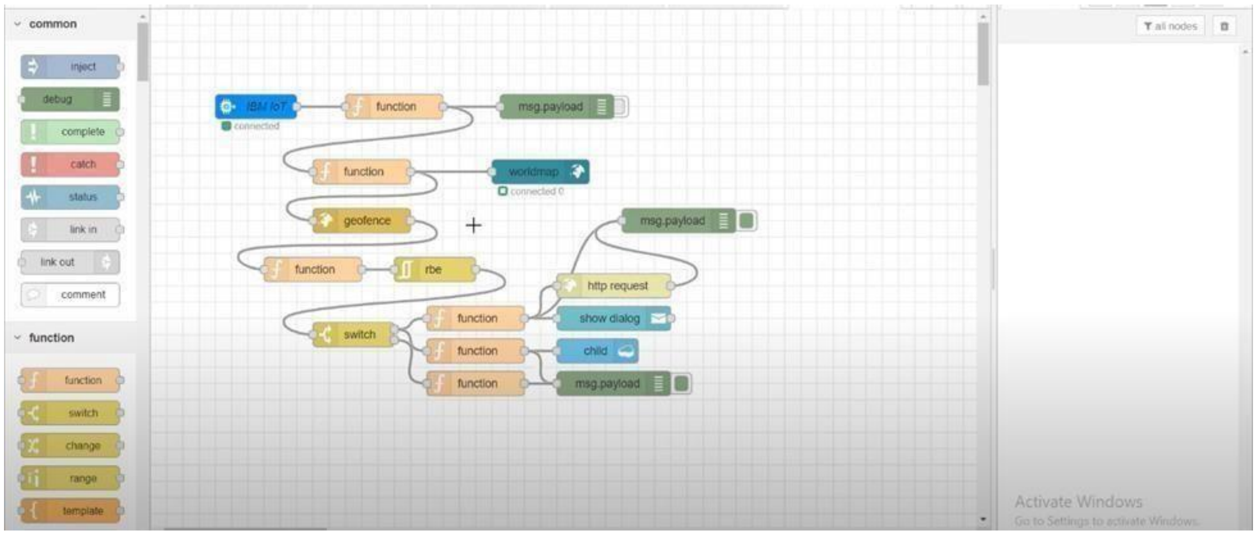
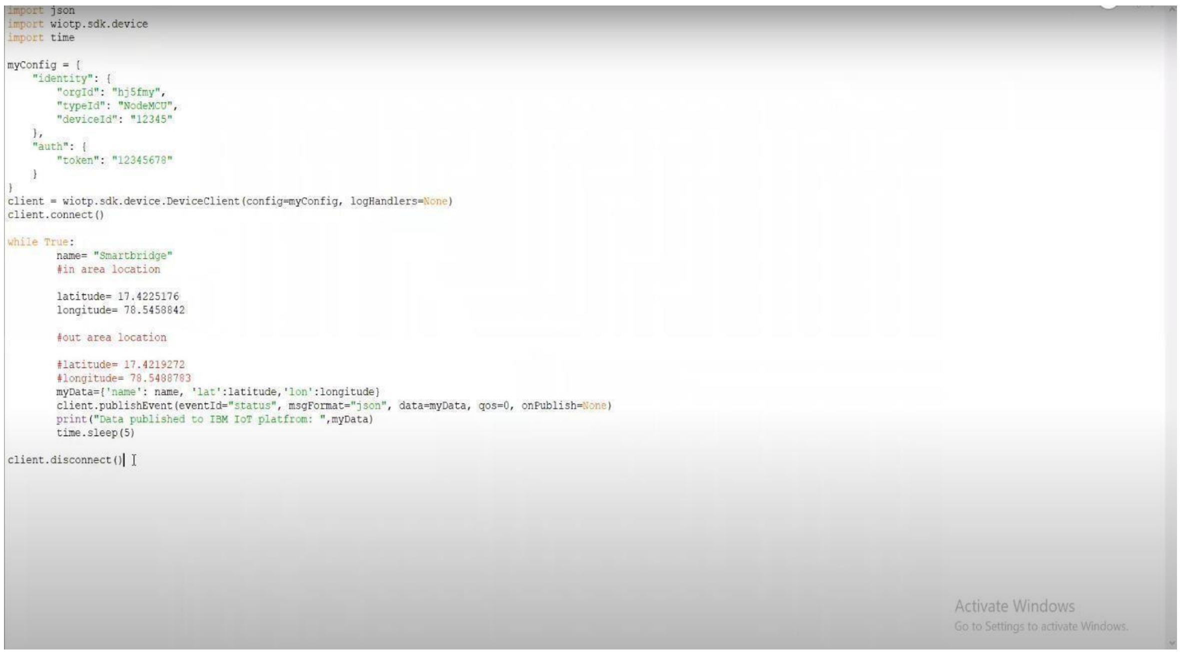
Team id: PNT2022TMID49101
Project Name: Smart Waste Management for Metropolitan Cities

NODE RED SERVICE

Step 1: Connect the blocks.



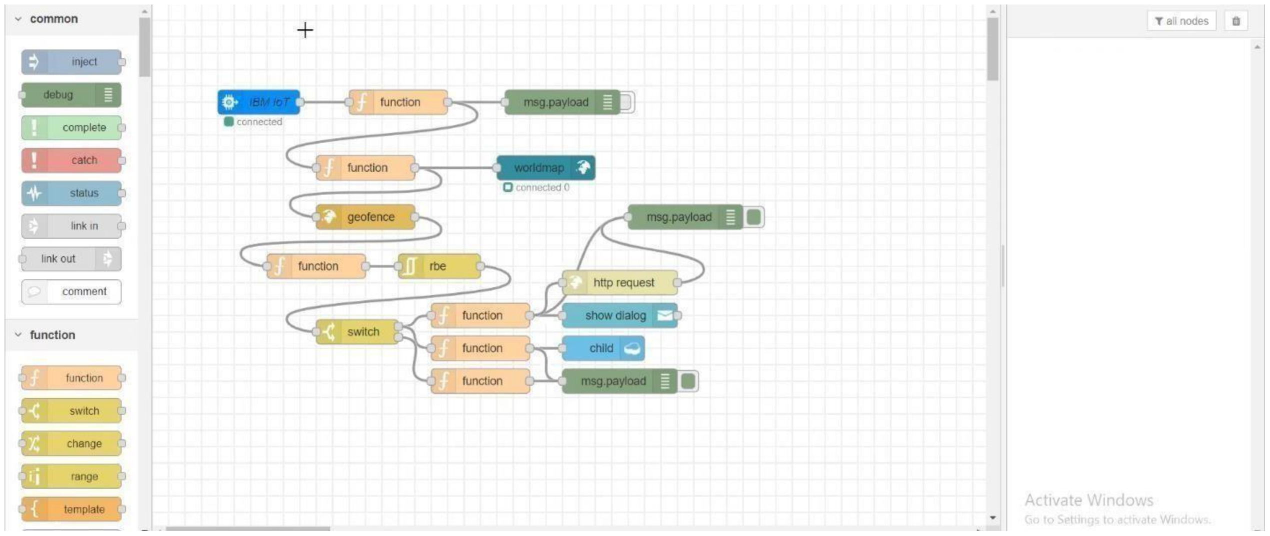


```
import json
import wiotp.sdk.device
import time

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= 78.5488783
        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 platfrom: ",myData)
        time.sleep(5)

client.disconnect()
```

Activate Windows
Go to Settings to activate Windows.

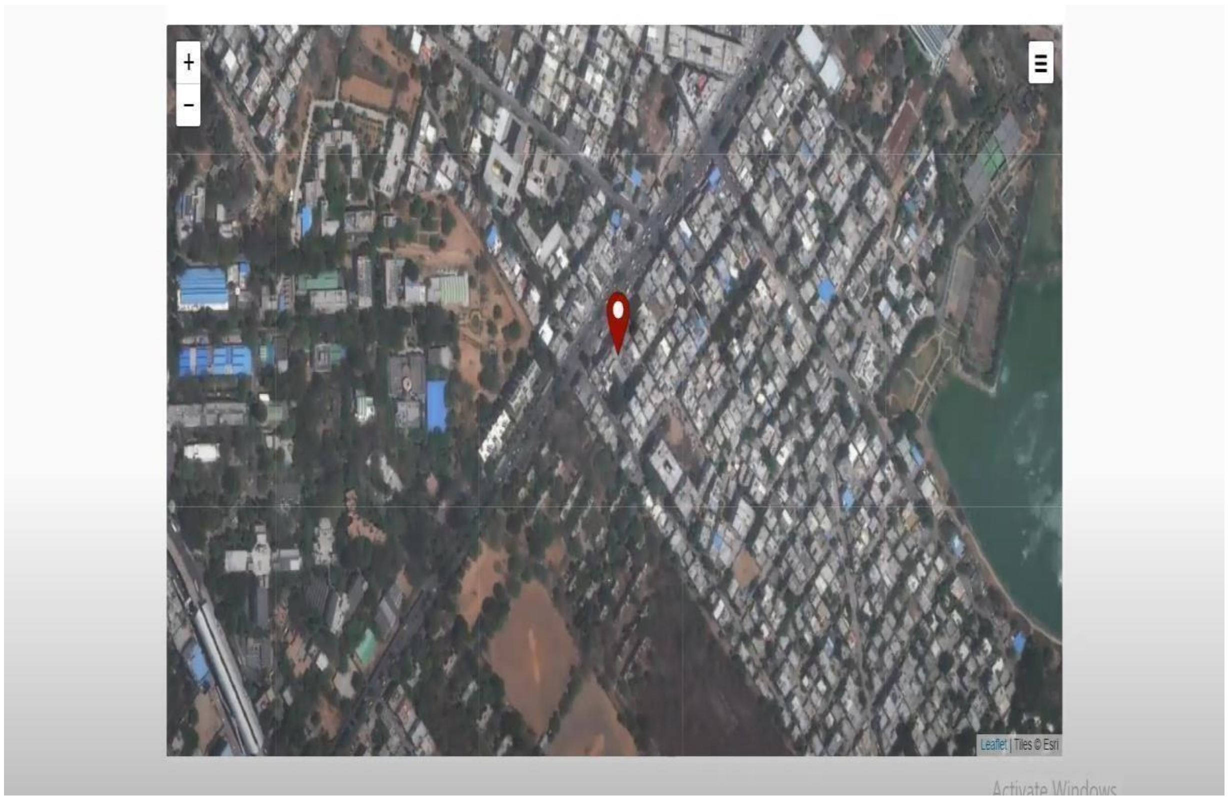
Step 2: Create python code.

Step 3: Click the geo-fence node.

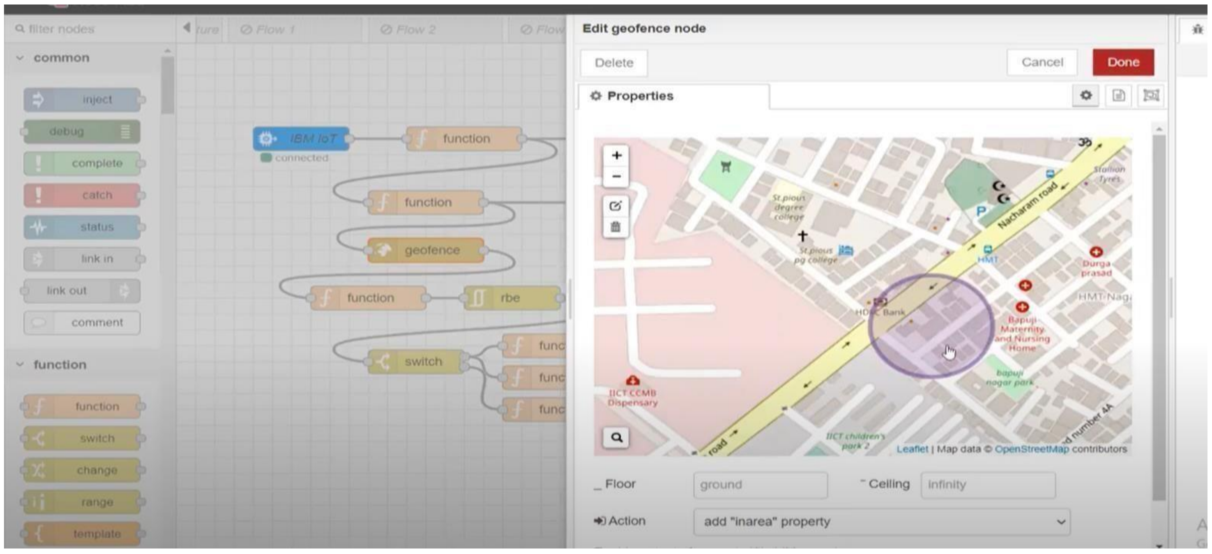


Step 4: Create the geo-fence area in the map.

Step

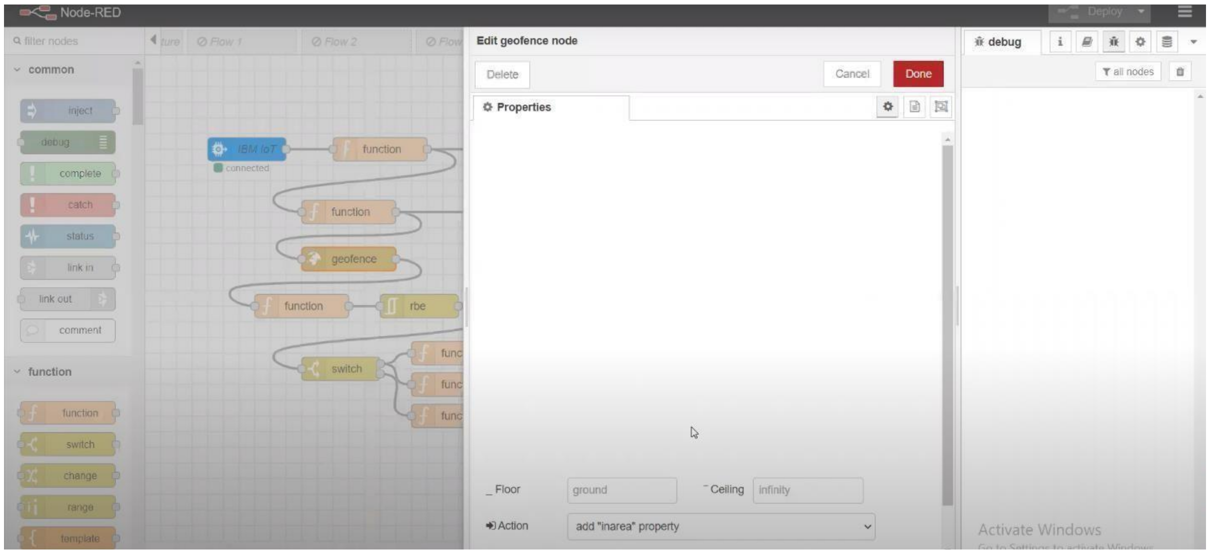


5: Create geo-fence in a particular area.



Floor: ground
Ceiling: infinity
Action: add "inarea" property

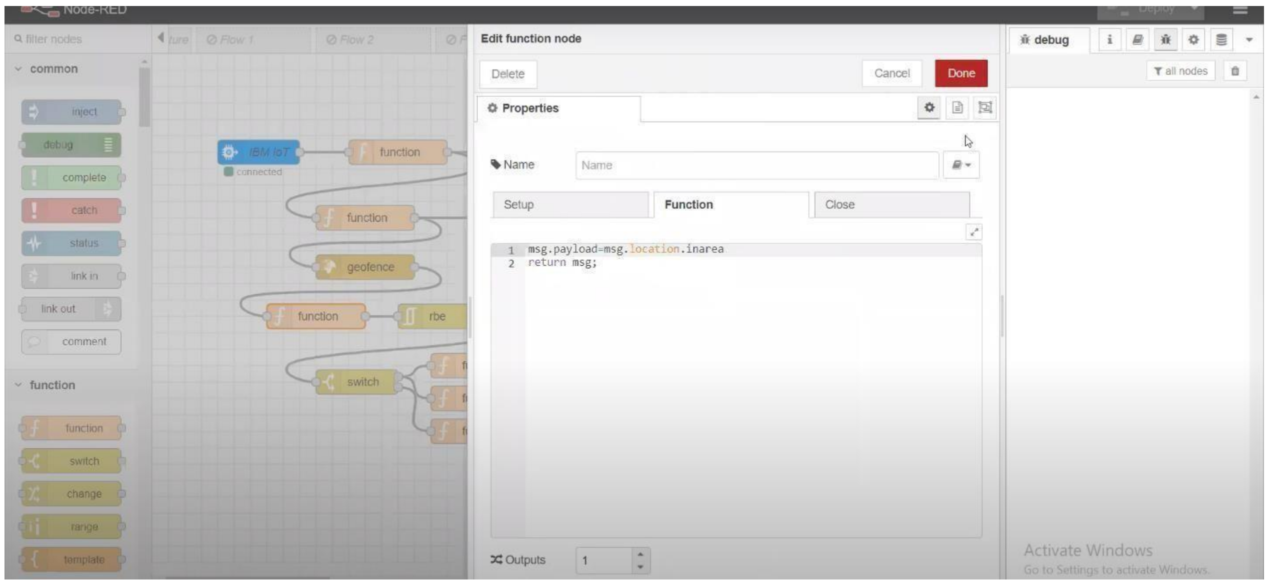
Step 6: Select the function block.



Floor ground Ceiling infinity

Action add "inarea" property

Step 7: Select the message payload.



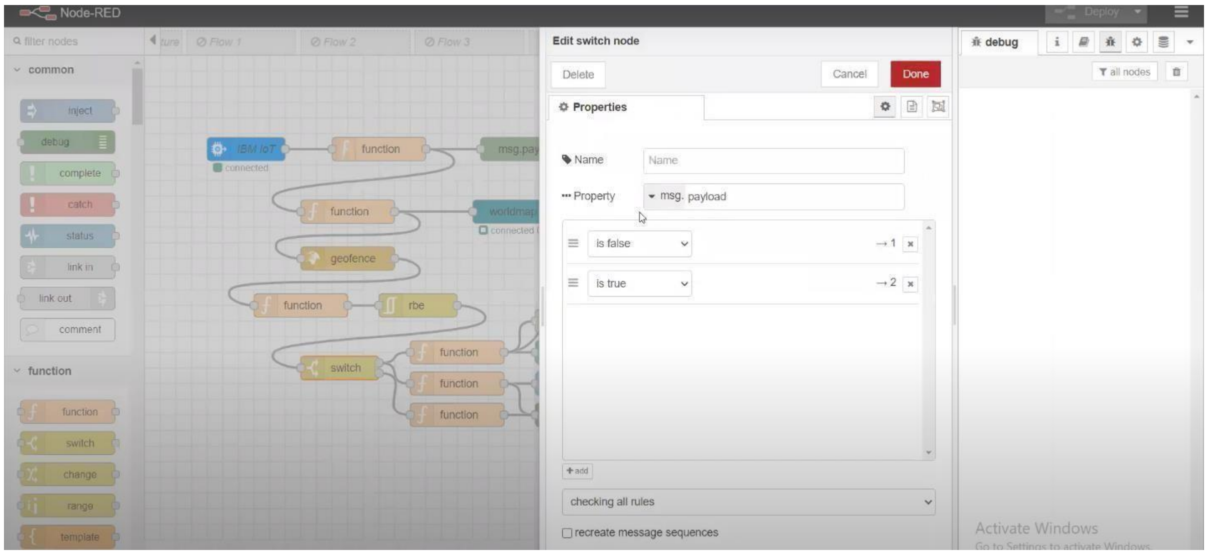
Name Name

```
msg.payload=msg.location.inarea
return msg;
```

Outputs 1

Step 8: To identify the person in area.

Step



Edit switch node

Delete Cancel Done

Properties

Name: Name

Property: msg. payload

is false → 1

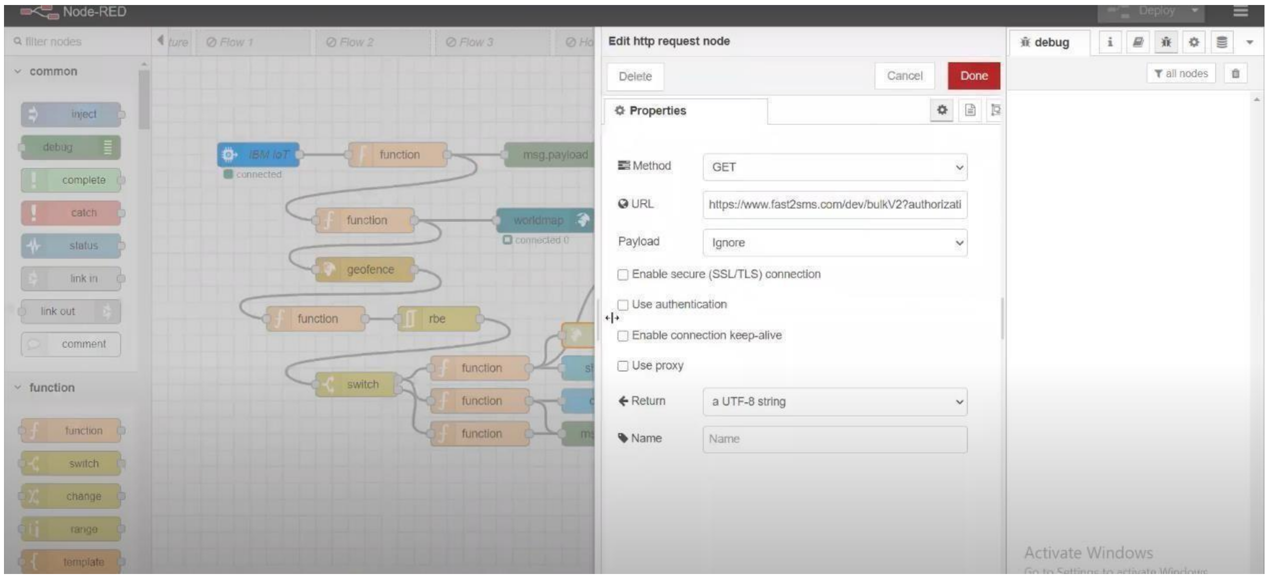
is true → 2

+add

checking all rules

recreate message sequences

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



Edit http request node

Delete Cancel Done

Properties

Method: GET

URL: https://www.fast2sms.com/dev/bulkV2?authorizati

Payload: Ignore

Enable secure (SSL/TLS) connection

Use authentication

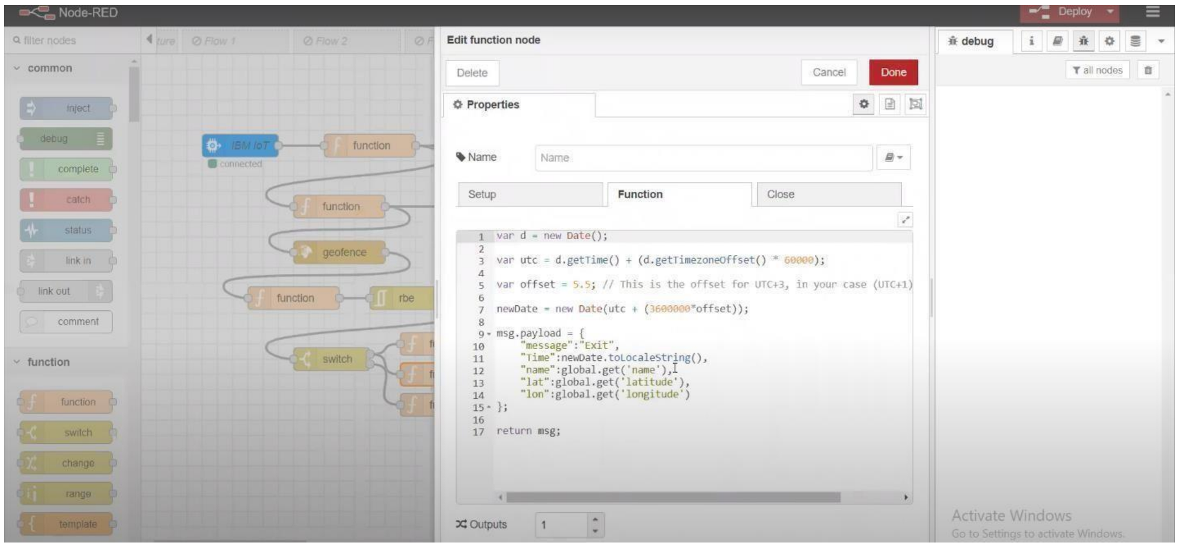
Enable connection keep-alive

Use proxy

Return: a UTF-8 string

Name: Name

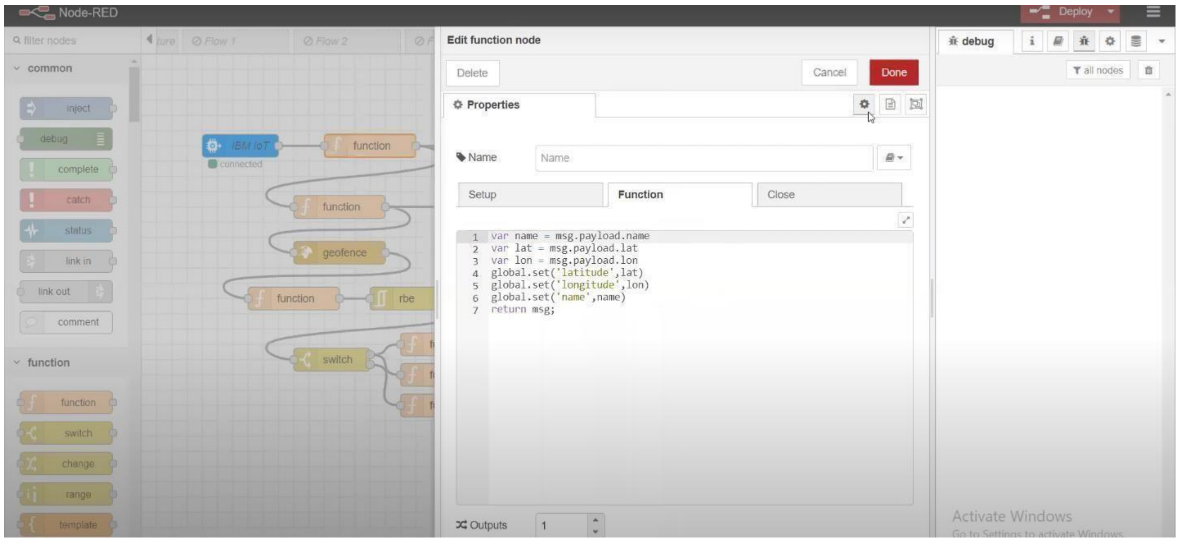
Step 10: For sending the message with time.



```javascript
var d = new Date();

var utc = d.getTime() + (d.getTimezoneOffset() * 60000);

var offset = 5.5; // This is the offset for UTC+3, in your case (UTC+1)

newDate = new Date(utc + (3600000*offset));

msg.payload = {
    "message":"Exit",
    "Time":newDate.toLocaleString(),
    "name":global.get('name'),
    "lat":global.get('latitude'),
    "lon":global.get('longitude')
};

return msg;
```

Step

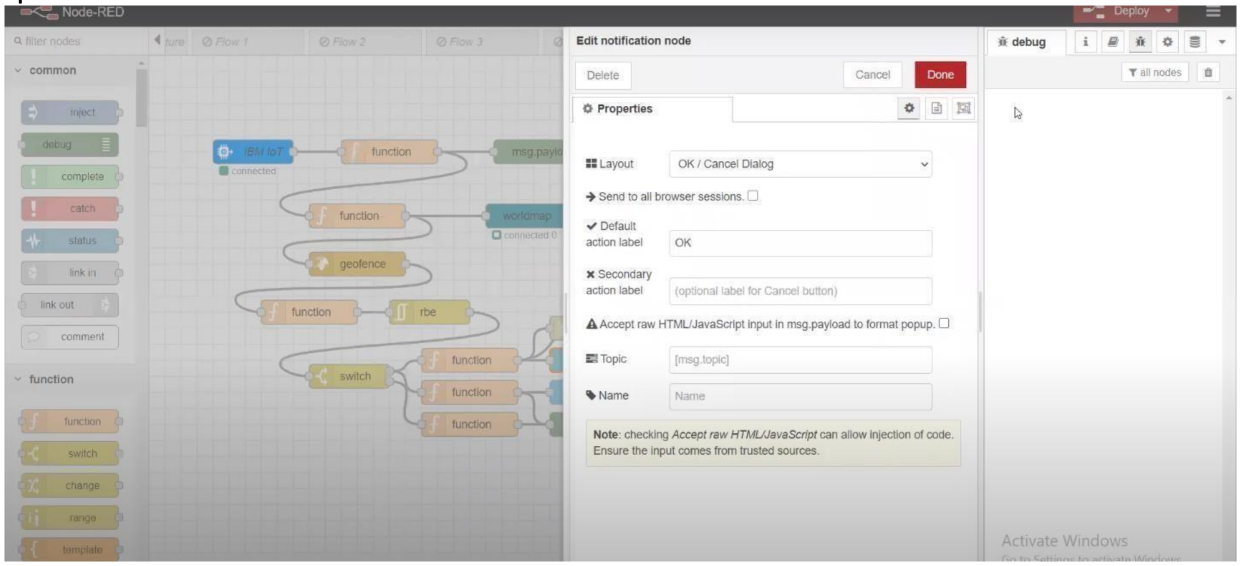
10: Click show dialog for notifying the popup alert.



```
var name = msg.payload.name
var lat = msg.payload.lat
var lon = msg.payload.lon
global.set('latitude',lat)
global.set('longitude',lon)
global.set('name',name)
return msg;
```

Outputs 1

Step



Edit notification node

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

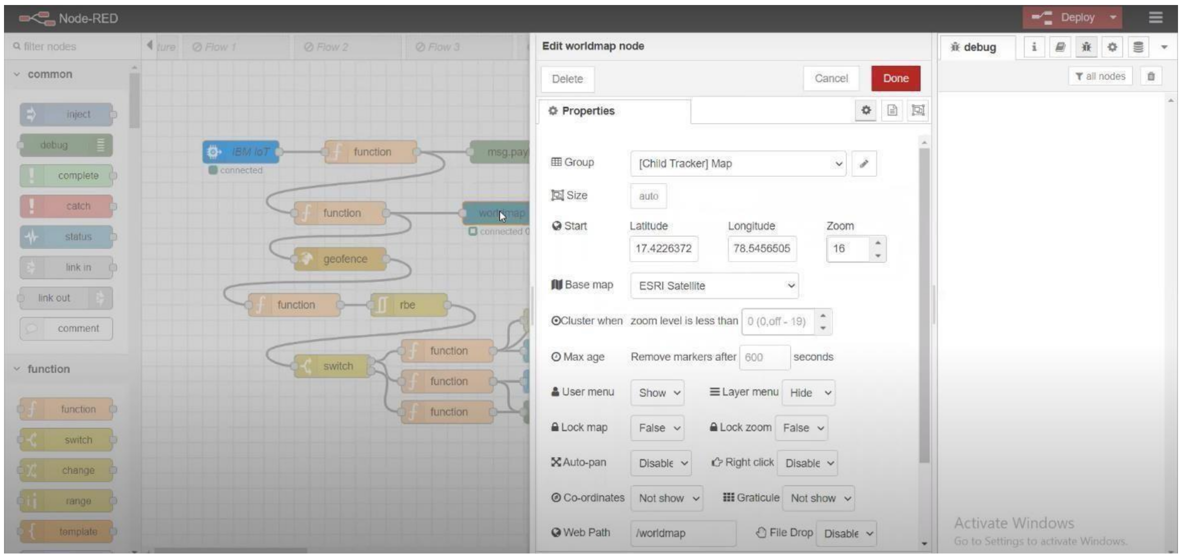
Note: checking *Accept raw HTML/JavaScript* can allow injection of code. Ensure the input comes from trusted sources.

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.



Edit worldmap node

Properties

Group: [Child Tracker] Map

Size: auto

Start: Latitude 17.4226372 Longitude 78.5456505 Zoom 16

Base map: ESRI Satellite

Cluster when zoom level is less than 0 (0,off - 19)

Max age: Remove markers after 600 seconds

User menu: Show Layer menu: Hide

Lock map: False Lock zoom: False

Auto-pan: Disable Right click: Disable

Co-ordinates: Not show Graticule: Not show

Web Path: /worldmap File Drop: Disable