- SPRINT 2 REPORT
- SMARTFARMER – IoT ENABLED SMART FARMING APPLICAION

TEAM ID – PNT2022TMID22142

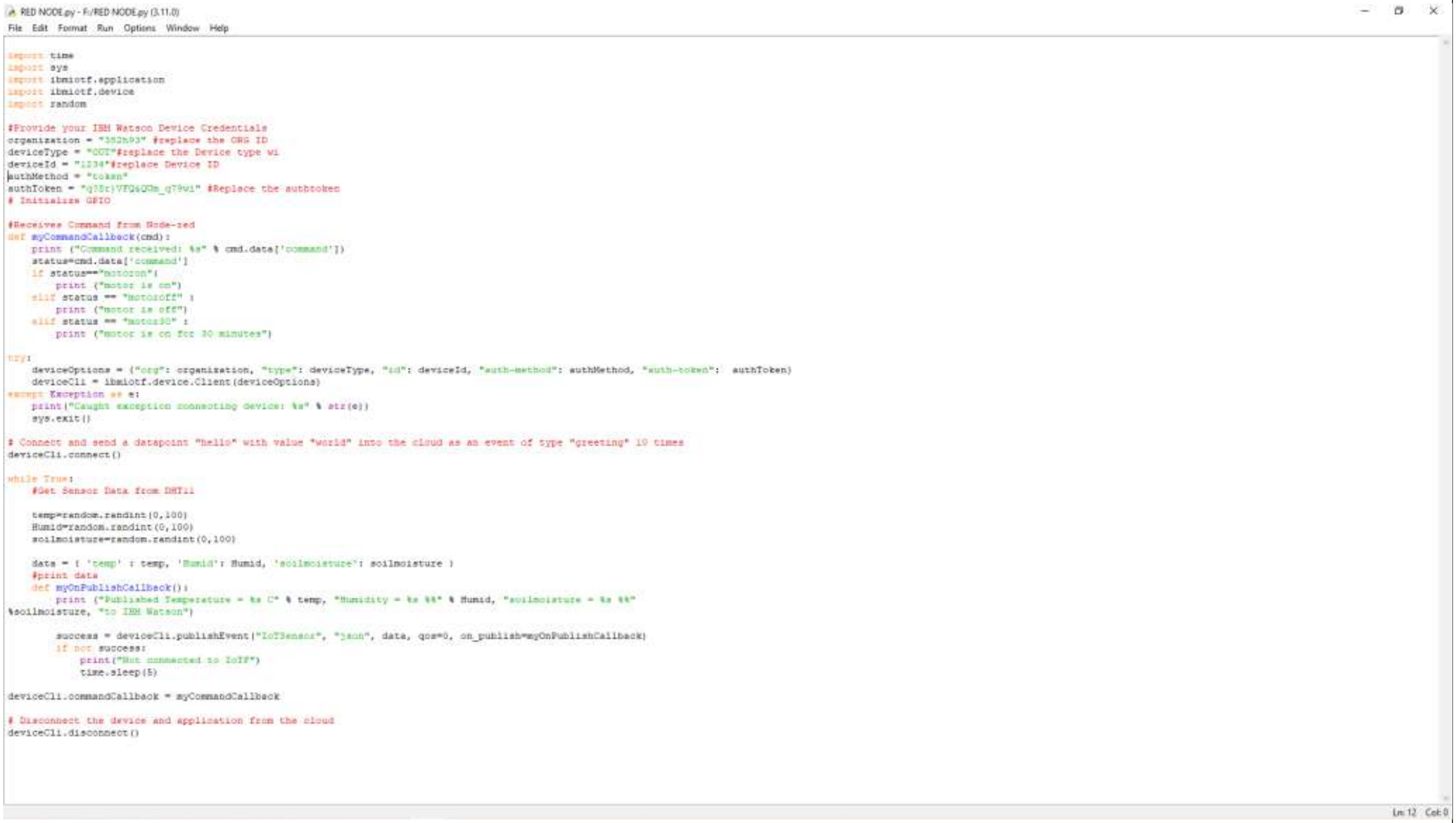
- **TEAM LEADER : NAVEEN V M**
- **TEAM MEMBER : GHAJENDHIRAN J**
- **TEAM MEMBER : GANESA MOORTHI M**
- **TEAM MEMBER : KOUSHIK P H**

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

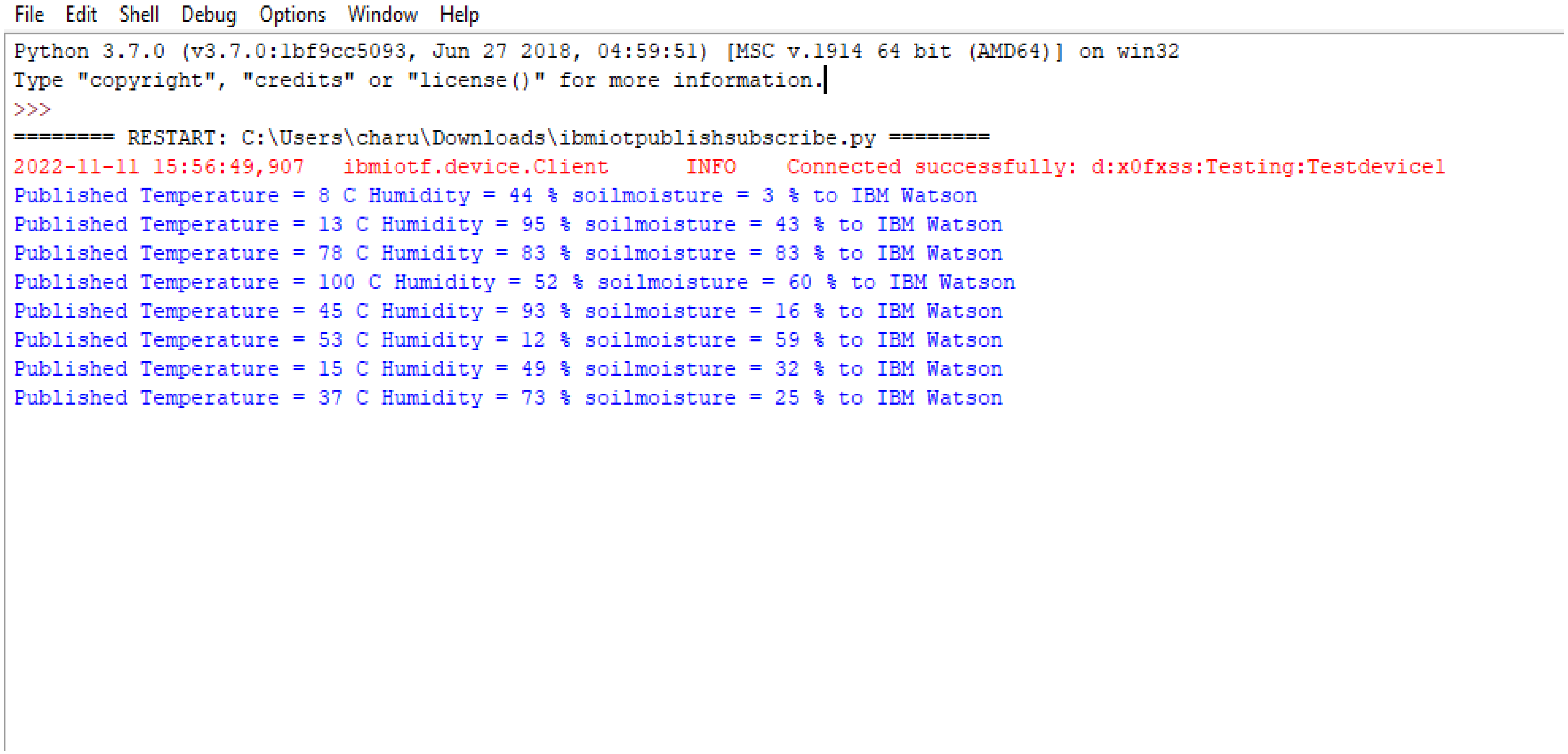
S.NO	Tools & Technology Used
1	Python 3.7.0
2	IBM Cloud
3	Node-Red

Python Script:

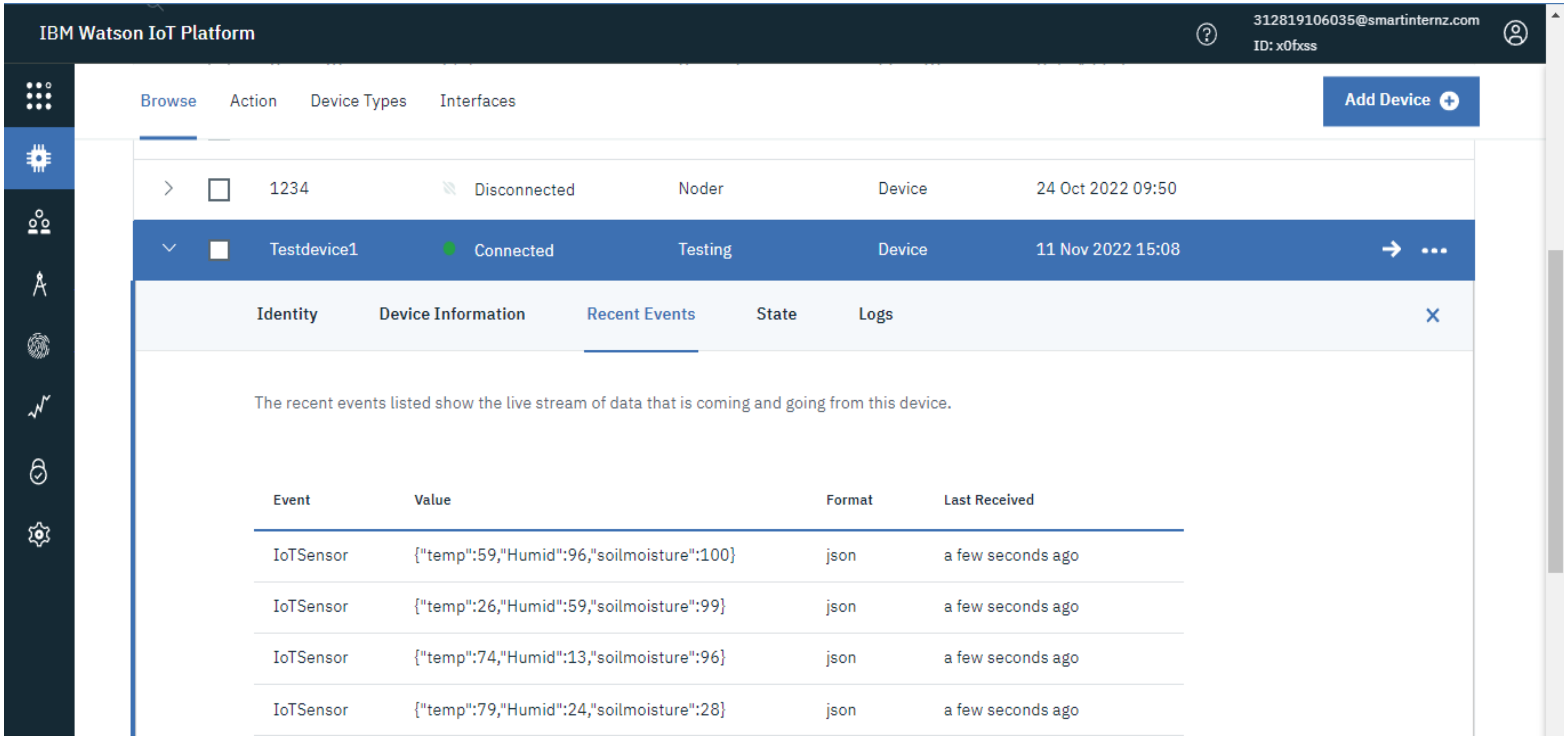

RED NODE.py - F:/RED NODE.py (3.11.0)
File Edit Format Run Options Window Help
Ln: 12 Col: 0

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

#Provide your IBM Watson Device Credentials
organization = "352h93" #replace the ORG ID
deviceType = "OOT"#replace the Device type wi
deviceId = "1234"#replace Device ID
authMethod = "token"
authToken = "q?3r)VFQ&QOm_q79vi" #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:



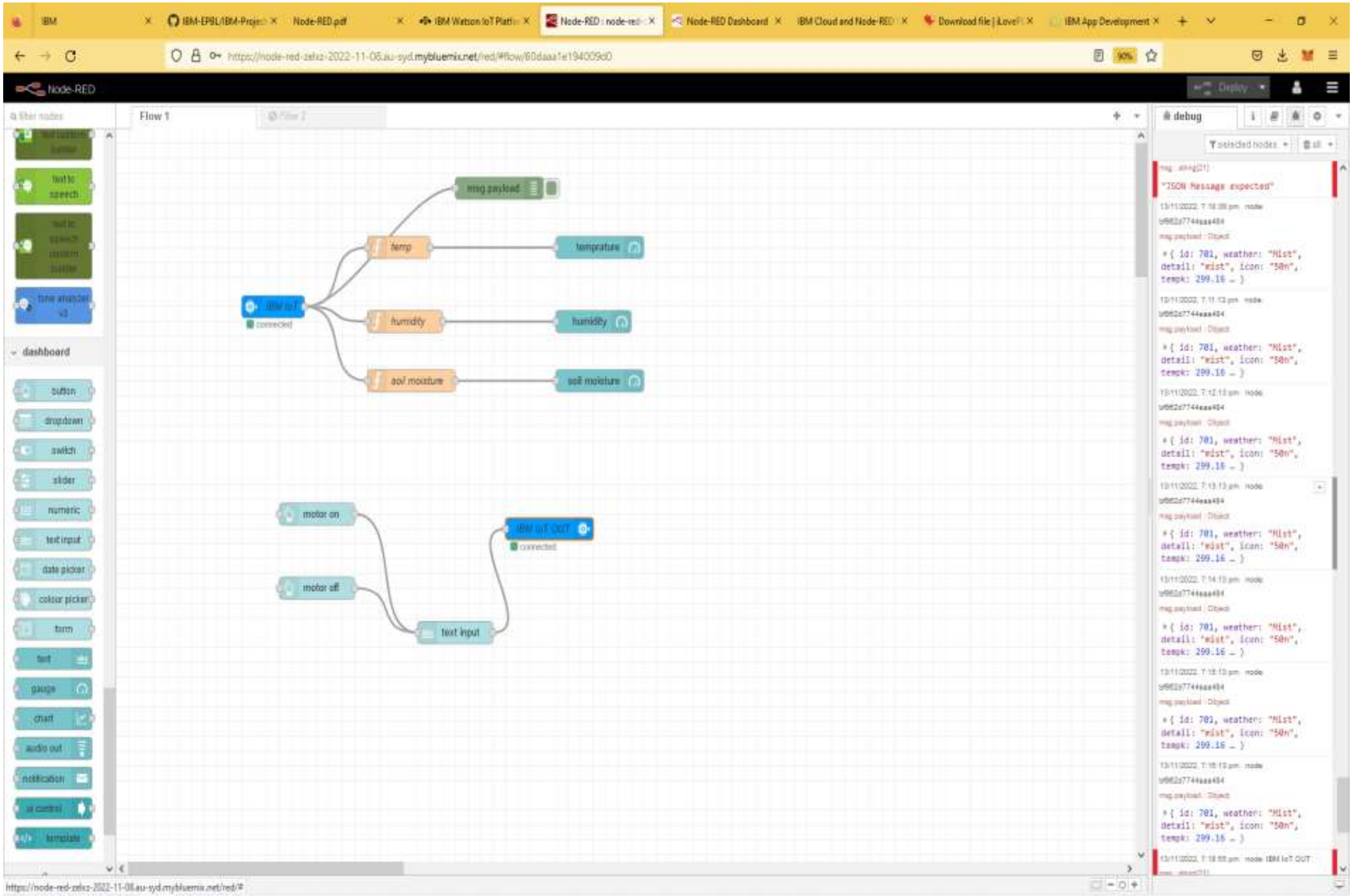
```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:\Users\charu\Downloads\ibmiotpublishsubscribe.py ========
2022-11-11 15:56:49,907   ibmiotf.device.Client      INFO    Connected successfully: d:x0fxss:Testing:Testdevice1
Published Temperature = 8 C Humidity = 44 % soilmoisture = 3 % to IBM Watson
Published Temperature = 13 C Humidity = 95 % soilmoisture = 43 % to IBM Watson
Published Temperature = 78 C Humidity = 83 % soilmoisture = 83 % to IBM Watson
Published Temperature = 100 C Humidity = 52 % soilmoisture = 60 % to IBM Watson
Published Temperature = 45 C Humidity = 93 % soilmoisture = 16 % to IBM Watson
Published Temperature = 53 C Humidity = 12 % soilmoisture = 59 % to IBM Watson
Published Temperature = 15 C Humidity = 49 % soilmoisture = 32 % to IBM Watson
Published Temperature = 37 C Humidity = 73 % soilmoisture = 25 % to IBM Watson
```



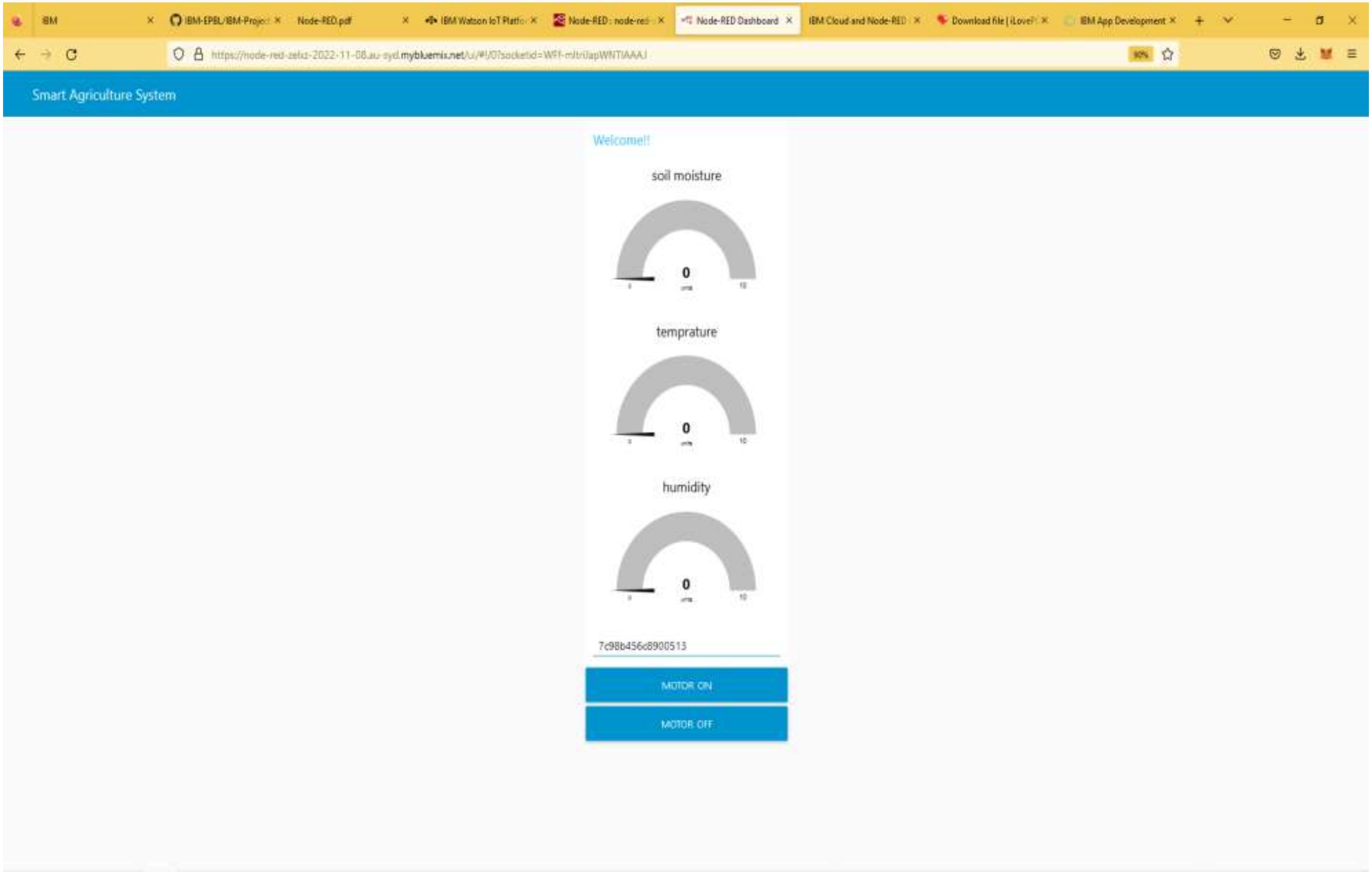
› □	1234	Disconnected	Noder	Device	24 Oct 2022 09:50
⌄ □	Testdevice1	● Connected	Testing	Device	11 Nov 2022 15:08

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

Event	Value	Format	Last Received
IoTSensor	{"temp":59,"Humid":96,"soilmoisture":100}	json	a few seconds ago
IoTSensor	{"temp":26,"Humid":59,"soilmoisture":99}	json	a few seconds ago
IoTSensor	{"temp":74,"Humid":13,"soilmoisture":96}	json	a few seconds ago
IoTSensor	{"temp":79,"Humid":24,"soilmoisture":28}	json	a few seconds ago



```
msg : string[21]
"JSON Message expected"
13/11/2022, 7:10:35 pm  node: b9862d7744eaa484
msg.payload : Object
▸{ id: 701, weather: "Mist", detail: "mist", icon: "50n", tempk: 299.16 … }
13/11/2022, 7:11:13 pm  node: b9862d7744eaa484
msg.payload : Object
▸{ id: 701, weather: "Mist", detail: "mist", icon: "50n", tempk: 299.16 … }
13/11/2022, 7:12:13 pm  node: b9862d7744eaa484
msg.payload : Object
▸{ id: 701, weather: "Mist", detail: "mist", icon: "50n", tempk: 299.16 … }
13/11/2022, 7:13:13 pm  node: b9862d7744eaa484
msg.payload : Object
▸{ id: 701, weather: "Mist", detail: "mist", icon: "50n", tempk: 299.16 … }
13/11/2022, 7:14:13 pm  node: b9862d7744eaa484
msg.payload : Object
▸{ id: 701, weather: "Mist", detail: "mist", icon: "50n", tempk: 299.16 … }
13/11/2022, 7:15:13 pm  node: b9862d7744eaa484
msg.payload : Object
▸{ id: 701, weather: "Mist", detail: "mist", icon: "50n", tempk: 299.16 … }
13/11/2022, 7:16:13 pm  node: b9862d7744eaa484
msg.payload : Object
▸{ id: 701, weather: "Mist", detail: "mist", icon: "50n", tempk: 299.16 … }
13/11/2022, 7:18:55 pm  node: IBM IoT OUT
msg : string[21]
```



Smart Agriculture System

Welcome!!

soil moisture

0

temprature

0

humidity

0

7c98b456c8900513