README

This project was done by Global Rain by the developer Thomas Fiske. The purpose of this project was to connect a CRUD module to a Dash application that allowed the user to use the interactive Dash app to find dogs that fit certain criteria. This application makes use of MongoDB as well as Python. It utilizes a database that was made in Mongo and pulls information from that database in order to show widgets and graphs to the user. The user will be allowed to use different filtering options to look for certain types of dogs. The user requested that they be able to see dogs that could be used for different tasks such as Rescue or tracking in different environments. These different filtering options are built into the Dash app. The app also has a reset button to reset the filtering option chosen by the user.

MongoDB was used for this project because of its ability to interact with Python. The Mongo database was easily assessable through a python CRUD module and then that could be used for information for a Dash app as well. Another reason that MongoDB was used was that it can quickly access data in its databases and that can be manipulated easily with Python. This is part of the required functionality for the application because it needs to be able to access the data quickly for the user.

The Dash component of this application was used to create an interface that the user could interact with. This allowed for HTML code to be used to display different widgets, graphs, and tables that the user could interact with. Dash is the framework that this application is run through and it allows for data to be manipulated through different app call backs. It also allows for different parts of the application to work with other parts with its app callback functions. A good example of this would be the filter that is written in Python interacting with the HTML coded graph.

This project was made with a few different steps, these are the ones that I followed:

1: Create a database within MongoDB ( “AAC”)

2: Create a user within your MongoDB ( “accuser”)

3: Create a crud module in Python. This is done by creating a Create, Read, Update, and Delete method using Python that will load your Mongo database.

4: Create a Dash app. More information: <https://dash.plotly.com/>

5: Load your Python/MongoDB Crud module into your Dash app.

6. Create the required functionality to display graphs and other data. This includes different app call backs to allow the different parts of the application to work with one another.

The challenges that I faced with this application was authentication. I have run into an error that does not allow me to authentication my self and I have not been able to fully access the application. I will continue to figure out this authentication problem. This application requires a username and a passcode, both of which appear to be correct but I am still unable to get authentication.

Another challenge that I faced was making the CRUD Module. The read method was used a lot in the application and I ran into several challenges with it. The way that I fixed this was to continually look at different examples of the read method in a CRUD module, as well as different syntax for MongoDB because it is deeply connected to the module. I was able to figure it out after help from my professor and different examples from the resources provided in the course that I was a part of.

Contact

Thomas Fiske, SNHU

[Thomas.fiske@snhu.edu](mailto:Thomas.fiske@snhu.edu)