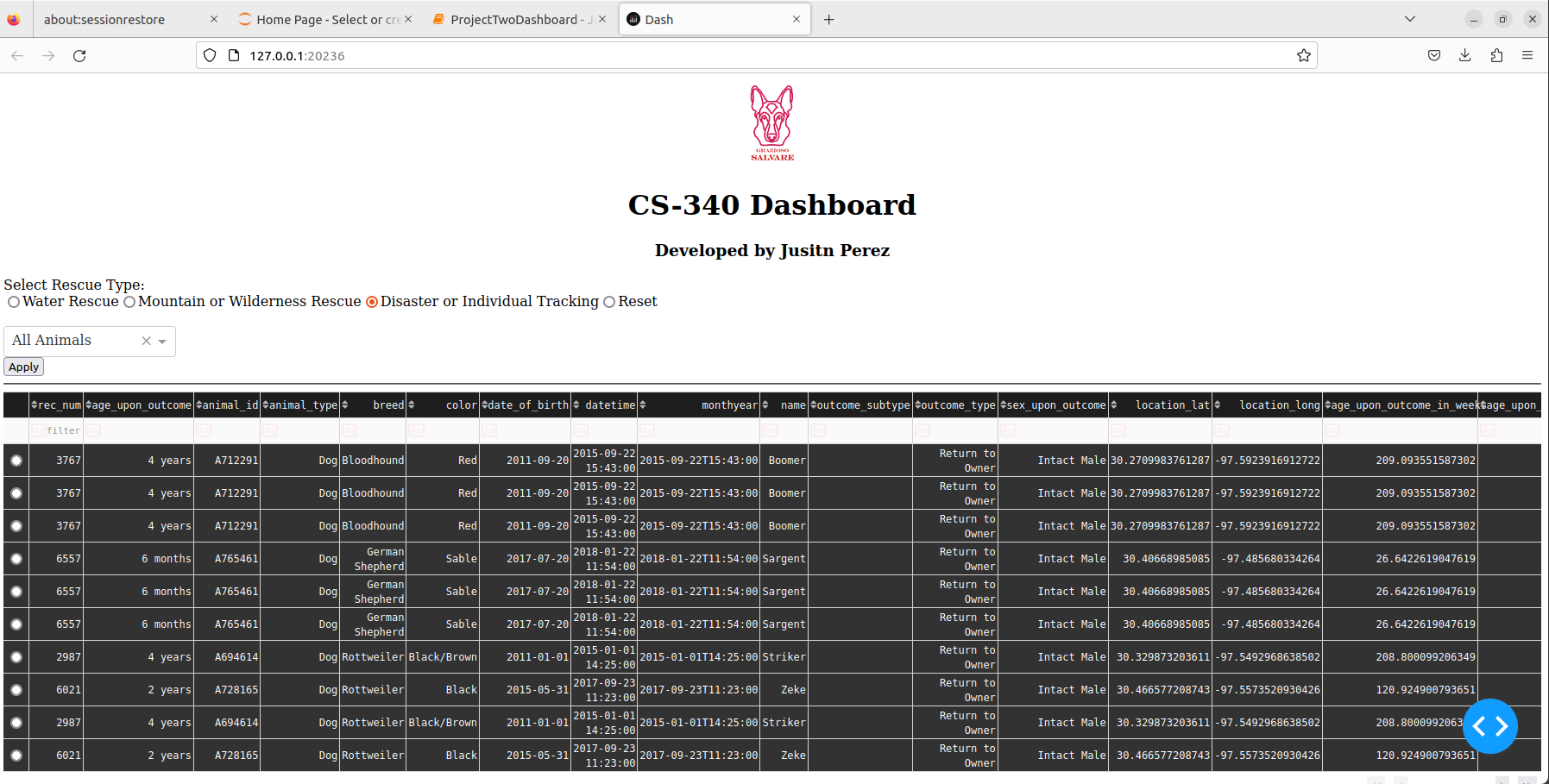
- Screenshot of the graphs and charts representing animal data.











---

## Tools and Rationale

### MongoDB

MongoDB was chosen for its flexibility and scalability. As a NoSQL database, it allows for the storage of complex and varied data structures, which is ideal for the diverse data found in an animal shelter. MongoDB's document-oriented nature makes it easy to interface with Python, particularly through the use of the pymongo library. This allows for seamless integration and efficient data retrieval.

### Dash

Dash was used to build the interactive web application due to its simplicity and effectiveness in creating data-driven applications. Dash combines the power of Plotly for interactive graphs and charts with the simplicity of Flask to handle backend operations. This makes it an ideal choice for developing a responsive and interactive dashboard.

---

## Installation and Setup

### Dependencies

The project requires the following Python packages:

- pandas

- dash

- dash\_leaflet

- jupyter-dash

- pymongo

- base64

You can install these dependencies using:

```bash

pip install -r requirements.txt

* **Configuration**

**class AnimalShelter:**

def \_\_init\_\_(self, username, password, host, port, db, col):

self.client = MongoClient(f'mongodb://{username}:{password}@{host}:{port}/{db}?authSource=admin')

self.database = self.client[db]

self.collection = self.database[col]

Ensure your MongoDB instance is running and accessible. The database should contain a collection named animals within the AAC database.

* **Running the Application**

python app.py, The application will be accessible at <http://127.0.0.1:8050/>.

**Project Steps:**

1. **Setup MongoDB:** Configured MongoDB with the animal shelter data.
2. **CRUD Module:** Developed the animal\_shelter.py module to handle CRUD operations.
3. **Data Retrieval:** Used the CRUD module to fetch data and load it into a Pandas DataFrame.
4. **Dashboard Development:** Created the Dash application layout including the data table, map visualization, and graphs.
5. **Interactive Features:** Added interactivity to the data table, allowing for filtering and selection, and linked it to the map and graphs.
6. **Testing and Deployment:** Tested the application locally and deployed it for review.

**Challenges and Solutions:**

**Database Connection Issues:**

**Challenge**: Encountered issues with connecting to the MongoDB instance.

**Solution**: Verified the connection string and credentials, and ensured the MongoDB service was running and accessible.

**Data Display and Formatting:**

**Challenge**: Properly displaying and formatting the data in the Dash data table.

**Solution**: Utilized Pandas to preprocess the data and removed any unnecessary columns that could cause issues.

**Map Visualization:**

**Challenge**: Integrating the map visualization with animal location data.

**Solution**: Used the dash\_leaflet library to create interactive maps and linked the data table selections to update the map dynamically.

**Resources:**

* Dash Documentation
* [MongoDB Documentation](https://docs.mongodb.com/)
* Pandas Documentation
* Dash Leaflet Documentation

License:

This project is licensed under the MIT License.