

PROJECT DEVELOPMENT PHASE
SPRINT 2

TEAM ID	PNT2022TMID46063
PROJECT NAME	IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE
DATE	31 OCTOBER 2022

STEP 1: Write a python code for randomize Soil Moisture ,Temperature and Humidity.

```
sensor.py - C:/Users/murug/Desktop/sensor.py (3.7.0)
File Edit Format Run Options Window Help

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

# Provide your IBM Watson Device Credentials
organization = "8guz7t" # replace the ORG ID
deviceType = "weather_monitor" # replace the Device type
deviceId = "b827ebd607b5" # replace Device ID
authMethod = "token"
authToken = "LWVpQPqEVL66HWN48Z" # Replace the authToken

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    print(cmd)

try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-token": authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)
    #.....

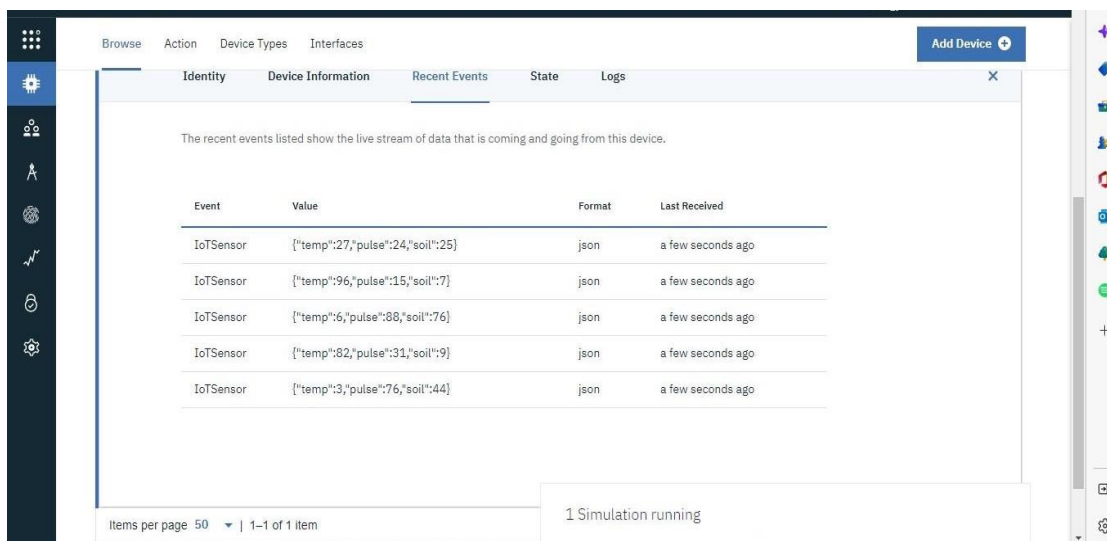
except Exception as e:
    print("Caught exception connecting device: %s" % str(e))
    sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times
deviceCli.connect()

while True:
    temp=random.randint(0,100)
    pulse=random.randint(0,100)
    soil=random.randint(0,100)

    data = { 'temp': temp, 'pulse': pulse, 'soil':soil}
    #print data
    def myOnPublishCallback():
```

STEP 2: Run the python code it send data to IBM IoT Watson Platform.



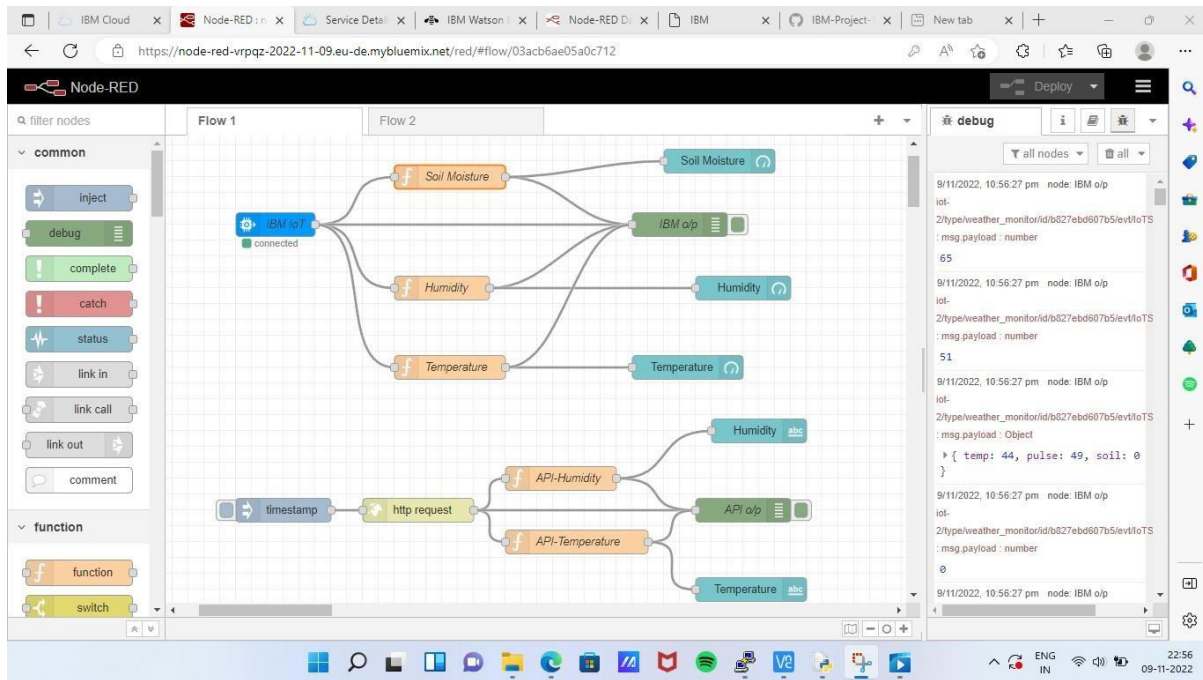
The screenshot shows the IBM IoT Platform dashboard with the 'Recent Events' tab selected. It displays a table of live data streams from an IoT sensor. The table has four columns: Event, Value, Format, and Last Received. There are five rows of data, each representing a sensor reading. The status at the bottom indicates '1 Simulation running'.

Event	Value	Format	Last Received
IoTSensor	["temp":27,"pulse":24,"soil":25]	json	a few seconds ago
IoTSensor	["temp":96,"pulse":15,"soil":7]	json	a few seconds ago
IoTSensor	["temp":6,"pulse":88,"soil":76]	json	a few seconds ago
IoTSensor	["temp":82,"pulse":31,"soil":9]	json	a few seconds ago
IoTSensor	["temp":3,"pulse":76,"soil":44]	json	a few seconds ago

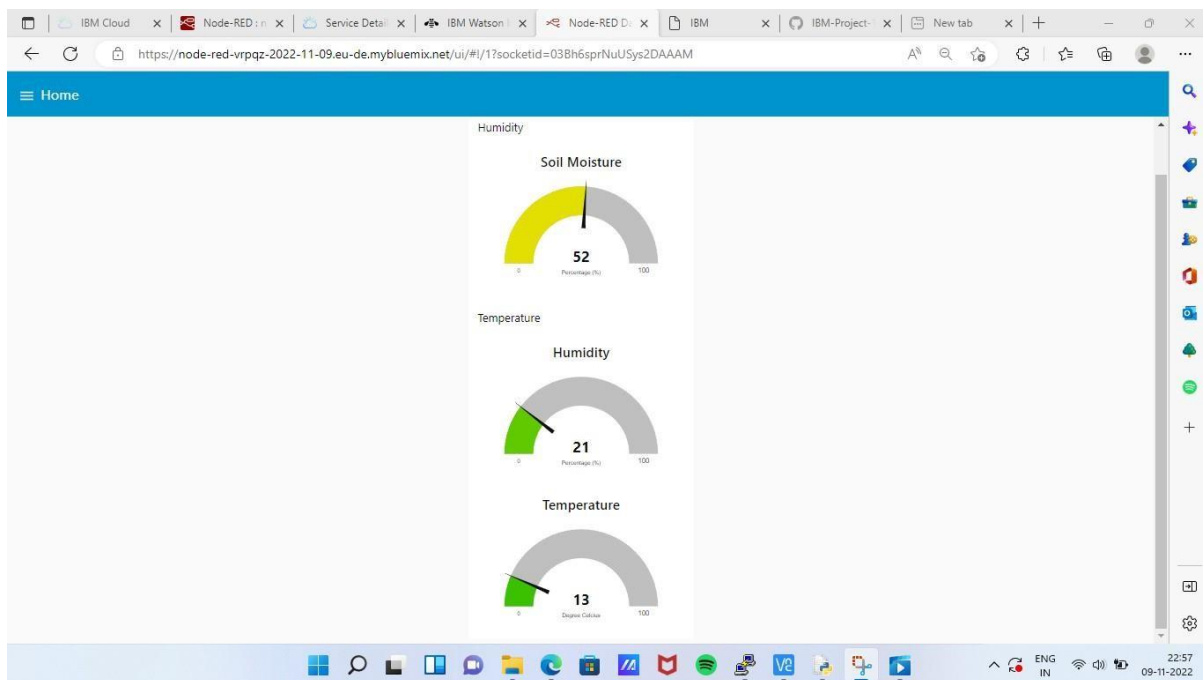
Items per page 50 | 1-1 of 1 item

1 Simulation running

STEP 3: Open Node-RED flow dashboard.



STEP 4: Open Node-RED user interface to show the Soil Moisture, Humidity and Temperature value in gauge.



PYTHON CODE :

```

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

# Provide your IBM Watson Device Credentials organization =
"8gyz7t" # replace the ORG ID deviceType = "weather_monitor" #
replace the Device type deviceId = "b827ebd607b5" # replace
Device ID authMethod = "token" authToken =
"LWVpQPpVQ166HWN48f" # Replace the authtoken def
myCommandCallback(cmd):

    print("Command received: %s" % cmd.data['command'])
    print(cmd)


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()


# Connect and send a datapoint "hello" with value "world" into the cloud as
an event of type "greeting" 10 times deviceCli.connect()

```

```

while True:
    temp=random.randint(0,100)
    pulse=random.randint(0,100)
    soil=random.randint(0,100)

data = { 'temp' : temp, 'pulse': pulse , 'soil':soil }
    #print data def
    myOnPublishCallback():
    print ("Published
    Temperature = %s C" %
    temp, "Humidity = %s
    %% "
    % pulse,"Soil Moisture = %s %% " % soil,"to IBM Watson")

    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoTTF")
        time.sleep(1)

    deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the
cloud deviceCli.disconnect()

```

Node-RED :

```

[{"id":"b42b5519fee73ee2","type":"ibmiot
in","z":"03acb6ae05a0c712","authentication":"apiKey","apiKey":"ef745d48e3

```

```
5ccc0","inputType":"evt","logicalInterface":"","ruleId":"","deviceId":"b827ebd607b5","applicationId":"","deviceType":"weather_monitor","eventType":"+","commandType":"","format":"json","name":"IBM IoT","service":"registered","allDevices":"","allApplications":"","allDeviceType":s":"","allLogicalInterfaces":"","allEvents":true,"allCommands":"","allFormats":,"","qos":0,"x":270,"y":180,"wires":[["50b13e02170d73fc","d7da6c2f5302ffaf","a949797028158f3f","a71f164bc378bcf1"]]},{"id":"50b13e02170d73fc","type":"function","z":"03acb6ae05a0c712","name":"Soil Moisture","func":"msg.payload = msg.payload.soil;\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[,"x":490,"y":120,"wires":[["a949797028158f3f","ba98e701f55f04fe"]]},{"id":"d7da6c2f5302ffa","type":"function","z":"03acb6ae05a0c712","name":"Humidity","func":"msg.payload = msg.payload.pulse;\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[,"x":480,"y":260,"wires":[["a949797028158f3f","70a5b076eeb80b70"]]},{"id":"a949797028158f3f","type":"debug","z":"03acb6ae05a0c712","name":"IBM o/p","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":false,"payload","targetType":"msg","statusVal":"","statusType":"auto","x":780,"y":180,"wires":[]},{"id":"70a5b076eeb80b70","type":"ui_gauge","z":"03acb6ae05a0c712","name":"","group":"f4cb8513b95c98a4","order":6,"width":0,"height":0,"gtype":"gage","title":"Humidity","label":"Percentage (%)","format":"{{ value }}","min":0,"max":100,"colors":["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":"","className":"","x":860,"y":260,"wires":[]},{"id":"b9832c19b922be3e","type":"http request","z":"03acb6ae05a0c712","name":"","method":"GET","ret":"obj","payloads":"ignore","url":"http://api.openweathermap.org/data/2.5/weather?q=Chinchwad,%20IN&appid=6aa2b89eb478ce7baebf384e671bfd15","tls":"","persist":false,"proxy":"","authType":"","senderr":false,"x":450,"y":540,"wires":[["f7c149
```

a
3169164e8","c2e6d49c5aa44698","6d207fb212acdac3"]]}, {"id": "d55b317d0ec9acfc", "type": "inject", "z": "03acb6ae05a0c712", "name": "", "props": [{"p": "payload"}, {"p": "topic", "vt": "str"}], "repeat": "", "crontab": "", "once": false, "onceDelay": 0.1, "topic": "", "payload": "", "payloadType": "date", "x": 280, "y": 540, "wires": [{"b9832c19b922be3e"}]}, {"id": "6d207fb212acdac3", "type": "debug", "z": "03acb6ae05a0c712", "name": "API o/p", "active": true, "tosidebar": true, "console": false, "tostatus": false, "complete": "payload", "targetType": "msg", "statusVal": "", "statusType": "auto", "x": 860, "y": 540, "wires": []}, {"id": "f7c149a3169164e8", "type": "function", "z": "03acb6ae05a0c712", "name": "API-Humidity", "func": "msg.payload=msg.payload.main.pulse;\nreturn msg;", "outputs": 1, "noerr": 0, "initialize": "", "finalize": "", "libs": [], "x": 630, "y": 500, "wires": [{"6d207fb212acdac3", "23e82e5991b96c8d"}]}, {"id": "c2e6d49c5aa44698", "type": "function", "z": "03acb6ae05a0c712", "name": "API-Temperature", "func": "msg.payload=msg.payload.main.temp;\nreturn msg;", "outputs": 1, "noerr": 0, "initialize": "", "finalize": "", "libs": [], "x": 650, "y": 580, "wires": [{"6d207fb212acdac3", "3e9b68204bef0552"}]}, {"id": "a71f164bc378bcf1", "type": "function", "z": "03acb6ae05a0c712", "name": "Temperature", "func": "msg.payload=msg.payload.temp;\nreturn msg;", "outputs": 1, "noerr": 0, "initialize": "", "finalize": "", "libs": [], "x": 490, "y": 360, "wires": [{"8e8b63b110c5ec2d", "a949797028158f3f"}]}, {"id": "8e8b63b110c5ec2d", "type": "ui_gauge", "z": "03acb6ae05a0c712", "name": "", "group": "f4cb8513b95c98a4", "order": 11, "width": "0", "height": "0", "gtype": "gage", "title": "Temperature", "label": "Degree Celcius", "format": "{{ value }}", "min": 0, "max": "100", "colors": ["#00b500", "#e6600", "#ca3838"], "seg1": "", "seg2": "", "className": "", "x": 790, "y": 360, "wires": []

```
{,"id":"3e9b68204bef0552","type":"ui_text","z":"03acb6ae05a0c712","group":
":
"f4cb8513b95c98a4","order":2,"width":"0","height":"0","name":"","label":"Te
mperature","format":"{{msg.payload}}","layout":"row-
spread","className":"","x":870,"y":640,"wires":[]},{ "id":"23e82e5991b96c8d
",
,"type":"ui_text","z":"03acb6ae05a0c712","group":"f4cb8513b95c98a4","order
":1,"width":"0","height":"0","name":"","label":"Humidity","format":"{{msg.pa
yload}}","layout":"row-
spread","className":"","x":880,"y":440,"wires":[]},{ "id":"ba98e701f55f04fe",
"type":"ui_gauge","z":"03acb6ae05a0c712","name":"","group":"f4cb8513b95c
9
8a4","order":1,"width":"0","height":"0","gtype":"gage","title":"Soil
Moisture","label":"Percentage
(%)","format":"{{value}}","min":0,"max":"100","colors":["#00b500","#e6e60
0
","#ca3838"],"seg1":"","seg2":"","className":"","x":830,"y":100,"wires":[]},{
"
id":"ef745d48e395ccc0","type":"ibmiot","name":"weather_monitor","keepaliv
e
":"60","serverName":"","cleansession":true,"appId":"","shared":false},{ "id":"f
4
cb8513b95c98a4","type":"ui_group","name":"monitor","tab":"1f4cb829.2fdee
8
","order":2,"disp":true,"width":"6","collapse":false,"className":""},{ "id":"1f4
c
b829.2fdee8","type":"ui_tab","name":"Home","icon":"dashboard","order":3,"d
i
sabled":false,"hidden":false}]}
```