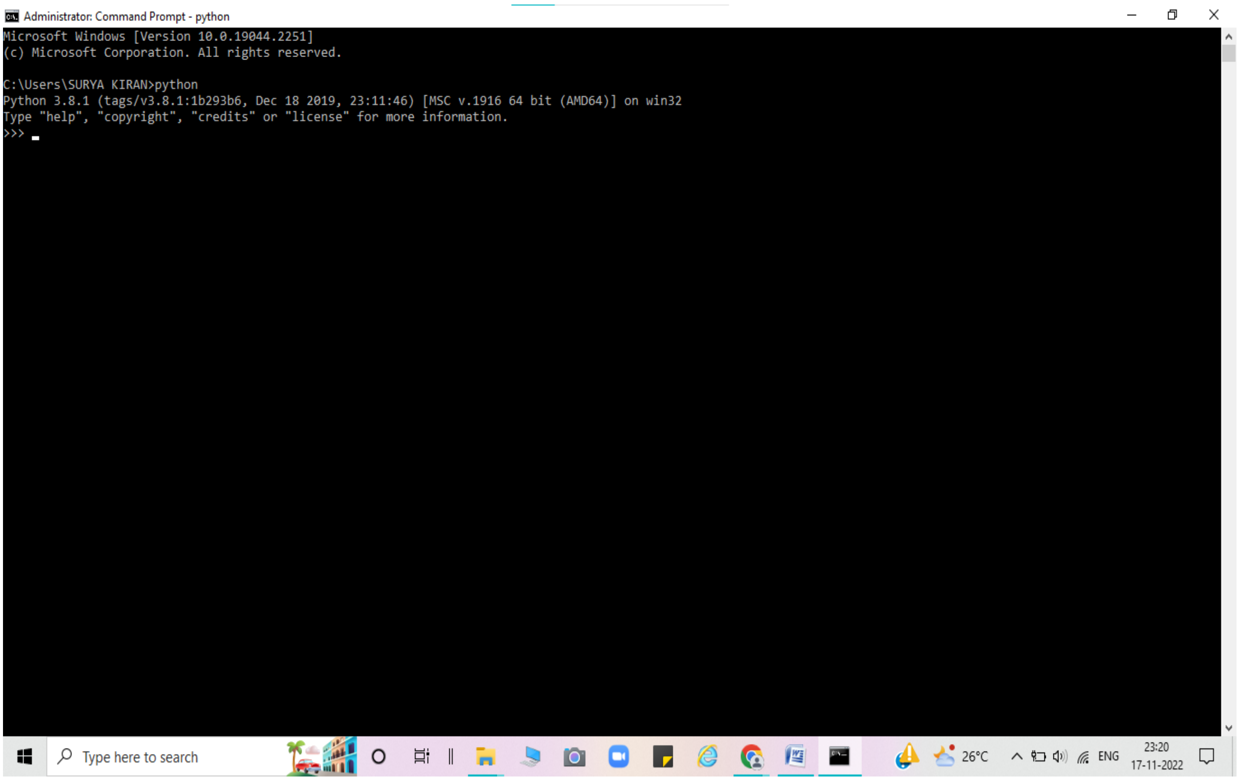
DEVELOP A PYTHON SCRIPT TO PUBLISH AND SUBSRIBE TO IBM IOT PLATFORM.

Team ID	PNT2022TMID24904
Project Name	Smart Farmer-IoT Enabled Smart Farming Application

Python IDE:

- Install Python3 compiler
- Install any python IDE to execute python scripts, in my case I usedSpyder to execute the code.



```
Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\SURYA KIRAN>python
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 23:11:46) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

PYTHON CODE:

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


 #Provide your IBM Watson Device Credentials
 organization = "tu54k3"
 deviceType = "sensor"
 deviceId = "sensor"
 authMethod = "use-
 token-auth"
 authToken = "12345678


 def myCommandCallback(cmd):
     print("Command received: %s" % cmd.data['command'])
     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:


    temperature=random.randint(15,10)
    humidity=random.randint(10,100)
    soil= random.randint(20,100)


    data = {'temperature' : temperature, 'humidity': humidity ,'soil':soil}
    #print data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temperature, "Humidity = %s
%%" % humidity, "soil Moisture = %s %%"% soil,"to IBM Watson")
```

```
        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(1)

        deviceCli.commandCallback = myCommandCallback

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