Project Development Phase
Sprint-2

Date	5 November 2022
Team ID	PNT2022TMID35841
Project Name	Gas Leakage Monitoring and Alerting System

Sprint Target:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Action	USN-4	As a user, I can get the notification if there is a gas leak	15	High	SUDARSHAN A R, B ADITYA, PREETHI B
		USN-5	As a user, I get the instant message	5	Low	SUDARSHAN A R B ADITYA, PREETHI B

Introduction:

In this Sprint-2, we have added data publishing to the IoT device and Node-Red. We have also added notification in the app if there is a gas leakage.

Publishing to IBM cloud

Using a Python script Gas concentration values in ppm are generated and published to IBM cloud

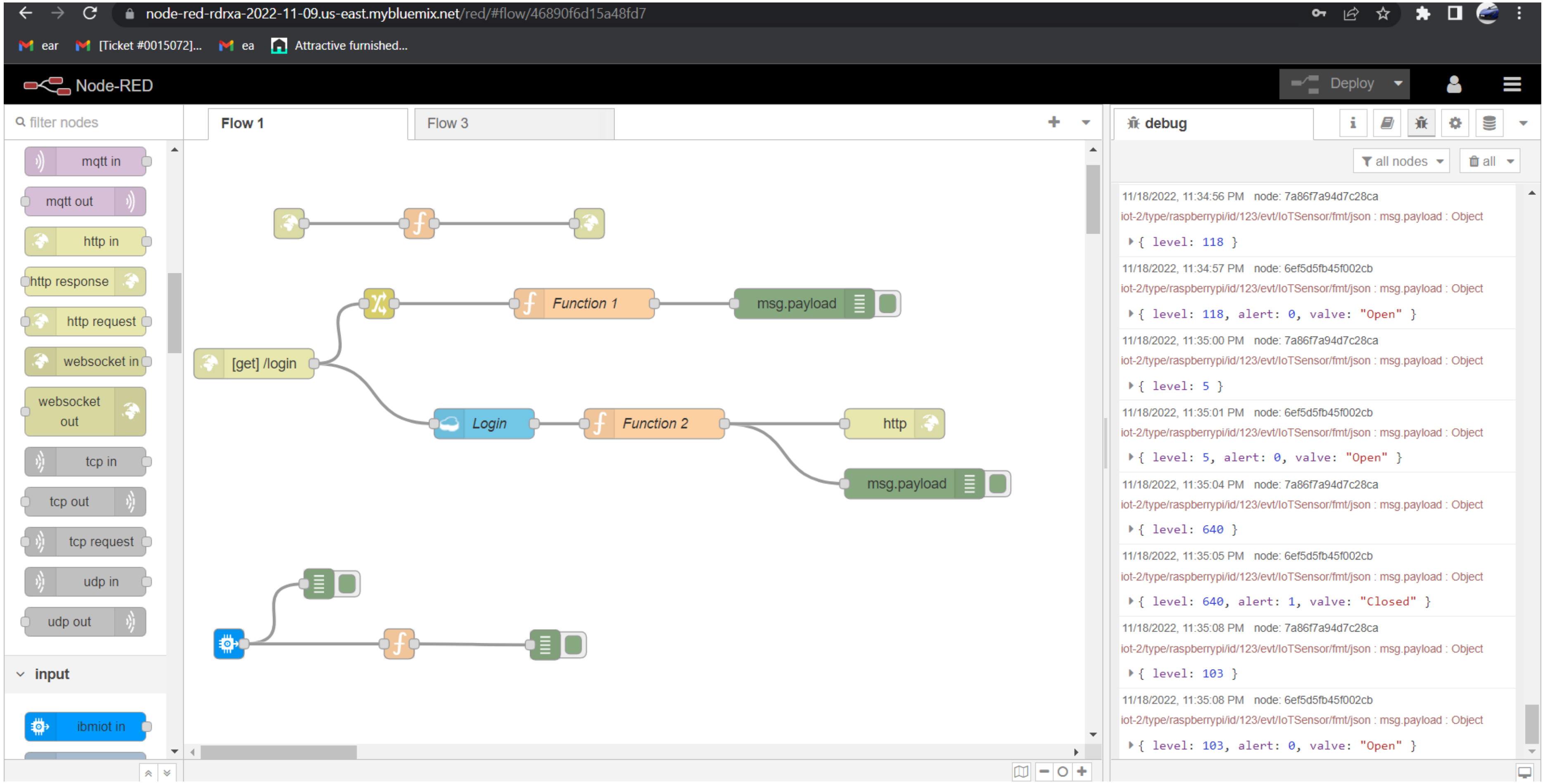
Code:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random


#Provide your IBM Watson Device Credentials
organization = "lqca59"
deviceType = "raspberrypi"
deviceId = "123"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO


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")
```

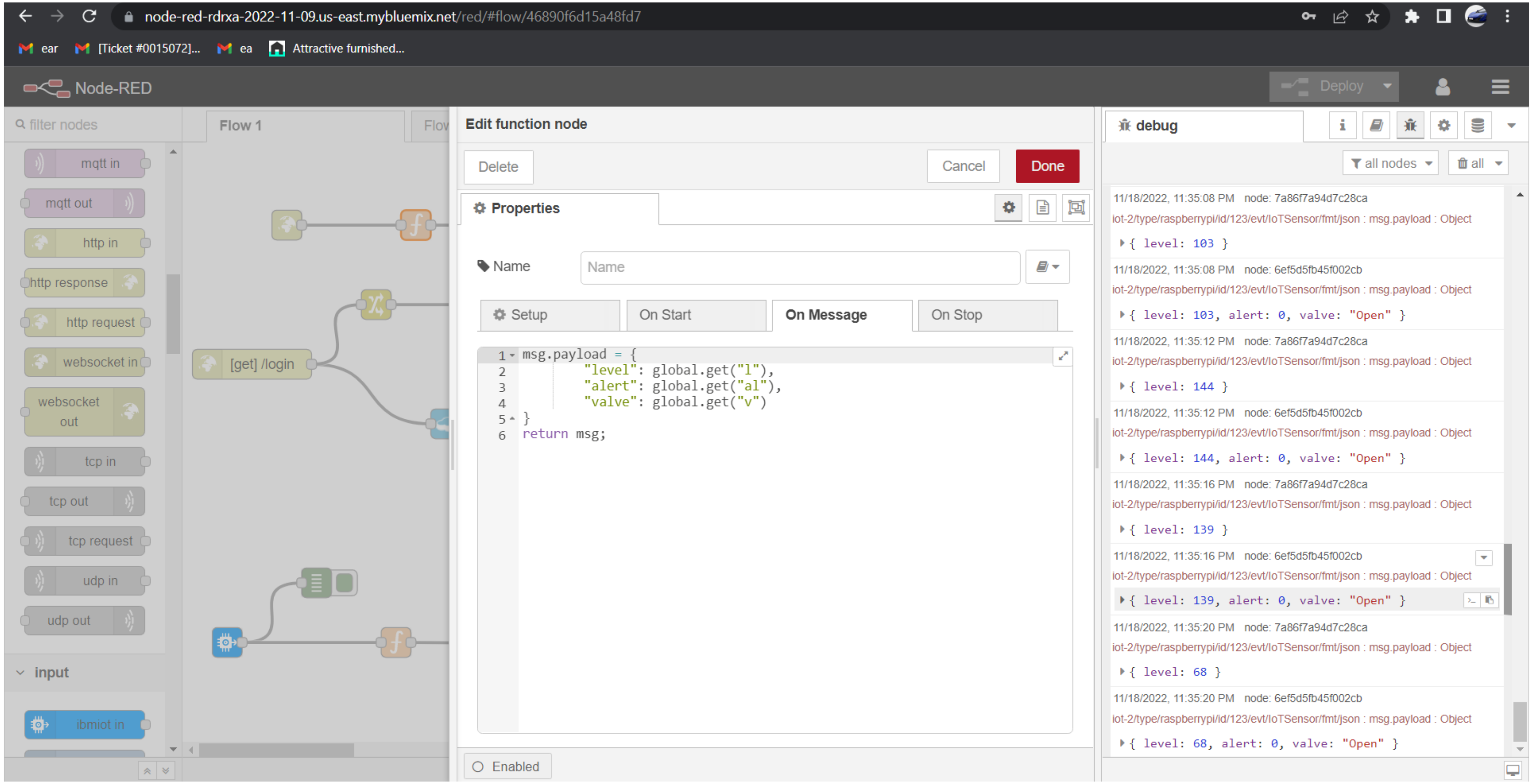
```
try:
        deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-token": authToken}
        deviceCli = ibmiotf.device.Client(deviceOptions)
        #..............................................

except Exception as e:
        print("Caught exception connecting device: %s" % str(e))
        sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times
deviceCli.connect()

while True:
    #Get Sensor Data from DHT11
    x = random.random()
    if x < 0.8:
      gas_level = random.randint(0, 150)
    else:
      gas_level = random.randint(500, 1000)
    # gas_level=random.randint(0,1000)

    data = {'level': gas_level}
    #print data
    def myOnPublishCallback():
      print ("Published Gas Level in Air = %s ppm" % gas_level, "to IBM Watson")

    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
    if not success:
      print("Not connected to IoTF")
    time.sleep(4)

    deviceCli.commandCallback = myCommandCallback
```

```
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

Node-red:

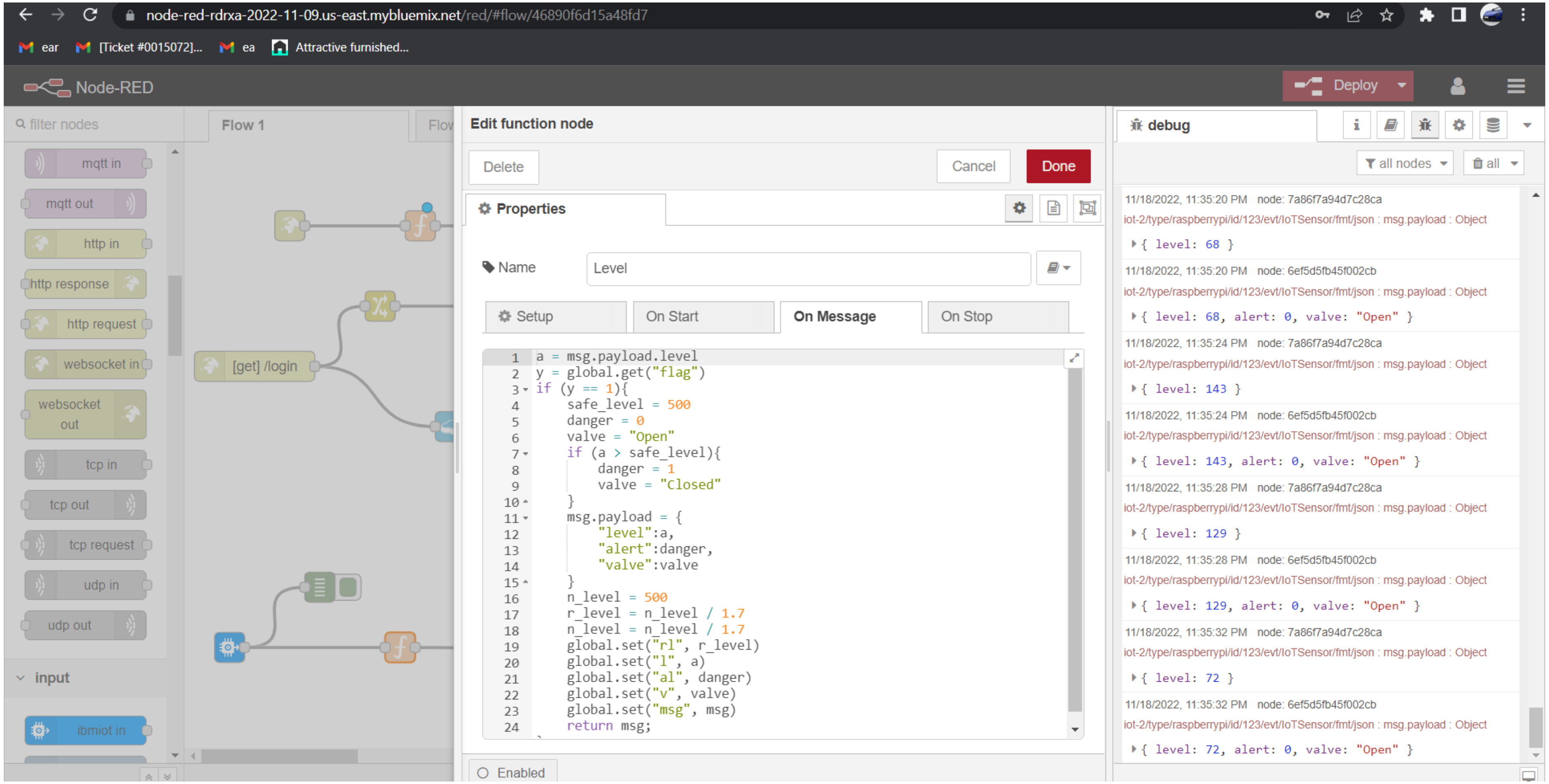


```
11/18/2022, 11:34:56 PM   node: 7a86f7a94d7c28ca
iot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
 ▸{ level: 118 }
11/18/2022, 11:34:57 PM   node: 6ef5d5fb45f002cb
iot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
 ▸{ level: 118, alert: 0, valve: "Open" }
11/18/2022, 11:35:00 PM   node: 7a86f7a94d7c28ca
iot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
 ▸{ level: 5 }
11/18/2022, 11:35:01 PM   node: 6ef5d5fb45f002cb
iot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
 ▸{ level: 5, alert: 0, valve: "Open" }
11/18/2022, 11:35:04 PM   node: 7a86f7a94d7c28ca
iot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
 ▸{ level: 640 }
11/18/2022, 11:35:05 PM   node: 6ef5d5fb45f002cb
iot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
 ▸{ level: 640, alert: 1, valve: "Closed" }
11/18/2022, 11:35:08 PM   node: 7a86f7a94d7c28ca
iot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
 ▸{ level: 103 }
11/18/2022, 11:35:08 PM   node: 6ef5d5fb45f002cb
iot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
 ▸{ level: 103, alert: 0, valve: "Open" }
```



```
msg.payload = {
          "level": global.get("l"),
          "alert": global.get("al"),
          "valve": global.get("v")
}
return msg;
```

```
11/18/2022, 11:35:08 PM   node: 7a86f7a94d7c28ca
iot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
▸{ level: 103 }
11/18/2022, 11:35:08 PM   node: 6ef5d5fb45f002cb
iot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
▸{ level: 103, alert: 0, valve: "Open" }
11/18/2022, 11:35:12 PM   node: 7a86f7a94d7c28ca
iot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
▸{ level: 144 }
11/18/2022, 11:35:12 PM   node: 6ef5d5fb45f002cb
iot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
▸{ level: 144, alert: 0, valve: "Open" }
11/18/2022, 11:35:16 PM   node: 7a86f7a94d7c28ca
iot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
▸{ level: 139 }
11/18/2022, 11:35:16 PM   node: 6ef5d5fb45f002cb
iot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
▸{ level: 139, alert: 0, valve: "Open" }
11/18/2022, 11:35:20 PM   node: 7a86f7a94d7c28ca
iot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
▸{ level: 68 }
11/18/2022, 11:35:20 PM   node: 6ef5d5fb45f002cb
iot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
▸{ level: 68, alert: 0, valve: "Open" }
```



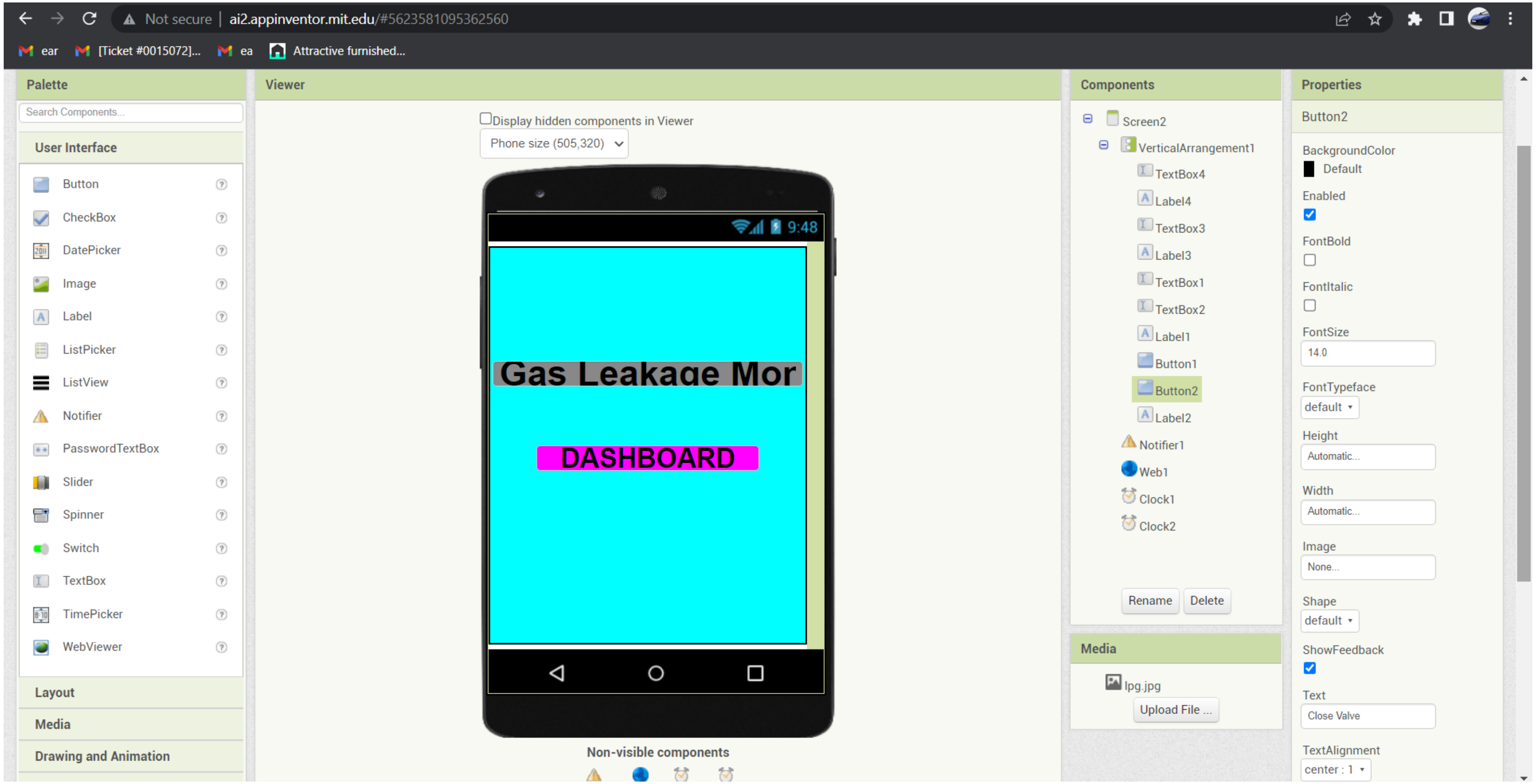
Name: Level

```
a = msg.payload.level
y = global.get("flag")
if (y == 1){
    safe_level = 500
    danger = 0
    valve = "Open"
    if (a > safe_level){
        danger = 1
        valve = "Closed"
    }
    msg.payload = {
        "level":a,
        "alert":danger,
        "valve":valve
    }
    n_level = 500
    r_level = n_level / 1.7
    n_level = n_level / 1.7
    global.set("rl", r_level)
    global.set("l", a)
    global.set("al", danger)
    global.set("v", valve)
    global.set("msg", msg)
    return msg;
```

```
11/18/2022, 11:35:20 PM   node: 7a86f7a94d7c28ca
iot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
▸{ level: 68 }
11/18/2022, 11:35:20 PM   node: 6ef5d5fb45f002cb
iot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
▸{ level: 68, alert: 0, valve: "Open" }
11/18/2022, 11:35:24 PM   node: 7a86f7a94d7c28ca
iot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
▸{ level: 143 }
11/18/2022, 11:35:24 PM   node: 6ef5d5fb45f002cb
iot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
▸{ level: 143, alert: 0, valve: "Open" }
11/18/2022, 11:35:28 PM   node: 7a86f7a94d7c28ca
iot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
▸{ level: 129 }
11/18/2022, 11:35:28 PM   node: 6ef5d5fb45f002cb
iot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
▸{ level: 129, alert: 0, valve: "Open" }
11/18/2022, 11:35:32 PM   node: 7a86f7a94d7c28ca
iot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
▸{ level: 72 }
11/18/2022, 11:35:32 PM   node: 6ef5d5fb45f002cb
iot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
▸{ level: 72, alert: 0, valve: "Open" }
```

Notification in App:

If the Gas concentration increases 500ppm, an alert is issued.

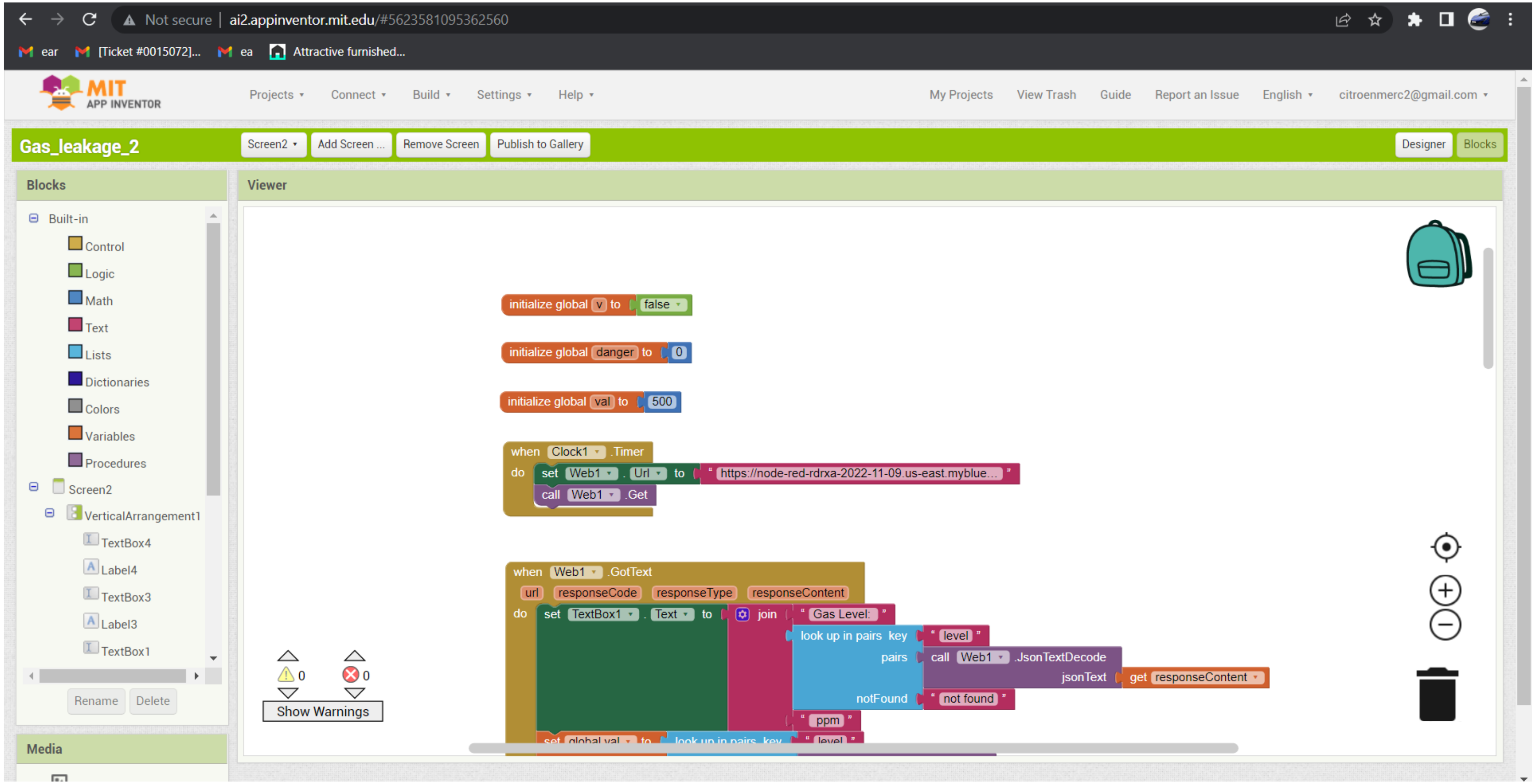


Screen2
- VerticalArrangement1
  - TextBox4
  - Label4
  - TextBox3
  - Label3
  - TextBox1
  - TextBox2
  - Label1
  - Button1
  - Button2
  - Label2
- Notifier1
- Web1
- Clock1
- Clock2

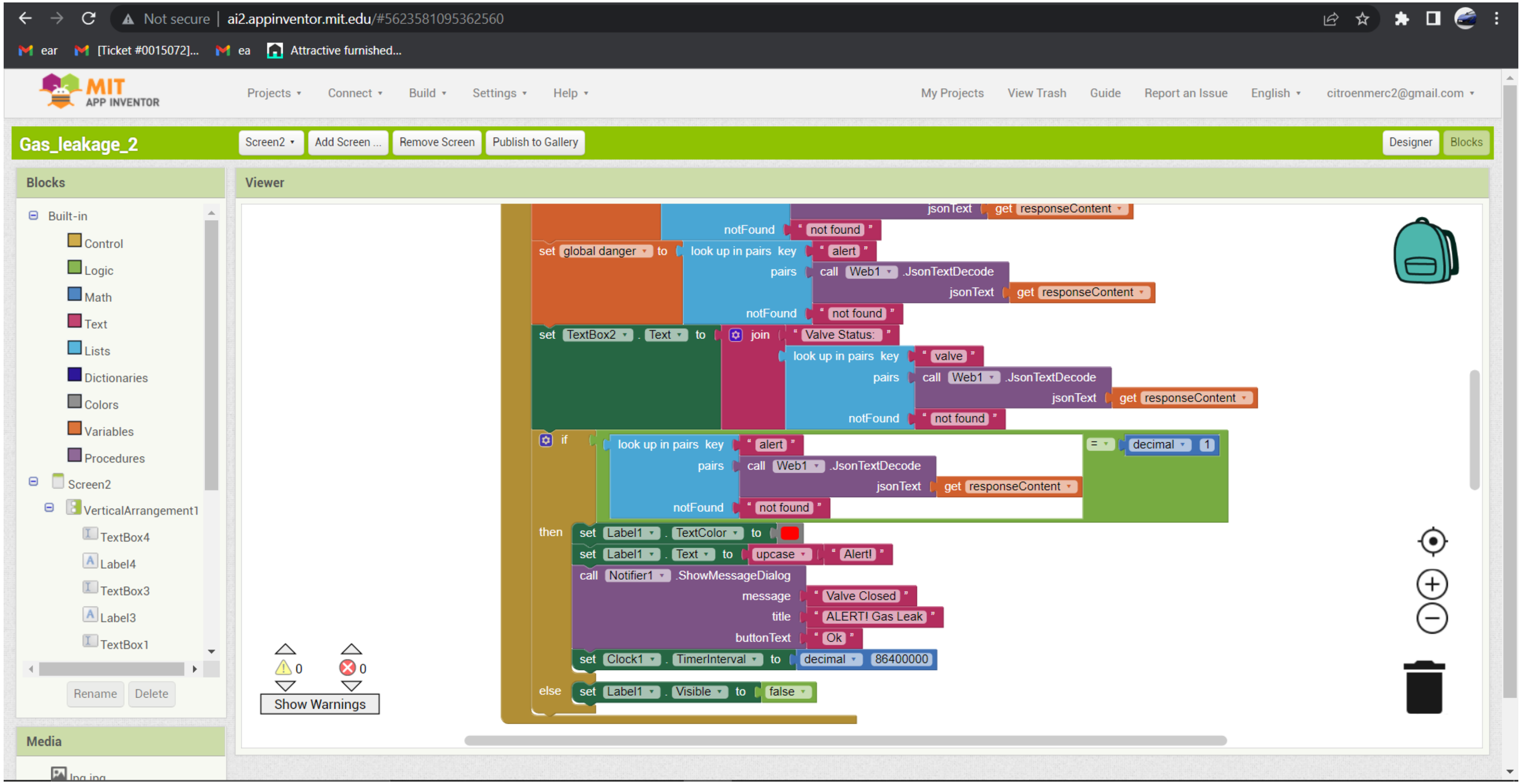
lpg.jpg

Button2

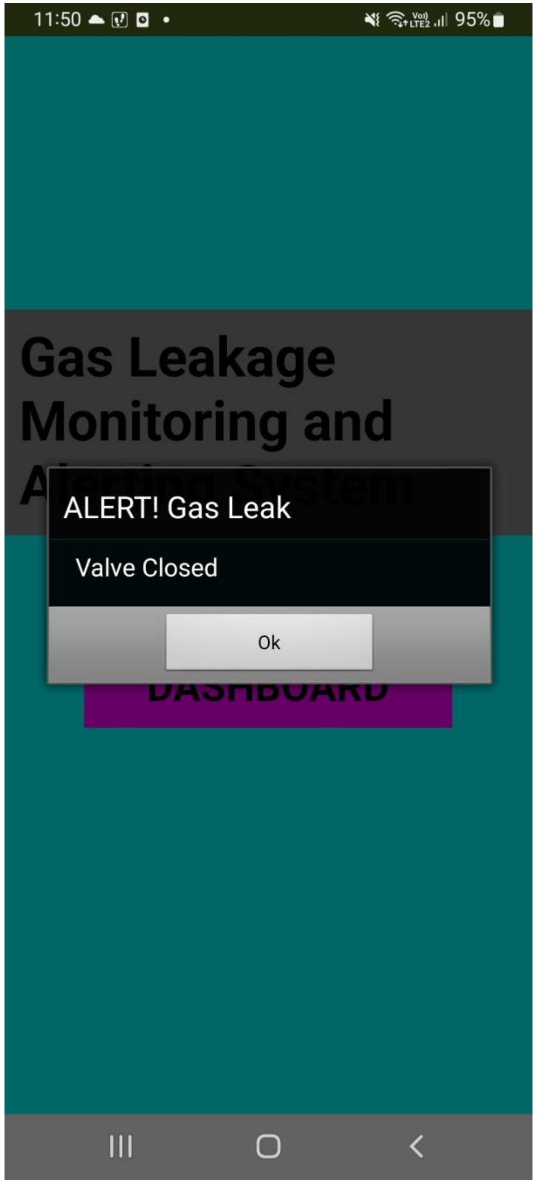
BackgroundColor: Default
Enabled: ☑
FontBold: □
FontItalic: □
FontSize: 14.0
FontTypeface: default
Height: Automatic...
Width: Automatic...
Image: None...
Shape: default
ShowFeedback: ☑
Text: Close Valve
TextAlignment: center : 1



```
initialize global v to false

initialize global danger to 0

initialize global val to 500

when Clock1 .Timer
do  set Web1 . Url to " https://node-red-rdrxa-2022-11-09.us-east.myblue... "
    call Web1 .Get

when Web1 .GotText
 url  responseCode  responseType  responseContent
do  set TextBox1 . Text to  join  " Gas Level: "
                                  look up in pairs  key  " level "
                                                  pairs  call Web1 .JsonTextDecode
                                                                       jsonText  get responseContent
                                               notFound  " not found "
                                  " ppm "
    set global val to  look up in pairs  key  " level "
```



```
                                        jsonText  get responseContent
                      notFound  " not found "
set global danger to  look up in pairs key  " alert "
                      pairs  call Web1 .JsonTextDecode
                                       jsonText  get responseContent
                      notFound  " not found "
set TextBox2 . Text to  join  " Valve Status: "
                              look up in pairs key  " valve "
                              pairs  call Web1 .JsonTextDecode
                                               jsonText  get responseContent
                              notFound  " not found "
if    look up in pairs key  " alert "                              =  decimal  1
      pairs  call Web1 .JsonTextDecode
                       jsonText  get responseContent
      notFound  " not found "
then  set Label1 . TextColor to  🟥
      set Label1 . Text to  upcase  " Alert! "
      call Notifier1 .ShowMessageDialog
                      message  " Valve Closed "
                        title  " ALERT! Gas Leak "
                   buttonText  " Ok "
      set Clock1 . TimerInterval to  decimal  86400000
else  set Label1 . Visible to  false
```



ALERT! Gas Leak

Valve Closed

In the upcoming sprints, we will add the dashboard and UI elements.