**Allergy Notification System based on iot**

**Introduction**

a.overview

The Internet of Things is a newly emerging term for the new generation of the internet which allows understanding between interconnected devices.IoT acts as an assistant in healthcare and plays an extremely important role in wide scope of medicinal services applications.Most of the applications implemented using node-red and mit app inventor.In upcoming days applications of Iot can save life of so many people by monitoring the people activities.

b.Purpose

The main purpose of the project is send message to person.Most of the people are allergenic to common substances like hotair,cloudwaves,chemicals,

Smoke,humidity,etc.Sometimes the cases may become worse.This smart allergen notifier heps the people in knowing about the allergic conditions and intimates him about it.There will be a smart device which monitors the parameters like dust level,temperature,humidity,UV radiations and air quality.

If any of the parameters are high in the environment the person will be notified to take necessary precautions.

**Literature Summary**

a.Existing Problem

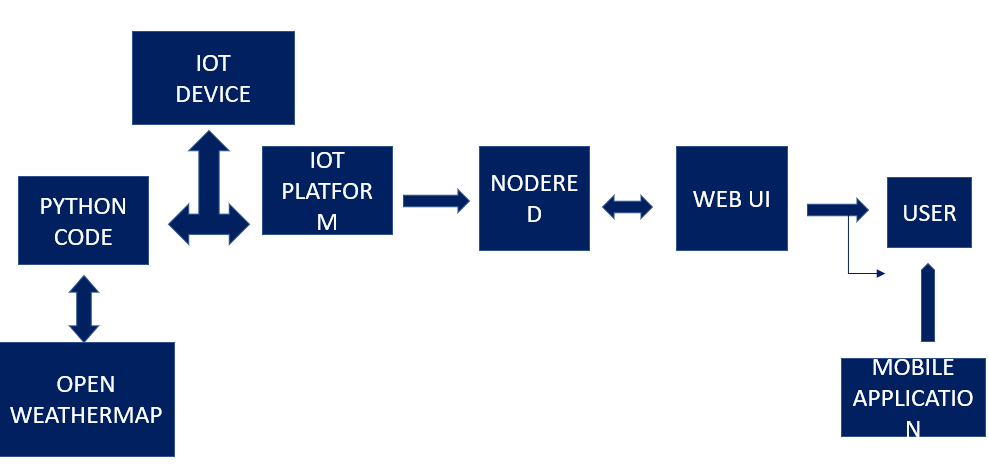
As we noted,and many know all too well,a change in the season can trigger allergies.Below are some common allergies associated with the transition of seasons,weather.the better that you understand how seasonal weather impact your symptoms,the better you will be able to prepare for your personal allergy season.Since most allergy medications are more effective if taken before the onset of symptoms,this foresight can be very valuable

b. Proposed Solution

Through this project we can save may life from this allergies .In this project the main element is sending message.Whenever the temperature or any other parameters which cause allegry are noticed by the ibm cloud it will send message to the given phonenumber.He/She can who see the message move to safe place.it works as indicator and alerts the people in short time.

**Theoretical Analysis**

a.Block Diagram



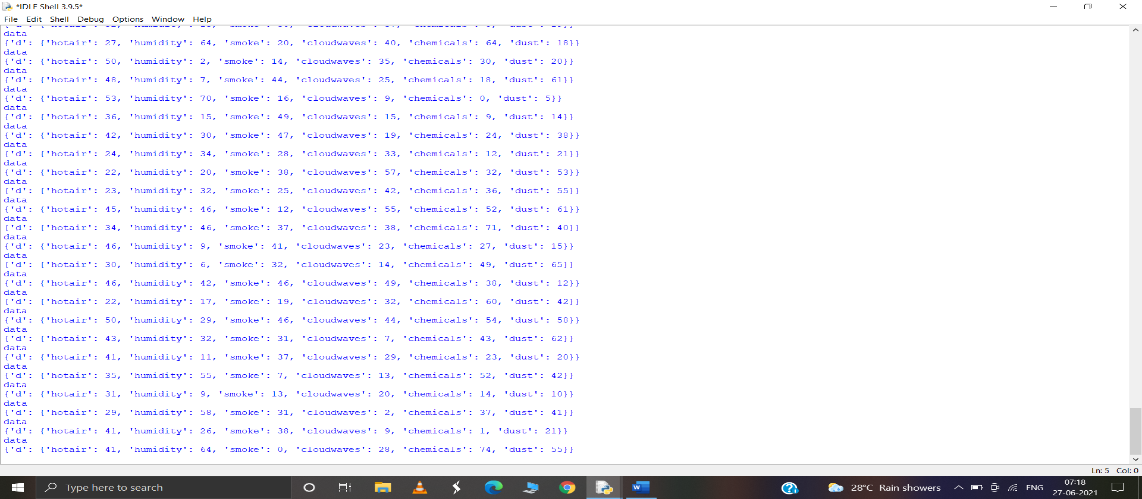
This is the blockdiagram of the allergy notification system

b. Hardware/Software Designing

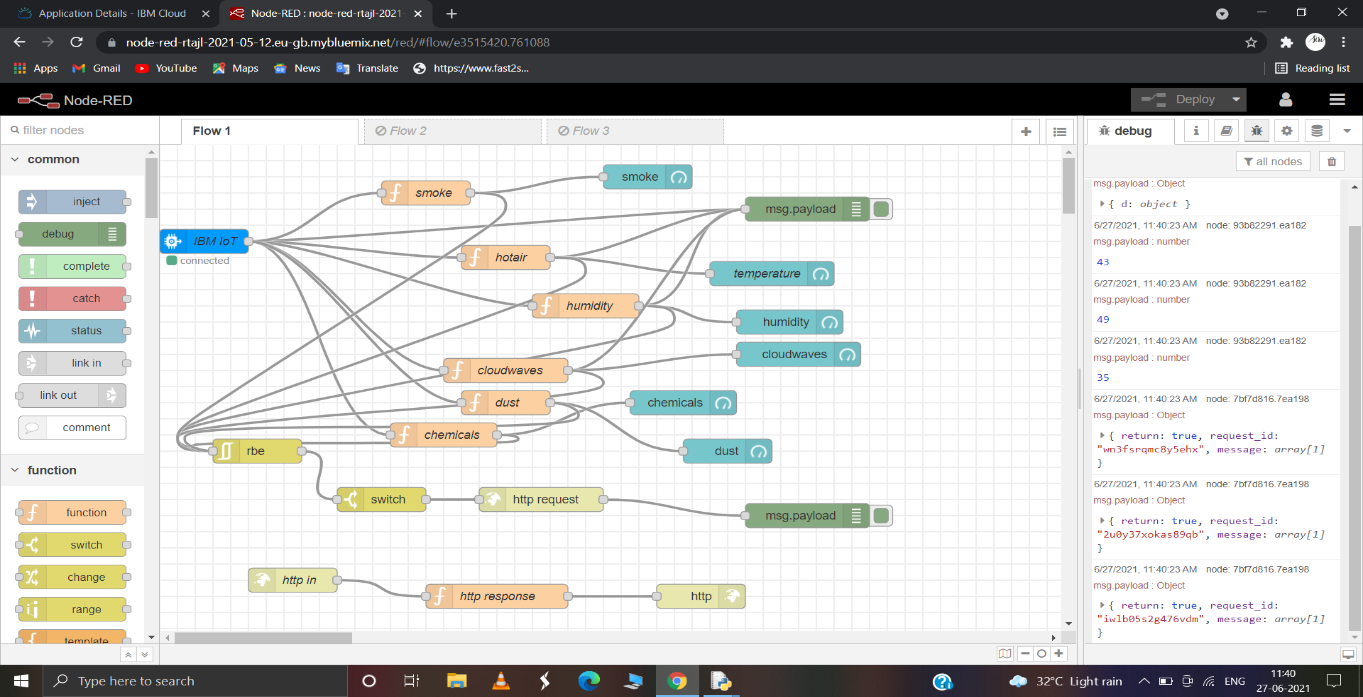
* In this we used phythoncode
* Nodered platform to store data
* Mit App inventor to create mobile application
* Fastsms url to send message

**Experimental Investigations**

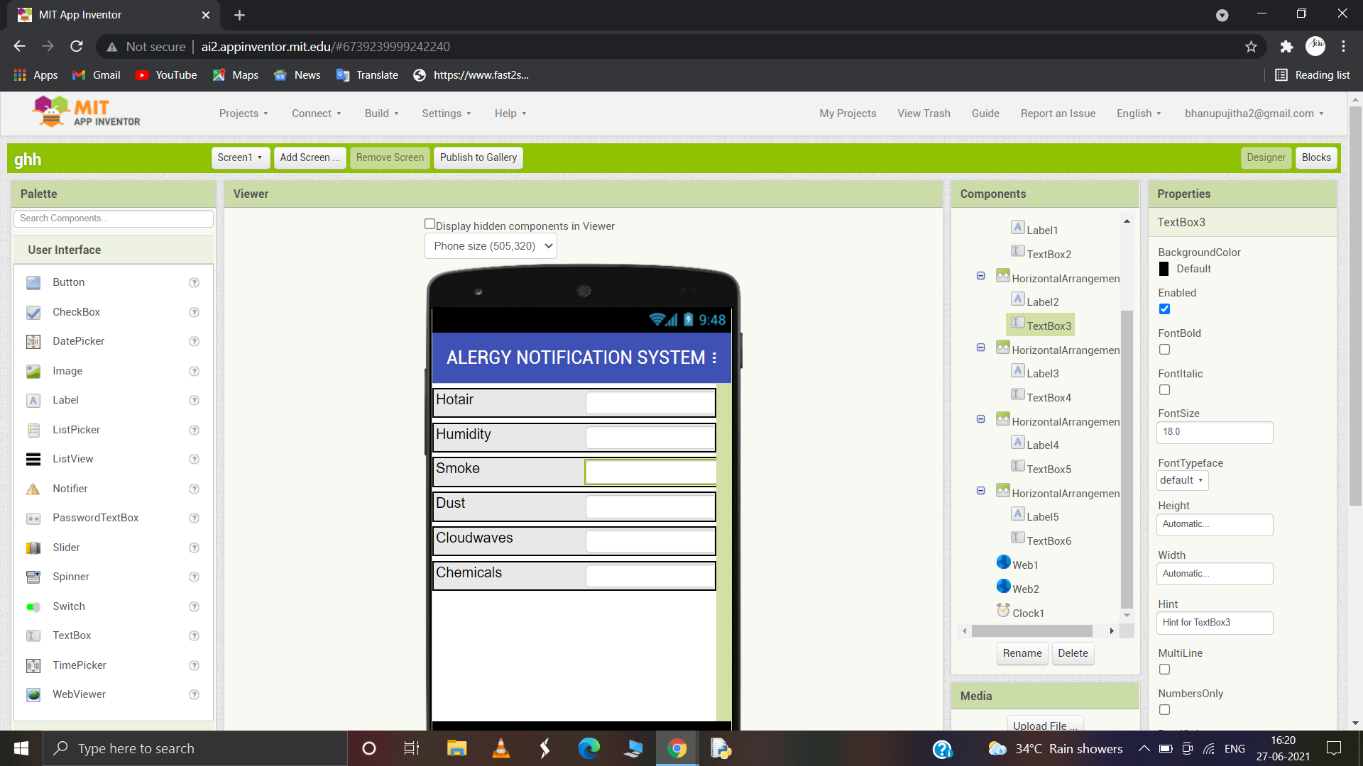
In this project we first we connected device with device credentials and then generated random values of required parameters like hotair,humidity,cloudwaves by using random function in pythoncode.



Through the code the data is sent to ibm cloud input node.To observe the Ui dashboard we need function node it is connected to ibm cloud input.Debug node which is connected to function node is to deplay the message in debug panel.

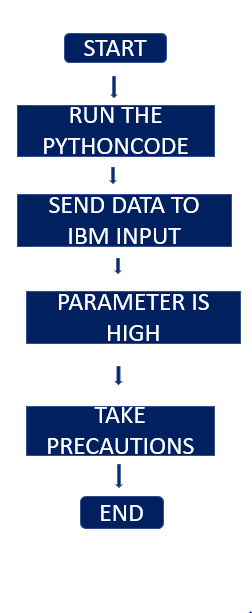
Then for getting message we used three nodes rbe,switch and http node.In http node fastsmas link is given that link contain required information to whih number we have to send message and what is the message we have to send.Switch is used to set condition whenever the we need to send take precautions.

Then we created an mit app in following process first in mit app inventor we created design of required parameters which we have to display.Then created blocks according to it.For data transmission from node-red to mit app we made required connections in node-red by using function and http node.



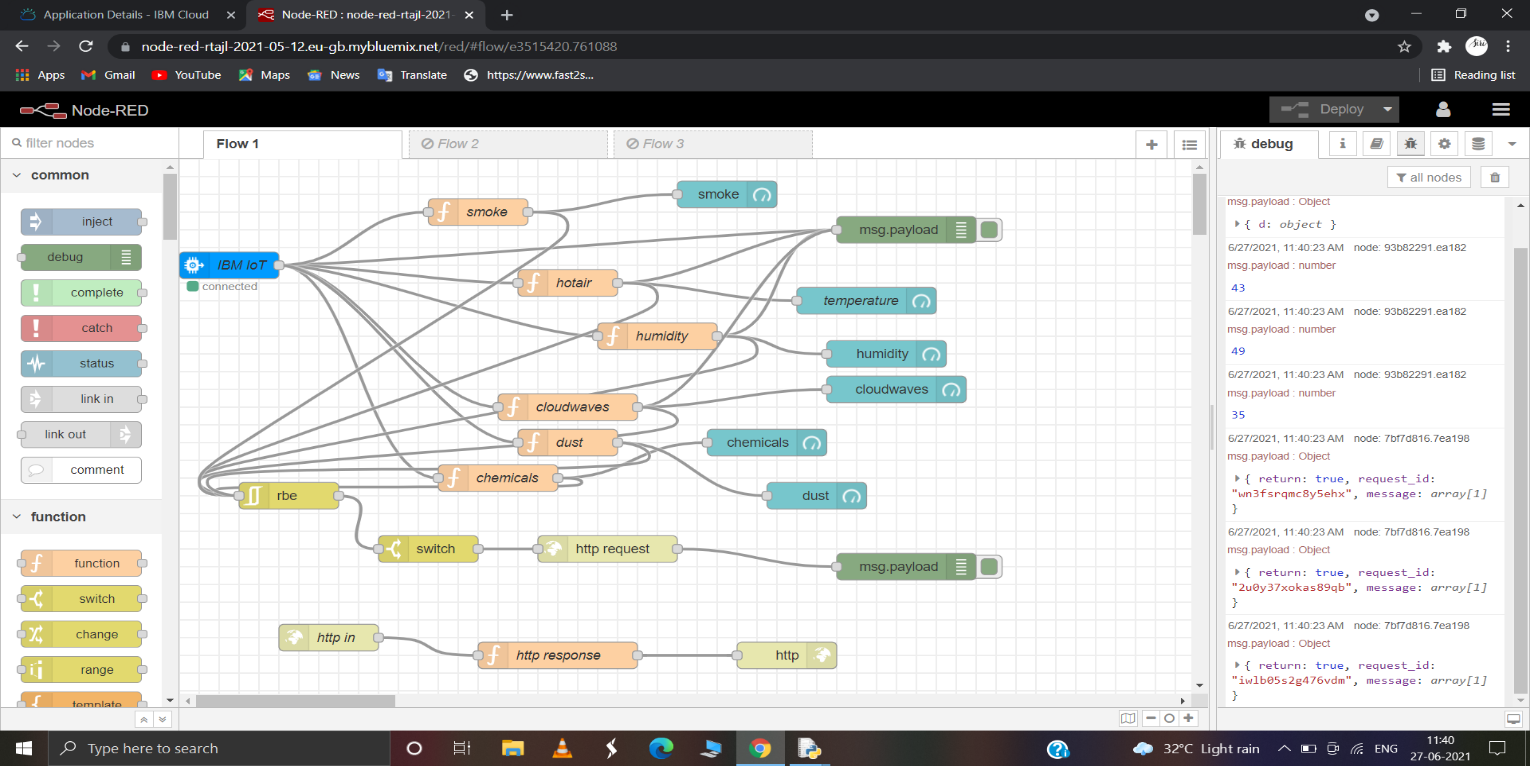
The data is displayed on the mit app when the related code is run.Finally when the code run the generated is went to ibm input and then it is displayed on mit app and send message to given phone number.

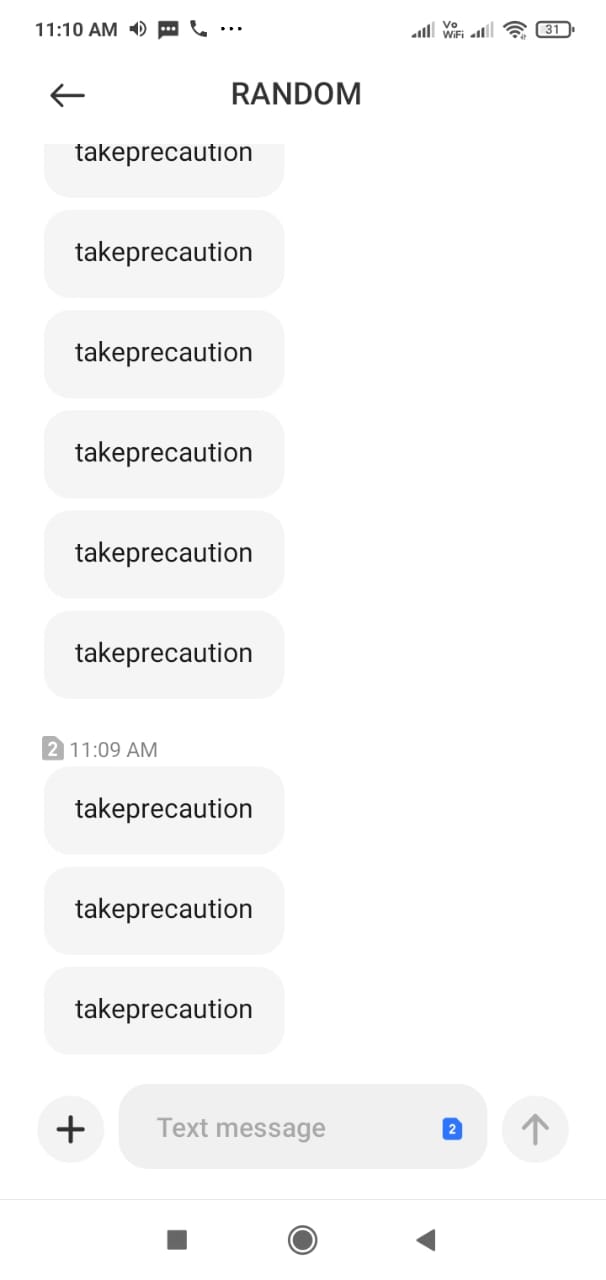
**Flowchart**

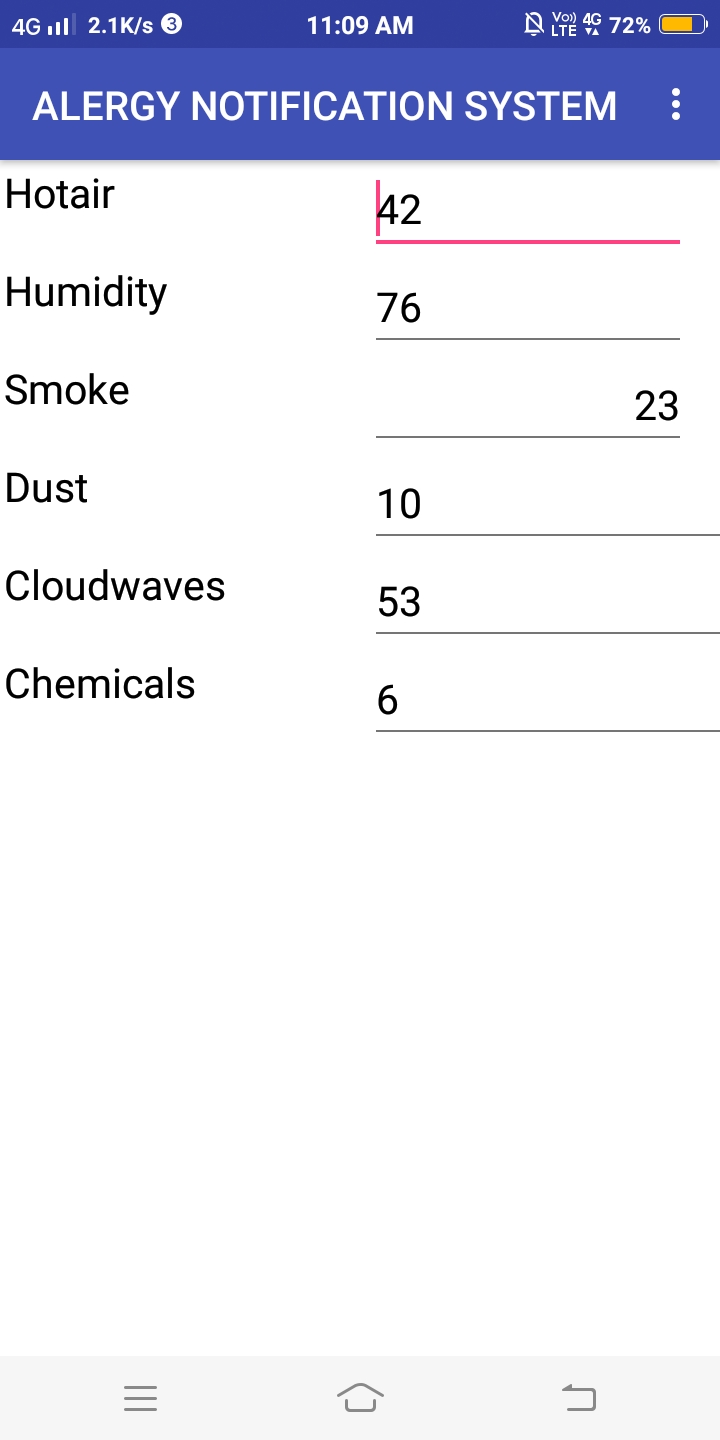


**Result**

Message sent to the given phone number has take precautions and displayed the required parameters in mobile application.







**Advantages & Disadvantages**

a.Advantages

* + Its highly accurate and reliable.
  + Consumes less time to send notification about weather changes.
  + Alert message saves the life of a person.
  + It helps in faster detection of input sensors.

b.Disadvantages

* The sensors are costly for hardware designing.
* If there are errors debugging is difficult.
* Due to poor signals one may not receive the notification.

**Applications**

* Monitoring health
* Pollution control
* Weather alerting systems

**Conclusion**

Allergy notification system helps the people in less time.Its easily identifies the changes in the weather parameters like humidity,hotair, cloudwaves, dust level, etc.Wherever identifies the changes sends message in few minutes to the given phone number in the url.The url is given through http

Response.It have many advantages.

**Future Scope**

* In this if we add few parameters we can use it to pollution control.
* It can be included in health monitoring system.

**Bibilography**

<https://github.com/gnaneshwarbandari/IOT/blob/main/ibm_code.py>

<https://thesmartbridge.com/documents/projects/SmartHomeAutomationusingIBMCloud.pdf>

https://www.hindawi.com/journals/wcmc/2021/5592454/

Appendix

a.Source code

code for generating random parameters and send to ibm input node

import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

import json

#Provide your IBM Watson Device Credentials

organization = "8fy5zu"

deviceType = "iotdevice"

deviceId = "1001"

authMethod = "token"

authToken = "1234567890"

print("dataaaa")

# Initialize the device client.

T=0

H=0

def myCommandCallback(cmd):

print("Command received: %s" % cmd.data['command'])

if cmd.data['command']=='lighton':

print("LIGHT ON IS RECEIVED")

elif cmd.data['command']=='lightoff':

print("LIGHT OFF IS RECEIVED")

if cmd.command == "setInterval":

if 'interval' not in cmd.data:

print("Error - command is missing required information: 'interval'")

else:

interval = cmd.data['interval']

elif cmd.command == "print":

if 'message' not in cmd.data:

print("Error - command is missing required information: 'message'")

else:

print(cmd.data['message'])

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:

print("data")

T=random.randint(20,55)

H=random.randint(0,80)

U=random.randint(0,50)

S=random.randint(0,60)

C=random.randint(0,75)

D=random.randint(0,65)

#Send Temperature & Humidity to IBM Watson

data = {"d":{ 'hotair' : T, 'humidity': H,'smoke': U,'cloudwaves': S,'chemicals': C,'dust': D}}

print (data)

def myOnPublishCallback():

print ("Published Hotair = %s C" % t, "Humidity = %s %%" % h, "Smoke = %s %%" % u,"Cloudwaves = %s %%" % s,"Chemicals = %s %%" % c,"Dust = %s %%" % d, "to IBM Watson")

success = deviceCli.publishEvent("Data", "json", data, qos=0, on\_publish=myOnPublishCallback)

if not success:

print("Not connected to IoTF")

time.sleep(1)

deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud

deviceCli.disconnect()

b. UI output Screenshot

UI OUTPUT OF NODERED AND MIT APP UI

