Design and Analysis of IoT Software Systems
SOFE 4610U
Fall 2021

IoT Final Project Report
Generator Temperature Monitoring System

December 6th, 2021

Group 6	
Owais Quadri	100697281
Raveenth Maheswaran	100704540
Sabesan Sivakumar	100701928

Backend SRC - https://github.com/OwaisQuadri/ros-temphumid/
Frontend SRC: https://github.com/raveen15/ThermoCheck

Objective

To create an IoT application using the concepts learned in class to help a real world application.

Our Project

In many large scale buildings there is a need for a huge amount of electricity to run and operate. The way these factories get and produce electricity is through industrial size generators which take a huge amount of space and cost thousands of dollars. It is important to make sure this generator stays running and performs at their peak efficiency. Our application ThermoCheck is a web application that will allow the user to see the current temperature and humidity values of the generator to make sure they are operating in a stable environment to ensure longevity and reliability of the generators.

Project Design and Architecture

Hardware Design

To achieve the humidity and temperature values we use a DHT11 Temperature and Humidity sensor provided for us in our IOT kits. This sensor is an example of a basic, low cost sensor, it uses a capacitive humidity sensor and a thermostat to collect both temperature and humidity values. The NodeMCU is what will be used to power the sensor and also allows data transfer over WiFi. In other words the NodeMCU retrieves the temperature and humidity values from the DHT11 sensor and sends the data through WiFi.

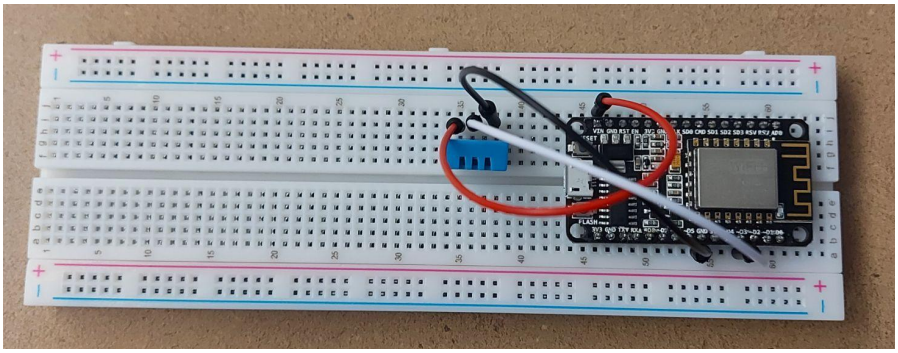
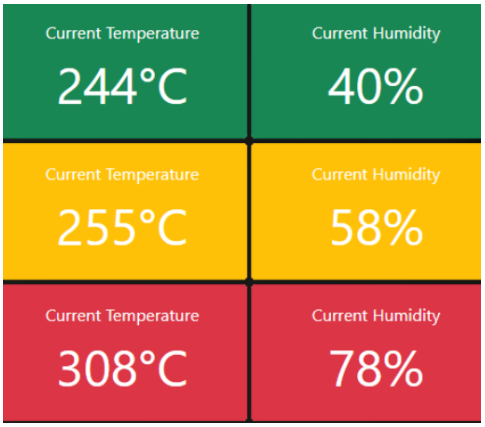


Figure 1: Hardware Setup

In the figure shown above is how the DHT11 Temperature and Humidity Sensor was connected to the NodeMCU board. The black wire represents the ground wire which was connected to the ground pins from both components. The white wire is the data wire meaning that it is connected to the data out pin on the DHT11 sensor to D3/GPIO0 pin on the NodeMCU board. Finally the red wire is voltage wire which is the wire responsible for giving power to the the DHT11 sensor. It is connected to the VCC pin on the DHT11 sensor to the VIN pin on the NodeMCU board. The NodeMCU board gets power via micro usb cable.

Web Application Design

To access our website the user will just need to go to this link https://www.thermocheck.xyz/ .The temperature and humidity values will be displayed at the forefront of our website.If the values of the generator are at stable conditions (Temperature <250 and Humidity<50%) then they will appear green to showcase healthy values, if the values are moderate(Temperature <300 and Humidity<75%) they will appear yellow to showcase moderate values. If the values are at dangerous levels (Temperature >300 and Humidity>75%) they will appear red. The user is also alerted of any dangerous levels when the website is updated through javascript each minute

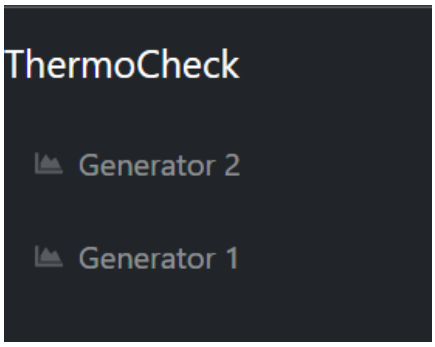


Next to the current temperature and humidity readings will be a table that displays 60 temperature and humidity ratings from the past hour alongside the time at when it was recorded. That way the user can see if the temperature was consistent or if the temperature has fluctuated in the past hour

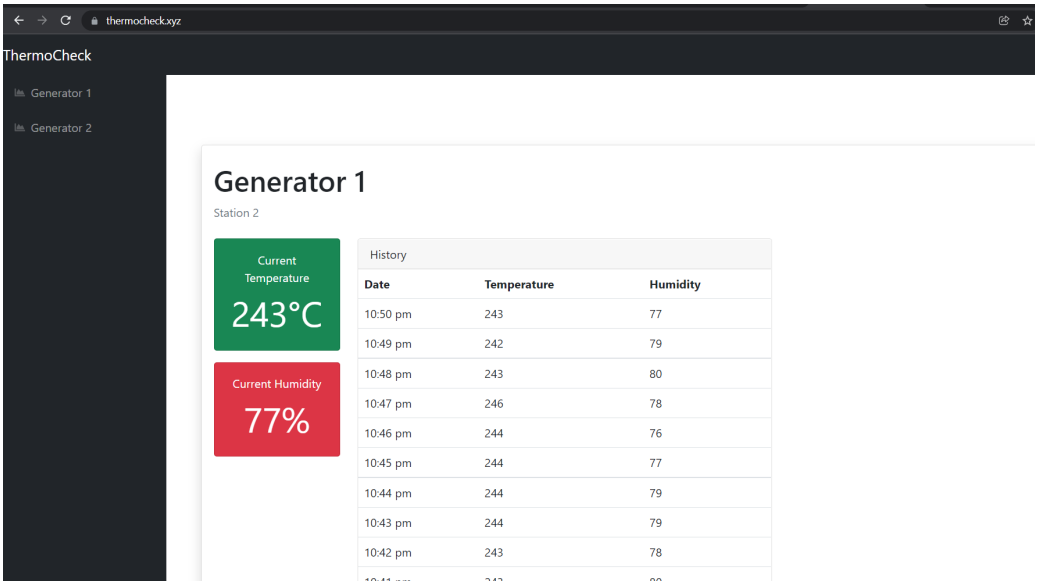
History

Date	Temperature	Humidity
10:34 am	277	59
10:33 am	292	61
10:32 am	292	56
10:31 am	254	45
10:30 am	261	45
10:29 am	255	48
10:28 am	257	53
10:27 am	300	48
10:26 am	274	55
10:25 am	249	66
10:24 am	277	42
10:23 am	255	54

On the left of our webpage will be a list of all the current generators connected to our applications. Multiple generators can be connected to our application. The user will be able to click on the generator and it will redirect them to the generators page and show current and past hour temperature and humidity values.



Webpage



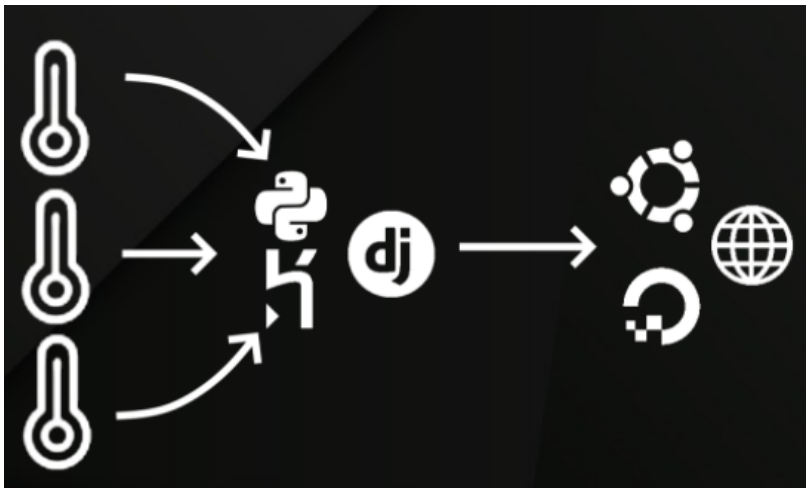
Demo Link:

https://drive.google.com/file/d/13lO846tY9wHI4pELv2PVcRK5YRgE15-f/view?usp=sharing

Architectural Design Decisions

For our architectural design, we implemented Django API and HTTP protocol to transfer the temperature and humidity values from the DHT11 sensor to our website. Our other option was to implement MQTT as our protocol to publish our data and to subscribed clients from an MQTT broker. However, we chose to use HTTP because it has more reliable and secure data transmission by implementing basic authentication tokens from our DHT11 sensor to the website. Also, this would allow us to get the current value of all the generators in one response instead of subscribing to an MQTT broker and then publishing the temperature and humidity values from the sensor.HTTP is a request/response model meaning it sends a request to the server and awaits an HTTP response back versus MQTT where it is publish and subscribe which is asynchronous service to service communication.

Deployment Design Decisions



For the deployment, we chose to deploy our application programming interface on Heroku since our API was developed in django and we are familiar with working with Heroku from previous activities and exercises. For the deployment of the web application, we used the Ubuntu virtual machine alongside the digital ocean platform that we learned how to implement in another one of the in-class-activities. When creating our cloud instance through Digital Ocean, we have been given a virtual host (which we use Ubuntu) with a given IP address. The Ubuntu host is where we configure our website (thermocheck.xyz) by using Apache 2 and the website has been purchased through godaddy.com. Once it was purchased, we placed the three Digital Ocean namespaces into our website. Once everything is configured, the website will be deployed based on the front end development code located in the directory ‘/var/www/thermocheck.xyz’. Within the code under ./js/scripts.js, this is where we retrieve our temperature, humidity, name, and date values from Django API and Heroku to display it on the website. It gets the information real time and outputs it for the user to see the values about each generator.

Conclusion

In conclusion, this project demonstrated the functionalities of IoT applications learned in class and implemented them to help a real world application of maintaining and monitoring industrial-sized generators in order to ensure the generators longevity, reliability and make sure they perform at their peak efficiency. Through extensive research and development, we have created an effective IoT system that would allow engineers to track generator temperatures and humidity levels to ensure that the generators are performing efficiently. If the system identifies dangerous temperature and humidity levels, the engineers will be notified through alerts so that they can take action immediately to prevent any damages from happening. Ultimately, we wanted our project to provide a service that will use the connection between our devices to save the time of our workers. After we realize the efficiency and effectiveness of our IoT application, we believe that IoT will be the future of industry monitoring technology.