

### Sprint Delivery – 3

Team ID	PNT2022TMID05128
Project Name	Project - Smart Waste Management System for Meteropolitan Cities

In the last sprint phase(sprint - 2), a node-MCU device is created in Watson IOT Platform and randomly generated.

Now, using PYTHON CODE, we need to generate random values for Garbage weight, Level and Location of the Bin.

For that we need Python 3.7.0 IDLE. First of all, require the ibmiotf package using this command in the command prompt : **pip install ibmiotf**.

The required library files are included in the code ,then functions are written

#### CODE:

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

#Provide your IBM Watson Device Credentials
organization = "8wd932"
deviceType = "Node_Mcu"
deviceId = "123456789"
authMethod = "token"
authToken = "123456789"

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

```

status=cmd.data['command']
if status == "lighton":
    print("led in on")
else :
    print ("led is off")

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
    time.sleep(5)
    ult_son=random.randint(0,80)
    weight=random.randint(0,100)
    lat = round(random.uniform(11.03, 11.50), 6)
    long = round(random.uniform(76.80, 76.90), 6)
    gps = str(lat) + str(',') + str(long)
    data = {'Ultrasonic' : ult_son, 'Weight' : weight , 'GPS' : gps}
    #print data
    def myOnPublishCallback():
        print ("Published Ultrasonic = %s Cm" %ult_son, "Weight:%s kg" %weight, "GPS: %s" %gps)

```

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

```
    if not success:
```

```
        print("Not connected to IoTTF")
```

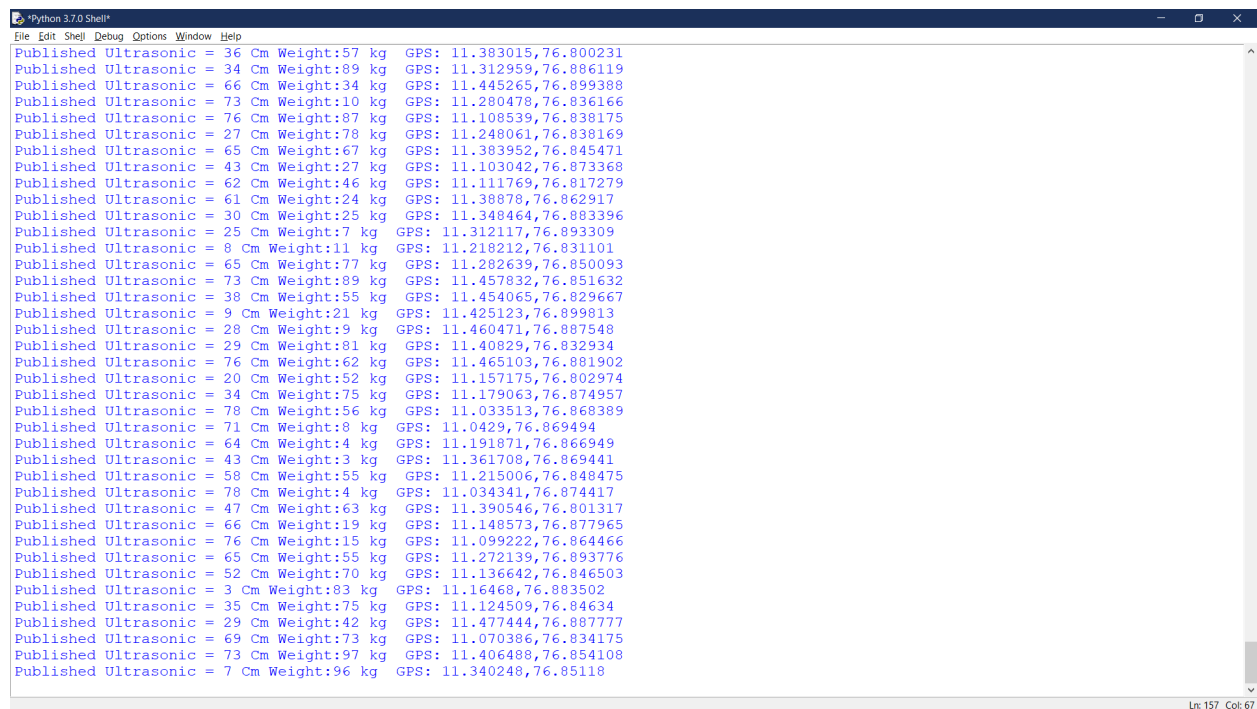
```
    time.sleep(1)
```

```
    deviceCli.commandCallback = myCommandCallback
```

```
# Disconnect the device and application from the cloud
```

```
deviceCli.disconnect()
```

## OUTPUT:



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Published Ultrasonic = 36 Cm Weight:57 kg GPS: 11.383015,76.800231
Published Ultrasonic = 34 Cm Weight:89 kg GPS: 11.312959,76.886119
Published Ultrasonic = 66 Cm Weight:34 kg GPS: 11.445265,76.899388
Published Ultrasonic = 73 Cm Weight:10 kg GPS: 11.280478,76.836166
Published Ultrasonic = 76 Cm Weight:87 kg GPS: 11.108539,76.838175
Published Ultrasonic = 27 Cm Weight:78 kg GPS: 11.248061,76.838169
Published Ultrasonic = 65 Cm Weight:67 kg GPS: 11.383952,76.845471
Published Ultrasonic = 43 Cm Weight:27 kg GPS: 11.103042,76.873368
Published Ultrasonic = 62 Cm Weight:46 kg GPS: 11.111769,76.817279
Published Ultrasonic = 61 Cm Weight:24 kg GPS: 11.38878,76.862917
Published Ultrasonic = 30 Cm Weight:25 kg GPS: 11.348464,76.883396
Published Ultrasonic = 25 Cm Weight:7 kg GPS: 11.312117,76.893309
Published Ultrasonic = 8 Cm Weight:11 kg GPS: 11.218212,76.831101
Published Ultrasonic = 65 Cm Weight:77 kg GPS: 11.282639,76.850093
Published Ultrasonic = 73 Cm Weight:89 kg GPS: 11.457832,76.851632
Published Ultrasonic = 38 Cm Weight:55 kg GPS: 11.454065,76.829667
Published Ultrasonic = 9 Cm Weight:21 kg GPS: 11.425123,76.899813
Published Ultrasonic = 28 Cm Weight:9 kg GPS: 11.460471,76.887548
Published Ultrasonic = 29 Cm Weight:81 kg GPS: 11.40829,76.832934
Published Ultrasonic = 76 Cm Weight:62 kg GPS: 11.465103,76.881902
Published Ultrasonic = 20 Cm Weight:52 kg GPS: 11.157175,76.802974
Published Ultrasonic = 34 Cm Weight:75 kg GPS: 11.179063,76.874957
Published Ultrasonic = 78 Cm Weight:56 kg GPS: 11.033513,76.868389
Published Ultrasonic = 71 Cm Weight:8 kg GPS: 11.0429,76.869494
Published Ultrasonic = 64 Cm Weight:4 kg GPS: 11.191871,76.866949
Published Ultrasonic = 43 Cm Weight:3 kg GPS: 11.361708,76.869441
Published Ultrasonic = 58 Cm Weight:55 kg GPS: 11.215006,76.848475
Published Ultrasonic = 78 Cm Weight:4 kg GPS: 11.034341,76.874417
Published Ultrasonic = 47 Cm Weight:63 kg GPS: 11.390546,76.801317
Published Ultrasonic = 66 Cm Weight:19 kg GPS: 11.148573,76.877965
Published Ultrasonic = 76 Cm Weight:15 kg GPS: 11.099222,76.864466
Published Ultrasonic = 65 Cm Weight:55 kg GPS: 11.272139,76.893776
Published Ultrasonic = 52 Cm Weight:70 kg GPS: 11.136642,76.846503
Published Ultrasonic = 3 Cm Weight:83 kg GPS: 11.16468,76.883502
Published Ultrasonic = 35 Cm Weight:75 kg GPS: 11.124509,76.84634
Published Ultrasonic = 29 Cm Weight:42 kg GPS: 11.477444,76.887777
Published Ultrasonic = 69 Cm Weight:73 kg GPS: 11.070386,76.834175
Published Ultrasonic = 73 Cm Weight:97 kg GPS: 11.406488,76.854108
Published Ultrasonic = 7 Cm Weight:96 kg GPS: 11.340248,76.85118
Ln: 157 Col: 67
```

These randomly generated values can be seen in the IBM Watson IOT Platform which is the IBM Cloud in the area of device recent events with the help of IBM ibmiotf package and the device credentials.

The screenshot shows the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. The main content area displays a device named '123456789' with a status of 'Disconnected'. Below this, a tabbed interface shows 'Identity', 'Device Information', 'Recent Events', 'State', and 'Logs'. The 'Recent Events' tab is active, showing a table of events. The table has columns for 'Event', 'Value', 'Format', and 'Last Received'. The events are listed as 'IoTSensor' with various sensor data values, all in 'json' format, and received 'a few seconds ago' or 'a minute ago'. The bottom of the dashboard shows a 'python code.pdf' file and a 'Show all' button.

Event	Value	Format	Last Received
IoTSensor	{"Ultrasonic ":"65," Weight ":"79," GPS ":"11.4911..."}	json	a few seconds ago
IoTSensor	{"Ultrasonic ":"45," Weight ":"18," GPS ":"11.3037..."}	json	a minute ago
IoTSensor	{"Ultrasonic ":"27," Weight ":"96," GPS ":"11.2514..."}	json	a minute ago
IoTSensor	{"Ultrasonic ":"13," Weight ":"21," GPS ":"11.1429..."}	json	a minute ago
IoTSensor	{"Ultrasonic ":"64," Weight ":"98," GPS ":"11.2969..."}	json	a minute ago

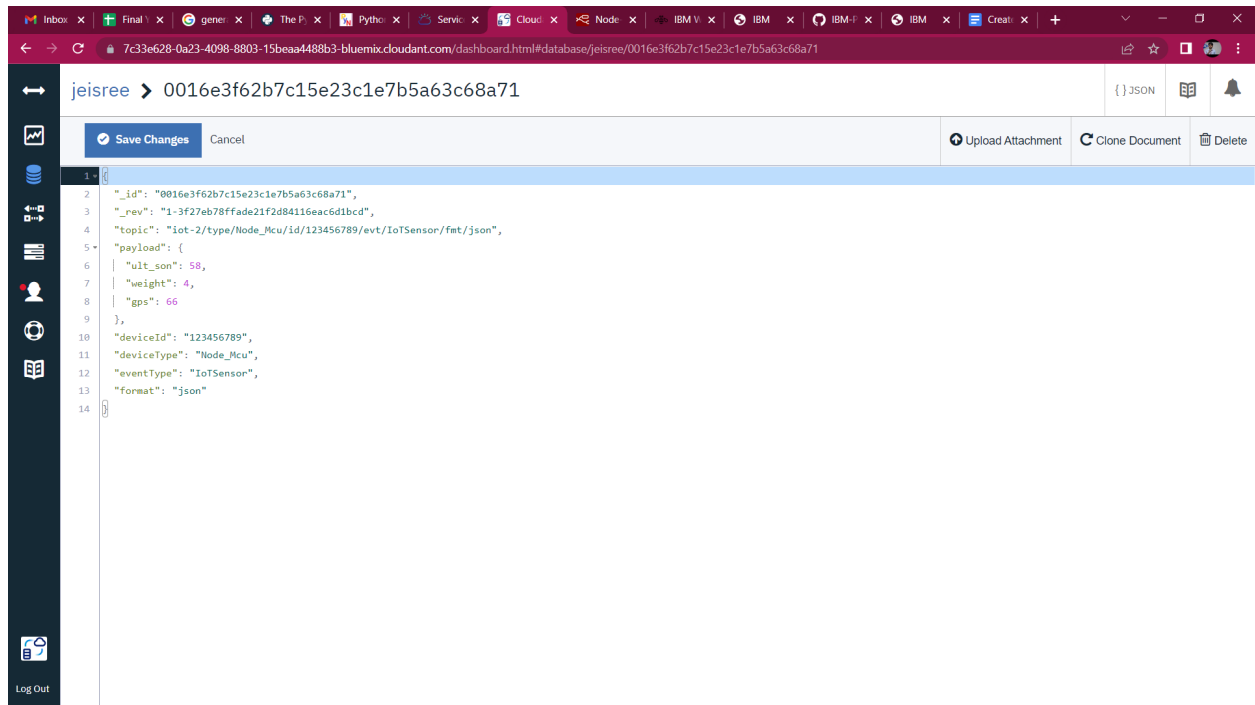
The values are stored in the IBM Cloudant database

Storing data in cloud:

The screenshot shows the IBM Cloudant database interface. The top navigation bar includes 'Inbox', 'Final', 'gener', 'The P...', 'Pytho', 'Servic', 'Cloud', 'Node', 'IBM V...', 'IBM', 'IBM', 'IBM', 'Creat', and '+'. The main content area displays a list of documents in a table view. The table has columns for 'id', 'key', and 'value'. The documents are listed with their IDs, keys, and values, all in 'json' format. The bottom of the dashboard shows a 'Showing document 1 - 20' message and a 'Documents per page: 20' dropdown.

id	key	value
0016e3f62b7c15e23c1e7b5a63c68a71	0016e3f62b7c15e23c1e7b5a63c68a71	{"rev": "1-3f27eb78fade21f2d84116eac6d1..."}
0016e3f62b7c15e23c1e7b5a63f84f6b	0016e3f62b7c15e23c1e7b5a63f84f6b	{"rev": "1-f6669174f89d6e7f7cfceda2e7242..."}
0016e3f62b7c15e23c1e7b5a63f8501b	0016e3f62b7c15e23c1e7b5a63f8501b	{"rev": "1-c214a24910f5ba9be4fa6f0afb88..."}
0287daa99d661fedea6b791e9bda7e09	0287daa99d661fedea6b791e9bda7e09	{"rev": "1-d789db870287944b5078d26e110..."}
0287daa99d661fedea6b791e9be096ef	0287daa99d661fedea6b791e9be096ef	{"rev": "1-a2d7d66ba56a7a1cc5104e91a5db..."}
0287daa99d661fedea6b791e9be14164	0287daa99d661fedea6b791e9be14164	{"rev": "1-7ac16f7ebe86339fb29f073179cf..."}
0287daa99d661fedea6b791e9be190e9	0287daa99d661fedea6b791e9be190e9	{"rev": "1-965049720b984431e9ba166ecd3..."}
0287daa99d661fedea6b791e9be296ae	0287daa99d661fedea6b791e9be296ae	{"rev": "1-be78a9697bc93f0e8f906311c01d..."}
0287daa99d661fedea6b791e9be31e46	0287daa99d661fedea6b791e9be31e46	{"rev": "1-568b19d9165ae273fb341f7fb087..."}
0287daa99d661fedea6b791e9be53b6c	0287daa99d661fedea6b791e9be53b6c	{"rev": "1-887d0de5e1e003227eb3c46bdfb3..."}
0287daa99d661fedea6b791e9bec2d70	0287daa99d661fedea6b791e9bec2d70	{"rev": "1-9a0e2541646db77d8695dea28b3..."}
0287daa99d661fedea6b791e9bef2238	0287daa99d661fedea6b791e9bef2238	{"rev": "1-4401b89c1f1771872e9aac8ff89c7..."}
0287daa99d661fedea6b791e9bf29a5b	0287daa99d661fedea6b791e9bf29a5b	{"rev": "1-1cc9c75b3a3e8c31c51aacbd2c3e..."}
0287daa99d661fedea6b791e9bf3be6b	0287daa99d661fedea6b791e9bf3be6b	{"rev": "1-7953ca4ab2dee473031b7473f014..."}

Actual Json format stored in cloud:



The screenshot shows a web browser window with multiple tabs. The active tab is titled "jeisree" and displays a JSON document. The URL in the address bar is "7c33e628-0a23-4098-8803-15beaa4488b3-bluemix.cloudant.com/dashboard.html#database/jeisree/0016e3f62b7c15e23c1e7b5a63c68a71". The JSON document is displayed in a text editor with a dark background and light-colored text. The document contains a single JSON object with the following fields: "\_id", "\_rev", "topic", "payload", "deviceId", "deviceType", "eventType", and "format". The "payload" field is an object containing "uit\_son", "weight", and "gps".

```
1 {
2   "_id": "0016e3f62b7c15e23c1e7b5a63c68a71",
3   "_rev": "1-3f27eb78ffade21f2d84116eac6d1bcd",
4   "topic": "iot-2/type/Node_Mcu/1d/123456789/evt/IoTSensor/fmt/json",
5   "payload": {
6     | "uit_son": 58,
7     | "weight": 4,
8     | "gps": 66
9   },
10  "deviceId": "123456789",
11  "deviceType": "Node_Mcu",
12  "eventType": "IoTSensor",
13  "format": "json"
14 }
```