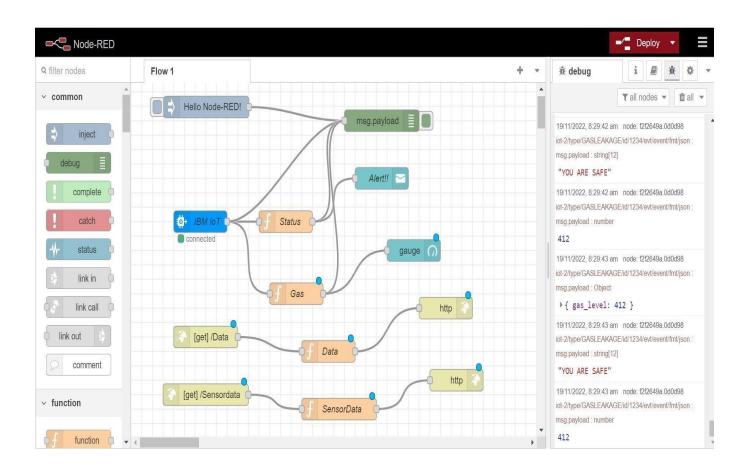
SPRINT-IV WEB UI

Team ID : PNT2022TMID41330

Project Name: Gas Leakage Monitoring and Alerting System

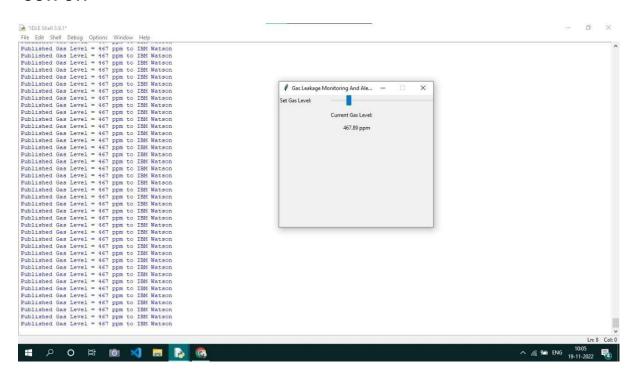
Web UI:



CODE:

```
Final.py - C:/Python/Python310/Final.py (3.10.7)
                                                                                                                                                         o ×
File Edit Format Run Options Window Help
# current gas level label
current gas label = ttk.Label(root,text='Current Gas Level:') current gas label.grid(row=1,columnspan=2,sticky='n',ipadx=10,ip ady=10)
# Gas level label (value gets displayed here)
gas_label = ttk.Label(root,text=str(get_current_gas()) +" ppm") gas_label.grid(row=2,columnspan=2,sticky='n')
 def publisher thread():
thread = Thread(target=publish_data) thread.start()
 def publish_data():
# Exception Handling 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() # Connect to IBM Watson IoT Platform while True:
gas_level = int(current_gas.get())
data = {'gas_level' : gas_level} def myOnPublishCallback():
print("Published Gas_Level = %s ppm" % gas_level, "to IBM W
success = deviceCli.publishEvent("event", "json", data, qos=0, on_publish=myOnPublishCallback)
if not success:
print("Not connected to IoTF") time.sleep(1)
publisher_thread()
root.mainloop() # startup Tkinter GUI
# Disconnect the device and application from the cloud deviceCli.disconnect()
                                                                                                                                                          Ln: 76 Col: 0
```

OUTPUT:



Testing Web UI:

