IBM

SPRINT 3 REPORT

IOT ENABLED SMART FARMING APPLICATION

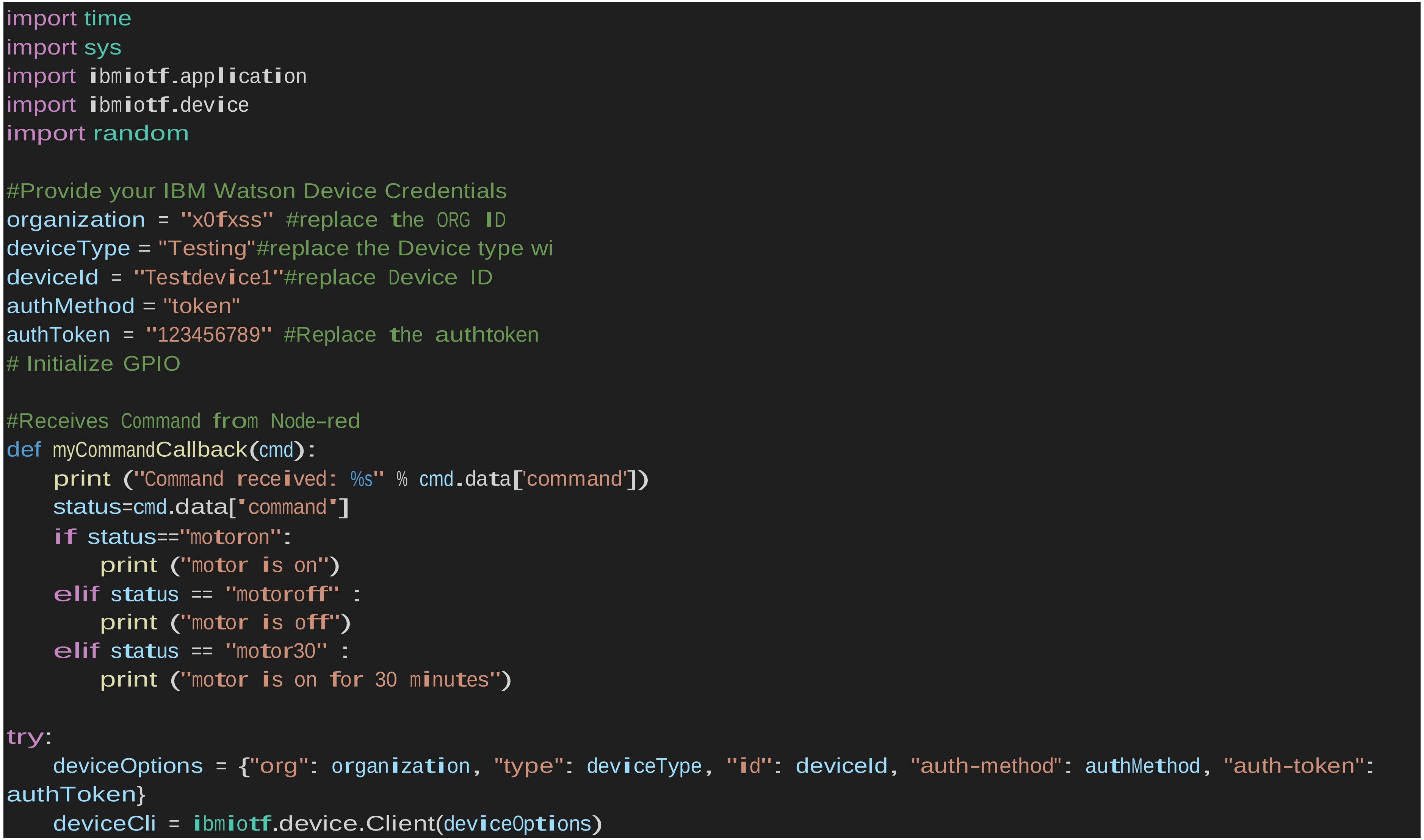
TEAM ID – PNT2022TMID39204
Alphino Alex. U
Kishore. A
Ajith Kumar. R
Ragul. M
Jayanth. S

Project Tracker

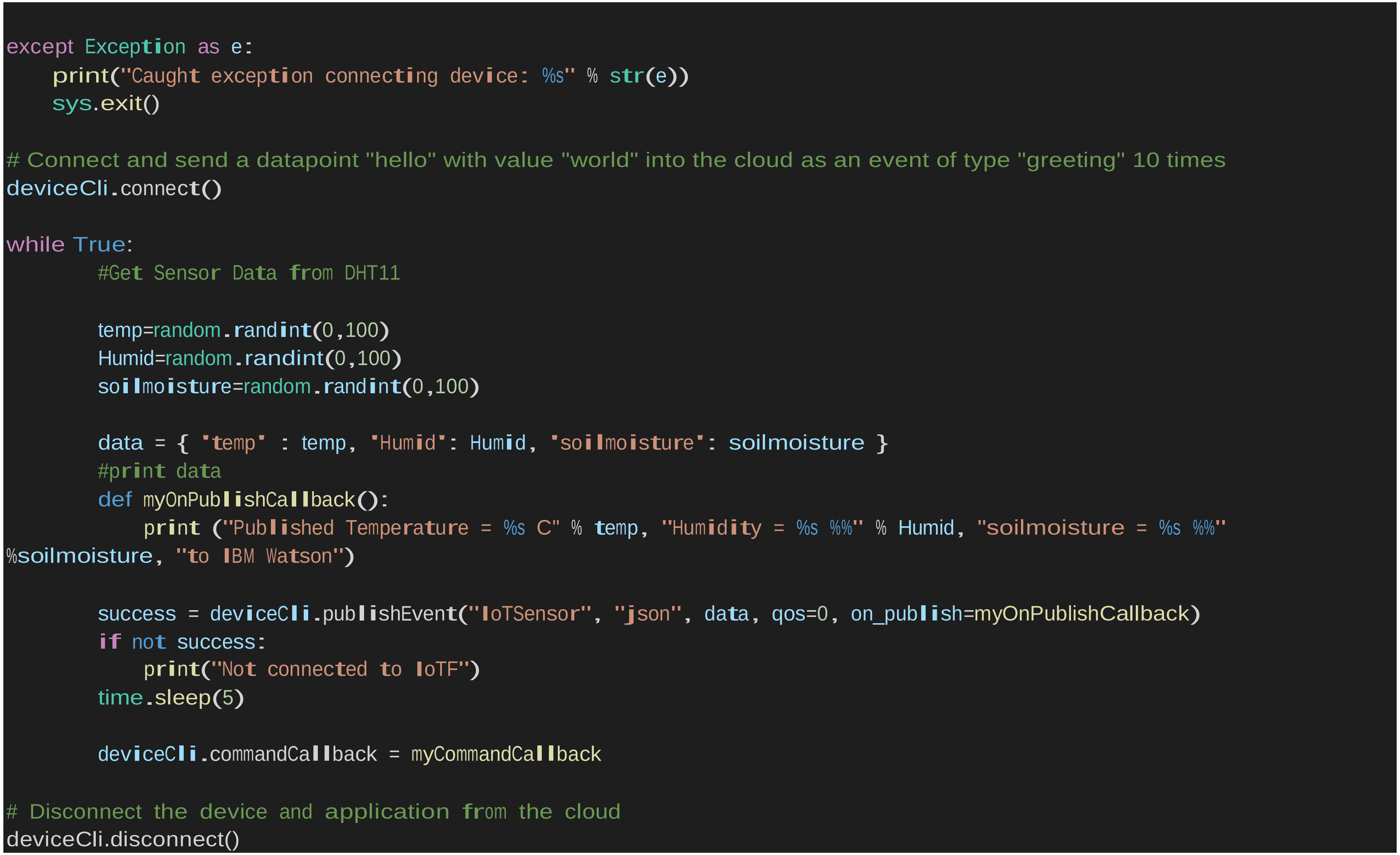
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	15	5 Days	26 Oct 2022	30 Oct 2022	15	30 Oct 2022
Sprint-2	15	7 Days	31 Oct 2022	06 Nov 2022	15	07 Nov 2022
Sprint-3	15	6 Days	07 Nov 2022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



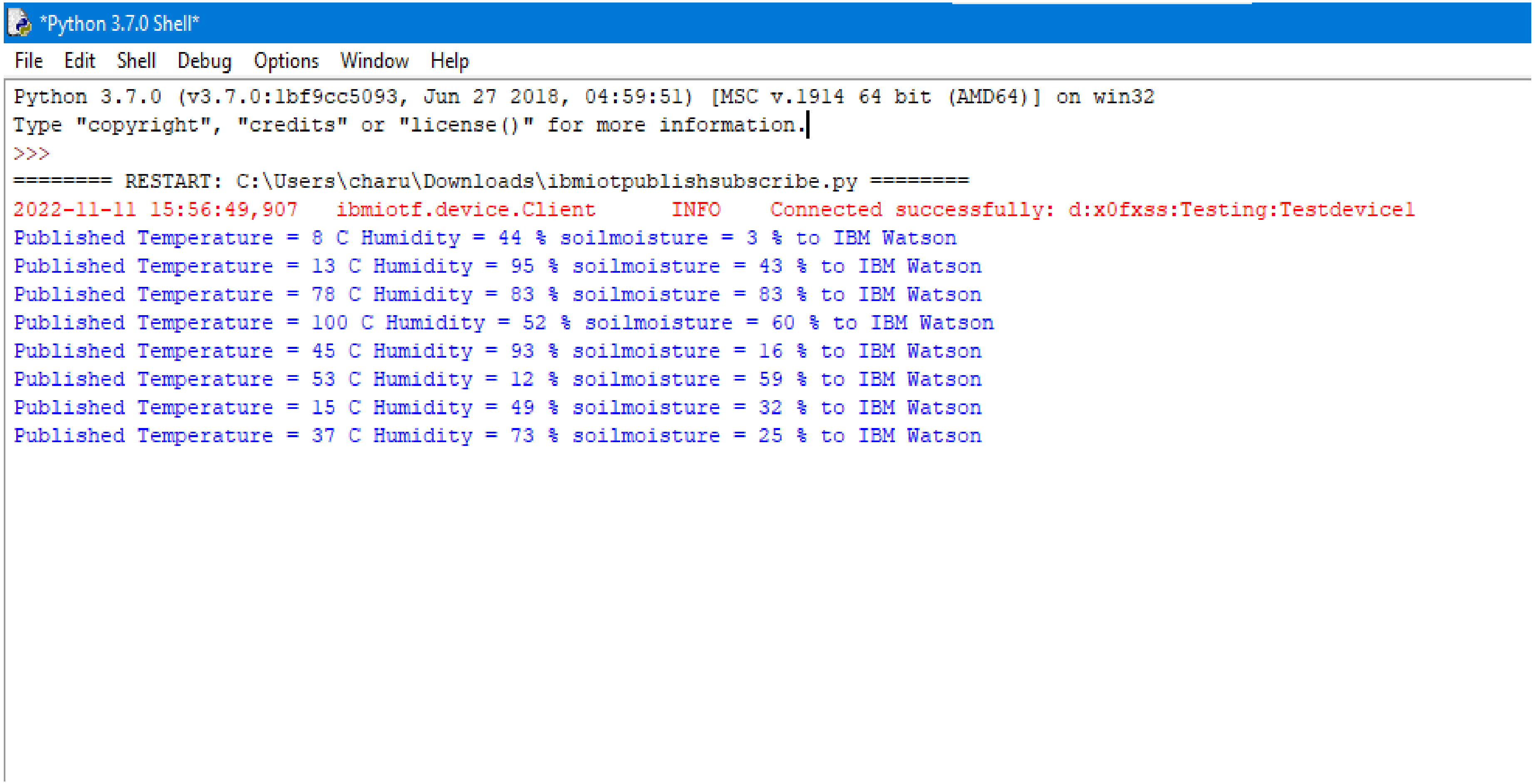
```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#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)
```



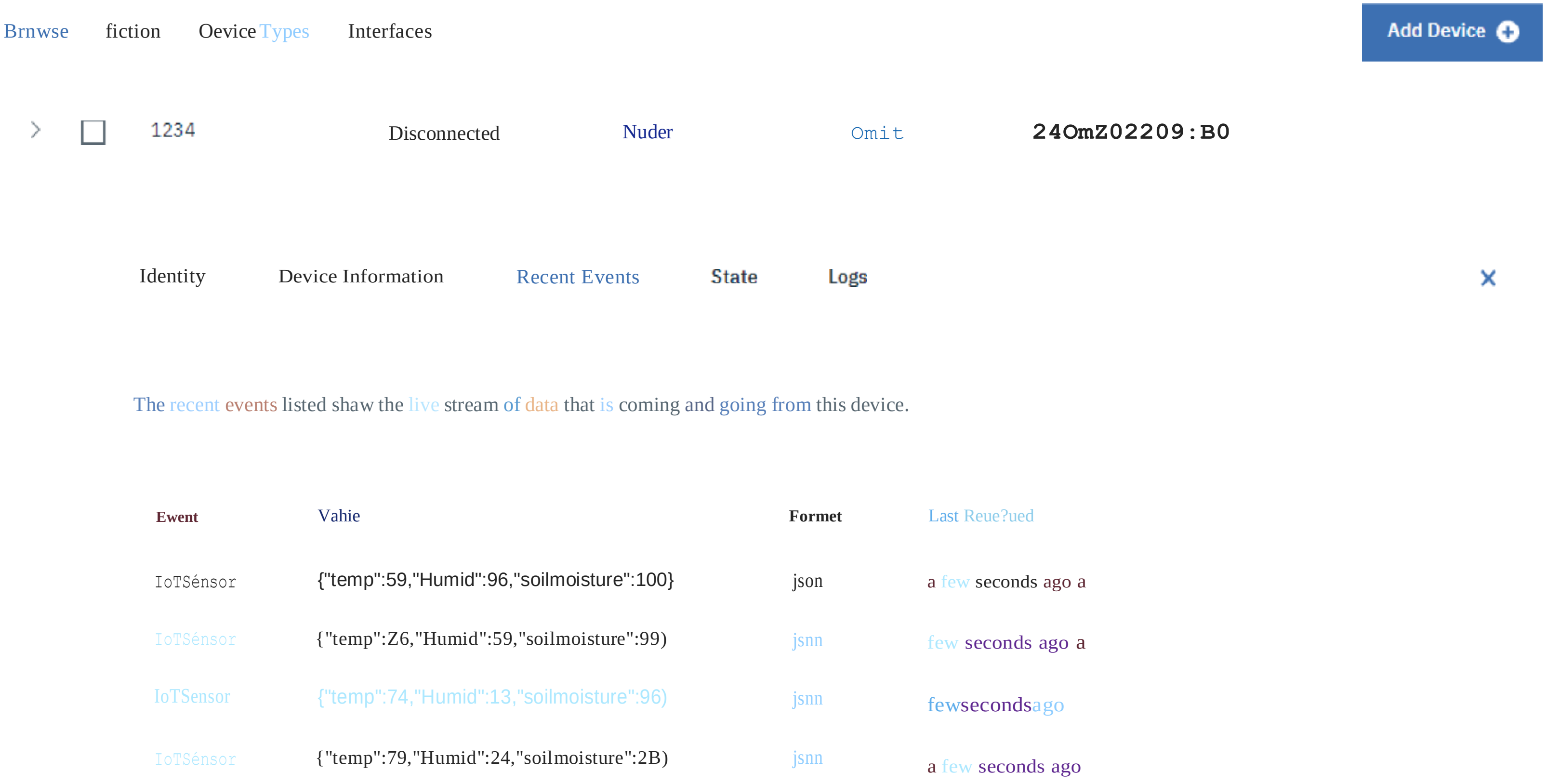
```
except Exception as e:
    print("Caught exception connecting device: %s" % str(e))
    sys.exit()

# 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.



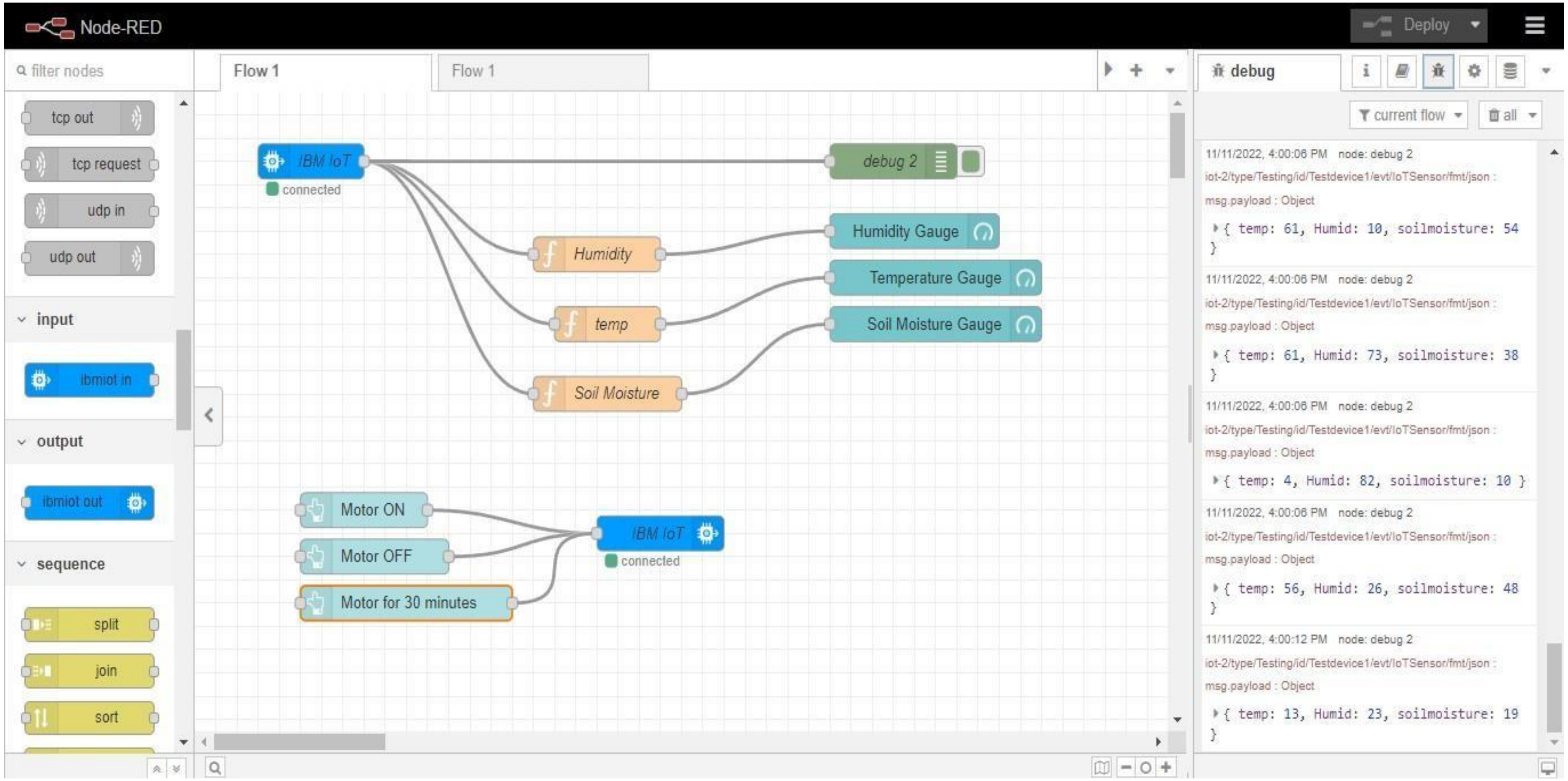
```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:\Users\charu\Downloads\ibmiotpublishsubscribe.py ========
2022-11-11 15:56:49,907   ibmiotf.device.Client      INFO    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
```



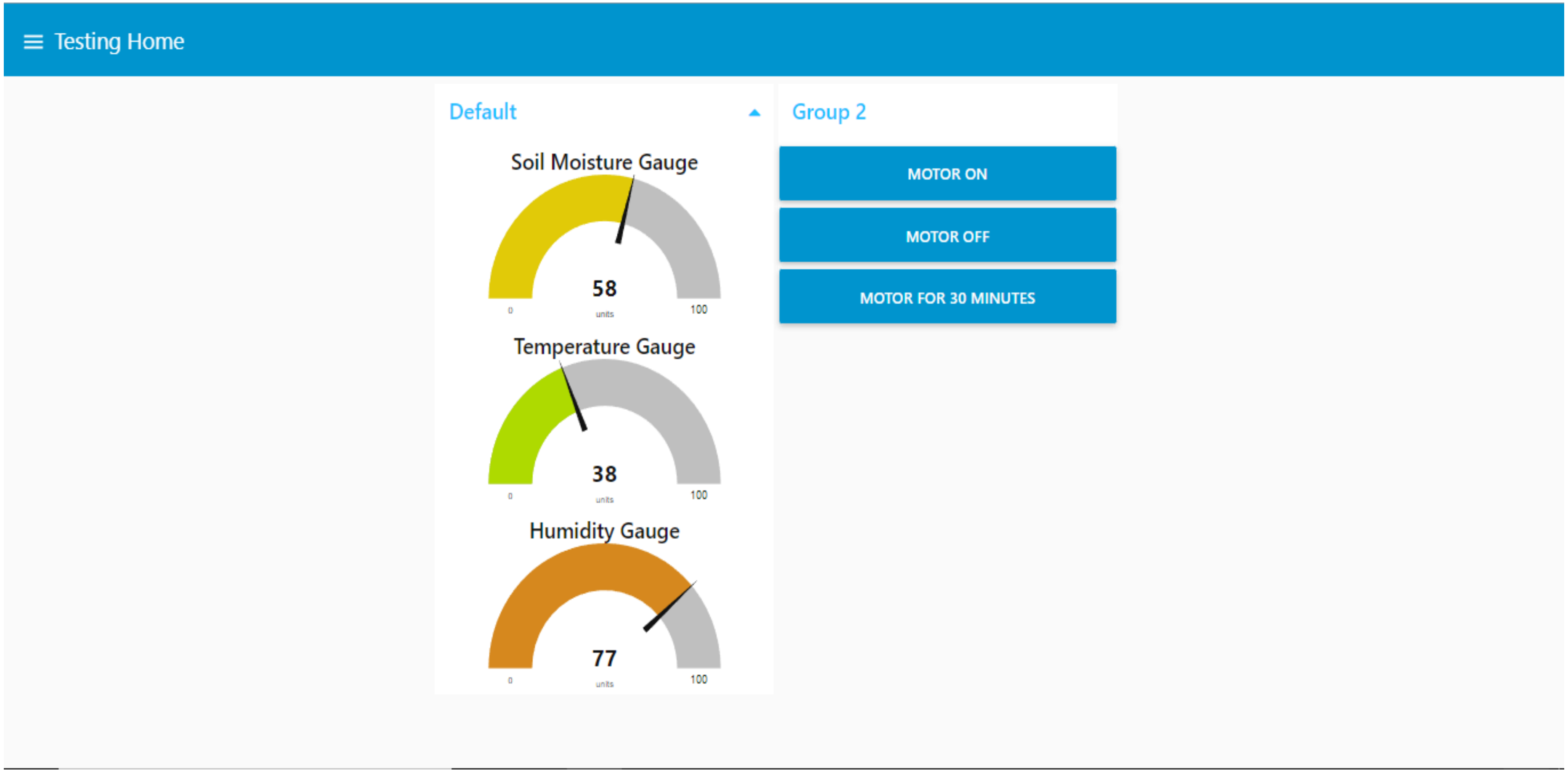
1234 Disconnected Nuder Omit 24OmZ02209:B0

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

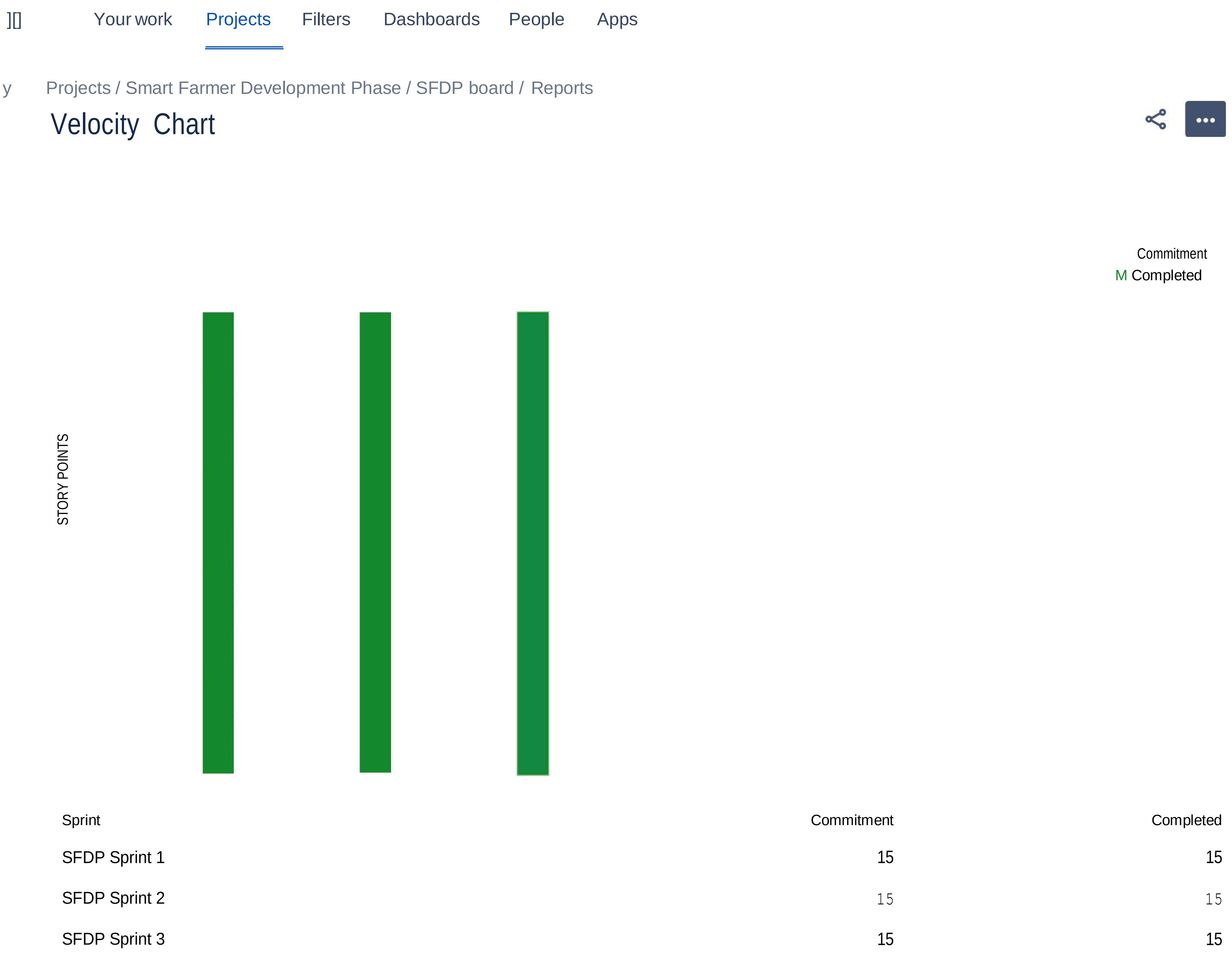
Ewent	Vahie	Formet	Last Reue?ued
IoTSénsor	{"temp":59,"Humid":96,"soilmoisture":100}	json	a few seconds ago a
IoTSénsor	{"temp":Z6,"Humid":59,"soilmoisture":99)	jsnn	few seconds ago a
IoTSensor	{"temp":74,"Humid":13,"soilmoisture":96)	jsnn	fewsecondsago
IoTSénsor	{"temp":79,"Humid":24,"soilmoisture":2B)	jsnn	a few seconds ago



```
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
 }
```



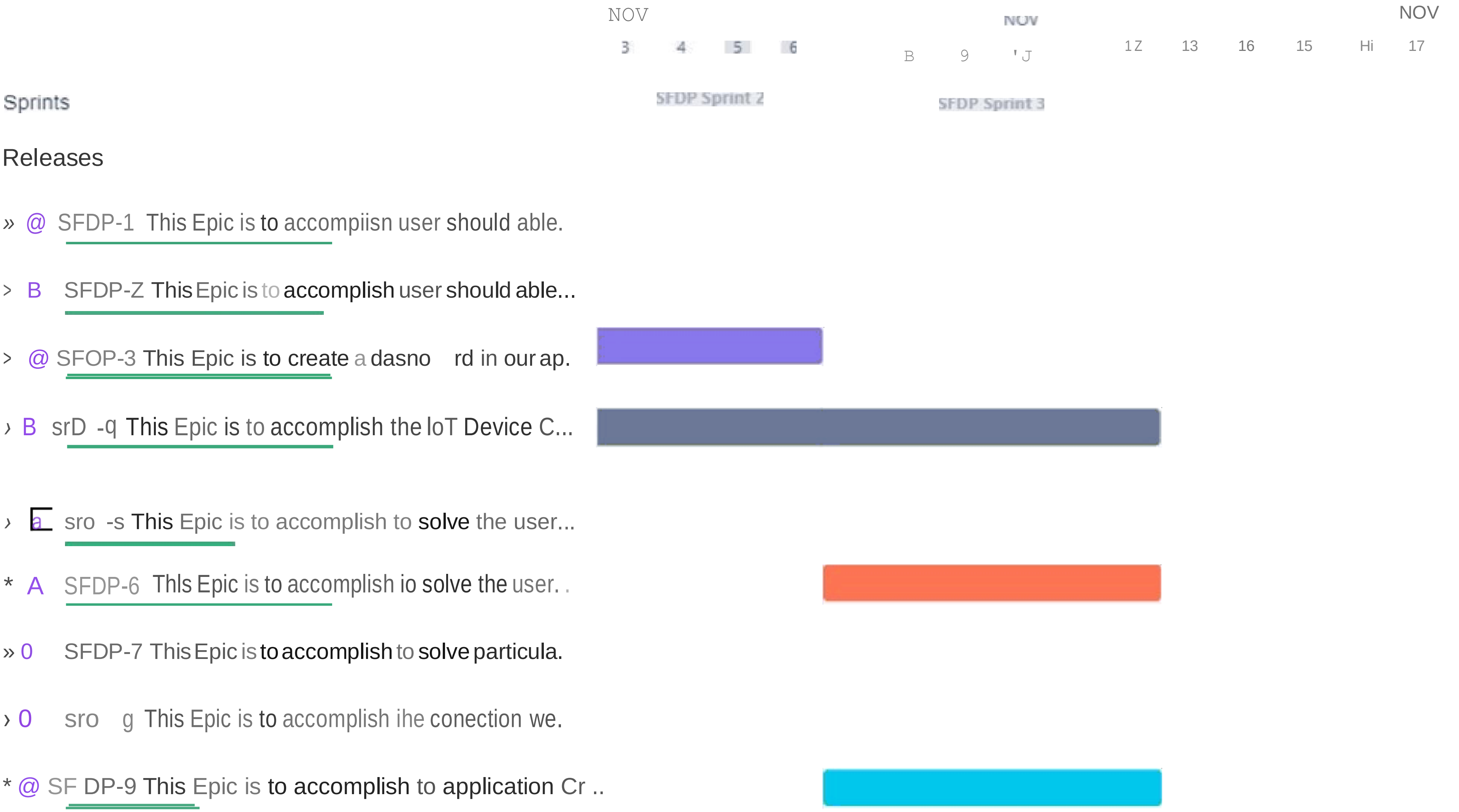
Data are successfully received and displayed.



Projects / Smart Farmer Development Phase / SFDP board / Reports

Velocity Chart

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



Sprints

Releases

» @ SFDP-1 This Epic is to accompiisn user should able.

> B SFDP-Z This Epic is to accomplish user should able...

> @ SFOP-3 This Epic is to create a dasno rd in our ap.

› B srD -q This Epic is to accomplish the loT Device C...

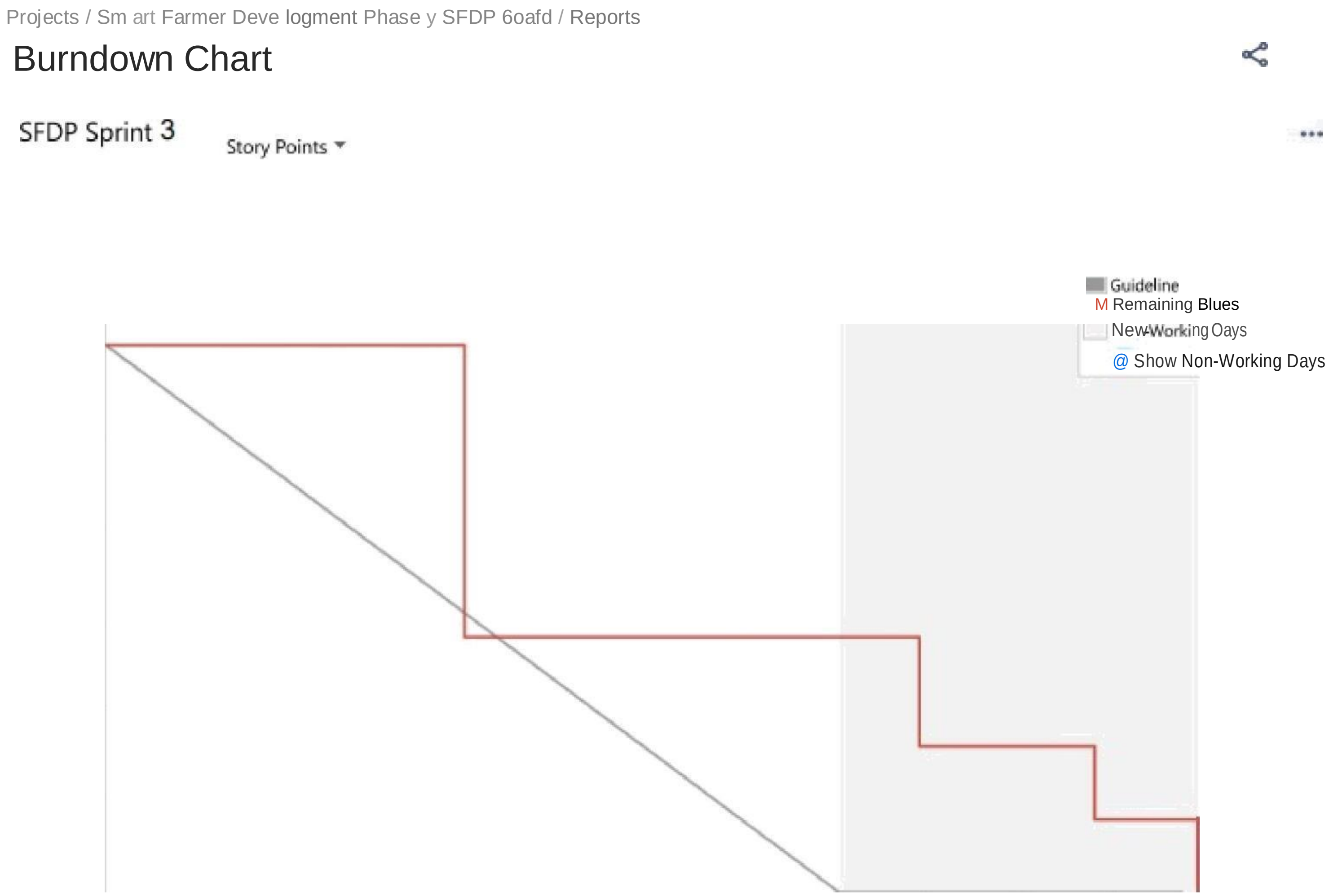
› a sro -s This Epic is to accomplish to solve the user...

* A SFDP-6 Thls Epic is to accomplish io solve the user. .

» 0 SFDP-7 This Epic is to accomplish to solve particula.

› 0 sro g This Epic is to accomplish ihe conection we.

* @ SF DP-9 This Epic is to accomplish to application Cr ..



Projects / Sm art Farmer Deve logment Phase y SFDP 6oafd / Reports

Burndown Chart

SFDP Sprint 3

Story Points ▾