| Date | 19.11.2022 |
|---|---|
| Team ID | PNT2022TMID11788 |
| Project title | **SmartFarmer - IoT Enabled Smart Farming Application** |

## Project Tracker

| Sprint | Total Story Points | Duration | Sprint Start Date | Sprint End Date (Planned) | Story Points Completed (as on Planned End Date) | Sprint Release Date (Actual) |
|---|---|---|---|---|---|---|
| Sprint-1 | 15 | 5 Days | 26 Oct 2022 | 30 Oct 2022 | 15 | 30 Oct 2022 |
| Sprint-2 | 15 | 7 Days | 31 Oct 2022 | 06 Nov 2022 | 15 | 07 Nov 2022 |
| Sprint-3 | 15 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 15 | 13 Nov 2022 |
| Sprint-4 | 15 | 6 Days | 13 Nov 2022 | 18 Nov 2022 | | 18 Nov 2022 – 19 Nov 2022 |

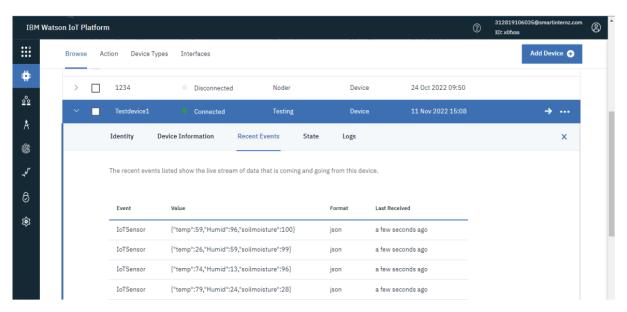| S.NO | Tools & Technology Used |
|---|---|
| 1 | Python 3.7.0 |
| 2 | IBM Cloud |
| 3 | Node-Red |

## Python Script:

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

#Provide your IBM Watson Device Credentials
organization = "x0fxss" #replace the ORG ID
deviceType = "Testing"#replace the Device type wi
deviceId = "Testdevice1"#replace Device ID
authMethod = "token"
authToken = "123456789" #Replace the authtoken
# Initialize GPIO

#Receives Command from Node-red
def myCommandCallback(cmd):
    print ("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="motoron":
        print ("motor is on")
    elif status == "motoroff" :
        print ("motor is off")
    elif status == "motor30" :
        print ("motor is on for 30 minutes")

try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-token": authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)
```
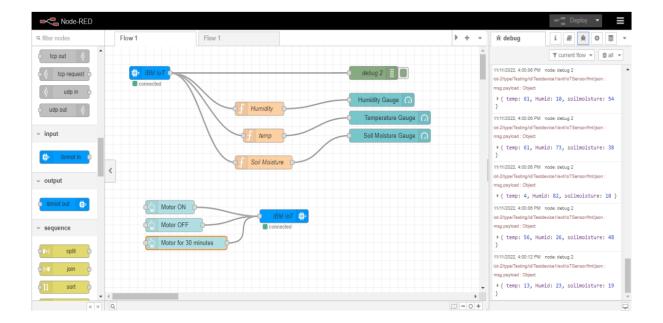
```
except Exception as e:
    print("Caught exception connecting device: %s" % str(e))
    sys.exit()

# 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)
        Humid=random.randint(0,100)
        soilmoisture=random.randint(0,100)

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

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(5)

        deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()
```
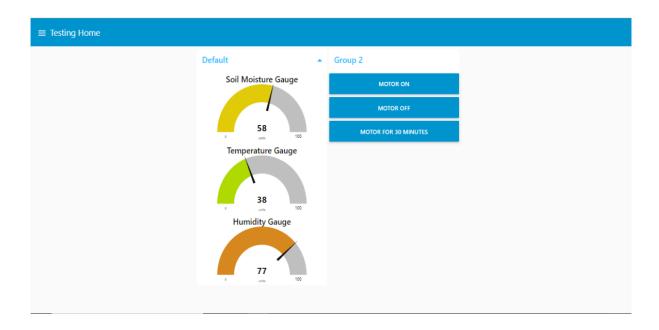
**OUTPUT:**

We are running python script to send data to IBM cloud and data is displayed in web-ui by using node-red.

Data are successfully received and displayed.