Smart Farmer - IoT Enabled Smart Farming Application

SPRINT- III

Team ID: PNT2022TMID50618

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

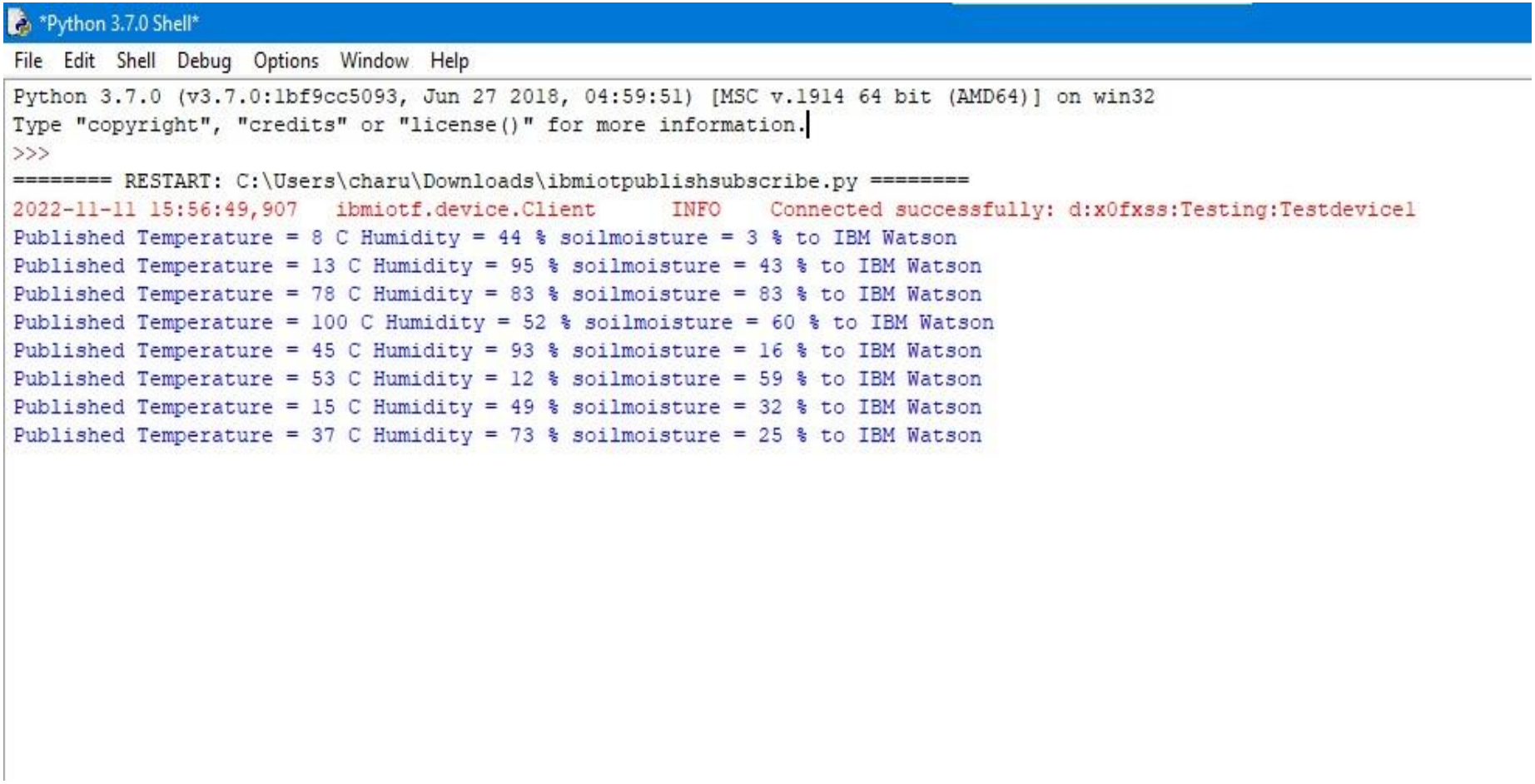
Python Script:

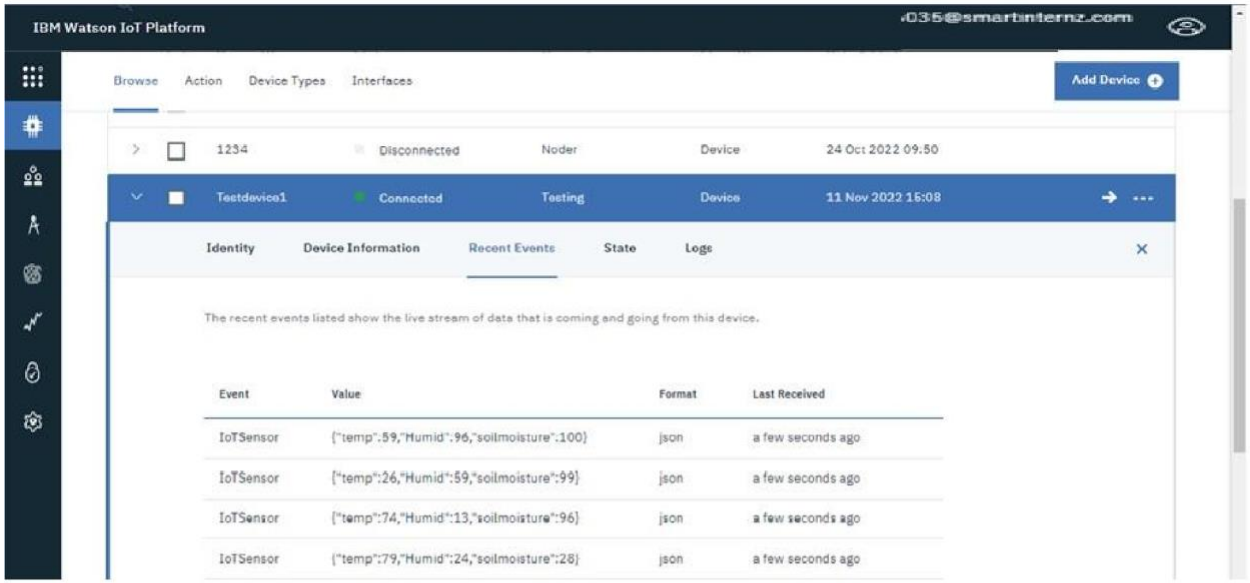
```
import  time  import  sys
import ibmiotf.application
import      ibmiotf.device
import random

#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)
```

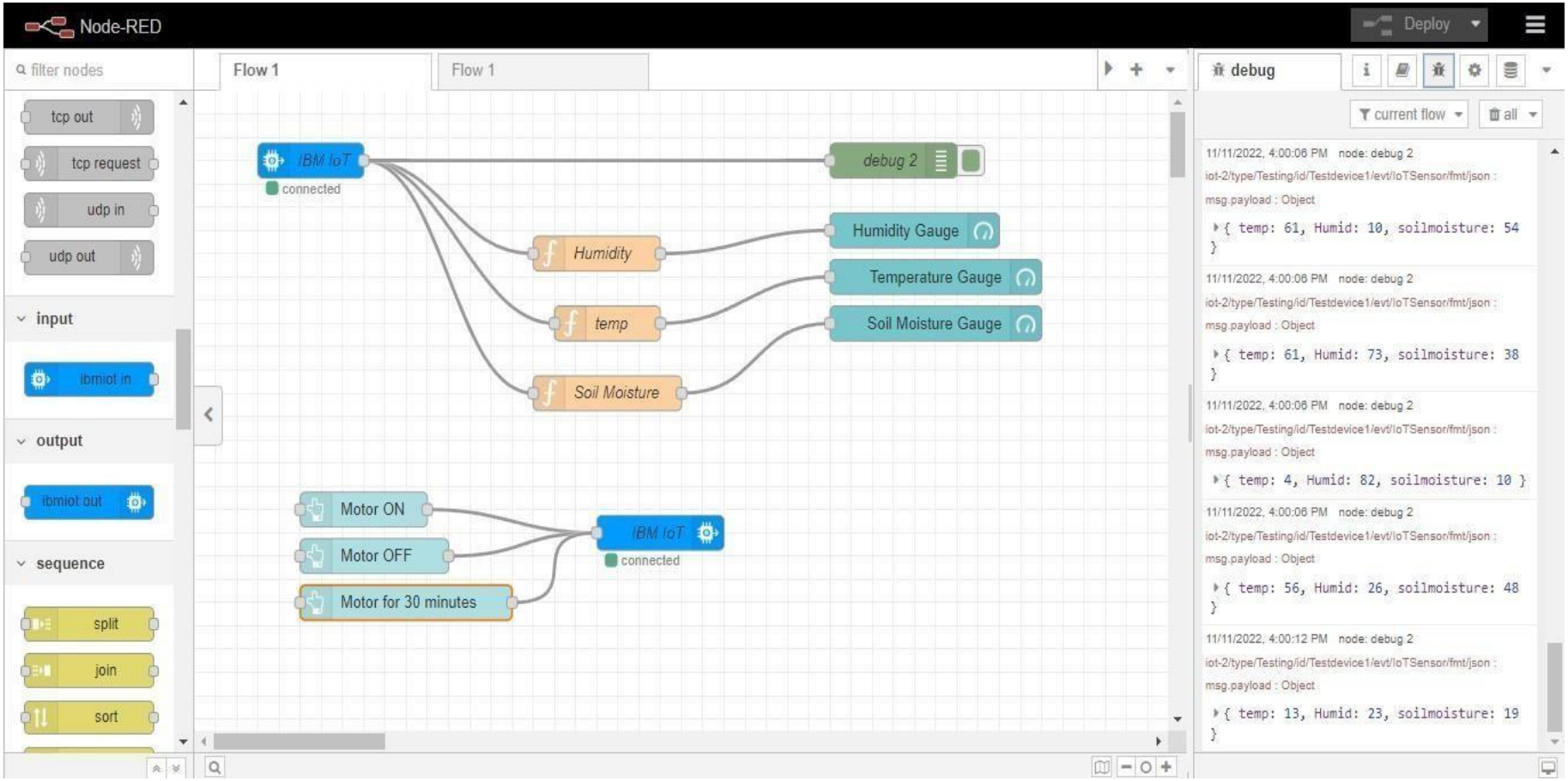
```
except Exception as e:
    print("Caught exception connecting device: %s" %
str(e)) sys.exit()
# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times
deviceCli.connect()
while True:
        #Get Sensor Data from DHT11

        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()
```

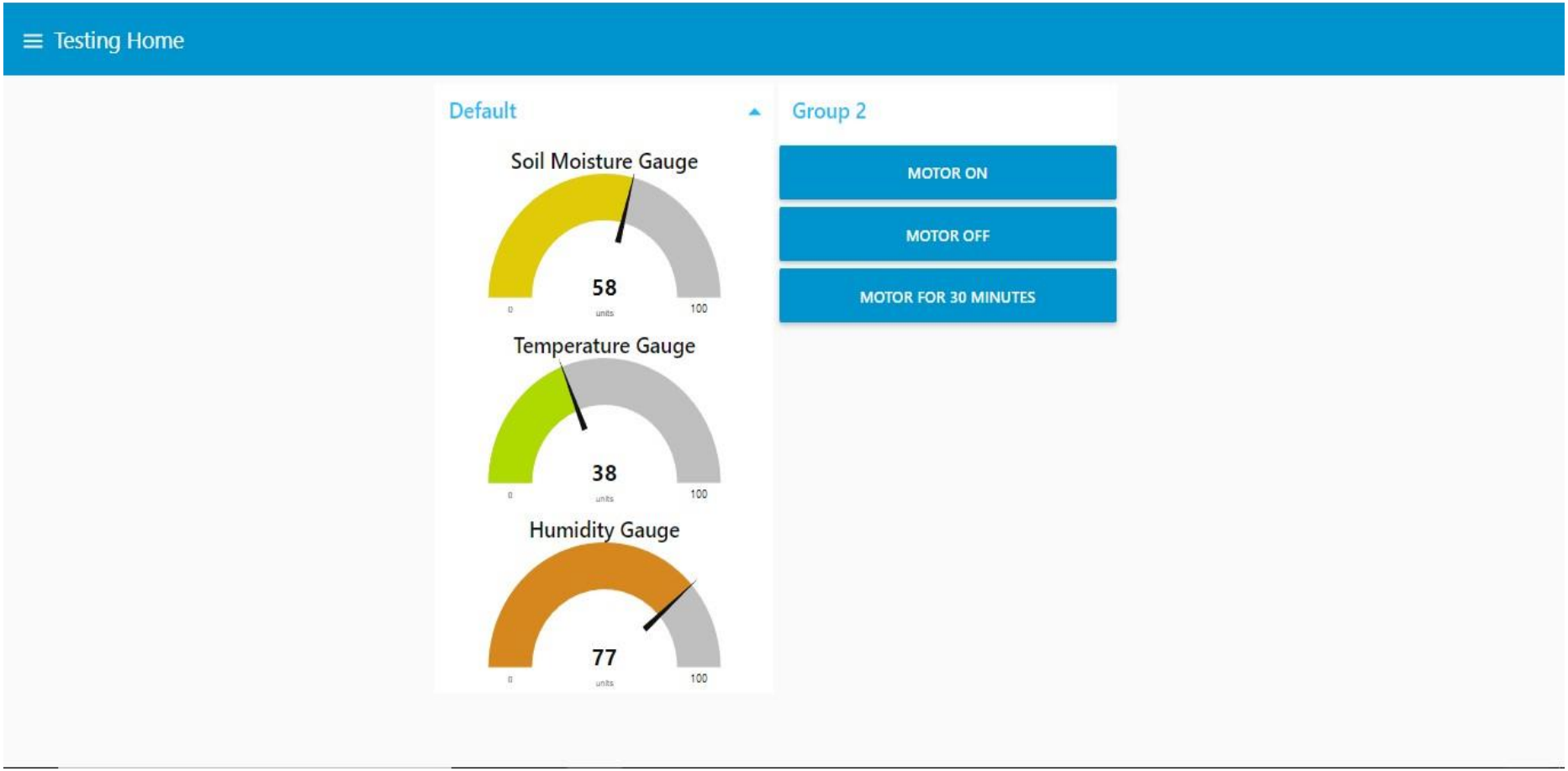
OUTPUT:

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



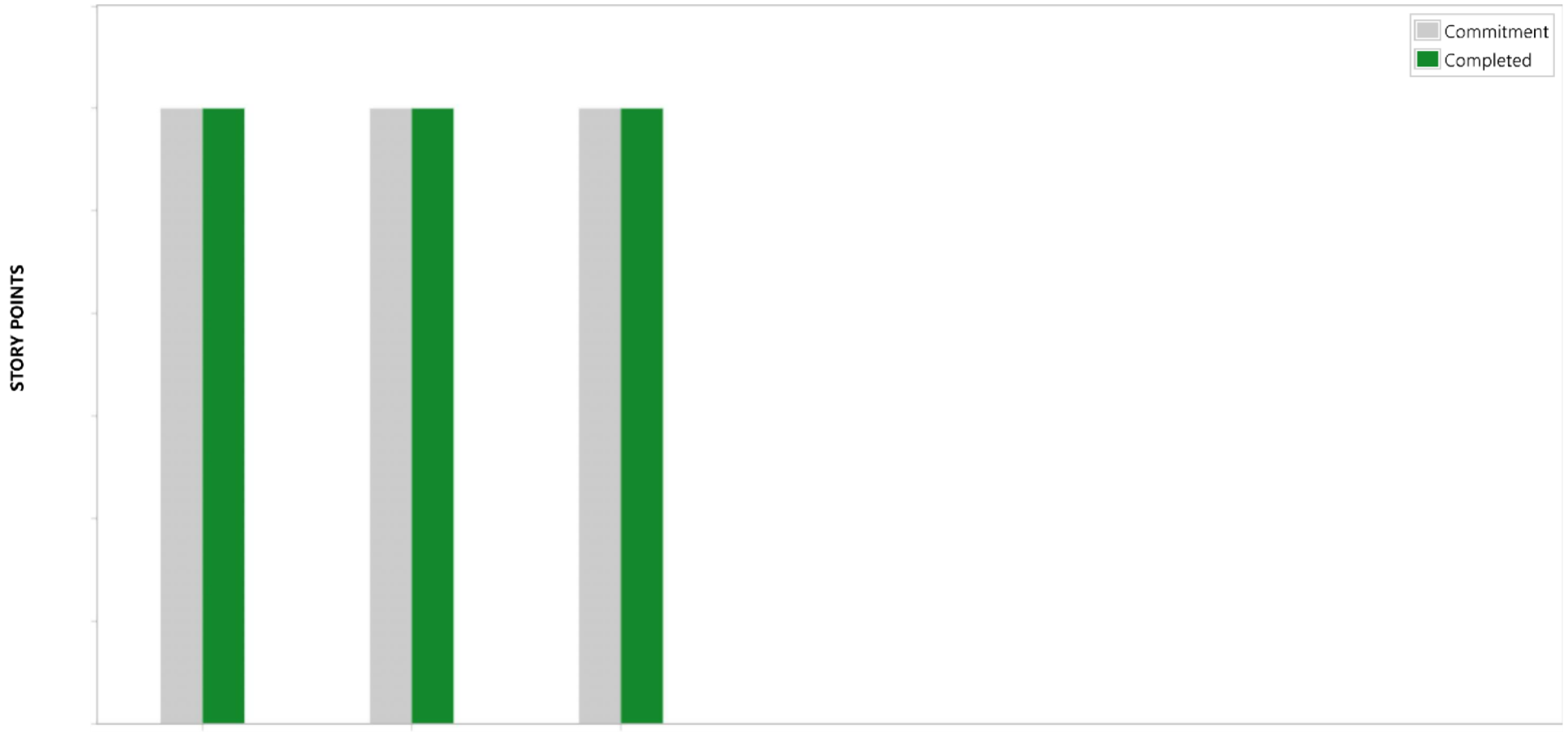

IBM Watson IoT Platform
035@smartinternz.com
Browse
Action
Device Types
Interfaces
Add Device
1234
Disconnected
Noder
Device
24 Oct 2022 09:50
Testdevice1
Connected
Testing
Device
11 Nov 2022 16:08
Identity
Device Information
Recent Events
State
Logs
The recent events listed show the live stream of data 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


Node-RED
Deploy
filter nodes
Flow 1
Flow 1
tcp out
tcp request
udp in
udp out
input
ibmiot in
output
ibmiot out
sequence
split
join
sort
IBM IoT
connected
debug 2
Humidity
Humidity Gauge
Temperature Gauge
temp
Soil Moisture Gauge
Soil Moisture
Motor ON
Motor OFF
Motor for 30 minutes
IBM IoT
connected
debug
current flow
all
11/11/2022, 4:00:06 PM node: debug 2
iot-2/type/Testing/id/Testdevice1/evt/IoTSensor/fmt/json : msg.payload : Object
{ temp: 61, Humid: 10, soilmoisture: 54 }
11/11/2022, 4:00:06 PM node: debug 2
iot-2/type/Testing/id/Testdevice1/evt/IoTSensor/fmt/json : msg.payload : Object
{ temp: 61, Humid: 73, soilmoisture: 38 }
11/11/2022, 4:00:06 PM node: debug 2
iot-2/type/Testing/id/Testdevice1/evt/IoTSensor/fmt/json : msg.payload : Object
{ temp: 4, Humid: 82, soilmoisture: 10 }
11/11/2022, 4:00:06 PM node: debug 2
iot-2/type/Testing/id/Testdevice1/evt/IoTSensor/fmt/json : msg.payload : Object
{ temp: 56, Humid: 26, soilmoisture: 48 }
11/11/2022, 4:00:12 PM node: debug 2
iot-2/type/Testing/id/Testdevice1/evt/IoTSensor/fmt/json : msg.payload : Object
{ temp: 13, Humid: 23, soilmoisture: 19 }

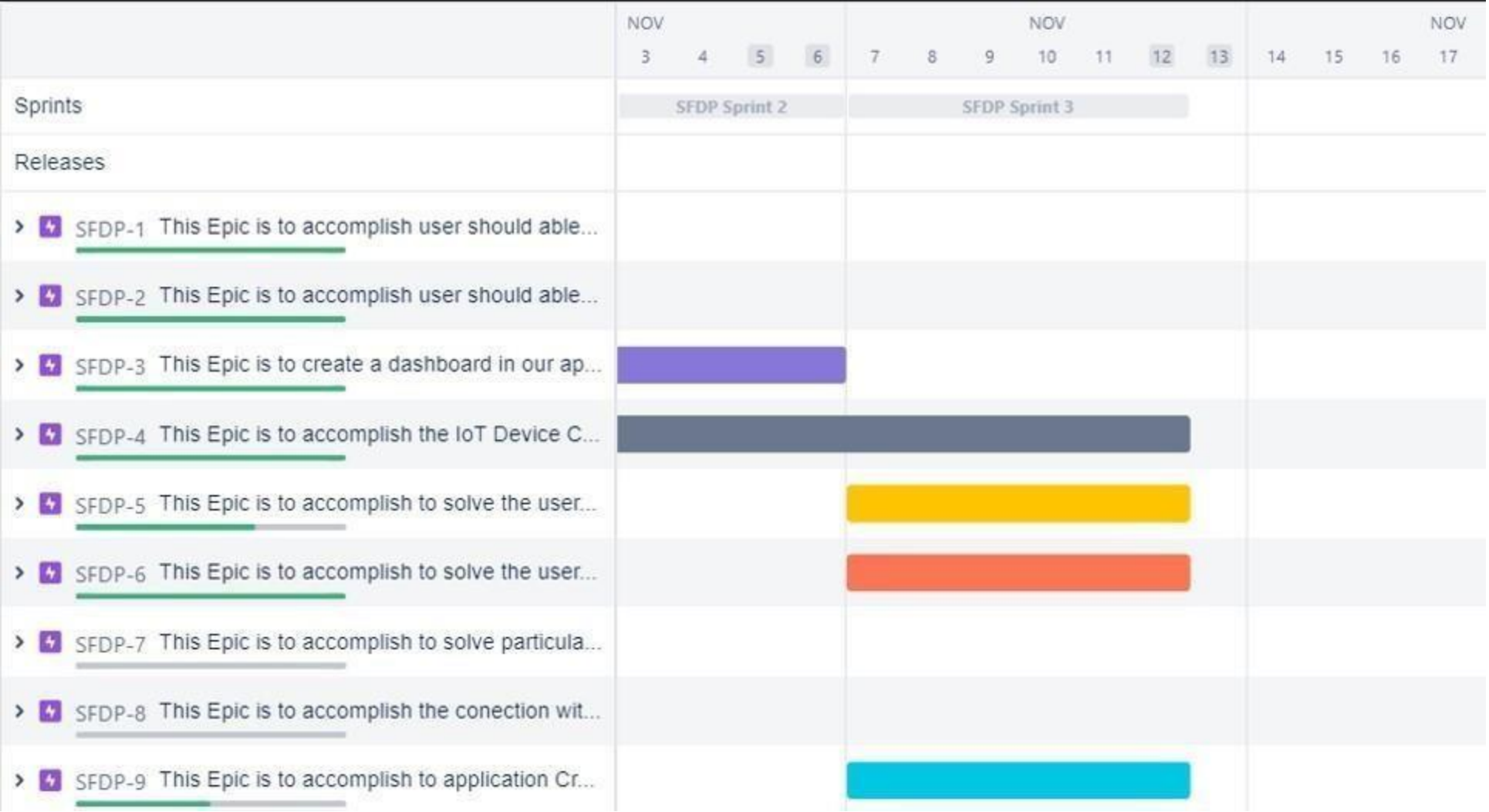


Data are successfully received and displayed.

Velocity Chart



Sprint	Commitment	Completed
SFDP Sprint 1	15	15
SFDP Sprint 2	15	15
SFDP Sprint 3	15	15


NOV
3
4
5
6
7
8
9
10
11
12
13
NOV
14
15
16
17
NOV
Sprints
SFDP Sprint 2
SFDP Sprint 3
Releases
SFDP-1 This Epic is to accomplish user should able...
SFDP-2 This Epic is to accomplish user should able...
SFDP-3 This Epic is to create a dashboard in our ap...
SFDP-4 This Epic is to accomplish the IoT Device C...
SFDP-5 This Epic is to accomplish to solve the user...
SFDP-6 This Epic is to accomplish to solve the user...
SFDP-7 This Epic is to accomplish to solve particula...
SFDP-8 This Epic is to accomplish the conection wit...
SFDP-9 This Epic is to accomplish to application Cr...

Projects / Smart Farmer Development Phase / SFDP board / Reports

Burndown Chart

SFDP Sprint 3 Story Points