

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 = "Iqca59"
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 IoT")
        time.sleep(4)

    deviceCli.commandCallback = myCommandCallback

```

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

Node-red:

The screenshot displays the Node-RED web interface in a browser. The address bar shows the URL: `node-red-rdxa-2022-11-09.us-east.mybluemix.net/red/#flow/46890f6d15a48fd7`. The interface includes a left sidebar with a "filter nodes" search bar and a list of nodes categorized under "input", including `ibmiot in`. The main workspace shows "Flow 1" with a diagram consisting of several nodes: a `[get] /login` node, a `Login` node, two `Function` nodes (labeled "Function 1" and "Function 2"), and `msg.payload` output nodes. The right sidebar features a "debug" tab showing a log of messages. The messages are JSON objects with fields like `level`, `alert`, and `valve`, along with timestamps and node IDs.

debug

all nodes

11/18/2022, 11:34:56 PM node: 7a86f7a94d7c28ca
lot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
▶ { level: 118 }

11/18/2022, 11:34:57 PM node: 6ef5d5fb45f002cb
lot-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
lot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
▶ { level: 5 }

11/18/2022, 11:35:01 PM node: 6ef5d5fb45f002cb
lot-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
lot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
▶ { level: 640 }

11/18/2022, 11:35:05 PM node: 6ef5d5fb45f002cb
lot-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
lot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
▶ { level: 103 }

11/18/2022, 11:35:08 PM node: 6ef5d5fb45f002cb
lot-2/type/raspberrypi/id/123/evt/IoTSensor/fmt/json : msg.payload : Object
▶ { level: 103, alert: 0, valve: "Open" }

The screenshot displays the Node-RED web interface in a browser. The top navigation bar includes a back arrow, a search icon, and the URL 'node-red-rdxa-2022-11-09-us-east.mybluemix.net/red/#flow/46890f6d15a48fd7'. The interface is divided into three main sections:

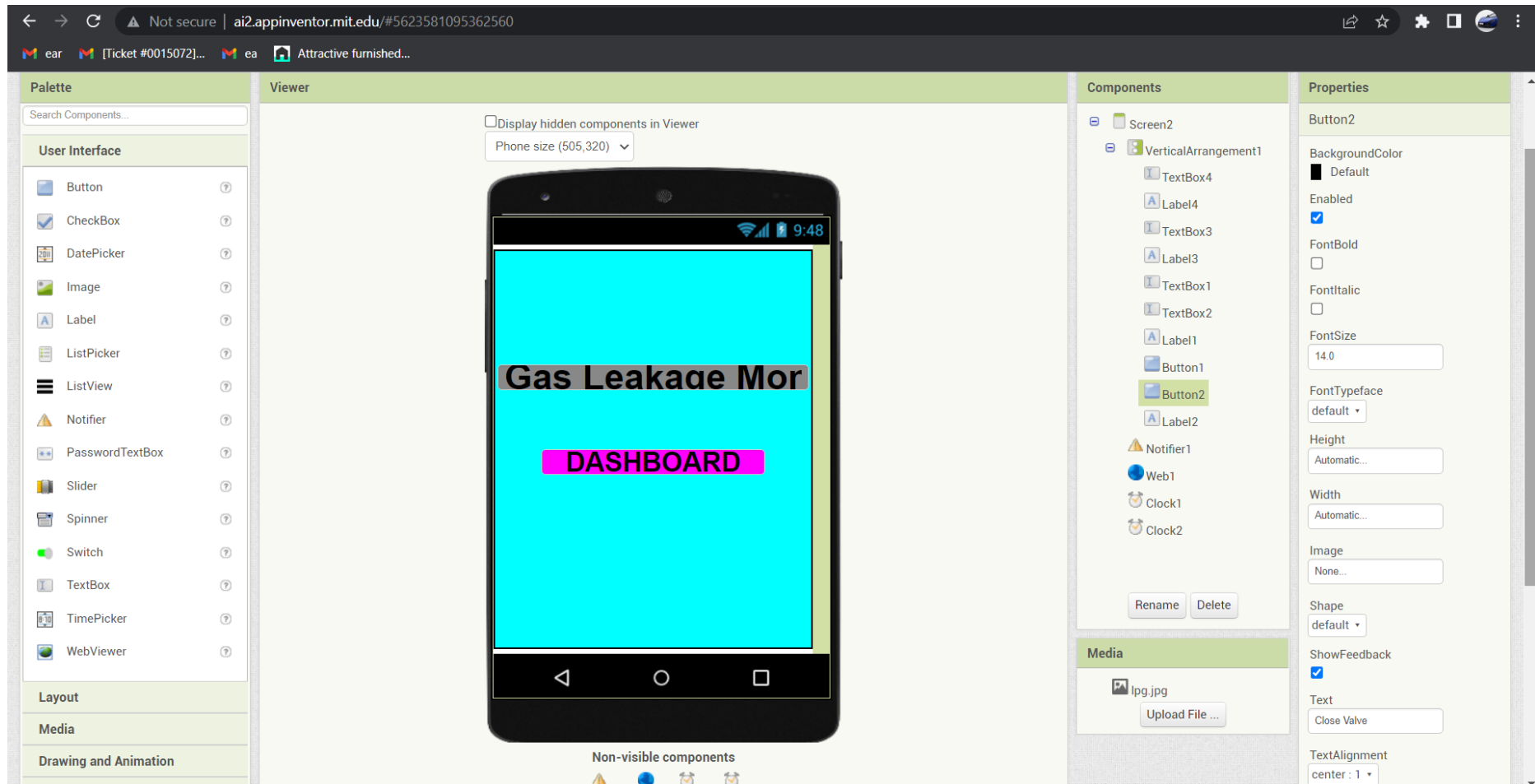
- Left Panel (Library):** A sidebar with a search bar 'filter nodes' and a list of input/output nodes categorized under 'input'. Visible nodes include 'mqtt in', 'mqtt out', 'http in', 'http response', 'http request', 'websocket in', 'websocket out', 'tcp in', 'tcp out', 'tcp request', 'udp in', 'udp out', and 'ibmiot in'.
- Center Panel (Flow Editor):** Shows 'Flow 1' with a visual programming flow. It includes a '[get] /login' node, a merge node (yellow circle with a cross), and a function node (orange circle with 'f').
- Right Panel (Edit function node):** A modal window for editing the selected function node. It has tabs for 'Properties', 'Setup', 'On Start', 'On Message', and 'On Stop'. The 'On Message' tab is active, showing a JavaScript code editor with the following code:

```
1 msg.payload = {  
2   "level": global.get("l"),  
3   "alert": global.get("al"),  
4   "valve": global.get("v")  
5 }  
6 return msg;
```
- Far Right Panel (Debug Console):** A 'debug' panel showing a log of messages. Each entry includes a timestamp, a node ID, and the message payload. The messages are JSON objects with 'level', 'alert', and 'valve' properties. For example, one message is:

```
{ level: 103 }
```


Notification in App:

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




← → ↺

Not secure | ai2.appinventor.mit.edu/#5623581095362560

🔗 ⭐ ⚙️ 📱 🌐

ear [Ticket #0015072]... ea Attractive furnished...

 Projects ▾ Connect ▾ Build ▾ Settings ▾ Help ▾

My Projects View Trash Guide Report an Issue English ▾ citroenmerc2@gmail.com ▾

Gas_leakage_2

Screen2 ▾ Add Screen ... Remove Screen Publish to Gallery

Designer Blocks

Blocks

Built-in

Control

Logic

Math

Text

Lists

Dictionaries

Colors

Variables

Procedures

Screen2

VerticalArrangement1

TextBox4

Label4

TextBox3

Label3

TextBox1

0 0

Show Warnings

Media

Viewer

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.GetText 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 look up in pairs key level

📁

🔍

+

-

🗑️

← → ↺

Not secure | ai2.appinventor.mit.edu/#5623581095362560

✎ ⭐ ⚙ □ 🌐

ear [Ticket #0015072]... ea Attractive furnished...

Projects ▾ Connect ▾ Build ▾ Settings ▾ Help ▾

My Projects View Trash Guide Report an Issue English ▾ citroenmerc2@gmail.com ▾

Gas_leakage_2

Screen2 ▾ Add Screen ... Remove Screen Publish to Gallery

Designer Blocks

Built-in

- Control
- Logic
- Math
- Text
- Lists
- Dictionaries
- Colors
- Variables
- Procedures

Screen2

- VerticalArrangement1
 - TextBox4
 - Label4
 - TextBox3
 - Label3
 - TextBox1

Rename Delete

Media

Viewer

⚠ 0 ✖ 0

Show Warnings

set global danger to look up in pairs key not found alert

call Web1 .JsonTextDecode jsonText get responseContent

notFound not found

set TextBox2 . Text to join " Valve Status: "

look up in pairs key valve

call Web1 .JsonTextDecode jsonText get responseContent

notFound not found

if look up in pairs key alert

call Web1 .JsonTextDecode jsonText get responseContent

notFound not found

then set Label1 . TextColor to red

set Label1 . Text to upcase " Alert! "

call Notifier1 . ShowMessageDialog

message " Valve Closed "

title " ALERT! Gas Leak "

buttonText " OK "

set Clock1 . TimerInterval to decimal 86400000

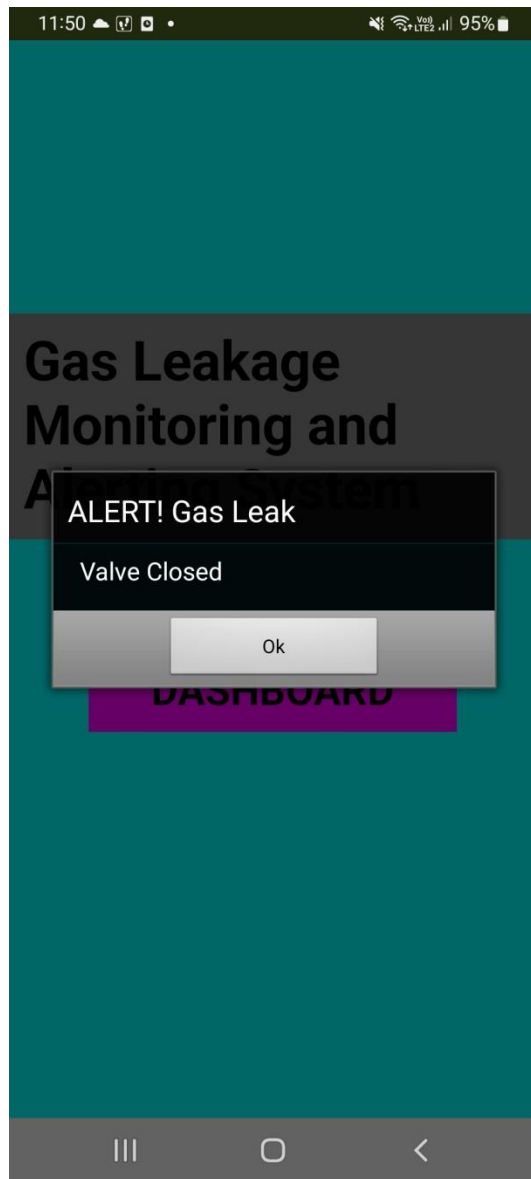
else set Label1 . Visible to false

🎯

+

-

🗑



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