

SPRINT - 4

Date	13 NOV 2022
Team ID	PNT2022TMID48570
Project Name	Smart Waste Management System for Metropolitan Cities

1, Simulate python code in Python IDE software to transmit data to IBM Watson IOT platform

Python code:

smartbin.py:

```
#Project: Smart Waste Management System for Metropolitan cities  
#Team ID: PNT2022TMID53567
```

```
#Installing necessary libraries
```

```
import wiotp.sdk.device  
import time import random  
import requests  
import math
```

```
#Configuration details for connecting python script to IBM Watson IOT Platform
```

```
myConfig = {  
    "identity": {  
        "orgId": "mldk59",  
        "typeId": "pythoncode",  
        "deviceId": "252525"  
    },  
    "auth": {  
        "token": "QZqODYo6U*Q6b+IpuC"  
    }  
}
```

```
def myCommandCallback(cmd):  
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])  
    m=cmd.data['command']
```

#Connecting the client to ibm watson iot platform

```
client = wiotp.sdk.device.DeviceClient(config=myConfig,logHandlers=None)
client.connect()
```

#Generate Random values for latitude, longitude in a circular distribution from the current location and

#alert the garbage collector to go to the particular location where the bin level and bin weight exceeds the threshold

while True:

```
    res = requests.get('https://ipinfo.io/')
    data = res.json()    loc =
    data['loc'].split(',')    theta =
    random.uniform(0,2*math.pi)    area =
    (0.05**2)*math.pi
    radius = math.sqrt(random.uniform(0,area/math.pi))
    latitude,longitude = [float(loc[0])+radius*math.cos(theta), float(loc[1])+radius*
    math.sin(theta)]

    binlevel=random.randint(10,100)
    binweight = random.randint(50,1500)

    if binweight>=1000 and binlevel>80:
        myData={'latitude':latitude, 'longitude':longitude,'binlevel':binlevel,
        'binweight':binweight}
        client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=
0, onPublish=None)
        ##print("Published data Successfully: %s", myData)
        print("BIN IS FULL..TIME TO EMPTY IT!!!\n",myData)
        client.commandCallback = myCommandCallback    time.sleep(2)
        #break

    else :
        print("BIN IS IN NORMAL LEVEL...")
        time.sleep(2)
```

Python IDE output:

```
JupyterLab - C:\Users\bgaya\Downloads\Scripts\smartbin.py (3.9.7)Python 3.9.7 ('tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit AMD...File Edit Format Run Options Window Help#Project: Smart Waste Management System for Metropolitan cities#Team ID: PNT2022TMD53567#Installing necessary librariesimport wiotp.sdk.deviceimport timeimport randomimport requestsimport math#Configuration details for connecting python script to IBM Watson IoT PlatformmyConfig = {'identity': { 'orgid': "mldkS5", 'typeId': "pythoncode", 'deviceid': "252525"}, }, "auth": { 'token': "QGqDOfEQU@Gb+IpuC"} }def myCommandCallback(cmd): print("Message received from IBM IoT Platform: %s" % cmd.data['command']) m=cmd.data['command']#Connecting the client to ibm watson iot platformclient = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)client.connect()#Generate random values for latitude, longitude in a circular distribution from the current location andset the garbage collector to go to the particular location where the bin level and bin weight exceeds thwhile True: res = requests.get('https://ipinfo.io/') data = res.json() loc = data['loc'].split(',') theta = random.uniform(0,2*math.pi) area = (0.05**2)*math.pi radius = math.sqrt(random.uniform(0,area/math.pi)) latitude,longitude = [float(loc[0])+radius*math.cos(theta), float(loc[1]+radius*math.sin(theta))] binlevel=random.randint(10,100) binweight = random.randint(50,1500)
```

2. Data is transferred to IBM Watson IoT platform.

IBM Platform output:

The screenshot displays the IBM Watson IoT Platform interface. At the top, the browser address bar shows the URL: `mdlk59.internetofthings.ibmcloud.com/dashboard/devices/browse`. The platform header includes the logo, navigation tabs (Browse, Action, Device Types, Interfaces), and a user profile section for '2019ec0032@svce.ac.in' with ID 'mdlk59'. A blue 'Add Device' button is visible on the right.

The main content area shows a list of devices. The selected device has the following details:

- Device ID:** 252525
- Status:** Connected
- Device Type:** pythoncode
- Class ID:** Device
- Date Added:** Nov 5, 2022 8:24 PM
- Descriptive Location:** (empty)

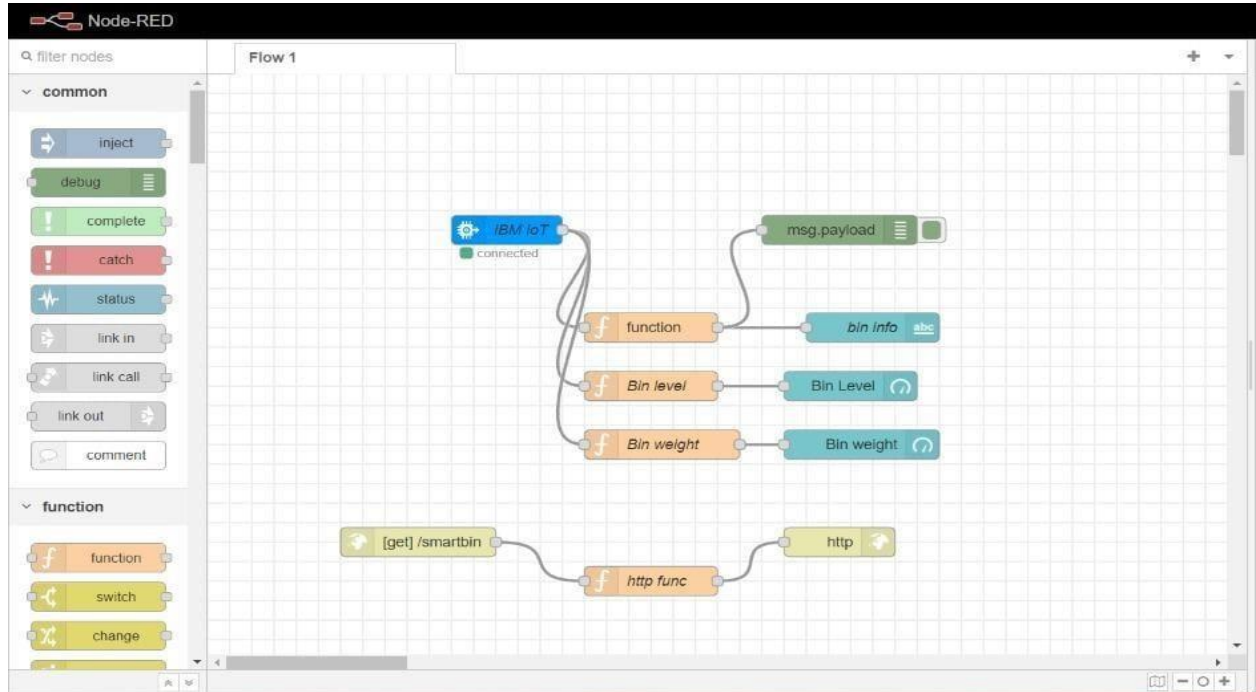
Below the device details, there are tabs for Identity, Device Information, Recent Events, State, and Logs. The 'Recent Events' tab is active, showing a message: 'The recent events listed show the live stream of data that is coming and going from this device.'

A table lists the recent events:

Event	Value	Format	Last Received
status	<code>{"latitude":93.38291147072071,"longitude":85....</code>	json	a few seconds ago
status	<code>{"latitude":123.34633147794314,"longitude":7....</code>	json	a few seconds ago
status	<code>{"latitude":120.69034946242466,"longitude":9....</code>	json	2 minutes ago
status	<code>{"latitude":82.92484862339958,"longitude":93....</code>	json	2 minutes ago
status	<code>{"latitude":44.682952261624024,"longitude":9....</code>	json	2 minutes ago

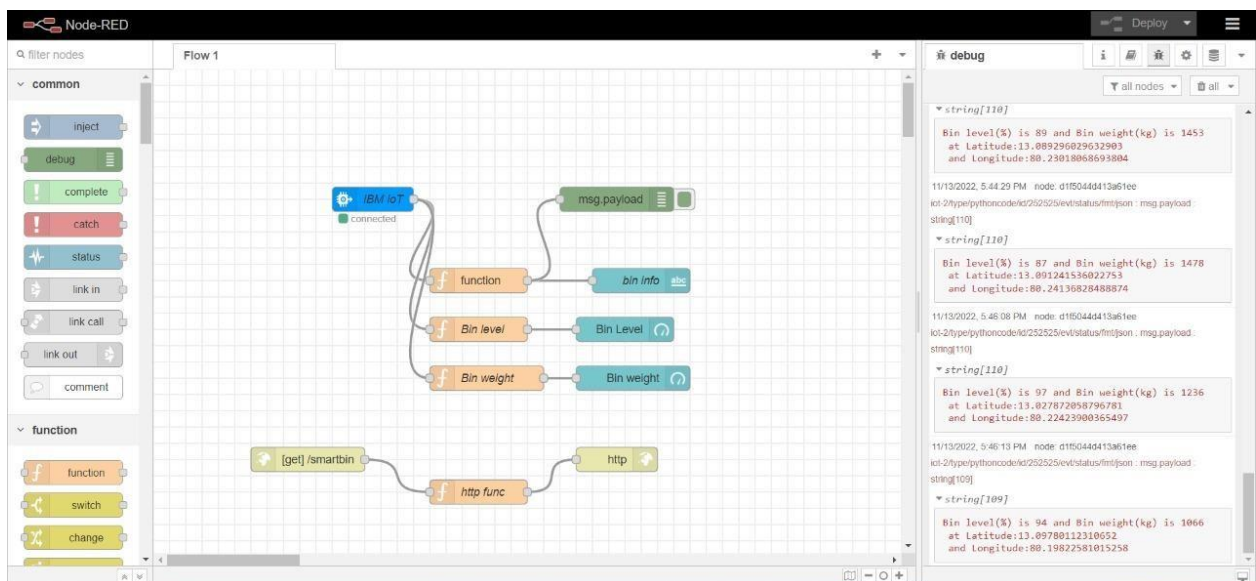
3. Data transfer from IBM Watson IOT platform and Python IDE to Node RED.

Node-RED:



4. Node-RED Connection setup for data transmission from IBM Watson IoT platform to Node-RED dashboard and viewing in Web UI .

Node-RED:



5. Storing database in IBM Cloudant DB

The screenshot shows the IBM Cloudant Databases interface. On the left is a sidebar with navigation icons. The main area is titled "Databases" and contains a table of "Your Databases". The table has columns: Name, Size, # of Docs, Partitioned, and Actions. There are four databases listed: login_credentials, noderedwjdty20221105, sample, and sensor_data. Each database has a size, number of documents, and is marked as "No" for partitioned. The Actions column contains icons for adding, deleting, and locking documents. At the bottom, it says "Showing 1-4 of 4 databases. Databases per page: 20".

Name	Size	# of Docs	Partitioned	Actions
login_credentials	13.7 KB	111	No	[Add] [Delete] [Lock]
noderedwjdty20221105	37.4 KB	4	No	[Add] [Delete] [Lock]
sample	59.4 KB	351	No	[Add] [Delete] [Lock]
sensor_data	15.7 KB	90	No	[Add] [Delete] [Lock]

The screenshot shows the IBM Cloudant interface for the "sensor_data" database. The left sidebar has a "Query" section. The main area shows a list of documents in JSON format. The table has columns: id, key, and value. The documents are listed with their IDs and keys, and the values are JSON objects. At the bottom, it says "Showing document 1 - 20. Documents per page: 20".

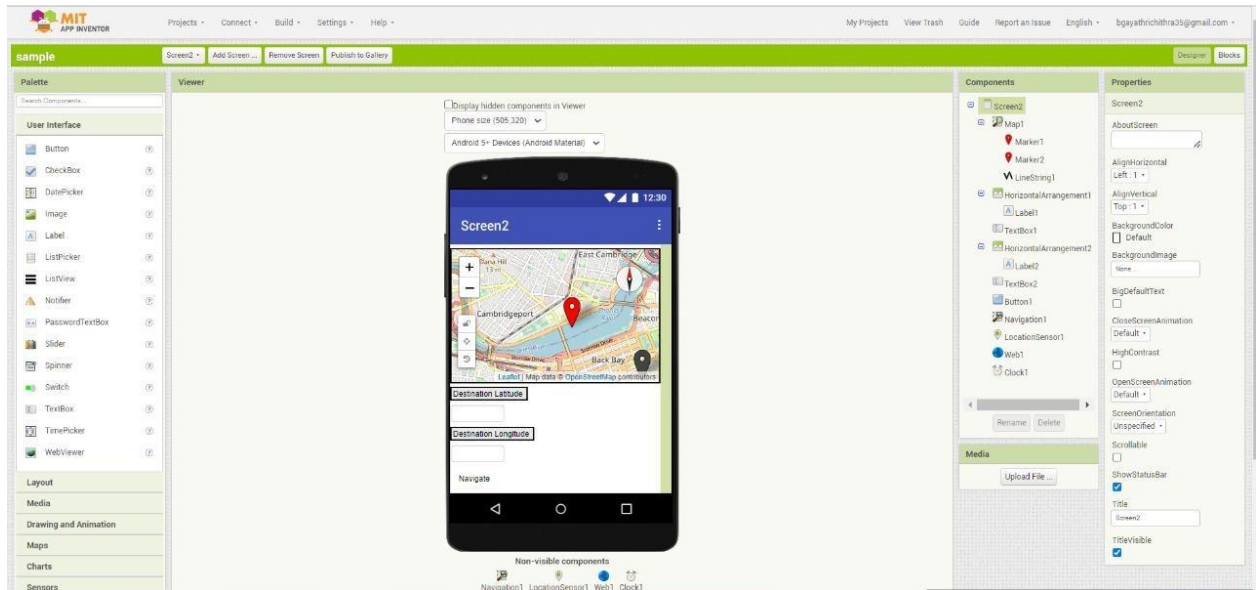
id	key	value
0198213c192cb2c244cc2433f1...	0198213c192cb2c244cc2433f1...	{ "rev": "1-cde2dd17c519394df..." }
0198213c192cb2c244cc2433f1...	0198213c192cb2c244cc2433f1...	{ "rev": "1-d26c5b40891e13c6c..." }
0198213c192cb2c244cc2433f1...	0198213c192cb2c244cc2433f1...	{ "rev": "1-cde2dd17c519394df..." }
0198213c192cb2c244cc2433f1...	0198213c192cb2c244cc2433f1...	{ "rev": "1-f96eb0460bc16cfab0..." }
1a921f21cbe229b86f599acb45...	1a921f21cbe229b86f599acb45...	{ "rev": "1-7226f08794cd47b7c..." }
1a921f21cbe229b86f599acb45...	1a921f21cbe229b86f599acb45...	{ "rev": "1-1bbdd9a985bd56cf9..." }
20a854e5445fa818e6c1de049...	20a854e5445fa818e6c1de049...	{ "rev": "1-7226f08794cd47b7c..." }
20a854e5445fa818e6c1de049...	20a854e5445fa818e6c1de049...	{ "rev": "1-3ad288ecad57f039e..." }
20a854e5445fa818e6c1de049...	20a854e5445fa818e6c1de049...	{ "rev": "1-1bbdd9a985bd56cf9..." }
298ed6f9b3b815f5ac7c061a...	298ed6f9b3b815f5ac7c061a...	{ "rev": "1-4e7240f6e5307a1b9..." }

6. Data is stored in JSON format

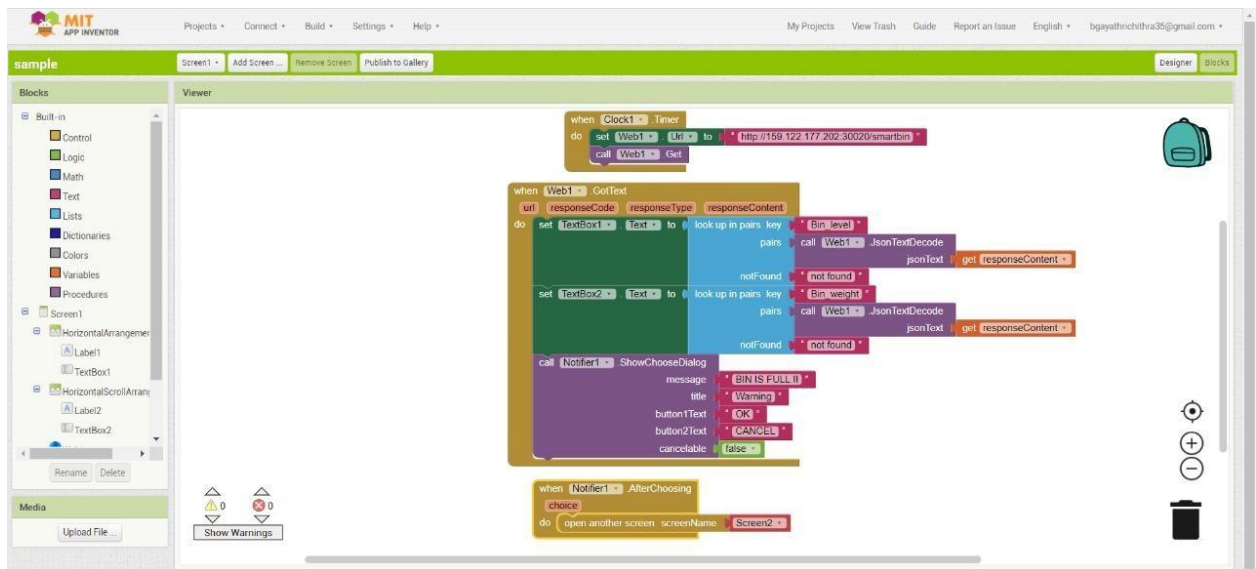
The screenshot shows the IBM Cloudant interface for the "sensor_data" database, displaying a single document in JSON format. The document is a JSON object with the following structure:

```
{
  "_id": "0198213c192cb2c244cc2433f1802b91",
  "_rev": "1-cde2dd17c519394df774730c495f8b",
  "topic": "iot-2/type/SWMSGHC/id/ibmproject/evt/data/fmt/json",
  "payload": {
    "Warning!!": "244.971e6"
  },
  "deviceId": "ibmproject",
  "deviceType": "SWMSGHC",
  "eventType": "data",
  "format": "json"
}
```

Web UI:



Screen 1 blocks:



Screen 2 blocks:



8. Install MIT AI2 Companion in phone and scan the QR code showed in AI connect

