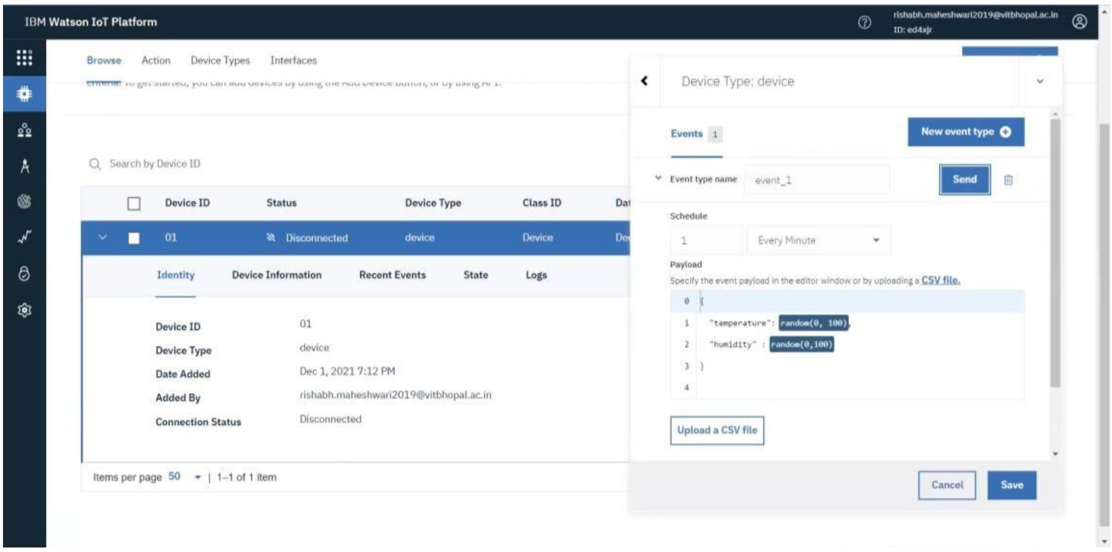
Assignment 3

Develop a code to upload the water tank level and light intensity values to the IBM IoT platform and visualize them in the web application.

Procedure:

1. Open Watson IoT Platform from Services in IBM Cloud. Create a new Simulation on a existing/new Device.



IBM Watson IoT Platform

rishabh.maheshwari2019@vitbhopal.ac.in
ID: ed4xjr

Browse Action Device Types Interfaces

Search by Device ID

□	Device ID	Status	Device Type	Class ID	Dat
□	01	Disconnected	device	Device	De

Identity | Device Information | Recent Events | State | Logs

Device ID	01
Device Type	device
Date Added	Dec 1, 2021 7:12 PM
Added By	rishabh.maheshwari2019@vitbhopal.ac.in
Connection Status	Disconnected

Items per page 50 | 1–1 of 1 item

Device Type: device

Events 1 New event type

Event type name: event_1 Send

Schedule: 1 Every Minute

Payload

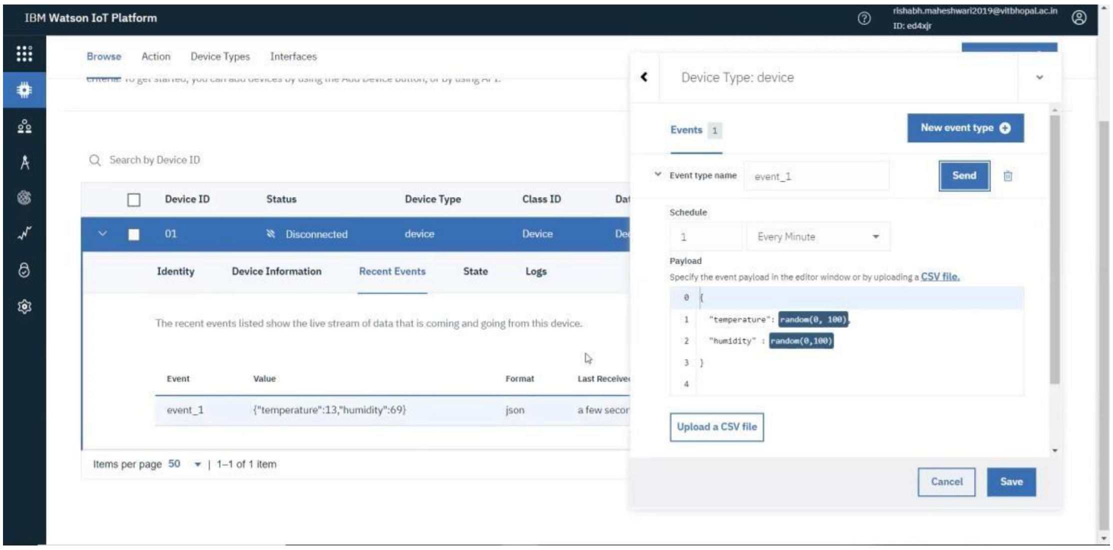
Specify the event payload in the editor window or by uploading a CSV file.

```
{
  "temperature": random(0, 100),
  "humidity" : random(0,100)
}
```

Upload a CSV file

Cancel Save

2. To verify that the service is running Run the service manually and check the output in Recent Events.



Device ID	Status	Device Type	Class ID	Da
01	Disconnected	device	Device	De

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Receive
event_1	{"temperature":13,"humidity":69}	json	a few secor

Items per page 50 | 1–1 of 1 item

Event type name: event_1

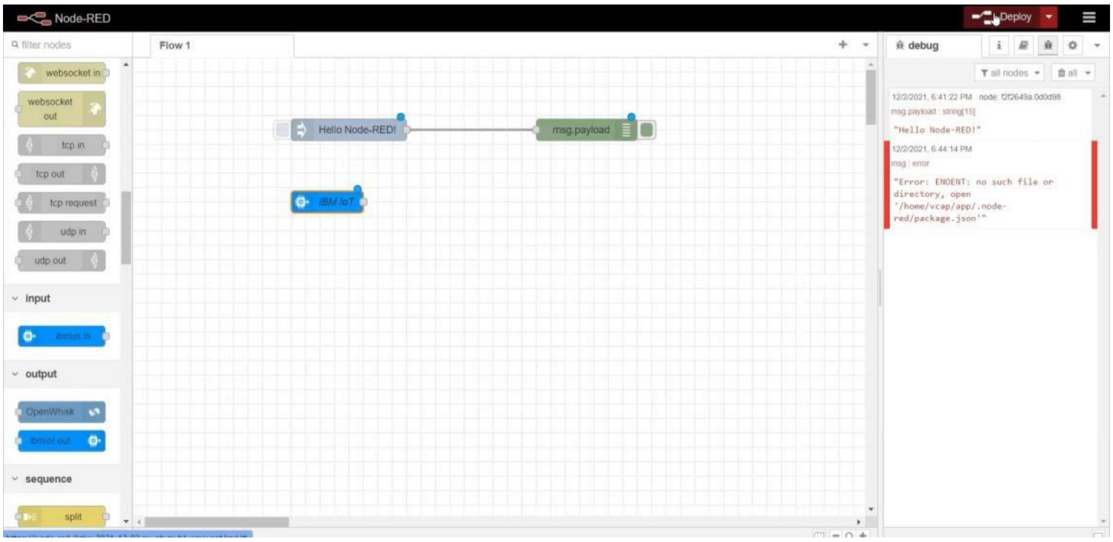
Schedule: 1 Every Minute

Payload

Specify the event payload in the editor window or by uploading a CSV file.

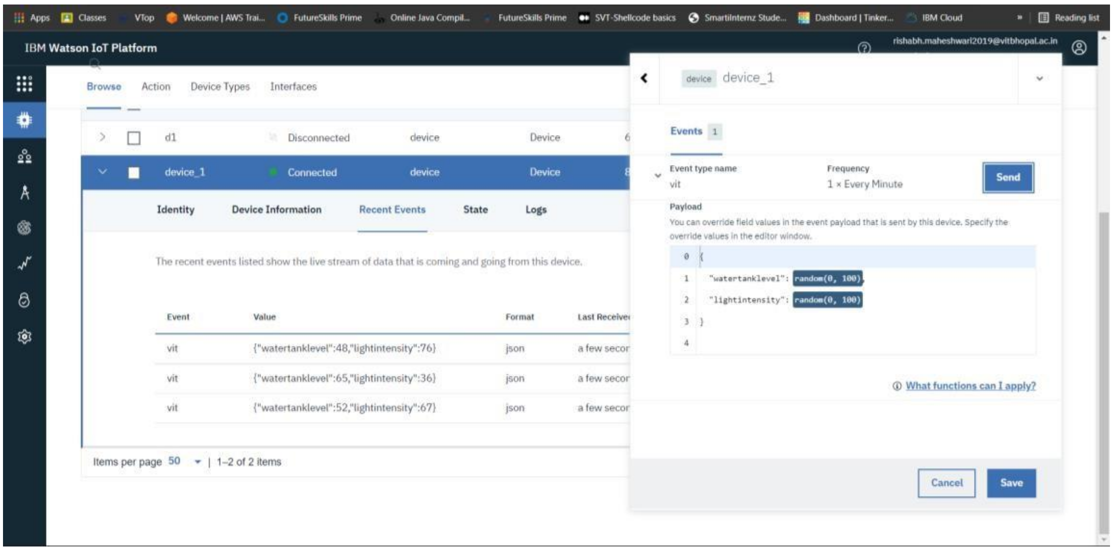
```
{
  "temperature": random(0, 100),
  "humidity" : random(0,100)
}
```

3. Launch Node-RED app and add an Input pallet, You should see an ERROR message on the Debug console as shown below.



```
12/2/2021, 6:41:22 PM  node: f2f2649a.0d0d98
msg.payload : string[15]
"Hello Node-RED!"
12/2/2021, 6:44:14 PM
msg : error
"Error: ENOENT: no such file or
directory, open
'/home/vcap/app/.node-
red/package.json'"
```

4. Edit the JSON data in the Watson IoT Platform.



d1	Disconnected	device	Device	6
device_1	Connected	device	Device	8

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Receive
vit	{"watertanklevel":48,"lightintensity":76}	json	a few secor
vit	{"watertanklevel":65,"lightintensity":36}	json	a few secor
vit	{"watertanklevel":52,"lightintensity":67}	json	a few secor

Items per page 50 | 1–2 of 2 items

device device_1

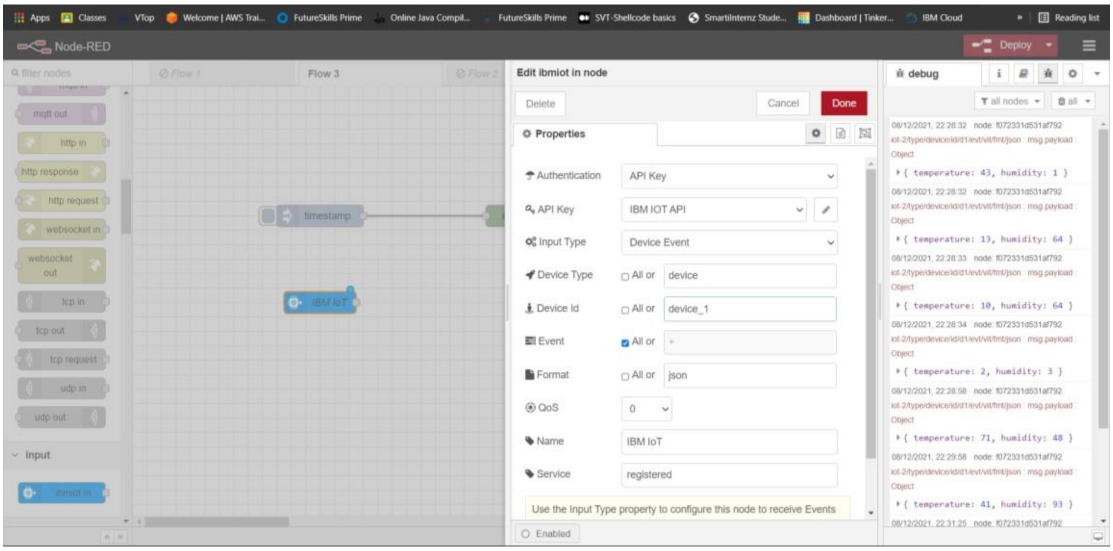
Events 1

Event type name: vit
Frequency: 1 × Every Minute

Payload
You can override field values in the event payload that is sent by this device. Specify the override values in the editor window.

```
{
  "watertanklevel": random(0, 100),
  "lightintensity": random(0, 100)
}
```

5. Configure the INPUT pallet. Add the details related to your Watson IoT Platform along with the API key generated on the same.



Edit ibmiot in node

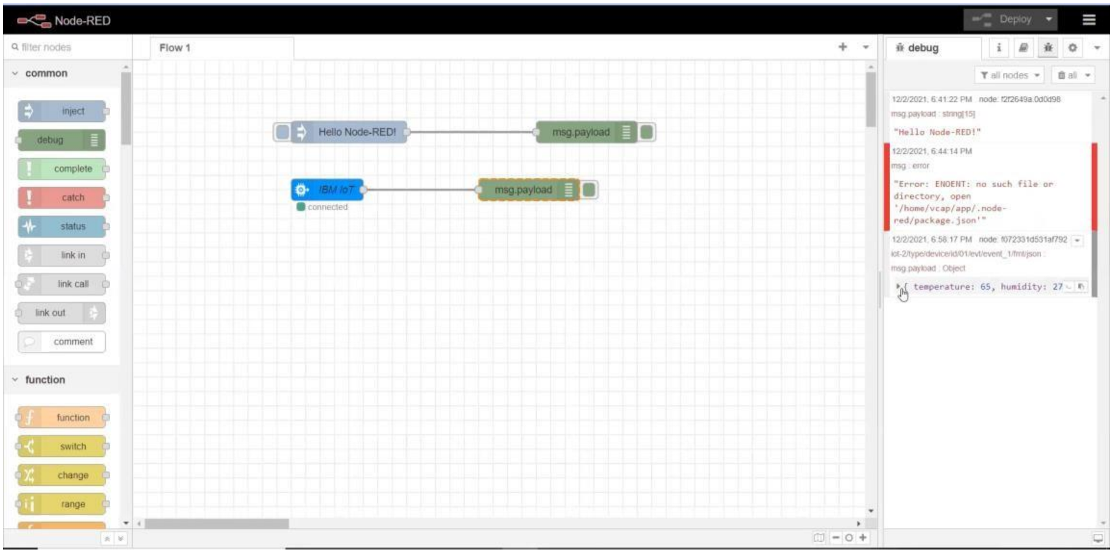
Authentication	API Key
API Key	IBM IOT API
Input Type	Device Event
Device Type	□ All or device
Device Id	□ All or device_1
Event	☑ All or +
Format	□ All or json
QoS	0
Name	IBM IoT
Service	registered

Use the Input Type property to configure this node to receive Events

○ Enabled

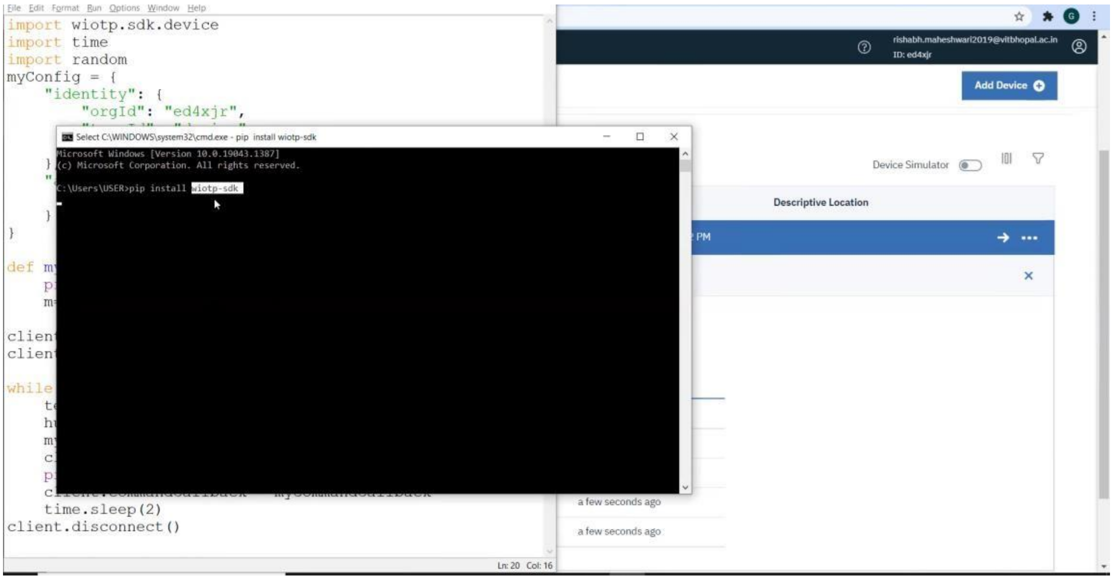
```
08/12/2021, 22:28:32  node: f072331d531af792
iot-2/type/device/id/d1/evt/vit/fmt/json : msg.payload :
Object
▸{ temperature: 43, humidity: 1 }
08/12/2021, 22:28:32  node: f072331d531af792
iot-2/type/device/id/d1/evt/vit/fmt/json : msg.payload :
Object
▸{ temperature: 13, humidity: 64 }
08/12/2021, 22:28:33  node: f072331d531af792
iot-2/type/device/id/d1/evt/vit/fmt/json : msg.payload :
Object
▸{ temperature: 10, humidity: 64 }
08/12/2021, 22:28:34  node: f072331d531af792
iot-2/type/device/id/d1/evt/vit/fmt/json : msg.payload :
Object
▸{ temperature: 2, humidity: 3 }
08/12/2021, 22:28:58  node: f072331d531af792
iot-2/type/device/id/d1/evt/vit/fmt/json : msg.payload :
Object
▸{ temperature: 71, humidity: 48 }
08/12/2021, 22:29:58  node: f072331d531af792
iot-2/type/device/id/d1/evt/vit/fmt/json : msg.payload :
Object
▸{ temperature: 41, humidity: 93 }
08/12/2021, 22:31:25  node: f072331d531af792
```

6. After adding the message payload pallet the output should also be visible on the debug console of Node-RED platform.



```
12/2/2021, 6:41:22 PM   node: f2f2649a.0d0d98
msg.payload : string[15]
"Hello Node-RED!"
12/2/2021, 6:44:14 PM
msg : error
"Error: ENOENT: no such file or directory, open '/home/vcap/app/.node-red/package.json'"
12/2/2021, 6:58:17 PM   node: f072331d531af792
iot-2/type/device/id/01/evt/event_1/fmt/json :
msg.payload : Object
▶ { temperature: 65, humidity: 27
```

7. Before executing our code we need to install wiotp.sdk module using the command pip install wiotp-sdk.



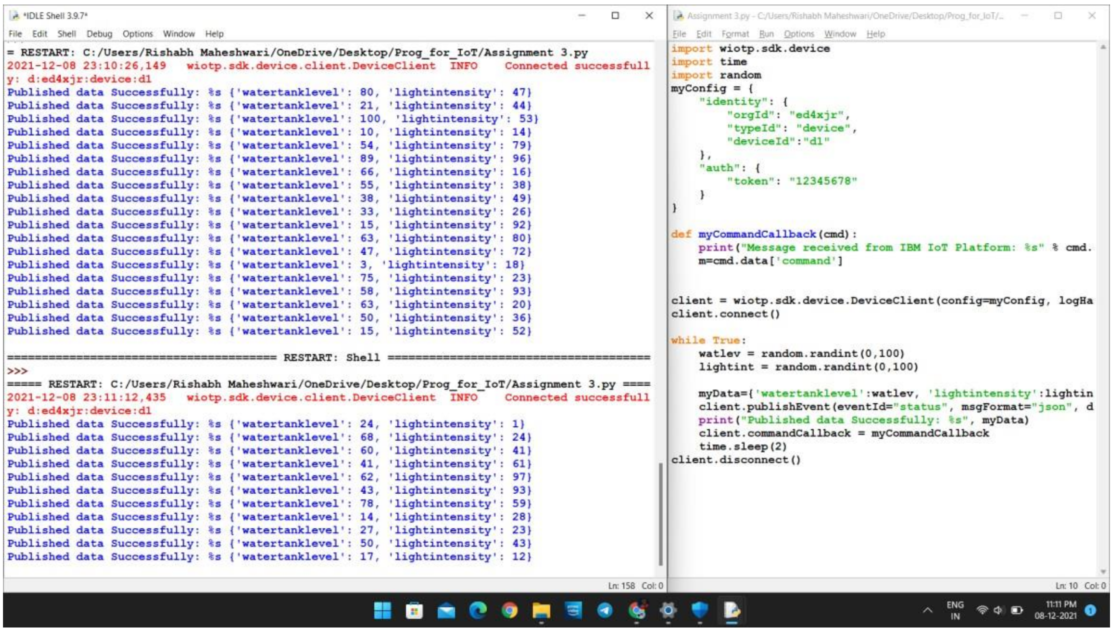
File Edit Format Run Options Window Help
Select C:\WINDOWS\system32\cmd.exe - pip install wiotp-sdk
rishabh.maheshwari2019@vitbhopal.ac.in
ID: ed4xjr
Add Device
Device Simulator
Descriptive Location
a few seconds ago
a few seconds ago
Ln: 20 Col: 16

```
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "ed4xjr",
```

```
    time.sleep(2)
client.disconnect()
```

```
Microsoft Windows [Version 10.0.19043.1387]
(c) Microsoft Corporation. All rights reserved.

C:\Users\USER>pip install wiotp-sdk
```

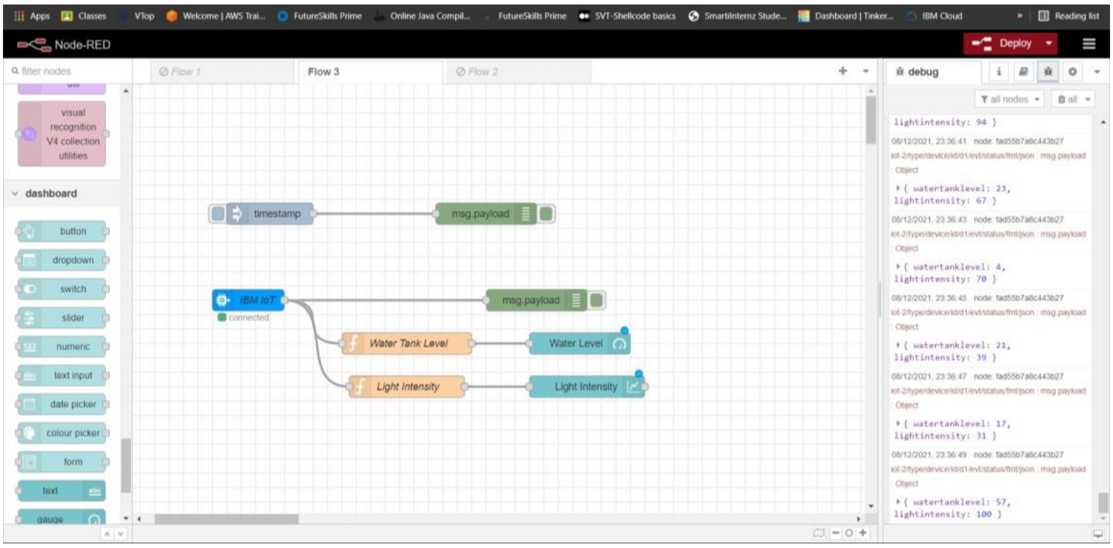
8. Now for sending the data by using Python IDLE, we execute the program code so that the program starts sending data to the IBM Watson Iot Platform. This data id further sent to the Node-RED service amd is visible on the debug console in JSON format.


IDLE Shell 3.9.7
File Edit Shell Debug Options Window Help
Ln: 158 Col: 0
Assignment 3.py - C:/Users/Rishabh Maheshwari/OneDrive/Desktop/Prog_for_IoT/...
File Edit Format Run Options Window Help
Ln: 10 Col: 0
ENG IN
11:11 PM
08-12-2021

```
= RESTART: C:/Users/Rishabh Maheshwari/OneDrive/Desktop/Prog_for_IoT/Assignment 3.py
2021-12-08 23:10:26,149   wiotp.sdk.device.client.DeviceClient  INFO    Connected successfull
y: d:ed4xjr:device:d1
Published data Successfully: %s {'watertanklevel': 80, 'lightintensity': 47}
Published data Successfully: %s {'watertanklevel': 21, 'lightintensity': 44}
Published data Successfully: %s {'watertanklevel': 100, 'lightintensity': 53}
Published data Successfully: %s {'watertanklevel': 10, 'lightintensity': 14}
Published data Successfully: %s {'watertanklevel': 54, 'lightintensity': 79}
Published data Successfully: %s {'watertanklevel': 89, 'lightintensity': 96}
Published data Successfully: %s {'watertanklevel': 66, 'lightintensity': 16}
Published data Successfully: %s {'watertanklevel': 55, 'lightintensity': 38}
Published data Successfully: %s {'watertanklevel': 38, 'lightintensity': 49}
Published data Successfully: %s {'watertanklevel': 33, 'lightintensity': 26}
Published data Successfully: %s {'watertanklevel': 15, 'lightintensity': 92}
Published data Successfully: %s {'watertanklevel': 63, 'lightintensity': 80}
Published data Successfully: %s {'watertanklevel': 47, 'lightintensity': 72}
Published data Successfully: %s {'watertanklevel': 3, 'lightintensity': 18}
Published data Successfully: %s {'watertanklevel': 75, 'lightintensity': 23}
Published data Successfully: %s {'watertanklevel': 58, 'lightintensity': 93}
Published data Successfully: %s {'watertanklevel': 63, 'lightintensity': 20}
Published data Successfully: %s {'watertanklevel': 50, 'lightintensity': 36}
Published data Successfully: %s {'watertanklevel': 15, 'lightintensity': 52}

======================================== RESTART: Shell ========================================
>>>
===== RESTART: C:/Users/Rishabh Maheshwari/OneDrive/Desktop/Prog_for_IoT/Assignment 3.py ====
2021-12-08 23:11:12,435   wiotp.sdk.device.client.DeviceClient  INFO    Connected successfull
y: d:ed4xjr:device:d1
Published data Successfully: %s {'watertanklevel': 24, 'lightintensity': 1}
Published data Successfully: %s {'watertanklevel': 68, 'lightintensity': 24}
Published data Successfully: %s {'watertanklevel': 60, 'lightintensity': 41}
Published data Successfully: %s {'watertanklevel': 41, 'lightintensity': 61}
Published data Successfully: %s {'watertanklevel': 62, 'lightintensity': 97}
Published data Successfully: %s {'watertanklevel': 43, 'lightintensity': 93}
Published data Successfully: %s {'watertanklevel': 78, 'lightintensity': 59}
Published data Successfully: %s {'watertanklevel': 14, 'lightintensity': 28}
Published data Successfully: %s {'watertanklevel': 27, 'lightintensity': 23}
Published data Successfully: %s {'watertanklevel': 50, 'lightintensity': 43}
Published data Successfully: %s {'watertanklevel': 17, 'lightintensity': 12}
```

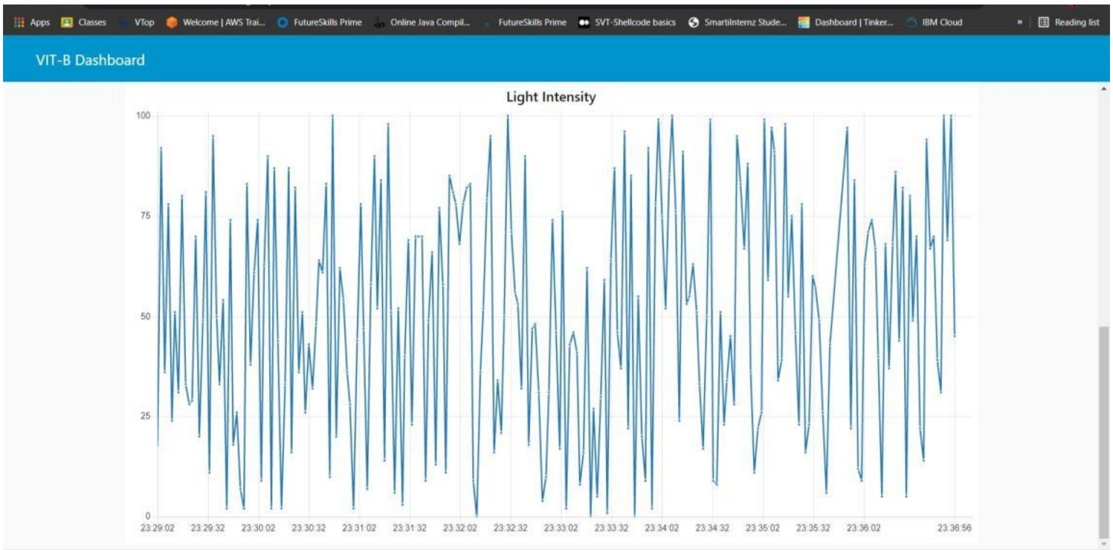
```python
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "ed4xjr",
        "typeId": "device",
        "deviceId":"d1"
    },
    "auth": {
        "token": "12345678"
    }
}

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


client = wiotp.sdk.device.DeviceClient(config=myConfig, logHa
client.connect()

while True:
    watlev = random.randint(0,100)
    lightint = random.randint(0,100)

    myData={'watertanklevel':watlev, 'lightintensity':lightin
    client.publishEvent(eventId="status", msgFormat="json", d
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```

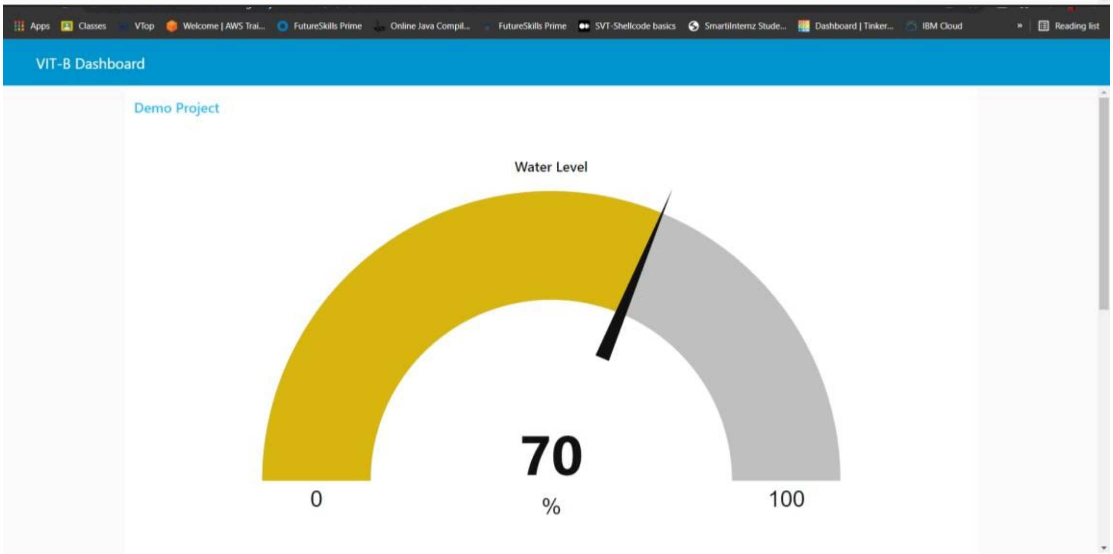
9. Now add the function and dashboard pallets to the flow. Configure them using the correct IDs’. Hence, the data will be available in Readable format.



```
lightintensity: 94 }
08/12/2021, 23:36:41  node: fad55b7a8c443b27
iot-2/type/device/id/d1/evt/status/fmt/json : msg.payload
: Object
▸{ watertanklevel: 23,
lightintensity: 67 }
08/12/2021, 23:36:43  node: fad55b7a8c443b27
iot-2/type/device/id/d1/evt/status/fmt/json : msg.payload
: Object
▸{ watertanklevel: 4,
lightintensity: 70 }
08/12/2021, 23:36:45  node: fad55b7a8c443b27
iot-2/type/device/id/d1/evt/status/fmt/json : msg.payload
: Object
▸{ watertanklevel: 21,
lightintensity: 39 }
08/12/2021, 23:36:47  node: fad55b7a8c443b27
iot-2/type/device/id/d1/evt/status/fmt/json : msg.payload
: Object
▸{ watertanklevel: 17,
lightintensity: 31 }
08/12/2021, 23:36:49  node: fad55b7a8c443b27
iot-2/type/device/id/d1/evt/status/fmt/json : msg.payload
: Object
▸{ watertanklevel: 57,
lightintensity: 100 }
```

10. To visualize the data in application such as chrome, just copy the Node-RED link till .net and paste the url appended by /ui in search bar. The visualization can be depicted as shown below.





Daksh Pathak
19BCG10043