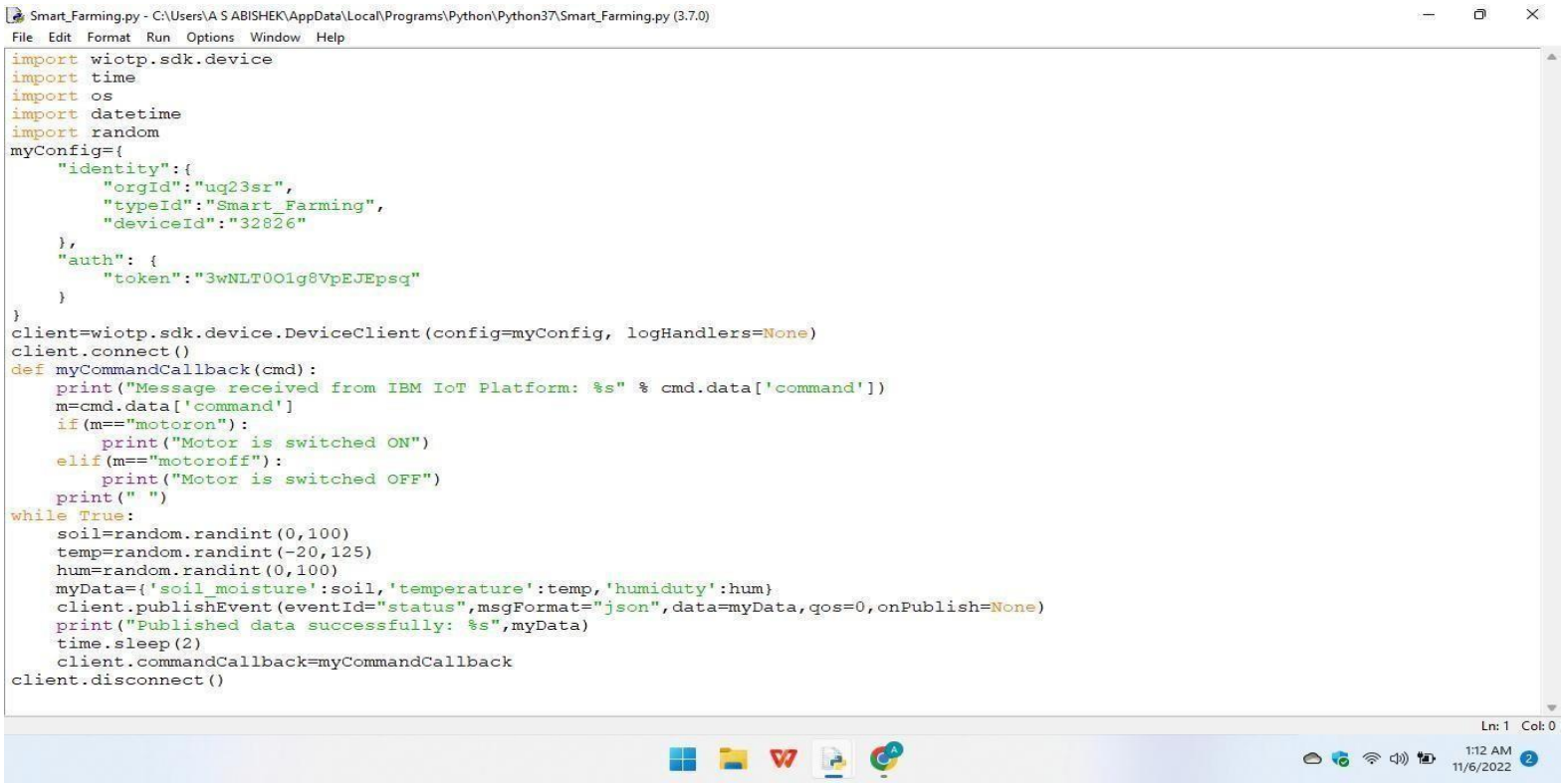
SMART FARMER – IOT ENABLEDD SMARTFARMING APPLICATION

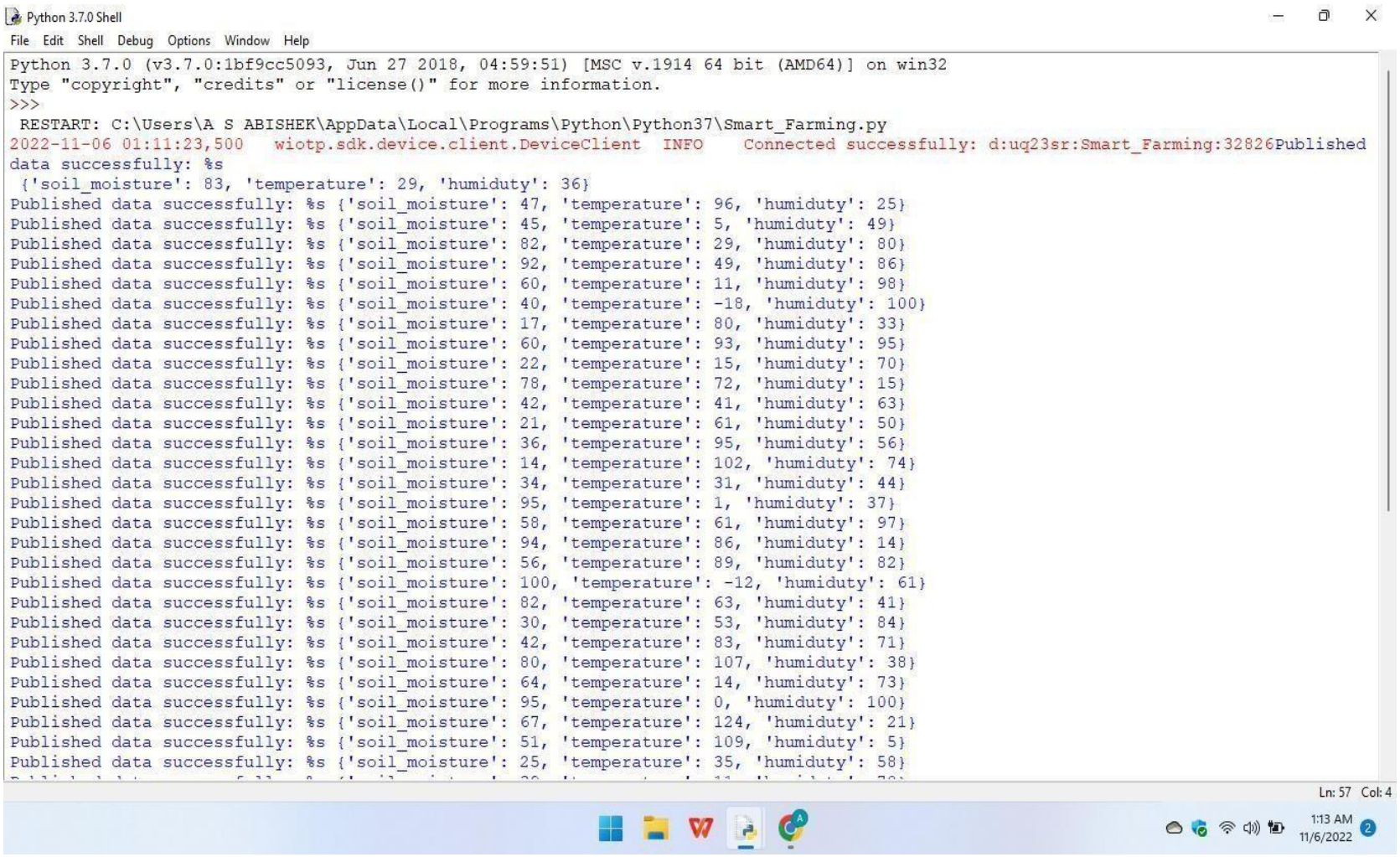
PROJECT DEVELOPMENT – DELIVERY OF

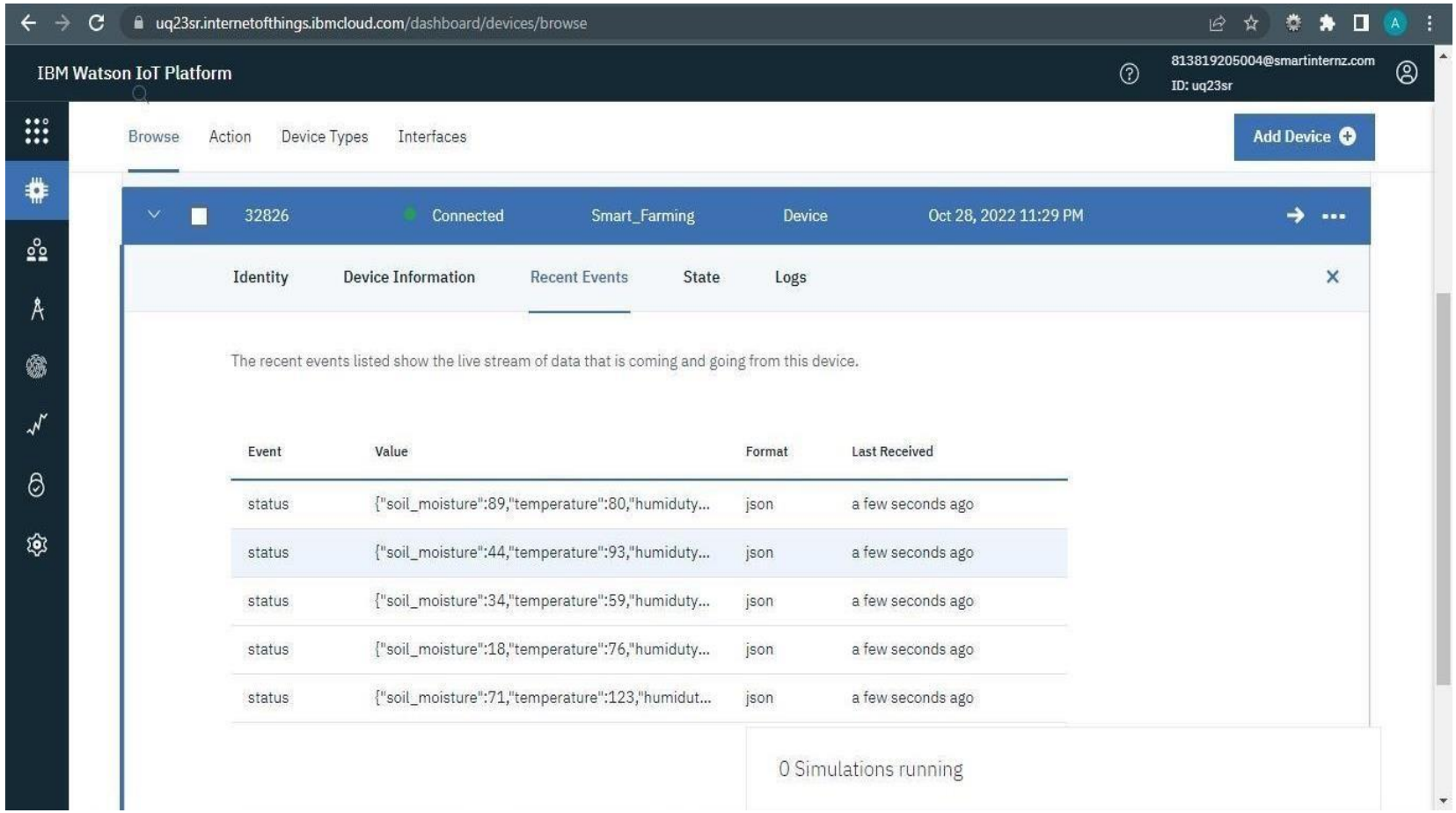
SPRINT – 2

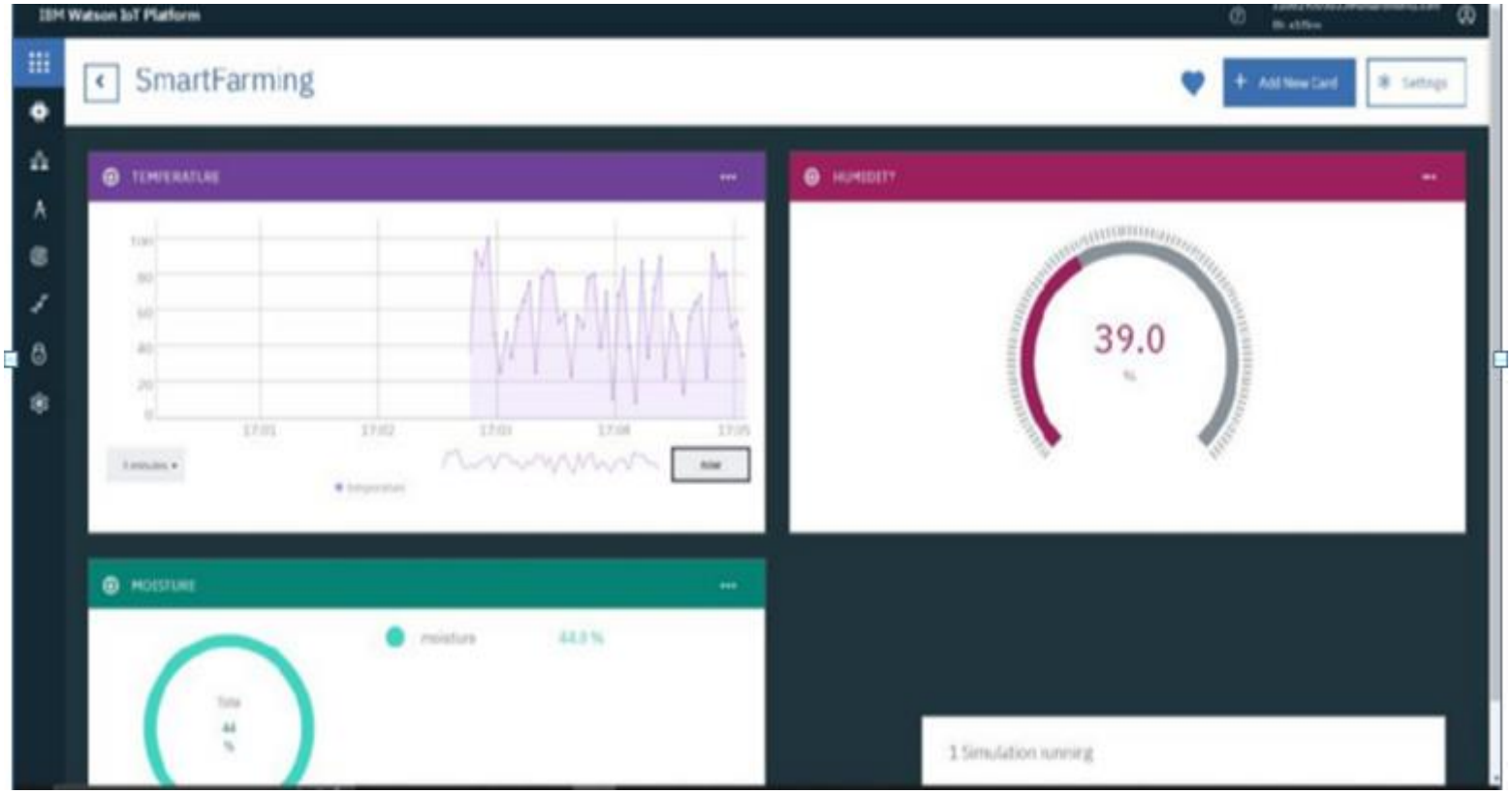
TITLE: SMART FARMER – IOT ENABLEDD SMART FARMING APPLICATION

TEAM: PNT2022TMID47823


Smart_Farming.py - C:\Users\A S ABISHEK\AppData\Local\Programs\Python\Python37\Smart_Farming.py (3.7.0)
File Edit Format Run Options Window Help
import wiotp.sdk.device
import time
import os
import datetime
import random
myConfig={
"identity":{
"orgId":"uq23sr",
"typeId":"Smart_Farming",
"deviceId":"32826"
},
"auth": {
"token":"3wNLT0O1g8VpEJEpsq"
}
}
client=wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()
def myCommandCallback(cmd):
print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
m=cmd.data['command']
if(m=="motoron"):
print("Motor is switched ON")
elif(m=="motoroff"):
print("Motor is switched OFF")
print(" ")
while True:
soil=random.randint(0,100)
temp=random.randint(-20,125)
hum=random.randint(0,100)
myData={'soil_moisture':soil,'temperature':temp,'humiduty':hum}
client.publishEvent(eventId="status",msgFormat="json",data=myData,qos=0,onPublish=None)
print("Published data successfully: %s",myData)
time.sleep(2)
client.commandCallback=myCommandCallback
client.disconnect()
Ln: 1 Col: 0
1:12 AM
11/6/2022


Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
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\A S ABISHEK\AppData\Local\Programs\Python\Python37\Smart_Farming.py
2022-11-06 01:11:23,500 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:uq23sr:Smart_Farming:32826Published data successfully: %s
{'soil_moisture': 83, 'temperature': 29, 'humiduty': 36}
Published data successfully: %s {'soil_moisture': 47, 'temperature': 96, 'humiduty': 25}
Published data successfully: %s {'soil_moisture': 45, 'temperature': 5, 'humiduty': 49}
Published data successfully: %s {'soil_moisture': 82, 'temperature': 29, 'humiduty': 80}
Published data successfully: %s {'soil_moisture': 92, 'temperature': 49, 'humiduty': 86}
Published data successfully: %s {'soil_moisture': 60, 'temperature': 11, 'humiduty': 98}
Published data successfully: %s {'soil_moisture': 40, 'temperature': -18, 'humiduty': 100}
Published data successfully: %s {'soil_moisture': 17, 'temperature': 80, 'humiduty': 33}
Published data successfully: %s {'soil_moisture': 60, 'temperature': 93, 'humiduty': 95}
Published data successfully: %s {'soil_moisture': 22, 'temperature': 15, 'humiduty': 70}
Published data successfully: %s {'soil_moisture': 78, 'temperature': 72, 'humiduty': 15}
Published data successfully: %s {'soil_moisture': 42, 'temperature': 41, 'humiduty': 63}
Published data successfully: %s {'soil_moisture': 21, 'temperature': 61, 'humiduty': 50}
Published data successfully: %s {'soil_moisture': 36, 'temperature': 95, 'humiduty': 56}
Published data successfully: %s {'soil_moisture': 14, 'temperature': 102, 'humiduty': 74}
Published data successfully: %s {'soil_moisture': 34, 'temperature': 31, 'humiduty': 44}
Published data successfully: %s {'soil_moisture': 95, 'temperature': 1, 'humiduty': 37}
Published data successfully: %s {'soil_moisture': 58, 'temperature': 61, 'humiduty': 97}
Published data successfully: %s {'soil_moisture': 94, 'temperature': 86, 'humiduty': 14}
Published data successfully: %s {'soil_moisture': 56, 'temperature': 89, 'humiduty': 82}
Published data successfully: %s {'soil_moisture': 100, 'temperature': -12, 'humiduty': 61}
Published data successfully: %s {'soil_moisture': 82, 'temperature': 63, 'humiduty': 41}
Published data successfully: %s {'soil_moisture': 30, 'temperature': 53, 'humiduty': 84}
Published data successfully: %s {'soil_moisture': 42, 'temperature': 83, 'humiduty': 71}
Published data successfully: %s {'soil_moisture': 80, 'temperature': 107, 'humiduty': 38}
Published data successfully: %s {'soil_moisture': 64, 'temperature': 14, 'humiduty': 73}
Published data successfully: %s {'soil_moisture': 95, 'temperature': 0, 'humiduty': 100}
Published data successfully: %s {'soil_moisture': 67, 'temperature': 124, 'humiduty': 21}
Published data successfully: %s {'soil_moisture': 51, 'temperature': 109, 'humiduty': 5}
Published data successfully: %s {'soil_moisture': 25, 'temperature': 35, 'humiduty': 58}
Ln: 57 Col: 4
1:13 AM
11/6/2022


uq23sr.internetofthings.ibmcloud.com/dashboard/devices/browse
IBM Watson IoT Platform
813819205004@smartinternz.com
ID: uq23sr
Browse
Action
Device Types
Interfaces
Add Device
32826
Connected
Smart_Farming
Device
Oct 28, 2022 11:29 PM
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
status
{"soil_moisture":89,"temperature":80,"humiduty...
json
a few seconds ago
status
{"soil_moisture":44,"temperature":93,"humiduty...
json
a few seconds ago
status
{"soil_moisture":34,"temperature":59,"humiduty...
json
a few seconds ago
status
{"soil_moisture":18,"temperature":76,"humiduty...
json
a few seconds ago
status
{"soil_moisture":71,"temperature":123,"humidut...
json
a few seconds ago
0 Simulations running



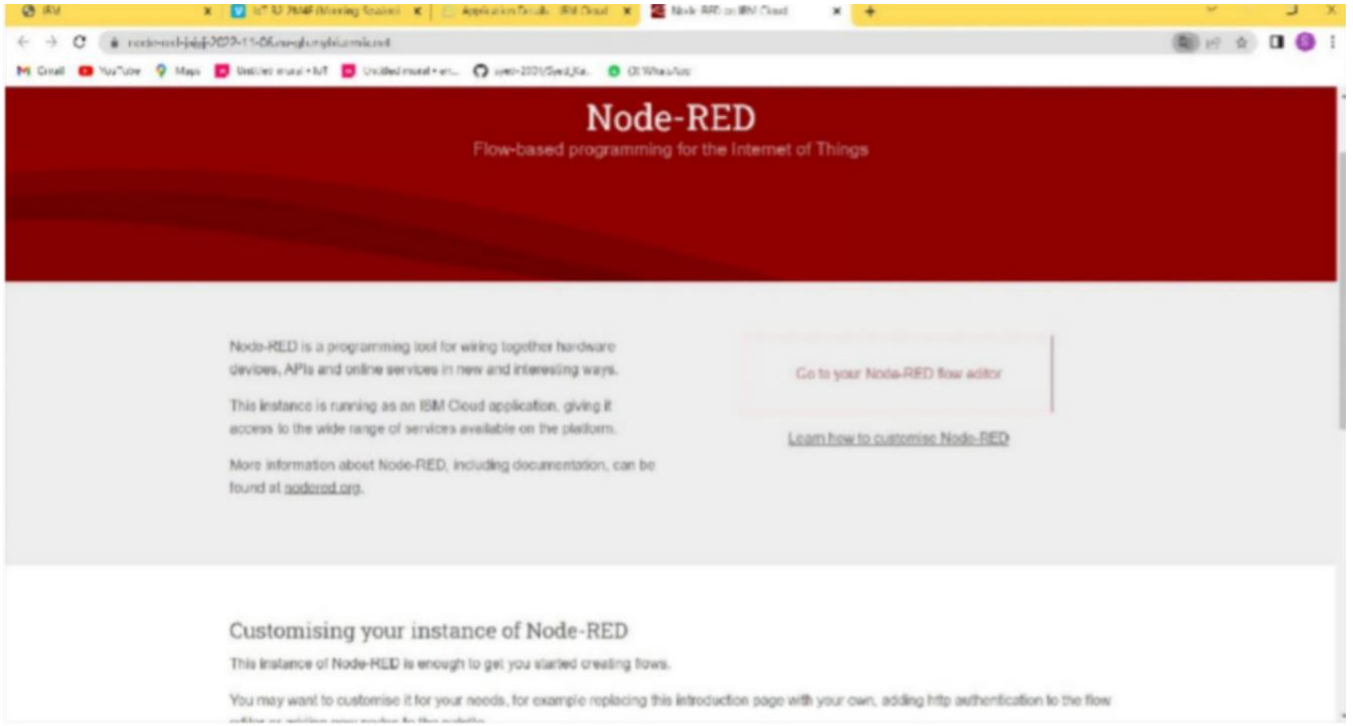
Explanation :

- To create device, in the home page of IBM cloud click on thecatalog on the top and click on IoT platform
- Click on launch button, then the IBM Watson platform will bedisplayed and Click on create device to create.
- After activating device simulator and check whether the code isrunning .
- Go to board and create a new board by filling the details.
- Fill the detail to get temperature graph, select the color from the option and repeat the same process to get the humidity graph, we
  get the final graph.
- Finally an IBM Watson cloud for IoT and a device is createdsuccessfully.

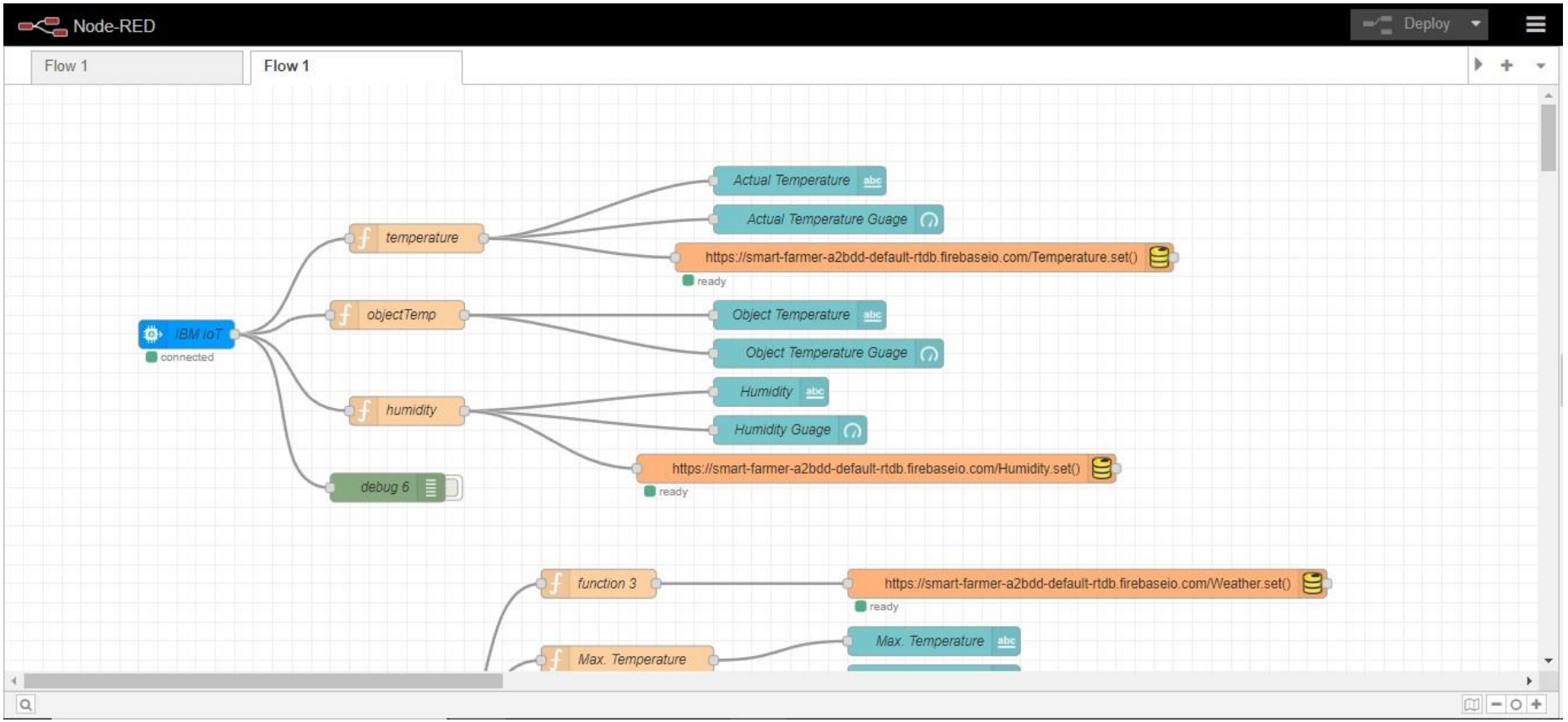
Note red

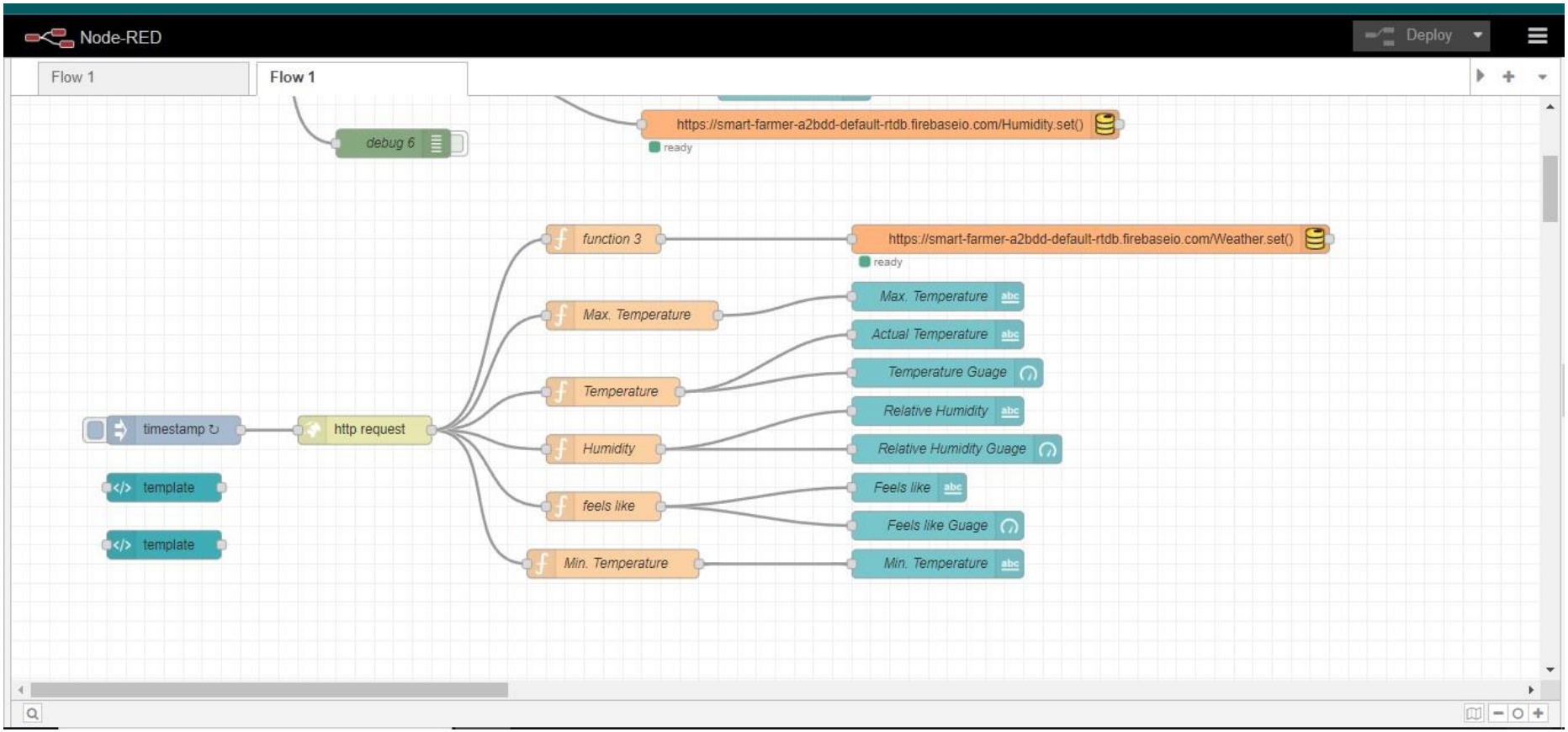
A flow-based programming tool for wiring together hardware devices, APIs, and online services. Create event-based apps with simple flow-based programming. Get started with Node-RED. Download Node-RED.
Search for Node Red in the catalog search box and click on get startbutton.

Do the required process and finish the installation steps.After the installation we can able to access the Node Red service. Then we can able to use nodes and connect then with other node .We can able to Run the node red by pressing on Deploy.


Node-RED
Flow-based programming for the Internet of Things
Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.
This instance is running as an IBM Cloud application, giving it access to the wide range of services available on the platform.
More information about Node-RED, including documentation, can be found at nodered.org.
Go to your Node-RED flow editor
Learn how to customise Node-RED
Customising your instance of Node-RED
This instance of Node-RED is enough to get you started creating flows.
You may want to customise it for your needs, for example replacing this introduction page with your own, adding http authentication to the flow

Node Red Connection:




Node-RED
Deploy
Flow 1
Flow 1
debug 6
https://smart-farmer-a2bdd-default-rtdb.firebaseio.com/Humidity.set()
ready
function 3
https://smart-farmer-a2bdd-default-rtdb.firebaseio.com/Weather.set()
ready
Max. Temperature
Max. Temperature
Actual Temperature
Temperature
Temperature Guage
Relative Humidity
timestamp
http request
Humidity
Relative Humidity Guage
template
feels like
Feels like
Feels like Guage
template
Min. Temperature
Min. Temperature