

ASSIGNMENT-3

Task: 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.

Code:

```
import wiotp.sdk.device

import time

import random

myConfig = {

    "identity": {

        "orgId": "y9045l",

        "typeId": "mobile",

        "deviceId": "09876"

    },

    "auth": {

        "token": "12345678"

    }

}

def myCommandCallback(cmd):

    print("Message received from IBM IoT Platform: %s" %

cmd.data['command'])

    m=cmd.data['command']
```

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

```
while True:
```

```
    level=random.randint(0,500)
```

```
    light=random.randint(0,100)
```

```
    myData={'waterlevel':level, 'lightintensity':light}
```

```
    client.publishEvent(eventId="status", msgFormat="json", data=myData,
qos=0, onPublish=None)
```

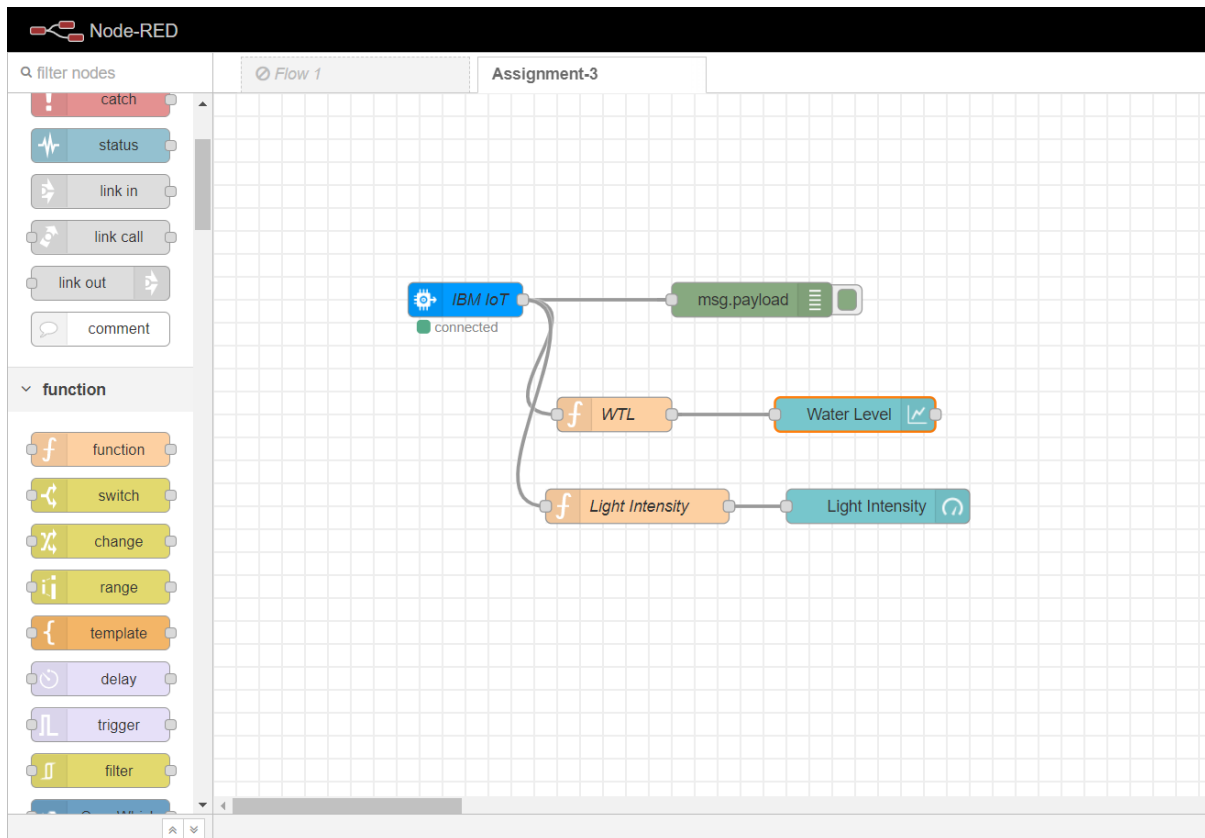
```
    print("Published data Successfully: %s", myData)
```

```
    client.commandCallback = myCommandCallback
```

```
    time.sleep(2)
```

```
client.disconnect()
```

Node Red Graph:



Debug – output:

The image shows the Node-RED debug console output. The output is a list of messages received from the "IBM IoT" node. Each message is a JSON object containing "waterlevel" and "lightintensity" values. The messages are as follows:

- `{ waterlevel: 111, lightintensity: 69 }`
- `{ waterlevel: 313, lightintensity: 77 }`
- `{ waterlevel: 99, lightintensity: 46 }`
- `{ waterlevel: 194, lightintensity: 0 }`
- `{ waterlevel: 415, lightintensity: 85 }`
- `{ waterlevel: 390, lightintensity: 71 }`

Web – Application Graph:

