README – CS 340 Project Two

This project is a full-stack dashboard application developed for Grazioso Salvare, a search-and-rescue organization that works with shelter animals. The dashboard was designed to help identify potential candidates for search-and-rescue training based on specific breed and age criteria. It connects to a MongoDB database and provides an intuitive interface where users can interactively filter data, visualize breed distributions, and view the geolocation of selected animals.

To develop this application, I used several tools and frameworks. MongoDB was used as the backend database because of its flexibility with unstructured data and strong support for Python integration through the pymongo library. Python and Dash were used to create the dashboard interface, with components such as radio buttons for filtering, interactive tables, pie charts using Plotly Express, and geolocation maps using Dash Leaflet. I also used base64 encoding to embed the Grazioso Salvare logo directly in the dashboard, avoiding any external file dependency issues.

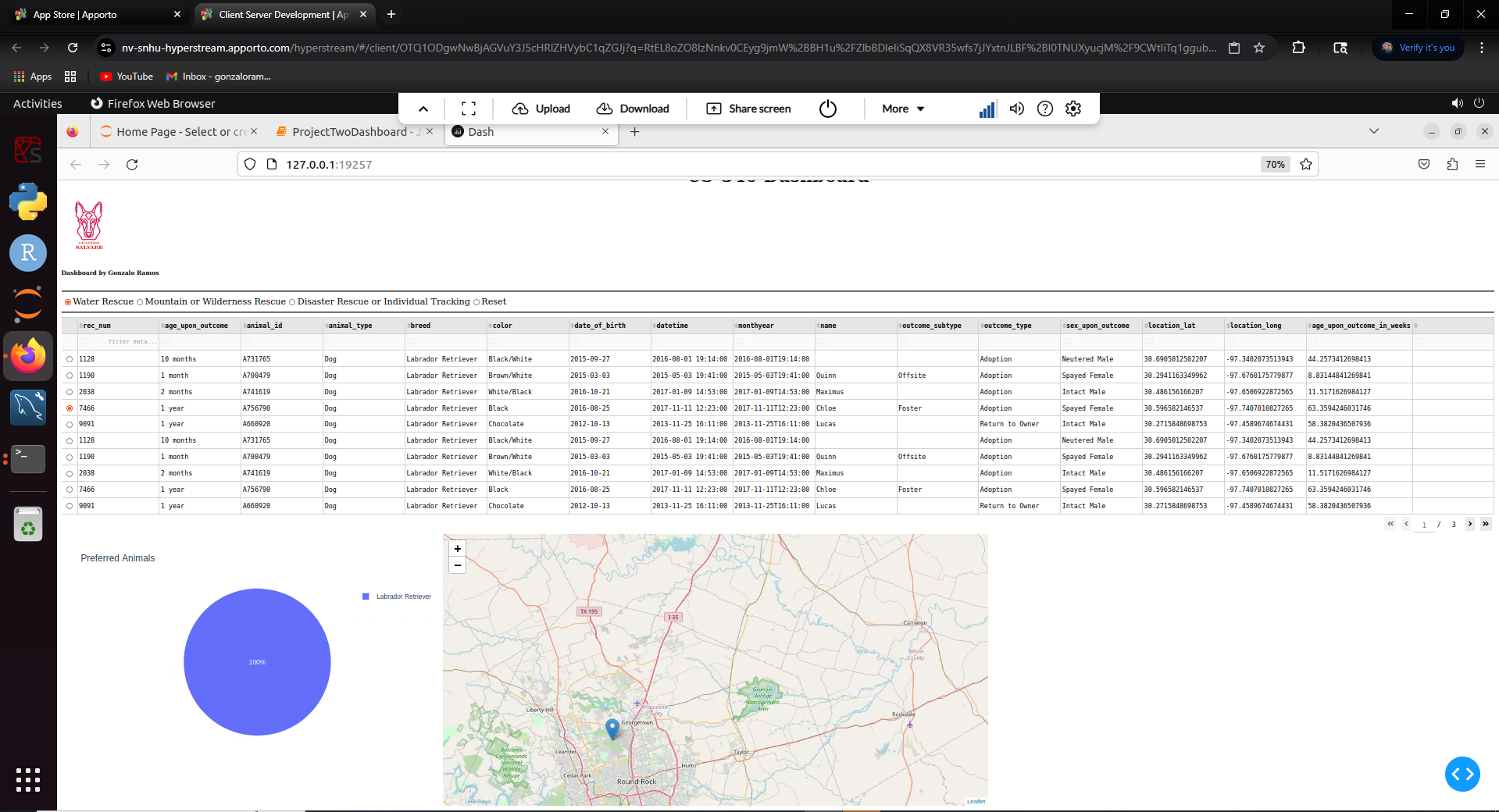
To run the application, all files must be placed in the same directory. This includes the ProjectTwoDashboard.ipynb notebook, the animal\_shelter.py CRUD module, the aac\_shelter\_outcomes.csv data file, and the grazioso\_salvare\_logo.png image. The dashboard is run using app.run\_server(mode='inline') in the final cell of the notebook. Once running, users can select a filter type, view a filtered data table, explore breed distributions in a pie chart, and select a row to see its location on the map.

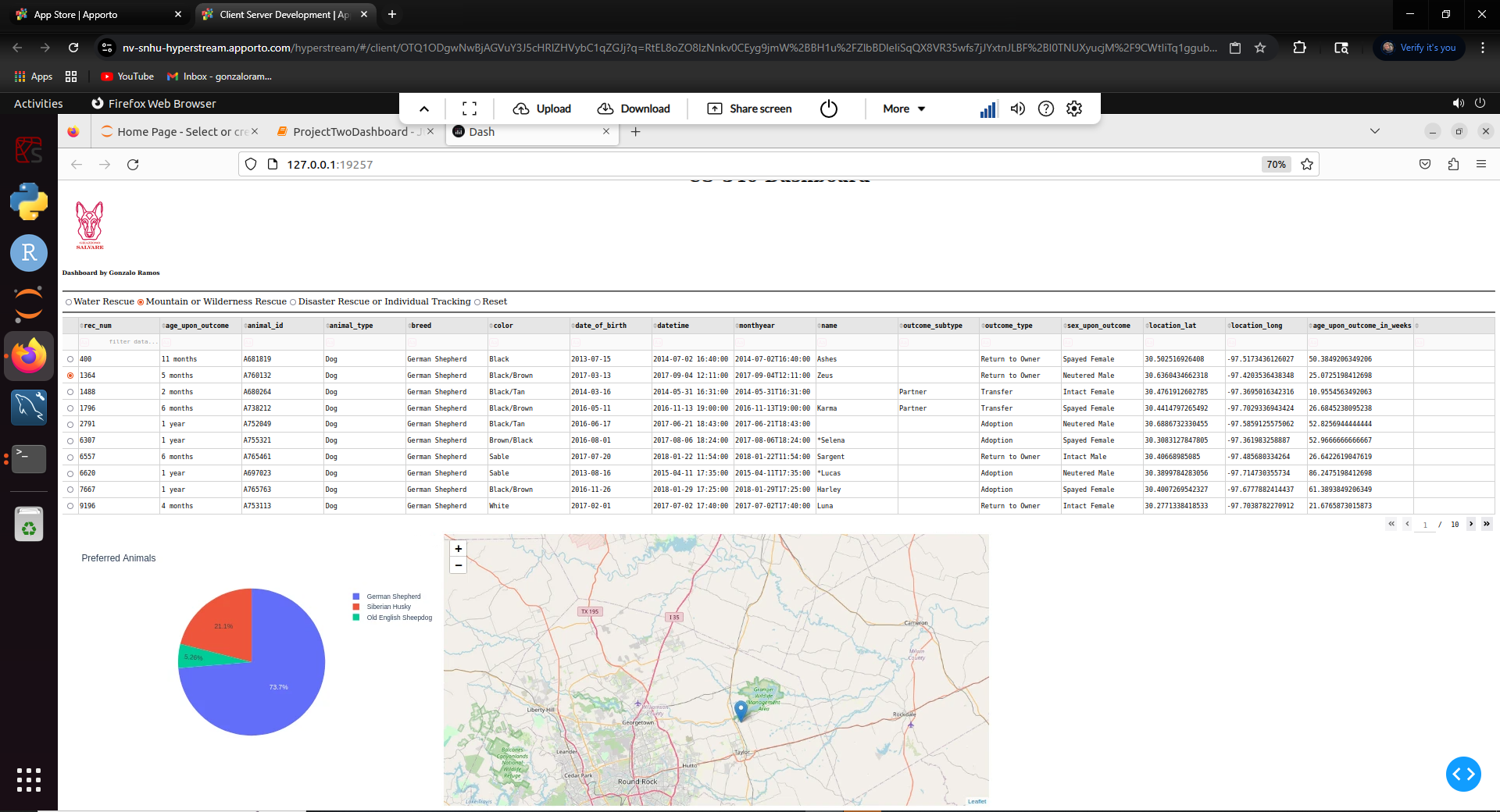
During development, a few challenges were encountered. Initially, I experienced issues connecting to the MongoDB database due to incorrect host and port configurations, which were later resolved by confirming the Apporto connection string. I also had to ensure only animals under two years old were shown for rescue qualification, and that certain breeds matched specific rescue roles. One critical bug was caused by the \_id field returned by MongoDB, which had to be removed from the DataFrame to avoid breaking the Dash data table component. Lastly, embedding the Grazioso Salvare logo using base64 allowed the image to display reliably inside the Jupyter notebook without path errors.

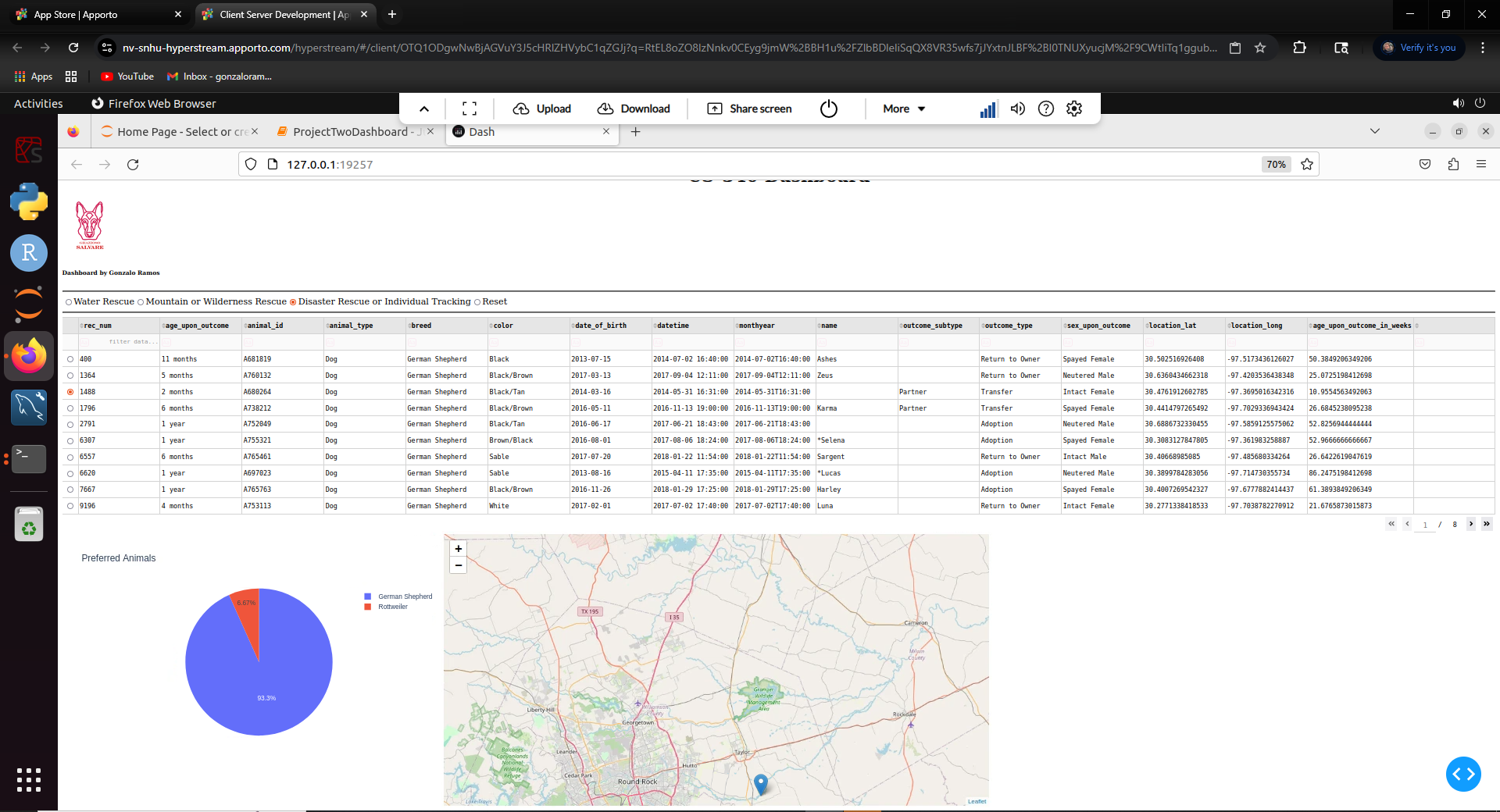
Below are the required screenshots demonstrating the dashboard’s functionality:

A screenshot of a computer

AI-generated content may be incorrect.**Figure 1: Reset View (Default)**

**Figure 2: Water Rescue Filter**

**Figure 3: Mountain or Wilderness Rescue Filter**

**Figure 4: Disaster Rescue or Individual Tracking Filter**