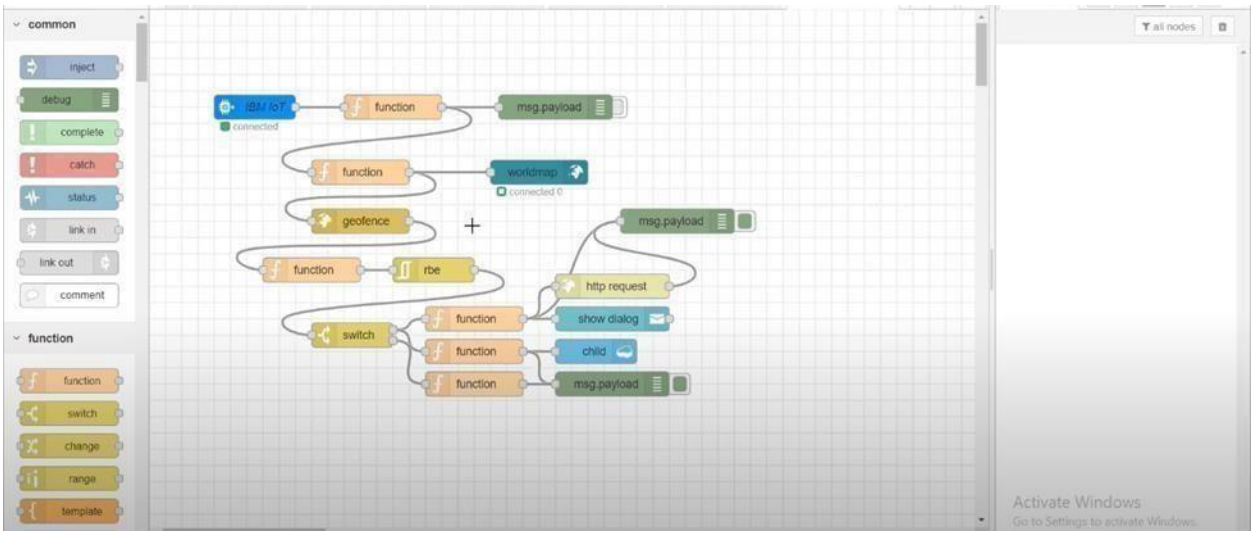
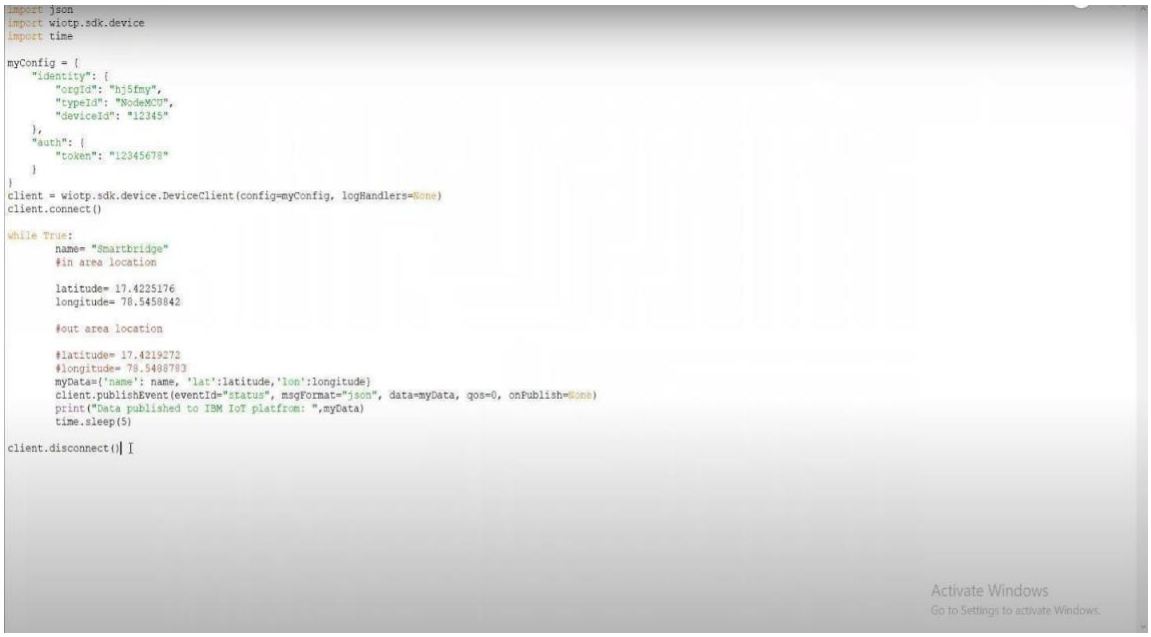
Team ID	**PNT2022TMID54021**
Project Name	**Smart Waste Management for Metropolitan Cities**

NODE RED SERVICE

Step 1: Connect the blocks.

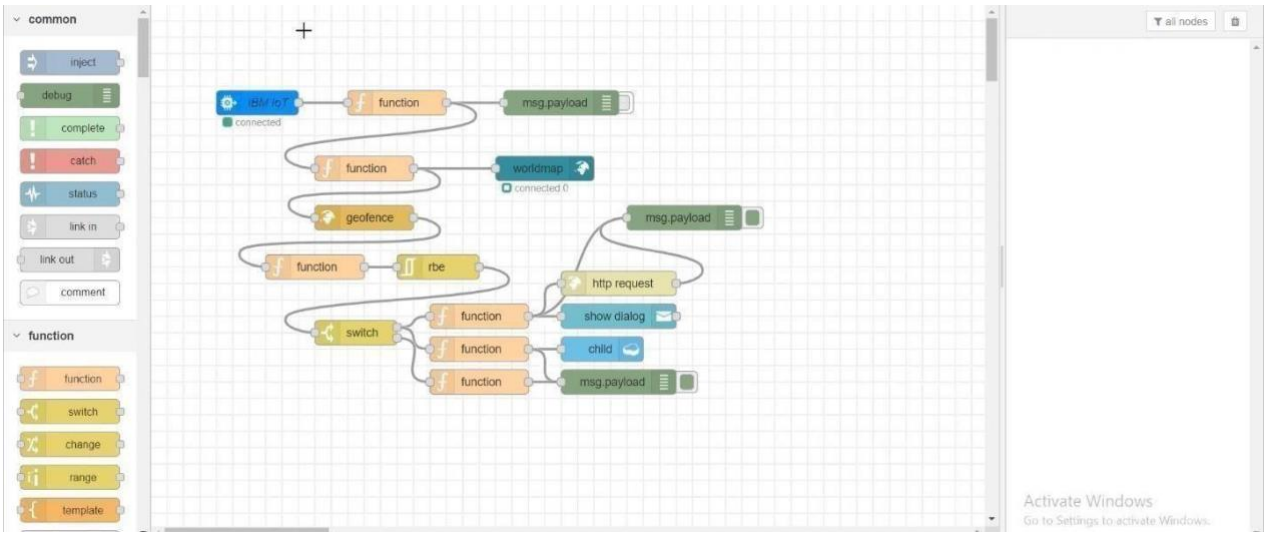




```
import json
import wiotp.sdk.device
import time

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= 78.5498703
        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 platfrom: ",myData)
        time.sleep(5)

client.disconnect()
```

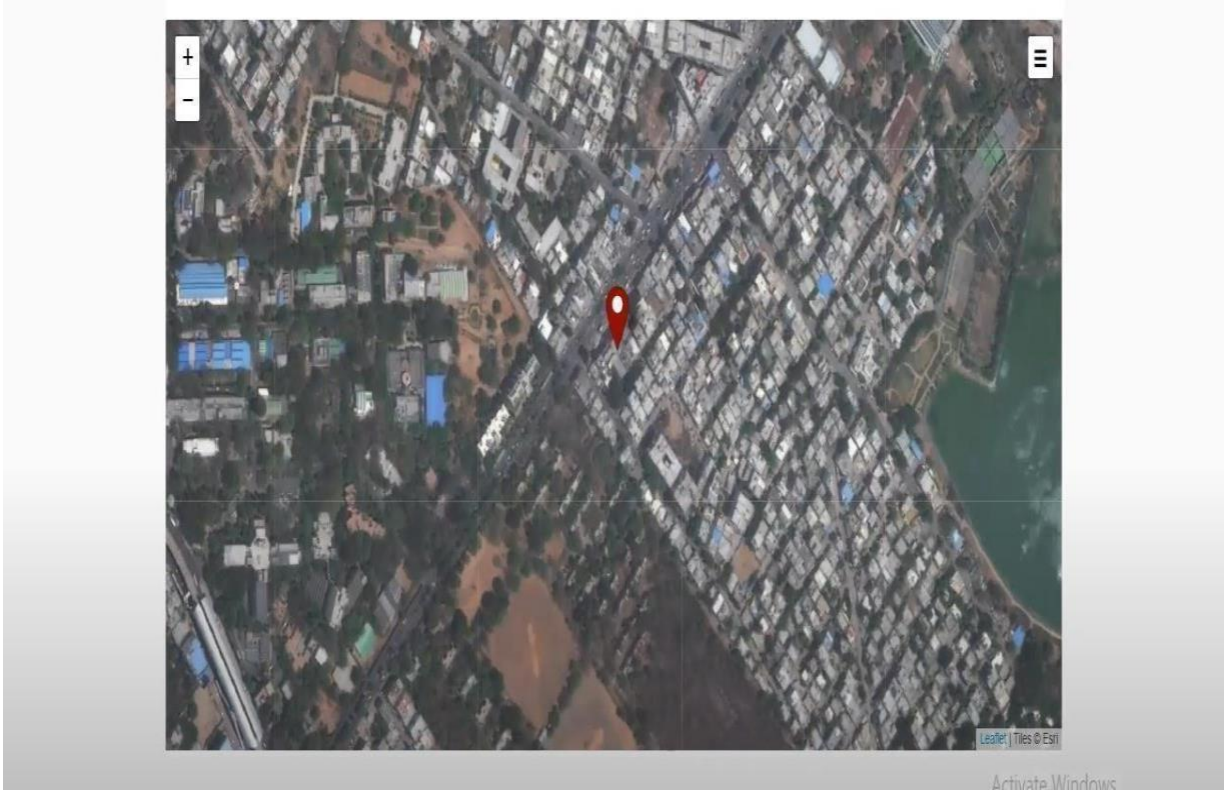
Step 2: Create python code.

Step 3: Click the geo-fence node.

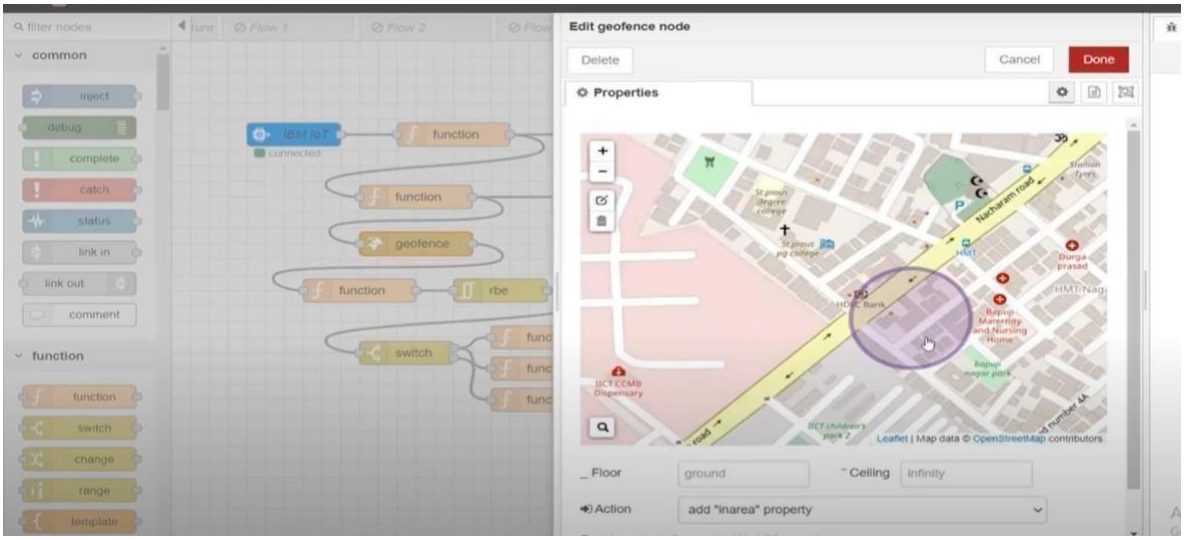


Step 4: Create the geo-fence area in the map.

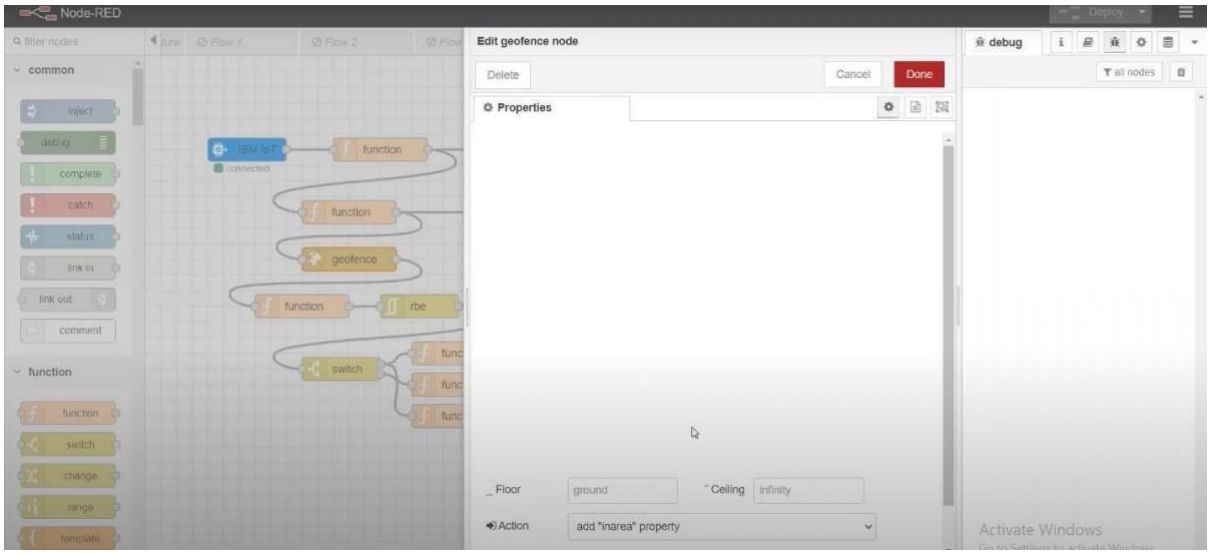
Step



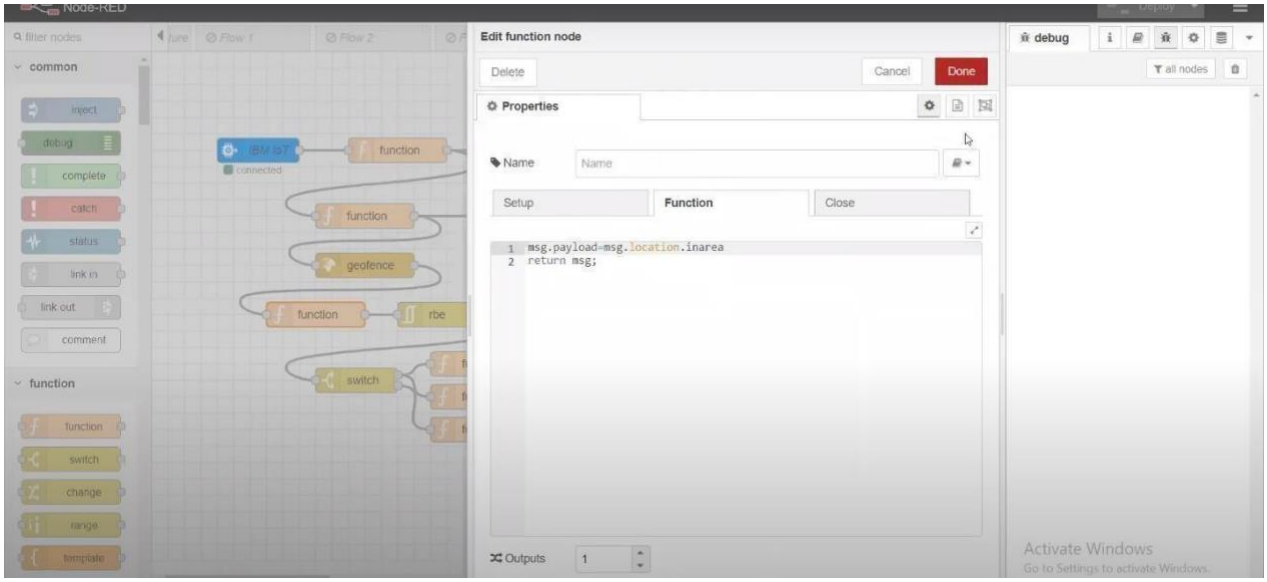
5: Create geo-fence in a particular area.



Step 6: Select the function block.

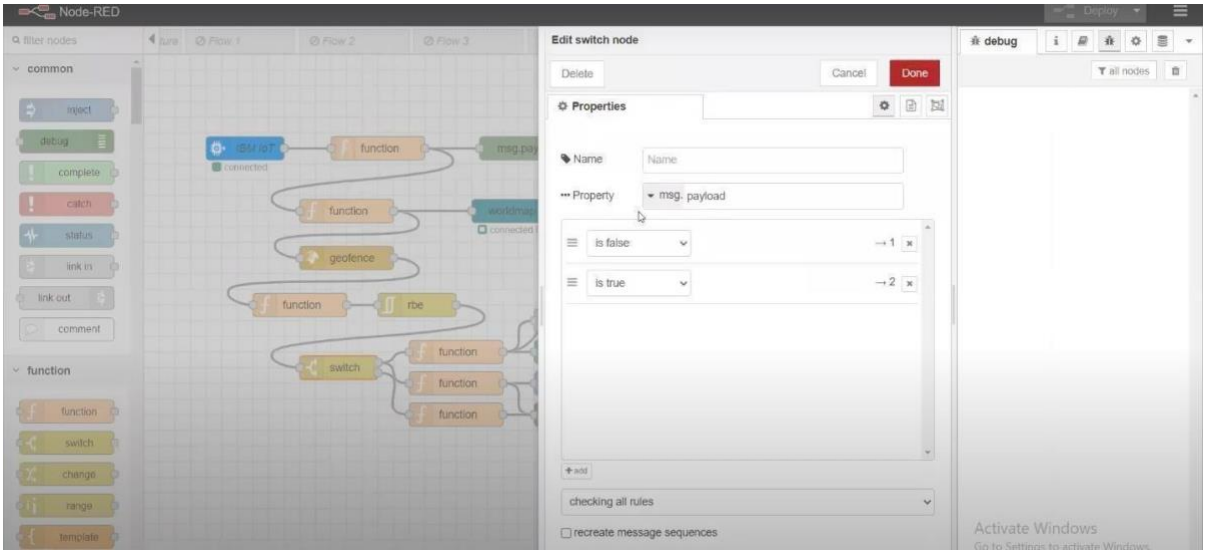


Step 7: Select the message payload.

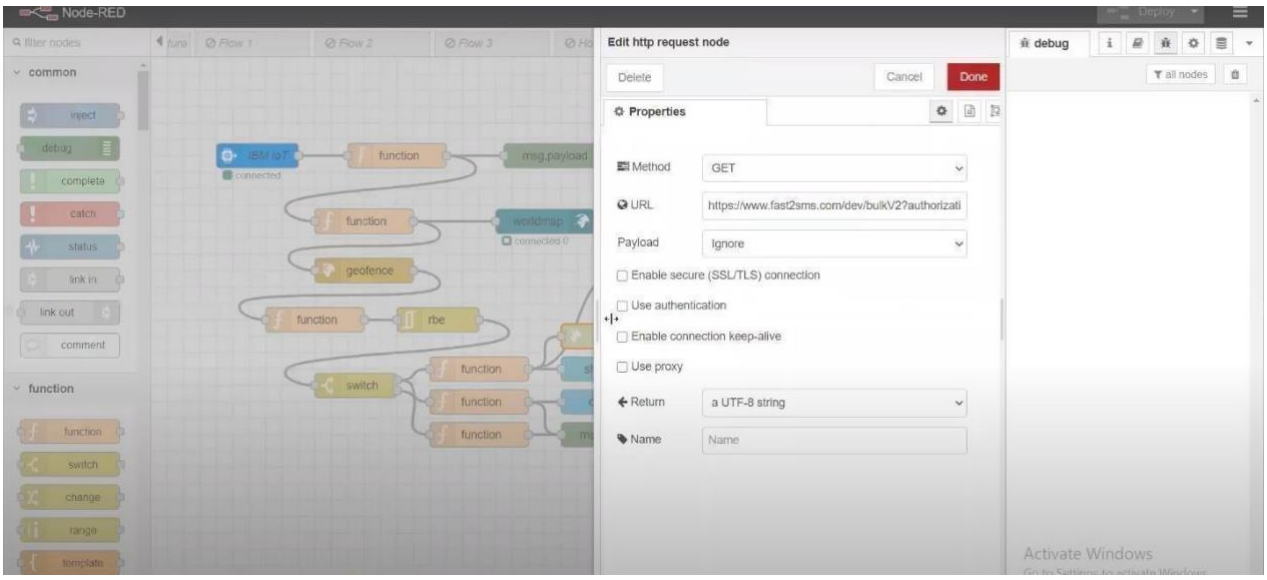


Step 8: To identify the person in area.

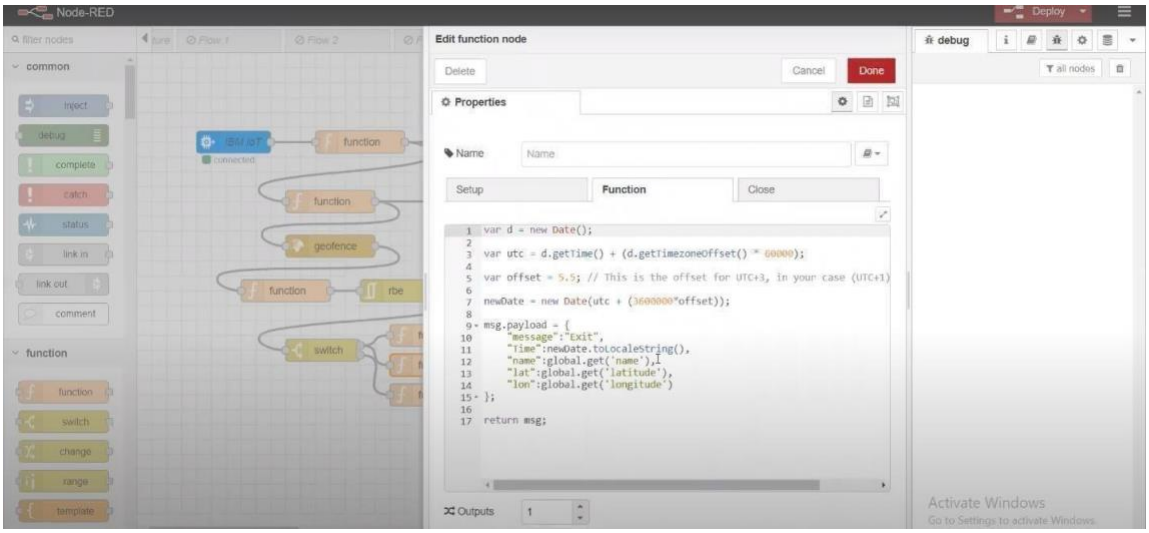
Step



9: Select the http request to send message to parent or guardian.

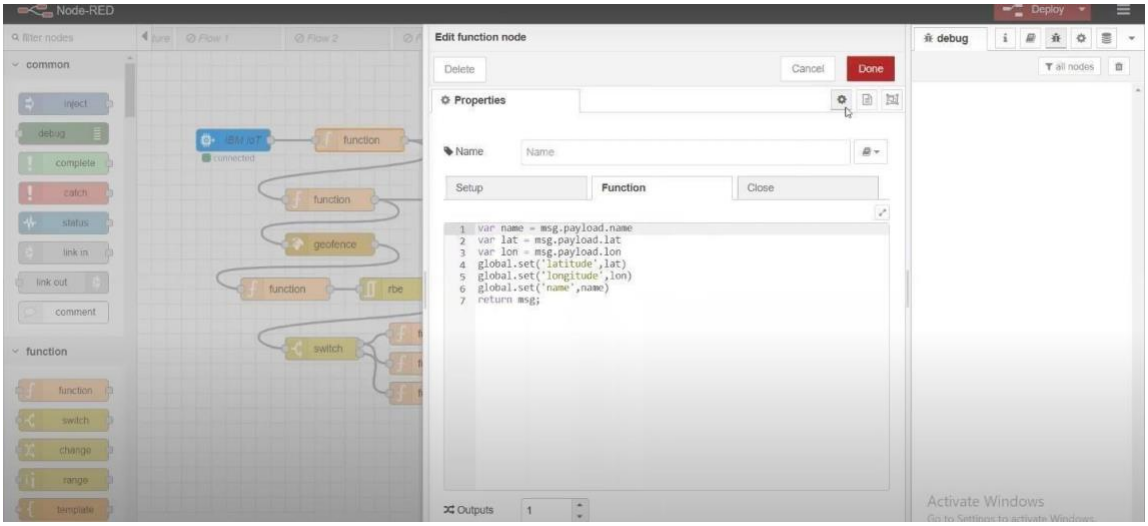


Step 10: For sending the message with time.

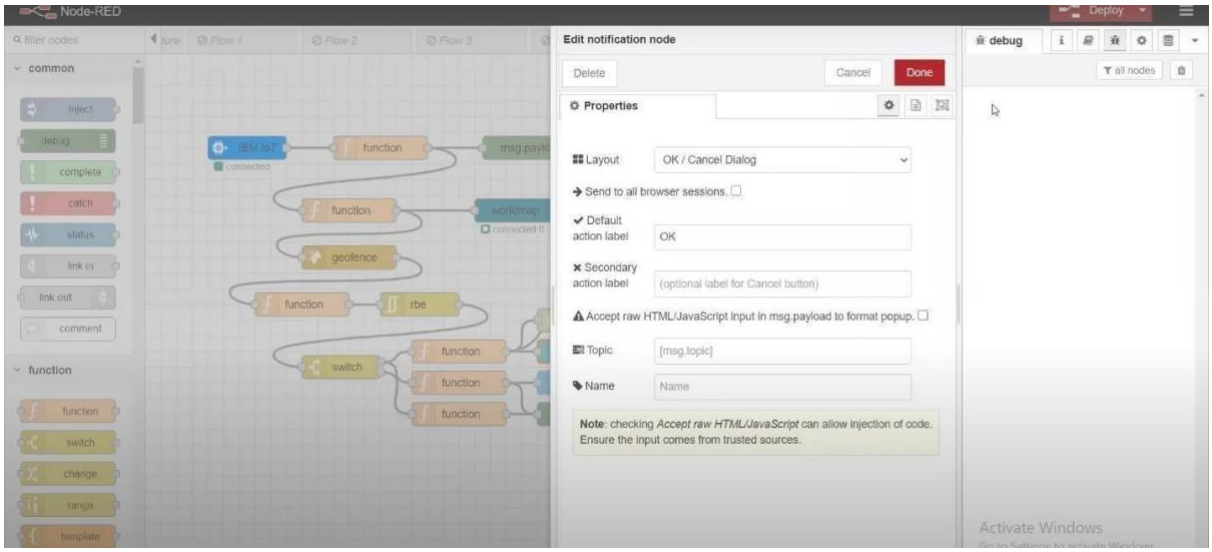

Node-RED
Deploy
filter nodes
common
inject
debug
complete
catch
status
link in
link out
comment
function
function
switch
change
range
template
Flow 1
Flow 2
IBM IoT
connected
function
function
geofence
function
rbe
switch
Edit function node
Delete
Cancel
Done
Properties
Name
Name
Setup
Function
Close
1 var d = new Date();
2
3 var utc = d.getTime() + (d.getTimezoneOffset() * 60000);
4
5 var offset = 5.5; // This is the offset for UTC+3, in your case (UTC+1)
6
7 newDate = new Date(utc + (3600000*offset));
8
9 msg.payload = {
10 "message":"Exit",
11 "time":newDate.toLocaleString(),
12 "name":global.get('name'),
13 "lat":global.get('latitude'),
14 "lon":global.get('longitude')
15 };
16
17 return msg;
Outputs
1
debug
all nodes
Activate Windows
Go to Settings to activate Windows.

Step

10: Click show dialog for notifying the popup alert.



Step


Node-RED
Deploy
filter nodes
Flow 1
Flow 2
Flow 3
common
inject
debug
complete
catch
status
link in
link out
comment
function
function
switch
change
range
template
function
function
msg.payl
function
worldmap
geofence
function
rbe
switch
function
function
function
connected
Edit notification node
Delete
Cancel
Done
Properties
Layout
OK / Cancel Dialog
Send to all browser sessions.
Default action label
OK
Secondary action label
(optional label for Cancel button)
Accept raw HTML/JavaScript input in msg.payload to format popup.
Topic
[msg.topic]
Name
Name
Note: checking Accept raw HTML/JavaScript can allow injection of code. Ensure the input comes from trusted sources.
debug
all nodes
Activate Windows

Step

Step 11: Create another payload and to pass the data to geo-fence and world map.

Step

12: Click the world map to see the location.