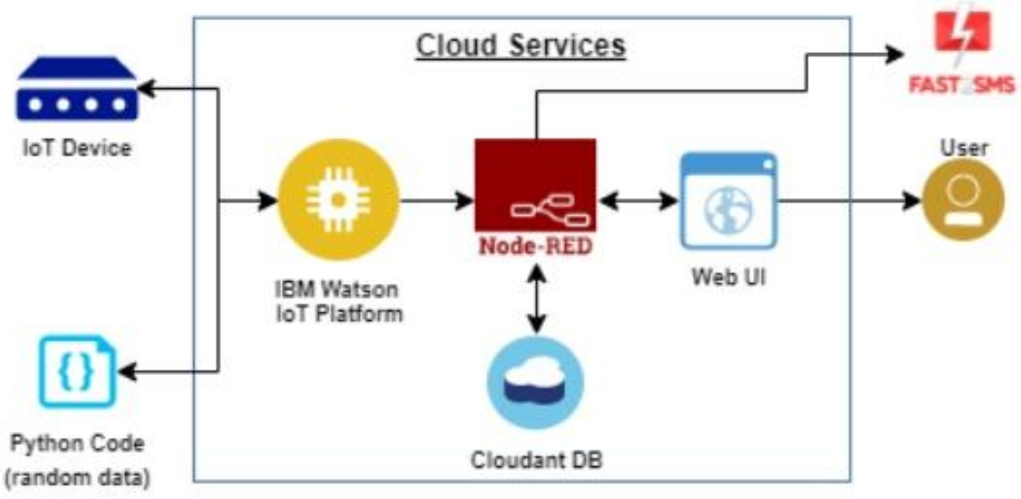
FINAL DELIVERABLES

Date	18 November 2022
Team ID	PNT2022TMID21384
Project Name	Gas Leakage monitoring & Alerting system
Project Team Members	19D051 - MUKILAN T V 19D068 – RAGUL SHANKAR S 19D079 – SANTHOSH S 19D083 – SEEMAN CHAKKARAVARTHY V

OBJECTIVE:

- The design of a sensor-based automatic gas leakage detector with an alert and control system has been proposed.
- This is an affordable, less power using, lightweight, portable, safe, user friendly, efficient, multi featured and simple system device for detecting gas.
- Gas leakage detection will not only provide us with significance in the health department but it will also lead to raise our economy, because when gas leaks it not only contaminates the atmosphere, but also wastage of gases will hurt our economy.
- The need for ensuring safety in workplaces is expected to be the key driving force for the market over the coming years.

FLOW OF THE PROJECT:

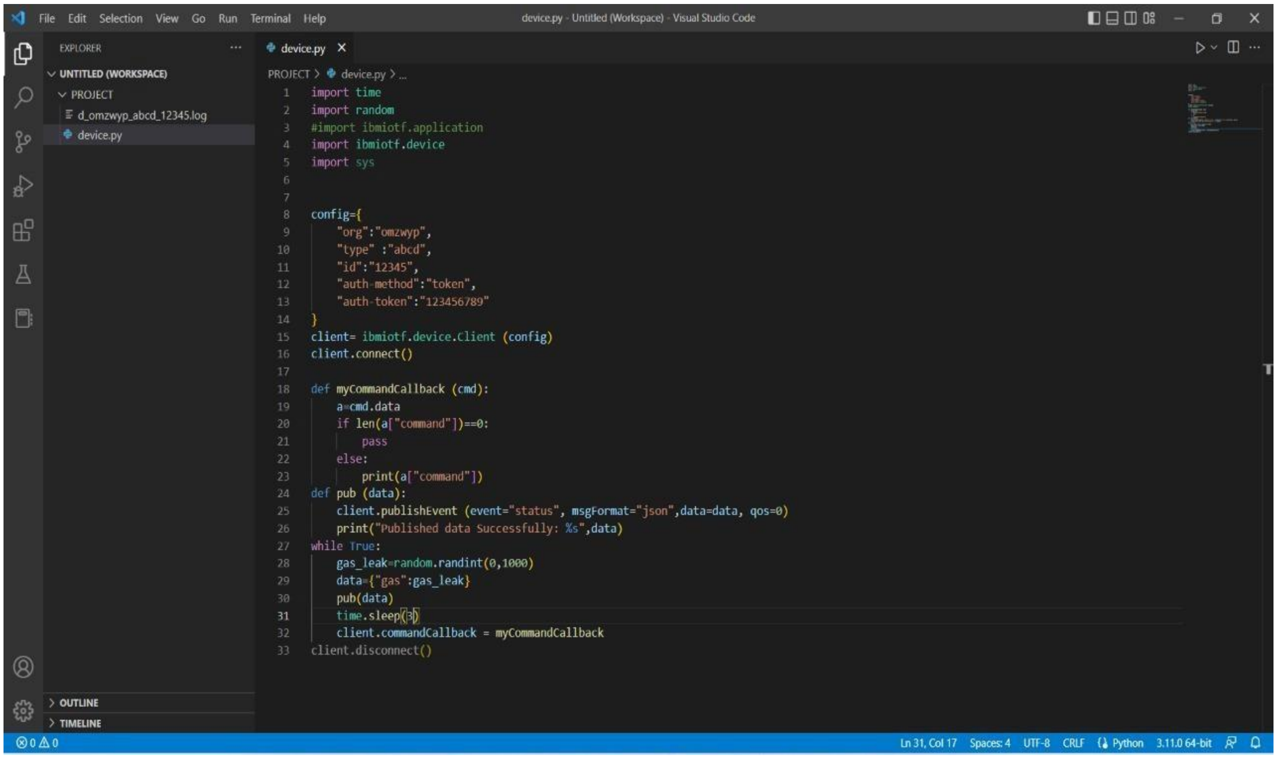


WAYS TO ACHIEVE THE PROJECT FLOW:

STEP 1:

Python code:

Develop a python code for connection establishment between the compiler and the IOT cloud platform with the credentials in the Watson IOT platform.

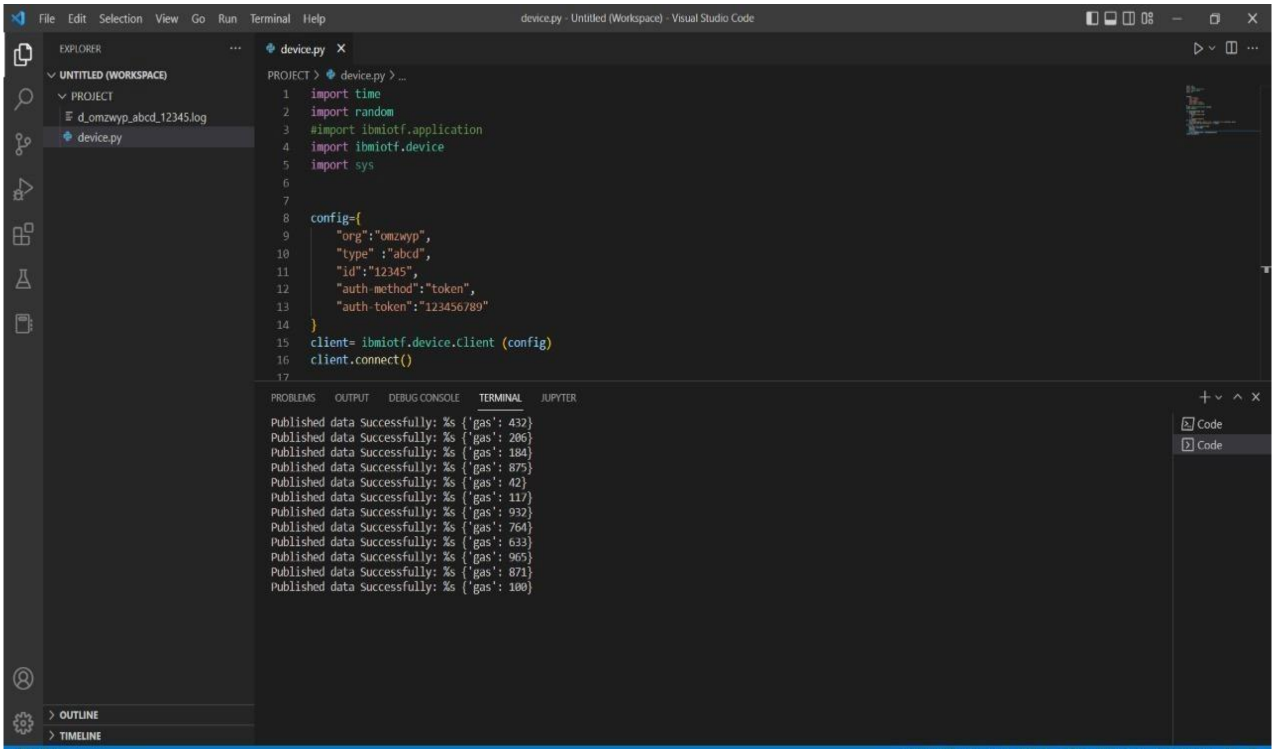

File Edit Selection View Go Run Terminal Help
device.py - Untitled (Workspace) - Visual Studio Code
EXPLORER
UNTITLED (WORKSPACE)
PROJECT
d_omzwyp_abcd_12345.log
device.py
PROJECT > device.py > ...
OUTLINE
TIMELINE
Ln 31, Col 17 Spaces: 4 UTF-8 CRLF Python 3.11.0 64-bit

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


config={
    "org":"omzwyp",
    "type" :"abcd",
    "id":"12345",
    "auth-method":"token",
    "auth-token":"123456789"
}
client= ibmiotf.device.Client (config)
client.connect()

def myCommandCallback (cmd):
    a=cmd.data
    if len(a["command"])==0:
        pass
    else:
        print(a["command"])
def pub (data):
    client.publishEvent (event="status", msgFormat="json",data=data, qos=0)
    print("Published data Successfully: %s",data)
while True:
    gas_leak=random.randint(0,1000)
    data={"gas":gas_leak}
    pub(data)
    time.sleep(3)
    client.commandCallback = myCommandCallback
client.disconnect()
```

program output:


File Edit Selection View Go Run Terminal Help
device.py - Untitled (Workspace) - Visual Studio Code
EXPLORER
UNTITLED (WORKSPACE)
PROJECT
d_omzwyp_abcd_12345.log
device.py
PROJECT > device.py > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
Code
Code
OUTLINE
TIMELINE

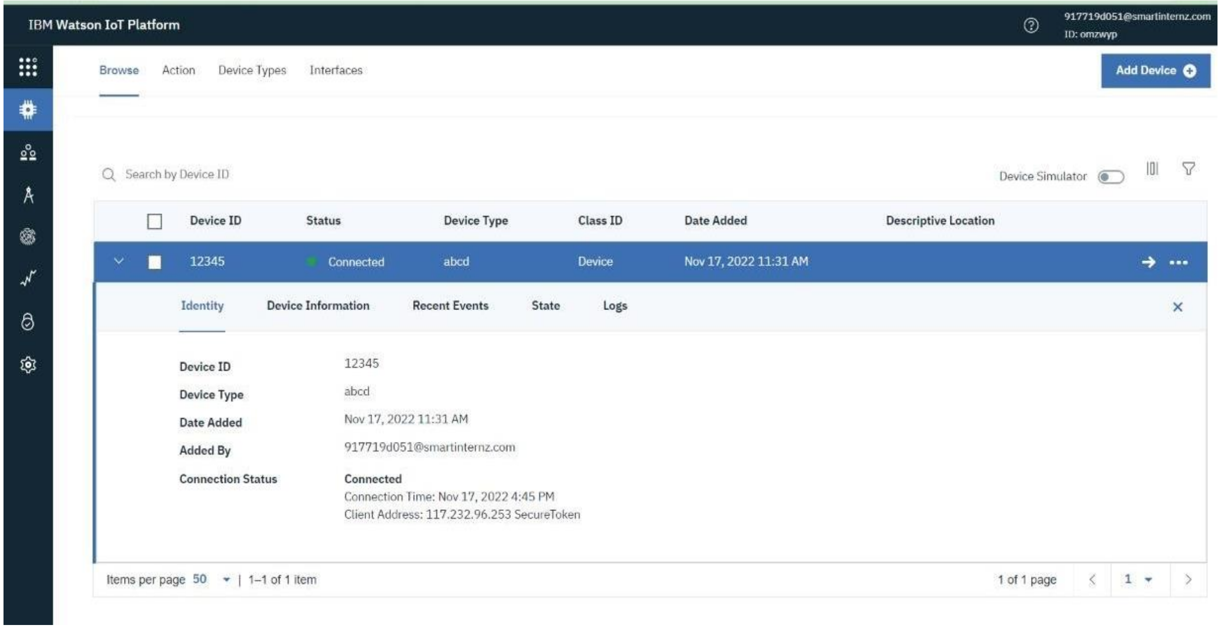
```python
import time
import random
#import ibmiotf.application
import ibmiotf.device
import sys


config={
    "org":"omzwyp",
    "type" :"abcd",
    "id":"12345",
    "auth-method":"token",
    "auth-token":"123456789"
}
client= ibmiotf.device.Client (config)
client.connect()
```

```
Published data Successfully: %s {'gas': 432}
Published data Successfully: %s {'gas': 206}
Published data Successfully: %s {'gas': 184}
Published data Successfully: %s {'gas': 875}
Published data Successfully: %s {'gas': 42}
Published data Successfully: %s {'gas': 117}
Published data Successfully: %s {'gas': 932}
Published data Successfully: %s {'gas': 764}
Published data Successfully: %s {'gas': 633}
Published data Successfully: %s {'gas': 965}
Published data Successfully: %s {'gas': 871}
Published data Successfully: %s {'gas': 100}
```

Step 2:

Create an IBM Watson Device and note down the credentials, after that create an App "Standard App" and node down the API key and Token.

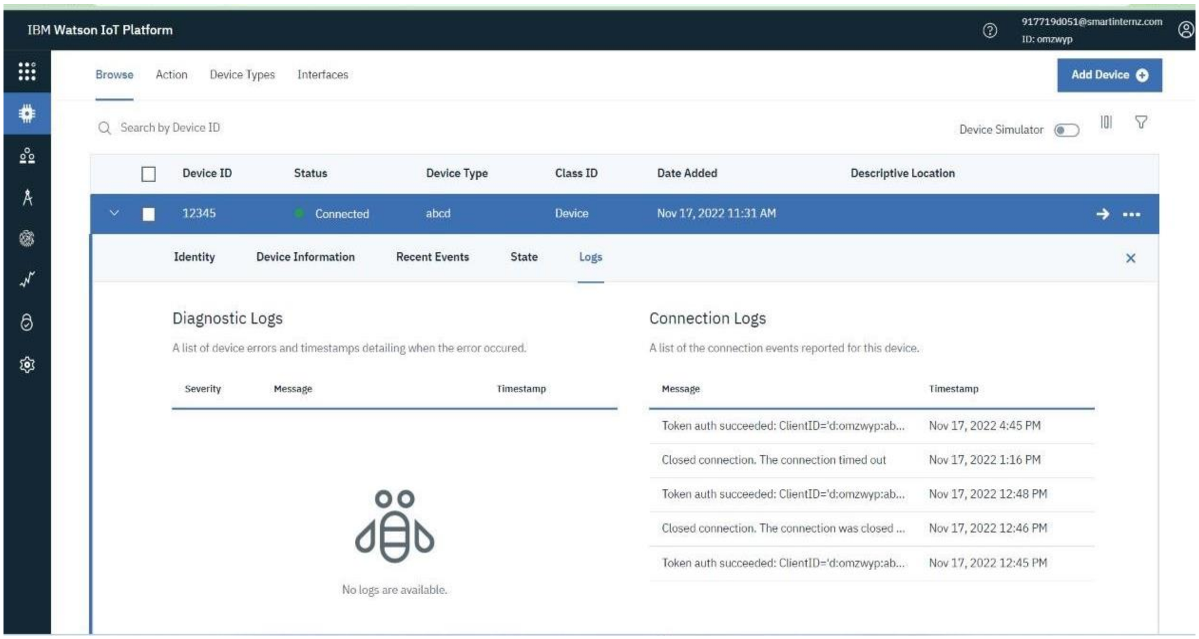


□	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
□	12345	● Connected	abcd	Device	Nov 17, 2022 11:31 AM	

Identity | Device Information | Recent Events | State | Logs

Device ID	12345
Device Type	abcd
Date Added	Nov 17, 2022 11:31 AM
Added By	917719d051@smartinternz.com
Connection Status	Connected Connection Time: Nov 17, 2022 4:45 PM Client Address: 117.232.96.253 SecureToken

Connection is established:



□	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
□	12345	● Connected	abcd	Device	Nov 17, 2022 11:31 AM	

Identity | Device Information | Recent Events | State | Logs

Diagnostic Logs

A list of device errors and timestamps detailing when the error occured.

Severity	Message	Timestamp

No logs are available.

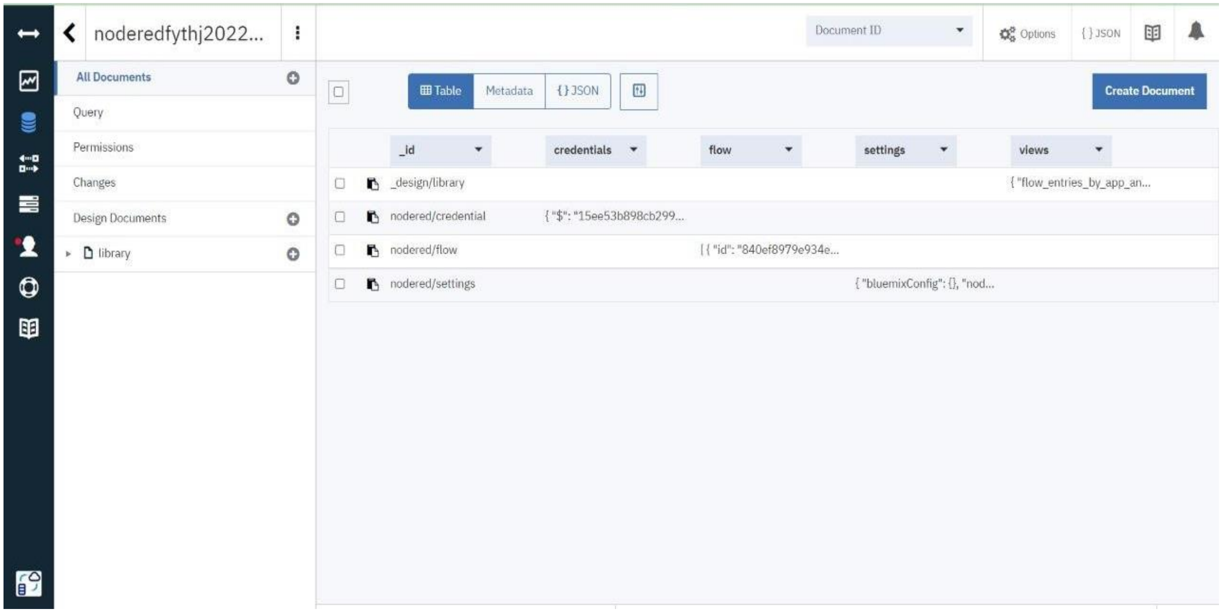
Connection Logs

A list of the connection events reported for this device.

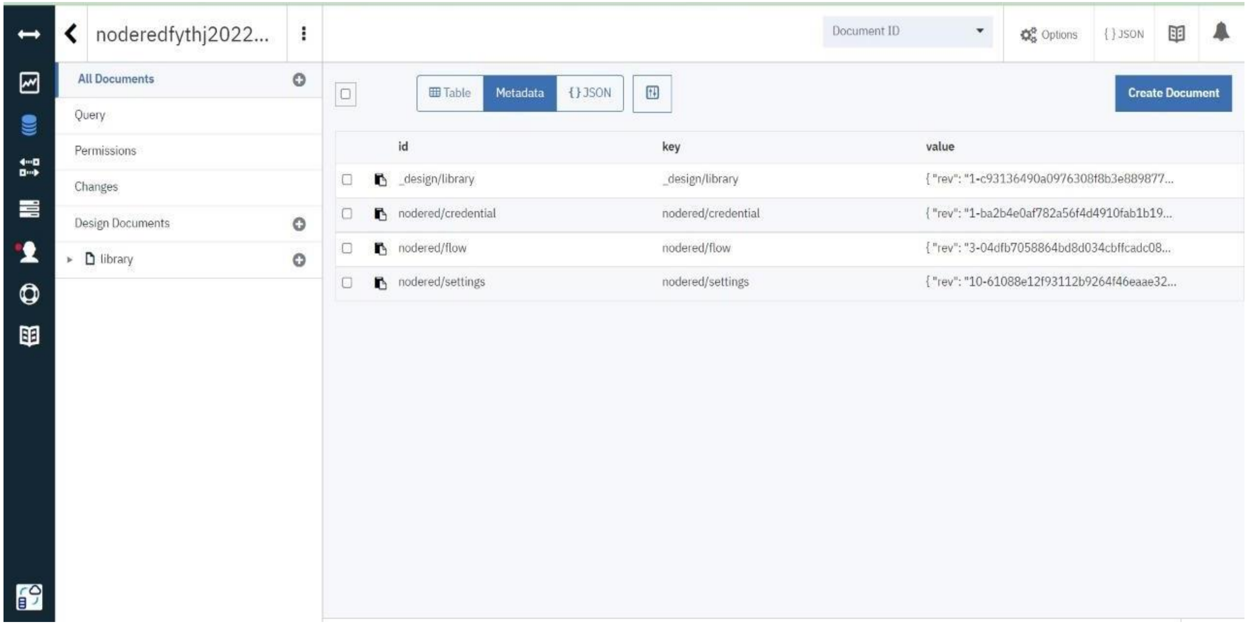
Message	Timestamp
Token auth succeeded: ClientID='d:omzwyp:ab...	Nov 17, 2022 4:45 PM
Closed connection. The connection timed out	Nov 17, 2022 1:16 PM
Token auth succeeded: ClientID='d:omzwyp:ab...	Nov 17, 2022 12:48 PM
Closed connection. The connection was closed ...	Nov 17, 2022 12:46 PM
Token auth succeeded: ClientID='d:omzwyp:ab...	Nov 17, 2022 12:45 PM

STEP 3:

Storing data in IBM cloudant DB through node-RED functions.



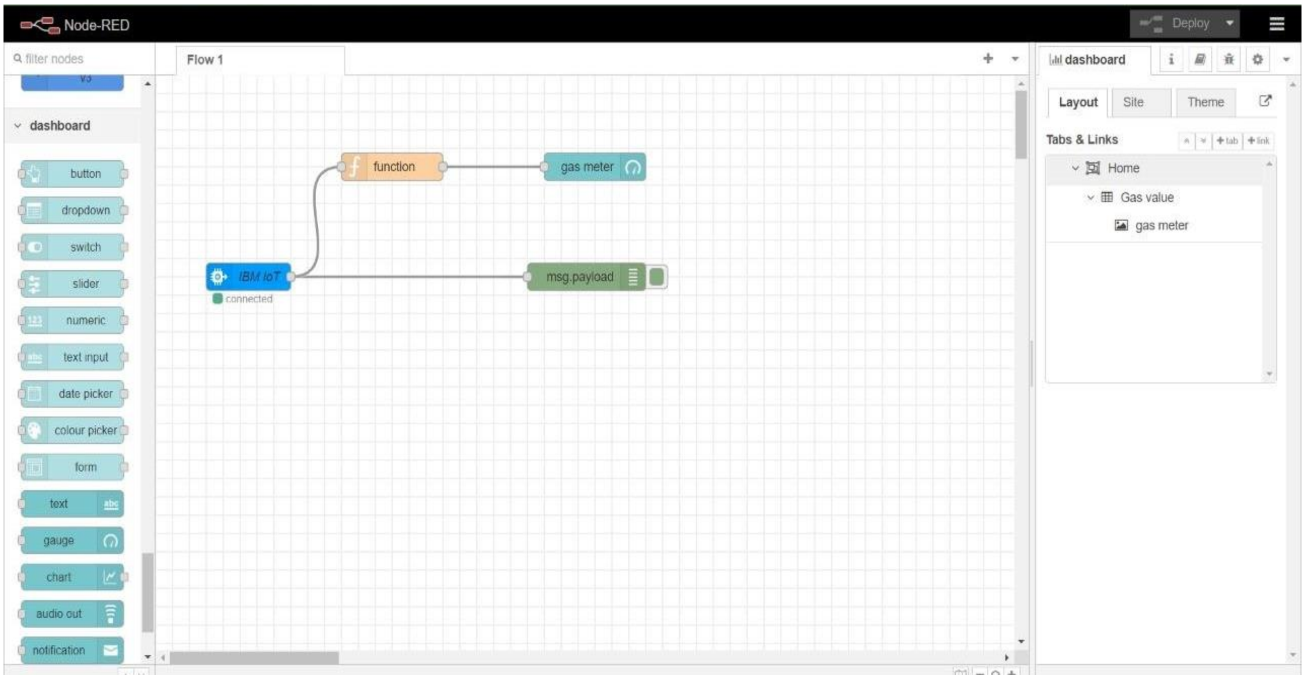
_id	credentials	flow	settings	views
_design/library				{ "flow_entries_by_app_an...
nodered/credential	{ "$": "15ee53b898cb299...			
nodered/flow		[{ "id": "840ef8979e934e...		
nodered/settings			{ "bluemixConfig": {}, "nod...	



id	key	value
_design/library	_design/library	{ "rev": "1-c93136490a0976308f8b3e889877...
nodered/credential	nodered/credential	{ "rev": "1-ba2b4e0af782a56f4d4910fab1b19...
nodered/flow	nodered/flow	{ "rev": "3-04dfb7058864bd8d034cbffcadc08...
nodered/settings	nodered/settings	{ "rev": "10-61088e12f93112b9264f46eaae32...

STEP 4:

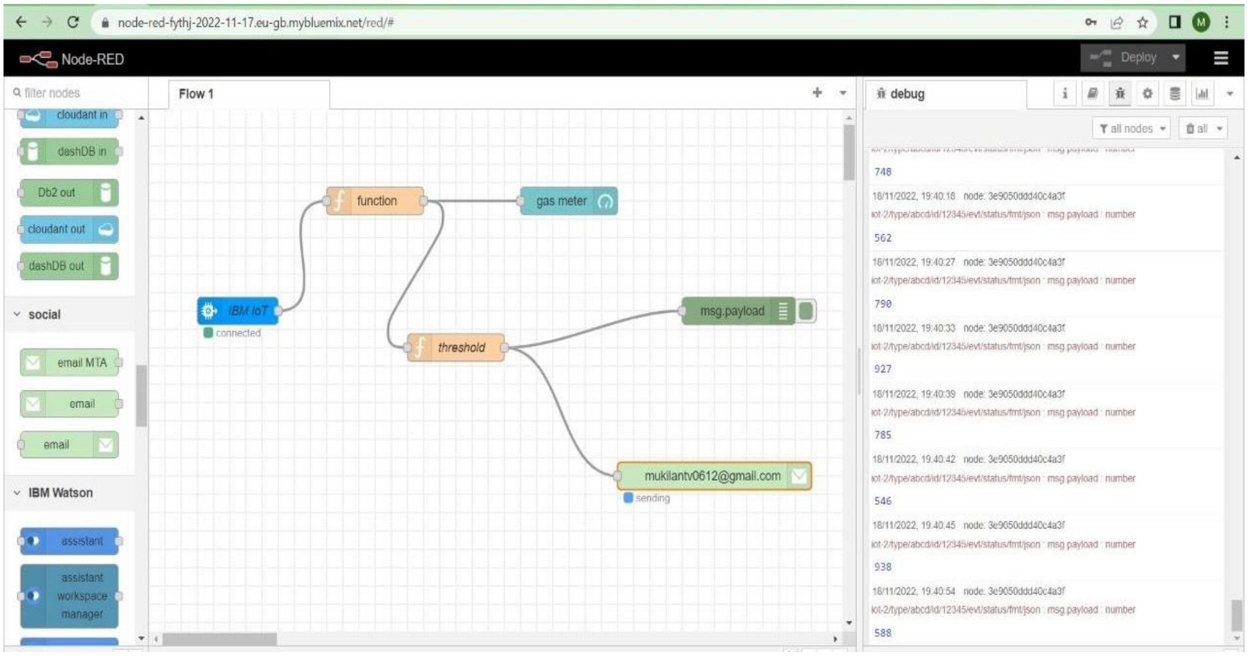
The board connect with the cloud and node-red platform and send information about the gas Leakage.



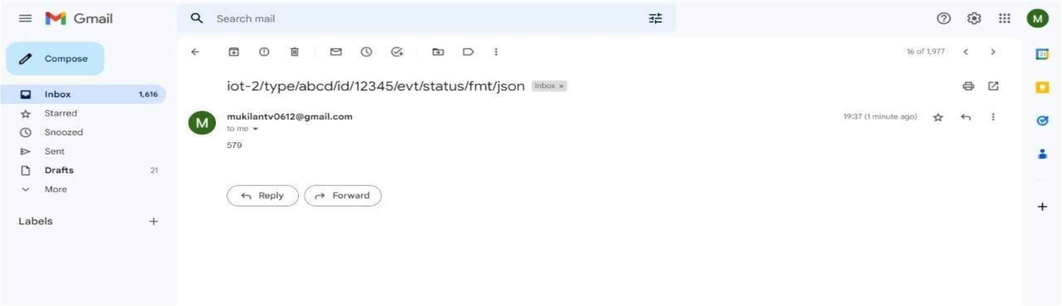
Step 5:

To send Alert Notification to the user

I. Notification through E-Mail:



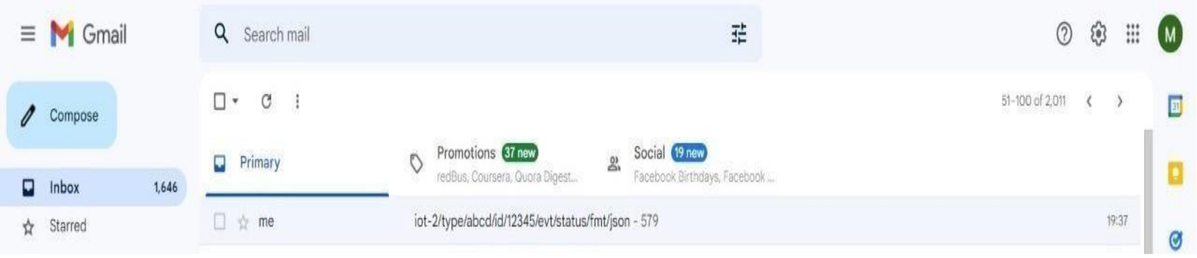
```
iot-2/type/abcd/id/12345/evt/status/fmt/json : msg.payload : number
748
18/11/2022, 19:40:18   node: 3e9050ddd40c4a3f
iot-2/type/abcd/id/12345/evt/status/fmt/json : msg.payload : number
562
18/11/2022, 19:40:27   node: 3e9050ddd40c4a3f
iot-2/type/abcd/id/12345/evt/status/fmt/json : msg.payload : number
790
18/11/2022, 19:40:33   node: 3e9050ddd40c4a3f
iot-2/type/abcd/id/12345/evt/status/fmt/json : msg.payload : number
927
18/11/2022, 19:40:39   node: 3e9050ddd40c4a3f
iot-2/type/abcd/id/12345/evt/status/fmt/json : msg.payload : number
785
18/11/2022, 19:40:42   node: 3e9050ddd40c4a3f
iot-2/type/abcd/id/12345/evt/status/fmt/json : msg.payload : number
546
18/11/2022, 19:40:45   node: 3e9050ddd40c4a3f
iot-2/type/abcd/id/12345/evt/status/fmt/json : msg.payload : number
938
18/11/2022, 19:40:54   node: 3e9050ddd40c4a3f
iot-2/type/abcd/id/12345/evt/status/fmt/json : msg.payload : number
588
```



iot-2/type/abcd/id/12345/evt/status/fmt/json Inbox ×

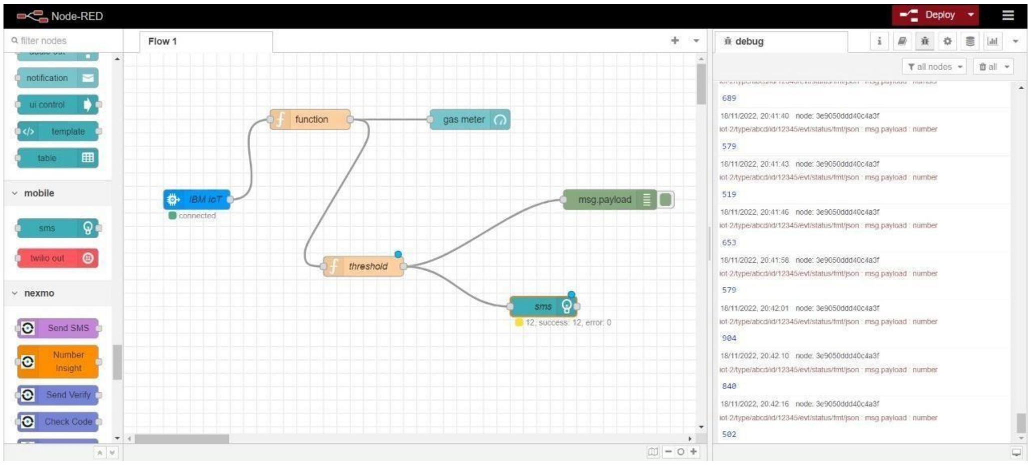
mukilantv0612@gmail.com
to me
19:37 (1 minute ago)

579

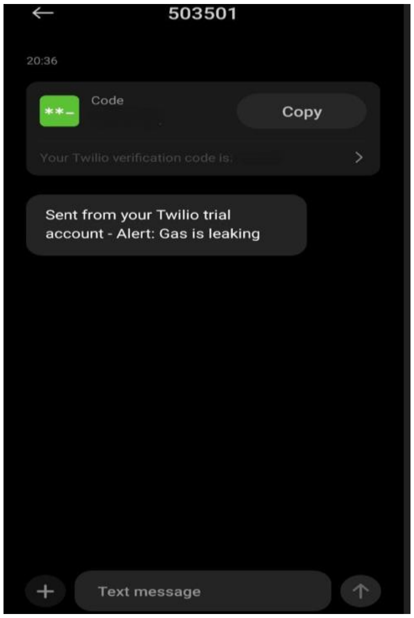


me iot-2/type/abcd/id/12345/evt/status/fmt/json - 579 19:37

II. Notification through SMS:



```
689
18/11/2022, 20:41:40   node: 3e9050ddd40c4a3f
iot-2/type/abcd/id/12345/evt/status/fmt/json : msg.payload : number
579
18/11/2022, 20:41:43   node: 3e9050ddd40c4a3f
iot-2/type/abcd/id/12345/evt/status/fmt/json : msg.payload : number
519
18/11/2022, 20:41:46   node: 3e9050ddd40c4a3f
iot-2/type/abcd/id/12345/evt/status/fmt/json : msg.payload : number
653
18/11/2022, 20:41:58   node: 3e9050ddd40c4a3f
iot-2/type/abcd/id/12345/evt/status/fmt/json : msg.payload : number
579
18/11/2022, 20:42:01   node: 3e9050ddd40c4a3f
iot-2/type/abcd/id/12345/evt/status/fmt/json : msg.payload : number
904
18/11/2022, 20:42:10   node: 3e9050ddd40c4a3f
iot-2/type/abcd/id/12345/evt/status/fmt/json : msg.payload : number
840
18/11/2022, 20:42:16   node: 3e9050ddd40c4a3f
iot-2/type/abcd/id/12345/evt/status/fmt/json : msg.payload : number
502
```

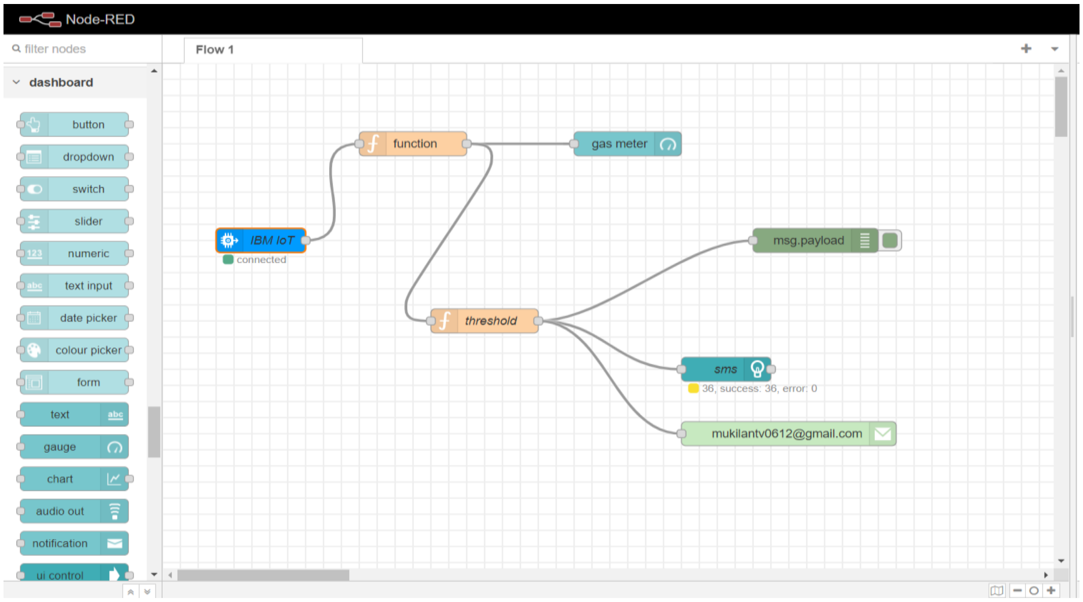


20:36

Your Twilio verification code is.

Sent from your Twilio trial account - Alert: Gas is leaking

III. Combined:



CONCLUSION:

The objectives are achieved and the data flow is constructed as per the project flow mentioned in the smartinternz guided project.