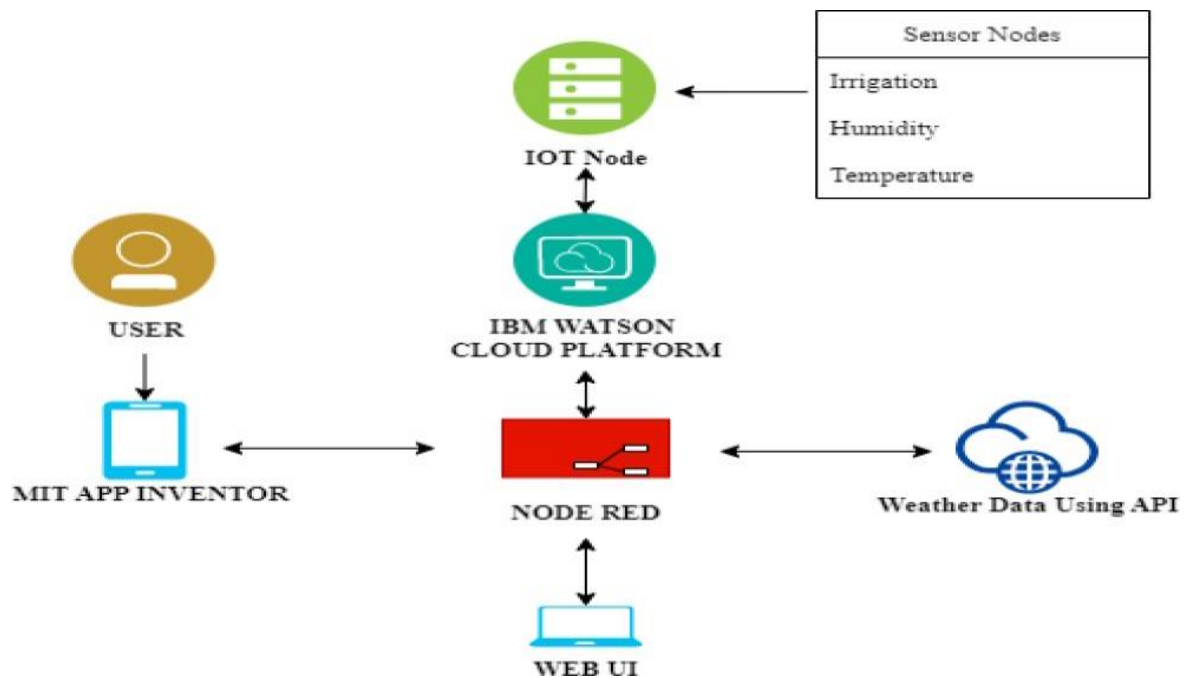


Project Development

Delivery of Sprint - 4

Team ID	PNT2022TMID26519
Team Members	Monish Kumar V, Mohana Priya K, Swetha G, Sanjay Kumar V
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

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 ()
```

```
sanjay.py - C:\Users\B.SOMESHWARAN\Downloads\sanjay.py (3.8.10)
File Edit Format Run Options Window Help

import wiotp.sdk.device
import time
import os
import datetime
import random
myconfig = {
    "identity": {
        "orgId": "ga4sji1",
        "typeId": "NodeMCU",
        "deviceId": "12345"
    },
    "auth": {
        "token": "CK2i+2FzgnY2FWE9y6"
    }
}
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, '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 :

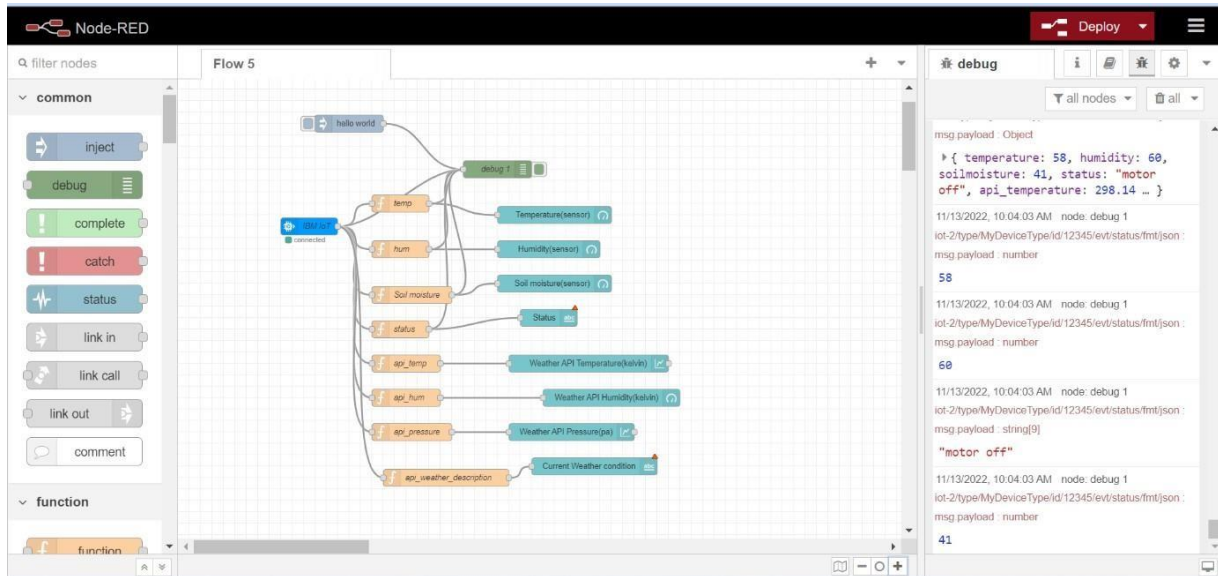
```
*IDLE Shell 3.8.10*
File Edit Shell Debug Options Window Help

Published data Successfully: %s ('soil_moisture': 74, 'temperature': 29, 'humidity': 98)
Published data Successfully: %s ('soil_moisture': 36, 'temperature': 7, 'humidity': 69)
Published data Successfully: %s ('soil_moisture': 20, 'temperature': 77, 'humidity': 56)
Published data Successfully: %s ('soil_moisture': 72, 'temperature': 103, 'humidity': 59)
Published data Successfully: %s ('soil_moisture': 77, 'temperature': -12, 'humidity': 12)
Published data Successfully: %s ('soil_moisture': 49, 'temperature': 46, 'humidity': 34)
Published data Successfully: %s ('soil_moisture': 13, 'temperature': 102, 'humidity': 29)
Published data Successfully: %s ('soil_moisture': 33, 'temperature': 12, 'humidity': 32)
Published data Successfully: %s ('soil_moisture': 47, 'temperature': 101, 'humidity': 86)
Published data Successfully: %s ('soil_moisture': 2, 'temperature': 94, 'humidity': 26)
Published data Successfully: %s ('soil_moisture': 81, 'temperature': 81, 'humidity': 73)
Published data Successfully: %s ('soil_moisture': 69, 'temperature': 18, 'humidity': 97)
Published data Successfully: %s ('soil_moisture': 96, 'temperature': 107, 'humidity': 20)
Published data Successfully: %s ('soil_moisture': 8, 'temperature': 84, 'humidity': 30)
Published data Successfully: %s ('soil_moisture': 77, 'temperature': 91, 'humidity': 99)
Published data Successfully: %s ('soil_moisture': 13, 'temperature': 78, 'humidity': 29)
Published data Successfully: %s ('soil_moisture': 0, 'temperature': 75, 'humidity': 26)
Published data Successfully: %s ('soil_moisture': 6, 'temperature': 105, 'humidity': 22)
Published data Successfully: %s ('soil_moisture': 72, 'temperature': 49, 'humidity': 16)
Published data Successfully: %s ('soil_moisture': 4, 'temperature': 113, 'humidity': 94)
Published data Successfully: %s ('soil_moisture': 14, 'temperature': 84, 'humidity': 82)
Published data Successfully: %s ('soil_moisture': 72, 'temperature': 20, 'humidity': 65)
Published data Successfully: %s ('soil_moisture': 73, 'temperature': 111, 'humidity': 58)
Published data Successfully: %s ('soil_moisture': 58, 'temperature': 97, 'humidity': 50)
Published data Successfully: %s ('soil_moisture': 48, 'temperature': -5, 'humidity': 62)
Published data Successfully: %s ('soil_moisture': 91, 'temperature': 107, 'humidity': 22)
Published data Successfully: %s ('soil_moisture': 39, 'temperature': 13, 'humidity': 83)
Published data Successfully: %s ('soil_moisture': 76, 'temperature': 38, 'humidity': 53)
Published data Successfully: %s ('soil_moisture': 43, 'temperature': 19, 'humidity': 17)
Published data Successfully: %s ('soil_moisture': 80, 'temperature': 118, 'humidity': 66)
Published data Successfully: %s ('soil_moisture': 93, 'temperature': -2, 'humidity': 6)
Published data Successfully: %s ('soil_moisture': 71, 'temperature': 19, 'humidity': 62)
Published data Successfully: %s ('soil_moisture': 45, 'temperature': 59, 'humidity': 84)
Published data Successfully: %s ('soil_moisture': 52, 'temperature': 101, 'humidity': 12)
Published data Successfully: %s ('soil_moisture': 33, 'temperature': 24, 'humidity': 74)
Published data Successfully: %s ('soil_moisture': 16, 'temperature': 11, 'humidity': 0)
Published data Successfully: %s ('soil_moisture': 93, 'temperature': 93, 'humidity': 59)
Published data Successfully: %s ('soil_moisture': 44, 'temperature': 96, 'humidity': 67)
Published data Successfully: %s ('soil_moisture': 54, 'temperature': 33, 'humidity': 52)
Published data Successfully: %s ('soil_moisture': 30, 'temperature': 73, 'humidity': 75)
```

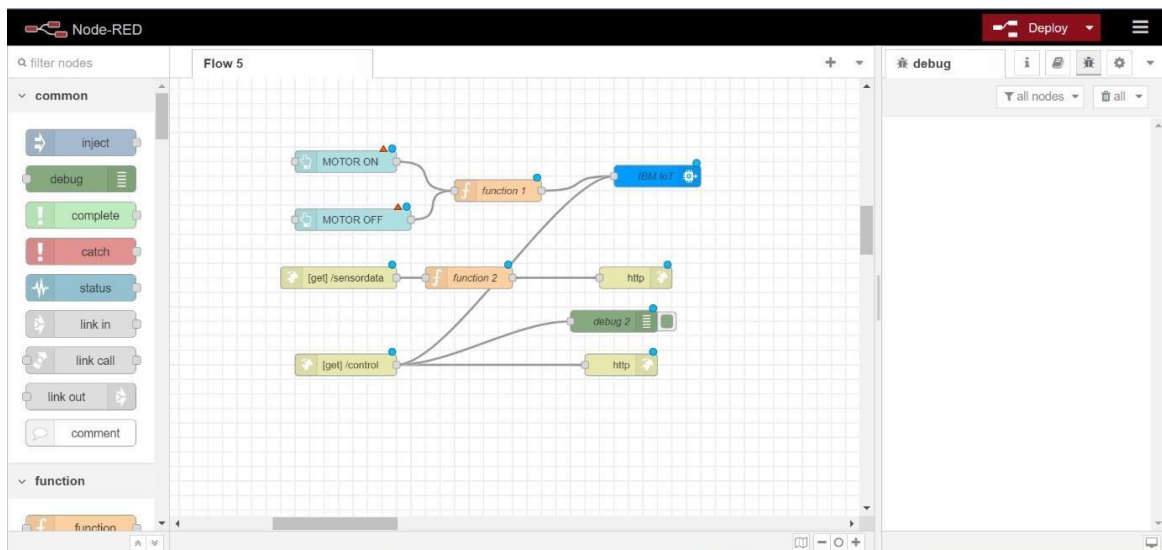
NODE RED Flow Connections

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

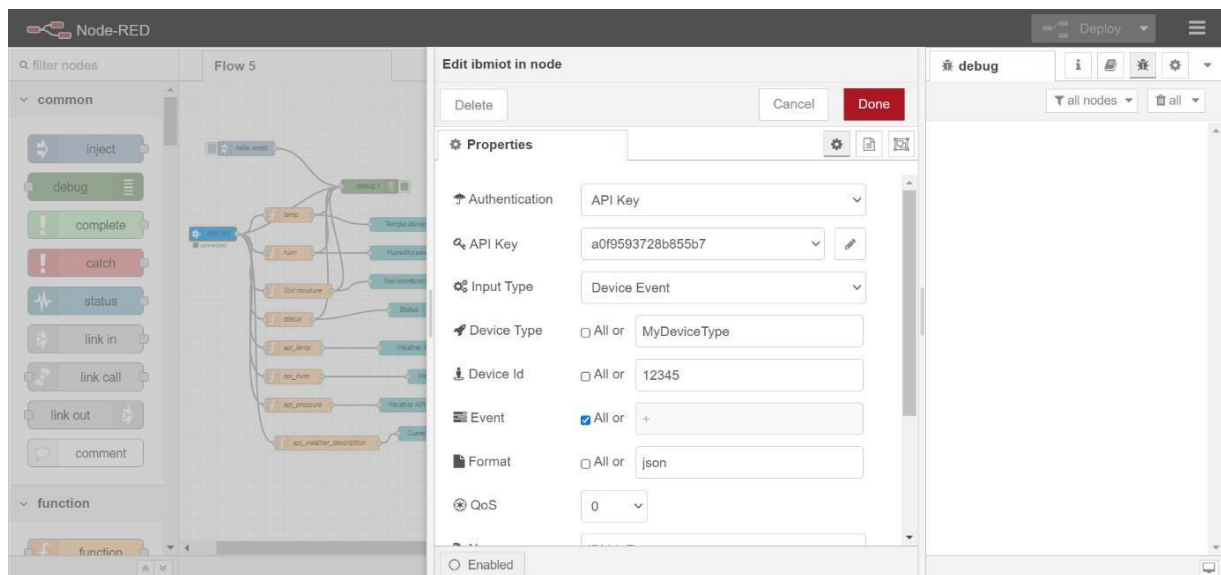
Flow:1



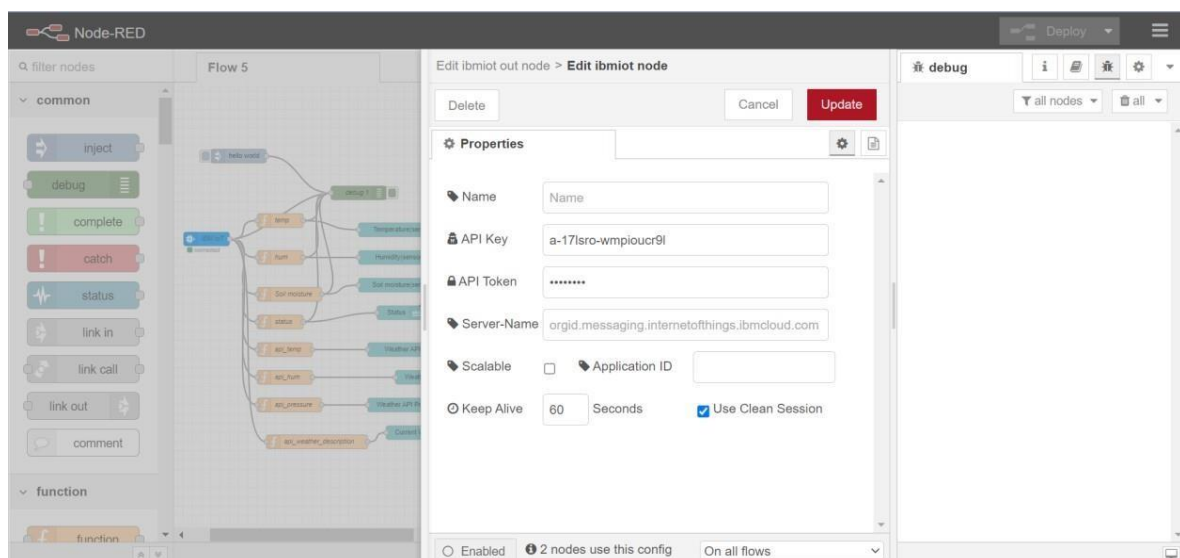
Flow:2



Flow:1 Configuring All Nodes With IBM IOT Platform



Flow:2 Configuring All Nodes With IBM IOT Platform



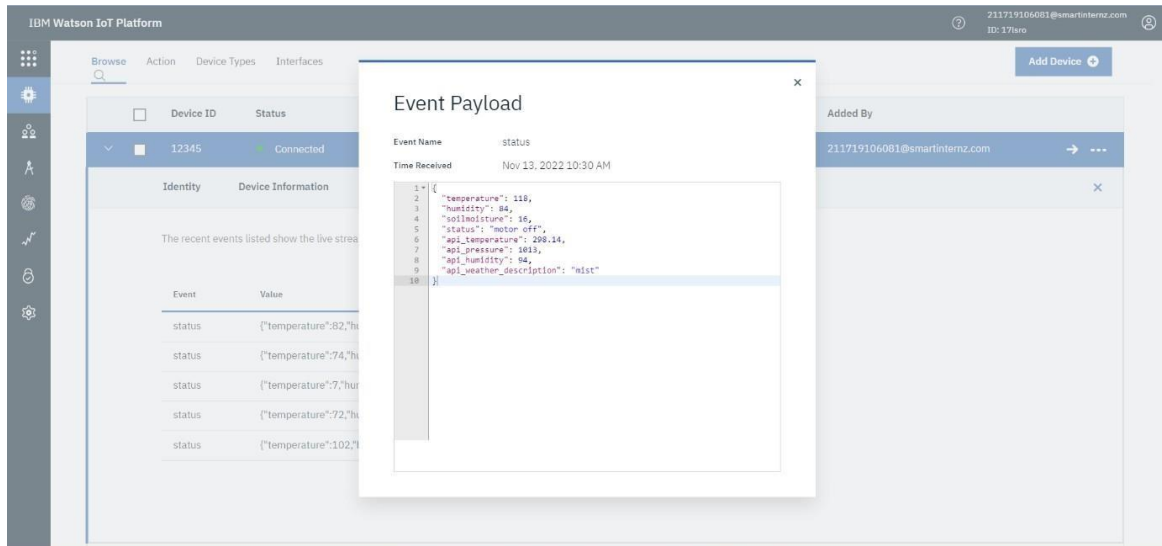
Execution of Python Program

```
C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.8_3.8.2800.0_x64_qbz5n2kfra8p0\python3.8.exe
erature': 298.14, 'api_pressure': 1013, 'api_humidity': 94, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': 2, 'humidity': 93, 'soilmoisture': 52, 'status': 'motor off', 'api_tempe
rature': 298.14, 'api_pressure': 1013, 'api_humidity': 94, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': 100, 'humidity': 100, 'soilmoisture': 63, 'status': 'motor off', 'api_te
mperature': 298.14, 'api_pressure': 1013, 'api_humidity': 94, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': -3, 'humidity': 9, 'soilmoisture': 28, 'status': 'motor off', 'api_tempe
rature': 298.14, 'api_pressure': 1013, 'api_humidity': 94, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': 96, 'humidity': 93, 'soilmoisture': 24, 'status': 'motor off', 'api_temp
erature': 298.14, 'api_pressure': 1013, 'api_humidity': 94, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': -5, 'humidity': 64, 'soilmoisture': 99, 'status': 'motor off', 'api_tem
perature': 298.14, 'api_pressure': 1013, 'api_humidity': 94, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': 8, 'humidity': 40, 'soilmoisture': 24, 'status': 'motor off', 'api_tempe
rature': 298.14, 'api_pressure': 1013, 'api_humidity': 94, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': 15, 'humidity': 25, 'soilmoisture': 70, 'status': 'motor off', 'api_tem
perature': 298.14, 'api_pressure': 1013, 'api_humidity': 94, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': 116, 'humidity': 59, 'soilmoisture': 65, 'status': 'motor off', 'api_tem
perature': 298.14, 'api_pressure': 1013, 'api_humidity': 94, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': 72, 'humidity': 71, 'soilmoisture': 13, 'status': 'motor off', 'api_tem
perature': 298.14, 'api_pressure': 1013, 'api_humidity': 94, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': 104, 'humidity': 82, 'soilmoisture': 90, 'status': 'motor off', 'api_tem
perature': 298.14, 'api_pressure': 1013, 'api_humidity': 94, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': 63, 'humidity': 82, 'soilmoisture': 98, 'status': 'motor off', 'api_tem
perature': 298.14, 'api_pressure': 1013, 'api_humidity': 94, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': 27, 'humidity': 57, 'soilmoisture': 21, 'status': 'motor off', 'api_tem
perature': 298.14, 'api_pressure': 1013, 'api_humidity': 94, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': 107, 'humidity': 57, 'soilmoisture': 44, 'status': 'motor off', 'api_tem
perature': 298.14, 'api_pressure': 1013, 'api_humidity': 94, 'api_weather_description': 'mist'}
Published data Successfully: %s {'temperature': -15, 'humidity': 67, 'soilmoisture': 41, 'status': 'motor off', 'api_tem
perature': 298.14, 'api_pressure': 1013, 'api_humidity': 94, 'api_weather_description': 'mist'}
```

Web UI Output

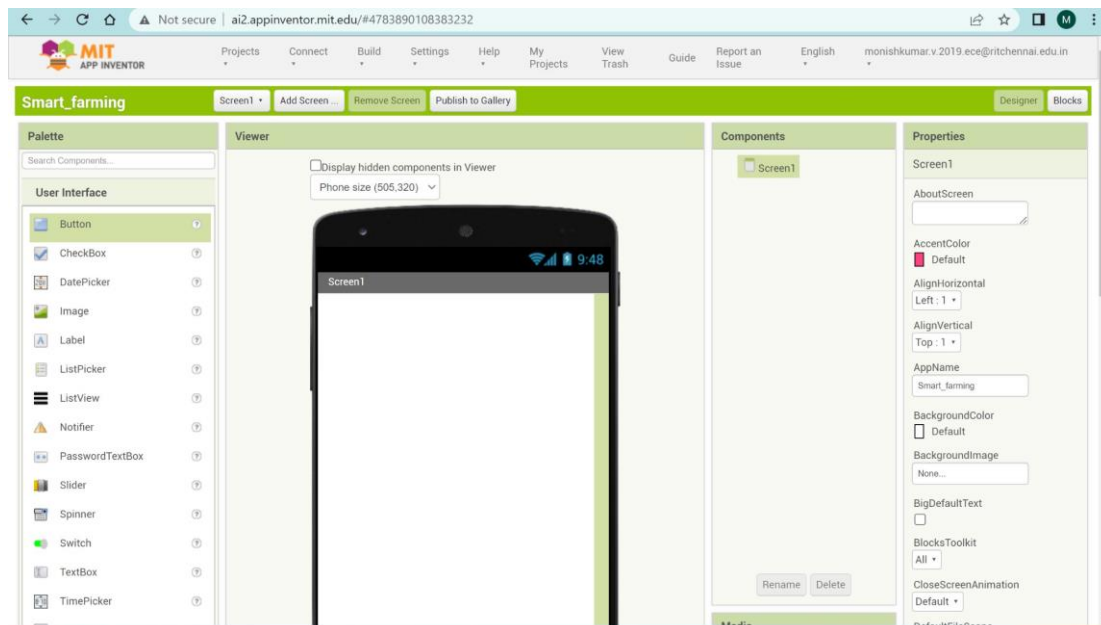


IBM Watson IoT Platform Device Connect & Live Data

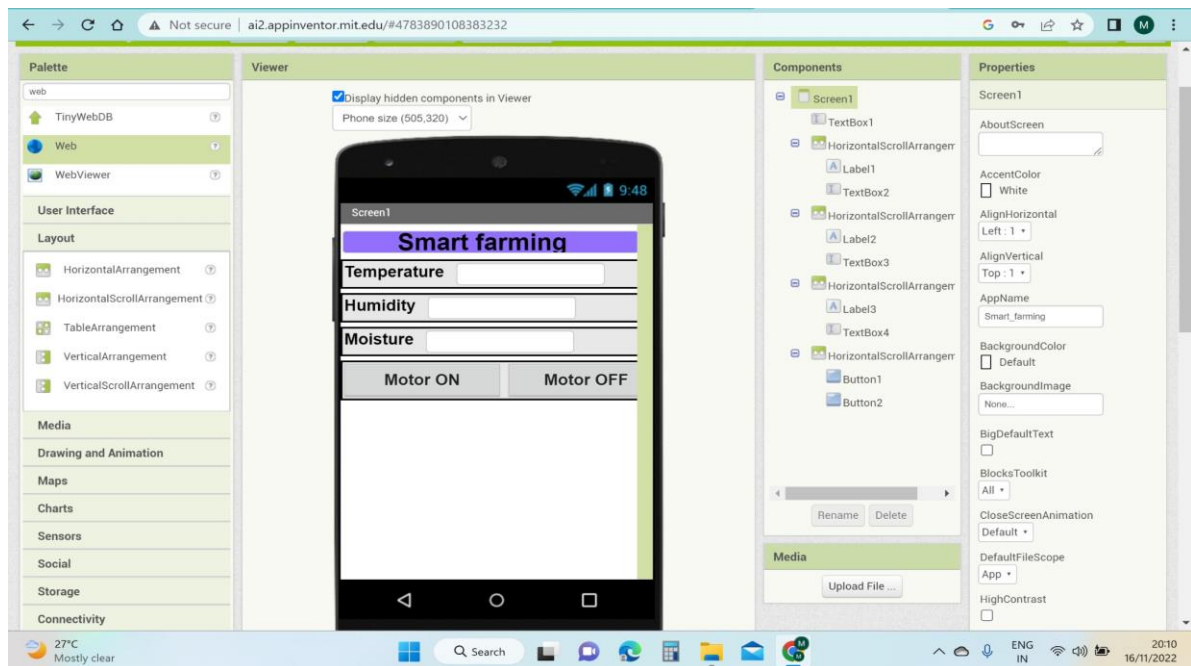


MIT APP INVENTOR

Step 1: Login Into MIT App Inventor

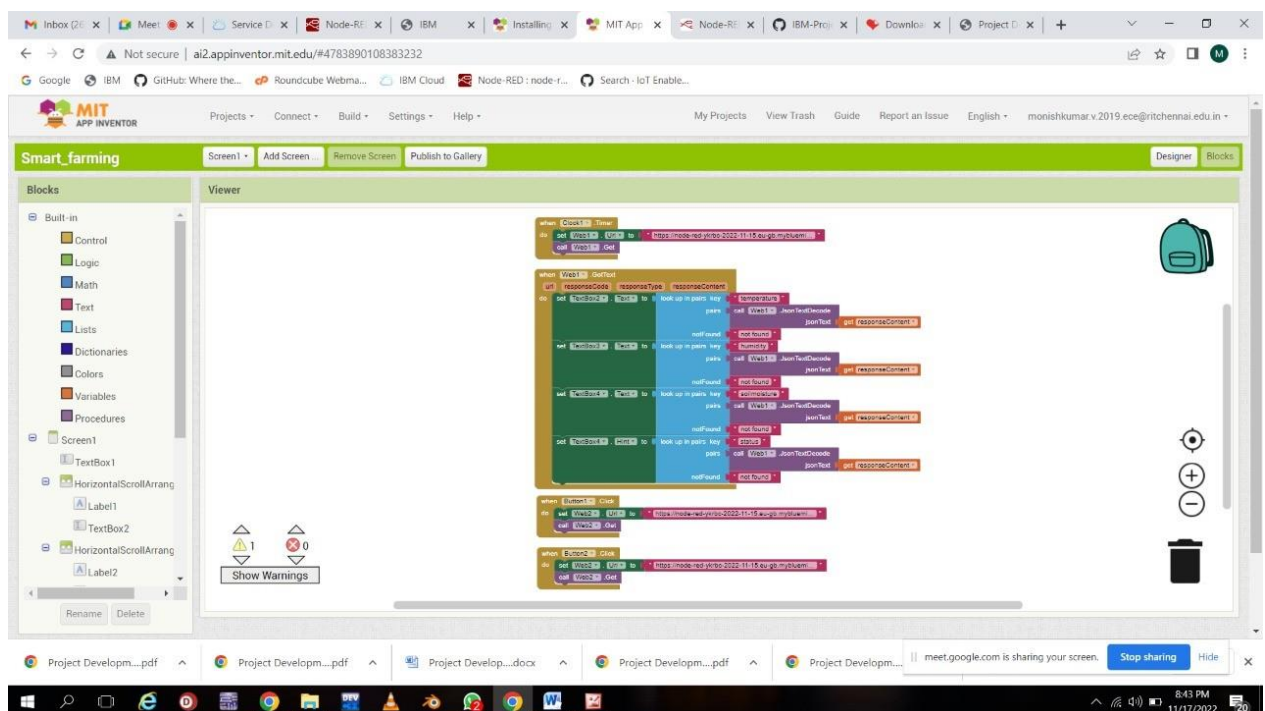


Step 2: Create Your User Interface By Using the Preset Tools



Step 3: Back End Process

- Specify the Cloud URL Details to Receive the Data From Node Red.
- Command 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

