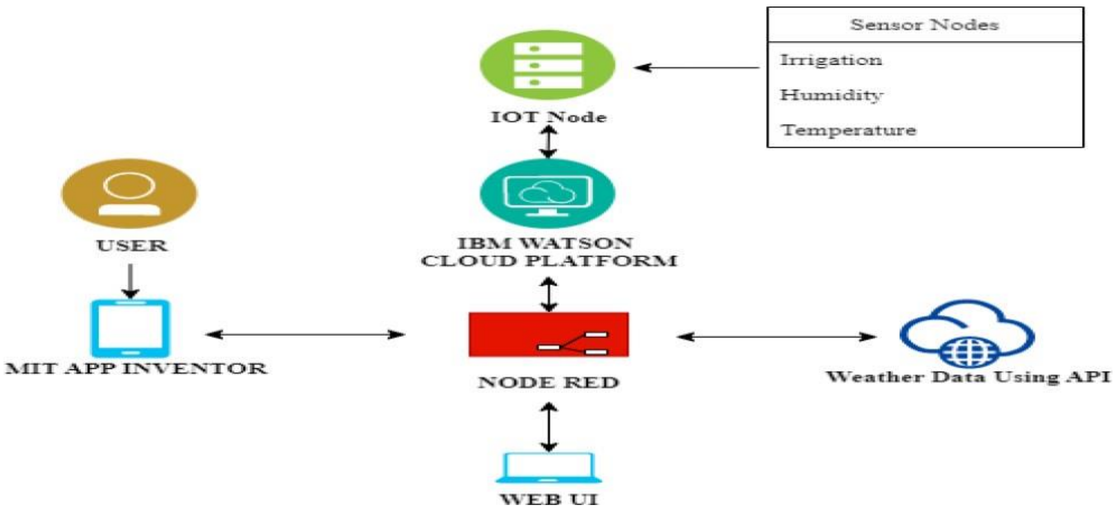
Project Development -Delivery of Sprint-4

Date	17 NOV 2022
Team ID	PNT2022TMID48510
Project Name	Project -Smart farmer-IOT enabled smart Farming Application

Flow Diagram



Python Code:

- **For Connecting IBM Cloud**
- **For NODE RED**
- **Weather Map Information**
- **MIT App Inventor**

```
#IBM Watson IOT
Platform#pip install wiotp-
sdk import
wiotp.sdk.device import
time
```

```
import random
import requests, json

ms=0
# Enter your API key here
api_key = "a0db30a689a774b93ffcb58ef2eddfda"#
base_url variable to store url
base_url = "http://api.openweathermap.org/data/2.5/weather?"#
Give city name
city_name = 'Chennai, IN'
# complete_url variable to store#
complete url address
complete_url = base_url + "appid=" + api_key + "&q=" + city_name

status='motor off'
myConfig = {
   "identity": {
      "orgId":
      "17lsro",
      "typeId":
      "MyDeviceType",
      "deviceId":"12345"
   },
   "auth": {
      "token": "GkatKdiUS?UVHKvnAD"
   }
}
```

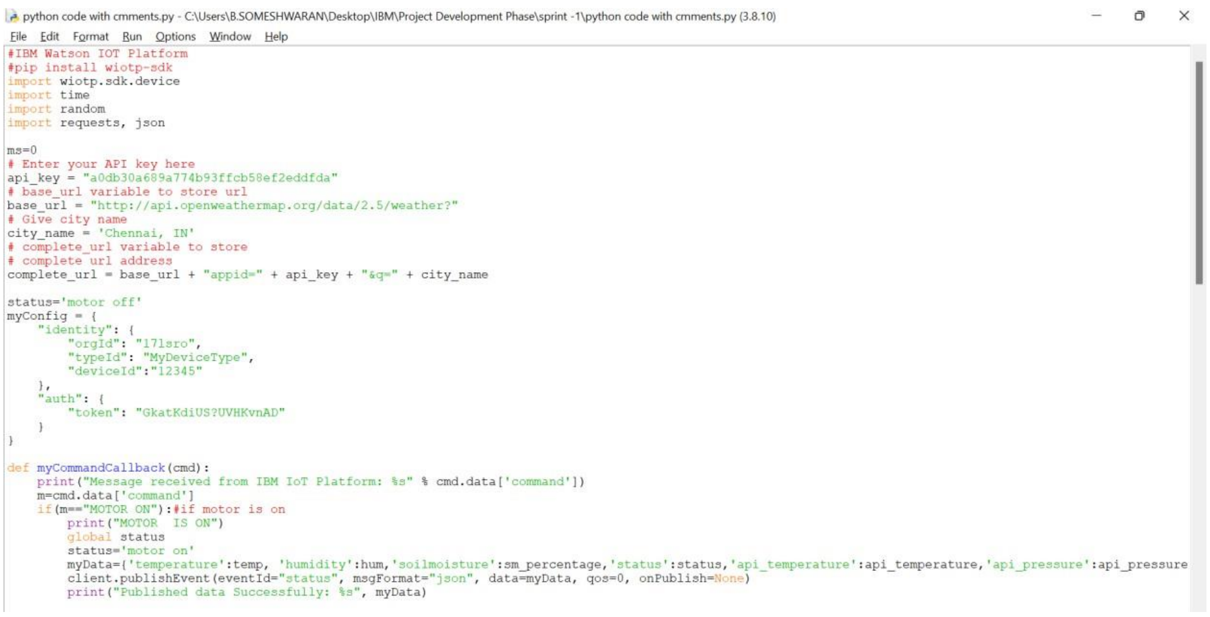
```
def myCommandCallback(cmd):
   print("Message received from IBM IoT Platform: %s"
%cmd.data['command'])
   m=cmd.data['command']
   if(m=="MOTOR ON"):#if motor is on
      print("MOTOR  IS ON")
      global status
      status='motor on'
      myData={'temperature':temp,
'humidity':hum,'soilmoisture':sm_percentage,'status':status,'api_temperature':
api_temperature,'api_pressure':api_pressure,'api_humidity':api_humidity,'api
_weather_description':api_weather_description}
      client.publishEvent(eventId="status", msgFormat="json", data=myData,
qos=0, onPublish=None)
      print("Published data Successfully: %s", myData)


      time.sleep(2)


   elif(m=="MOTOR OFF"):#if motor is
      offprint("MOTOR IS OFF")


      status='motor off' myData={'temperature':temp,
'humidity':hum,'soilmoisture':sm_percentage,'status':status,'api_temperature':
api_temperature,'api_pressure':api_pressure,'api_humidity':api_humidity,'api
_weather_description':api_weather_description}
      client.publishEvent(eventId="status", msgFormat="json", data=myData,
qos=0, onPublish=None)
      print("Published data Successfully: %s", myData)
```

```
        time.sleep(2)


client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    # get method of requests module#
    return response object
    response = requests.get(complete_url)#
    json method of response object
    # convert json format data into
    # python format data
    x = response.json()
    # Now x contains list of nested dictionaries
    # Check the value of "cod" key is equal to
    # "404", means city is found otherwise,
    # city is not found
    if x["cod"] !=
    "404":





        y = x["main"]
        api_temperature = y["temp"]#getting api temperature data

        api_pressure = y["pressure"]#getting api pressure data

        api_humidity = y["humidity"] #getting api humidity data

        z = x["weather"]

        api_weather_description = z[0]["description"]#getting api weather
condition data

    temp=random.randint(-20,125)#geneating ranom values for temperature
    hum=random.randint(0,100)#geneating ranom values for humidity
    soilmoisture=random.randint(0,1023)#analog sensor
    sm_percentage=(soilmoisture/1023)*100
    sm_percentage=int(sm_percentage)#geneating ranom values for
soilmoisture
    myData={'temperature':temp,
'humidity':hum,'soilmoisture':sm_percentage,'status':status,'api_temperature':
api_temperature,'api_pressure':api_pressure,'api_humidity':api_humidity,'api
_weather_description':api_weather_description}
    client.publishEvent(eventId="status", msgFormat="json", data=myData,qos=0,
onPublish=None)
```

print("Published data Successfully: %s", myData) client.commandCallback = myCommandCallbacktime.sleep(2)

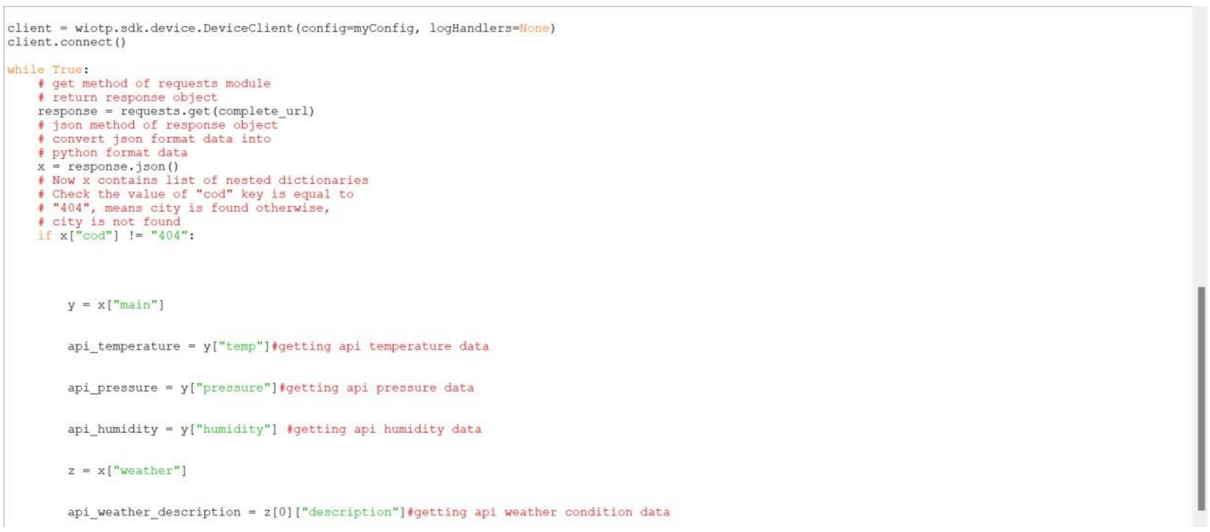
time.sleep(2)

client.disconnect()

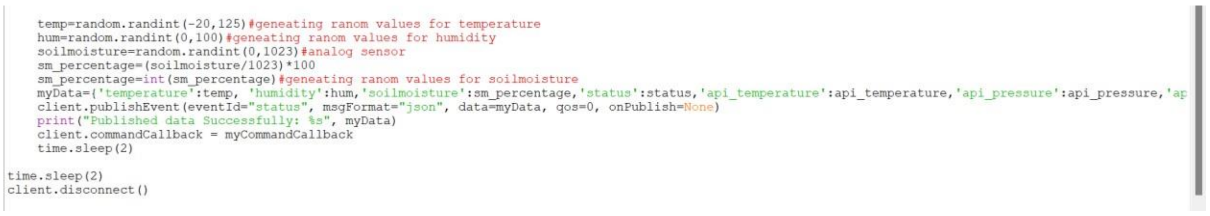


python code with cmments.py - C:\Users\B.SOMESHWARAN\Desktop\IBM\Project Development Phase\sprint -1\python code with cmments.py (3.8.10)

```
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
import requests, json

ms=0
# Enter your API key here
api_key = "a0db30a689a774b93ffcb58ef2eddfda"
# base_url variable to store url
base_url = "http://api.openweathermap.org/data/2.5/weather?"
# Give city name
city_name = 'Chennai, IN'
# complete_url variable to store
# complete url address
complete_url = base_url + "appid=" + api_key + "&q=" + city_name

status='motor off'
myConfig = {
    "identity": {
        "orgId": "171sro",
        "typeId": "MyDeviceType",
        "deviceId":"12345"
    },
    "auth": {
        "token": "GkatKdiUS?UVHKvnAD"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']
    if(m=="MOTOR ON"):#if motor is on
        print("MOTOR  IS ON")
        global status
        status='motor on'
        myData={'temperature':temp, 'humidity':hum,'soilmoisture':sm_percentage,'status':status,'api_temperature':api_temperature,'api_pressure':api_pressure
        client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
        print("Published data Successfully: %s", myData)
```

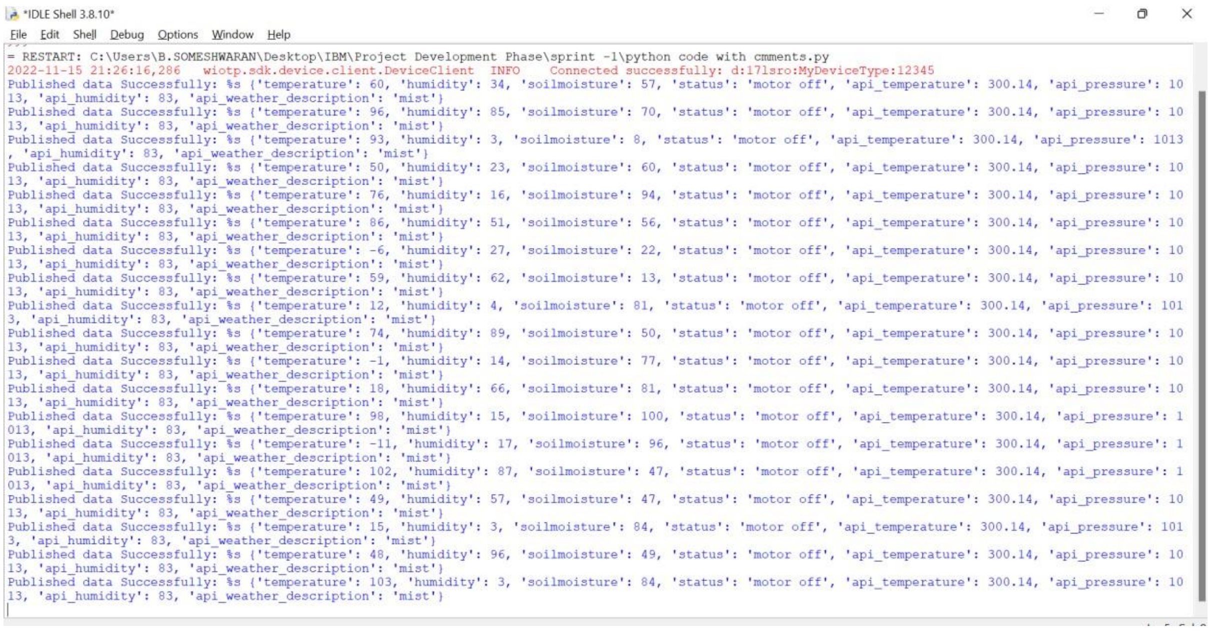


```
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    # get method of requests module
    # return response object
    response = requests.get(complete_url)
    # json method of response object
    # convert json format data into
    # python format data
    x = response.json()
    # Now x contains list of nested dictionaries
    # Check the value of "cod" key is equal to
    # "404", means city is found otherwise,
    # city is not found
    if x["cod"] != "404":



        y = x["main"]


        api_temperature = y["temp"]#getting api temperature data


        api_pressure = y["pressure"]#getting api pressure data


        api_humidity = y["humidity"] #getting api humidity data


        z = x["weather"]


        api_weather_description = z[0]["description"]#getting api weather condition data
```



```
    temp=random.randint(-20,125)#geneating ranom values for temperature
    hum=random.randint(0,100)#geneating ranom values for humidity
    soilmoisture=random.randint(0,1023)#analog sensor
    sm_percentage=(soilmoisture/1023)*100
    sm_percentage=int(sm_percentage)#geneating ranom values for soilmoisture
    myData={'temperature':temp, 'humidity':hum,'soilmoisture':sm_percentage,'status':status,'api_temperature':api_temperature,'api_pressure':api_pressure,'ap
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)

time.sleep(2)
client.disconnect()
```

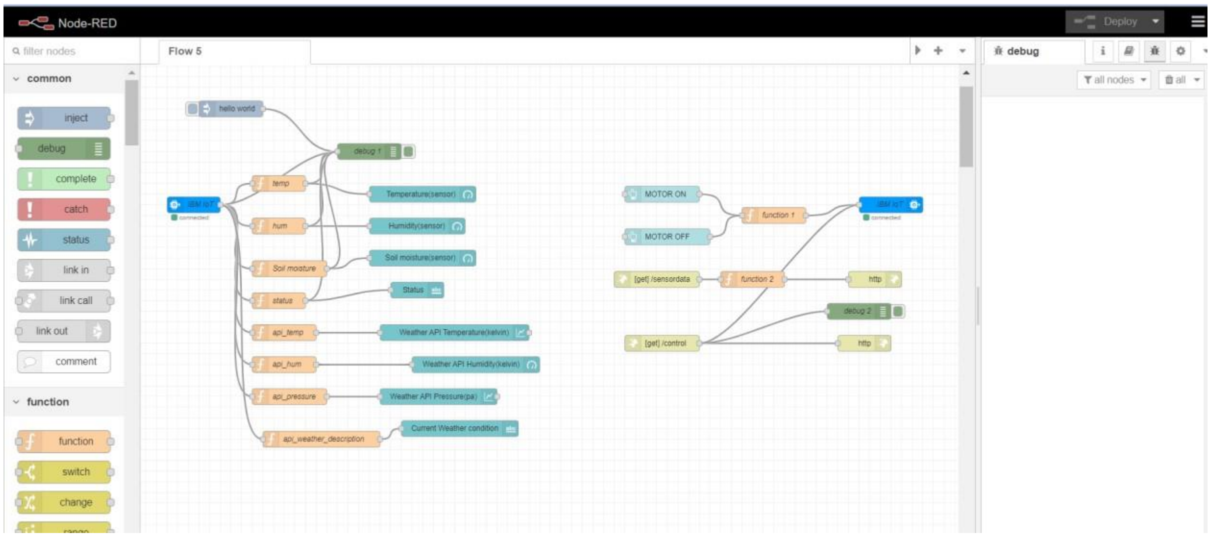
Running Module



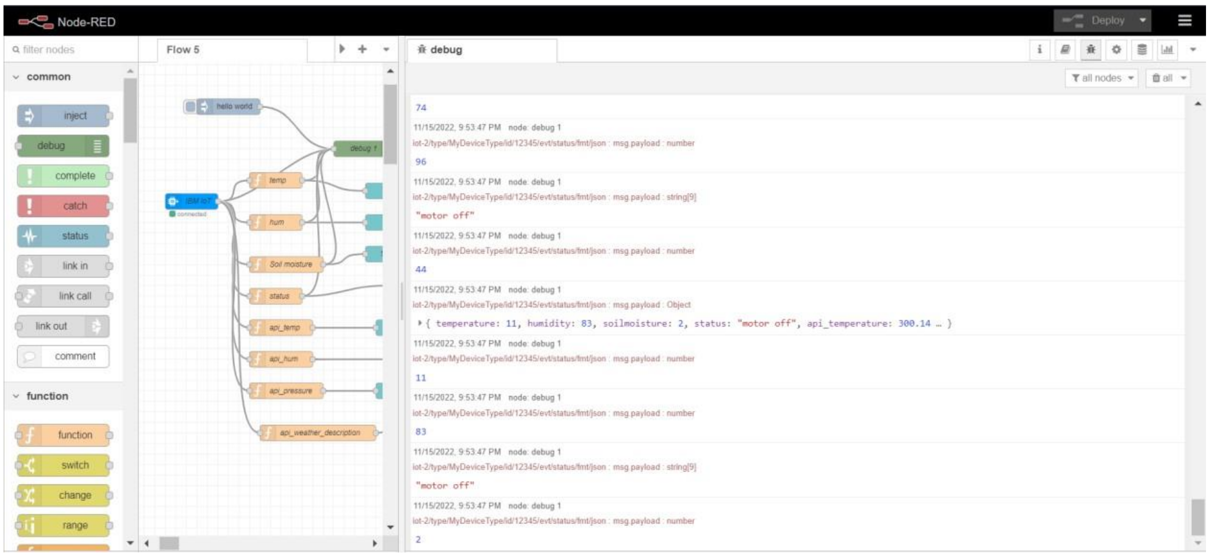
```
= RESTART: C:\Users\B.SOMESHWARAN\Desktop\IBM\Project Development Phase\sprint -1\python code with cmments.py
2022-11-15 21:26:16,286   wiotp.sdk.device.client.DeviceClient  INFO    Connected successfully: d:17lsro:MyDeviceType:12345
Published data Successfully: %s {'temperature': 60, 'humidity': 34, 'soilmoisture': 57, 'status': 'motor off', 'api_temperature': 300.14, 'api_pressure': 10
13, 'api_humidity': 83, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': 96, 'humidity': 85, 'soilmoisture': 70, 'status': 'motor off', 'api_temperature': 300.14, 'api_pressure': 10
13, 'api_humidity': 83, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': 93, 'humidity': 3, 'soilmoisture': 8, 'status': 'motor off', 'api_temperature': 300.14, 'api_pressure': 1013
, 'api_humidity': 83, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': 50, 'humidity': 23, 'soilmoisture': 60, 'status': 'motor off', 'api_temperature': 300.14, 'api_pressure': 10
13, 'api_humidity': 83, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': 76, 'humidity': 16, 'soilmoisture': 94, 'status': 'motor off', 'api_temperature': 300.14, 'api_pressure': 10
13, 'api_humidity': 83, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': 86, 'humidity': 51, 'soilmoisture': 56, 'status': 'motor off', 'api_temperature': 300.14, 'api_pressure': 10
13, 'api_humidity': 83, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': -6, 'humidity': 27, 'soilmoisture': 22, 'status': 'motor off', 'api_temperature': 300.14, 'api_pressure': 10
13, 'api_humidity': 83, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': 59, 'humidity': 62, 'soilmoisture': 13, 'status': 'motor off', 'api_temperature': 300.14, 'api_pressure': 10
13, 'api_humidity': 83, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': 12, 'humidity': 4, 'soilmoisture': 81, 'status': 'motor off', 'api_temperature': 300.14, 'api_pressure': 101
3, 'api_humidity': 83, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': 74, 'humidity': 89, 'soilmoisture': 50, 'status': 'motor off', 'api_temperature': 300.14, 'api_pressure': 10
13, 'api_humidity': 83, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': -1, 'humidity': 14, 'soilmoisture': 77, 'status': 'motor off', 'api_temperature': 300.14, 'api_pressure': 10
13, 'api_humidity': 83, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': 18, 'humidity': 66, 'soilmoisture': 81, 'status': 'motor off', 'api_temperature': 300.14, 'api_pressure': 10
13, 'api_humidity': 83, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': 98, 'humidity': 15, 'soilmoisture': 100, 'status': 'motor off', 'api_temperature': 300.14, 'api_pressure': 1
013, 'api_humidity': 83, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': -11, 'humidity': 17, 'soilmoisture': 96, 'status': 'motor off', 'api_temperature': 300.14, 'api_pressure': 1
013, 'api_humidity': 83, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': 102, 'humidity': 87, 'soilmoisture': 47, 'status': 'motor off', 'api_temperature': 300.14, 'api_pressure': 1
013, 'api_humidity': 83, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': 49, 'humidity': 57, 'soilmoisture': 47, 'status': 'motor off', 'api_temperature': 300.14, 'api_pressure': 10
13, 'api_humidity': 83, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': 15, 'humidity': 3, 'soilmoisture': 84, 'status': 'motor off', 'api_temperature': 300.14, 'api_pressure': 101
3, 'api_humidity': 83, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': 48, 'humidity': 96, 'soilmoisture': 49, 'status': 'motor off', 'api_temperature': 300.14, 'api_pressure': 10
13, 'api_humidity': 83, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': 103, 'humidity': 3, 'soilmoisture': 84, 'status': 'motor off', 'api_temperature': 300.14, 'api_pressure': 10
13, 'api_humidity': 83, 'api_weather_description': 'mist'}
```

NODE RED Flow Connections

- **Interfacing IBM Cloud**
- **Intefacing & Getting Sensor Datas**
- **Connecting MIT App Inventor**
- **Weather Map Parameters**

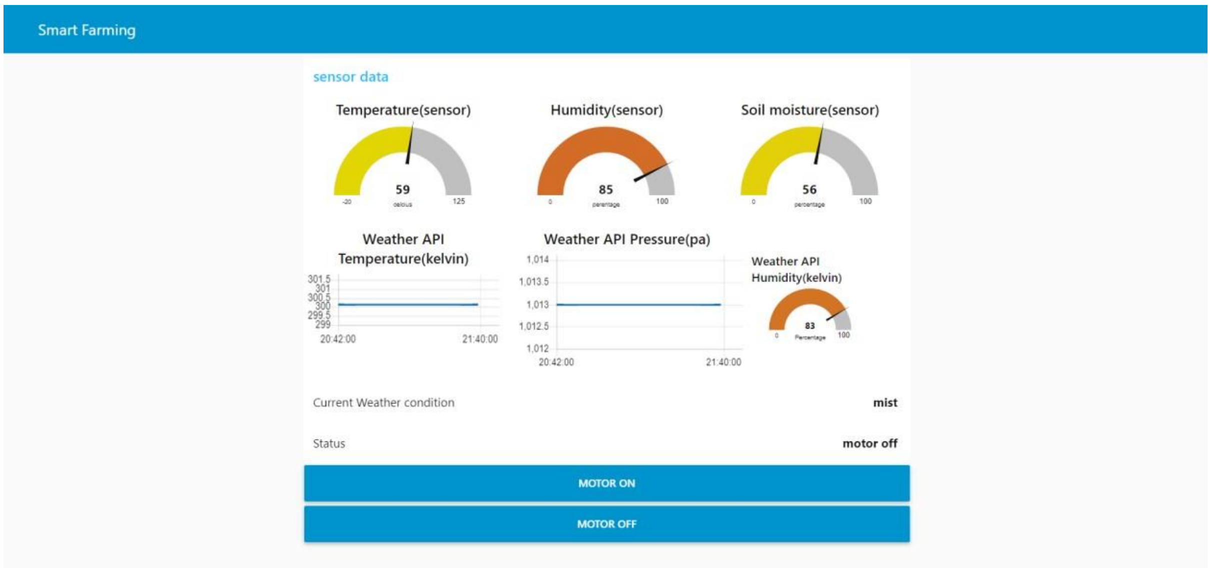


Live Publish Data Output Of Node Red



```
74
11/15/2022, 9:53:47 PM  node: debug 1
iot-2/type/MyDeviceType/id/12345/evt/status/fmt/json : msg.payload : number
96
11/15/2022, 9:53:47 PM  node: debug 1
iot-2/type/MyDeviceType/id/12345/evt/status/fmt/json : msg.payload : string[9]
"motor off"
11/15/2022, 9:53:47 PM  node: debug 1
iot-2/type/MyDeviceType/id/12345/evt/status/fmt/json : msg.payload : number
44
11/15/2022, 9:53:47 PM  node: debug 1
iot-2/type/MyDeviceType/id/12345/evt/status/fmt/json : msg.payload : Object
▸{ temperature: 11, humidity: 83, soilmoisture: 2, status: "motor off", api_temperature: 300.14 … }
11/15/2022, 9:53:47 PM  node: debug 1
iot-2/type/MyDeviceType/id/12345/evt/status/fmt/json : msg.payload : number
11
11/15/2022, 9:53:47 PM  node: debug 1
iot-2/type/MyDeviceType/id/12345/evt/status/fmt/json : msg.payload : number
83
11/15/2022, 9:53:47 PM  node: debug 1
iot-2/type/MyDeviceType/id/12345/evt/status/fmt/json : msg.payload : string[9]
"motor off"
11/15/2022, 9:53:47 PM  node: debug 1
iot-2/type/MyDeviceType/id/12345/evt/status/fmt/json : msg.payload : number
2
```

Web API Output


Smart Farming

sensor data

Temperature(sensor) 59 celcius (-20 – 125)

Humidity(sensor) 85 perentage (0 – 100)

Soil moisture(sensor) 56 percentage (0 – 100)

Weather API Temperature(kelvin)

Weather API Pressure(pa)

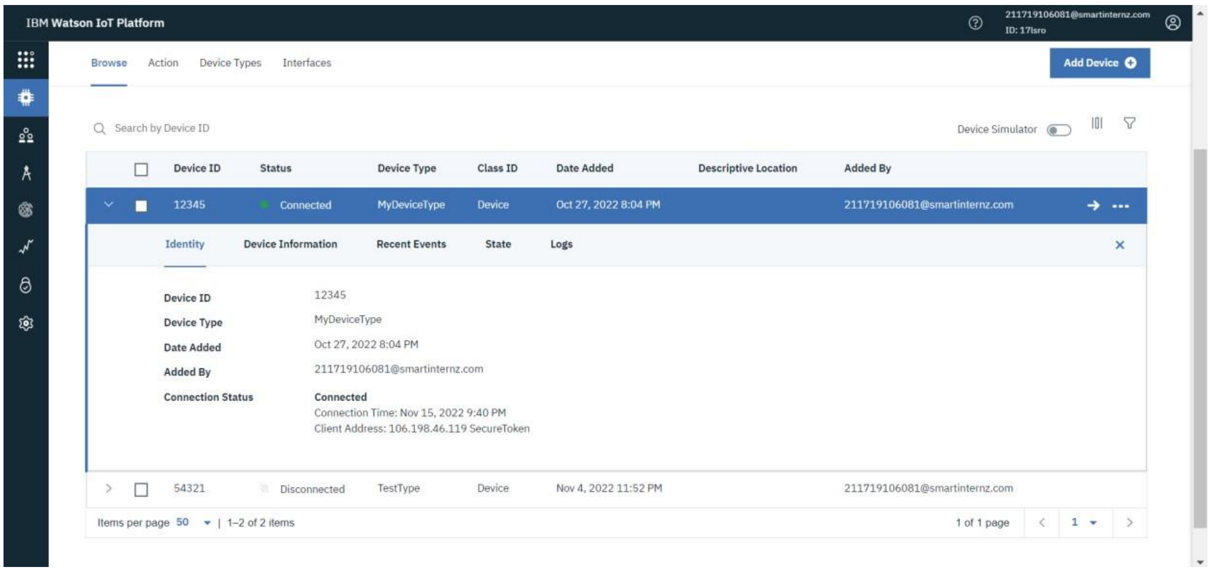
Weather API Humidity(kelvin) 83 Percentage (0 – 100)

Current Weather condition mist

Status motor off

IBM Watson IoT Platform

- **Device Connected Details**


IBM Watson IoT Platform

211719106081@smartinternz.com
ID: 17lsro

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location	Added By
12345	Connected	MyDeviceType	Device	Oct 27, 2022 8:04 PM		211719106081@smartinternz.com
54321	Disconnected	TestType	Device	Nov 4, 2022 11:52 PM		211719106081@smartinternz.com

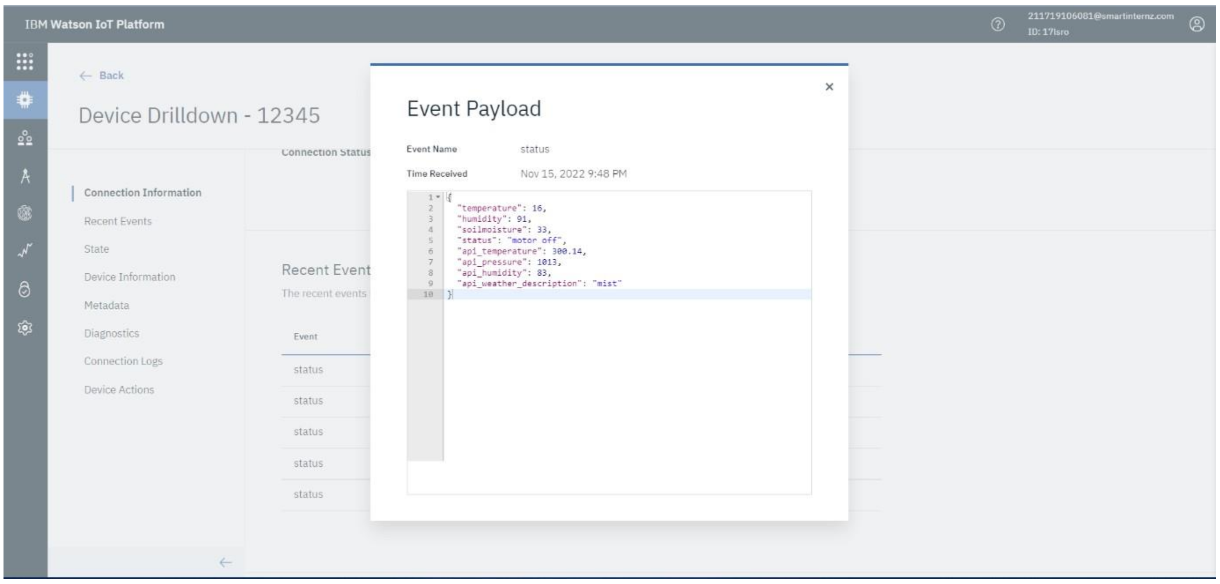
Device ID	12345
Device Type	MyDeviceType
Date Added	Oct 27, 2022 8:04 PM
Added By	211719106081@smartinternz.com
Connection Status	Connected Connection Time: Nov 15, 2022 9:40 PM Client Address: 106.198.46.119 SecureToken

Items per page 50 | 1–2 of 2 items

1 of 1 page 1

Live Date Output Of IBM Watson Iot Platform

- **Sensor Output Data**
- **Weather Condition**
- **Weather Map Parameters In Current Location**



Event Payload

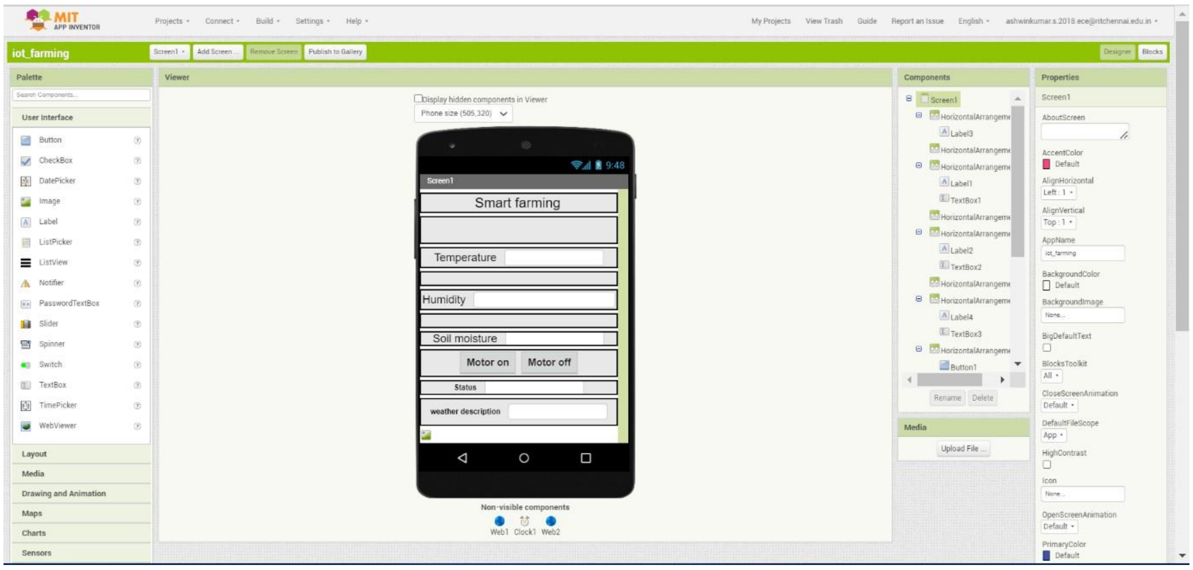
Event Name status

Time Received Nov 15, 2022 9:48 PM

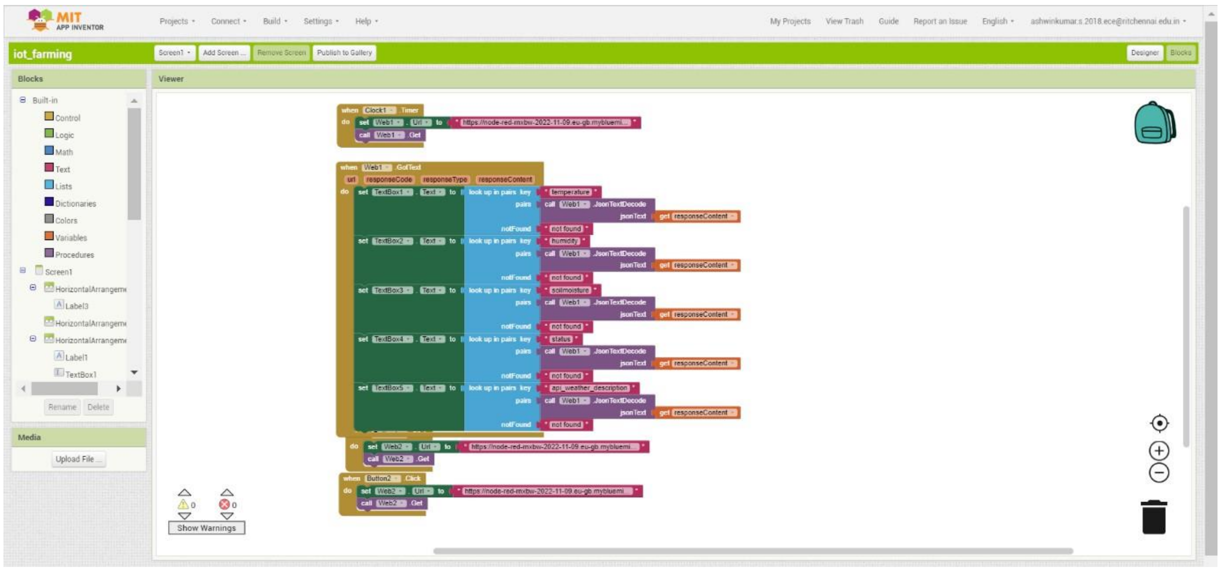
```
{
  "temperature": 16,
  "humidity": 91,
  "soilmoisture": 33,
  "status": "motor off",
  "api_temperature": 300.14,
  "api_pressure": 1013,
  "api_humidity": 83,
  "api_weather_description": "mist"
}
```

MIT APP INVENTOR

- **Design**

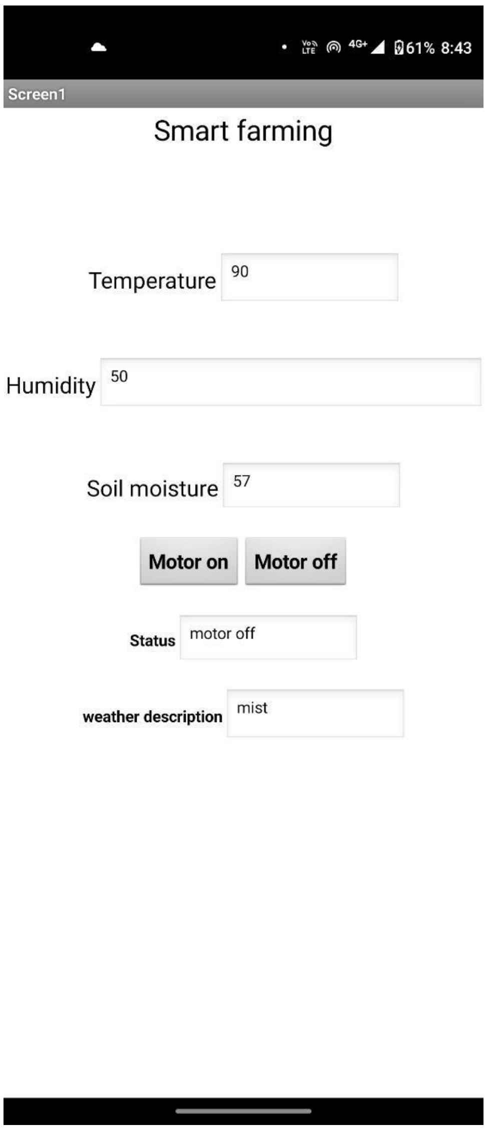


Back End Process(Block)



```
when Clock1 .Timer
do  set Web1 . Url to "https://node-red-mxbw-2022-11-09.eu-gb.mybluemi..."
    call Web1 .Get

when Web1 .GotText
 url  responseCode  responseType  responseContent
do  set TextBox1 . Text to  look up in pairs  key "temperature"
                                              pairs call Web1 .JsonTextDecode
                                                         jsonText get responseContent
                                              notFound "not found"
    set TextBox2 . Text to  look up in pairs  key "humidity"
                                              pairs call Web1 .JsonTextDecode
                                                         jsonText get responseContent
                                              notFound "not found"
    set TextBox3 . Text to  look up in pairs  key "soilmoisture"
                                              pairs call Web1 .JsonTextDecode
                                                         jsonText get responseContent
                                              notFound "not found"
    set TextBox4 . Text to  look up in pairs  key "status"
                                              pairs call Web1 .JsonTextDecode
                                                         jsonText get responseContent
                                              notFound "not found"
    set TextBox5 . Text to  look up in pairs  key "api_weather_description"
                                              pairs call Web1 .JsonTextDecode
                                                         jsonText get responseContent
                                              notFound "not found"

do  set Web2 . Url to "https://node-red-mxbw-2022-11-09.eu-gb.mybluemi..."
    call Web2 .Get

when Button2 .Click
do  set Web2 . Url to "https://node-red-mxbw-2022-11-09.eu-gb.mybluemi..."
    call Web2 .Get
```

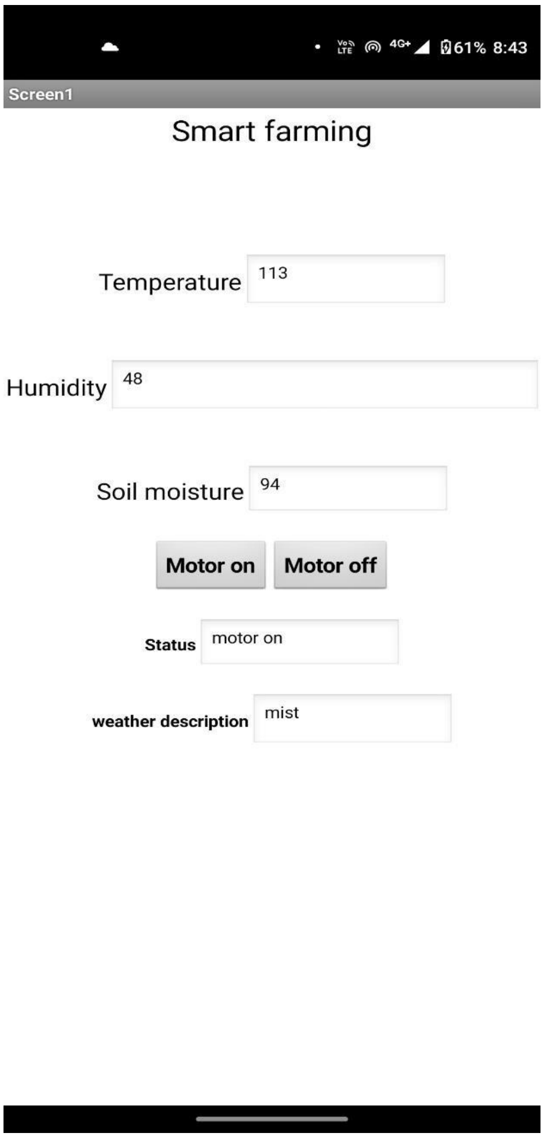
Mobile Application Ouput



Screen1

Smart farming

Temperature	90
Humidity	50
Soil moisture	57
Status	motor off
weather description	mist



Screen1

Smart farming

Temperature	113
Humidity	48
Soil moisture	94
Status	motor on
weather description	mist