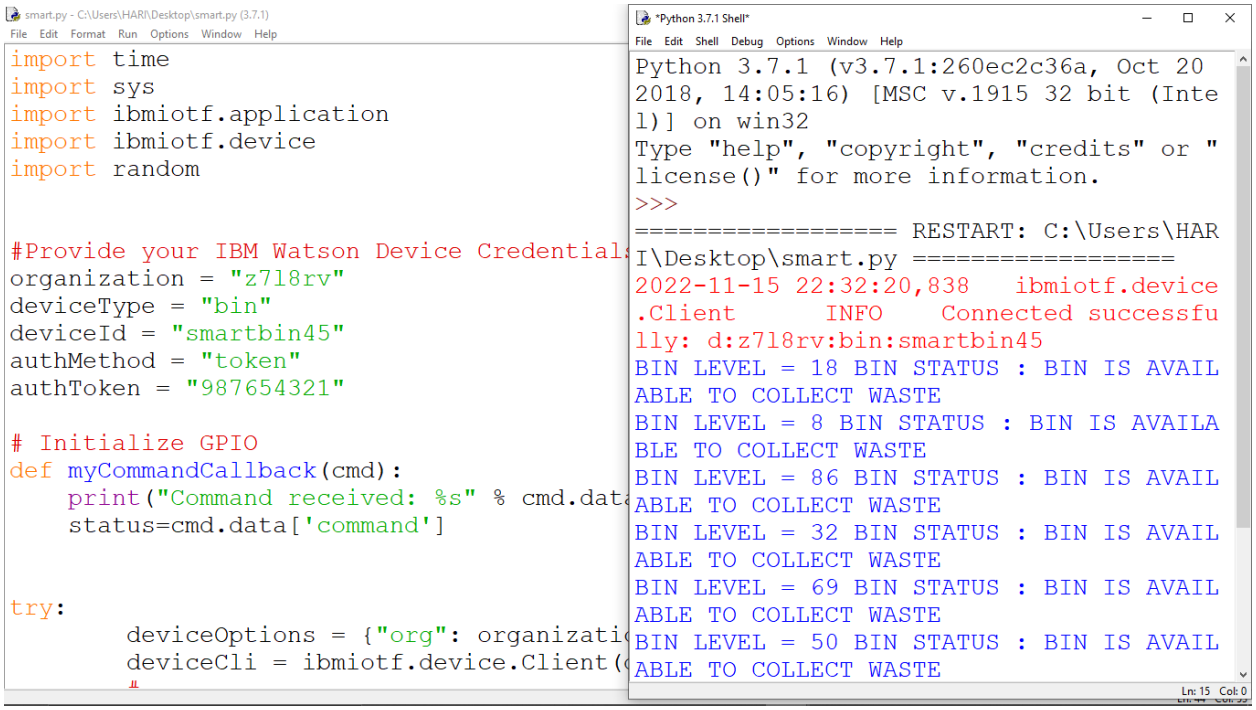
PYTHON SCRIPT

Team ID	PNT2022TMID03950
Project Name	Project – Smart waste Management System for Metropolitan cities

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


#Provide your IBM Watson Device Credentials
organization = "z7l8rv"
deviceType = "bin"
deviceId = "smartbin45"
authMethod = "token"
authToken = "987654321"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']


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 ultrosonic sensor
```

```
        t=random.randint(0,100)
        a="BIN IS GOING TO FULL" if t>=90 else "BIN IS AVAILABLE TO COLLECT WASTE"
        latitude=13.082680
        longitude=80.270721
        data = { 'BIN LEVEL' : t ,"status" : a,'latitude' : latitude,'longitude' : longitude}

        #print data
        def myOnPublishCallback():
            print ("BIN LEVEL = %s"% t,"BIN STATUS = %s"% a)



        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)

        if not success:
            print("Not connected to IoTF")
        time.sleep(10)

        deviceCli.commandCallback = myCommandCallback

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

PYTHON SIMULATION


smart.py - C:\Users\HARI\Desktop\smart.py (3.7.1)
File Edit Format Run Options Window Help
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "z7l8rv"
deviceType = "bin"
deviceId = "smartbin45"
authMethod = "token"
authToken = "987654321"
# Initialize GPIO
def myCommandCallback(cmd):
print("Command received: %s" % cmd.data
status=cmd.data['command']
try:
deviceOptions = {"org": organizati
deviceCli = ibmiotf.device.Client(
Python 3.7.1 Shell
File Edit Shell Debug Options Window Help
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20
2018, 14:05:16) [MSC v.1915 32 bit (Inte
l)] on win32
Type "help", "copyright", "credits" or "
license()" for more information.
>>>
========== RESTART: C:\Users\HAR
I\Desktop\smart.py ==========
2022-11-15 22:32:20,838 ibmiotf.device
.Client INFO Connected successfu
lly: d:z7l8rv:bin:smartbin45
BIN LEVEL = 18 BIN STATUS : BIN IS AVAIL
ABLE TO COLLECT WASTE
BIN LEVEL = 8 BIN STATUS : BIN IS AVAILA
BLE TO COLLECT WASTE
BIN LEVEL = 86 BIN STATUS : BIN IS AVAIL
ABLE TO COLLECT WASTE
BIN LEVEL = 32 BIN STATUS : BIN IS AVAIL
ABLE TO COLLECT WASTE
BIN LEVEL = 69 BIN STATUS : BIN IS AVAIL
ABLE TO COLLECT WASTE
BIN LEVEL = 50 BIN STATUS : BIN IS AVAIL
ABLE TO COLLECT WASTE
Ln: 15 Col: 0