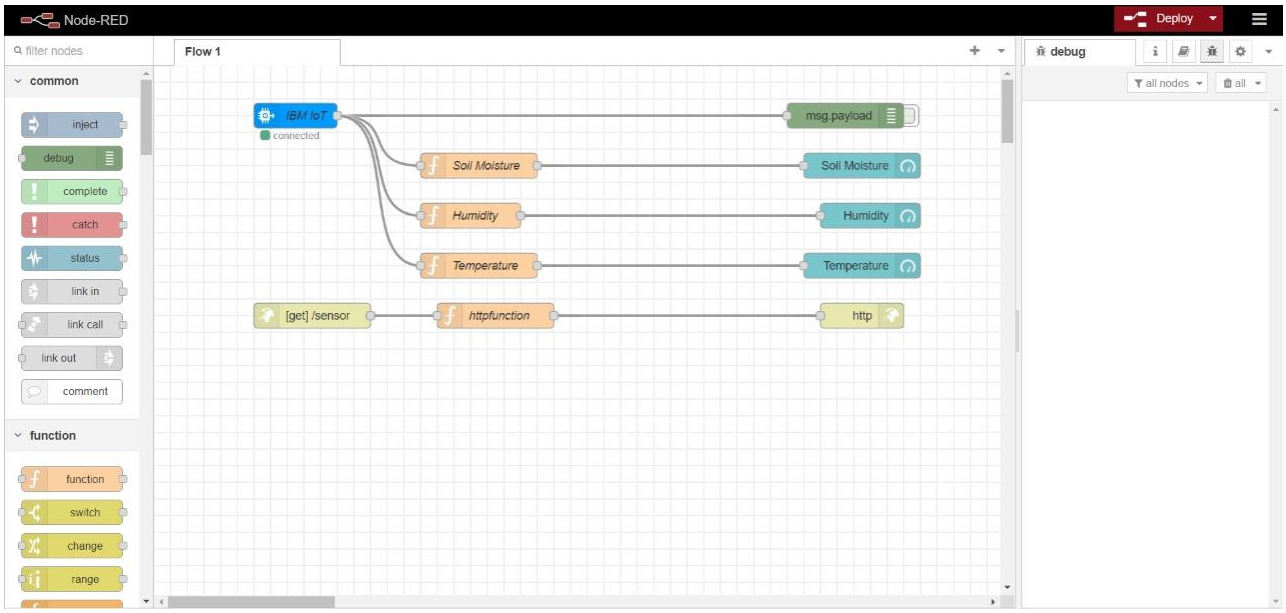
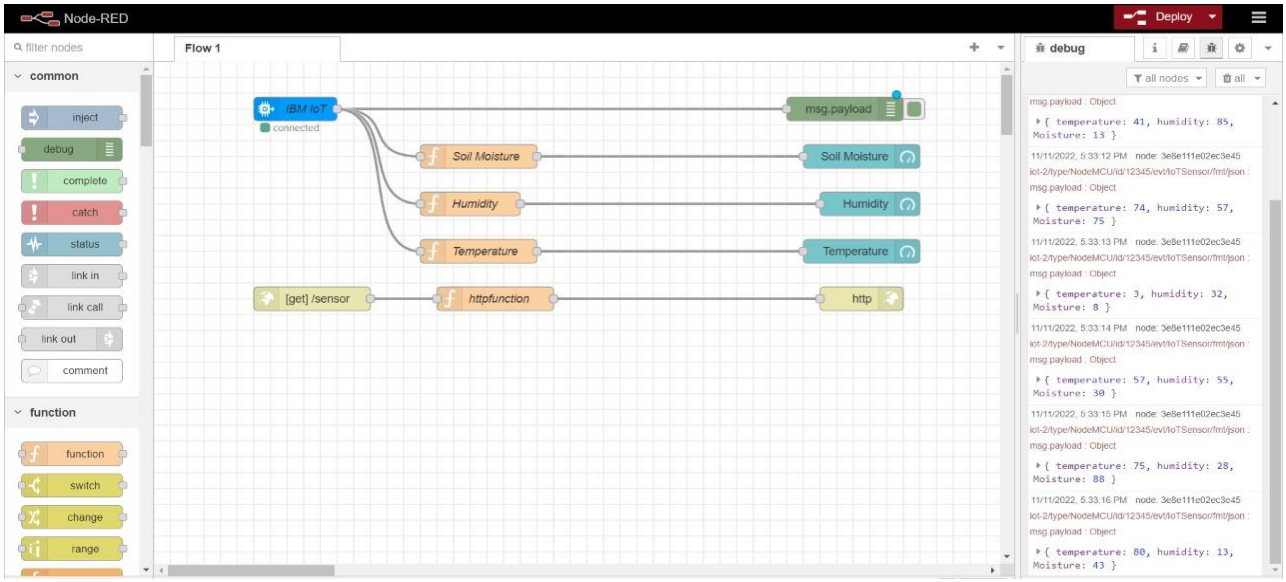
SPRINT 2

Date	5 November 2022
Team ID	PNT2022TMID53609
Project Name	Smart Farmer – IOT Enabled Smart Farming Application
Maximum Marks	8 Marks

Creation of Node Red Service for device events:

In the IBM Watson IOT platform, under the catalog, under the Node Red app service, an application is deployed using cloud foundry. In the cloud foundry, a group has been created and using the ci pipeline, the app url is obtained. Using the url, the Node red is launched. The IBM Watson IOT platform is connected to Node red using the ibm iot palette. Using appropriate palettes, the data published in the IBM IOT platform is printed in the debug window of Node red.





Code block for the function palette:

1) **Soil moisture:**

```
Soil = msg.payload.Moisture
msg.payload = "Soil Moisture : "
global.set('m',Soil)
msg.payload = Math.round(Soil)
return msg;
```

2) **Humidity:**

```
Humidity = msg.payload.humidity
msg.payload = "Humidity : "
global.set('h',Humidity)
msg.payload = Math.round(Humidity )
return msg;
```

3) **Temperature:**

```
Temperature = msg.payload.temperature
msg.payload = "Temperature : "
global.set('t',Temperature)
msg.payload =Math.round(Temperature)
return msg;
```

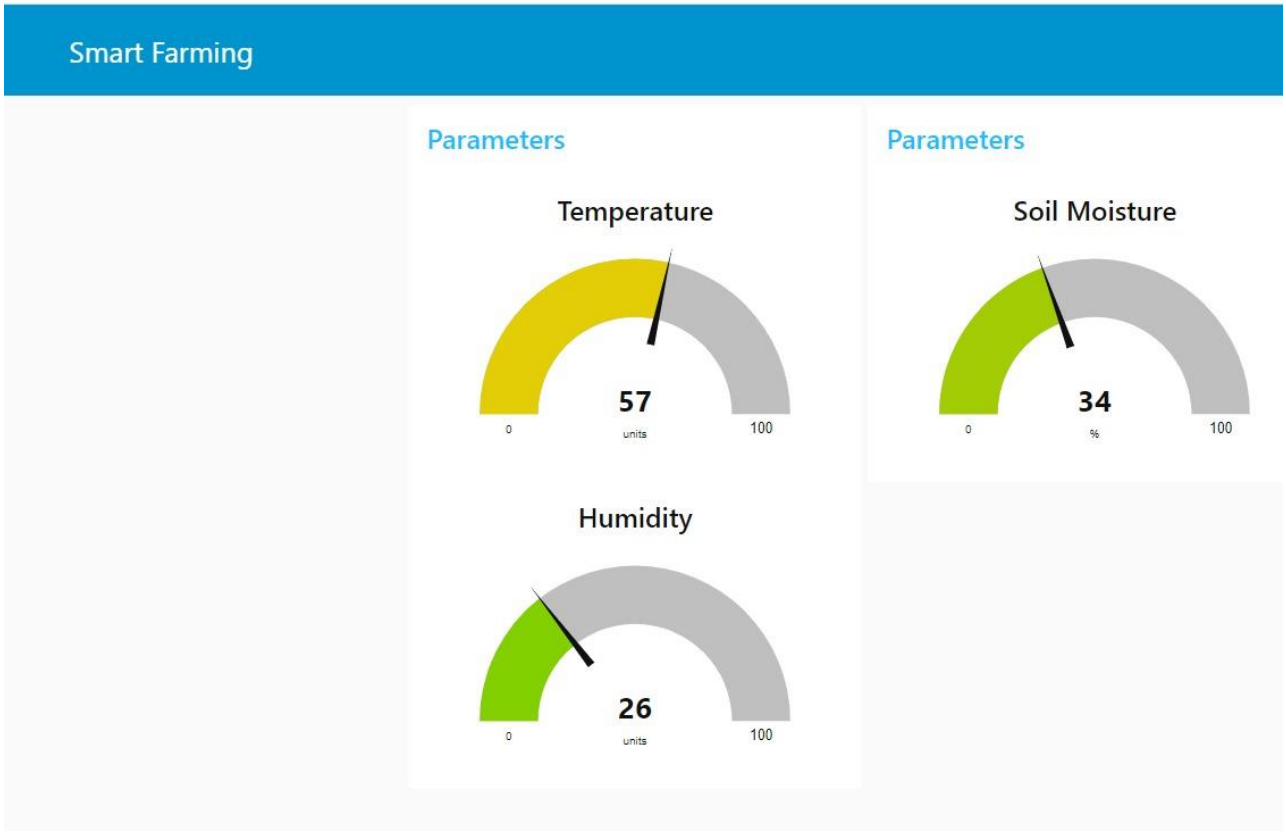
4) HTTP Function:

```
msg.payload = {"Temperature:": global.get('t'),"Humidity:":
global.get('h'),"Soil Moisture:": global.get('m')}

return msg;
```

Creation of website dashboard:

A website dashboard has been created using the gauge palette. It can be accessed by adding **"**/ui**"** in the main url of Node red. This dashboard displays the gauge representation of the data published in the IBM IOT platform.



Python code used:

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


#Provide your IBM Watson Device Credentials
organization = "nckdv7"
deviceType = "NodeMCU"
deviceId = "12345"
authMethod = "token"
authToken = "12345678"
```

```
# Initialize GPIO
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:
        #Get Sensor Data from DHT11
        temp=random.randint(0,100)
        pulse=random.randint(0,100)
        moisture= random.randint(0,100)
        humidity=random.randint(0,100);
        lat =  17
        lon = 18
        data = { 'temperature' : temp, 'humidity' : humidity, 'Moisture' : moisture}
        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % humidity,
"Soil Moisture = %s %%" % moisture,"to IBM Watson")

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

        deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```