Water Rescue Functionality:

A screenshot of a computer

AI-generated content may be incorrect.

Mountain or Wilderness Functionality:

A screenshot of a computer

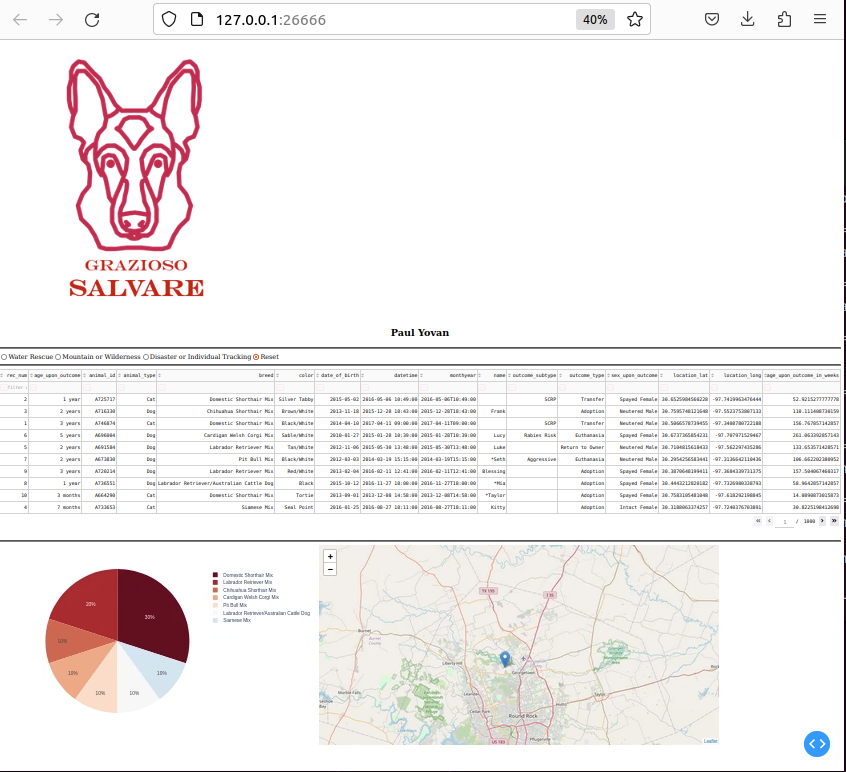
AI-generated content may be incorrect.

Disaster or Individual Tracking Functionality:

A screenshot of a computer

AI-generated content may be incorrect.

Reset Functionality:



Tools used to achieve this functionality:

MongoDB – Helped import the database given by Global Rain into my CRUD python file by having my file access MongoDB with given credentials.

Dash framework – Helps with how the display works by using html functions to structure the code and help make a more user-friendly interface by being able to import buttons that can help with filtering animals as well as searching for specific criteria through filter commands.

Steps taken to complete the project:

The first step I took was importing the database into MongoDB so that my python file could access the database successfully.

The second step I took was taking the already given code and then fixing the code that was easiest first and then working with the html functions to help make a database that I personally liked the display of.

The third step I took was figuring out how I could use a dash table to help make the display look neat and then also figuring out how to display the geographic map provided to help display the location of specific animals when searched for.

The final step I took was looking over my code and then working out any errors that may have arisen and then running the code again and making sure that every function that I implemented worked properly.

Challenges:

One challenge I had was actually successfully importing the database with MongoDB and what helped me in solving this problem was by researching the mongo import command and then applying what I already knew about the command into what I learned to finally successfully make it work properly with the new database.

The second challenge I had was importing the geographical map into the database without getting any errors regarding the wrong displays of where the specific dog was located and the way I resolved this was by researching a bit more about other users who have used a geographical map and how they successfully implemented it and then I applied their logic with my own to successfully make the map.

The final challenge I faced was making the buttons for filtering to actually successfully work as I had it set up to where when you clicked the button, nothing would display and the way I fixed that was by going through my code and seeing what may have been causing this and when I found out some reasons the error still persisted so I went and searched and figured out that I called the wrong value for the button I created as I gave each button a value for calling when I needed them which upon finding once I fixed that issue the filtering worked successfully.