Smart Farmer - IoT Enabled Smart Farming Application

TEAM ID: PNT2022TMID02873

TEAM MEMBERS:

- **ANTRO SAFIN.M**
- **AKILAN BIMBY.A**
- **ARUN KUMAR.M**
- **ANBU SELVAN.N**

MENTOR: U.M. RAMYA

CONTENTS

1. Introduction
2. Problem Statement
3. Proposed Solution
4. Theoretical Analysis
    - 4.1. Block Diagram
    - 4.2. Required Software installation
        - 4.2.1. Node-Red
        - 4.2.2. IBM Watson IoT Platform
        - 4.2.3. Python IDE
5. Building Project
    - 5.1. Create And Configure IBM Cloud Services
    - 5.2. Develop a Python Script
    - 5.3. Build A Web Application Using Node-RED
    - 5.4. Develop the Mobile Application
6. Flow chart
7. Observations & Results
8. Advantages & Disadvantages
9. Conclusion
10. Bibliography

Introduction:

The main aim of this project is to help farmers automate their farms by providing them with a Web and Mobile App through which they can monitor the parameters of the field like Temperature, soil moisture, humidity and etc and control the equipment like water motor and other devices remotely via internet without their actual presence in the field.

Problem Statement:

Farmers are to be present at farm for its maintenance irrespective of the weather conditions. They have to ensure that the crops are well watered and the farm status is monitored by them physically. Farmer have to stay most of the time in field in order to get a good yield. In difficult times like in the presence of pandemic also they have to work hard in their fields risking their lives to provide food for the country.

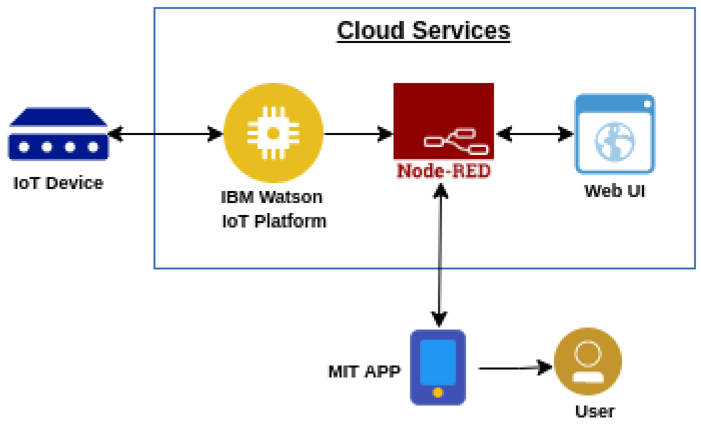
Proposed Solution:

In order to improve the farmer’s working conditions and make them easier, we introduce IoT services to him in which we use cloud services and internet to enable farmer to continue his work remotely via internet. He can monitor the field parameters and control the devices in farm.

Theoretical Analysis:

Block Diagram:

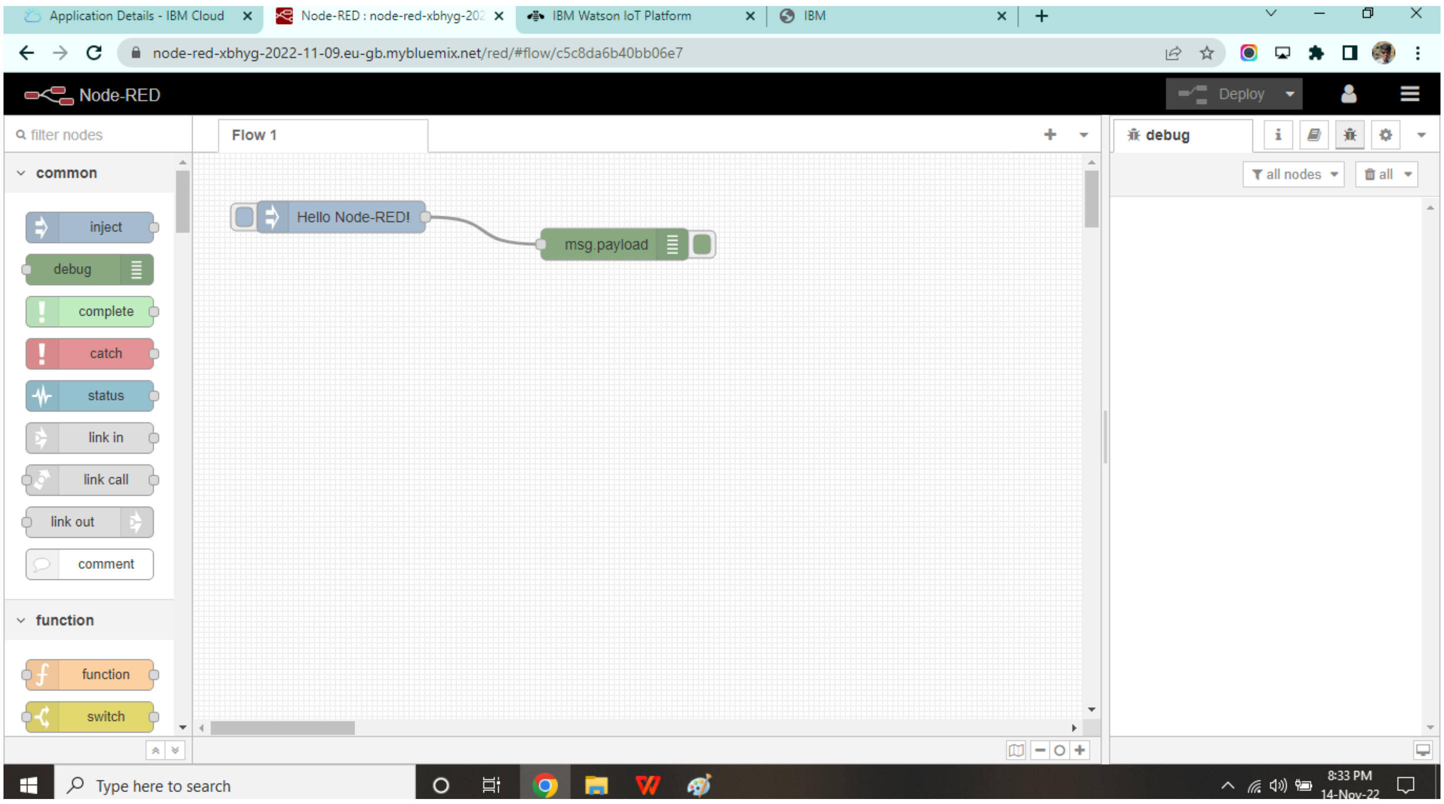
In order to implement the solution , the following approach as shown in the block diagram is used



Required Software Installation:

Node-Red:

Node-RED is a flow-based development tool for visual programming developed originally by IBM for wiring together hardware devices, APIs and online services as part of the Internet of Things.Node-RED provides a web browser-based flow editor, which can be used to create JavaScript functions.



Installation:

- Find the node-red app in ibm cloud catalog
- Install node-red app

To run the application:

- In resources find node-red app
- Click and visit the app url.
- Then open node-red editor.

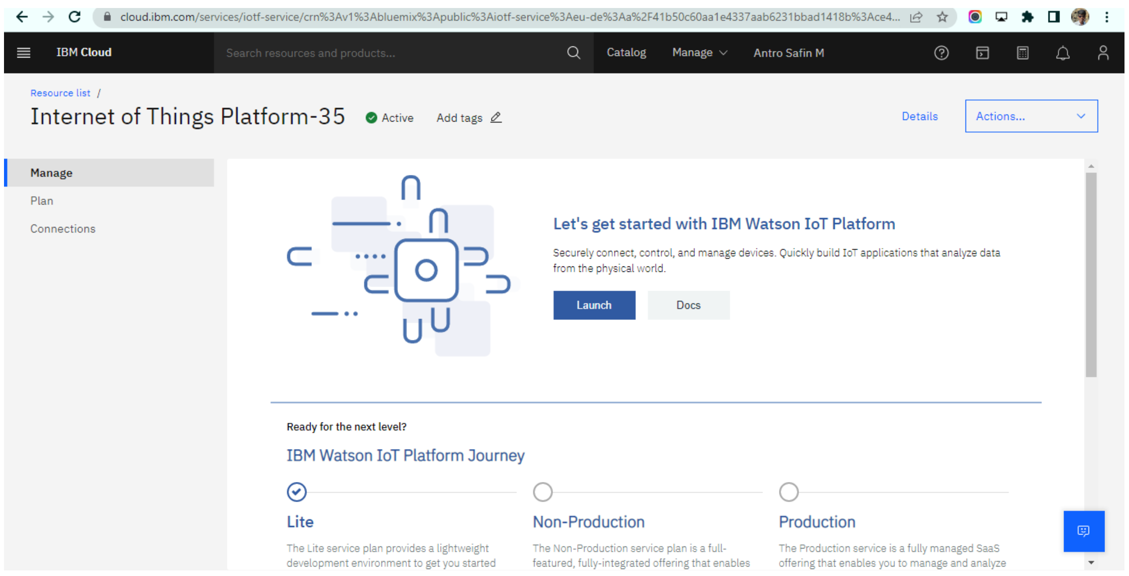
Installation of IBM IoT and Dashboard nodes for Node-Red:

In order to connect to IBM Watson IoT platform and create the Web App UI these nodes are required

- IBM IoT node
- Dashboard node

IBM Watson IoT Platform:

A fully managed, cloud-hosted service with capabilities for device registration, connectivity, control, rapid visualization and data storage. IBM Watson IoT Platform is a managed, cloud-hosted service designed to make it simple to derive value from your IoT devices.



Internet of Things Platform-35 Active Add tags

Let's get started with IBM Watson IoT Platform

Securely connect, control, and manage devices. Quickly build IoT applications that analyze data from the physical world.

Ready for the next level?

IBM Watson IoT Platform Journey

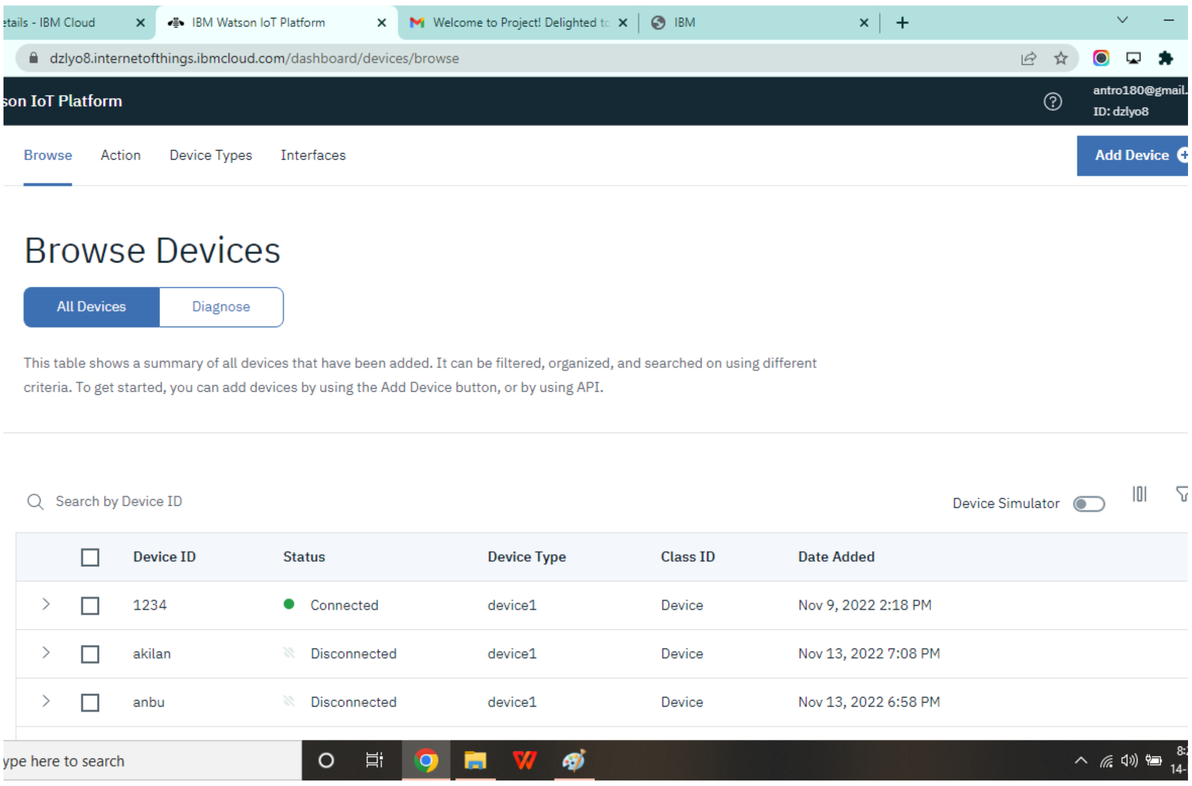
Lite
The Lite service plan provides a lightweight development environment to get you started

Non-Production
The Non-Production service plan is a full-featured, fully-integrated offering that enables

Production
The Production service is a fully managed SaaS offering that enables you to manage and analyze

Steps to configure:

- Create an account in IBM cloud using your email ID
- Create IBM Watson Platform in services in your IBM cloud account
- Launch the IBM Watson IoT Platform
- Create a new device
- Give credentials like device type, device ID, Auth. Token
- Create API key and store API key and token elsewhere.



Browse Devices

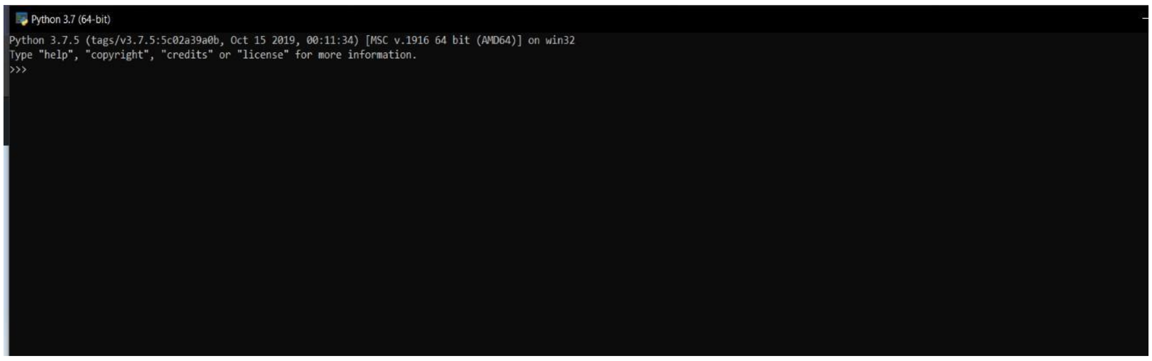
This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Device ID	Status	Device Type	Class ID	Date Added
1234	● Connected	device1	Device	Nov 9, 2022 2:18 PM
akilan	Disconnected	device1	Device	Nov 13, 2022 7:08 PM
anbu	Disconnected	device1	Device	Nov 13, 2022 6:58 PM

Python IDE

Install Python3 compiler

Install any python IDE to execute python scripts, in my case I used python IDLE to execute the code.

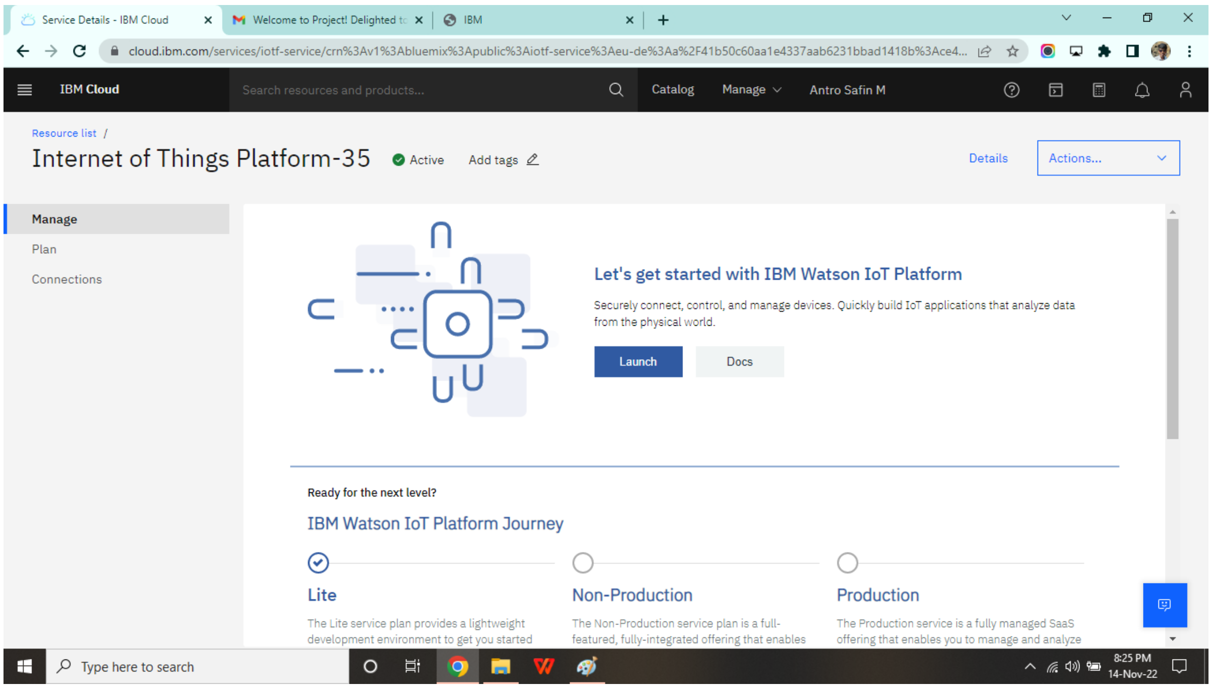


```
Python 3.7 (64-bit)
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Building Project:

1)Create The IBM Watson IoT Platform:

- IBM Watson IoT platform acts as the mediator to connect the web application to IoT device, so create the IBM Watson IoT platform.



Service Details - IBM Cloud | Welcome to Project! Delighted t | IBM

cloud.ibm.com/services/iotf-service/crn%3Av1%3Abluemix%3Apublic%3Aiotf-service%3Aeu-de%3Aa%2F41b50c60aa1e4337aab6231bbad1418b%3Ace4...

IBM Cloud | Search resources and products... | Catalog | Manage | Antro Safin M

Resource list /

Internet of Things Platform-35 · Active · Add tags · Details · Actions...

Manage · Plan · Connections

Let's get started with IBM Watson IoT Platform

Securely connect, control, and manage devices. Quickly build IoT applications that analyze data from the physical world.

Launch · Docs

Ready for the next level?

IBM Watson IoT Platform Journey

Lite — The Lite service plan provides a lightweight development environment to get you started

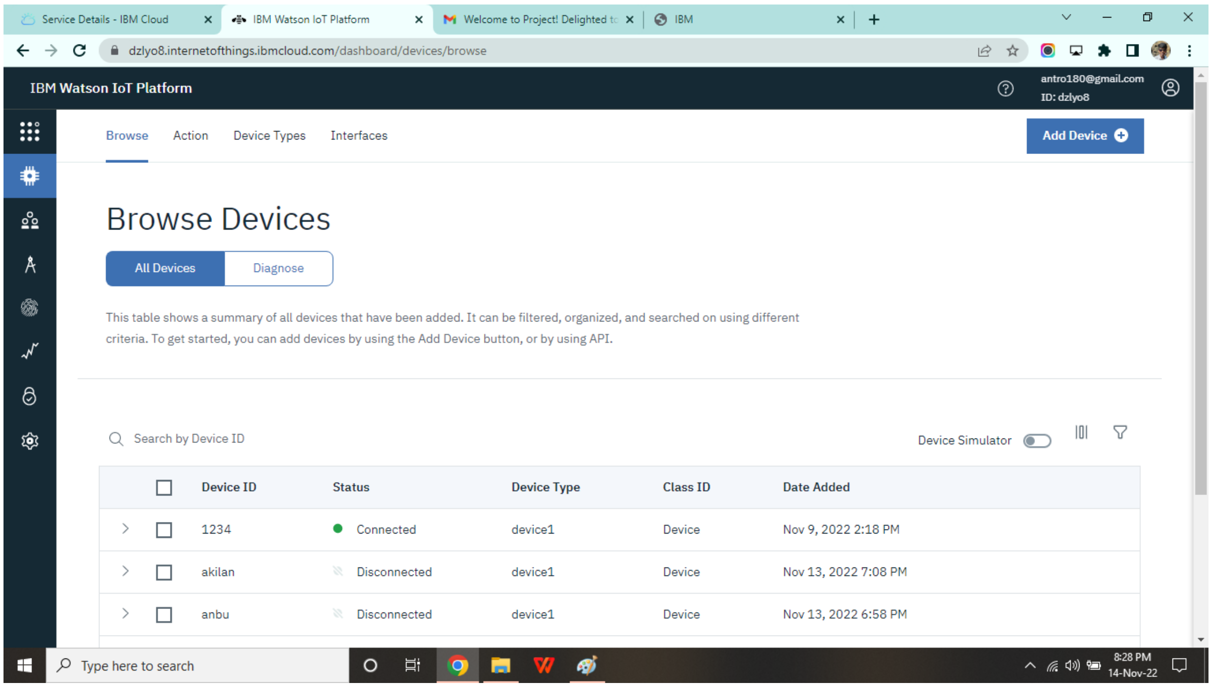
Non-Production — The Non-Production service plan is a full-featured, fully-integrated offering that enables

Production — The Production service is a fully managed SaaS offering that enables you to manage and analyze

Type here to search · 8:25 PM 14-Nov-22

2)Create The IBM Watson IoT Platform device:

- In order to connect the IoT device to the IBM cloud, create a device in the IBM Watson IoT platform and get the device credentials.



Browse Devices

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Device ID	Status	Device Type	Class ID	Date Added
1234	● Connected	device1	Device	Nov 9, 2022 2:18 PM
akilan	Disconnected	device1	Device	Nov 13, 2022 7:08 PM
anbu	Disconnected	device1	Device	Nov 13, 2022 6:58 PM

- **Device credentials:**

"orgId": "dzlyo8",

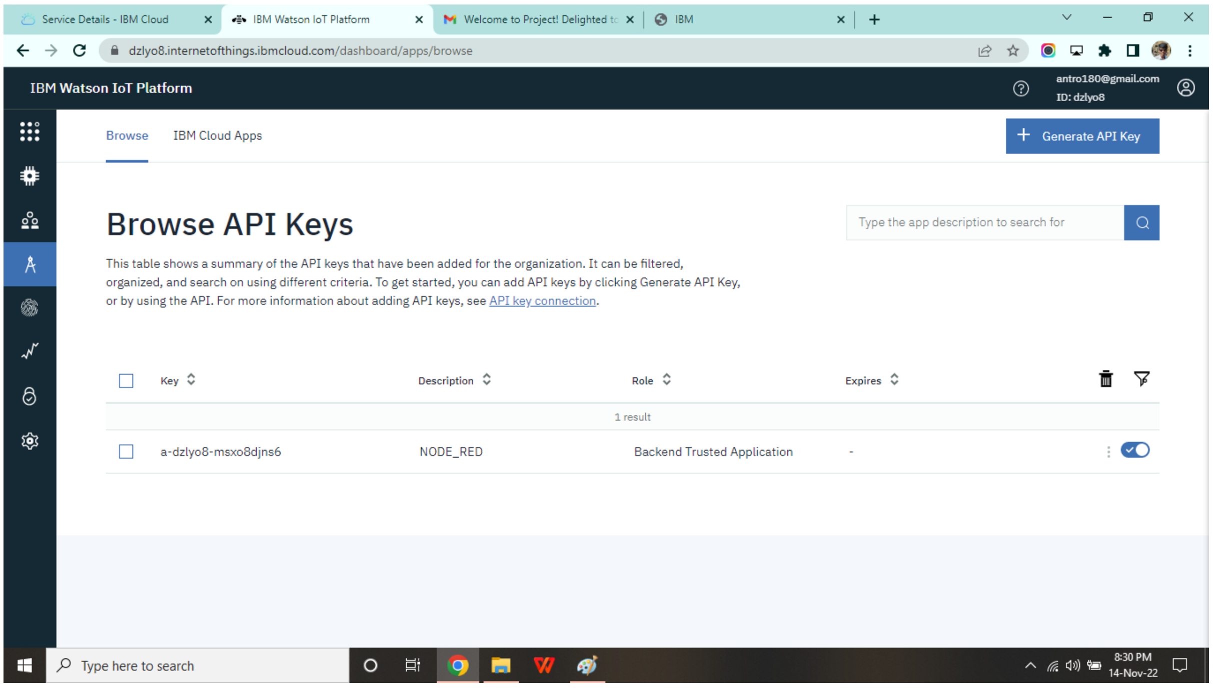
"typeId": "device1",

"deviceId": "1234"

"token": "123456789"

Create API Key:

- Configure the connection security and create API keys that are used in the Node-RED service for accessing the IBM IoT Platform.



IBM Watson IoT Platform

antro180@gmail.com
ID: dzlyo8

Browse IBM Cloud Apps

Generate API Key

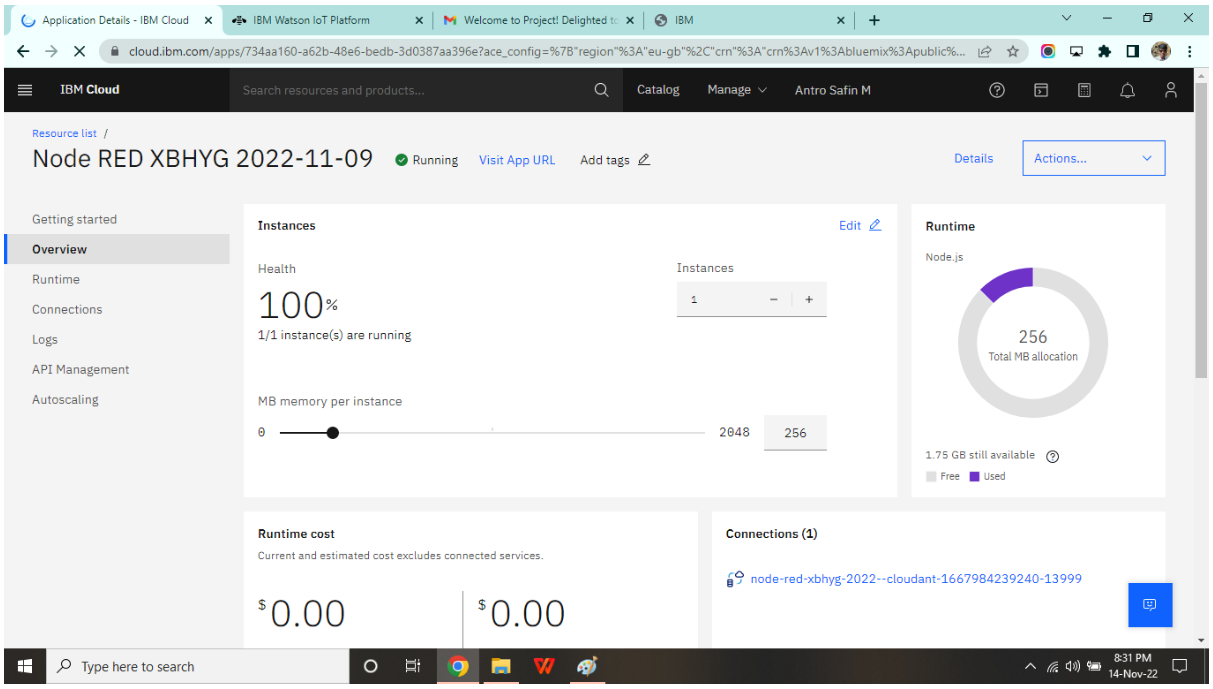
Browse API Keys

This table shows a summary of the API keys that have been added for the organization. It can be filtered, organized, and search on using different criteria. To get started, you can add API keys by clicking Generate API Key, or by using the API. For more information about adding API keys, see API key connection.

Key	Description	Role	Expires
a-dzlyo8-msxo8djns6	NODE_RED	Backend Trusted Application	-

1 result

Create Node-red App:



IBM Cloud

Resource list /

Node RED XBHYG 2022-11-09 Running Visit App URL Add tags

Instances

Health
100%
1/1 instance(s) are running

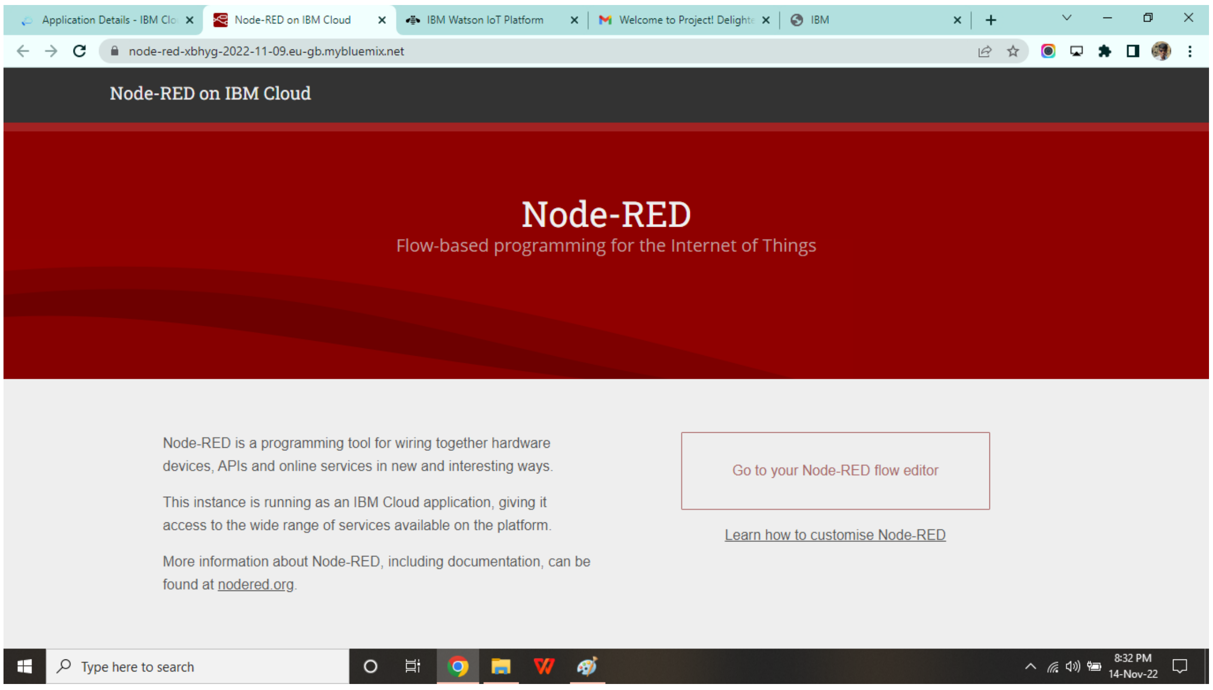
Instances: 1

MB memory per instance
0 — 2048 256

Runtime
Node.js
256 Total MB allocation
1.75 GB still available
Free Used

Runtime cost
Current and estimated cost excludes connected services.
$0.00 $0.00

Connections (1)
node-red-xbhyg-2022--cloudant-1667984239240-13999



Node-RED on IBM Cloud

Node-RED

Flow-based programming for the Internet of Things

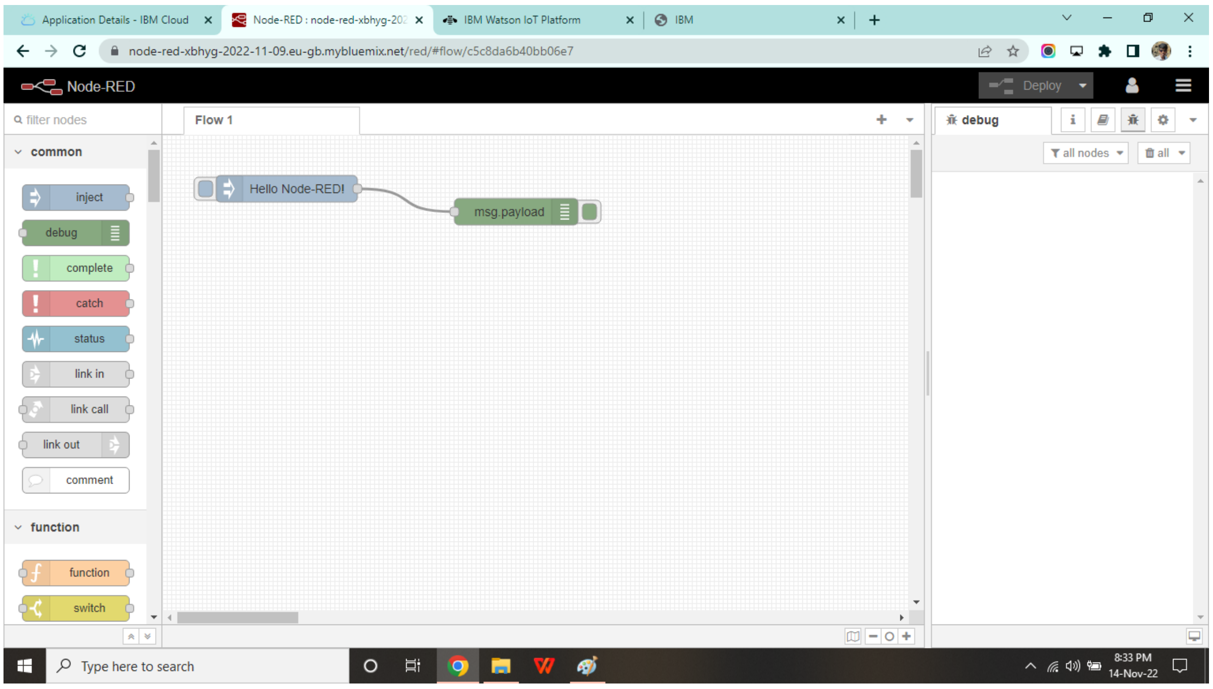
Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

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



Develop a Python Script:

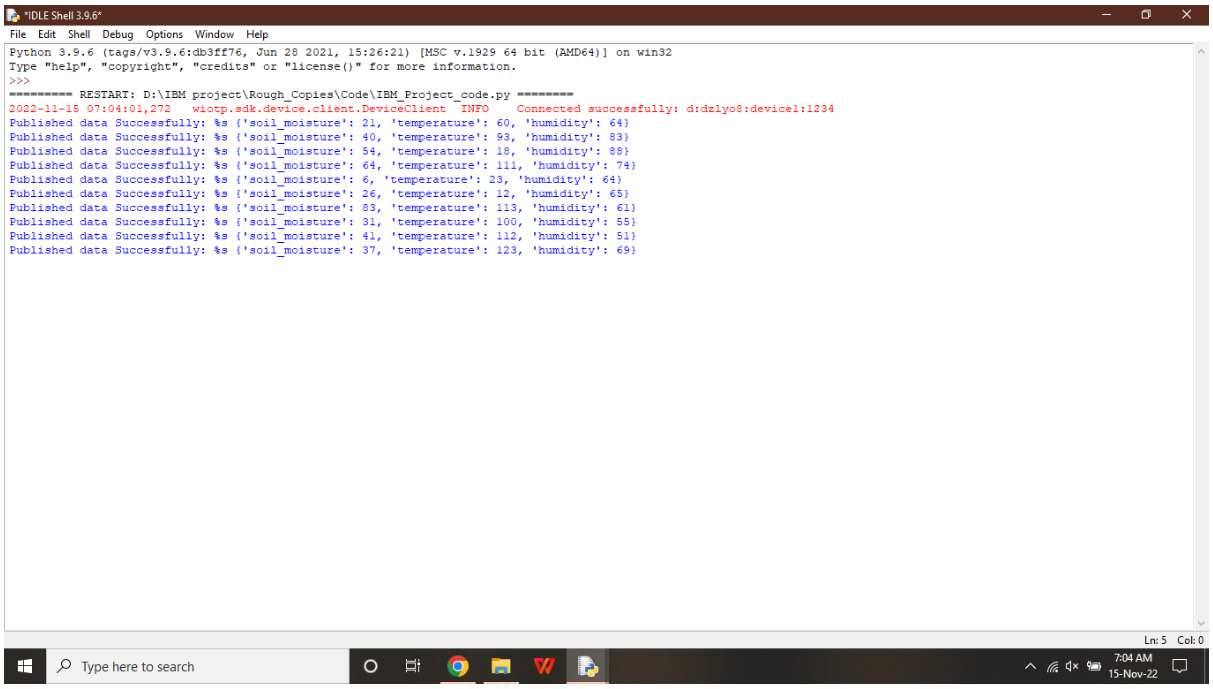
- Develop the python script to publish the data and subscribe the data from the IBM Watson IoT Platform.
- Install the package wiot-sdk to make use of the watson iot platform functions in python script.
- Get the device credentials from the iot platform device.

Python Code:

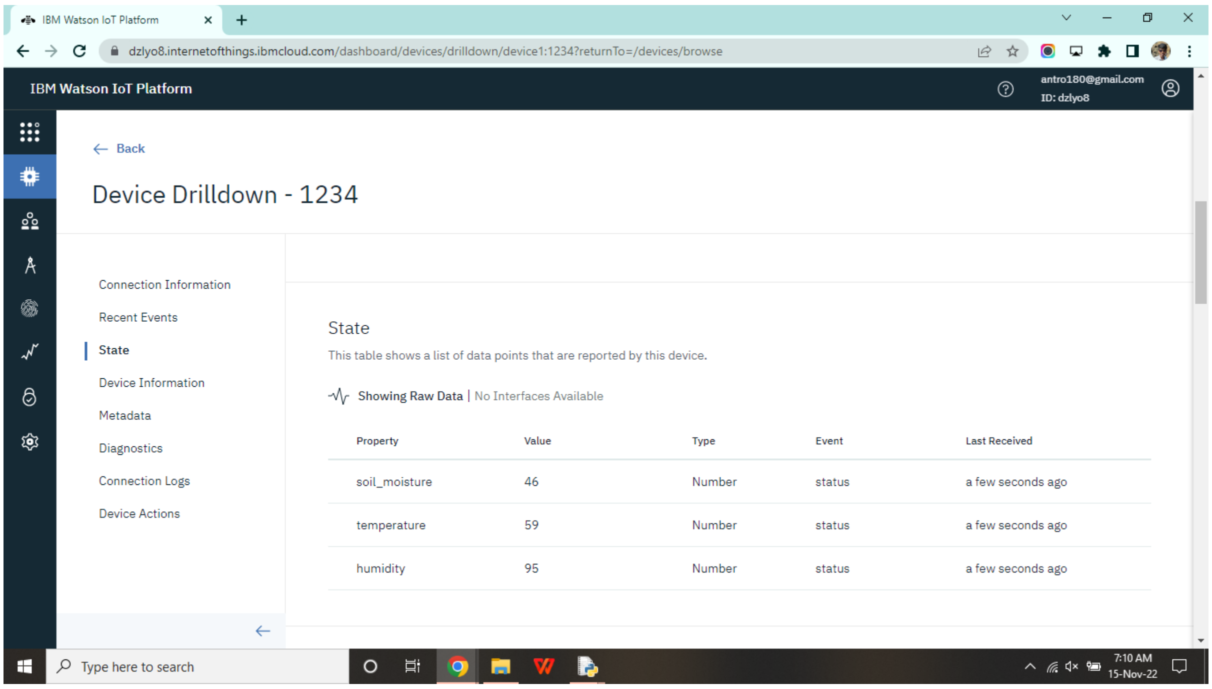
```
#import packages
import wiotp.sdk.device
import time
import os
import datetime
import random

#CallBack function to receive the commands from cloud
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(" ")

#Device credentials
myConfig = {
        "identity": {
            "orgId": "dzlyo8",
            "typeId": "device1",
            "deviceId": "1234"
        },
        "auth": {
            "token": "123456789"
        }
}

#Making Connection to cloud
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()
```

```
#Sending data for every 2 seconds to cloud
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.commandCallback = myCommandCallback
#Disconnected from cloud
client.disconnect()
```

Output:



```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========= RESTART: D:\IBM project\Rough_Copies\Code\IBM_Project_code.py ========
2022-11-15 07:04:01,272   wiotp.sdk.device.client.DeviceClient  INFO    Connected successfully: d:dzlyo8:device1:1234
Published data Successfully: %s {'soil_moisture': 21, 'temperature': 60, 'humidity': 64}
Published data Successfully: %s {'soil_moisture': 40, 'temperature': 93, 'humidity': 83}
Published data Successfully: %s {'soil_moisture': 54, 'temperature': 18, 'humidity': 88}
Published data Successfully: %s {'soil_moisture': 64, 'temperature': 111, 'humidity': 74}
Published data Successfully: %s {'soil_moisture': 6, 'temperature': 23, 'humidity': 64}
Published data Successfully: %s {'soil_moisture': 26, 'temperature': 12, 'humidity': 65}
Published data Successfully: %s {'soil_moisture': 83, 'temperature': 113, 'humidity': 61}
Published data Successfully: %s {'soil_moisture': 31, 'temperature': 100, 'humidity': 55}
Published data Successfully: %s {'soil_moisture': 41, 'temperature': 112, 'humidity': 51}
Published data Successfully: %s {'soil_moisture': 37, 'temperature': 123, 'humidity': 69}
```



Device Drilldown - 1234

State

This table shows a list of data points that are reported by this device.

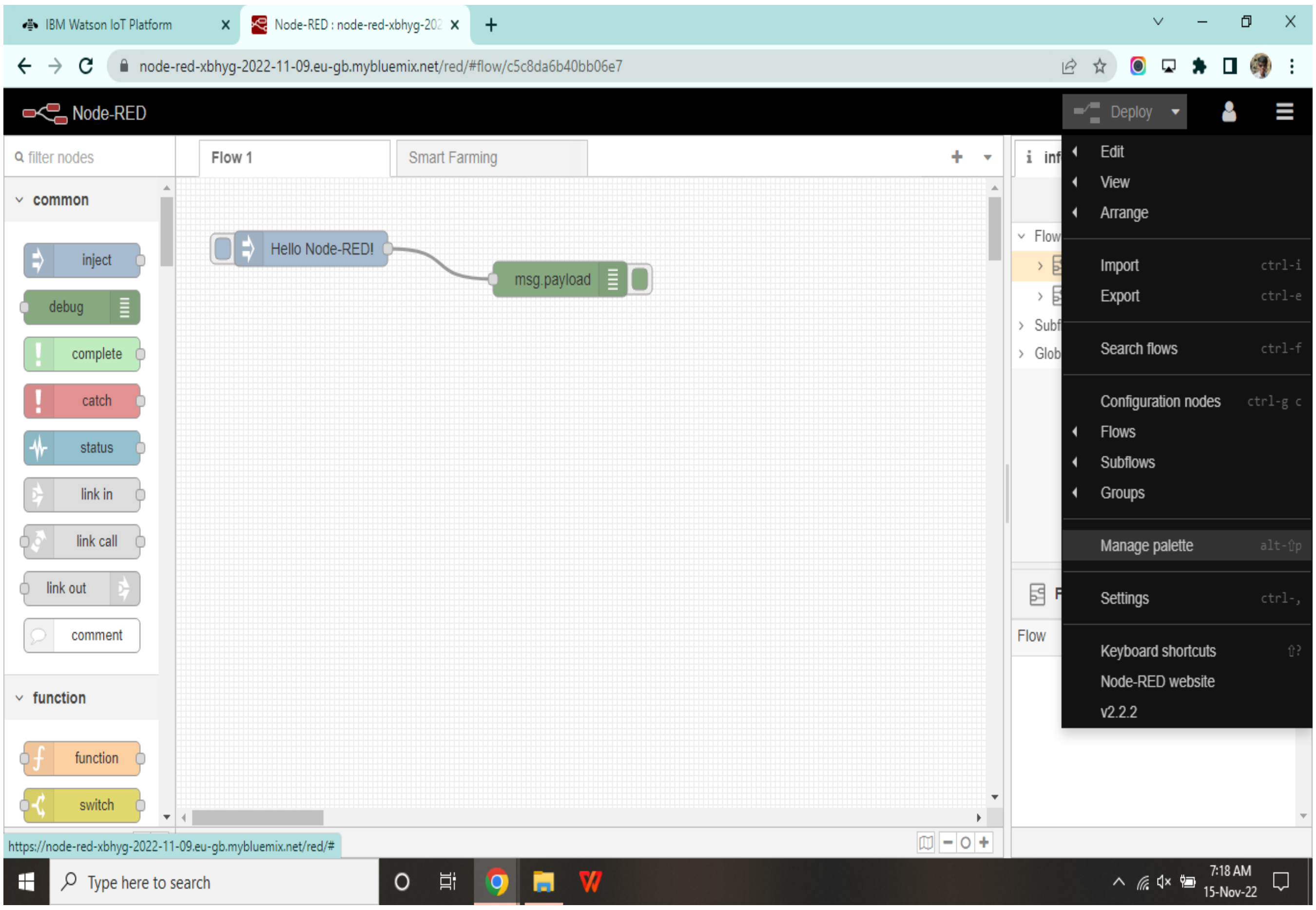
Showing Raw Data | No Interfaces Available

Property	Value	Type	Event	Last Received
soil_moisture	46	Number	status	a few seconds ago
temperature	59	Number	status	a few seconds ago
humidity	95	Number	status	a few seconds ago

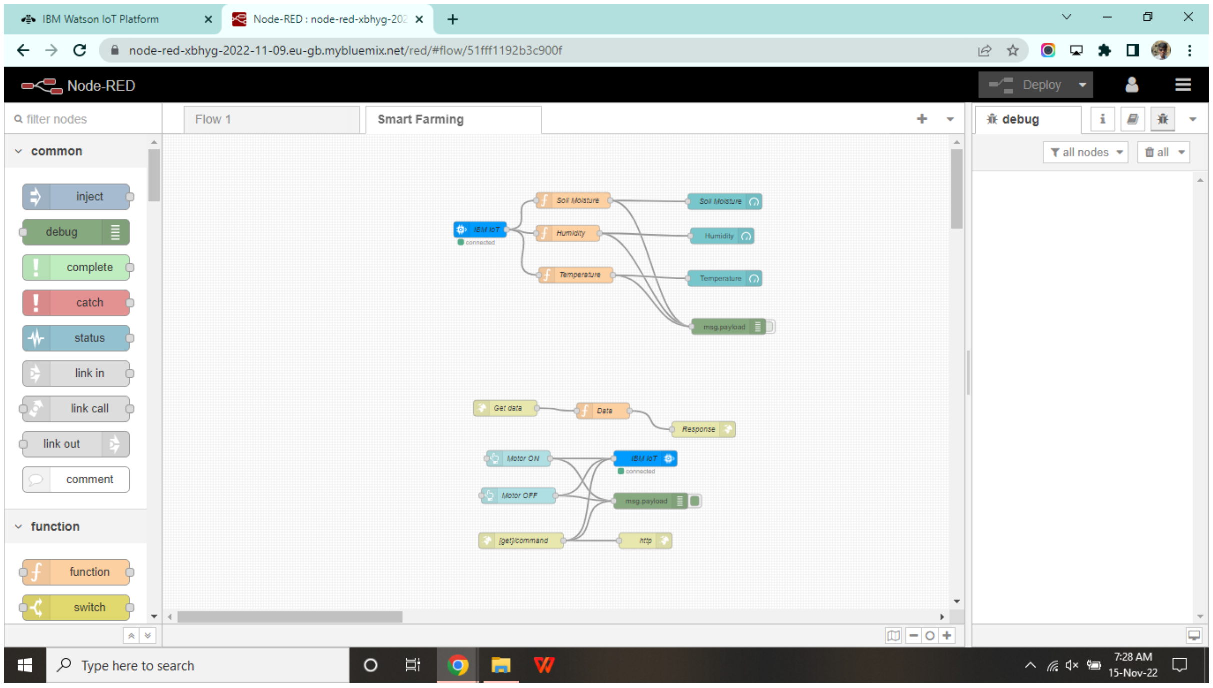
Build A Web Application Using Node-RED:

1) Install node-red-dashboard:

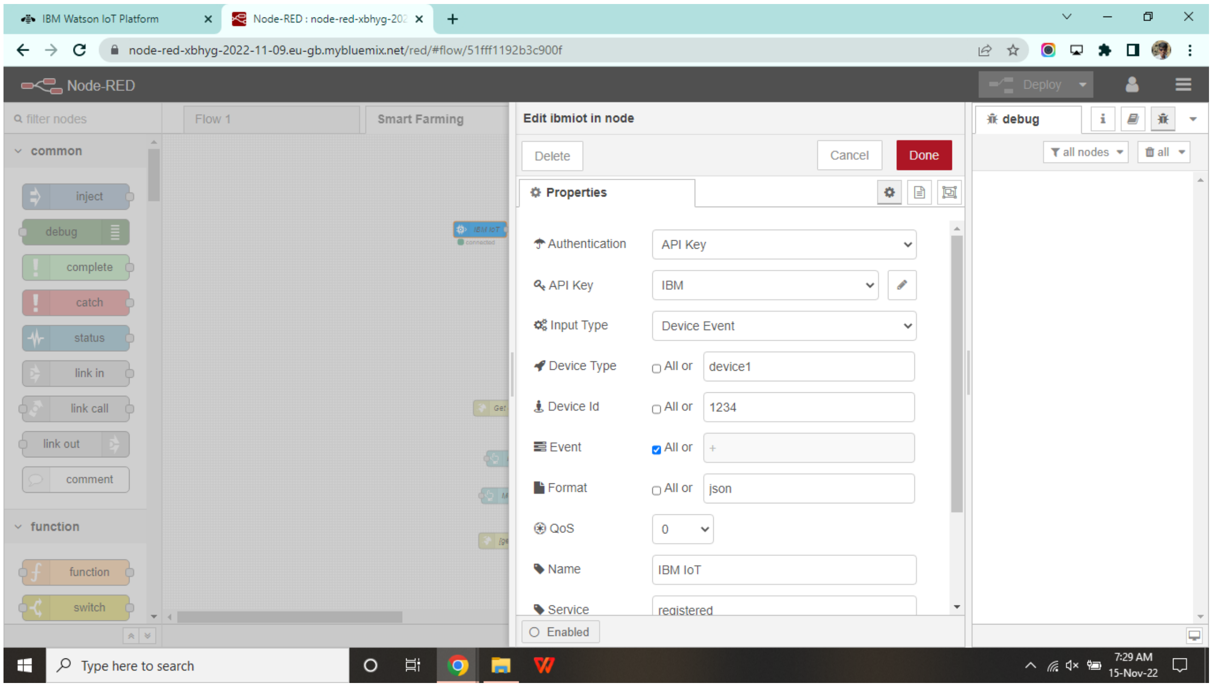
- To install node-red-dashboard, Search node red dashboard in the manage palette and install it.



2)Node-Red flow:



IBM IOT Node Details:


Edit ibmiot in node

Delete | Cancel | Done

Properties

Authentication: API Key

API Key: IBM

Input Type: Device Event

Device Type: ☐ All or device1

Device Id: ☐ All or 1234

Event: ☑ All or +

Format: ☐ All or json

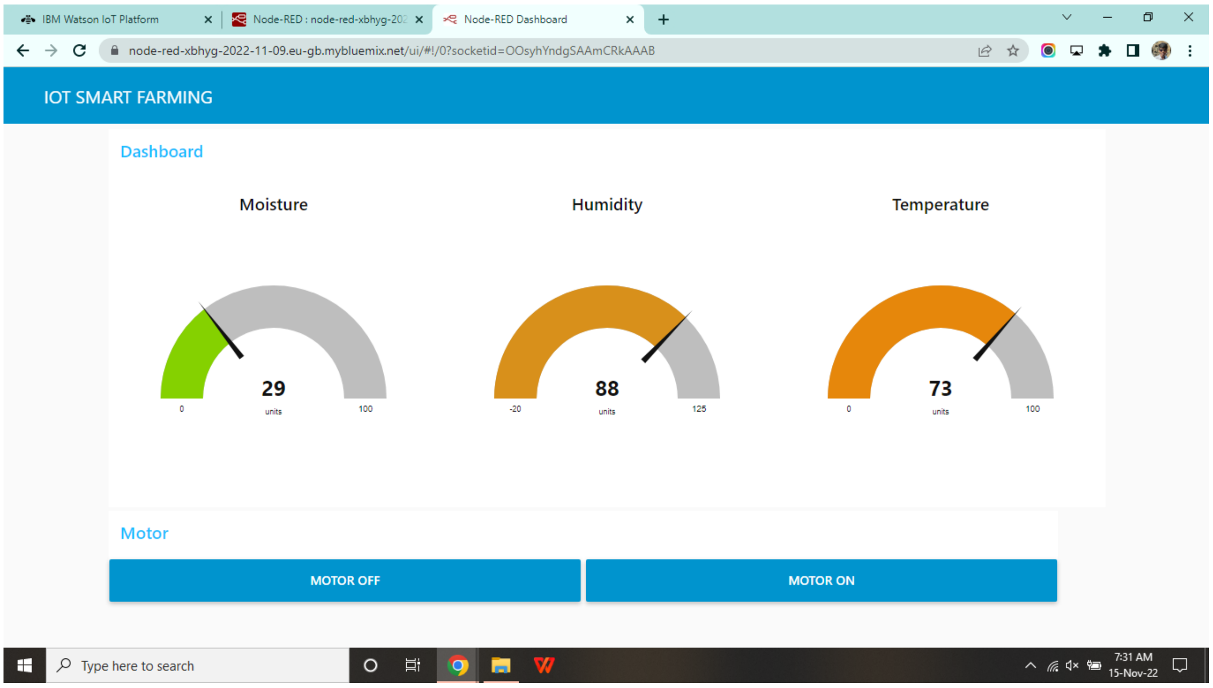
QoS: 0

Name: IBM IoT

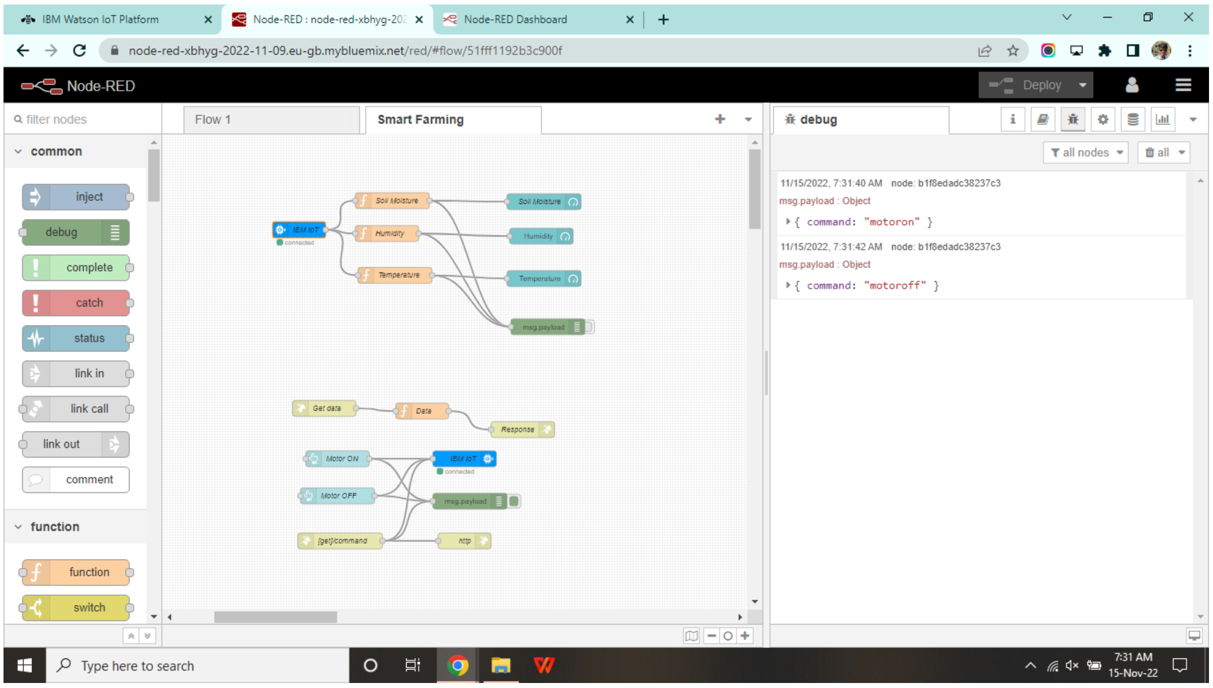
Service: registered

Enabled

Web UI:



Output:

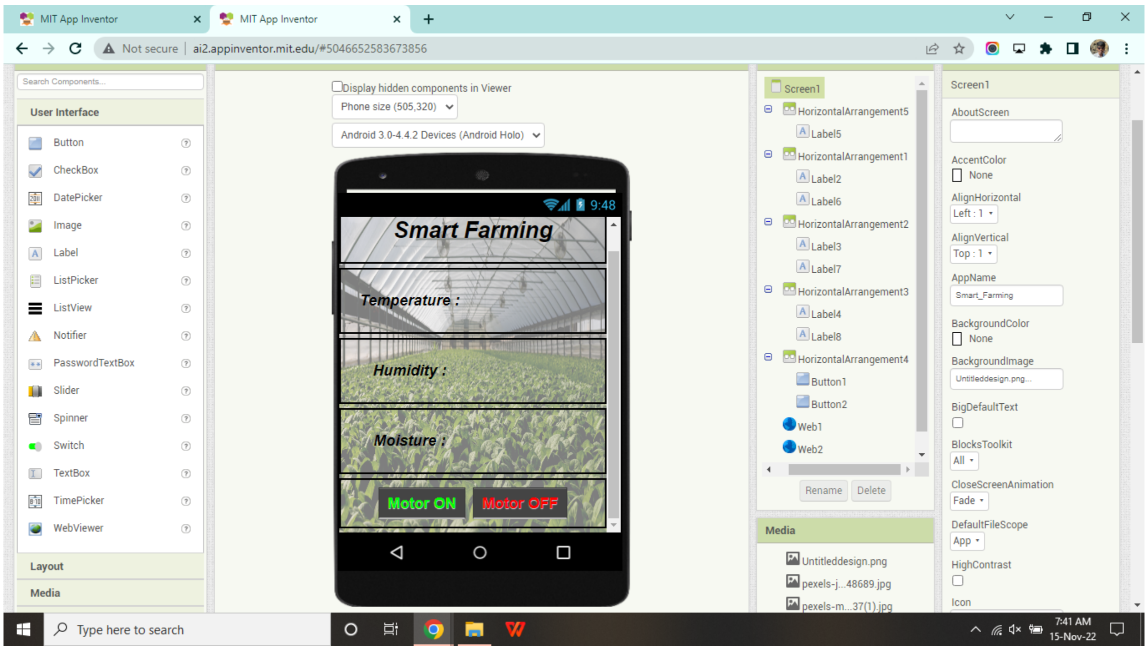


```
11/15/2022, 7:31:40 AM   node: b1f8edadc38237c3
msg.payload : Object
▸{ command: "motoron" }
11/15/2022, 7:31:42 AM   node: b1f8edadc38237c3
msg.payload : Object
▸{ command: "motoroff" }
```

Develop the Mobile Application:

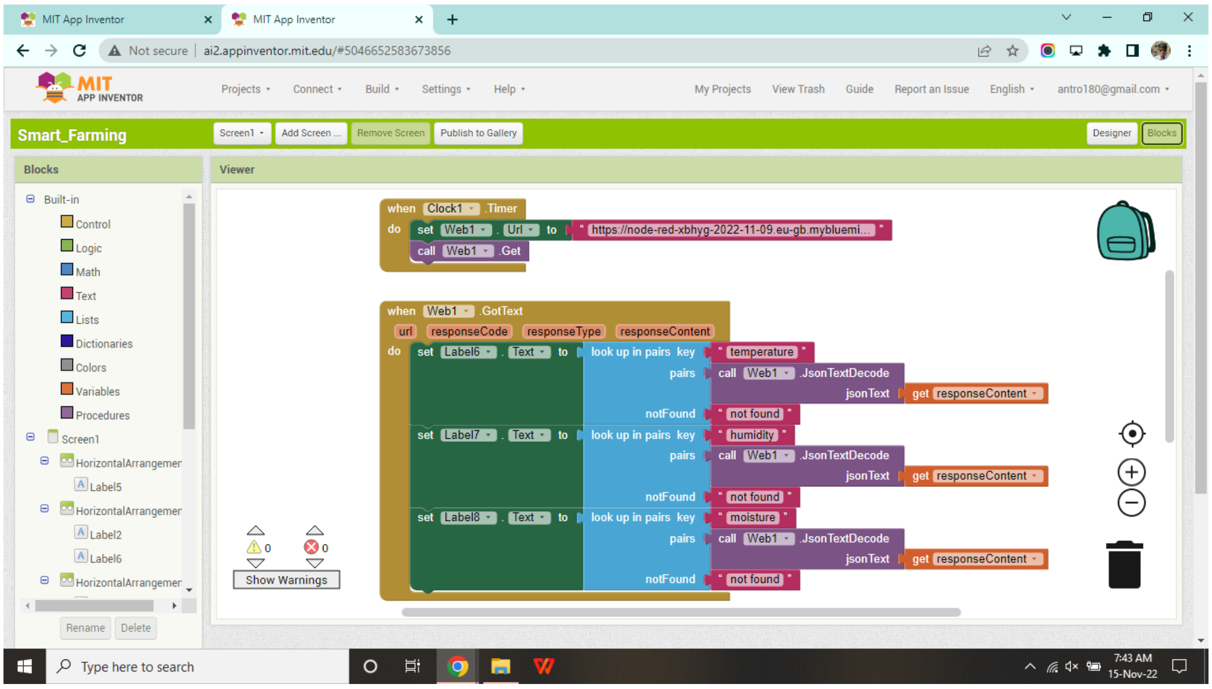
- Develop the mobile application using MIT App inventor which should display all the sensor parameters and have the buttons to control the motors.
- The mobile app should have the following features
  - Display the sensor parameters
  - Buttons for controlling the motors
  - Should communicate with IBM cloud using APIs to get the sensor data and send the commands.

Front-End:

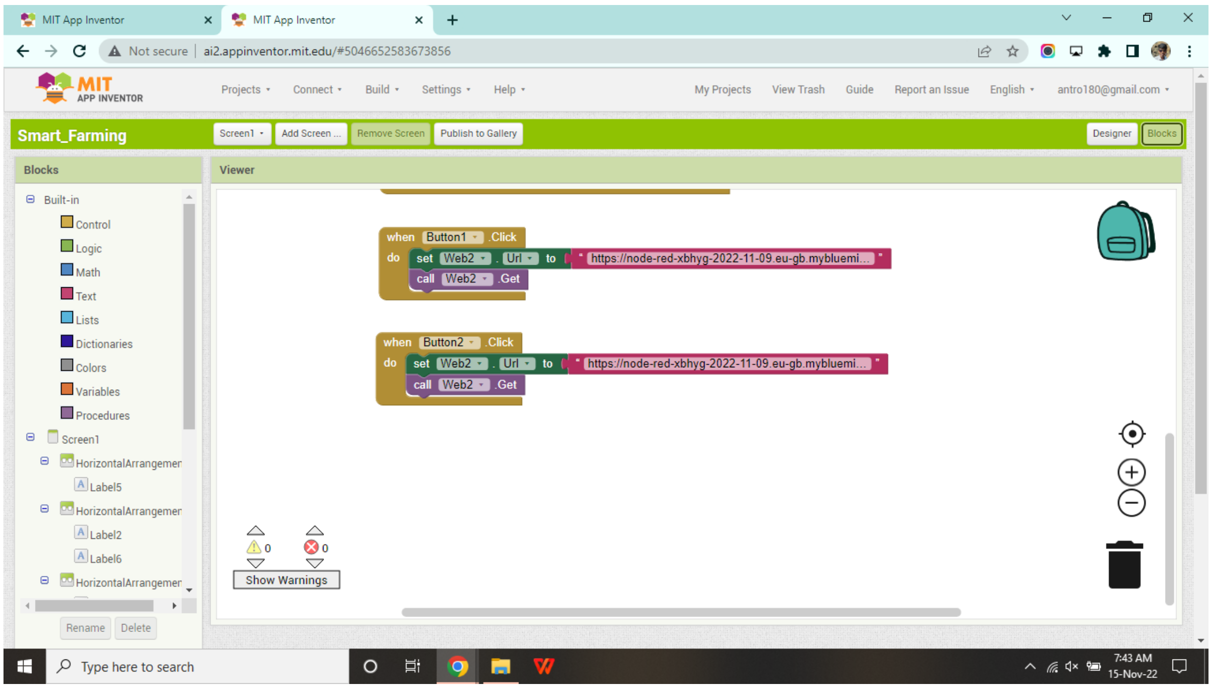


Screen1
AboutScreen
AccentColor None
AlignHorizontal Left : 1
AlignVertical Top : 1
AppName Smart_Farming
BackgroundColor None
BackgroundImage Untitleddesign.png...
BigDefaultText
BlocksToolkit All
CloseScreenAnimation Fade
DefaultFileScope App
HighContrast
Icon

Back-End Code:



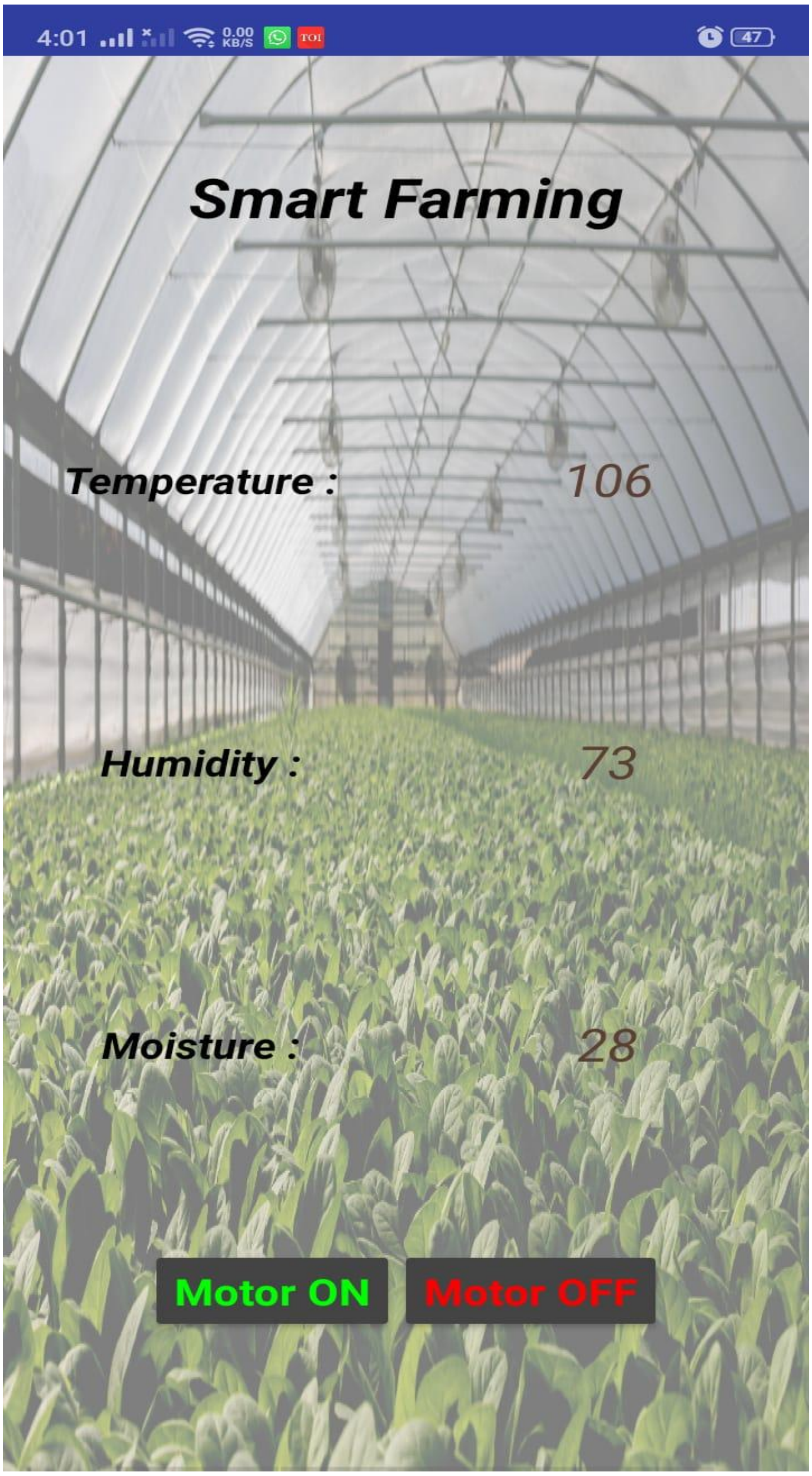
```
when Clock1 .Timer
do  set Web1 . Url to " https://node-red-xbhyg-2022-11-09.eu-gb.mybluemi... "
    call Web1 .Get

when Web1 .GotText
 url  responseCode  responseType  responseContent
do  set Label6 . Text to  look up in pairs  key      " temperature "
                                            pairs    call Web1 .JsonTextDecode
                                                              jsonText  get responseContent
                                            notFound " not found "
    set Label7 . Text to  look up in pairs  key      " humidity "
                                            pairs    call Web1 .JsonTextDecode
                                                              jsonText  get responseContent
                                            notFound " not found "
    set Label8 . Text to  look up in pairs  key      " moisture "
                                            pairs    call Web1 .JsonTextDecode
                                                              jsonText  get responseContent
                                            notFound " not found "
```



```
when Button1 .Click
do  set Web2 . Url to " https://node-red-xbhyg-2022-11-09.eu-gb.mybluemi... "
    call Web2 .Get

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

Output:

Android (.apk) Application:

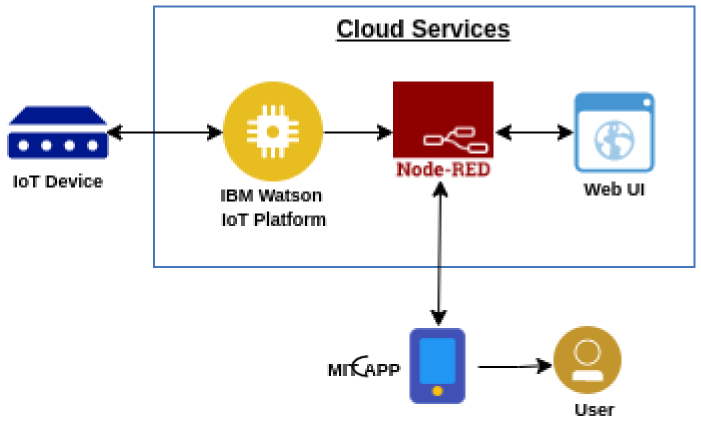


Temperature : 106

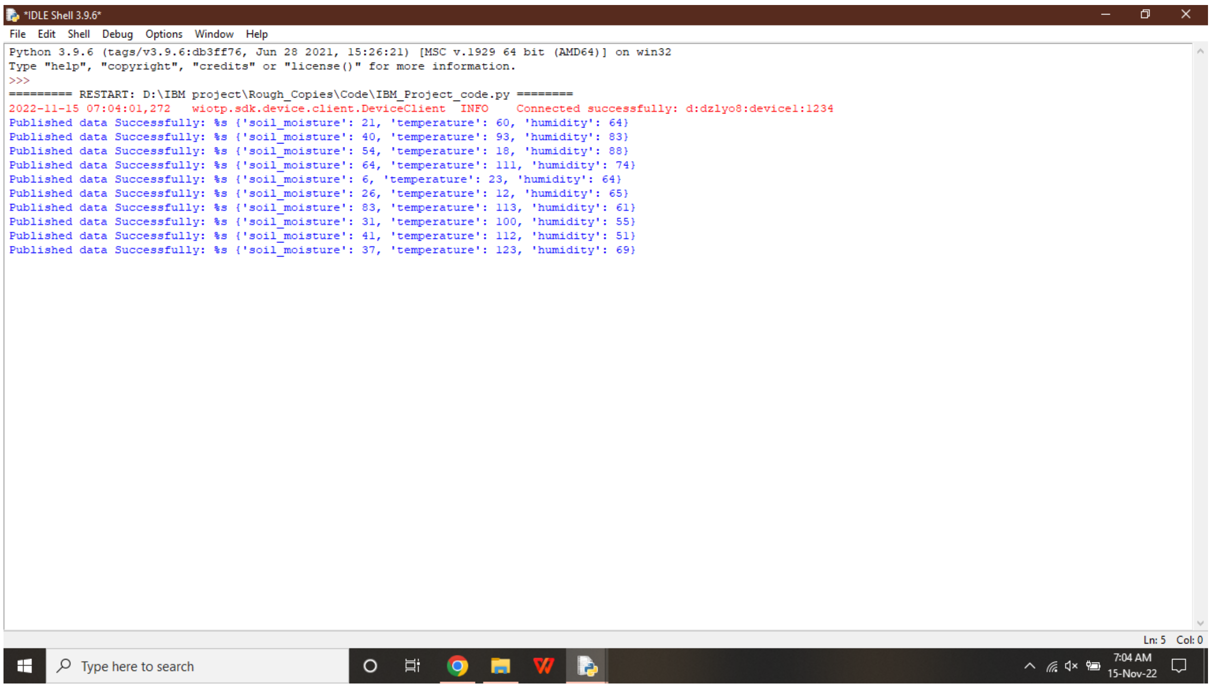
Humidity : 73

Moisture : 28

Flow Chart:



Observations & Results:



```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========= RESTART: D:\IBM project\Rough_Copies\Code\IBM_Project_code.py ========
2022-11-15 07:04:01,272   wiotp.sdk.device.client.DeviceClient  INFO    Connected successfully: d:dzlyo8:device1:1234
Published data Successfully: %s {'soil_moisture': 21, 'temperature': 60, 'humidity': 64}
Published data Successfully: %s {'soil_moisture': 40, 'temperature': 93, 'humidity': 83}
Published data Successfully: %s {'soil_moisture': 54, 'temperature': 18, 'humidity': 88}
Published data Successfully: %s {'soil_moisture': 64, 'temperature': 111, 'humidity': 74}
Published data Successfully: %s {'soil_moisture': 6, 'temperature': 23, 'humidity': 64}
Published data Successfully: %s {'soil_moisture': 26, 'temperature': 12, 'humidity': 65}
Published data Successfully: %s {'soil_moisture': 83, 'temperature': 113, 'humidity': 61}
Published data Successfully: %s {'soil_moisture': 31, 'temperature': 100, 'humidity': 55}
Published data Successfully: %s {'soil_moisture': 41, 'temperature': 112, 'humidity': 51}
Published data Successfully: %s {'soil_moisture': 37, 'temperature': 123, 'humidity': 69}
```

Advantages & Disadvantages:

Advantages:

- Farms can be monitored and controlled remotely.
- Increase in convenience to farmers.
- Less labor cost.
- Better standards of living.

Disadvantages:

- Lack of internet/connectivity issues.
- Added cost of internet and internet gateway infrastructure.
- Farmers wanted to adapt the use of WebApp.

Conclusion:

Thus the objective of the project to implement an IoT system in order to help farmers to control and monitor their farms has been implemented successfully.

Bibliography:

IBM cloud reference: https://cloud.ibm.com/

Python code reference: https://github.com/rachuriharish23/ibmsubscribe

IoT simulator : https://watson-iot-sensor-simulator.mybluemix.net/