

PROJECT DEVELOPMENT PHASE

SPRINT 2

TEAM ID	PNT2022TMID15739
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 = "sgys7e" # replace the ORG ID
deviceType = "weather_monitor" # replace the Device type
deviceId = "b827ebd607b5" # replace Device ID
authMethod = "token"
authToken = "LMVpQPvQ166HWN48Z" # 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.

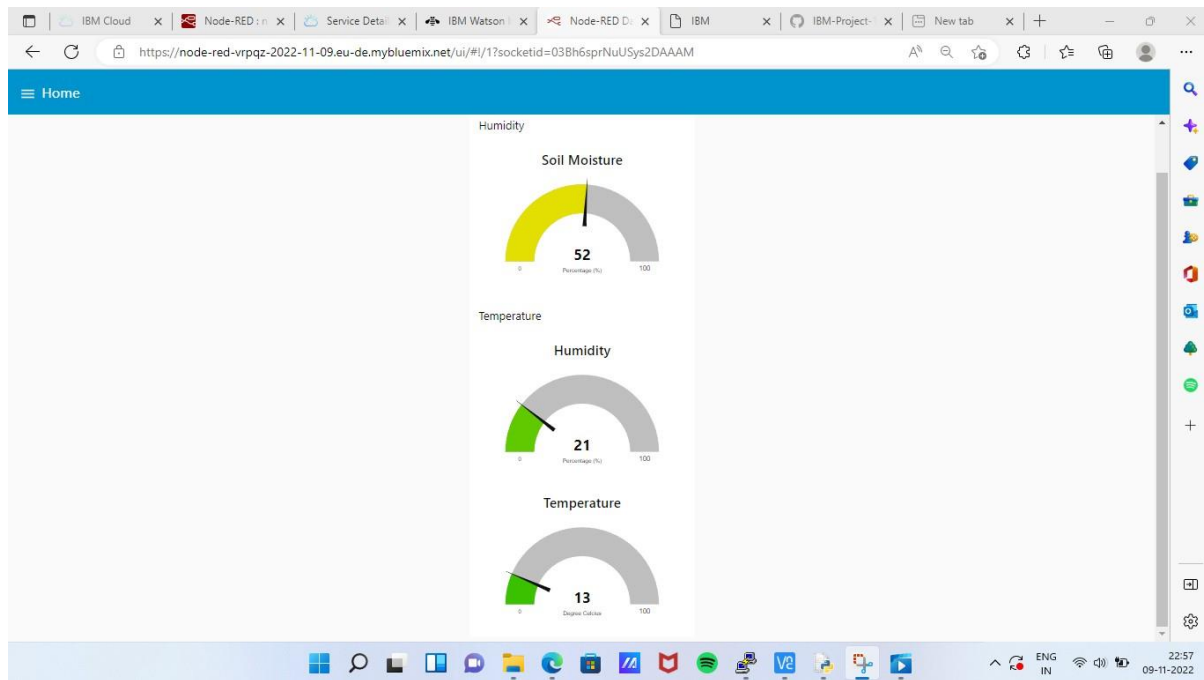
The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. The main content area is titled 'Recent Events' and displays a table of live data streams. The table has four columns: 'Event', 'Value', 'Format', and 'Last Received'. Below the table, it 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

STEP 3: Open Node-RED flow dashboard.

The screenshot shows the Node-RED flow dashboard. The main workspace displays a flow with several nodes. The flow starts with an 'IBM IoT' node, which connects to three function nodes labeled 'Soil Moisture', 'Humidity', and 'Temperature'. These function nodes then connect to corresponding output nodes labeled 'Soil Moisture', 'Humidity', and 'Temperature'. Below this, there is a 'timestamp' node connected to an 'http request' node, which then connects to two function nodes labeled 'API-Humidity' and 'API-Temperature'. These function nodes connect to output nodes labeled 'Humidity' and 'Temperature'. The right sidebar shows the 'debug' console with a log of messages received from the IBM IoT node, including timestamps and payloads.

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 = "LWVpQPaVQ166HWN48f" # 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 IoT")
```

```
    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":"ef745d48e39  
5ccc0","inputType":"evt","logicalInterface":"","ruleId":"","deviceId":"b827ebd  
607b5","applicationId":"","deviceType":"weather_monitor","eventType":"+","c  
ommandType":"","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  
f","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":"a949797028158  
f3f","type":"debug","z":"03acb6ae05a0c712","name":"IBM
```

o/p","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"p
ayload","targetType":"msg","statusVal":"","statusType":"auto","x":780,"y":180
,"wires":[]},{ "id":"70a5b076eeb80b70","type":"ui_gauge","z":"03acb6ae05a0c
712","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","payt
oqs":"ignore","url":"http://api.openweathermap.org/data/2.5/weather?q=Chinch
wad,%20IN&appid=6aa2b89eb478ce7baebf384e671bfd15","tls":"","persist":fal
se,"proxy":"","authType":"","senderr":false,"x":450,"y":540,"wires":[["f7c149a
3169164e8","c2e6d49c5aa44698","6d207fb212acdac3"]]}, {"id":"d55b317d0ec
9acfc","type":"inject","z":"03acb6ae05a0c712","name":"","props":[{"p":"paylo
ad"}, {"p":"topic","vt":"str"}],"repeat":"","crontab":"","once":false,"onceDelay":
0.1,"topic":"","payload":"","payloadType":"date","x":280,"y":540,"wires":[["b9
832c19b922be3e"]]}, {"id":"6d207fb212acdac3","type":"debug","z":"03acb6ae0
5a0c712","name":"API
o/p","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"p
ayload","targetType":"msg","statusVal":"","statusType":"auto","x":860,"y":540
,"wires":[]},{ "id":"f7c149a3169164e8","type":"function","z":"03acb6ae05a0c7
12","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":"c2e6d49c5aa44
698","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":"a71f164bc378bc
f1","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":"8e8b63b110c5e
c2d","type":"ui_gauge","z":"03acb6ae05a0c712","name":"","group":"f4cb8513
b95c98a4","order":11,"width":0,"height":0,"gtype":"gage","title":"Tempera
ture","label":"Degree
Celcius","format":"{{ value }}","min":0,"max":100,"colors":["#00b500","#e6e
600","#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":"f4cb8513b95c9
8a4","order":1,"width":"0","height":"0","gtype":"gage","title":"Soil
Moisture","label":"Percentage
(%)","format":{{value}}","min":0,"max":100,"colors":["#00b500","#e6e600
","#ca3838"],"seg1":"","seg2":"","className":"","x":830,"y":100,"wires":[]},{ "
id":"ef745d48e395ccc0","type":"ibmiot","name":"weather_monitor","keepalive
":"60","serverName":"","cleansession":true,"appId":"","shared":false},{ "id":"f4
cb8513b95c98a4","type":"ui_group","name":"monitor","tab":"1f4cb829.2fdee8
","order":2,"disp":true,"width":"6","collapse":false,"className":""},{ "id":"1f4c
b829.2fdee8","type":"ui_tab","name":"Home","icon":"dashboard","order":3,"di
sabled":false,"hidden":false}]
```