Sprint Delivery – 1

Date	17 November 2022
Team ID	PNT2022TMID28572
Project Name	Hazardous Area Monitoring for Industrial Power plant powered by IoT
Maximum Marks	4 Marks

1. Introduction

In some industrial plants, some areas are to be monitored from time to time. Sometimes the conditions may become critical, leading to property loss and human loss. Industrial accidents are as old as the industry itself and so are preventive measures. The Standards for Explosive Areas or Atmospheres have also evolved diversely worldwide, based on the local needs of the industries for the overall safe operation of the plants. Explosions and fire are two of the major constituents of these mishaps. Depending upon the environment, these can be termed 'Accidents' or fade away as simply the 'incidents' or 'Near Misses' in the safety officers' statistics. The first step to logically is to start defining and understanding some of the terms used in the whole scope of toss prevention in accidents due to explosion and fire.

2. Problem Statement

To fulfil the modern-day human lifestyle and requirements many industrial plants play a major role. But with the exponential increase in growth of the industrial plants the safety and security of Workers and Engineers exponentially go down. This is due to a lack of Knowledge of hazards in the industrial plants among the employees. To overcome this bad situation industrial plants, need to develop some industrial safety regulations and maintain them.

- A Plant Manager needs to be informed of possible hazardous areas because they could risk the workers' lives in the facility.
- A technician needs to be informed whenever he is entering a hazardous area because it can lead to dangerous health complications to his/her or may cause disaster.
- The Safety Inspector needs to easily examine whether the operations of the plant are hazardous or not. Because he needs to provide accurate reports to ensure safety.

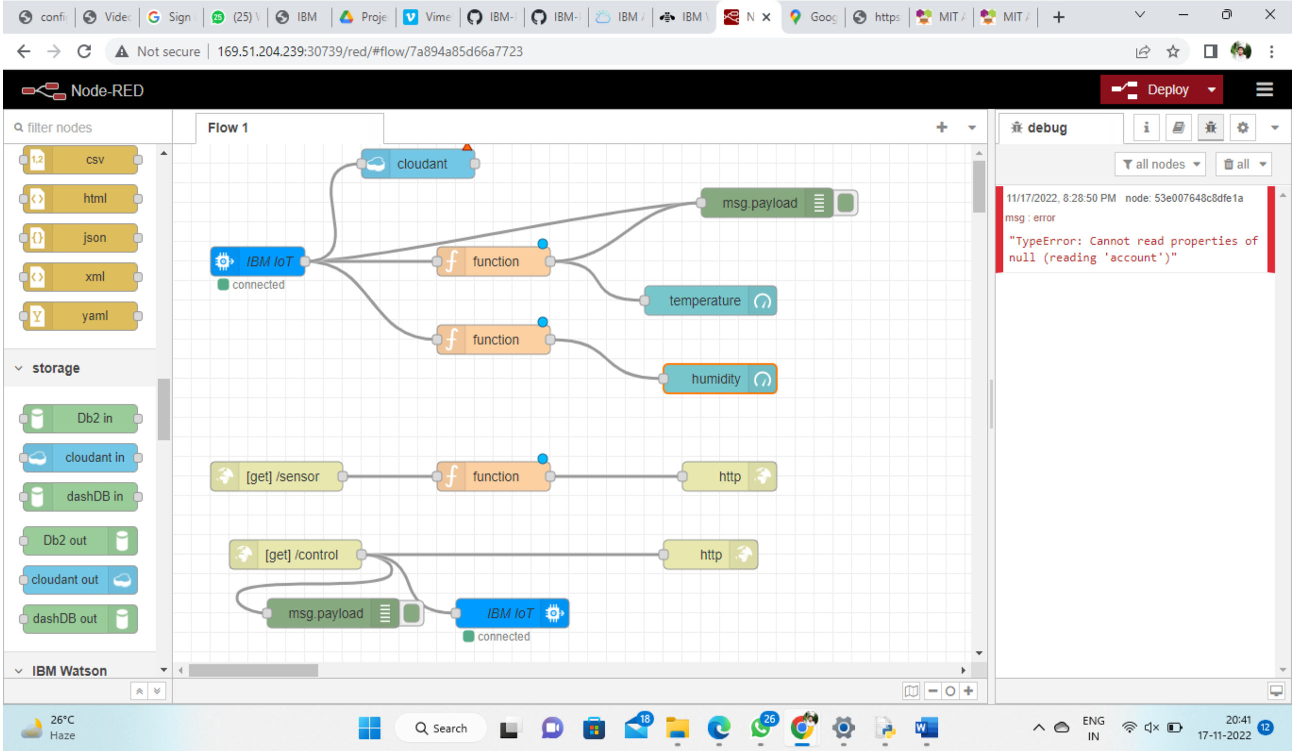
3. Proposed Solution

Hazardous Area Monitoring Industrial Plant powered by IoT, we are developing a system which will automatically monitor the industrial applications and generate Alerts/Alarms or make intelligent decisions using the concept of IoT. Every devout will be acting as a beacon and it is connected to temperature sensors. An alert message is also sent to the mobile whenever high temperature (or) toxic gases are detected within the area through SMS using API. Through this wearable gadget, the information is sent to the cloud database and through which the dashboard, the admins of that specific plant can see the information and take fundamental activities on the off chance that required.

4. Required Software Installation

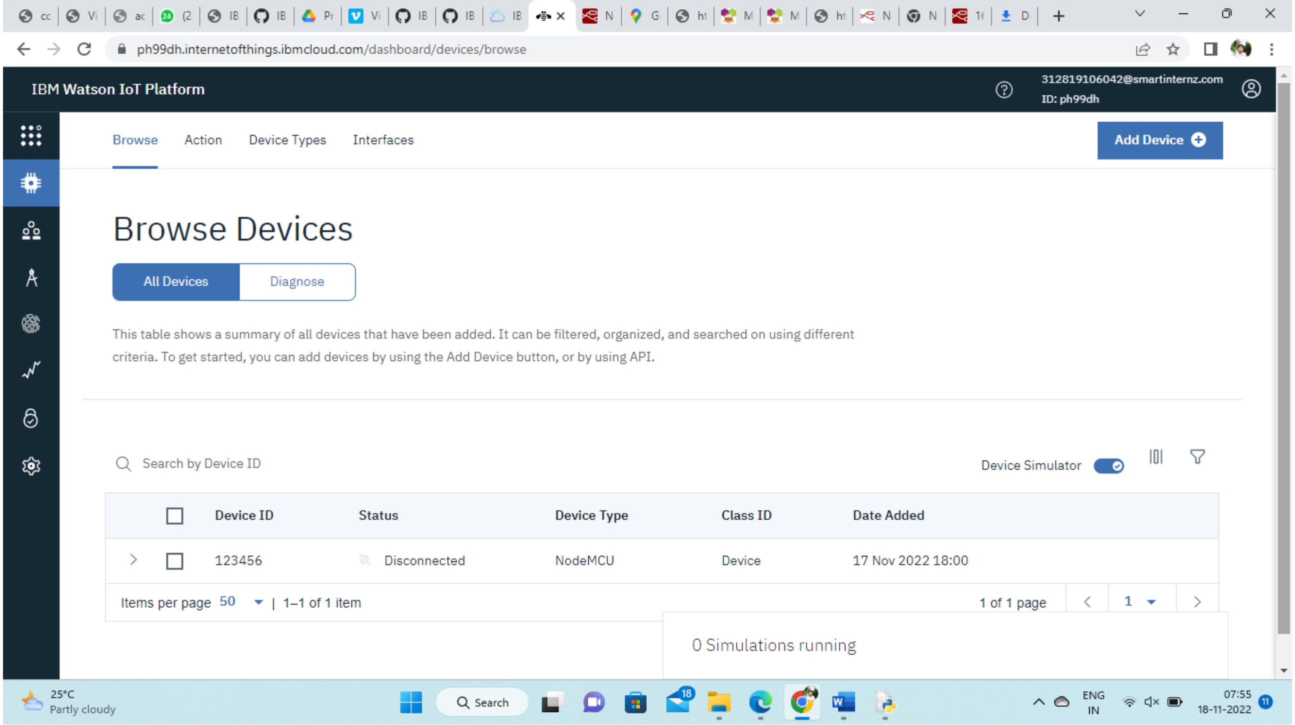
Node-Red

Node-RED is a flow-based development tool for visual programming developed originally by IBM for wiring together hardware devices, APIs and online services as part of the Internet of Things. Node-RED provides a web browser-based flow editor, which can be used to create JavaScript functions.



11/17/2022, 8:28:50 PM node: 53e007648c8dfe1a
msg : error
"TypeError: Cannot read properties of null (reading 'account')"

Installation of IBM IoT and Dashboard nodes for Node-Red:



ph99dh.internetofthings.ibmcloud.com/dashboard/devices/browse

IBM Watson IoT Platform

312819106042@smartinternz.com
ID: ph99dh

Browse Devices

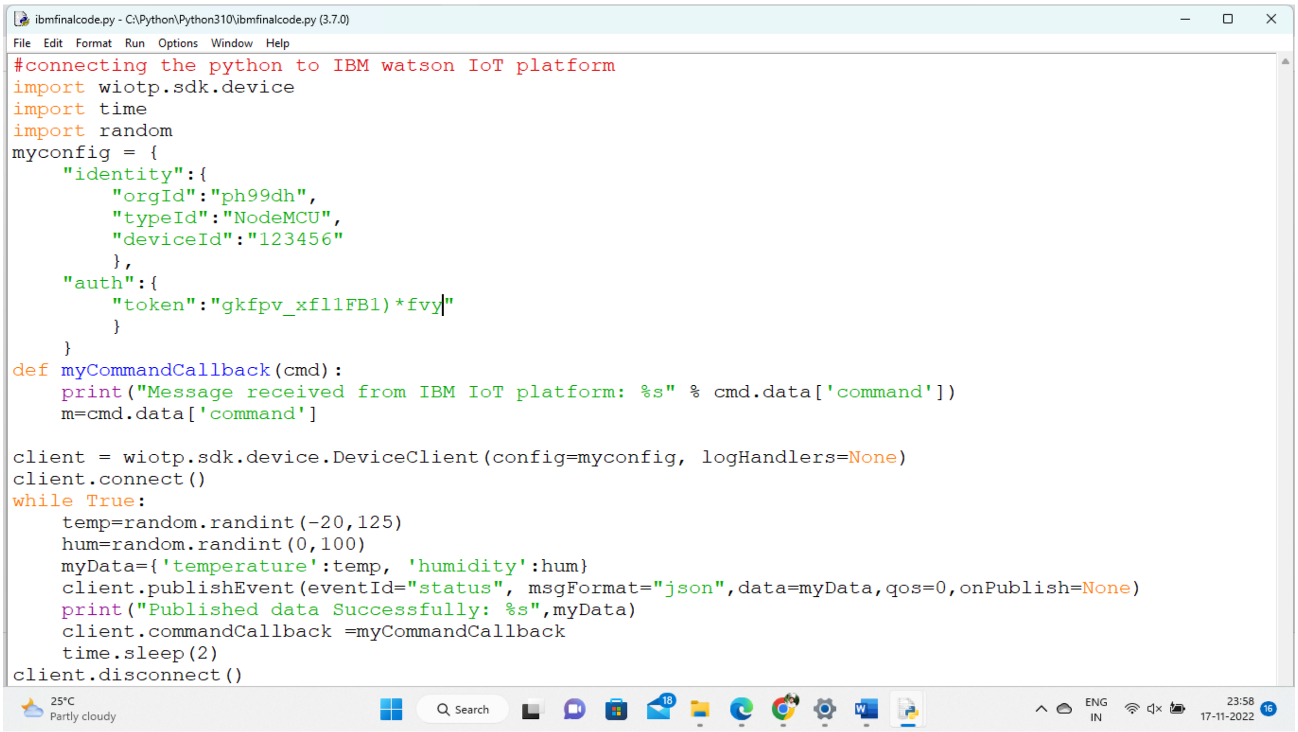
This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

	Device ID	Status	Device Type	Class ID	Date Added
□	123456	Disconnected	NodeMCU	Device	17 Nov 2022 18:00

Items per page 50 | 1–1 of 1 item   1 of 1 page   1

0 Simulations running

Python IDE:



ibmfinalcode.py - C:\Python\Python310\ibmfinalcode.py (3.7.0)

```
#connecting the python to IBM watson IoT platform
import wiotp.sdk.device
import time
import random
myconfig = {
    "identity":{
        "orgId":"ph99dh",
        "typeId":"NodeMCU",
        "deviceId":"123456"
        },
    "auth":{
        "token":"gkfpv_xfl1FB1)*fvy"
        }
    }
def myCommandCallback(cmd):
    print("Message received from IBM IoT platform: %s" % cmd.data['command'])
    m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myconfig, logHandlers=None)
client.connect()
while True:
    temp=random.randint(-20,125)
    hum=random.randint(0,100)
    myData={'temperature':temp, 'humidity':hum}
    client.publishEvent(eventId="status", msgFormat="json",data=myData,qos=0,onPublish=None)
    print("Published data Successfully: %s",myData)
    client.commandCallback =myCommandCallback
    time.sleep(2)
client.disconnect()
```