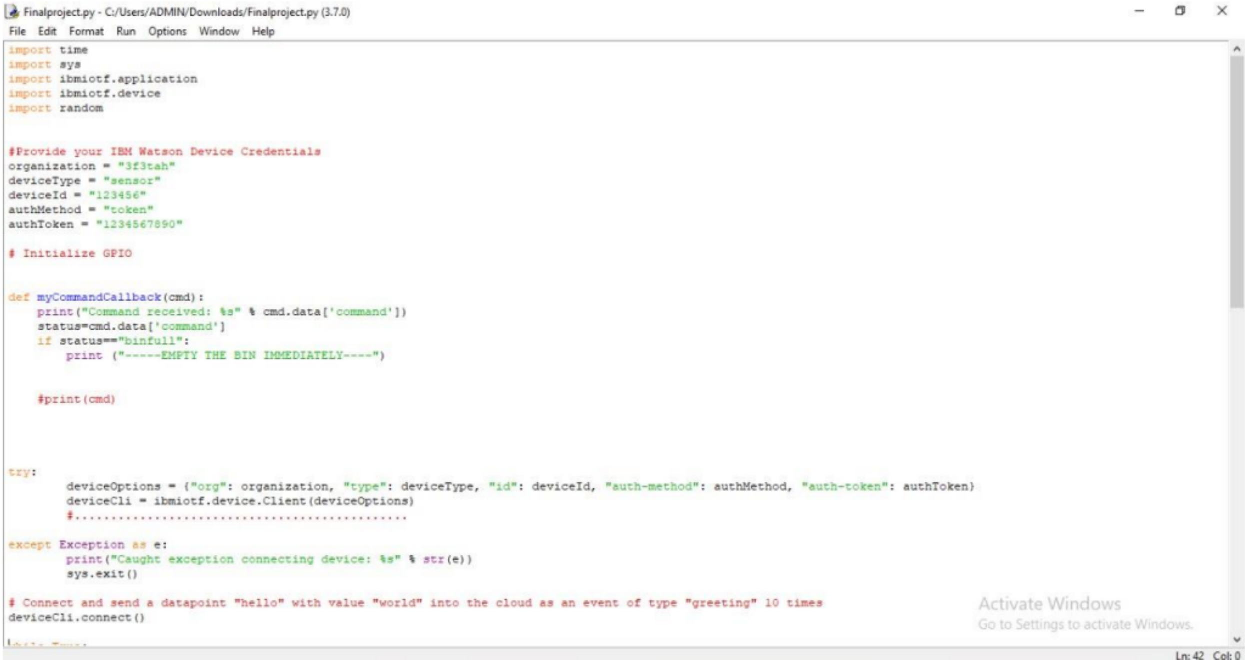
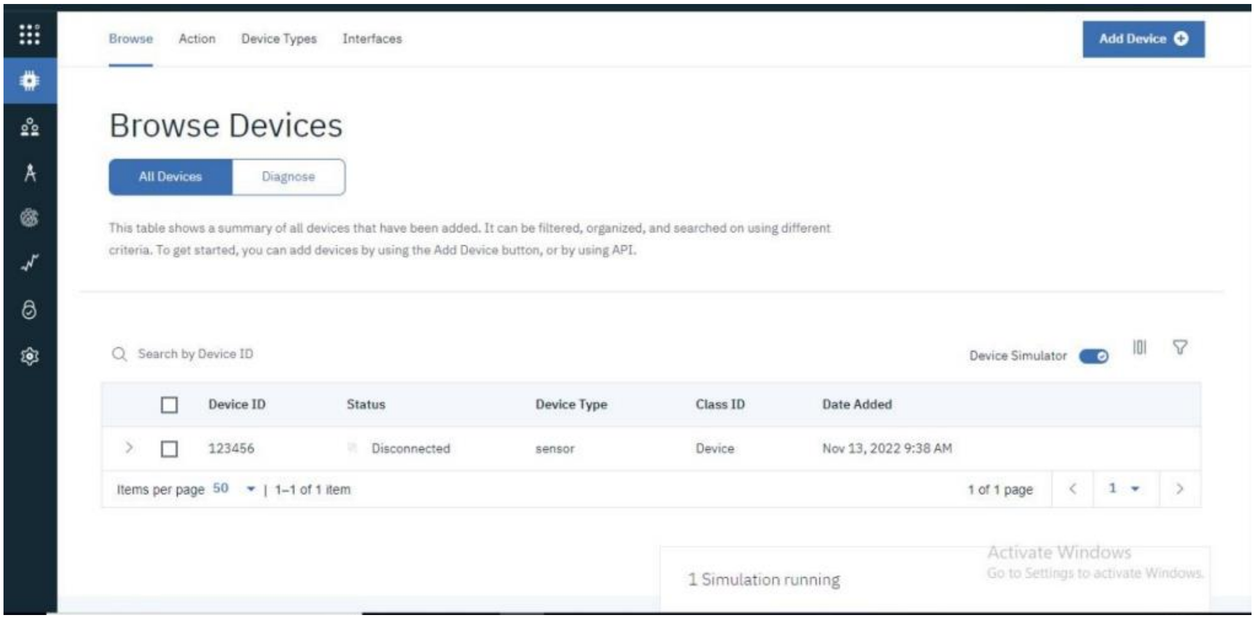
Sprint Delivery - 1

Team ID	PNT2022TMID11896
Project Name	Smart Waste Management System For Metropolitan

PYTHON CODE:


Finalproject.py - C:/Users/ADMIN/Downloads/Finalproject.py (3.7.0)
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 = "3f3tah"
deviceType = "sensor"
deviceId = "123456"
authMethod = "token"
authToken = "1234567890"

# Initialize GPIO


def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="binfull":
        print ("-----EMPTY THE BIN IMMEDIATELY----")


    #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:
```



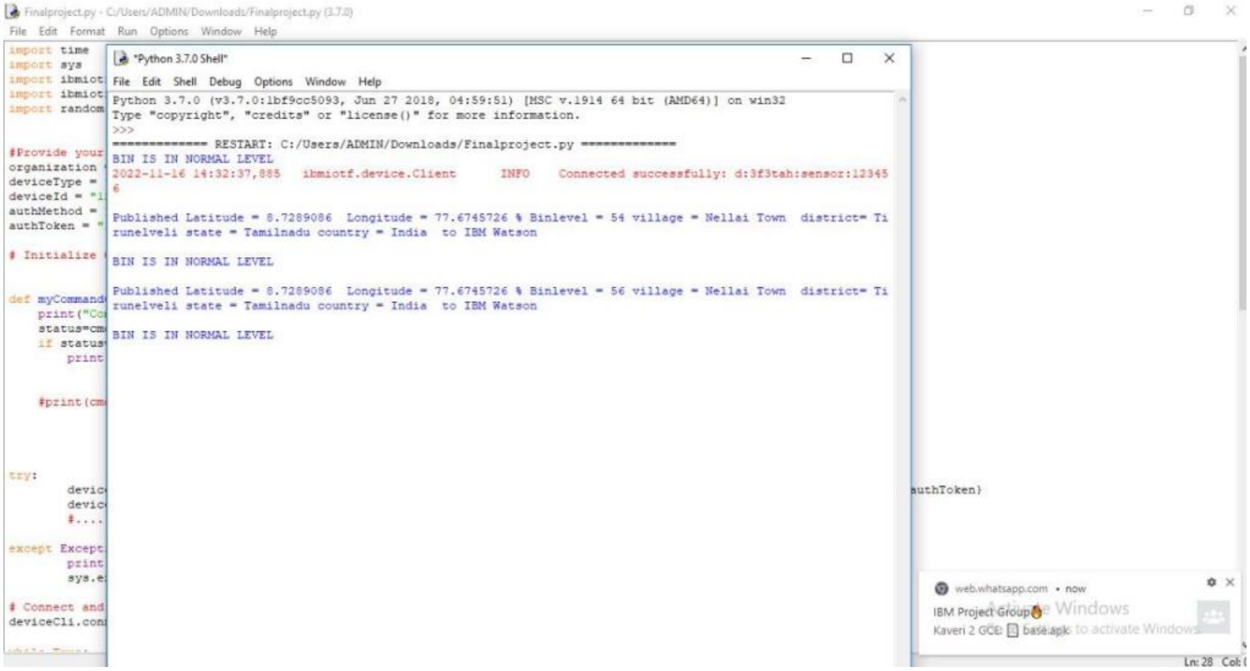
Browse Devices

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

	□	Device ID	Status	Device Type	Class ID	Date Added
>	□	123456	Disconnected	sensor	Device	Nov 13, 2022 9:38 AM

Items per page 50 | 1–1 of 1 item
1 of 1 page
1
1 Simulation running
Activate Windows
Go to Settings to activate Windows.

CODE OUTPUT:


Finalproject.py - C:/Users/ADMIN/Downloads/Finalproject.py (3.7.0)
File Edit Format Run Options Window Help
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
web.whatsapp.com • now
IBM Project Group
Kaveri 2 GCE: base.apk
Activate Windows
Ln: 28 Col: 0

```
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: C:/Users/ADMIN/Downloads/Finalproject.py =============
BIN IS IN NORMAL LEVEL
2022-11-16 14:32:37,885   ibmiotf.device.Client      INFO    Connected successfully: d:3f3tah:sensor:12345
6

Published Latitude = 8.7289086  Longitude = 77.6745726 % Binlevel = 54 village = Nellai Town  district= Ti
runelveli state = Tamilnadu country = India  to IBM Watson

BIN IS IN NORMAL LEVEL

Published Latitude = 8.7289086  Longitude = 77.6745726 % Binlevel = 56 village = Nellai Town  district= Ti
runelveli state = Tamilnadu country = India  to IBM Watson

BIN IS IN NORMAL LEVEL
```

PYTHON CODE:

```
import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

#Provide your IBM Watson Device Credentials

organization = "3f3tah"

deviceType = "sensor"

deviceId = "123456"

authMethod = "token"

authToken = "1234567890"

# Initialize GPIO
```

```
def myCommandCallback(cmd):
 print("Command received: %s" %
cmd.data['command'])
 status=cmd.data['command']
 if status=="binfull":
 print ("-----EMPTY THE BIN
IMMEDIATELY----")


 #print(cmd)


try:
deviceOptions = {"org": organization,
"type": deviceType, "id": deviceId, "authmethod": 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
```

```
binlevel=random.randint(10,100)
locationId=random.randint(1,5)
district="Tirunelveli"
state="Tamilnadu"
country="India"
if locationId == 1:
latitude=8.7060581
longitude=77.7633162
village="VM Chathiram"
elif locationId == 2:
latitude=8.7066676
longitude=77.732578
village="Perumalpuram"
elif locationId == 3:
latitude=8.7199159
longitude=77.725674
village="Palayamkottai"
elif locationId == 4:
latitude=8.7282671
longitude=77.7180244
village="Vannarpettai"
elif locationId == 5:
latitude=8.7289086
longitude=77.6745726
village="Nellai Town"
else:
print("No location Found!!")
```

```
 data = { 'latitude' : latitude, 'longitude':
longitude,'binlevel':
binlevel,'village':village,'district':district,'state':stat
e,'country':country }
 #print data
 def myOnPublishCallback():
 print ("Published Latitude = %s " %
latitude, "Longitude = %s %%" % longitude,
"Binlevel = %s" % binlevel,"village = %s " %
village,"district= %s" % district,"state = %s" %
state,"country = %s " % country, "to IBM
Watson\n")
 if binlevel >= 90:
 data={'Latitude':latitude,
'Longitude':longitude, 'Binlevel':binlevel,
'Village':village, 'District':district,
'State':state,'Country':country}

print("!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!BIN IS
FULL
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!\n")
 print("----------------------------------------
EMPTY THE BIN IMMEDIATELY----------------
------------------------------------\n")
 deviceCli.commandCallback =
myCommandCallback
 time.sleep(5)
 else:
 print("BIN IS IN NORMAL LEVEL\n")
 time.sleep(5)


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

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