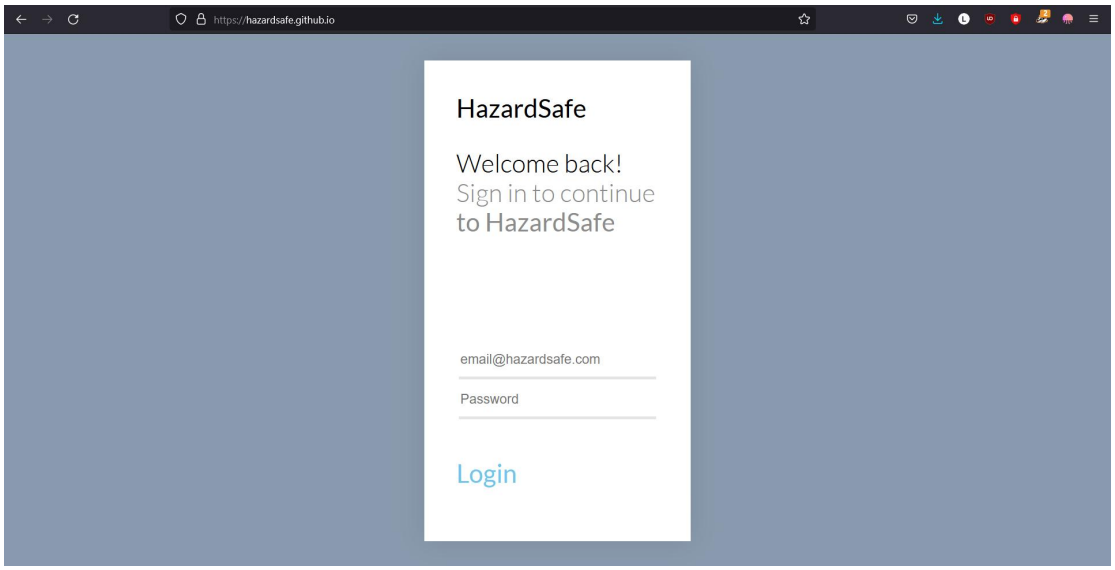
SPRINT 4

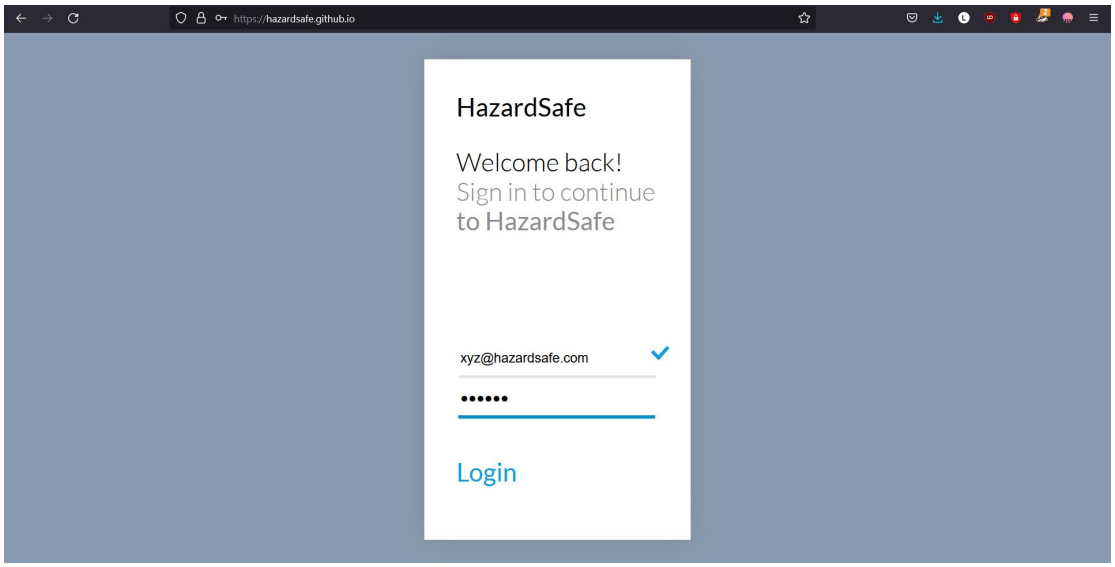
Date	14 November 2022
Team ID	PNT2022TMID27462
Project Name	Project - Hazardous Area Monitoring for Industrial Plant powered by IoT

Sprint 4 focuses on remote user functionality.

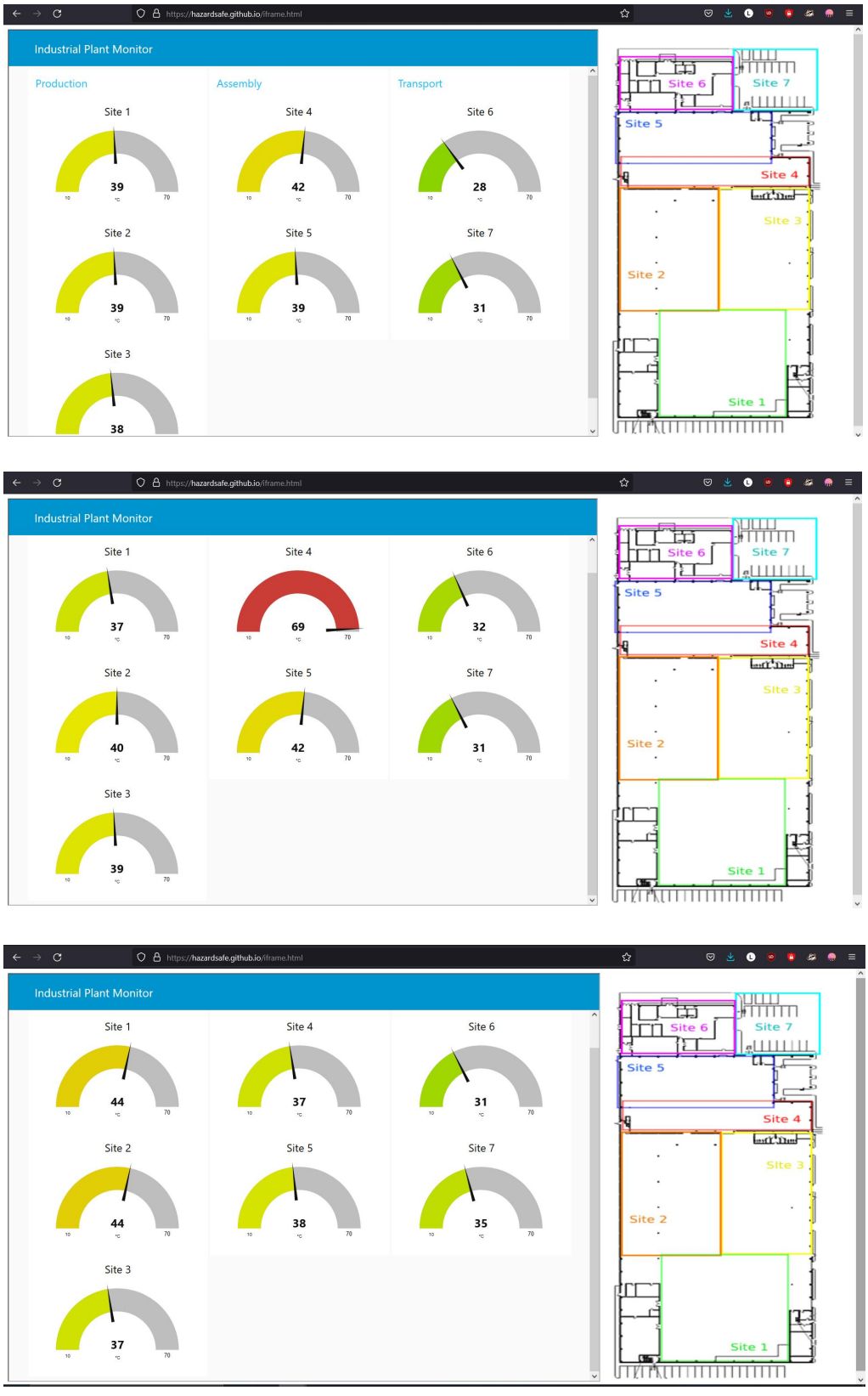
Web Application : hazardsafe.github.io

Login Page:



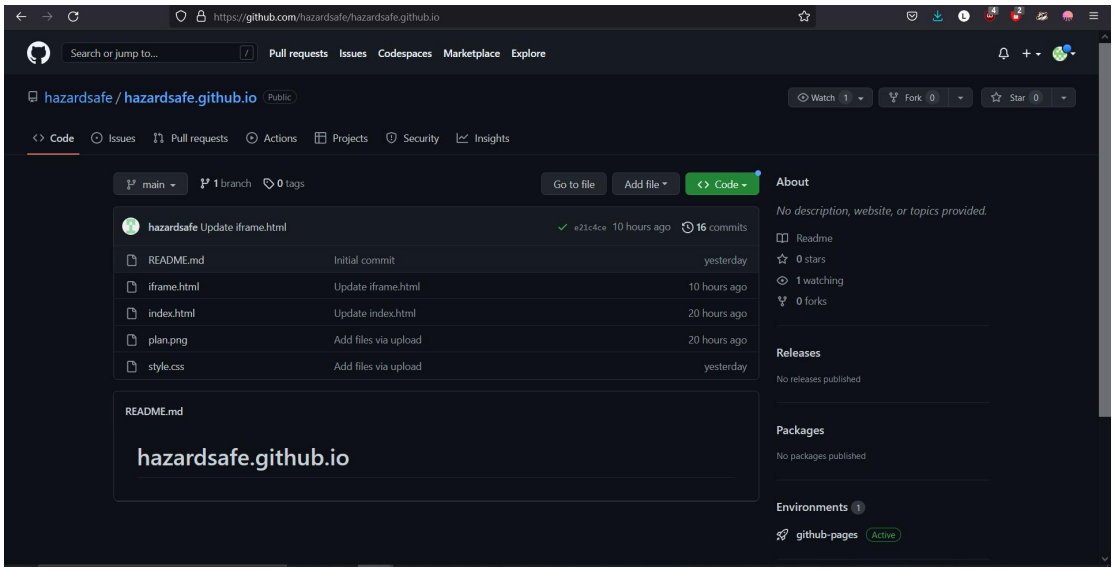


Full Site Remote Monitoring:



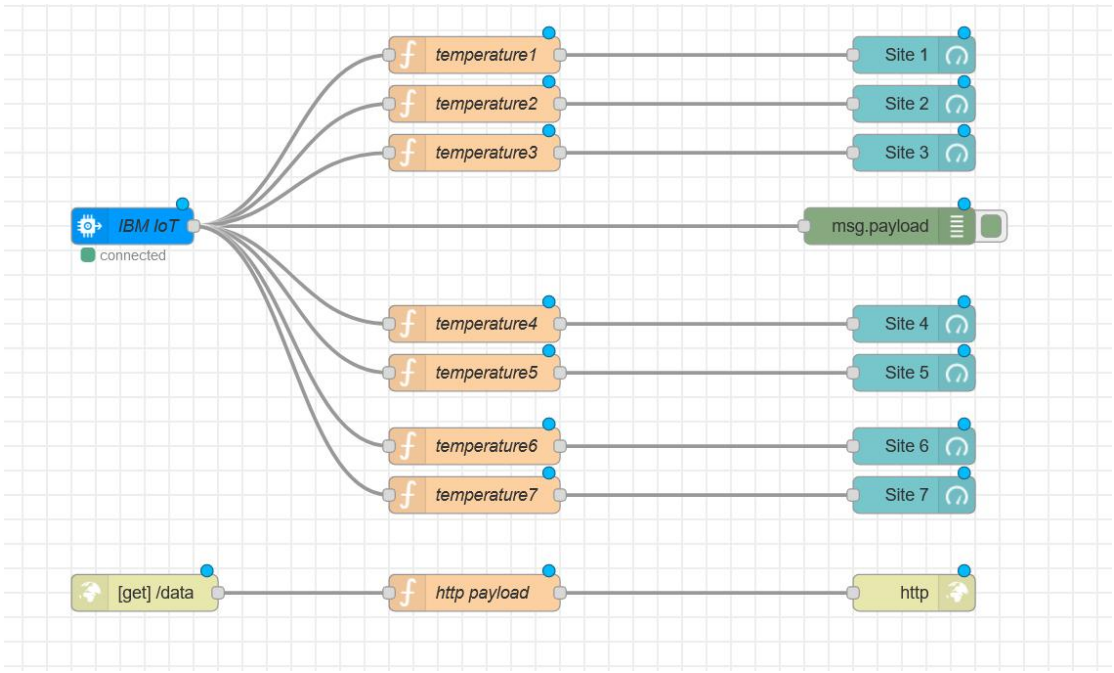
Real time data for the whole site can be monitored from this dashboard.

Website Hosting:



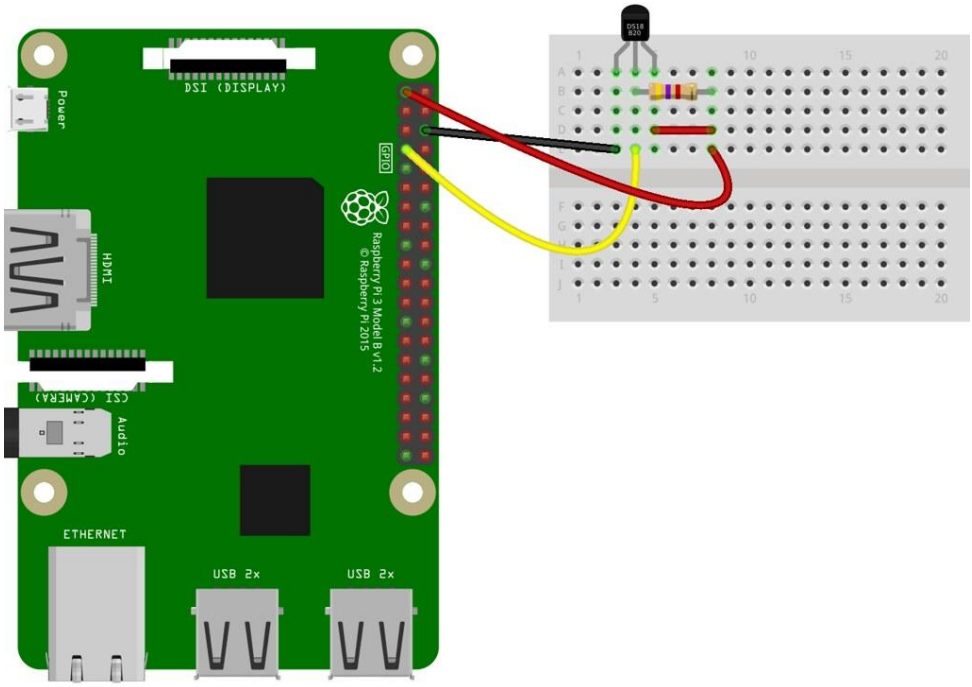
The website is hosted on [Github](#) for the time being.

Node-RED



1. Data is obtained from the IBM IoT device.
2. The temperature from each site is seperated and sent to its respective dashboard elements.
3. HTTP nodes send the JSON data to a URL where it can be used by other apps.

IoT Device:



Each device is running the following code to transmit its data to the cloud:

```
import time
import random
import paho.mqtt.client as mqtt
import json

#from w1thermsensor import W1ThermSensor
#sensor = W1ThermSensor()
#def Temp():
    #return temperature = sensor.get_temperature()

#The above lines of code would be used when getting temperature data from a DS18B20 sensor.
#Due to hardware limitations we are simulating values using random function.

def prodTemp():
    l = [40,40,39,39,39,38,38,37,37,37,41,42,44,41,42,42,43]
    return random.choice(l)

def assemTemp():
    l = [40,40,39,39,39,38,38,37,37,37,41,42,41,42,42,43,69,69,69]
    return random.choice(l)

def transTemp():
    l = [30,30,30,29,29,27,28,27,26,31,31,31,32,35,35,34,30,29,27]
    return random.choice(l)

#Due to hardware limitations we are simulating values for different areas using random
function.

ORG = "csgusn"
DEVICE_TYPE = "RPI"
TOKEN = "1123581321"
```

```
DEVICE_ID = "3c7c3f5b666d"

#Credentials of device as per created on IBM IoT platform.

server = ORG + ".messaging.internetofthings.ibmcloud.com";
pubTopic1 = "iot-2/evt/status1/fmt/json";

#Event named status1 sending data in json format.

authMethod = "use-token-auth";
token = TOKEN;
clientId = "d:" + ORG + ":" + DEVICE_TYPE + ":" + DEVICE_ID;

mqttc = mqtt.Client(client_id=clientId)
mqttc.username_pw_set(authMethod, token)
mqttc.connect(server, 1883, 60)

#Connecting via MQTT.

while True:

    tempDict = { "d": {"temperature1": prodTemp(),"temperature2": prodTemp(),"temperature3":
prodTemp(),"temperature4": assemTemp(),"temperature5": assemTemp(),"temperature6":
transTemp(),"temperature7": transTemp(),} };

    tempJson = json.dumps(tempDict);

    mqttc.publish(pubTopic1, tempJson)
    print("Reading Taken");

    time.sleep(5);
```