

Develop A Web Application Using Node-RED Service

Python code for transferring latitude and longitude:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

organization="mkgfko"
deviceType="raspberrypi"
deviceId="12345"
authMethod="token"
authToken="12345678"

try:
    deviceOptions={"org": organization,"type": deviceType,"id": deviceId,"auth-
method": authMethod,"auth-token": authToken}

    deviceCli=ibmiotf.device.Client(deviceOptions)
except Exception as e:
    print("caught exception connecting device:%s" % str(e))
    sys.exit()
    deviceCli.connect()

while True:
    #in data
    name="kowshik"
    #latitude=11.229592;
    #longitude= 78.171158;
    #out data
```

```

latitude=12.7345;

longitude=13.2020;

data={'lat':latitude,'lon':longitude,'name':name}

def myOnPublishCallback():

    print("published lattitude=%d" %latitude,"longitude=%d" %longitude,"to
ibm watson")

success=deviceCli.publishEvent("lotSensor","json",data,qos=0,on_publish=myOnP
ublishCallback)

if not success:

    print("Not connected to IoT")

    time.sleep(3)

deviceCli.disconnect()

```

OUTPUT:

```

location data.py - C:\Users\ELCOT\AppData\Local\Programs\Python\Python37\location data.py (3.7.0)
File Edit Format Run Options Window Help

import time
import sys
import ibmiotf.application

location data.py *Python 3.7.0 Shell
File Edit Format Shell Debug Options Window Help

Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\ELCOT\AppData\Local\Programs\Python\Python37\location data.py

organization: 2022-11-08 19:43:18,686 ibmiotf.device.Client INFO Connected successfully
deviceType: d:mkgfk:raspberrypi:12345
deviceId="published lattitude=11 longitude=78 to ibm watson"
authMethod="published lattitude=11 longitude=78 to ibm watson"
authToken="published lattitude=11 longitude=78 to ibm watson"

try:
    deviceCli.publishEvent("lotSensor","json",data,qos=0,on_publish=myOnPublishCallback)
except Exception as e:
    print(e)
    sys.exit(1)

deviceCli.disconnect()
while True:
    time.sleep(3)

```

PUBLISHED DATA IN IBM WATSON IOT PLATFORM:

Service Details - IBM Cloud x IBM Watson IoT Platform x Node-RED x +

mkgfko.internetofthings.ibmcloud.com/dashboard/devices/browse

IBM Watson IoT Platform kowshikp333777@gmail.com ID: mkgfko

Browse Action Device Types Interfaces Add Device +

Device ID	Status	Device Type	Class ID	Date Added	
12345	Connected	raspberrypi	Device	24 Oct 2022 22:40	→ ...

Identity Device Information Recent Events State Logs X

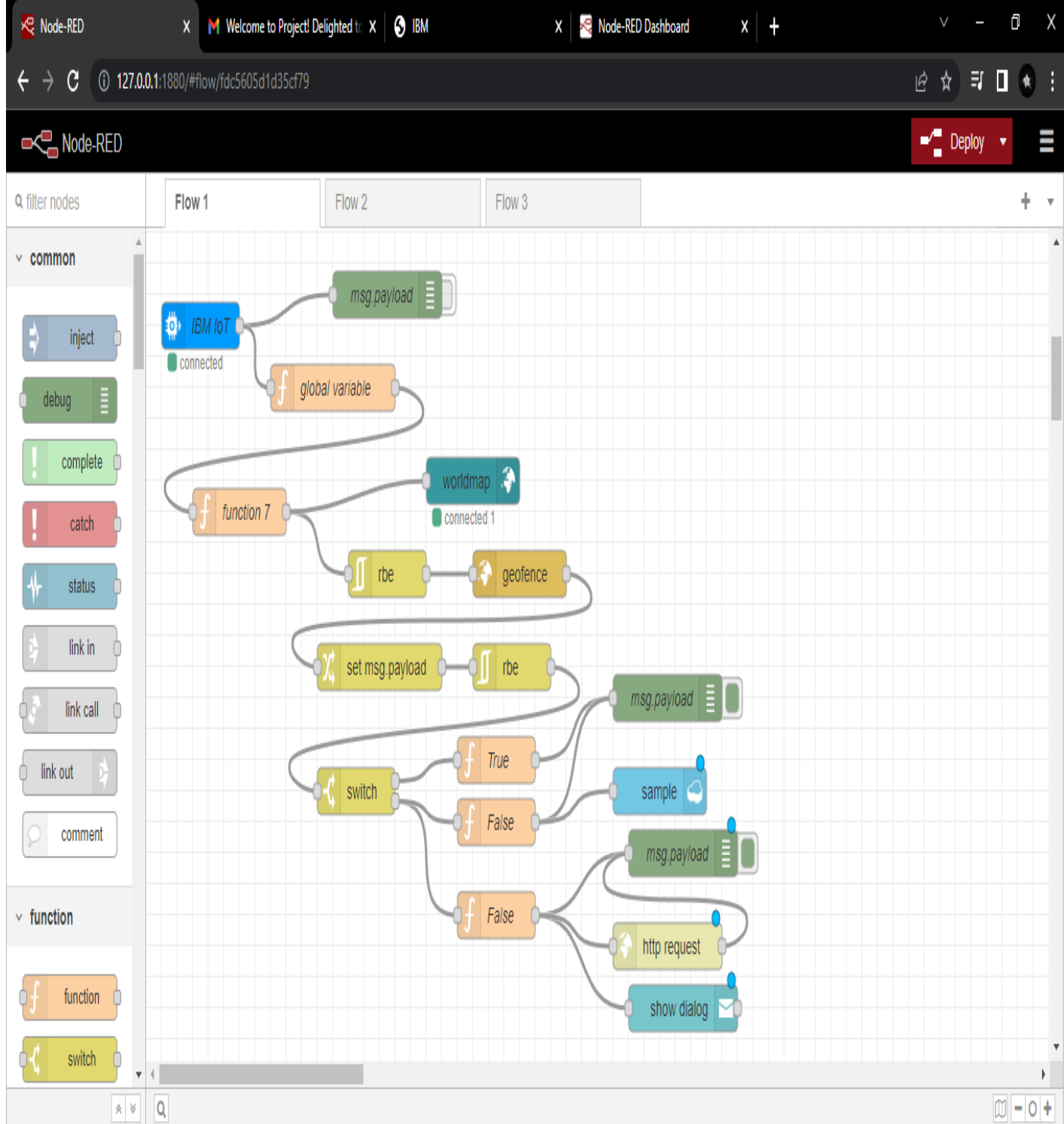
The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
IotSensor	{"lat":12.7345,"lon":13.202,"name":"kowshik"}	json	a few seconds ago
IotSensor	{"lat":12.7345,"lon":13.202,"name":"kowshik"}	json	a few seconds ago
IotSensor	{"lat":12.7345,"lon":13.202,"name":"kowshik"}	json	a few seconds ago
IotSensor	{"lat":12.7345,"lon":13.202,"name":"kowshik"}	json	a few seconds ago
IotSensor	{"lat":12.7345,"lon":13.202,"name":"kowshik"}	json	a few seconds ago

0 Simulations running

Type here to search 30°C 10:44 AM 11/9/2022

WEB APPLICATION USING NODE-RED:



WHEN WE GIVE IN AREA LOCATION:

```
import time
import sys
import ibmiotf.application
```

Node-RED interface showing a flow diagram and a debug console.

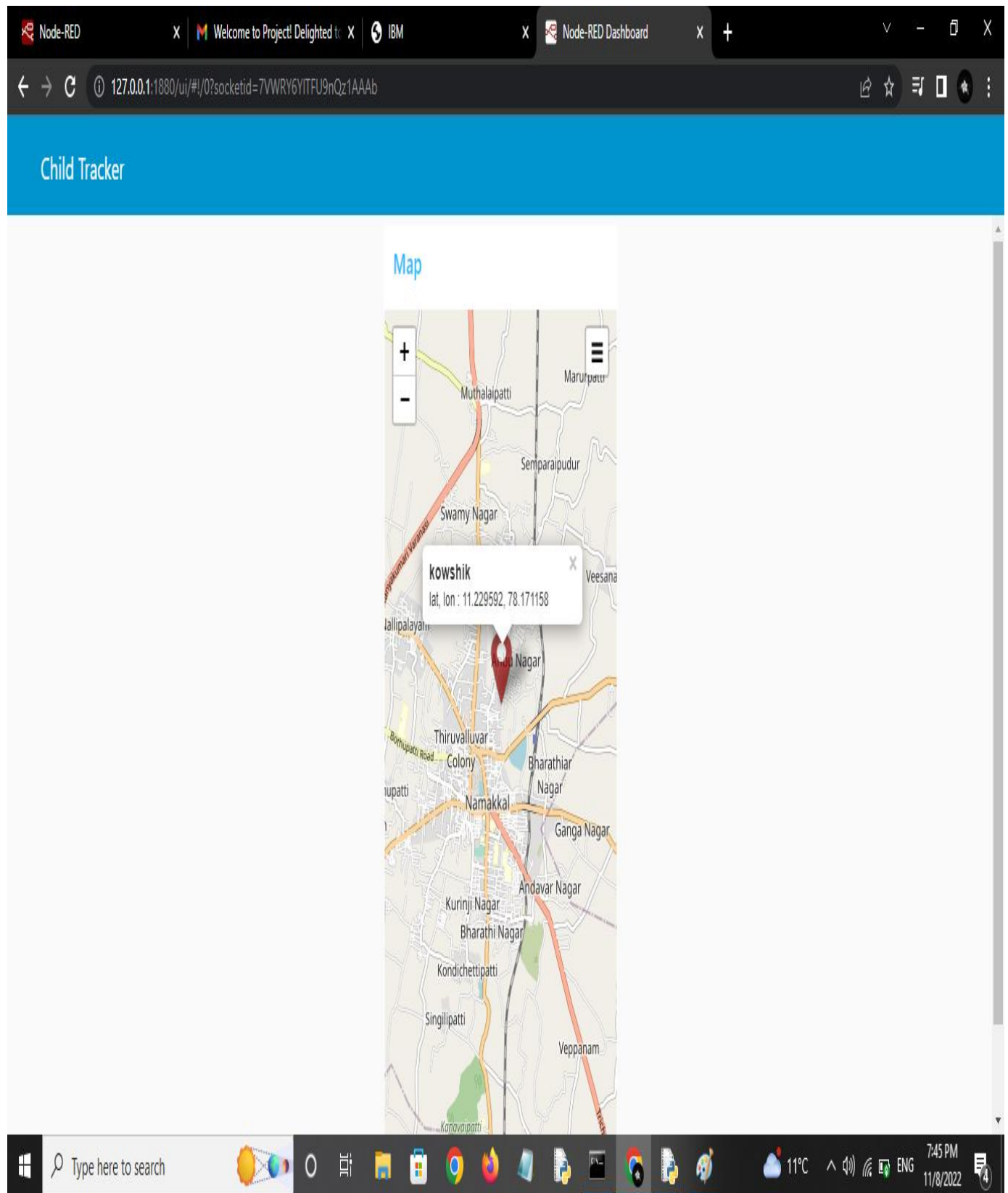
Flow Diagram (Flow 1):

- IBM IoT** (connected) node connects to a **msg.payload** node.
- The **msg.payload** node connects to a **global variable** node.
- The **global variable** node connects to a **function 7** node.
- The **function 7** node connects to a **worldmap** (connected 1) node.
- The **function 7** node connects to an **rbe** node.
- The **rbe** node connects to a **geofence** node.
- The **geofence** node connects to a **set msg.payload** node.
- The **set msg.payload** node connects to an **rbe** node.
- The **rbe** node connects to a **switch** node.
- The **switch** node has three outputs:
 - True**: connects to a **sample** node.
 - False**: connects to a **msg.payload** node.
 - False**: connects to an **http request** node.
- The **sample** node connects to a **msg.payload** node.
- The **http request** node connects to a **show dialog** node.

Debug Console:

11/8/2022, 7:44:02 PM node: msg.payload
iot-2/type/raspberrypi/id/12345/evt/lotSensor/fmt/json : msg.payload :
Object
» { message: "Entry", Date and Time: "11/9/2022, 9:14:01 AM", name: "kowshik", lat: 11.229592, lon: 78.171158 }

OUTPUT:



WHEN WE GIVE OUT AREA LOCATION:

The screenshot displays the Node-RED web interface in a browser. The address bar shows the URL `127.0.0.1:1880/#flow/fdc5605d1d35cf79`. The interface includes a left sidebar with node categories (common, function), a central workspace with a flow diagram, and a right sidebar with a debug console.

Flow Diagram:

- The flow starts with an **IBM IoT** node (connected).
- It connects to a **msg.payload** node.
- Next is a **global variable** node.
- Then a **function 7** node.
- The flow then splits into two paths:
 - Path 1:** **function 7** → **worldmap** (connected 1) → **rbe** → **geofence**.
 - Path 2:** **function 7** → **set msg.payload** → **rbe**.
- Both paths converge into a **switch** node.
- The **switch** node has three outputs:
 - True:** **True** node → **sample** node.
 - False (top):** **False** node → **msg.payload** node.
 - False (bottom):** **False** node → **http request** node → **show dialog** node.

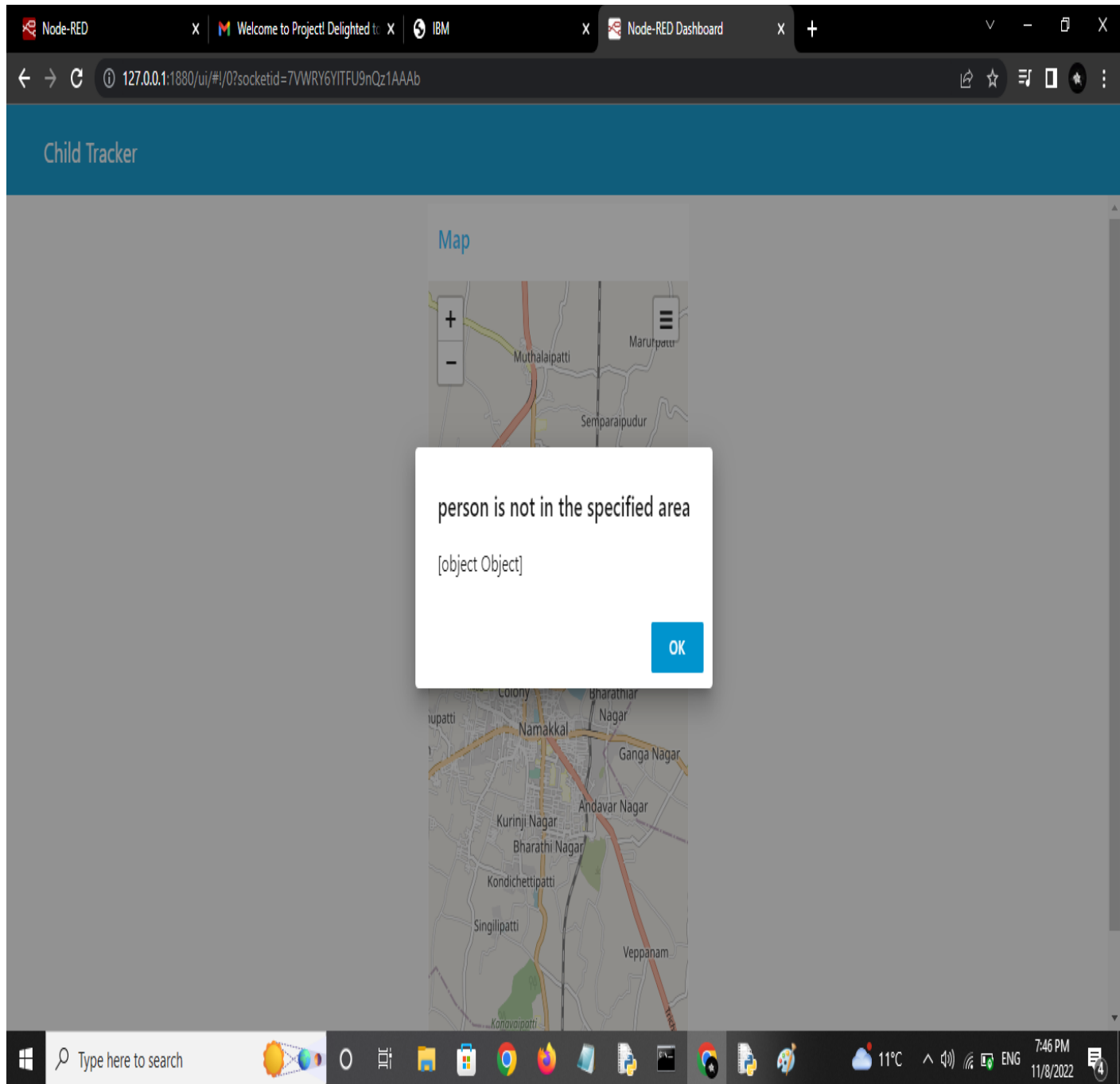
Debug Console:

The debug console shows three messages from the **node: msg.payload** node:

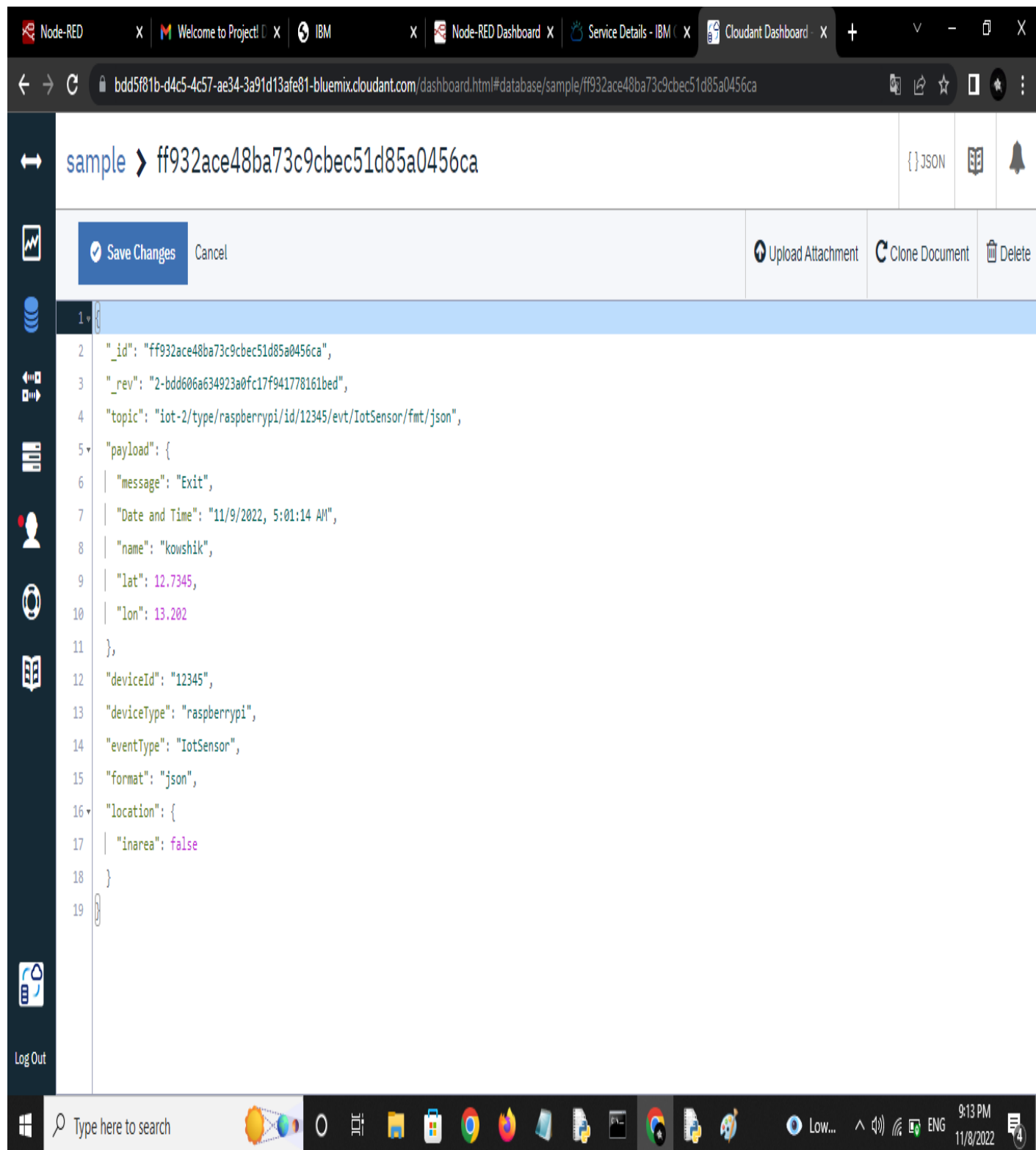
- `{ message: "Person is not in the specified.." }`
- `{ message: "Exit", Date and Time: "11/9/2022, 9:18:23 AM", name: "kowshik", lat: 12.7345, lon: 13.202 }`
- `string[111]`

The bottom of the image shows a Windows taskbar with the search bar and various application icons.

OUTPUT:



STORED DATA IN THE DATABASE WE HAVE CREATED:



The screenshot displays the IBM Cloudant Dashboard interface. The browser's address bar shows the URL: `bdd5f81b-d4c5-4c57-ae34-3a91d13afe81-bluemix.cloudant.com/dashboard.html#database/sample/ff932ace48ba73c9cbec51d85a0456ca`. The page title is "sample" followed by the document ID "ff932ace48ba73c9cbec51d85a0456ca". The document is displayed in JSON format, with a toggle for "JSON" visible. The document content is as follows:

```
{
  "_id": "ff932ace48ba73c9cbec51d85a0456ca",
  "_rev": "2-bdd606a634923a0fc17f941778161bed",
  "topic": "iot-2/type/raspberrypi/id/12345/evt/IotSensor/fmt/json",
  "payload": {
    "message": "Exit",
    "Date and Time": "11/9/2022, 5:01:14 AM",
    "name": "kowshik",
    "lat": 12.7345,
    "lon": 13.202
  },
  "deviceId": "12345",
  "deviceType": "raspberrypi",
  "eventType": "IotSensor",
  "format": "json",
  "location": {
    "inarea": false
  }
}
```

The interface includes a "Save Changes" button, an "Upload Attachment" button, a "Clone Document" button, and a "Delete" button. The document is displayed in a code editor with line numbers 1 through 19. The bottom of the screen shows the Windows taskbar with the search bar and various application icons.