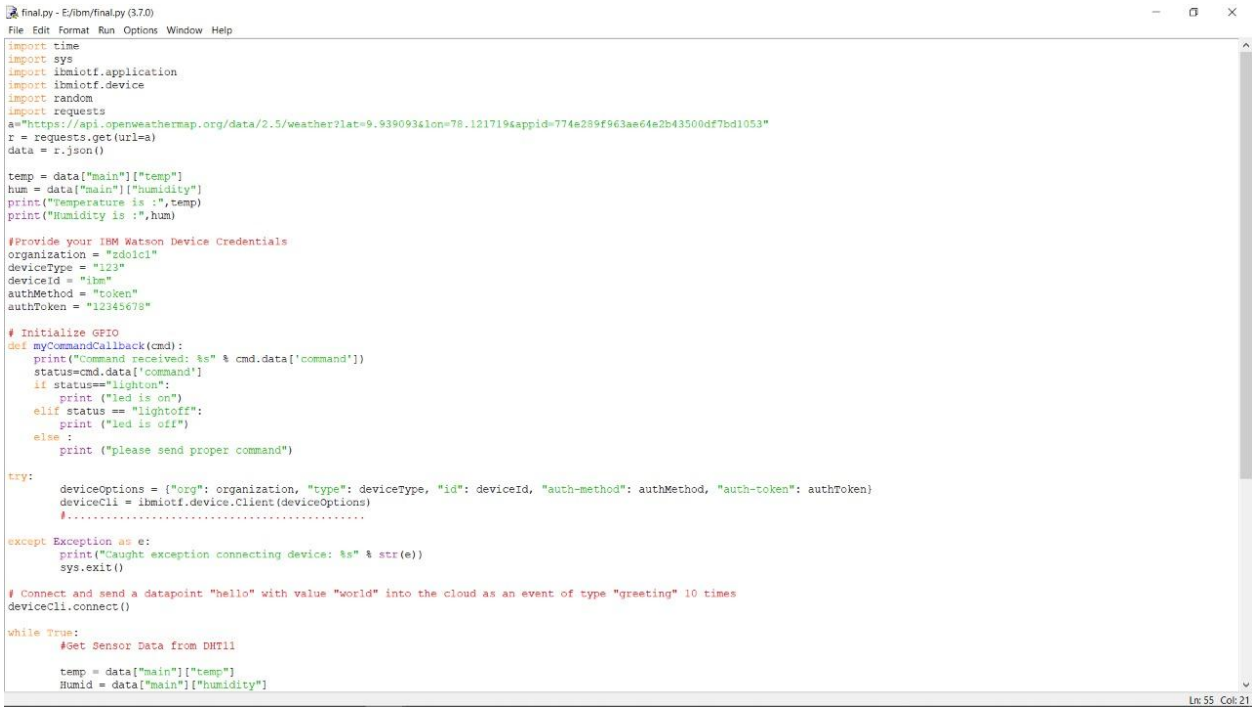
SPRINT 2

DATE	17 NOVEMBER 2022
TEAM ID	PNT2022TMID21273
TEAM NAME	SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

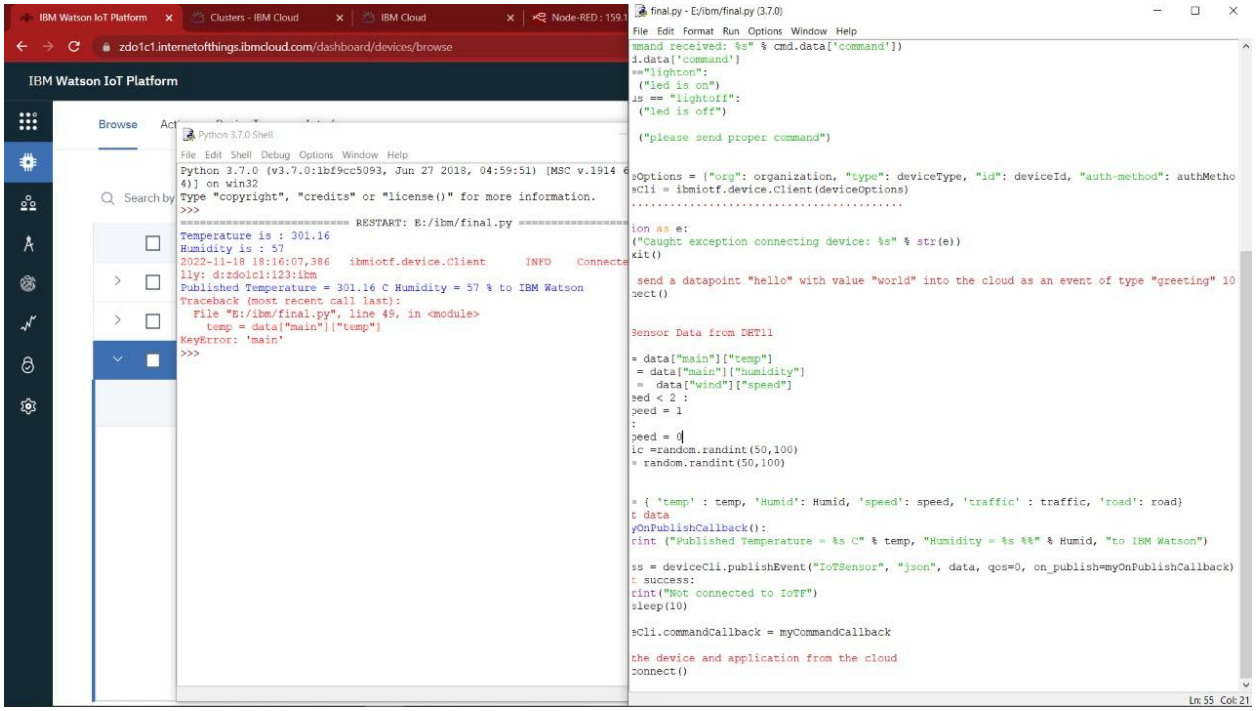
S.NO	**SPRINT**	**ACTIVITIES**
1	**SPRINT 2**	Creation of python script to extract important data like temperature , humidity and other weather conditions .
2	**SPRINT 2**	Creation of UI using MTI inventor app
3	**SPRINT 2**	Configure the connection security and create the API keys that are used in the NODE-RED service for accessing the IBM Iot
4	**SPRINT 2**	Create NODE-RED service

1.Creation of python script to extract important data like temperature , humidity and other weather conditions:



```
final.py - E:/ibm/final.py (3.7.0)
File  Edit  Format  Run  Options  Window  Help
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
import requests
a="https://api.openweathermap.org/data/2.5/weather?lat=9.939093&lon=78.121719&appid=774e289f963ae64e2b43500df7bd1053"
r = requests.get(url=a)
data = r.json()

temp = data["main"]["temp"]
hum = data["main"]["humidity"]
print("Temperature is :",temp)
print("Humidity is :",hum)

#Provide your IBM Watson Device Credentials
organization = "zdo1c1"
deviceType = "123"
deviceId = "ibm"
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")
    elif status == "lightoff":
        print ("led is off")
    else :
        print ("please send proper command")

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

        temp = data["main"]["temp"]
        Humid = data["main"]["humidity"]
```

Ln: 55 Col: 21



IBM Watson IoT Platform | Clusters - IBM Cloud | IBM Cloud | Node-RED : 159.1

zdo1c1.internetofthings.ibmcloud.com/dashboard/devices/browse

IBM Watson IoT Platform

Browse

Search by

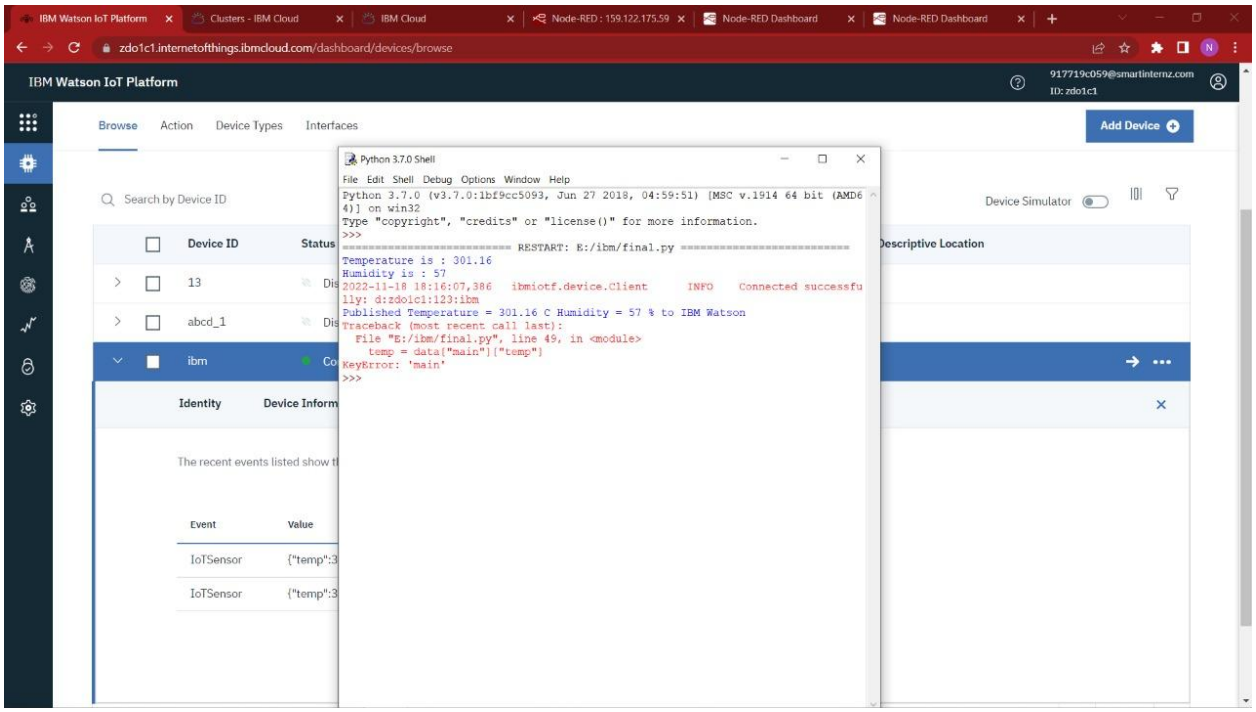
Python 3.7.0 Shell

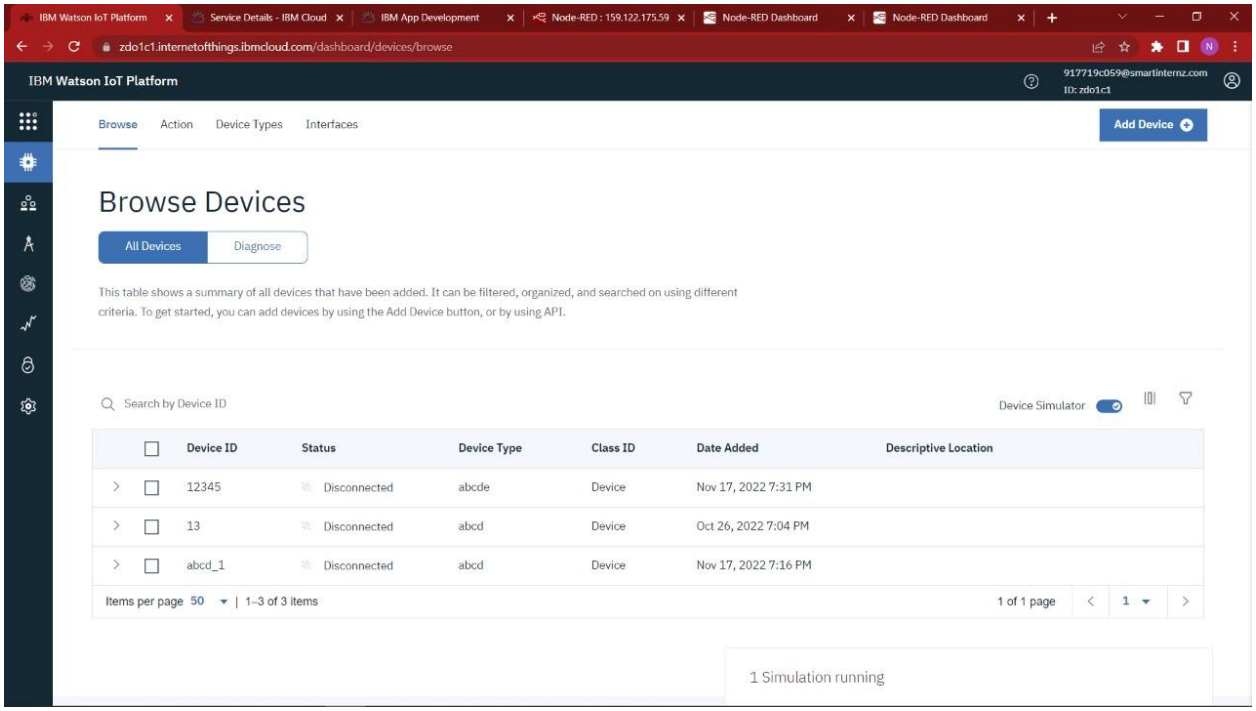
```
File  Edit  Shell  Debug  Options  Window  Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 6
4)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: E:/ibm/final.py ==========
Temperature is : 301.16
Humidity is : 57
2022-11-18 18:16:07,386   ibmiotf.device.Client      INFO    Connecte
lly: d:zdo1c1:123:ibm
Published Temperature = 301.16 C Humidity = 57 % to IBM Watson
Traceback (most recent call last):
  File "E:/ibm/final.py", line 49, in <module>
    temp = data["main"]["temp"]
KeyError: 'main'
>>>
```

final.py - E:/ibm/final.py (3.7.0)

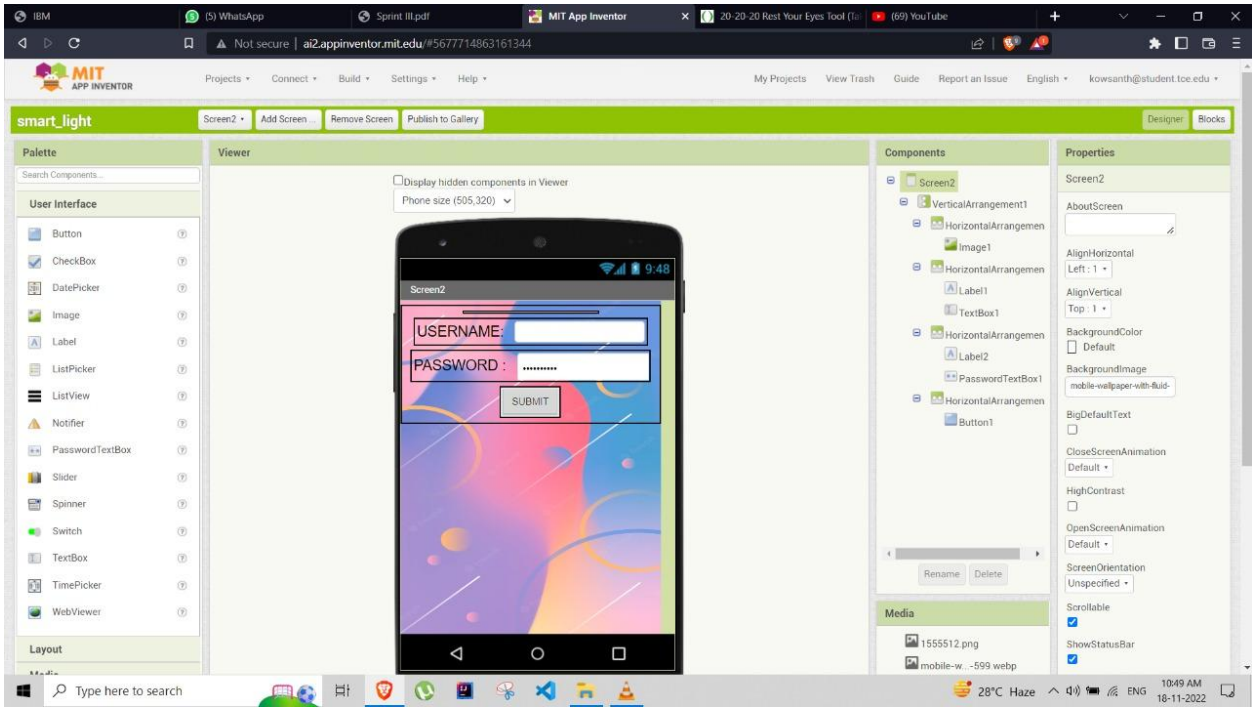
```
File  Edit  Format  Run  Options  Window  Help
mmand received: %s" % cmd.data['command'])
d.data['command']
=="lighton":
 ("led is on")
is == "lightoff":
 ("led is off")

 ("please send proper command")


sOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMetho
sCli = ibmiotf.device.Client(deviceOptions)
.....................................

ion as e:
("Caught exception connecting device: %s" % str(e))
xit()

 send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10
nect()


Sensor Data from DHT11

= data["main"]["temp"]
 = data["main"]["humidity"]
 =  data["wind"]["speed"]
eed < 2 :
peed = 1
:
peed = 0
ic =random.randint(50,100)
= random.randint(50,100)


= { 'temp' : temp, 'Humid': Humid, 'speed': speed, 'traffic' : traffic, 'road': road}
t data
yOnPublishCallback():
rint ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "to IBM Watson")

ss = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
t success:
rint("Not connected to IoTF")
sleep(10)

sCli.commandCallback = myCommandCallback

the device and application from the cloud
onnect()
```

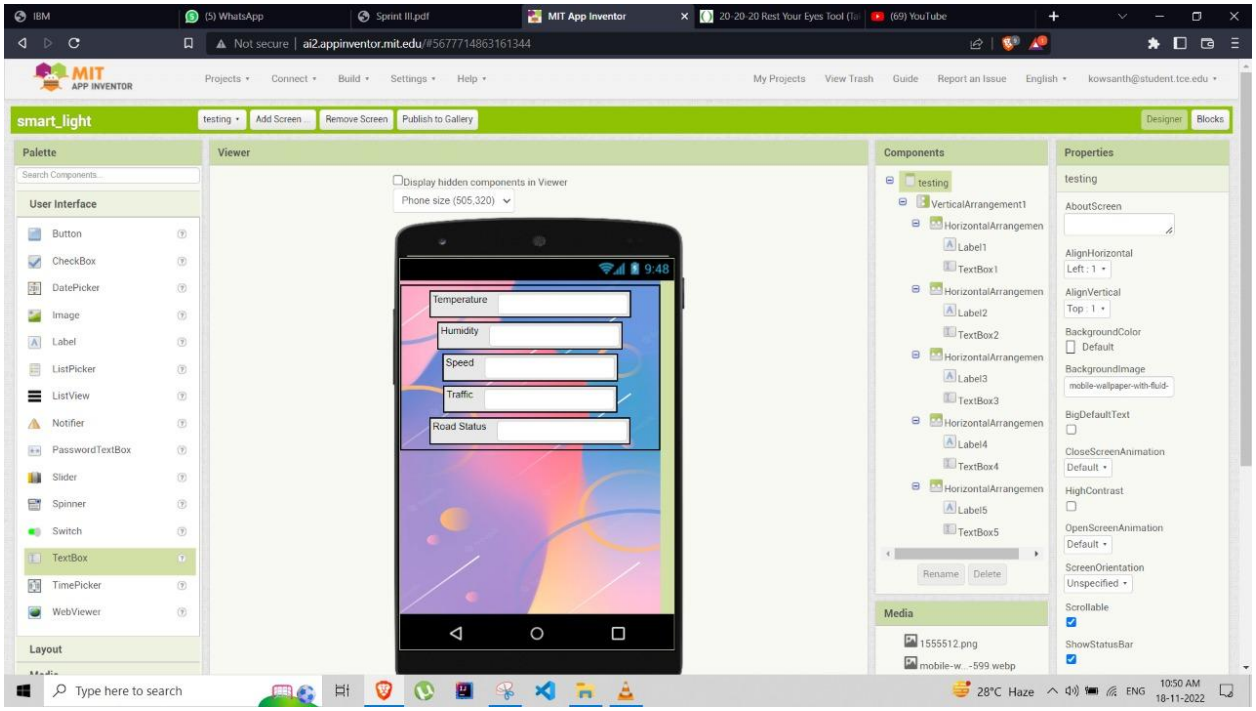
Ln: 55 Col: 21

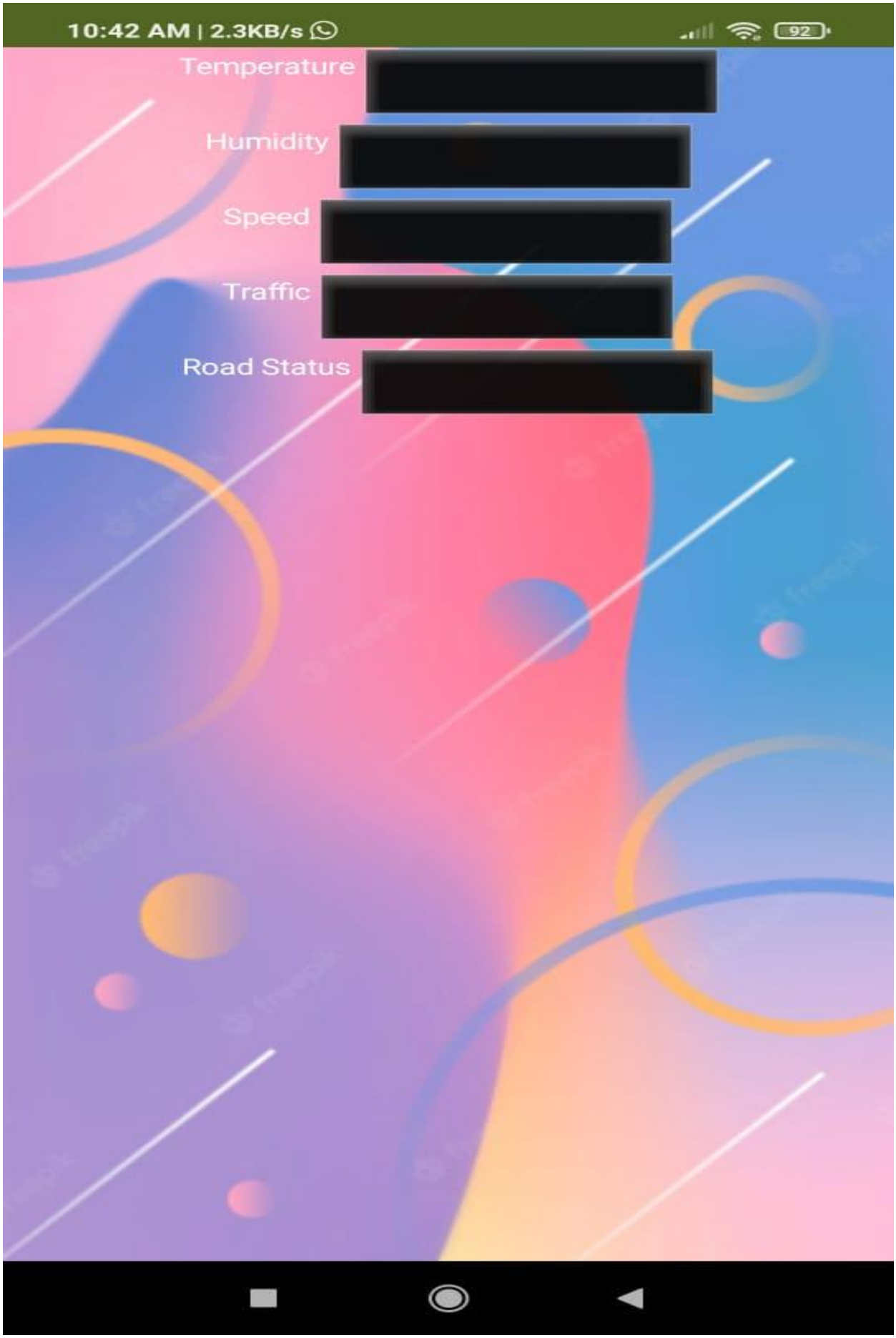

IBM Watson IoT Platform
zdo1c1.internetofthings.ibmcloud.com/dashboard/devices/browse
Browse
Action
Device Types
Interfaces
Add Device
Search by Device ID
Device Simulator
Device ID
Status
Descriptive Location
13
abcd_1
ibm
Identity
Device Inform
The recent events listed show t
Event
Value
IoTSensor
IoTSensor
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: E:/ibm/final.py ==========
Temperature is : 301.16
Humidity is : 57
2022-11-18 18:16:07,386 ibmiotf.device.Client INFO Connected successfully: d:zdo1c1:123:ibm
Published Temperature = 301.16 C Humidity = 57 % to IBM Watson
Traceback (most recent call last):
File "E:/ibm/final.py", line 49, in <module>
temp = data["main"]["temp"]
KeyError: 'main'
>>>


IBM Watson IoT Platform
zdo1c1.internetofthings.ibmcloud.com/dashboard/devices/browse
Browse
Action
Device Types
Interfaces
Add Device
Browse Devices
All Devices
Diagnose
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.
Search by Device ID
Device Simulator
Device ID
Status
Device Type
Class ID
Date Added
Descriptive Location
12345 Disconnected abcde Device Nov 17, 2022 7:31 PM
13 Disconnected abcd Device Oct 26, 2022 7:04 PM
abcd_1 Disconnected abcd Device Nov 17, 2022 7:16 PM
Items per page 50 | 1–3 of 3 items
1 of 1 page
1 Simulation running

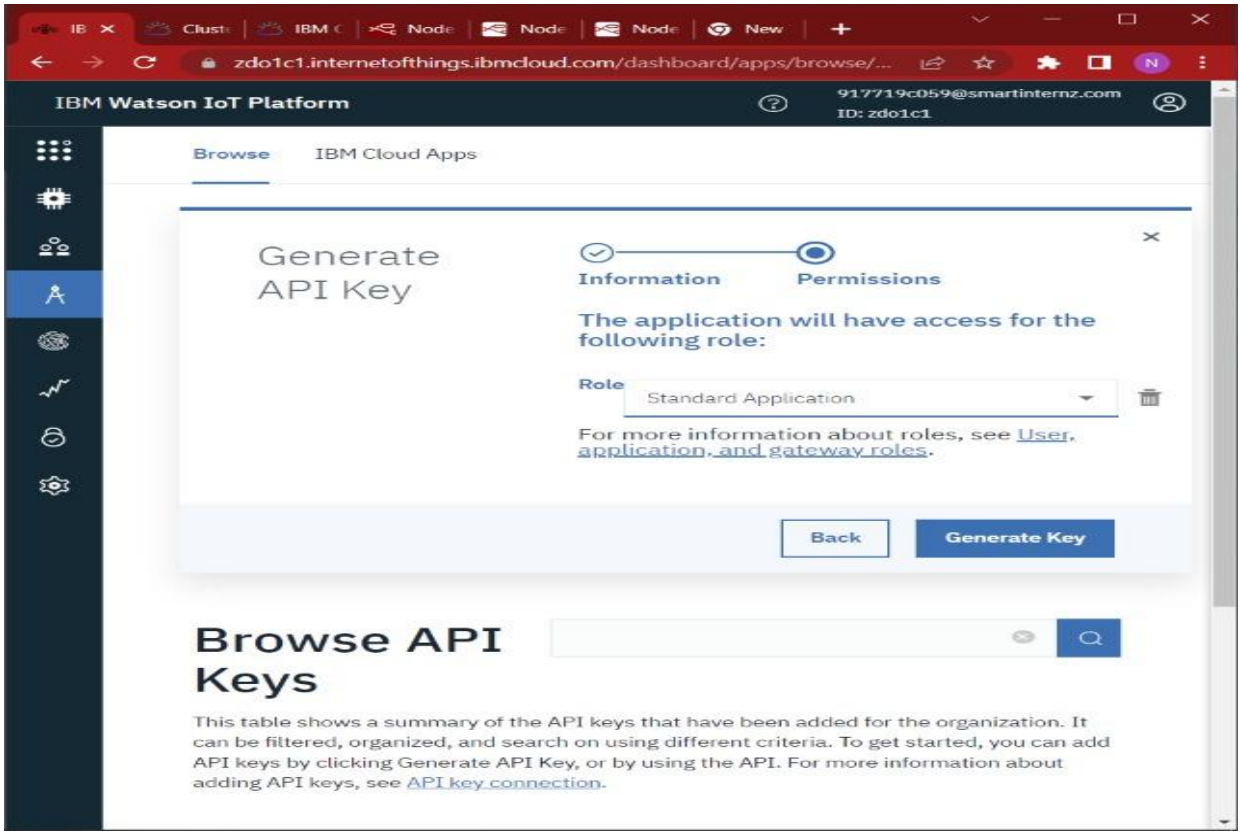
2.Creation of UI using MTI inventor app:

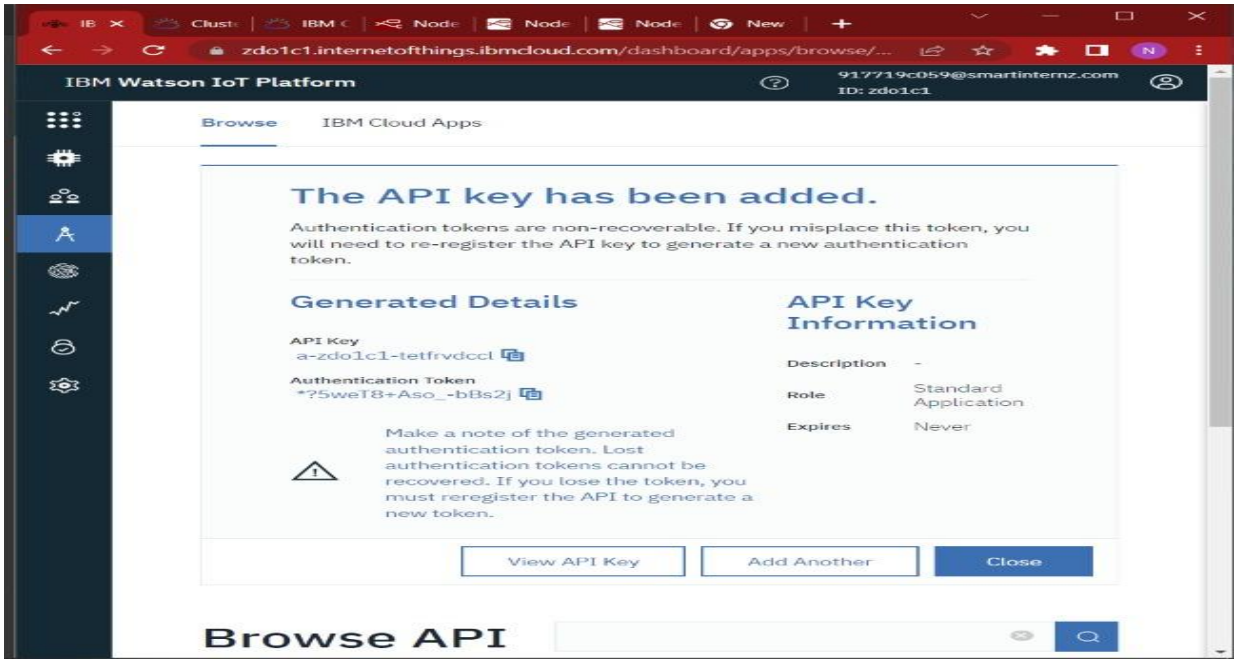




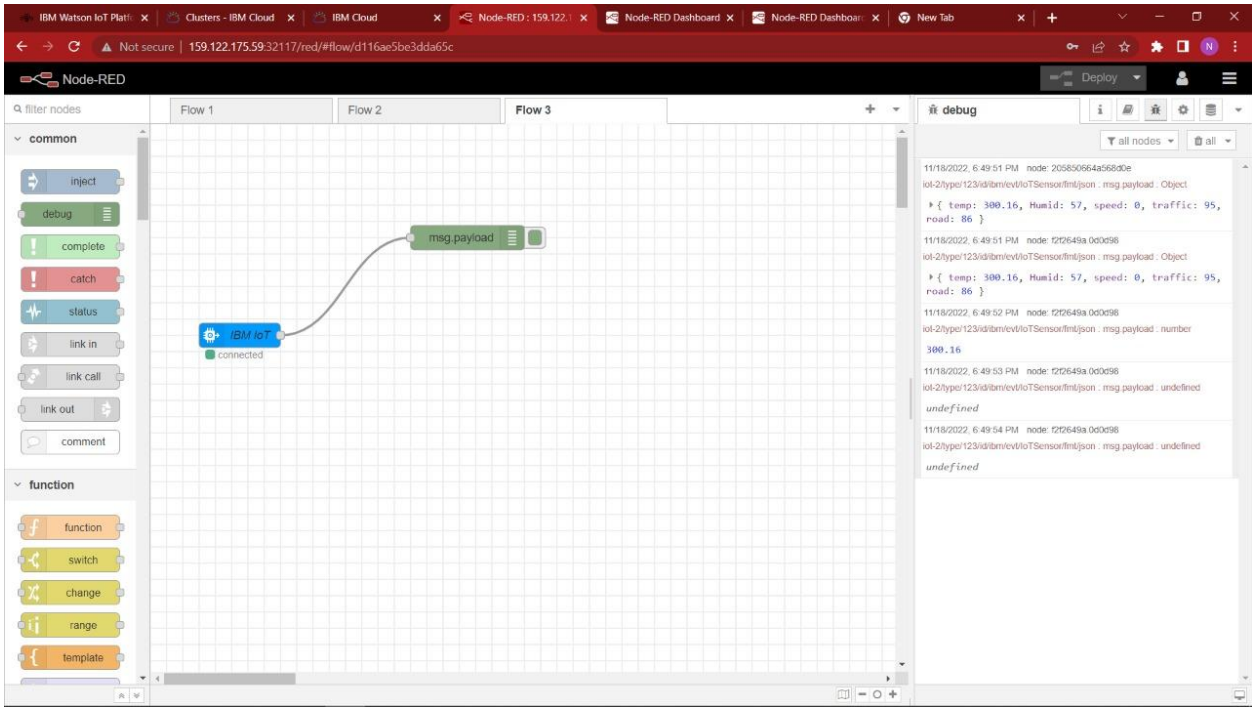

10:42 AM | 2.3KB/s
92
Temperature
Humidity
Speed
Traffic
Road Status

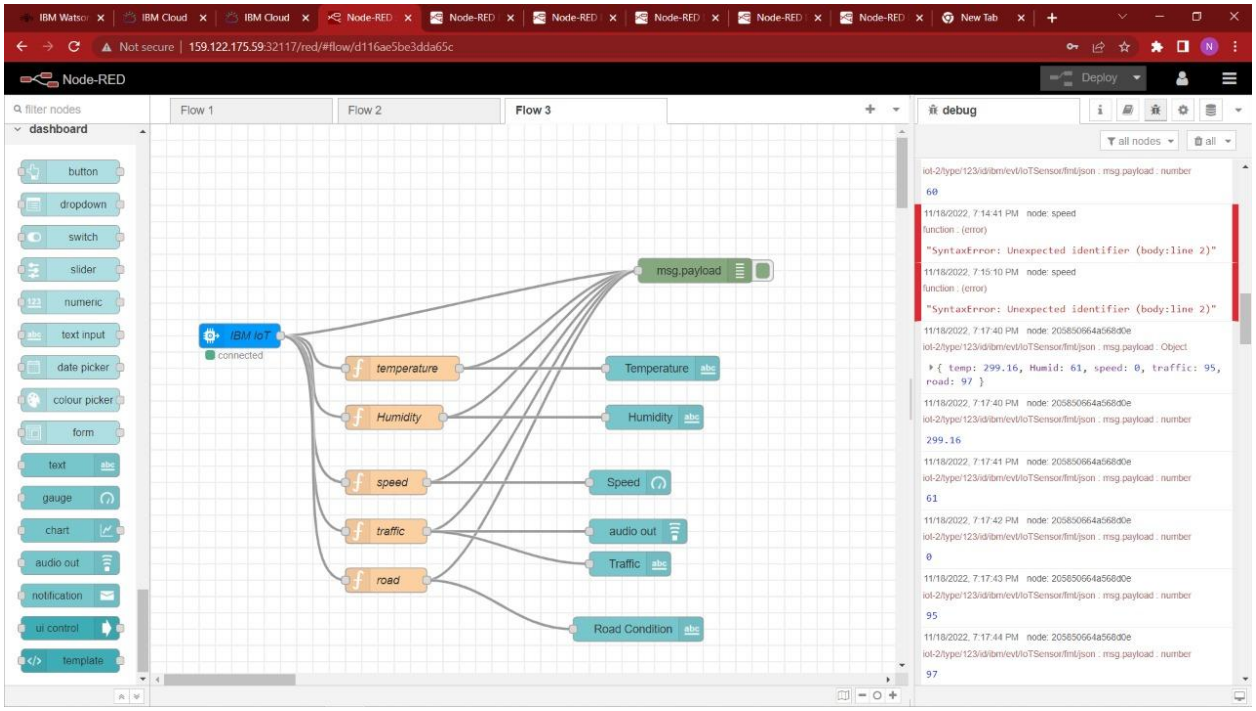
3.Configure the connection security and create the API keys that are used in the NODE-RED service for accessing the IBM Iot

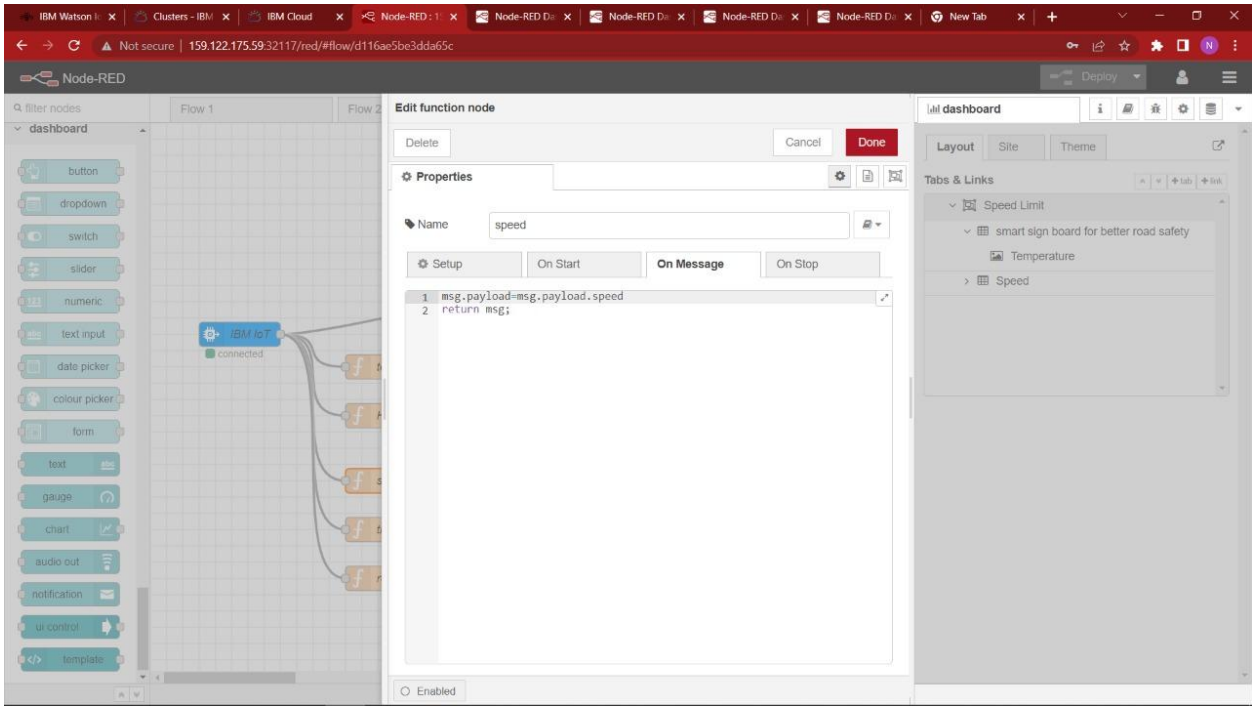


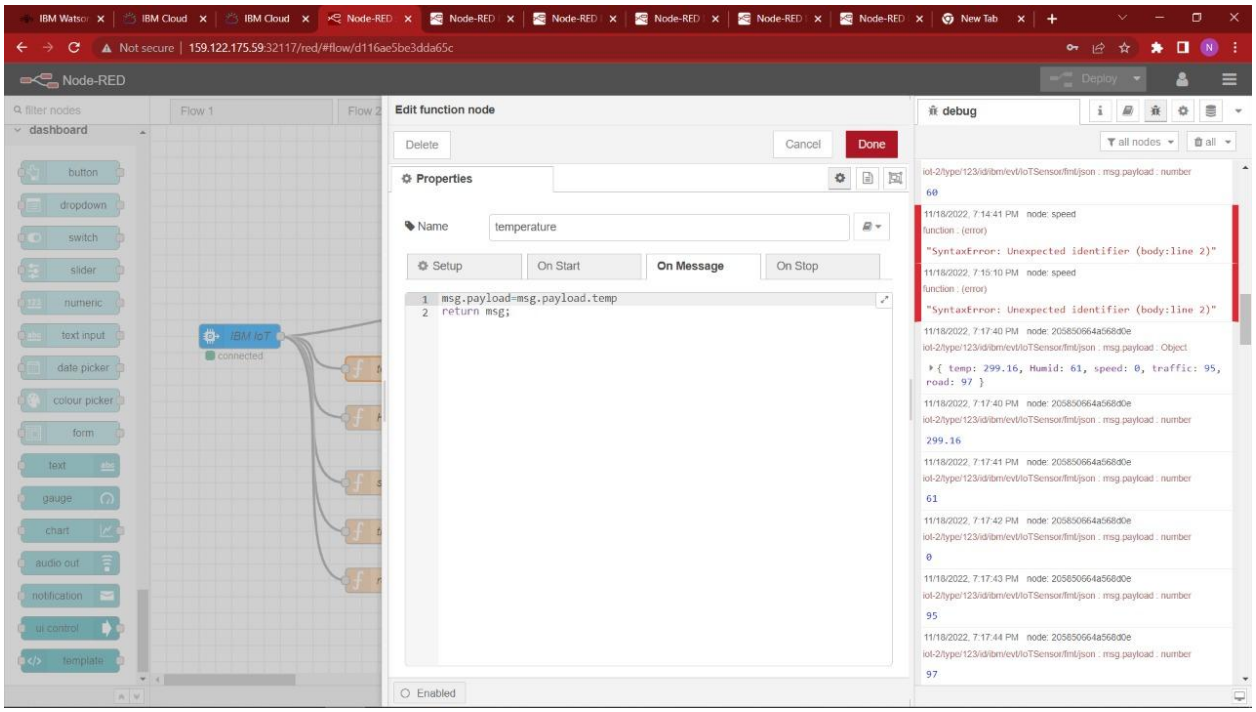


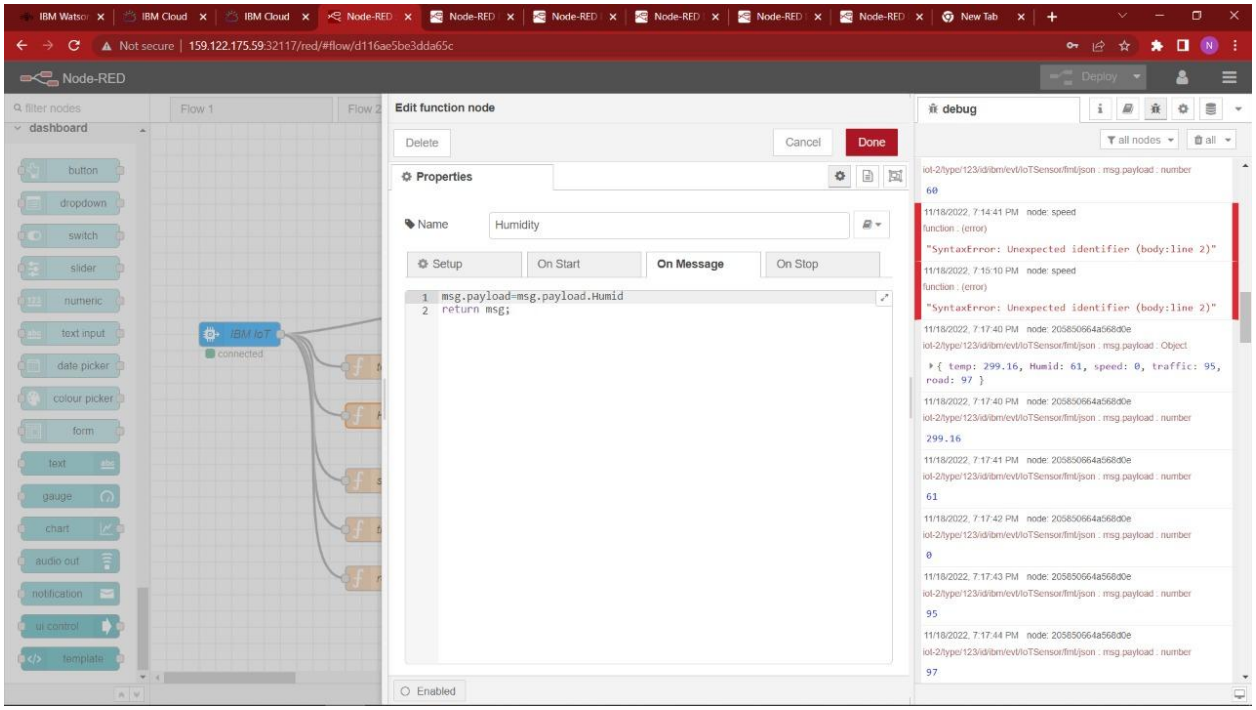
4.Create NODE-RED service:

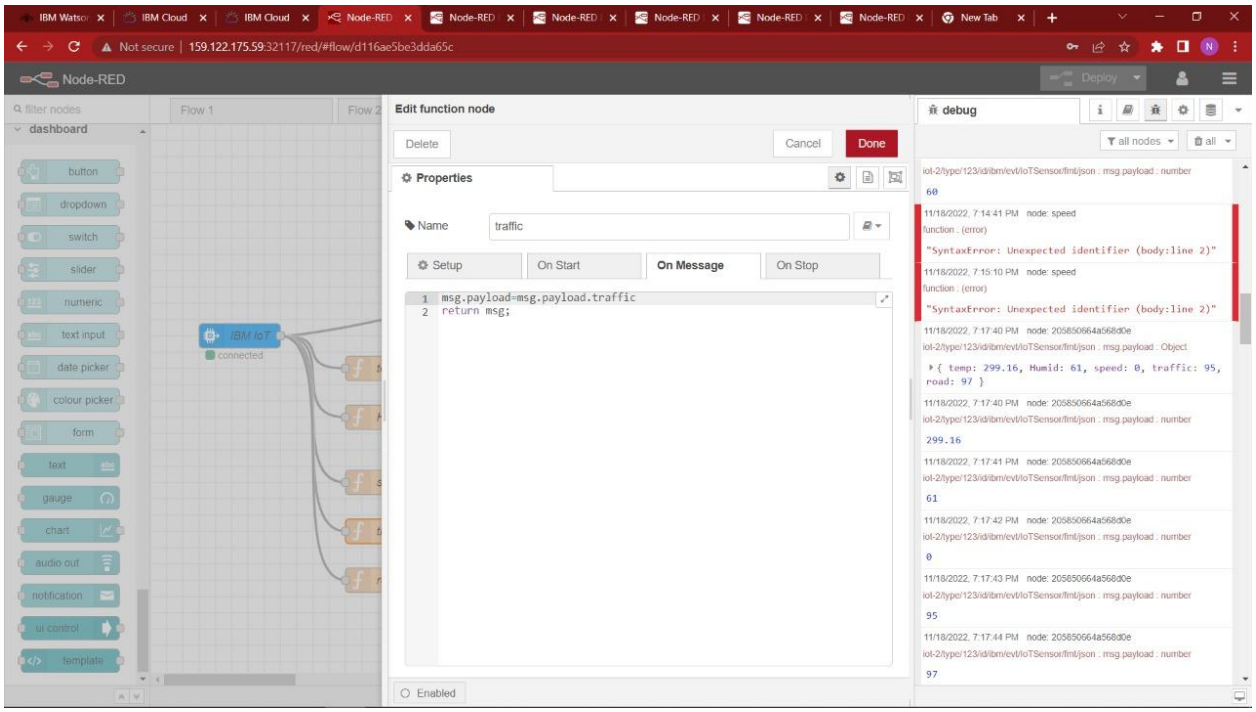


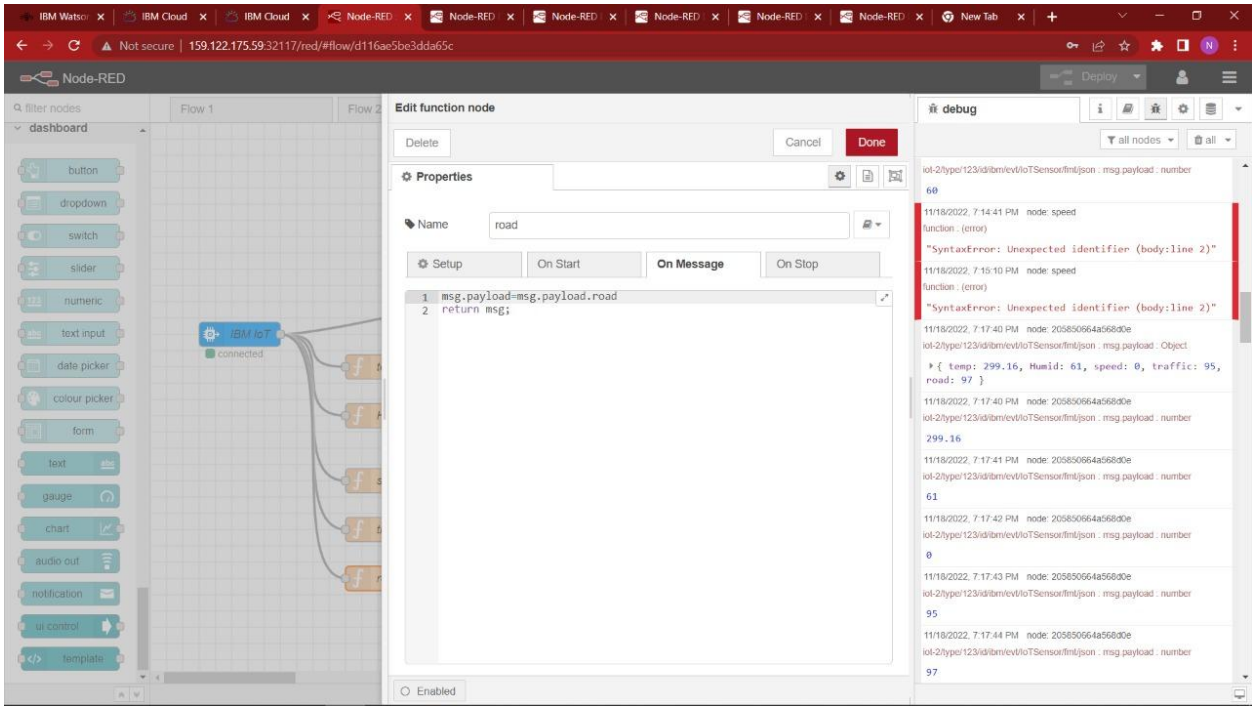


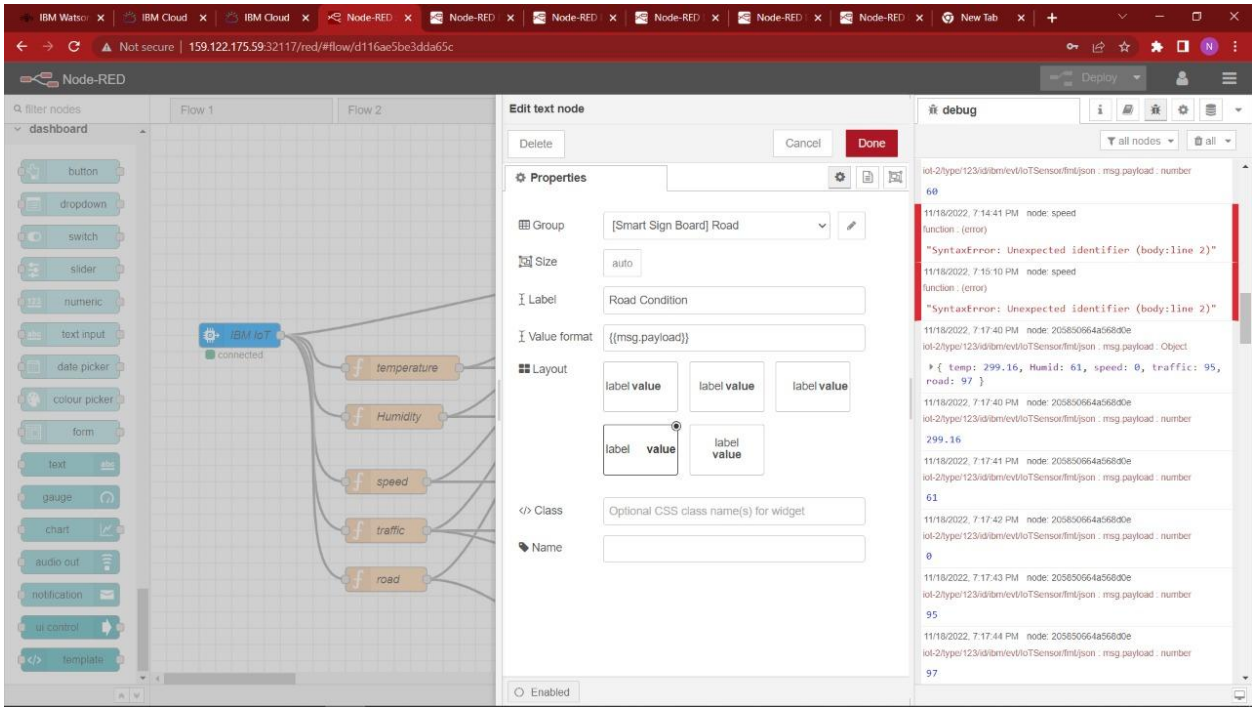

Node-RED
Edit function node
Delete
Cancel
Done
Properties
Name
speed
Setup
On Start
On Message
On Stop
msg.payload=msg.payload.speed
return msg;
Enabled
dashboard
Layout
Site
Theme
Tabs & Links
Speed Limit
smart sign board for better road safety
Temperature
Speed


Node-RED
Edit function node
Delete
Cancel
Done
Properties
Name
temperature
Setup
On Start
On Message
On Stop
msg.payload=msg.payload.temp
return msg;
Enabled
debug
60
"SyntaxError: Unexpected identifier (body:line 2)"
"SyntaxError: Unexpected identifier (body:line 2)"
{ temp: 299.16, Humid: 61, speed: 0, traffic: 95, road: 97 }
299.16
61
0
95
97


Not secure | 159.122.175.59:32117/red/#flow/d116ae5be3dda65c
Node-RED
Deploy
Edit function node
Delete
Cancel
Done
Properties
Name
Humidity
Setup
On Start
On Message
On Stop
1 msg.payload=msg.payload.Humid
2 return msg;
Enabled
debug
all nodes
"SyntaxError: Unexpected identifier (body:line 2)"
{ temp: 299.16, Humid: 61, speed: 0, traffic: 95, road: 97 }
299.16
61
0
95
97


Not secure | 159.122.175.59:32117/red/#flow/d116ae5be3dda65c
Node-RED
Deploy
Edit function node
Delete
Cancel
Done
Properties
Name
traffic
Setup
On Start
On Message
On Stop
1 msg.payload=msg.payload.traffic
2 return msg;
Enabled
debug
all nodes
"SyntaxError: Unexpected identifier (body:line 2)"
{ temp: 299.16, Humid: 61, speed: 0, traffic: 95, road: 97 }
299.16
61
0
95
97


Node-RED
Edit function node
Delete
Cancel
Done
Properties
Name
road
Setup
On Start
On Message
On Stop
msg.payload=msg.payload.road
return msg;
Enabled
debug
IBM IoT
connected
dashboard
button
dropdown
switch
slider
numeric
text input
date picker
colour picker
form
text
gauge
chart
audio out
notification
ui control
template


Node-RED
Edit text node
Delete
Cancel
Done
Properties
Group
[Smart Sign Board] Road
Size
auto
Label
Road Condition
Value format
{{msg.payload}}
Layout
label value
Class
Optional CSS class name(s) for widget
Name
Enabled
debug
IBM IoT
connected
temperature
Humidity
speed
traffic
road