# CS 340 Client/Server Development Project Two: Interactive Dashboard README

Professor Bill Chan

Mubeen Ahmed Khan

mubeenahmed.khan@snhu.edu

Southern New Hampshire University

18th April, 2025

## About the Project

This project delivers a full-stack dashboard solution for Grazioso Salvare, a global search-and-rescue dog training organization. Built using Python and Dash, the application connects to a MongoDB database of shelter animals and provides an interactive interface to identify, filter, and visualize animals suitable for search-and-rescue training.

### The dashboard supports:

* Interactive filtering based on rescue type
* Real-time updates from a MongoDB backend
* Geolocation mapping of selected animals
* Visual distribution of breed types via pie chart

The project demonstrates practical knowledge in database connectivity, dashboard development, data visualization, and user-centric design.

## Motivation

The inspiration for this project stems from the need to streamline how rescue organizations identify dogs with specific physical and behavioral traits. Using real-world data from animal shelters in the Austin, TX region, Grazioso Salvare can now efficiently locate candidates for training in water rescues, wilderness missions, disaster tracking, and individual search operations.

### Key goals included:

* Applying CRUD operations with live MongoDB data
* Creating a professional, intuitive dashboard interface
* Implementing real-time geospatial and statistical visualizations
* Demonstrating client-server architecture in action

## Getting Started

### MongoDB Configuration

Ensure you have access to a MongoDB server. You can use:

* MongoDB Atlas (cloud-hosted database)
* MongoDB Community Edition (installed locally)

Your MongoDB database should include:

* **Database:** aac
* **Collection:** animals

Authentication credentials are hardcoded for academic purposes but can be adapted for .env usage in production.

### Installation

#### Requirements:

* Python 3.9+
* Jupyter Notebook
* MongoDB instance (local or Atlas)

#### Install necessary packages:

pip install pymongo jupyter-dash dash dash-leaflet plotly pandas

#### File Structure

* **animal\_shelter.py** — Contains the AnimalShelter class used for MongoDB CRUD operations
* **ProjectTwoDashboard.ipynb** — Contains the Dash app implementation with all visualization and logic
* **Grazioso Salvare Logo.png** — Branding logo used in dashboard UI
* **README.md** — Documentation for running and understanding the project

#### Usage

##### Running the Dashboard

* Launch Jupyter Notebook.
* Open ProjectTwoDashboard.ipynb.
* Run all cells. The dashboard will launch in your browser via localhost (external mode).

##### Features

###### Interactive Filtering

* Column titles can be clicked on sorting
* Radio button options for Water Rescue, Mountain or Wilderness Rescue, Disaster/Tracking, and Reset
* Live filtering triggers MongoDB queries using PyMongo and updates the dashboard views

###### Pie Chart

* Displays breed distribution of the filtered results
* On-hover popups show percentages and breed names

###### Geolocation Map

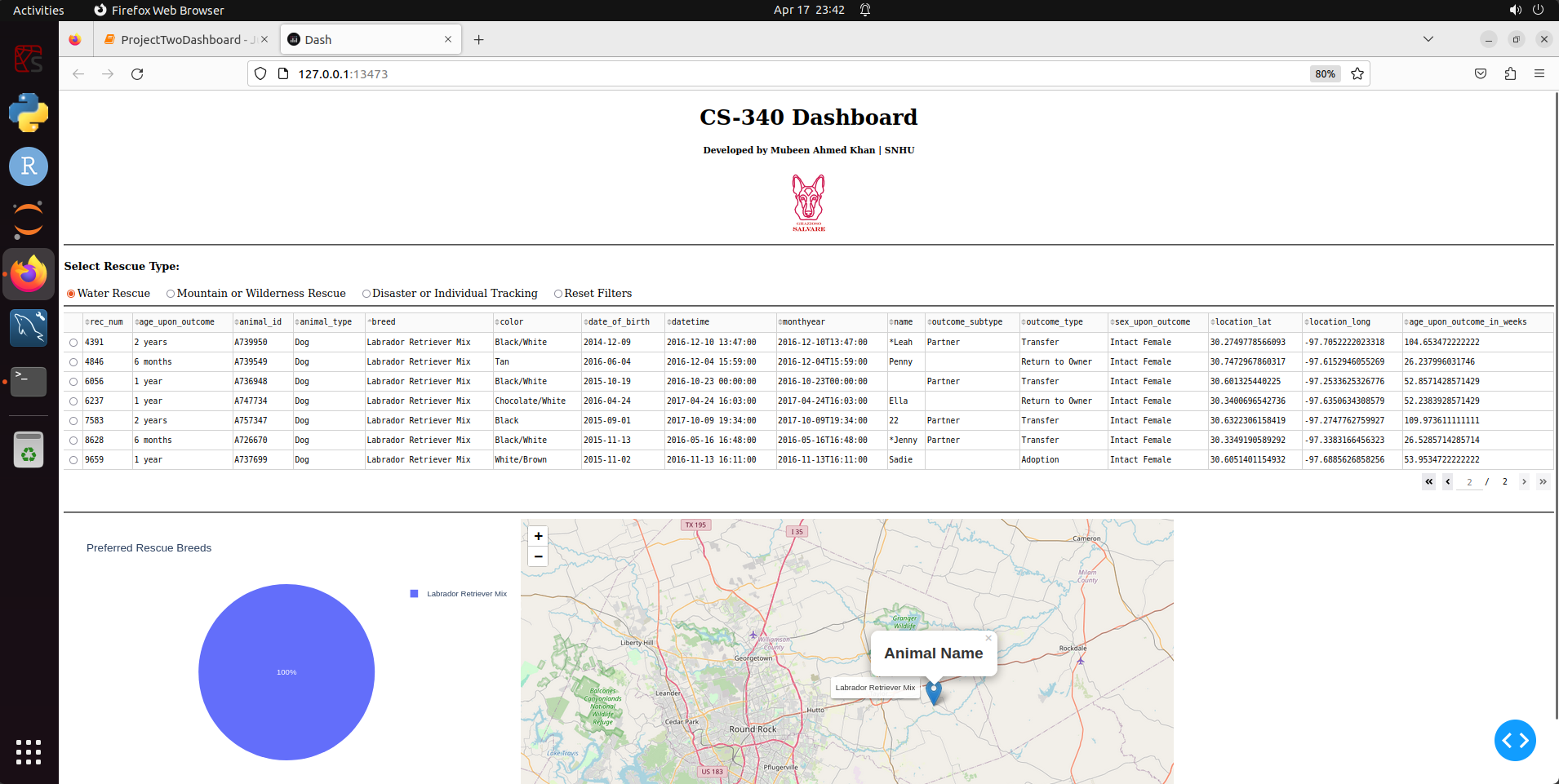
* Built with dash\_leaflet
* Updates location dynamically based on the selected row from the data table
* Markers show breed name (tooltip) and animal name (popup)

###### Data Table

* Interactive table with sortable columns and selectable rows
* Supports real-time filtering and direct integration with map updates

## Screenshots

| **Filter Applied** | **Data Table** | **Pie Chart** | **Geolocation Map** |
| --- | --- | --- | --- |
| **Water Rescue** | ✅ | ✅ | ✅ |
| **Mountain or Wilderness** | ✅ | ✅ | ✅ |
| **Disaster or Individual Tracking** | ✅ | ✅ | ✅ |
| **Reset (Unfiltered)** | ✅ | ✅ | ✅ |

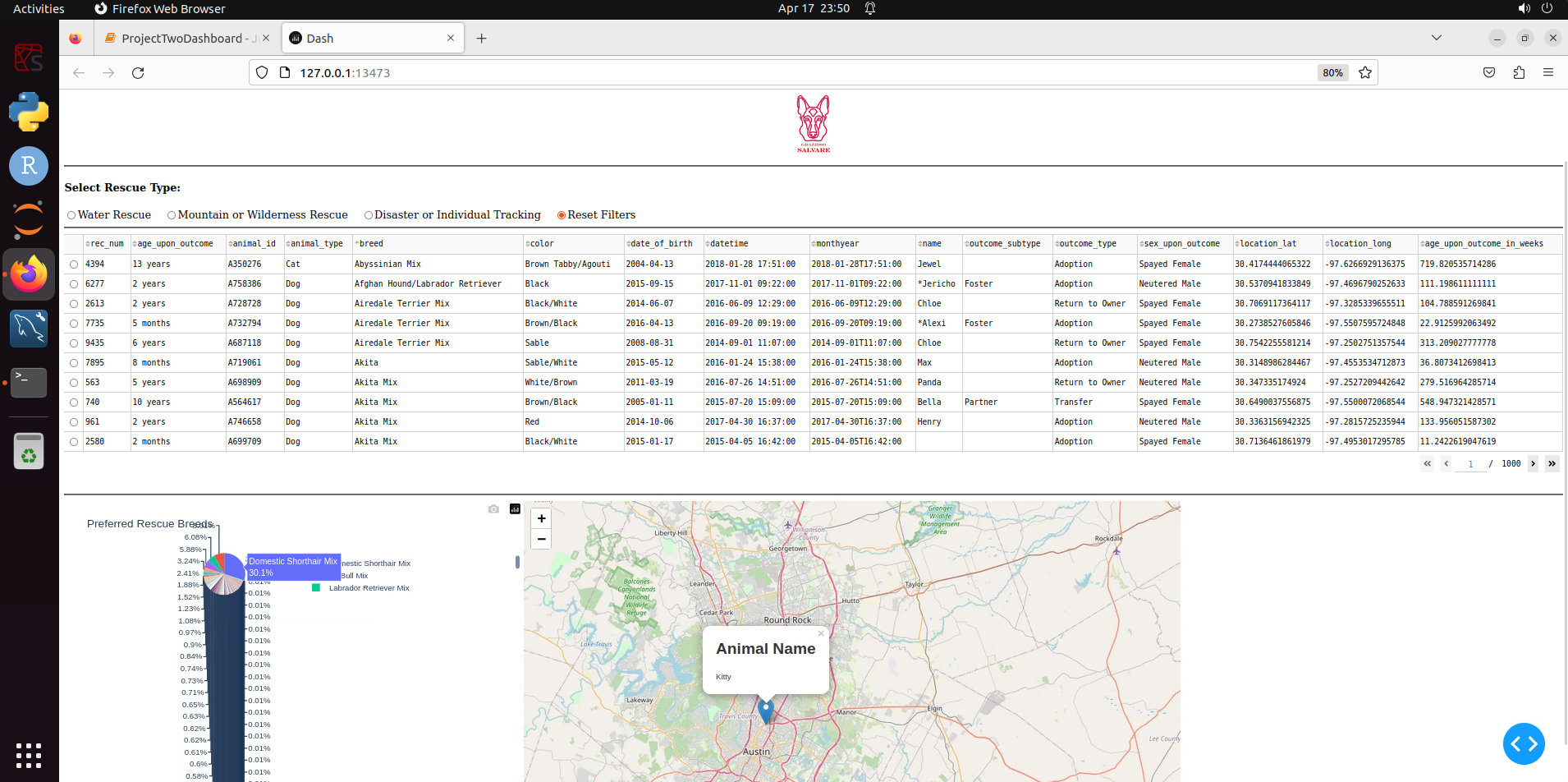


A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.



## Development Notes

### Challenges & Resolutions

* **Filter Logic:** Created build\_query() to handle MongoDB $in and range queries based on breed, age, and sex.
* **Map Integration:** Used dynamic dl.Marker objects with tooltip and popup attributes for clarity.
* **Performance:** Ensured smooth experience by filtering only on server side, and avoiding unnecessary front-end overhead.
* **Dashboard Usability:** Enhanced layout with horizontal filter alignment and improved label visibility.

## Future Improvements

* Add Update and Delete functionality
* Enable export of filtered data
* Add dropdown or multi-filtering capability
* Deploy the dashboard as a web app using Dash Enterprise or Heroku

## References

* https://www.mongodb.com/docs
* https://dash.plotly.com/
* https://www.dash-leaflet.com/
* https://pymongo.readthedocs.io/en/stable/

## Attribution

This project is developed for the CS-340: Client/Server Development course at Southern New Hampshire University.

© 2025 – Developed by Mubeen Ahmed Khan