

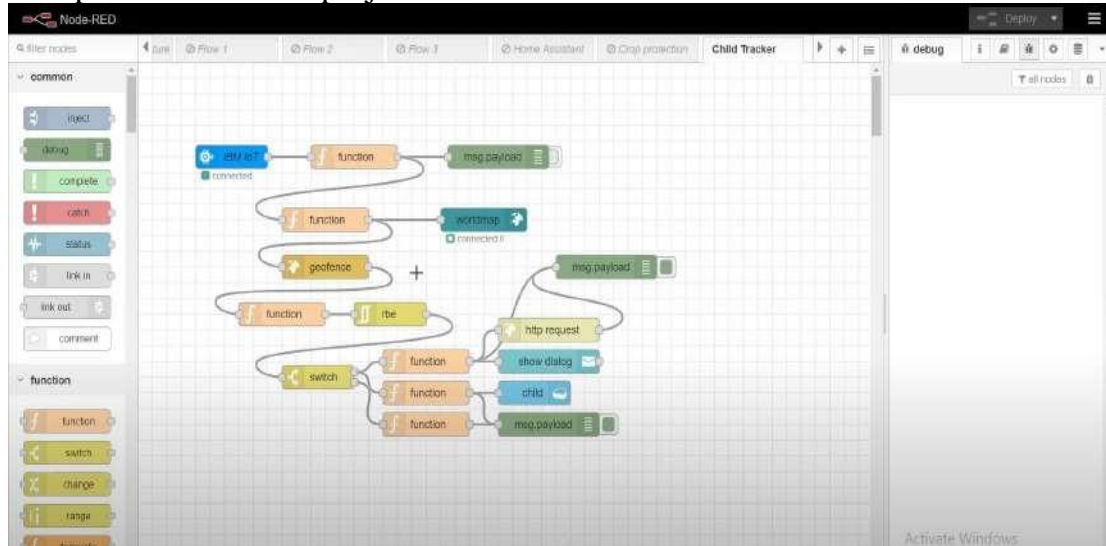
# Develop The Web Application Using Node-RED

## PNT2022TMID25493

### 1. To Develop the web application using Node-RED

Steps :

#### 1. Open a Node-RED project



#### 2. Add 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

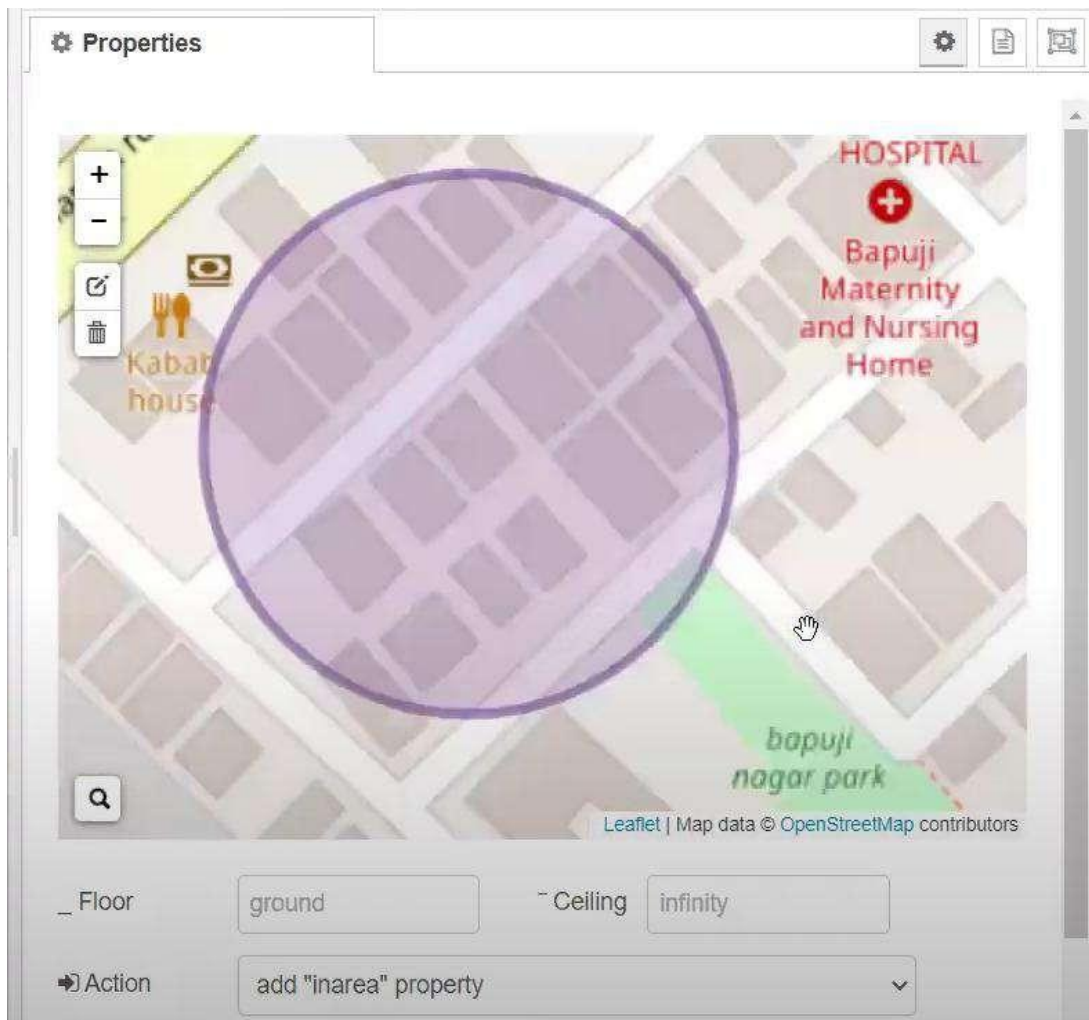
    latitude= 17.4225176
    longitude= 78.5458842

    #out area location

    #latitude= 17.4219272
    #longitude= 78.5488783
    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. Create the Geofence



#### 4. Edit the HTTP Request URL

The screenshot shows the 'Edit http request node' dialog box. At the top, there are buttons for 'Delete', 'Cancel', and 'Done'. Below these is a 'Properties' tab. The 'Method' is set to 'GET'. The 'URL' field is highlighted with a blue selection box containing the text 'payload}}&language=english&flash=0&numbers='. The 'Payload' is set to 'Ignore'. There are four unchecked checkboxes: 'Enable secure (SSL/TLS) connection', 'Use authentication', 'Enable connection keep-alive', and 'Use proxy'. The 'Return' dropdown is set to 'a UTF-8 string'. The 'Name' field contains the text 'Name'. To the right of the dialog is a 'debug' panel with a toolbar and a search bar labeled 'all nodes'.

Delete Cancel Done

Properties

Method GET

URL payload}}&language=english&flash=0&numbers=

Payload Ignore

☐ Enable secure (SSL/TLS) connection

☐ Use authentication

☐ Enable connection keep-alive

☐ Use proxy

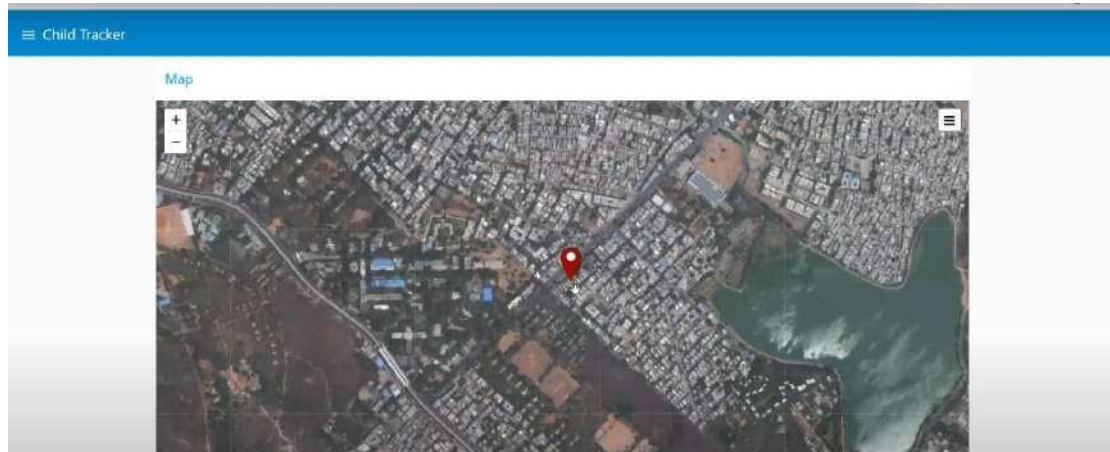
Return a UTF-8 string

Name Name

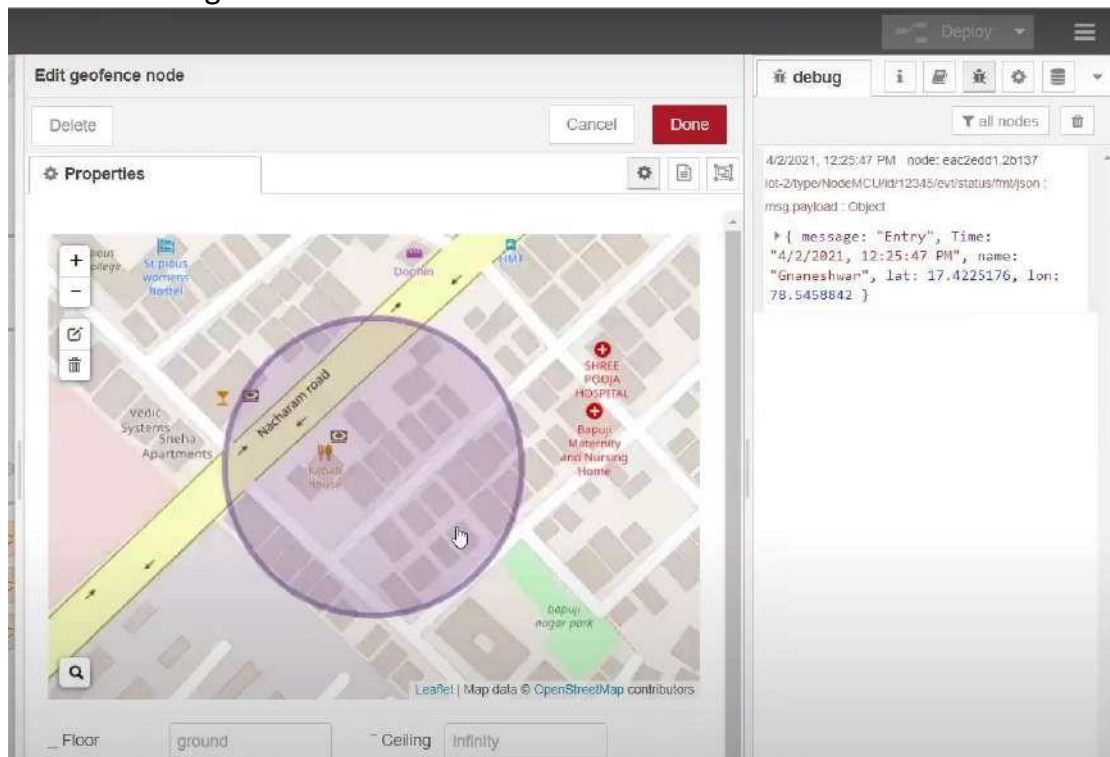
debug

all nodes

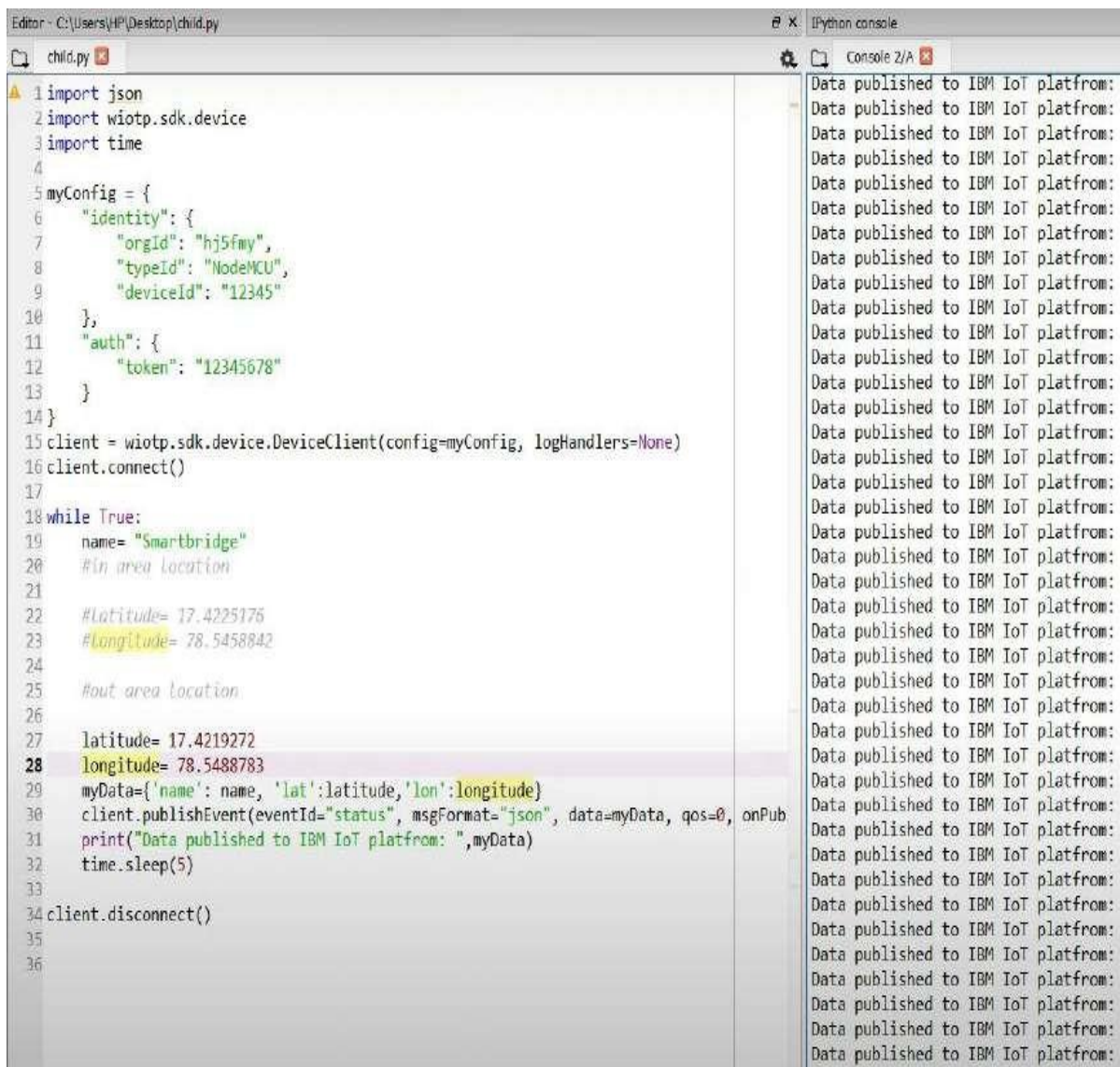
## 5. Locate the child



## 6. Create the geofence node



## 7. Python script send requests to IBM Cloud



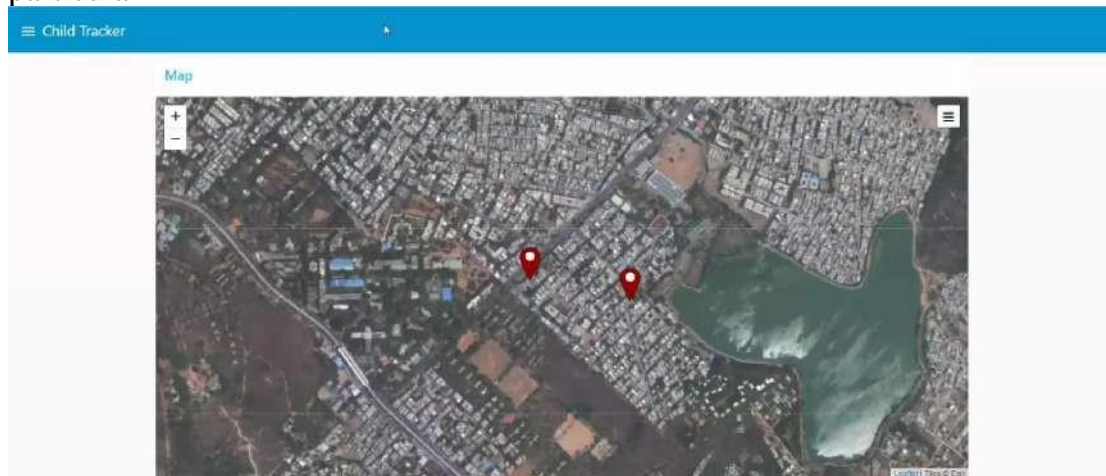
The screenshot shows a Python script in an IDE window titled 'Editor - C:\Users\HP\Desktop\chld.py'. The script is named 'chld.py' and contains the following code:

```
1 import json
2 import wiotp.sdk.device
3 import time
4
5 myConfig = {
6     "identity": {
7         "orgId": "hj5fmy",
8         "typeId": "NodeMCU",
9         "deviceId": "12345"
10    },
11    "auth": {
12        "token": "12345678"
13    }
14 }
15 client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
16 client.connect()
17
18 while True:
19     name= "Smartbridge"
20     #in area location
21
22     #Latitude= 17.4225176
23     #Longitude= 78.5458842
24
25     #out area location
26
27     latitude= 17.4219272
28     longitude= 78.5488783
29     myData={'name': name, 'lat':latitude,'lon':longitude}
30     client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPub
31     print("Data published to IBM IoT platform: ",myData)
32     time.sleep(5)
33
34 client.disconnect()
35
36
```

The console window on the right, titled 'Console 2/A', shows the output of the script. It displays a repeating message: 'Data published to IBM IoT platform:'. The message is truncated in the console view, but the full message as seen in the print statement is 'Data published to IBM IoT platform: ',myData'.



8. After running the script, the web UI shows “Person is not in the particular area”



**Conclusion:**

Developed the web application using Node-RED Successfully