

**GOVERNMENT COLLEGE OF ENGINEERING
CHETTIKARAI, DHARMAPURI**



SMART FARMER – IOT ENABLED SMART FARMING APPLICATION

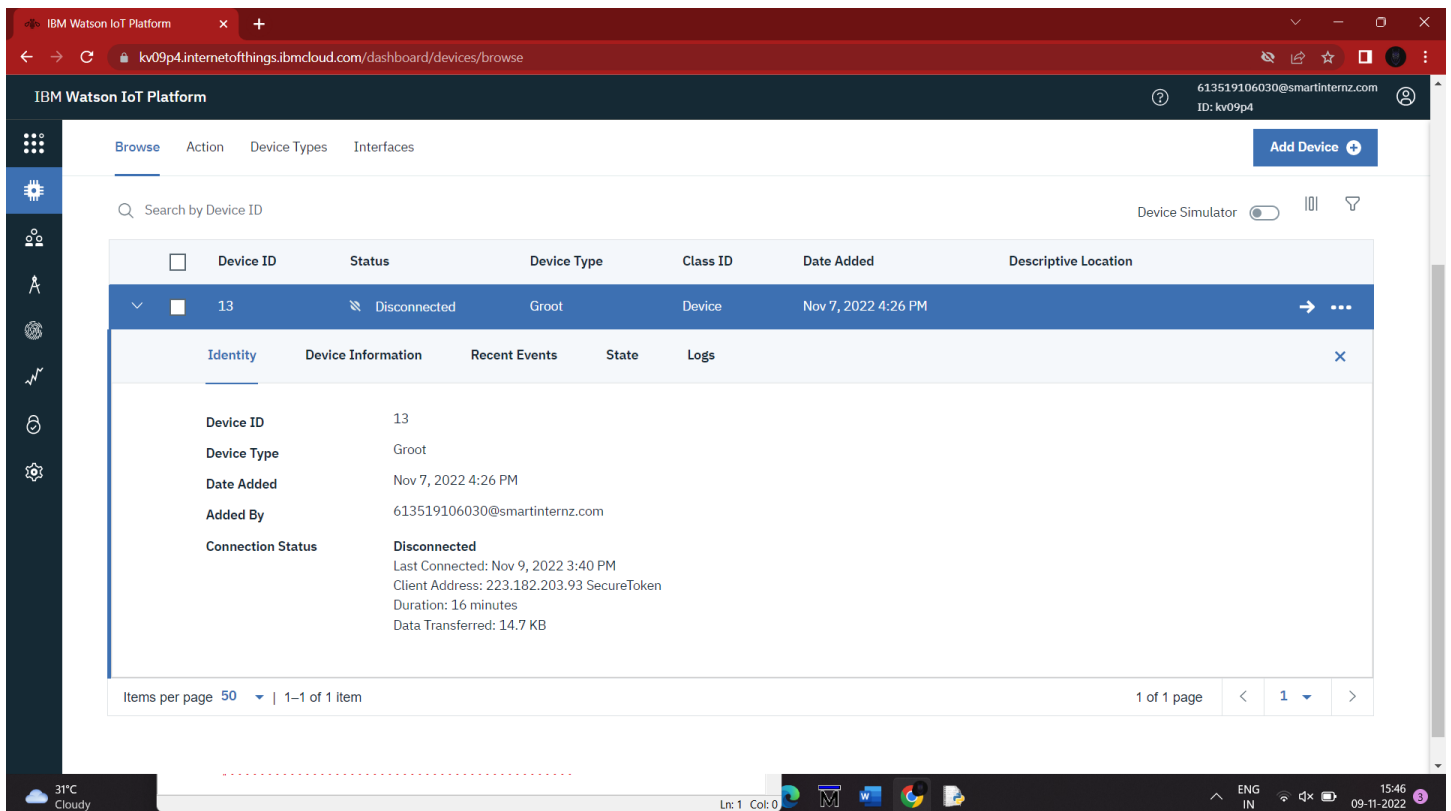
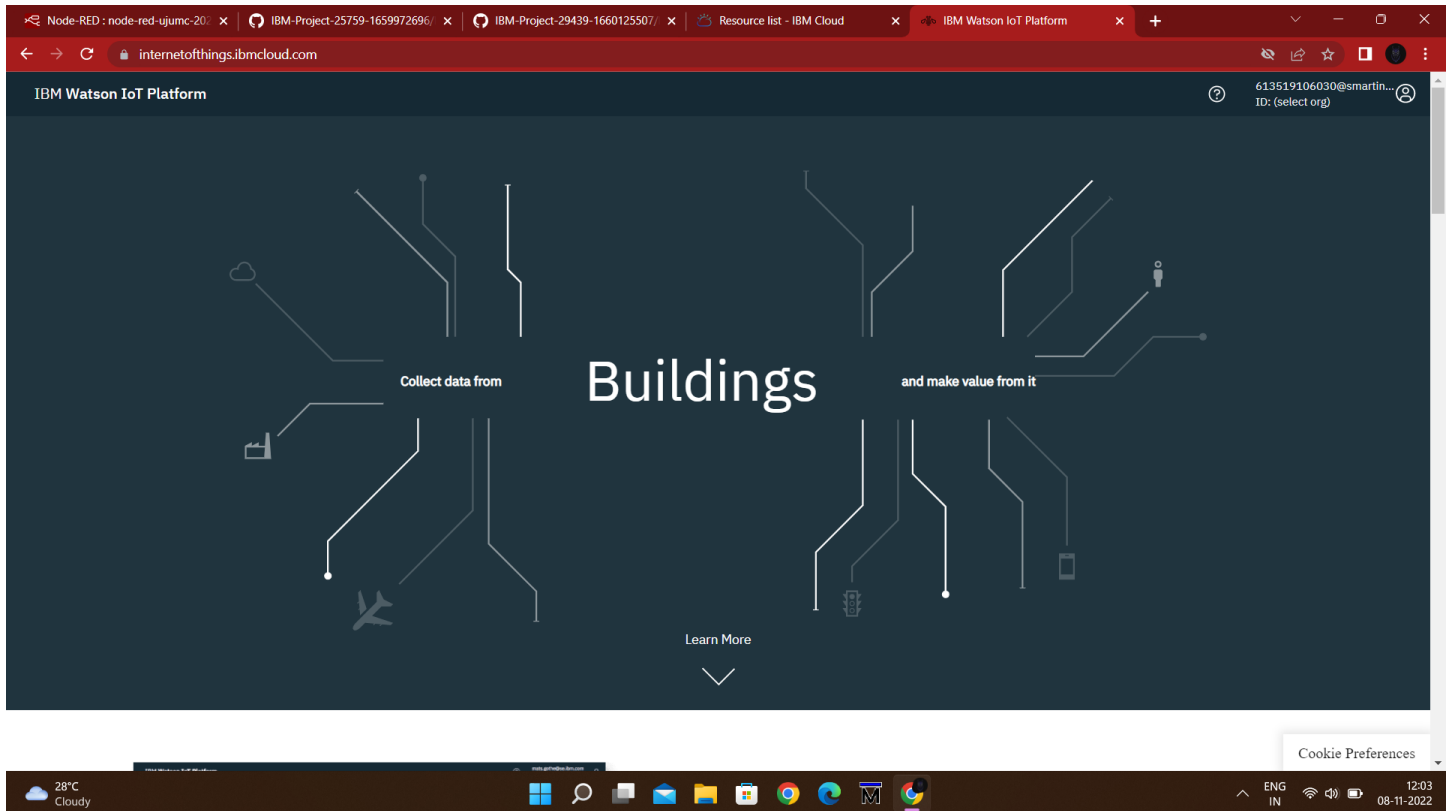
IBM NALAIYATHIRAN

Project Development -Delivery of Sprint 1

Creating And Connecting IBM cloud for Project and Python Code

TITLE	Smart Farmer IoT Enabled Smart Farming Application
DOMAIN NAME	INTERNET OF THINGS
TEAM ID	PNT2022TMID41287
TEAM LEADERNAME	MITHUN SRINIVASAN S
TEAM MEMBER NAME	ARUN KUMAR M AJITH KUMAR S RAVIN G
MENTOR NAME	Dr. DINESH G

Creating IBM cloud device:



Creating Python Code:

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

#Provide your IBM Watson Device Credentials
organization = "kv09p4"
deviceType = "Groot"
deviceId = "13"
authMethod = "token"
authToken = "12345678"
global y

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="motoron":
        print ("motor is on")
    if status=="motoroff" :
        print ("motor is off")
    if status=="manual" :
        print ("Motor Control is in Manual Mode")
    if status=="automatic" :
        print ("Motor control is in Automatic Mode")
    if soilmoisture > 600:
        print ("motor is on")
    #print(cmd)
```

```

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()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times
deviceCli.connect()

while True:
    #Get Sensor Data from DHT11
    temp=random.randint(0,100)
    Humid=random.randint(0,100)
    soilmoisture=random.randint(0,1023)
    Phlevel=random.randint(0,14)
    y=soilmoisture
    data = { 'temp' : temp, 'Humid': Humid,'soilmoisture' : soilmoisture , 'Phlevel' : Phlevel }
    #print data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "Humidity = %s %" % Humid,"Soil Moisture is %s %" % soilmoisture,"PH level is %s" %Phlevel , "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

# Disconnect the device and application from the cloud
deviceCli.disconnect()

```

Connecting IBM Watson and Python Code:

IBM Watson IoT Platform

kv09p4.internetofthings.ibmcloud.com/dashboard/devices/browse

613519106030@smartinternz.com
ID: kv09p4

Device ID: 13, Status: Connected, Device Type: Groot, Class ID: Device, Date Added: Nov 7, 2022 4:26 PM

Identity | Device Information | **Recent Events** | State | Logs

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

Event	Value	Format	Last Received
IoTSensor	{"temp":30,"Humid":55,"soilmoisture":413,"Phle..."	json	a few seconds ago
IoTSensor	{"temp":40,"Humid":51,"soilmoisture":489,"Phle..."	json	a few seconds ago
IoTSensor	{"temp":44,"Humid":40,"soilmoisture":784,"Phle..."	json	a few seconds ago
IoTSensor	{"temp":95,"Humid":34,"soilmoisture":808,"Phle..."	json	a few seconds ago
IoTSensor	{"temp":35,"Humid":34,"soilmoisture":21,"Phlev..."	json	a few seconds ago

Items per page: 50 | 1-1 of 1 item

IBM Watson IoT Platform

kv09p4.internetofthings.ibmcloud.com/dashboard/devices/browse

613519106030@smartinternz.com
ID: kv09p4

Device ID: 13, Status: Connected, Device Type: Groot, Class ID: Device, Date Added: Nov 7, 2022 4:26 PM

Identity | Device Information | **Recent Events** | State

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

Event	Value	Format
IoTSensor	{"temp":69,"Humid":23,"soilmoisture":172,"Phle..."	json
IoTSensor	{"temp":16,"Humid":47,"soilmoisture":276,"Phle..."	json
IoTSensor	{"temp":30,"Humid":55,"soilmoisture":413,"Phle..."	json
IoTSensor	{"temp":40,"Humid":51,"soilmoisture":489,"Phle..."	json
IoTSensor	{"temp":44,"Humid":40,"soilmoisture":784,"Phle..."	json

Items per page: 50 | 1-1 of 1 item

Python 3.7.0 Shell

```
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\IBM project\IBM iot.py =====
2022-11-09 15:49:31,491 ibmiotf.device.Client INFO Connected successfully: d:kv09p4:Groot:13
Published Temperature = 88 C Humidity = 8 % Soil Moisture is 596 % PH level is 13 to IBM Watson
Published Temperature = 34 C Humidity = 92 % Soil Moisture is 91 % PH level is 1 to IBM Watson
Published Temperature = 18 C Humidity = 13 % Soil Moisture is 1022 % PH level is 10 to IBM Watson
Published Temperature = 65 C Humidity = 16 % Soil Moisture is 315 % PH level is 14 to IBM Watson
Published Temperature = 84 C Humidity = 73 % Soil Moisture is 260 % PH level is 14 to IBM Watson
Published Temperature = 81 C Humidity = 45 % Soil Moisture is 283 % PH level is 0 to IBM Watson
Published Temperature = 30 C Humidity = 96 % Soil Moisture is 969 % PH level is 11 to IBM Watson
Published Temperature = 35 C Humidity = 34 % Soil Moisture is 21 % PH level is 4 to IBM Watson
Published Temperature = 95 C Humidity = 34 % Soil Moisture is 808 % PH level is 8 to IBM Watson
Published Temperature = 44 C Humidity = 40 % Soil Moisture is 784 % PH level is 1 to IBM Watson
Published Temperature = 40 C Humidity = 51 % Soil Moisture is 489 % PH level is 10 to IBM Watson
Published Temperature = 30 C Humidity = 55 % Soil Moisture is 413 % PH level is 10 to IBM Watson
Published Temperature = 16 C Humidity = 47 % Soil Moisture is 276 % PH level is 4 to IBM Watson
Published Temperature = 69 C Humidity = 23 % Soil Moisture is 172 % PH level is 6 to IBM Watson
```