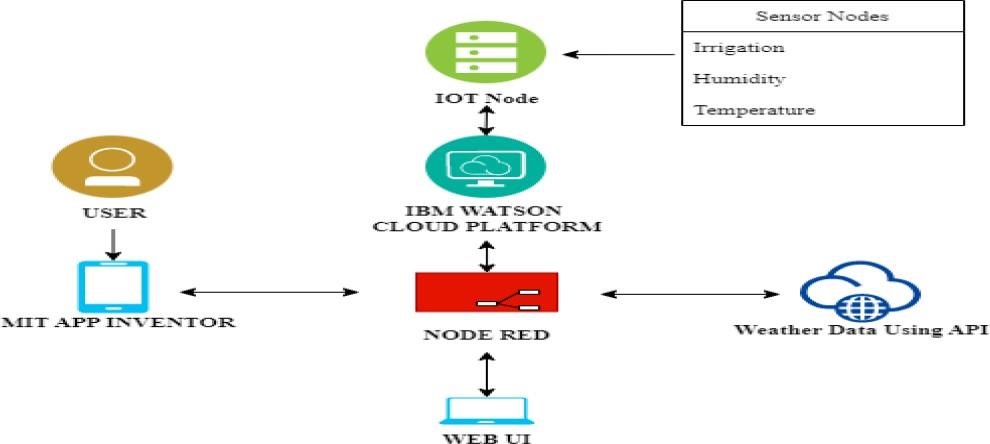
**Project Development**

**Sprint - 4**

|  |  |
| --- | --- |
| **Team ID** | PNT2022TMID02439 |
| **Project Name** | Project -Smart farmer-IOT enabled smart Farming Application |
| **Team Members** | Kamesh Kumar K, Kaviyaraj C, Javeed Akram Z , KavinSaran JK |

**Flow Diagram**



**Python Code:**

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

**Python Code:**

import wiotp.sdk.device import time import os import datetime import random

myconfig = {

"identity": {

"orgId": "ga4sjl",

"typeId": "NodeMCU",

"deviceId": "12345"

},

"auth": {

"token": "CK2!+2FzgnyZFWE9yW"

}

}

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"):

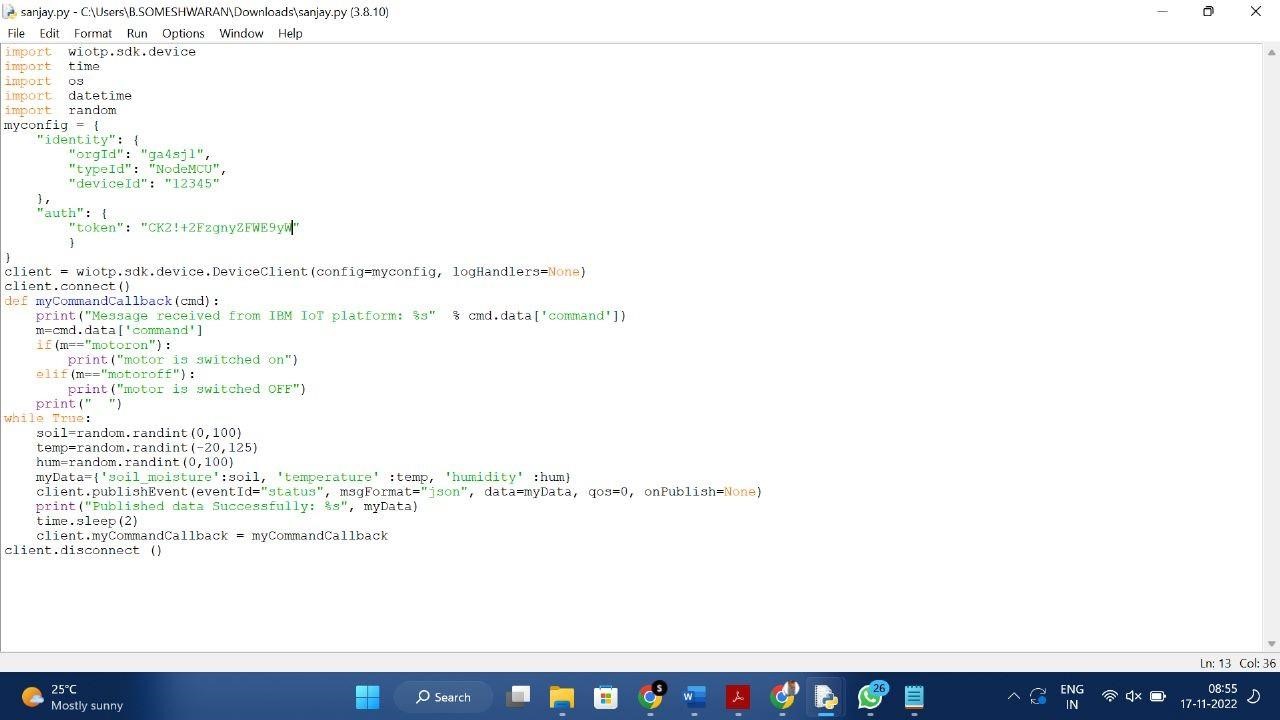
print("motor is switched OFF") print(" ") while True:

soil=random.randint(0,100) temp=random.randint(-20,125) hum=random.randint(0,100)

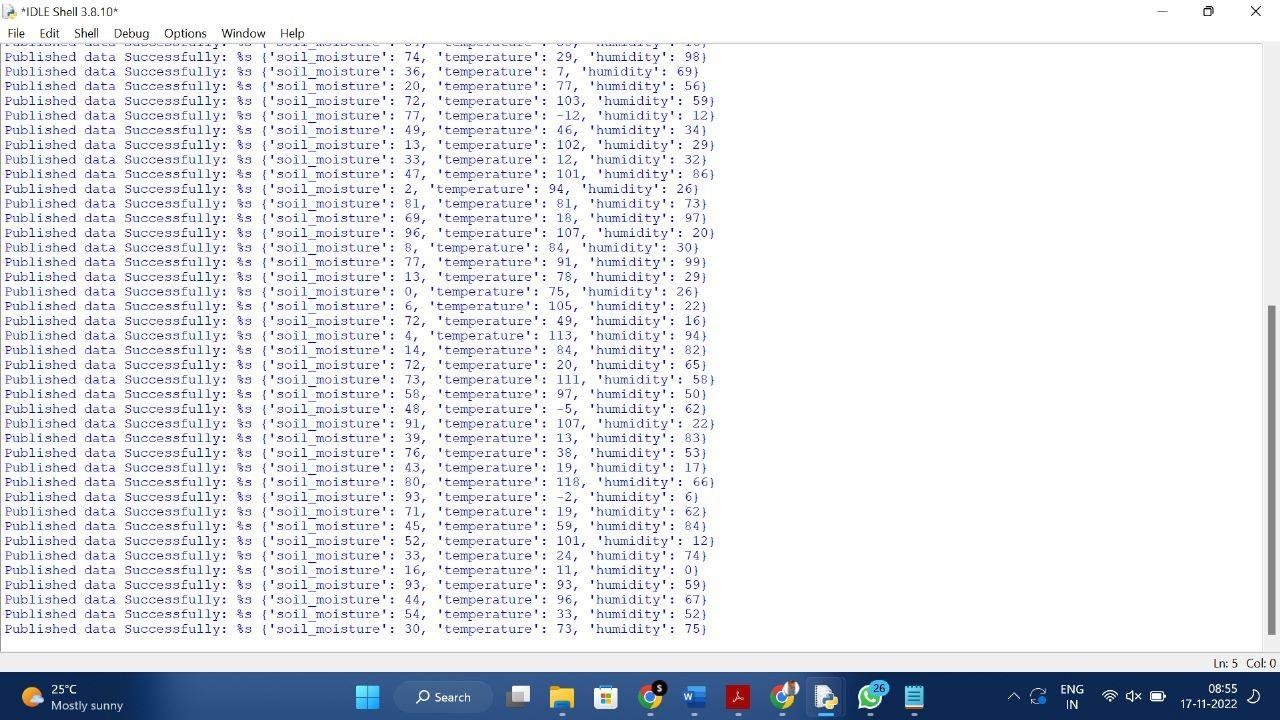
myData={'soilmoisture':soil, 'temperature' :temp, 'humidity' :hum} client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)

print("Published data Successfully: %s", myData) time.sleep(2)

client.myCommandCallback = myCommandCallback client.disconnect ()



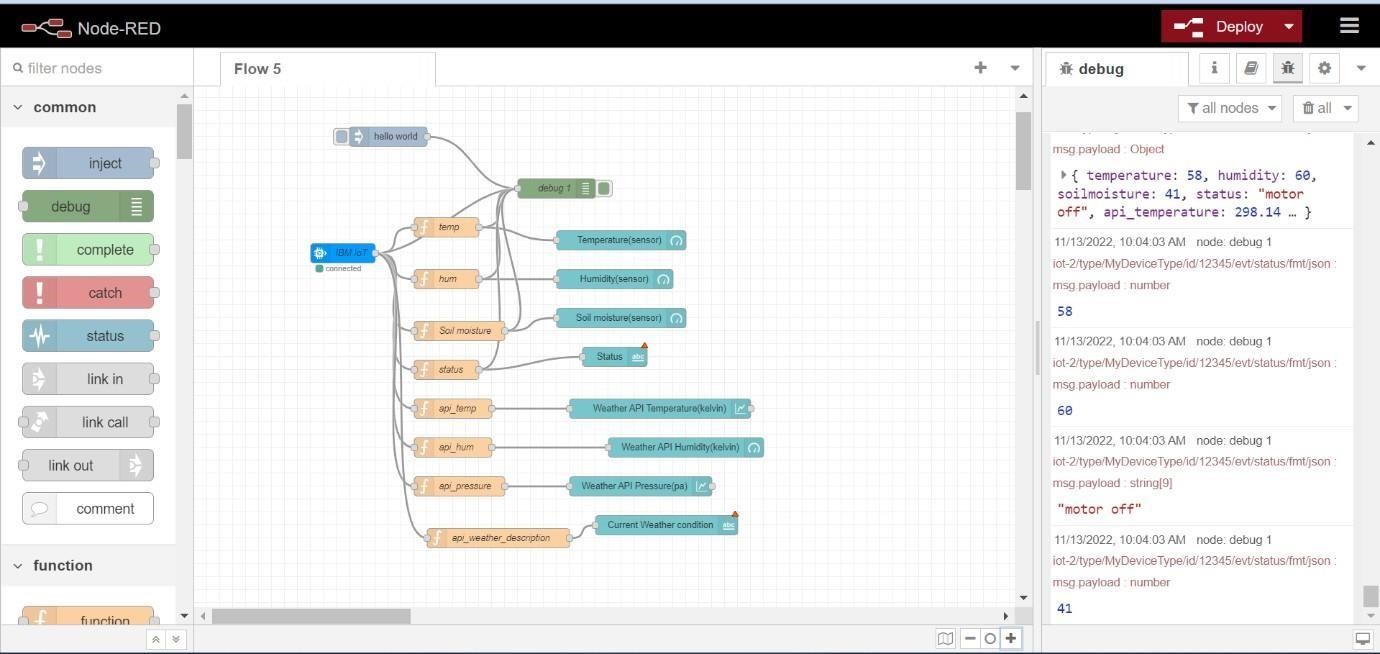
Running of programs :



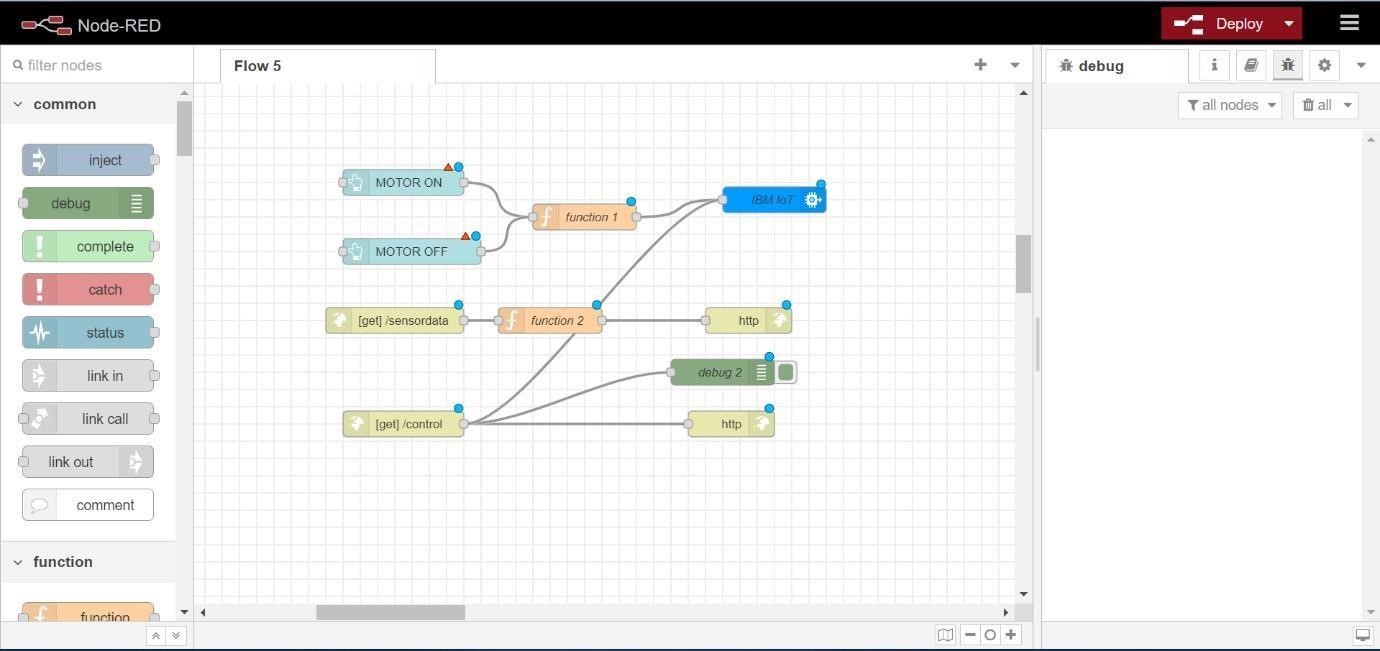
## NODE RED Flow Connections  Interfacing IBM Cloud

* **Intefacing & Getting Sensor Datas**
* **Connecting MIT App Inventor**
* **Weather Map Parameters**

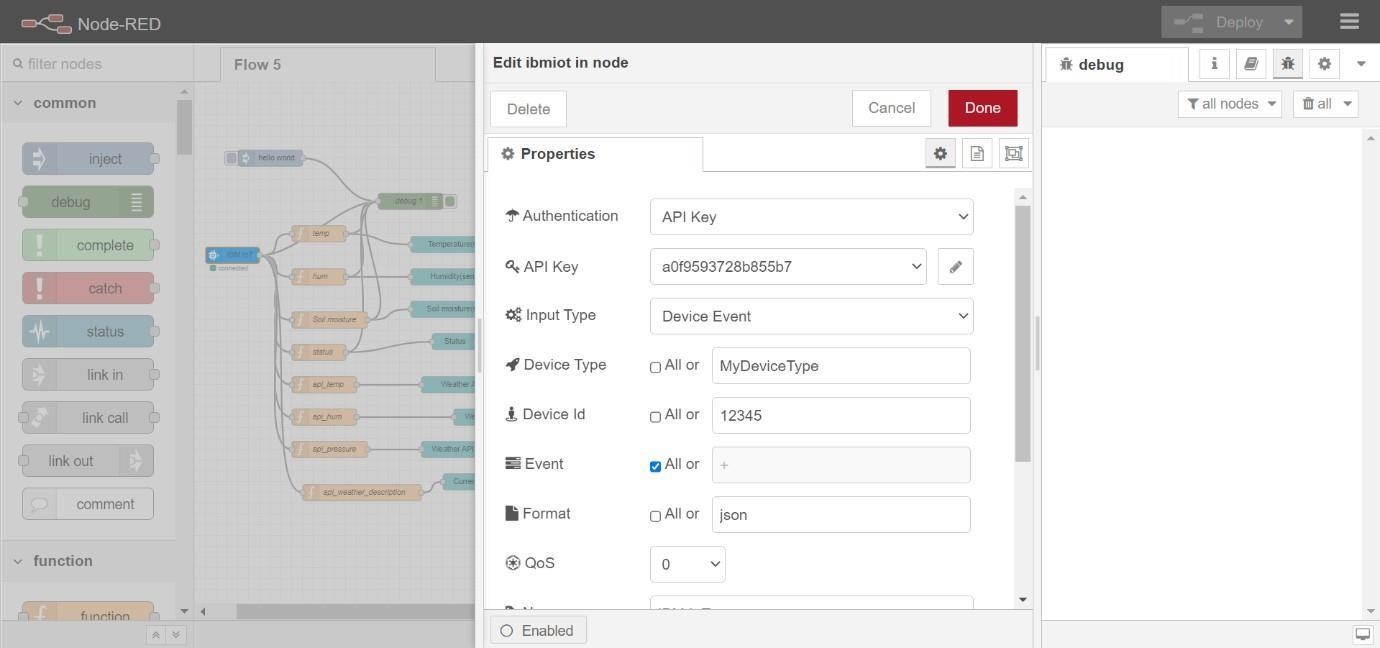
## Flow:1



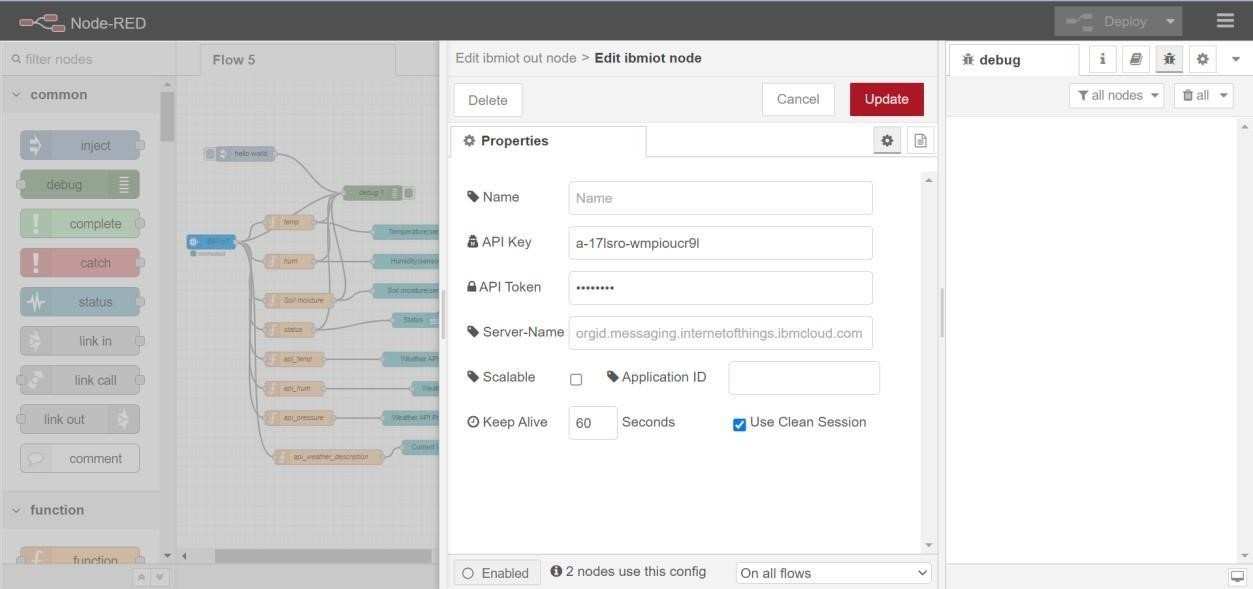
## Flow:2



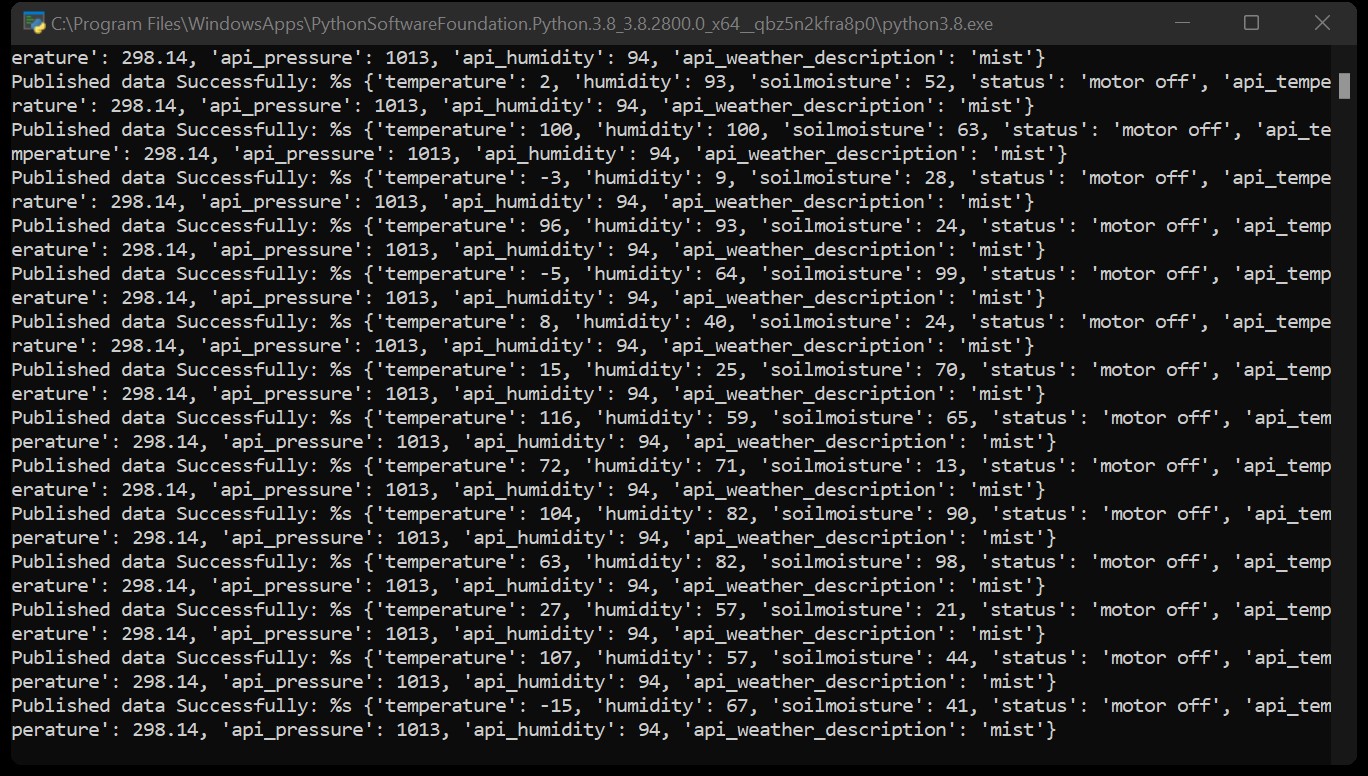
## Flow:1 Configuring All Nodes With IBM IOT Platform



## Flow:2 Configuring All Nodes With IBM IOT Platform



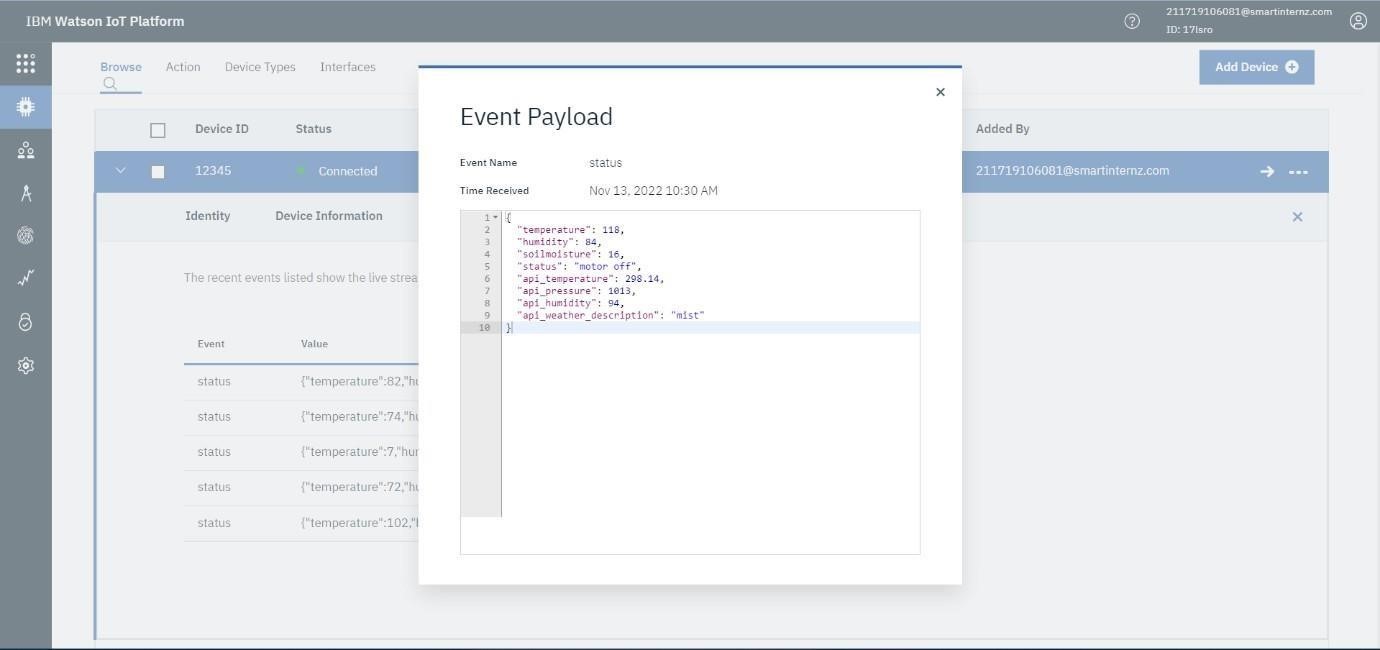
## Execution of Python Program



## Web UI Output

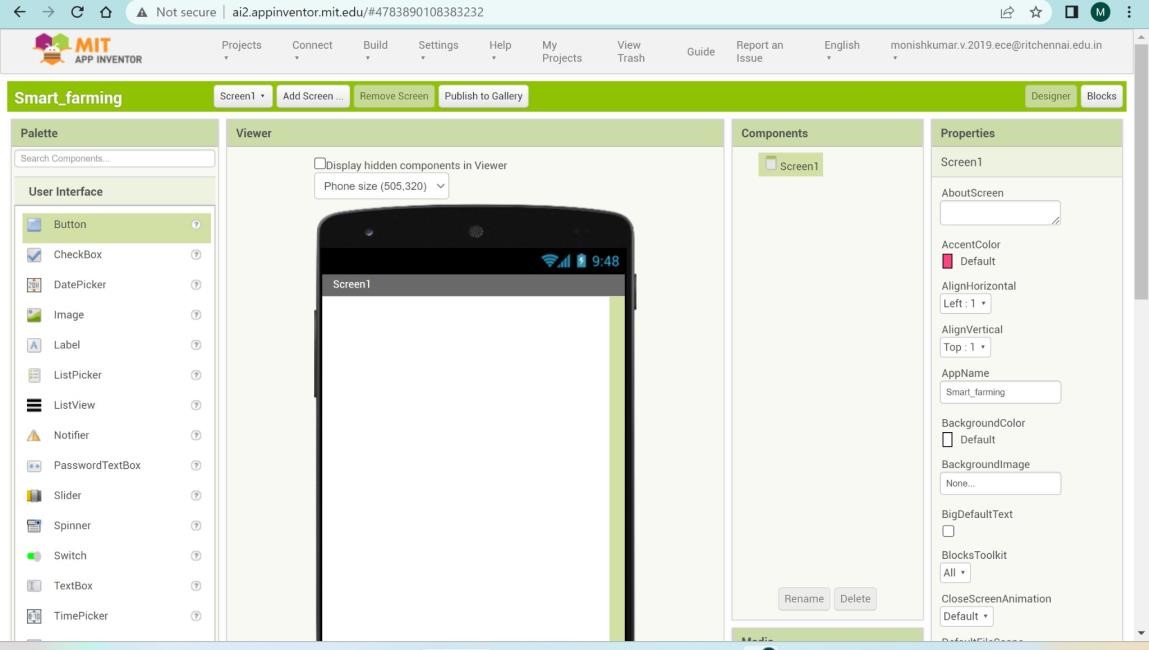


## IBM Watson IoT Platform Device Connect & Live Data

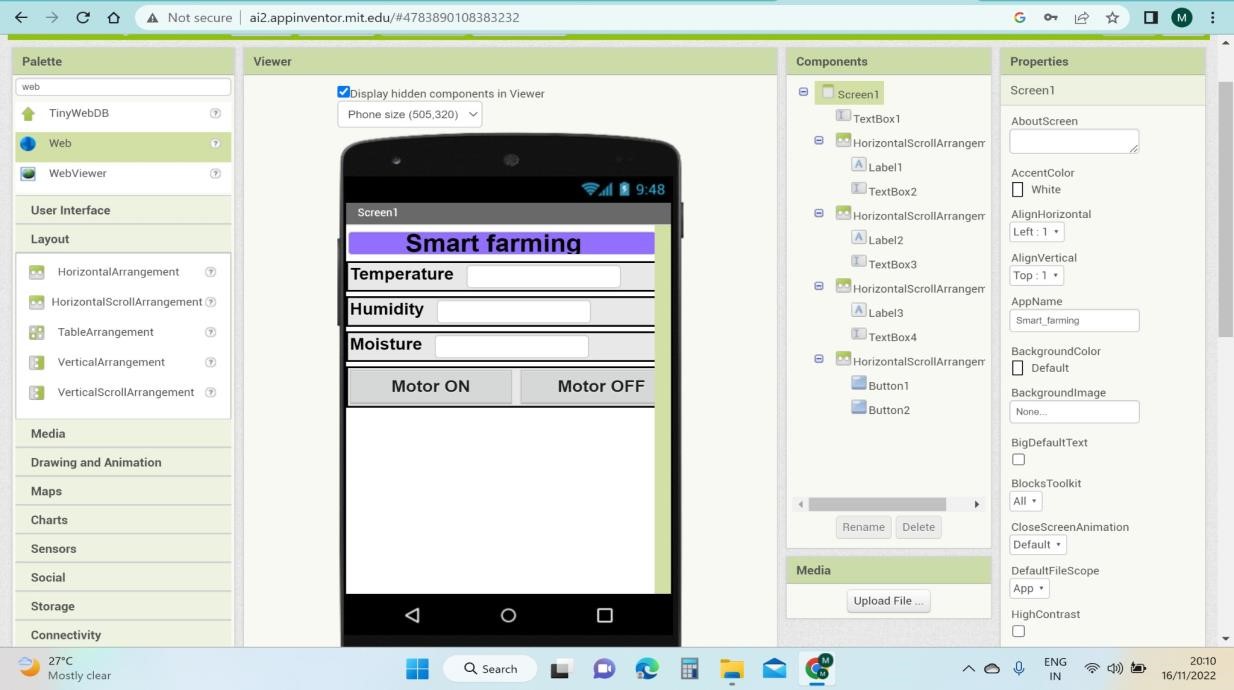


**MIT APP INVENTOR**

**Step 1:** Login Into MIT App Inventor

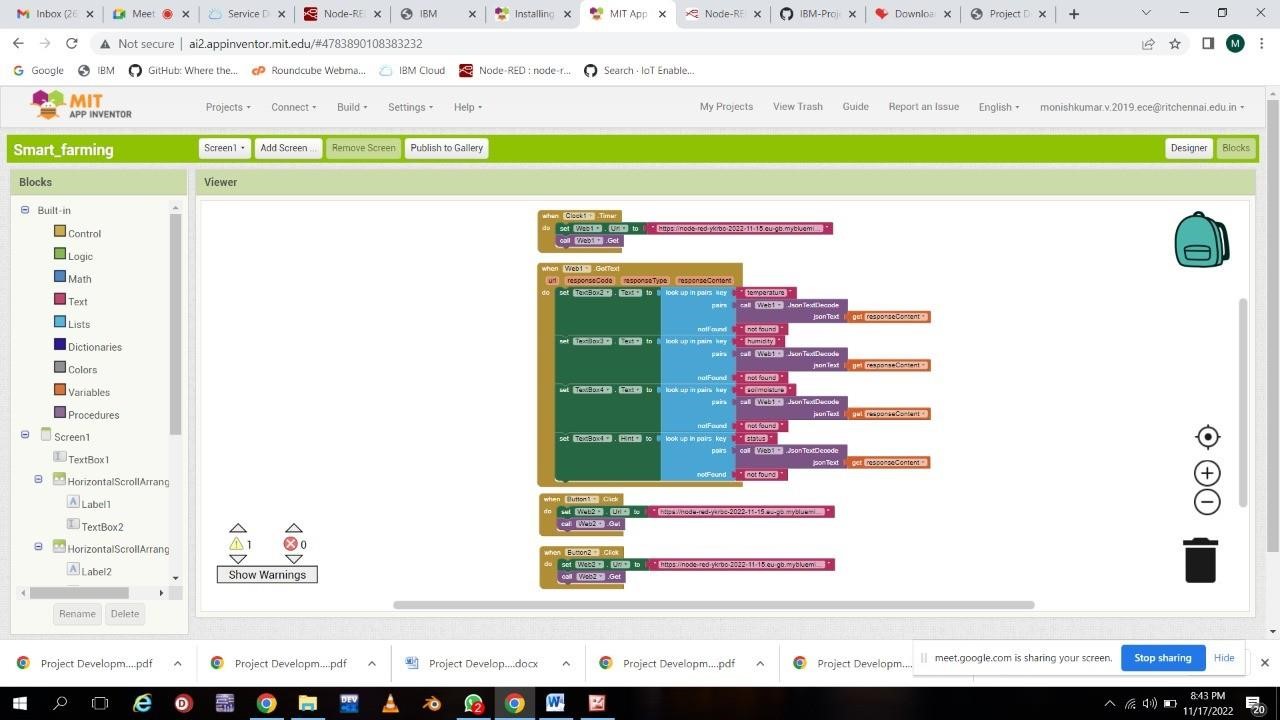


**Step 2:** Create Your User Interface By Using the Preset Tools



**Step 3:** Back End Process

* Specify the Cloud URL Details to Receive the Date From Node Red.
* Commend Request From App To Node RED to Turn ON /OFF Motor.
* Weather API Data is Displaced From Node RED.



**Step 4:** Live Output In Mobile Application

