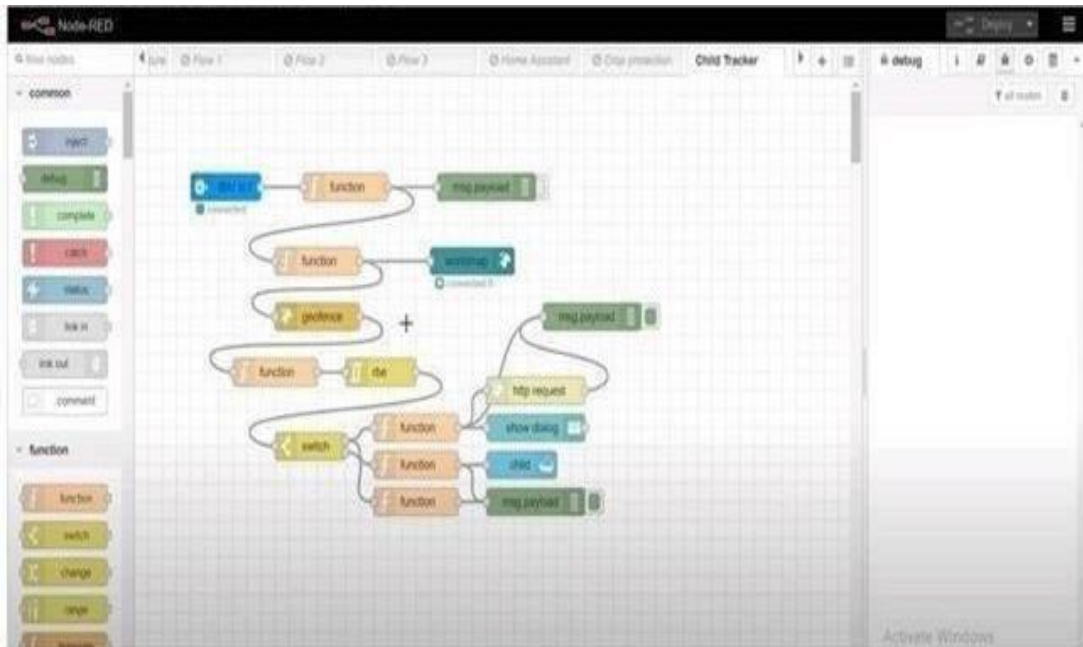


# SPRINT -3

## CREATING GEO-FENCING

### 1. BUILD A NODE- RED :



### 2. ADD A CODE TO GET CHILD LOCATION IN PYTHON :

```
import json
import wiotp.sdk.device
import time

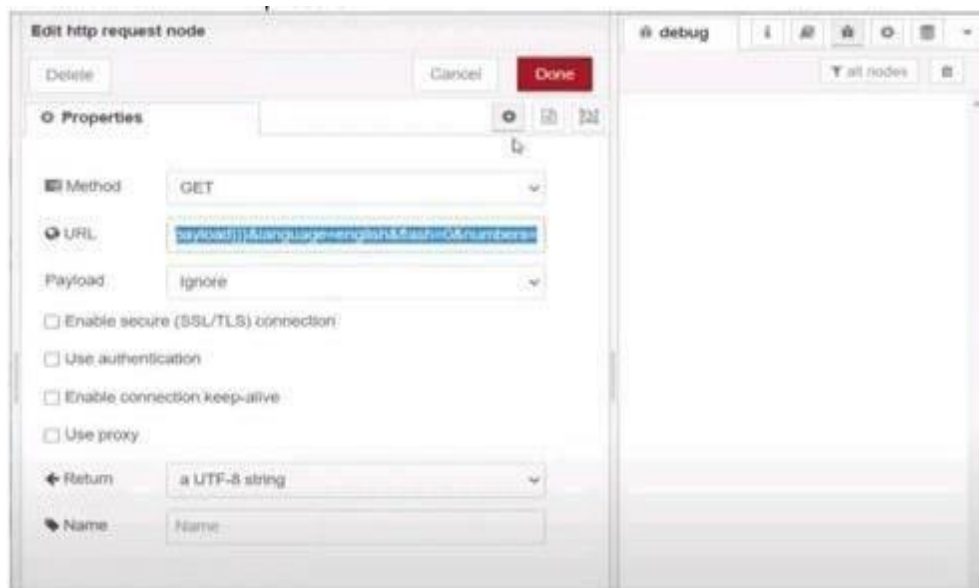
myConfig = {
    "identity": {
        "orgId": "hj5fmy",
        "typeId": "NodeMCU",
        "deviceId": "12345"
    },
    "auth": {
        "token": "12345678"
    }
}
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    name= "Smartbridge"
    #in area location
    latitude= 17.4225176
    longitude= 78.5458842

    #out area location
    #latitude= 17.4219272
    #longitude= 78.5489783
    myData={'name': name, 'lat':latitude, 'lon':longitude}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Data published to IBM IoT platform: ",myData)
    time.sleep(5)

client.disconnect()
```

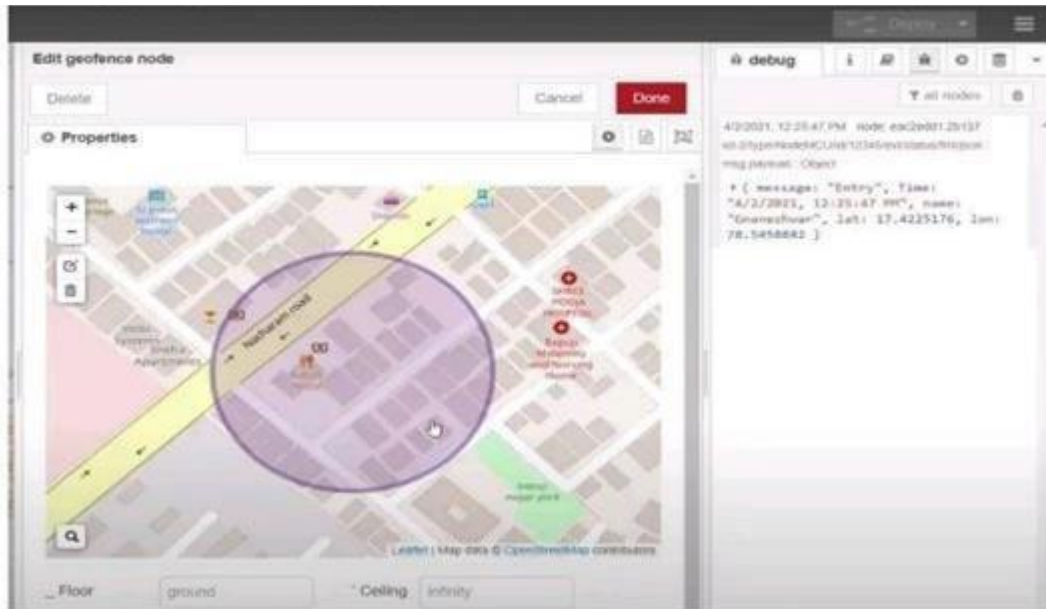
### 3. EDIT THE HTTP REQUEST URL :



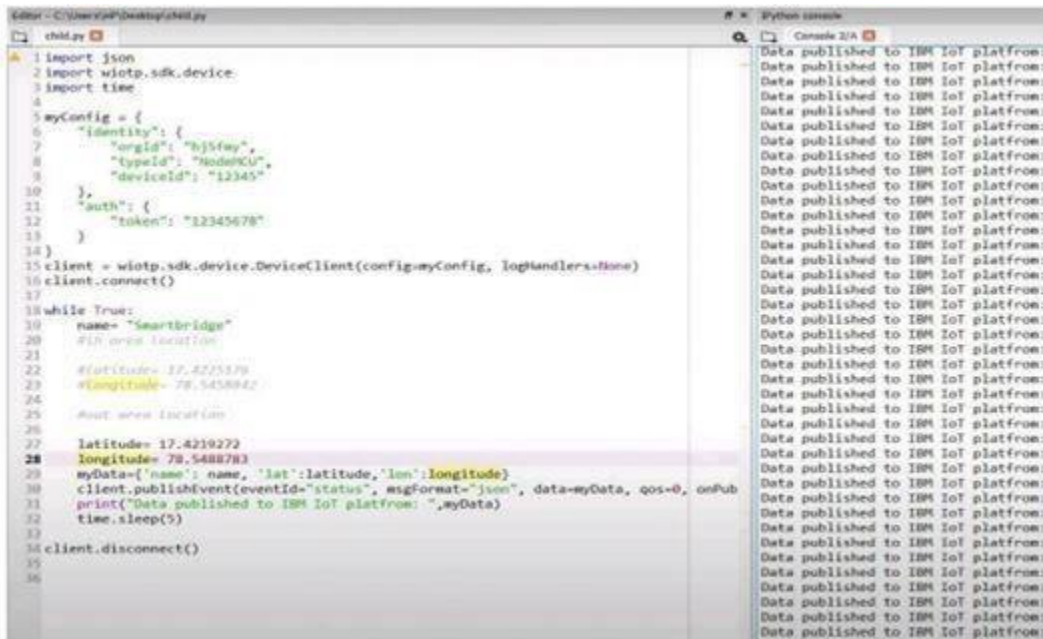
### 4. LIVE LOCATION OF THE CHILD



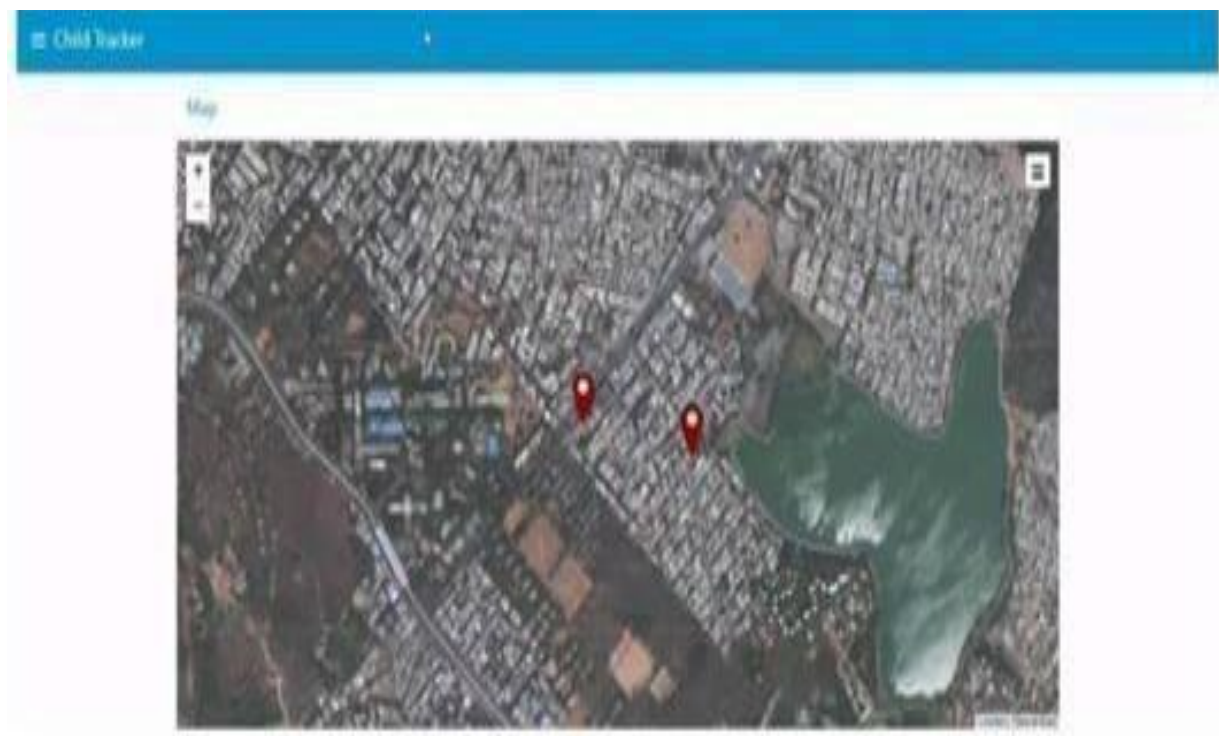
## 5. CREATE THE GEOFENCE NODE :



## 6.PYTHON SCRIPT SEND REQUEST TO IBM CLOUD :



**7.RESULT : “After Running The Script, The Web UI show  
The CurrentLocation & The Activity Of The Child”**



# SENSING THE TEMPERATURE AND HUMIDITY USING SENSORS.

Step 1: After creating the IBM Watson IoT platform, the code for measuring the temperature and humidity is implemented.

The screenshot displays the IBM Watson IoT Platform interface. On the left, a sidebar contains navigation icons. The main panel is titled 'IBM Watson IoT Platform' and features a 'Browse' tab. Below the tab, a table lists events with columns 'Event' and 'Value'. The table contains five rows, all labeled 'event\_1', with values representing random data for temperature, humidity, and a random number. A modal window on the right, titled 'Device Type: Iot', shows the configuration for 'event\_1'. It includes a 'Send' button, a 'Schedule' section set to 'Every Minute', and a 'Payload' section with a JSON structure for random data. The payload is defined as follows:

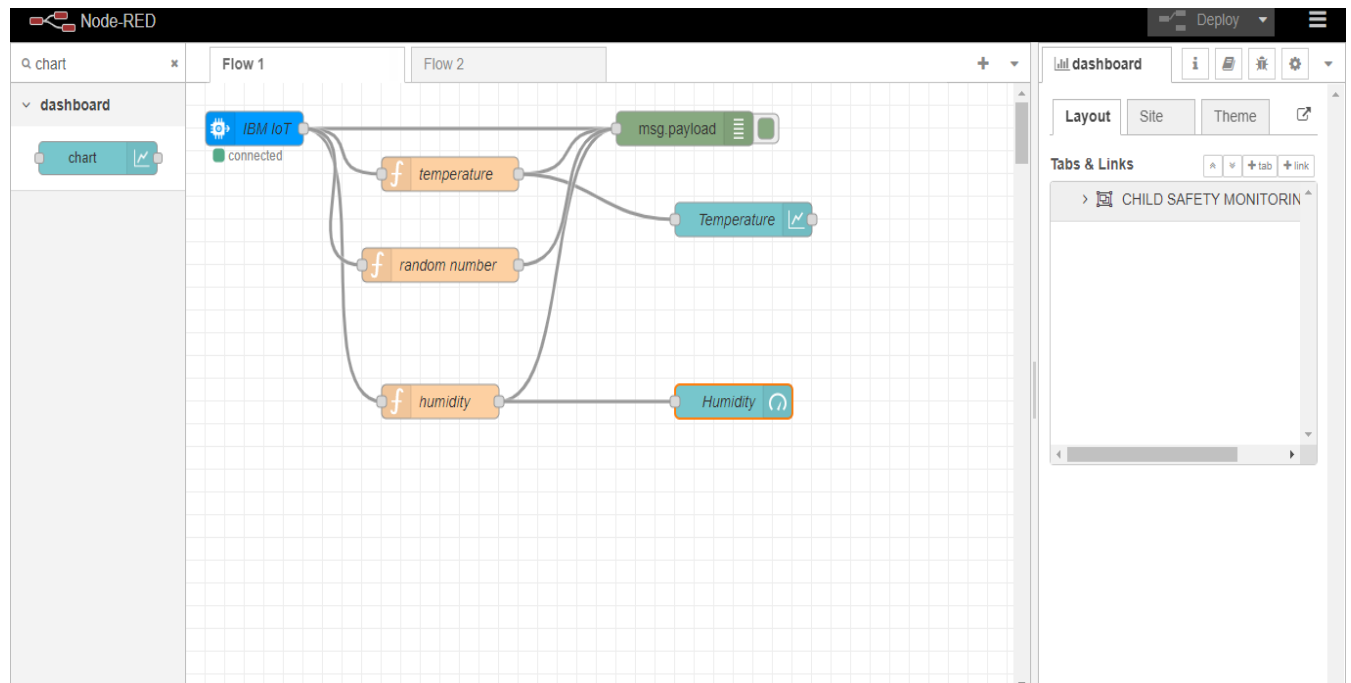
```
0 {
1   "randomNumber": random(0, 100),
2   "temp": random(0, 100),
3   "hum": random(0, 100)
4 }
5
```

At the bottom of the modal, there is an 'Upload a CSV file' button and 'Cancel' and 'Save' buttons.

Event	Value
event_1	{"randomNumber":34,"temp":12,"hum":77}
event_1	{"randomNumber":92,"temp":93,"hum":32}
event_1	{"randomNumber":82,"temp":84,"hum":38}
event_1	{"randomNumber":47,"temp":93,"hum":98}
event_1	{"randomNumber":76,"temp":46,"hum":24}

Items per page 50 | 1-1 of 1 item

Step 2: The blocks are connected in the Node-RED and simultaneously run the code in the IBM Watson IoT platform.



Step 3: After deploying the code in the Node-RED, the values of the temperature and humidity are randomly generated.

