

## USE DASHBOARD NODES FOR CREATING UI WEB APP

Date	16 NOVEMBER 2022
Team ID	PNT2022TMID32972
Project Name	<b>Project - Real-Time River Water Quality Monitoring and Control System</b>
Maximum Marks	4 Marks

Figure 1: when the light on and light off buttons are pressed

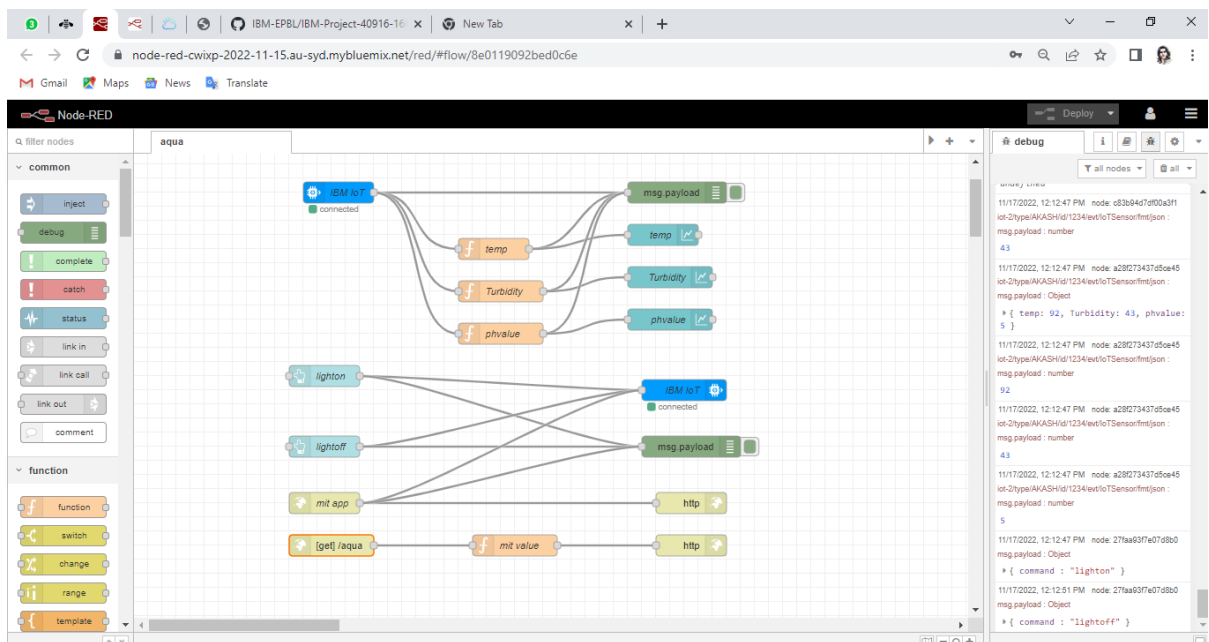
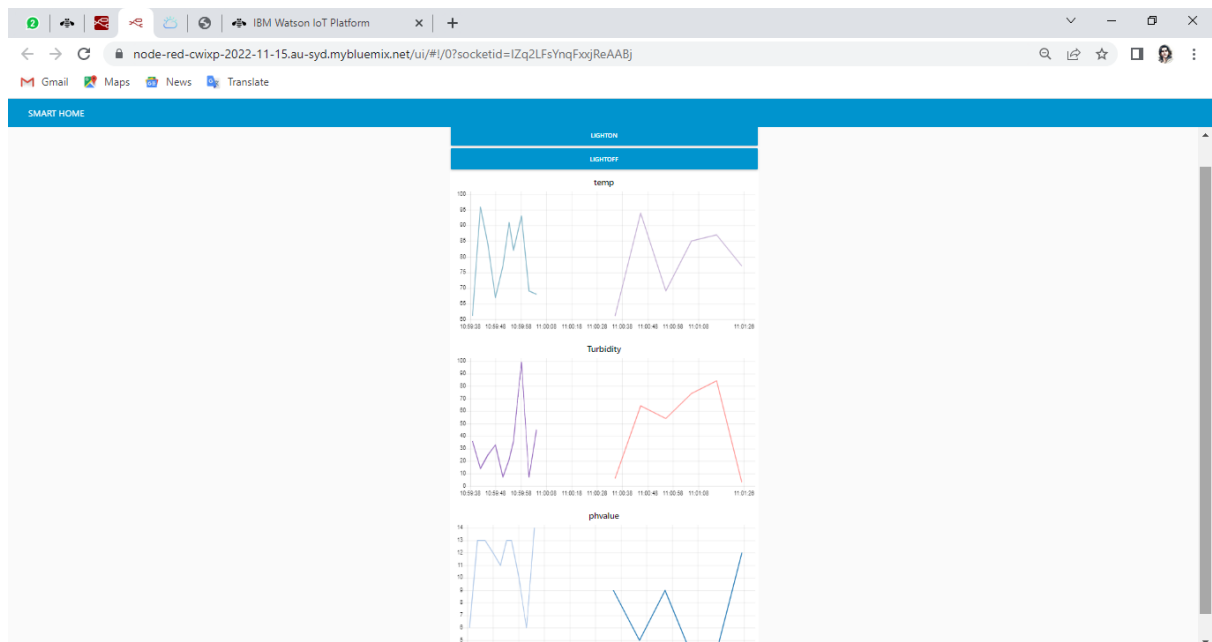
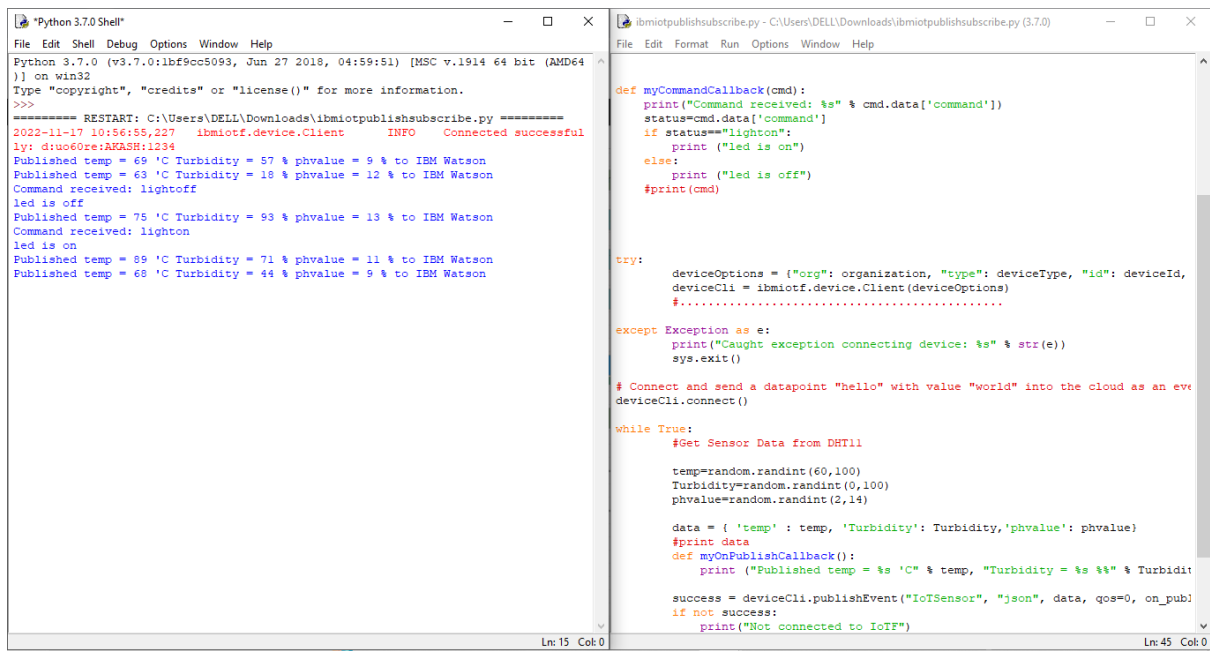


Figure 2: WEB UI Created using NodeRed Dashboard



**Figure 3: Python 3.7.0 software output when the LIGHT ON and LIGHT OFF buttons on the WEB UI is pressed**



The image shows two side-by-side windows of a Python 3.7.0 Shell. The left window displays the output of a script, showing a restart message and several lines of sensor data (temp, turbidity, phvalue) being published to IBM Watson. The right window shows the source code of the script, which includes a command callback function, a try block for device connection, and a while loop for data publishing.

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\DELL\Downloads\ibmiotpublishsubscribe.py =====
2022-11-17 10:56:55,227 ibmiotf.device.Client INFO Connected successful
ly: diuo60re:AKASH:1234
Published temp = 69 'C Turbidity = 57 % phvalue = 9 % to IBM Watson
Published temp = 63 'C Turbidity = 18 % phvalue = 12 % to IBM Watson
Command received: lightoff
led is off
Published temp = 75 'C Turbidity = 93 % phvalue = 13 % to IBM Watson
Command received: lighton
led is on
Published temp = 89 'C Turbidity = 71 % phvalue = 11 % to IBM Watson
Published temp = 68 'C Turbidity = 44 % phvalue = 9 % to IBM Watson
Ln:15 Col:0

ibmiotpublishsubscribe.py - C:\Users\DELL\Downloads\ibmiotpublishsubscribe.py (3.7.0)
File Edit Format Run Options Window Help
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    else:
        print ("led is off")
    #print(cmd)

try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,
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 eve
deviceCli.connect()

while True:
    #Get Sensor Data from DHT11
    temp=random.randint(60,100)
    Turbidity=random.randint(0,100)
    phvalue=random.randint(2,14)

    data = { 'temp' : temp, 'Turbidity': Turbidity, 'phvalue': phvalue}
    #print data
    def myOnPublishCallback():
        print ("Published temp = %s 'C" % temp, "Turbidity = %s %%" % Turbidit
success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publ
if not success:
    print("Not connected to IoT")
Ln:45 Col:0
```