PROJECT DEVELOPEMENT PHASE DELIVERY OF SPRINT – 3

Date	12 November 2022
Team ID	PNT2022TMID46424
Project Name	Project - Smart Waste Management System For Metropolitan Cities

PYTHON CODE: [To connect IBM WATSON]

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "cbseji"
deviceType = "abcd"
deviceId = "1234"
authMethod = "token"
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(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
      level=random.randint(0,100)
      weight=random.randint(0,100)
      data = { 'level' : level, 'weight': weight }
      #print data
      def myOnPublishCallback():
        print ("Published level = %s C" % level, "weight = %s %%"
 % weight, "to IBM Watson")
      success = deviceCli.publishEvent("IoTSensor", "json", data,
```

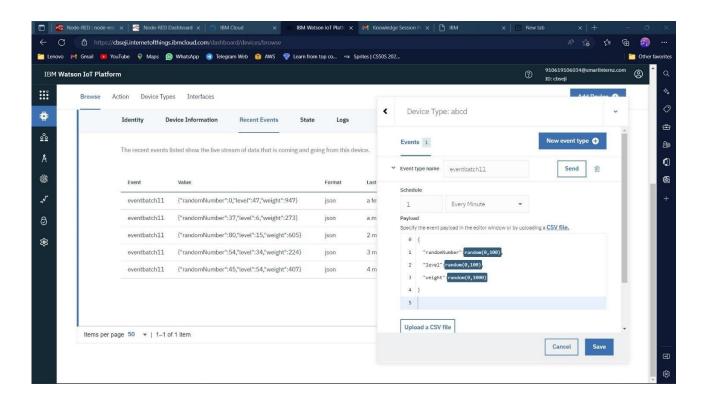
```
qos=0, on_publish=myOnPublishCallback)
if not success:
    print("Not connected to IoTF")
    time.sleep(1)
```

deviceCli.commandCallback = myCommandCallback

```
if (level>=75):
print("Full LED ON")
```

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

OUTPUT:



```
ee\Dropbox\PC\Downloads\ibmiotpublishsubscribe (1).py (3.7.0)
ibmiotpublishsubscribe (1).py - C:\Users\nav
 File Edit Format Run Options Window Help
import time
import sys
                                                          Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win3
import ibmiotf.application
import ibmiotf.device
                                                          Type "copyright", "credits" or "license()" for more information.
import random
                                                          = RESTART: C:\Users\navee\Dropbox\PC\Downloads\ibmiotpublishsubscribe (1).py =
                                                          2022-11-13 11:52:44,654 ibmiotf.device.Client INFO Connected successfully: d:cbseji:abc
#Provide your IBM Watson Device Credentials
                                                          d:1234
organization = "cbseji"
                                                          Published level = 82 C weight = 64 % to IBM Watson
deviceType = "abcd"
                                                          Full LED ON
deviceId = "1234"
                                                          Published level = 5 C weight = 2 % to IBM Watson
authMethod = "token"
                                                          Published level = 22 C weight = 57 % to IBM Watson
authToken = "12345678"
                                                          Published level = 83 C weight = 60 % to IBM Watson
                                                          Full LED ON
# Initialize GPIO
                                                          Published level = 16 C weight = 12 % to IBM Watson
                                                          Published level = 19 C weight = 91 % to IBM Watson
                                                          Published level = 35 C weight = 77 % to IBM Watson
def myCommandCallback(cmd):
                                                          Published level = 22 C weight = 46 % to IBM Watson
  print("Command received: %s" % cmd.data['command
                                                          Published level = 85 C weight = 68 % to IBM Watson
  status=cmd.data['command']
                                                          Full LED ON
  if status=="lighton":
                                                          Published level = 36 C weight = 88 % to IBM Watson
    print ("led is on")
                                                          Published level = 69 C weight = 72 % to IBM Watson
  else:
                                                          Published level = 14 C weight = 3 % to IBM Watson
    print ("led is off")
                                                          Published level = 99 C weight = 0 % to IBM Watson
  #print(cmd)
try:
         deviceOptions = {"org": organization, "type": de
```

