

SPRINT 3

Team ID	PNT2022TMID08036
Project Name	SmartFarmer - IoT Enabled Smart Farming Application

Python Script

```
import time
import sys
import ibmiotf.application
import ibmiotf.device import random

organization = "1nuzg6"
deviceType = "srimathi"
deviceId = "94868"
authMethod = "token"
authToken = "987654321"

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data)
    for key in cmd.data.keys():
        if key == 'motor':
            if cmd.data['motor'] == 'ON':
                print("MOTOR is turned ON")
            elif cmd.data['motor'] == 'OFF':
```

```

print("MOTOR is turned OFF")

try:
deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-
method": authMethod, "auth-token": authToken}
deviceCli = ibmiotf.device.Client(deviceOptions)
except Exception as e:
print("Caught exception connecting device: %s" % str(e))
sys.exit()
deviceCli.connect()
while True:
temp=random.randint(0,40)
Humid=random.randint(0,100)
moist=random.randint(0,40)
data = { 'temperature' : temp, 'humidity': Humid, 'soil_moisture':moist
}
def myOnPublishCallback():
print ("Published Temperature = %s C" % temp, "Humidity = %s %" % Humid,
"soil moisture =%s" % moist,"to IBM Watson")
success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
if not success:
print("Not connected to IoT")
time.sleep(10)
deviceCli.commandCallback = myCommandCallback
deviceCli.disconnect()

```

Python Script configured to IBM Watson IoT platform

IBM Watson IoT Platform

Browse Action Device Types Interfaces

Add Device +

Browse Devices

All Devices Diagnose

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.

Search by Device ID

Device Simulator

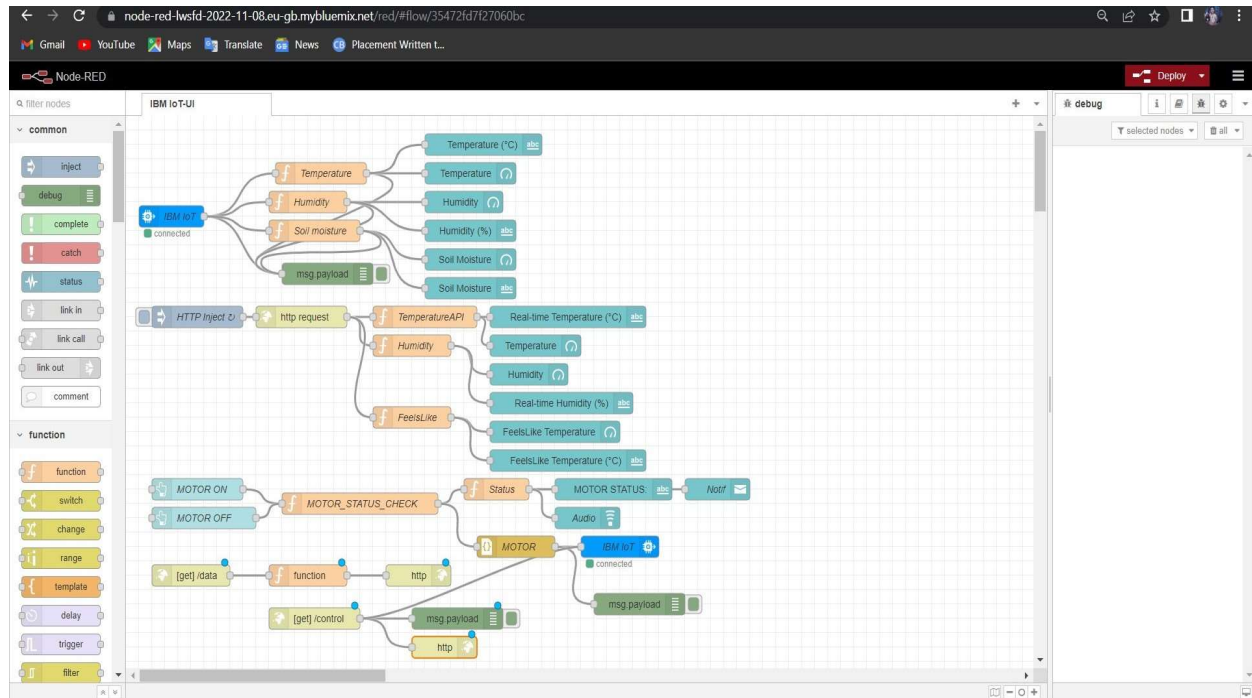
Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
12345	Connected	dominators	Device	Nov 3, 2022 3:08 PM	

Items per page 50 | 1-1 of 1 item

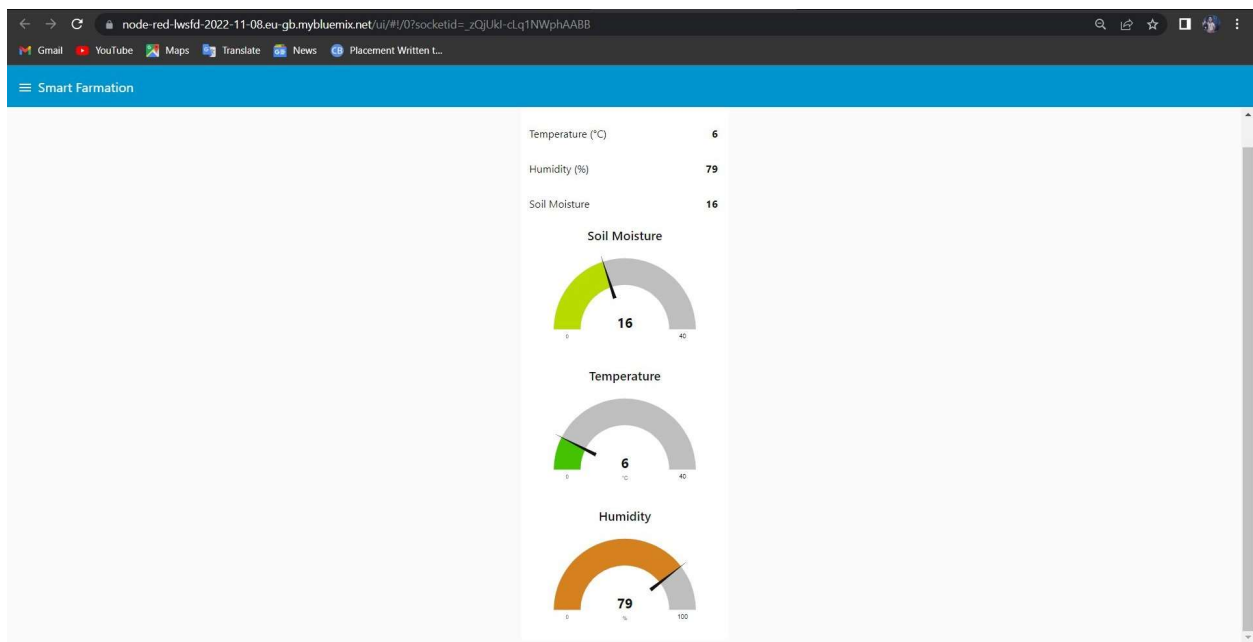
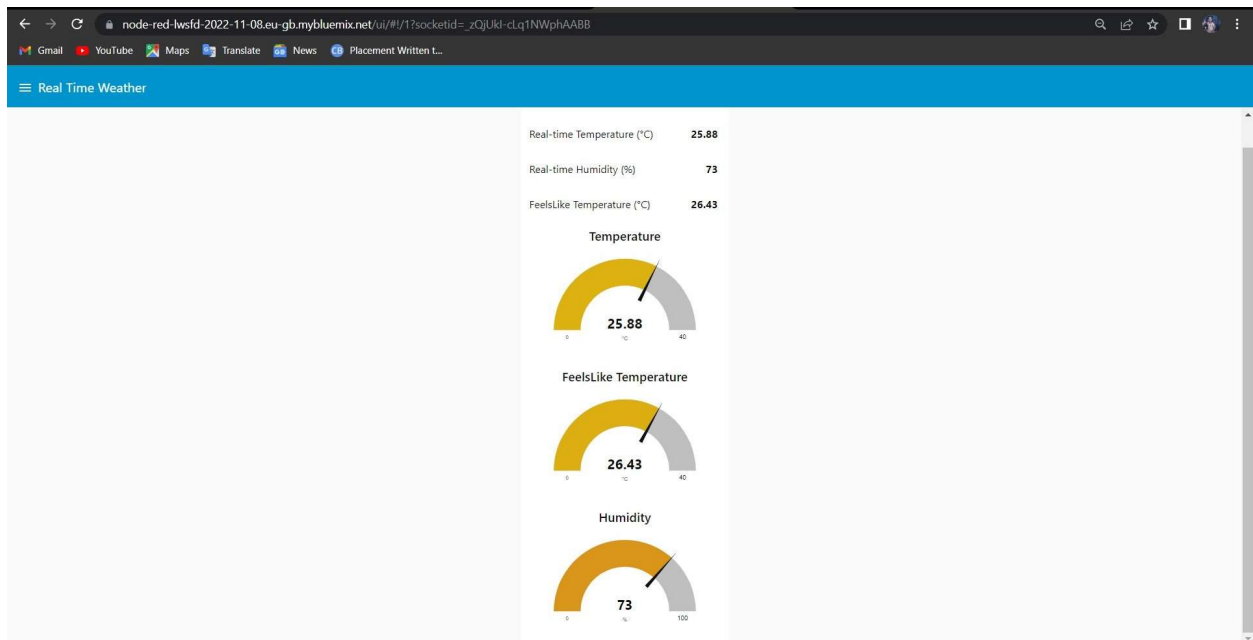
1 of 1 page

0 Simulations running

Node-Red Flow:



Node-Red Dashboard and Application website



The Sensors data in the python script will be received by IBM Watson IoT platform

The screenshot displays a Python script in an IDE Shell and the IBM Watson IoT Platform dashboard. The script, located at `C:\Users\sugen\OneDrive\Desktop\try1.py`, sends sensor data to the platform. The dashboard shows the device `12345` is connected, and the 'Recent Events' tab displays a live stream of data.

Python Script Output:

```
===== RESTART: C:\Users\sugen\OneDrive\Desktop\try1.py =====
2022-11-17 19:33:43,811 ibmiotf.device.Client INFO C
connected successfully: d:zxnybt:dominators:12345
Published Temperature = 22 C Humidity = 0 % to IBM Watson
Published Temperature = 25 C Humidity = 77 % to IBM Watson
Published Temperature = 19 C Humidity = 10 % to IBM Watson
Published Temperature = 39 C Humidity = 34 % to IBM Watson
Published Temperature = 22 C Humidity = 43 % to IBM Watson
Published Temperature = 1 C Humidity = 63 % to IBM Watson
```

IBM Watson IoT Platform Dashboard:

Device: 12345, Status: Connected, Last Received: Nov 3, 2022 3:08 PM

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
IoTSensor	{\"temperature\":1,\"humidity\":63,\"soil_moisture\":...	json	a few seconds ago
IoTSensor	{\"temperature\":22,\"humidity\":43,\"soil_moisture\"...	json	a few seconds ago
IoTSensor	{\"temperature\":39,\"humidity\":34,\"soil_moisture\"...	json	a few seconds ago

Items per page: 50 | 1-1 of 0 Simulations running