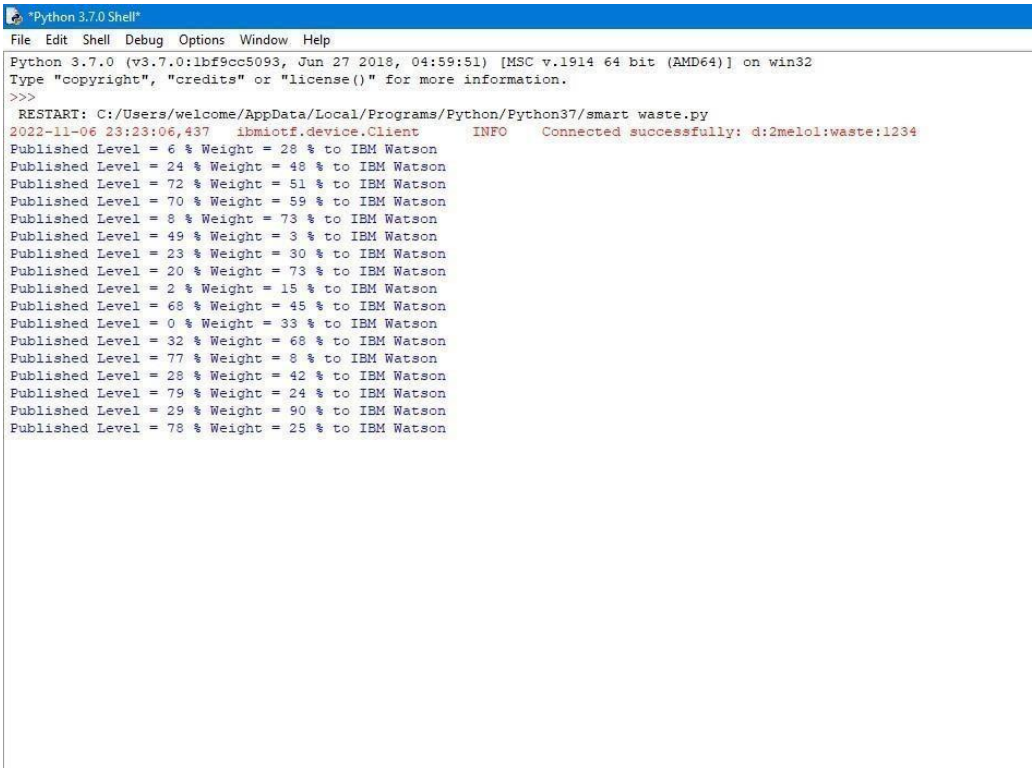
Team id:PNT2022TMID19083

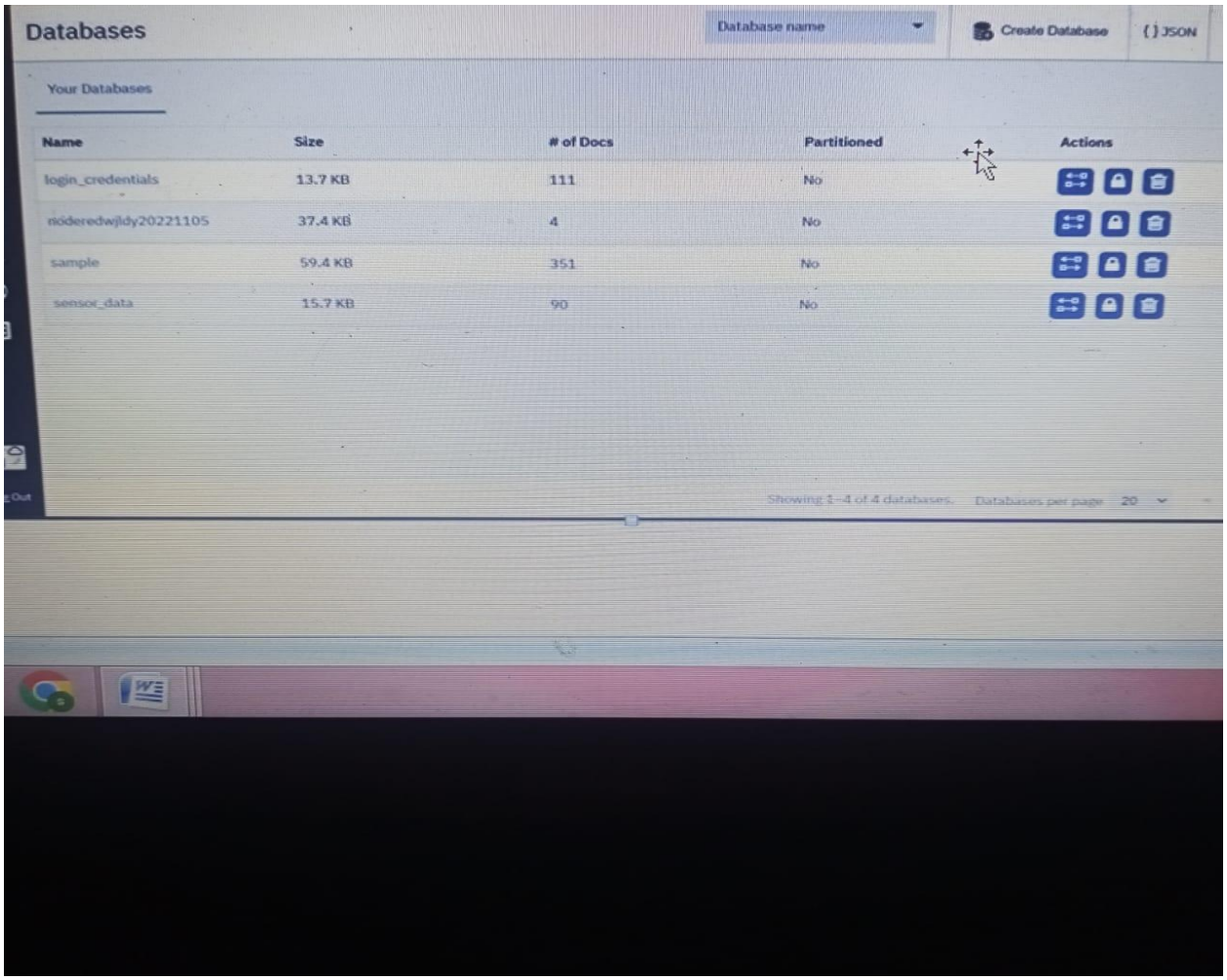
Python Code

```
import time import sys

import ibmiotf.application

import ibmiotf.device

importrandom


#Provide your IBM Watson Device Credentials organization

= "2melo1" deviceType =

"waste" deviceId = "1234"

authMethod ="token" authToken =

"12345678"


# Initialize GPIO


def myCommandCallback(cmd):

                                        print("Comman

dreceived: %s" % cmd.data['command'])

status=cmd.data['command']  if status=="waste

level":

      print ("waste level

   monitored")else :

      print ("weight level monitored")
```

```
   #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 %%" % level, "Weight = %s %%" % weight, "to IBM Watson")
```

```
        success = deviceCli.publishEvent("IoTSensor", "json",
 data, qos=0,on_publish=myOnPublishCallback)
        if not success:
        print("Notconnected      to
IoTF")
        time.sleep(20)
        deviceCli.commandCallbac
k=
        myCommandCallback


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




Databases
Database name
Create Database
{ } JSON
Your Databases
Name
Size
# of Docs
Partitioned
Actions
login_credentials
13.7 KB
111
No
noderedwjldy20221105
37.4 KB
4
No
sample
59.4 KB
351
No
sensor_data
15.7 KB
90
No
Showing 1–4 of 4 databases.
Databases per page
20