Justin Phung & Ryan Sanders
12/5/2024

Assignment 8 Report

GitHub Repository Link: https://github.com/BigWeege/Assignment-8-Group-11

Project Summary

Our project utilizes Google Cloud to host two virtual machines: one that acts as a client that asks queries and one that acts as a server that answers them. The server uses the pymongo module from Python to access a MongoDB database that is connected to a website called Dataniz. Dataniz generates the data for three IoT devices and MongoDB stores said data.

IoT Sensors and Data

These IoT devices consist of two refrigerators and a dishwasher. Each fridge has a thermistor that randomly generates temperatures between -10 and 4 degrees Celsius (the optimal temperature range for a fridge according to the USDA), an Ammeter that randomly generates currents between 3 and 6 Amperes (the optimal current range for a fridge according to EnergySage.com), and a Moisture Meter that randomly generates relative humidity percentages between 30 and 50 percent (the optimal relative humidity for a fridge according to Bodewell.com). Each dishwasher has a water consumption sensor that randomly generates water consumption rates between 0 and 4 gallons per cycle (the water consumption range of an Energy Star-rated dishwasher according to Mr. Appliance.com) and an ammeter that randomly generates currents between 10 and 12.5 Amperes (the average current range for a residential dishwasher according to ElectriciansServiceTeam.com).

Dataniz Metadata

We created two pieces of metadata for each of our devices one for the location and the other for the name of the device. The reason we decided for these pieces of metadata is in order to differentiate between the two devices that have the same name, that being the refrigerators, and the devices being in the same place, that being the kitchen. This differentiation is important so we know what IoT device we are talking about when answering the queries, like which is using the most power.

Algorithms, Calculations, and Conversions

Query 1 didn't require many conversions since Dataniz already generated moisture readings in terms of relative humidity percentage. I also didn't need to convert any timestamp readings since Dataniz creates timestamps in terms of UTC and the time function from Python's time module also generates a timestamp in terms of UTC for comparison. Query 2 didn't require any conversions since Dataniz already generated water consumption based on the average water consumption of a dishwasher per cycle. Query 3 also didn't require any conversions since power consumption in the form of kWh can be calculated by current multiplied by voltage and hours

used all divided by 1000. Since The hours of usage and the voltage of each device were all the same (the average voltage of a dishwasher and a fridge are both 120 V), each device's power consumption was based on its current, which Dataniz generated for each device.

Challenges

The challenges we faced with the project was the use of mongoDB, comparison of fields, determining time, and determining the location of the devices for the queries. Starting with mongo, both my partner and I were out of practice for using the database as it had been around a year and a half since we had used it for projects so we had to relearn many of its features. Our next challenge was determining the length of 3 hr for query 1 in order for it to correctly gather the data from the past 3 hrs. We solve this using the time() function to form the time module to know the current timestamp and subtract 10800 (3 hours converted to seconds) to know the timestamp 3 hours ago. Next was determining the location of the kitchen because Query 1 requires the server to know which Fridge device is located in the kitchen. We were able to solve this by using custom metadata to specify the location of each device within the house. The final challenge was comparing three differently-named fields for Query 3. I was able to use the $project query to create a new field for each device's current and used the $filter and $regexMatch functions so that the new field contained the sensor reading that contained the substring "Ammeter".

Dataniz Feedback

Some feedback that we have is that we had some inconveniences when adding metadata as in order to add it even if the field is filled when clicking add at the bottom it wouldn't be added unless the plus was clicked before adding. Another thing that could be improved was the login not allowing you to retry after inputting an incorrect account name/password. Lastly, the image being required in order to create a custom sensor was a bit inconvenient as it doesn't feel like it added anything to the sensor as it can be any image.