

| | |
|--------------|--|
| TEAM ID | PND2022TMID32705 |
| PROJECT NAME | REAL - TIME RIVER WATER QUALITY MONITORING AND CONTROL SYSTEM |

PROJECT DEVELOPMENT PHASE SPRINT 3

Develop a python script:

```
import ibmiotf.application
import ibmiotf.device
import time
import random
import sys
from twilio.rest import Client
import keys
Client = Client(keys.account_sid, keys.auth_token)
```

```
organization = "Bluemix Free"
deviceType = "Microcontroller_device"
deviceId = "deivanai"
authMethod = "token"
authToken = "deiva@1234"
```

```
pH = random.randint(1, 14)
turbidity = random.randint(1, 1000)
temperature = random.randint(0, 100)
humidity = random.randint(30, 60)
```

```
def myCommandCallback(cmd):
    print("Command Received: %s" % cmd.data['command'])
    print(cmd)
```

```
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()

deviceCli.connect()

while True:

    pH = random.randint(1, 14)
    turbidity = random.randint(1, 1000)
    temperature = random.randint(0, 100)
    humidity = random.randint(30, 60)

    data = {'pH': pH, 'turbid': turbidity, 'temp': temperature, 'humi' : 'humidity'}
    def SMS():
        message = Client.messages.create(
            body="ALERT!! THE WATER QUALITY IS DEGRADED",
            from_=keys.twilio_number,
            to = keys.target_number)
        print(message.body)

    if temperature>70 or pH<6 or turbidity>500 or humidity>40:
        SMS()

    def myOnPublishCallback():
        print("Published pH= %s" % pH, "Turbidity:%s" % turbidity, "Temperature:%s" %
temperature)

    success = deviceCli.publishEvent("demo", "json", data, qos=0,
on_publish=myOnPublishCallback)
    if not success:
        print("Not Connected to ibmiot")
        time.sleep(5)
        deviceCli.commandCallback = myCommandCallback

deviceCli.disconnect()

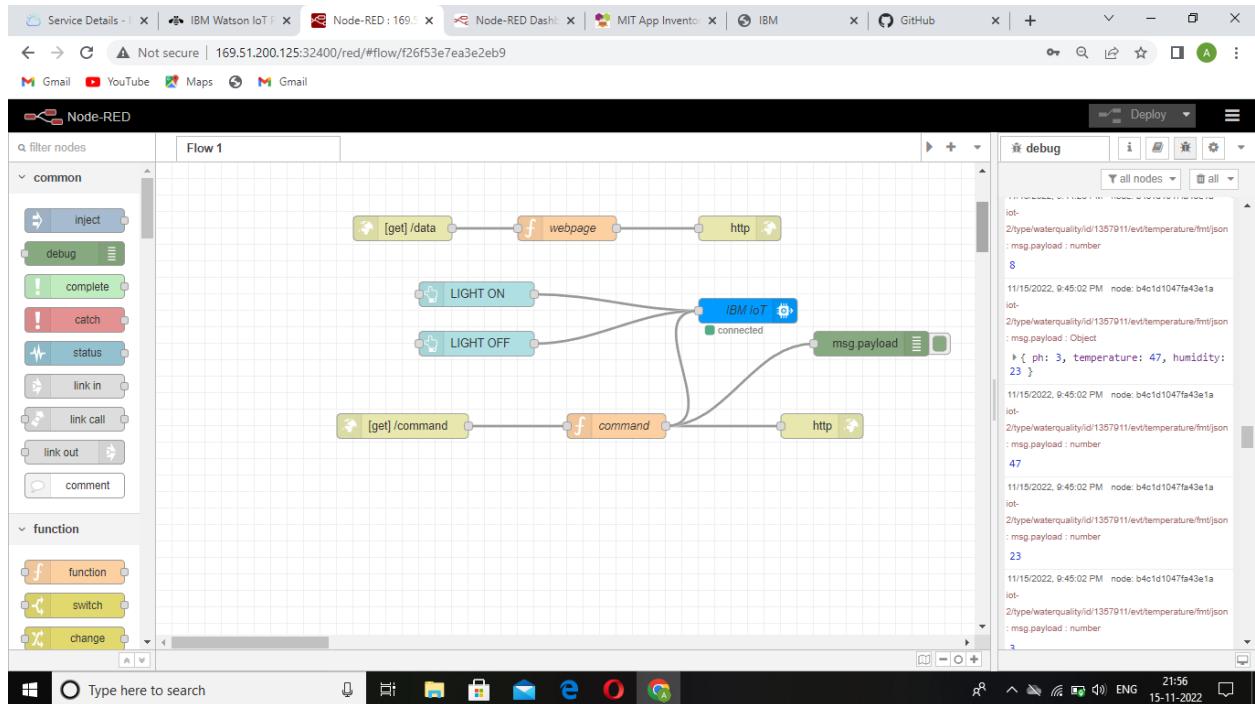
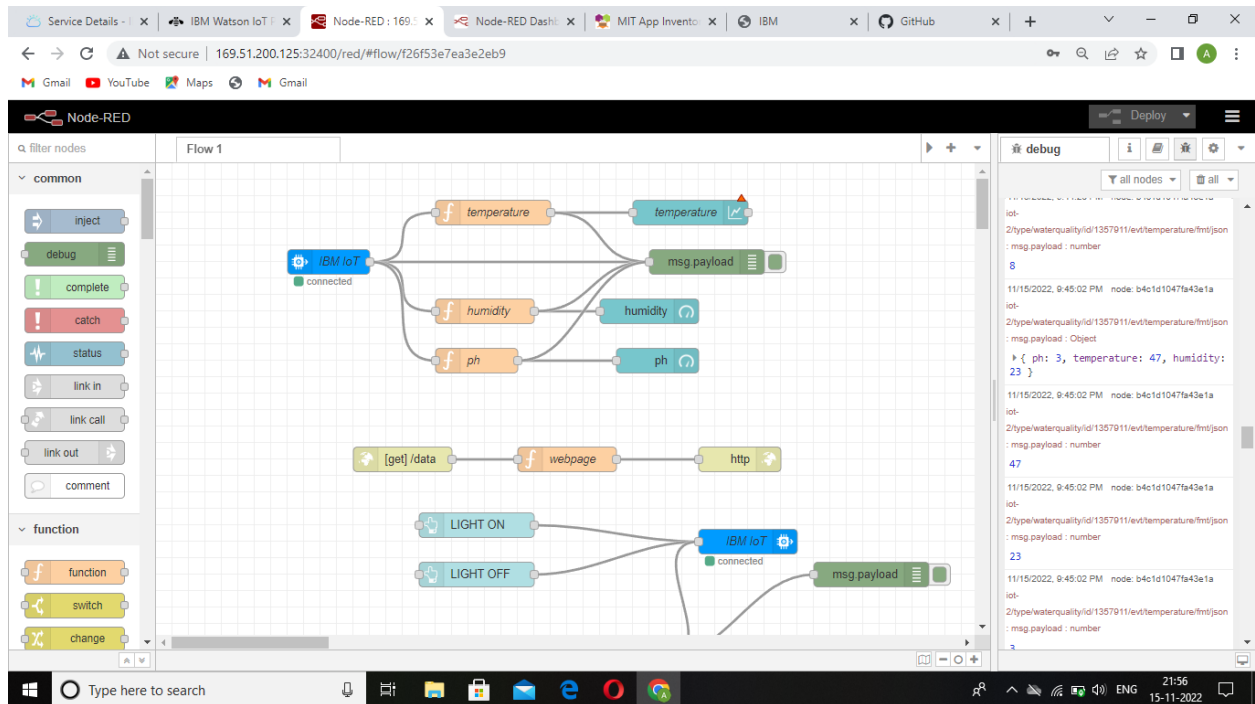
```

2.Executing the developed python script to send value to IoT Watson platform by the MQTT protocol:

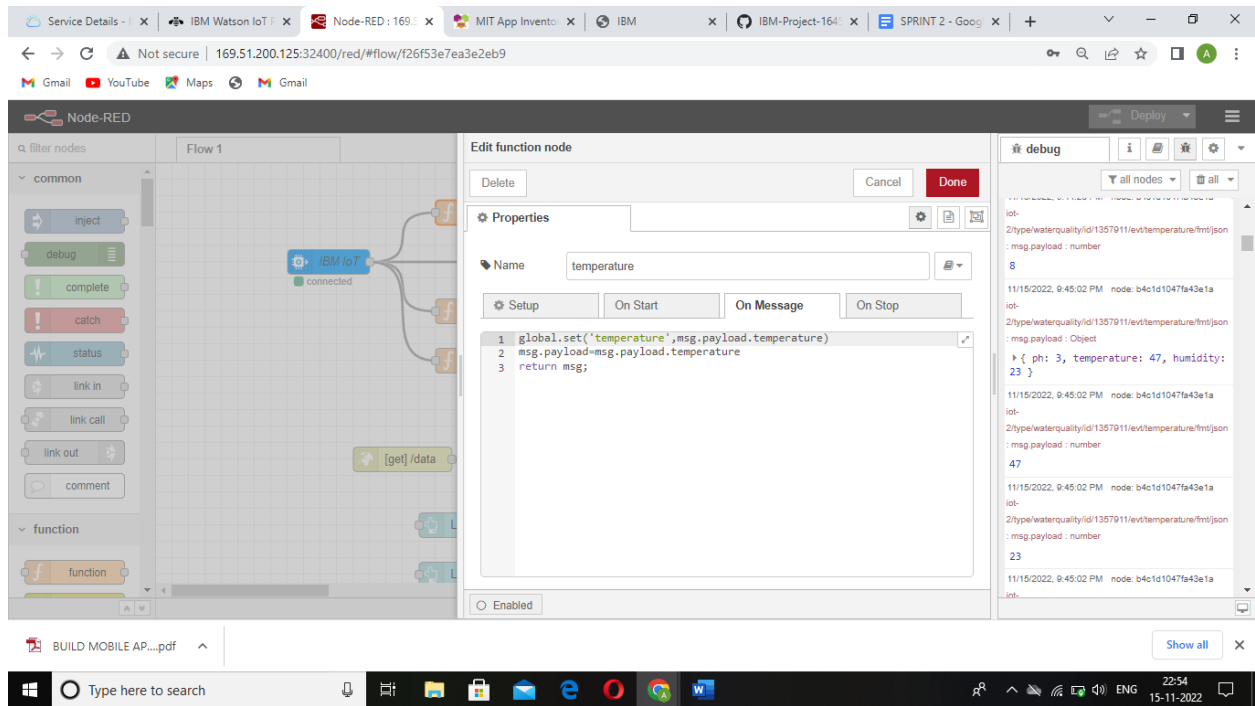
```
C:\Windows\System32\cmd.exe - python randscript.py
Microsoft Windows [Version 10.0.22000.1219]
(c) Microsoft Corporation. All rights reserved.

D:\IOT_PROJECT>python randscript.py
Temp: 87.80 F / 31.0 C    pH: 11.0  Turbidity:0.83NTU
Published
Temp: 93.20 F / 34.0 C    pH: 7.0   Turbidity:0.60NTU
Published
Temp: 86.00 F / 30.0 C    pH: 10.0  Turbidity:1.59NTU
Published
Temp: 104.00 F / 40.0 C   pH: 0.0   Turbidity:0.41NTU
Published
```

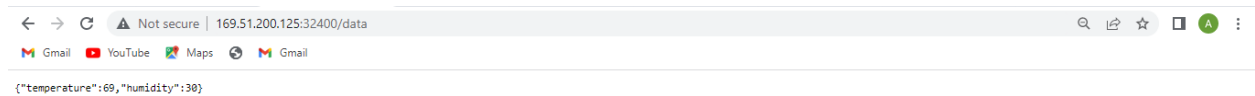
3. Sending the obtained values to Web UI dashboard and designed app :



4. Payload defined to obtain all the parameters in mobile app:



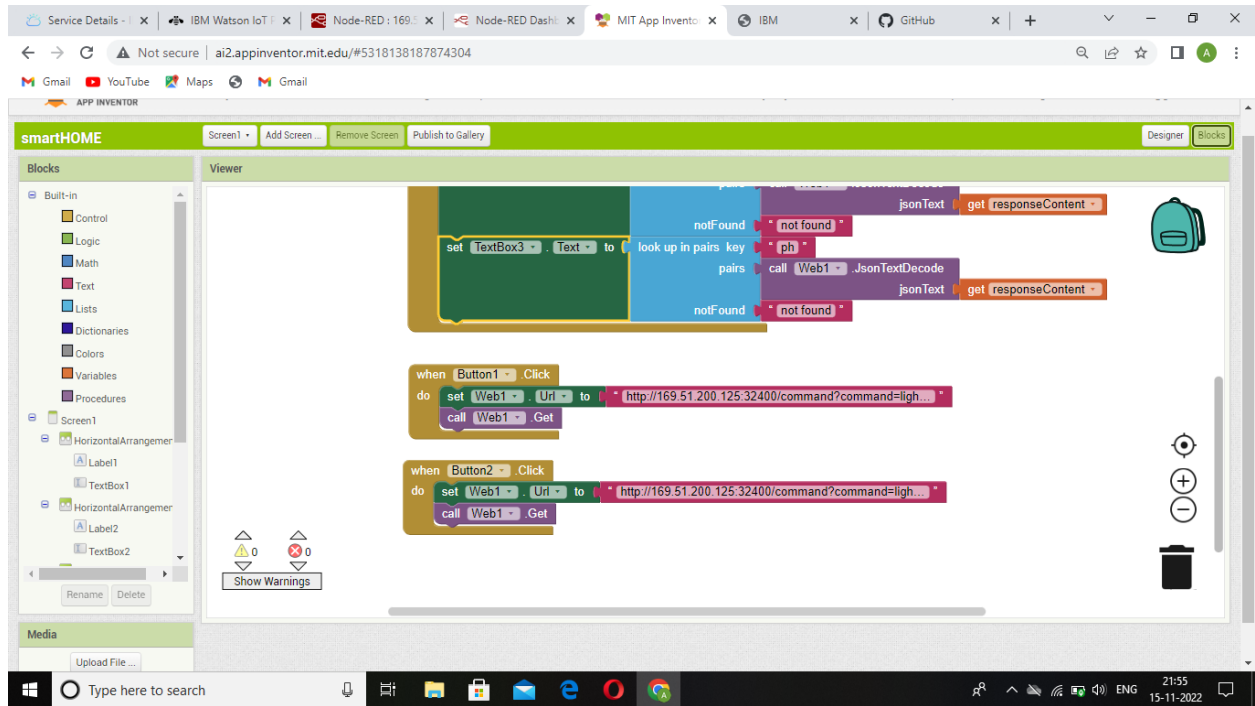
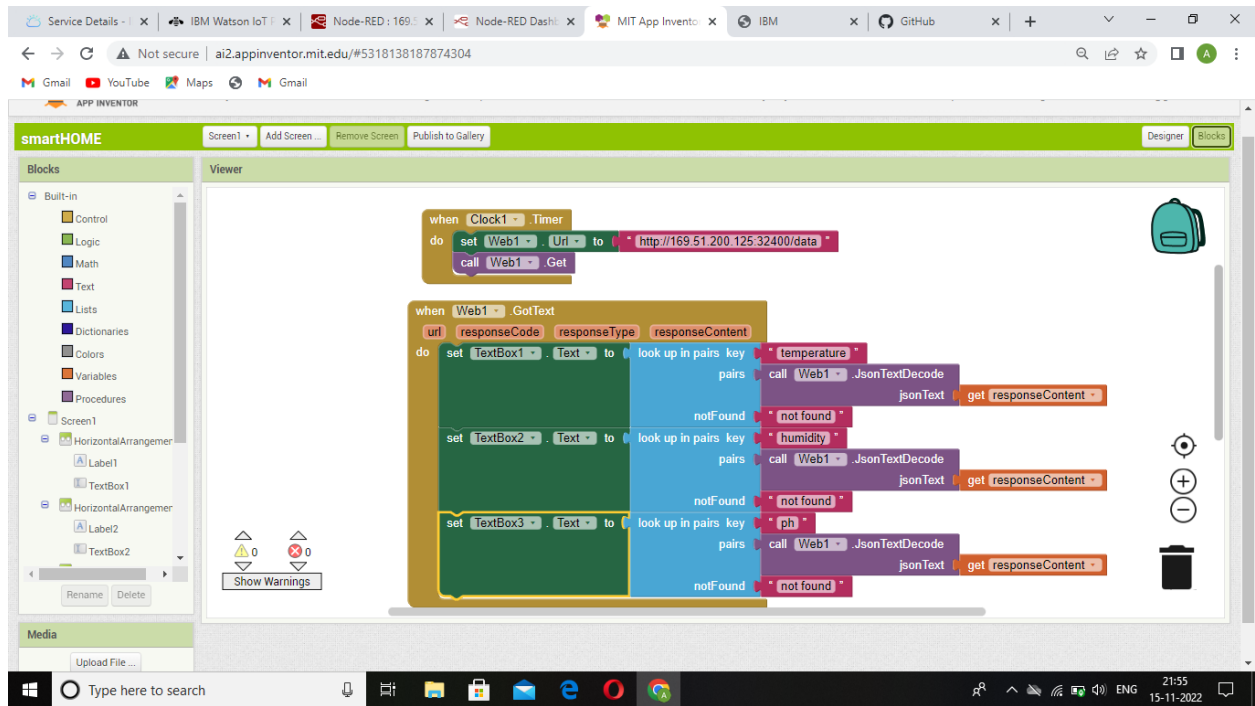
5.JSON object obtained in the specified URL:



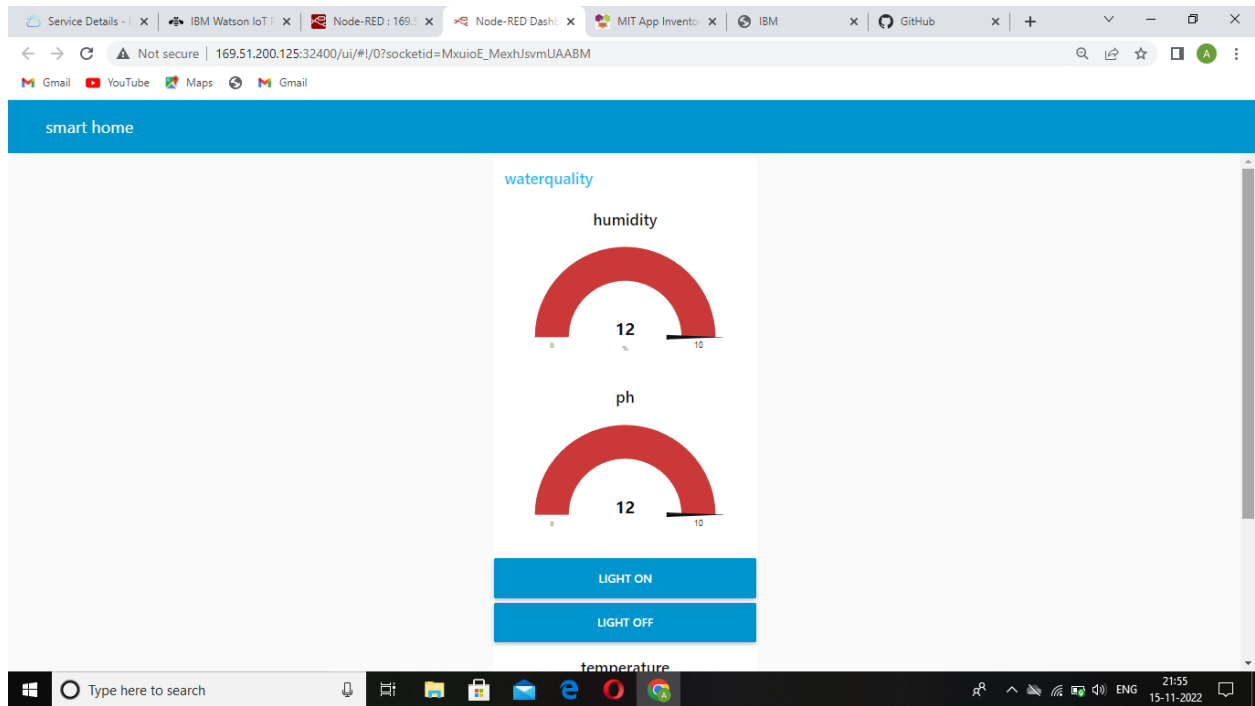
6.Mobile UI frontend to receive the data from Node-Red:



7. Configuring MIT Mobile app backend to receive the data from Node-Red:



8. Web UI dashboard:



9. Checking in mobile app whether data correctly received or not :

