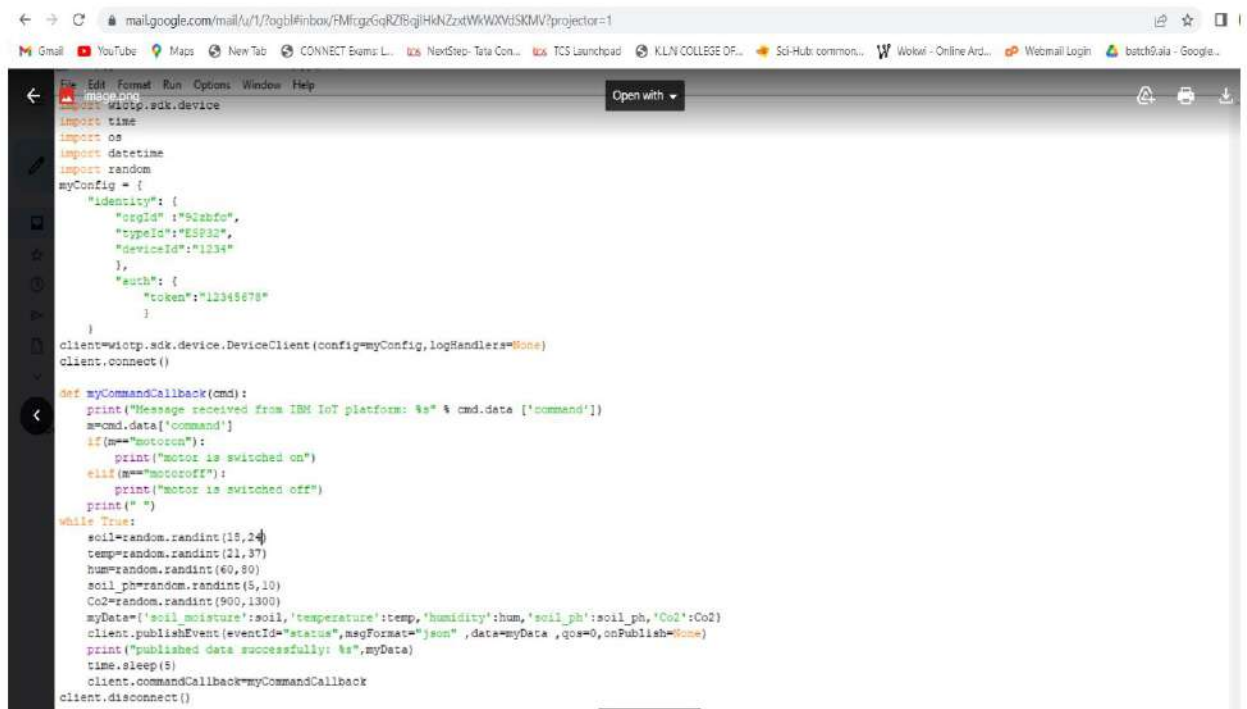


SPRINT 1- PYTHON PROGRAM TO GET CONNECTED WITH IBM IOT WATSON PLATFORM:



The screenshot shows a web browser window with a Google Mail page in the background. Overlaid on the browser is a code editor window displaying a Python script for connecting to the IBM IoT Watson Platform. The script includes imports for time, os, datetime, and random modules. It defines a configuration dictionary for a device and a client object. A callback function, myCommandCallback, is defined to handle incoming commands. The main loop generates random sensor data (soil moisture, temperature, humidity, soil pH, and CO2) and publishes it to the Watson IoT platform using the client's publishEvent method. The script also includes a sleep function to control the data publishing rate.

```
import time
import os
import datetime
import random

myConfig = {
    "identity": {
        "orgId": "92abfc",
        "typeId": "ESP32",
        "deviceId": "1234"
    },
    "auth": {
        "token": "12345678"
    }
}

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=="motoroft"):
        print("motor is switched off")
    print(" ")
while True:
    soil=random.randint(15,24)
    temp=random.randint(21,37)
    hum=random.randint(60,80)
    soil_ph=random.randint(5,10)
    Co2=random.randint(900,1300)
    myData={'soil_moisture':soil,'temperature':temp,'humidity':hum,'soil_ph':soil_ph,'Co2':Co2}
    client.publishEvent(eventId="status",msgFormat="json",data=myData ,qos=0,onPublish=None)
    print("published data successfully: %s",myData)
    time.sleep(5)
    client.commandCallback=myCommandCallback
client.disconnect()
```



The screenshot shows a Python 3.6.2 Shell window with the output of the Python script. The output indicates that the device successfully connected to the Watson IoT platform and published data. The data includes soil moisture, temperature, humidity, soil pH, and CO2 levels. The output is as follows:

```
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\kamalesh\Downloads\ibm p.py =====
2022-11-13 22:08:01.008 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:92abfc:ESP32:1234published data successfully: %s
{'soil_moisture': 24, 'temperature': 29, 'humidity': 63, 'soil_ph': 8, 'Co2': 1086}
published data successfully: %s {'soil_moisture': 20, 'temperature': 26, 'humidity': 66, 'soil_ph': 8, 'Co2': 1235}
```

