README

The functionality needed by Gravioso was an organized database that could be broken down into different search functionalities with some useful user widgets. For the project we broke the database entries into four different collections; All animals, animals fit for water rescue, animals fit for wilderness rescue, and animals fit for disaster rescue. As you can see in the follow screenshots, all four have been implemented.

A picture containing graphical user interface

Description automatically generatedA picture containing graphical user interface

Description automatically generatedGraphical user interface

Description automatically generated with medium confidenceGraphical user interface, application

Description automatically generated

MongoDB was used as the database structure for the project has it has all the functionality needed as well as ease of use. With MongoDB comes pymongo which is a python library for mongo which allows us to use the functionality of mongo through a python script. This is incredibly useful for turning the database into a user-friendly result. Another tool we used was the dash framework which allowed us to display our data in neat user-friendly manners such as the graphs and the maps that we created.

The project was entirely a learning experience since the first week of the course because coming into the course I had no experience with any databases at all. Building on the knowledge one week at a time really helped me to understand how to manipulate and interact with MongoDB. The most challenging part to me occurred when we started using python to interact with the databases as it was entirely new to me and seemed to come with a lot of difficulties as well. The biggest difficulty I ran into was when creating the map, I made a couple of syntax errors and instead of raising an error in the program it just failed to load the map. It took me awhile to troubleshoot this and to figure out what I was doing wrong but eventually I understood it.