**Python Source Code**

| Team ID | PNT2022TMID41546 |
| --- | --- |
| Project Name | Smart Waste Management system for Metropolitan Cities |

# Source code

```
Import time
Import sys
Import ibmiotf.application
Import ibmiotf.device
Import random
#Provide your IBM Watson
Device Credentials
Organization = “wgsy43”
deviceType = “NodeMCU”
deviceId = “12345”
  authMethod = “use-token-auth”
  authToken = “12345678”
  # Initialize GPIO
  Def myCommandCallback(cmd):
  Print(“Command received: %s” % cmd.data[‘command’]
```

```
Status=cmd.data['command']If
status==”lighton”:
    Print (“led is on”)
  Else :
    Print (“led is off”)

  #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€)
        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)
       Data = { ‘temp’ : temp, ‘Humid’: Humid }
       #print data
        Def myOnPublishCallback():
        Print (“Published Temperature = %s C” % temp, “Humidity = %s %%” % Humid, “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()
```