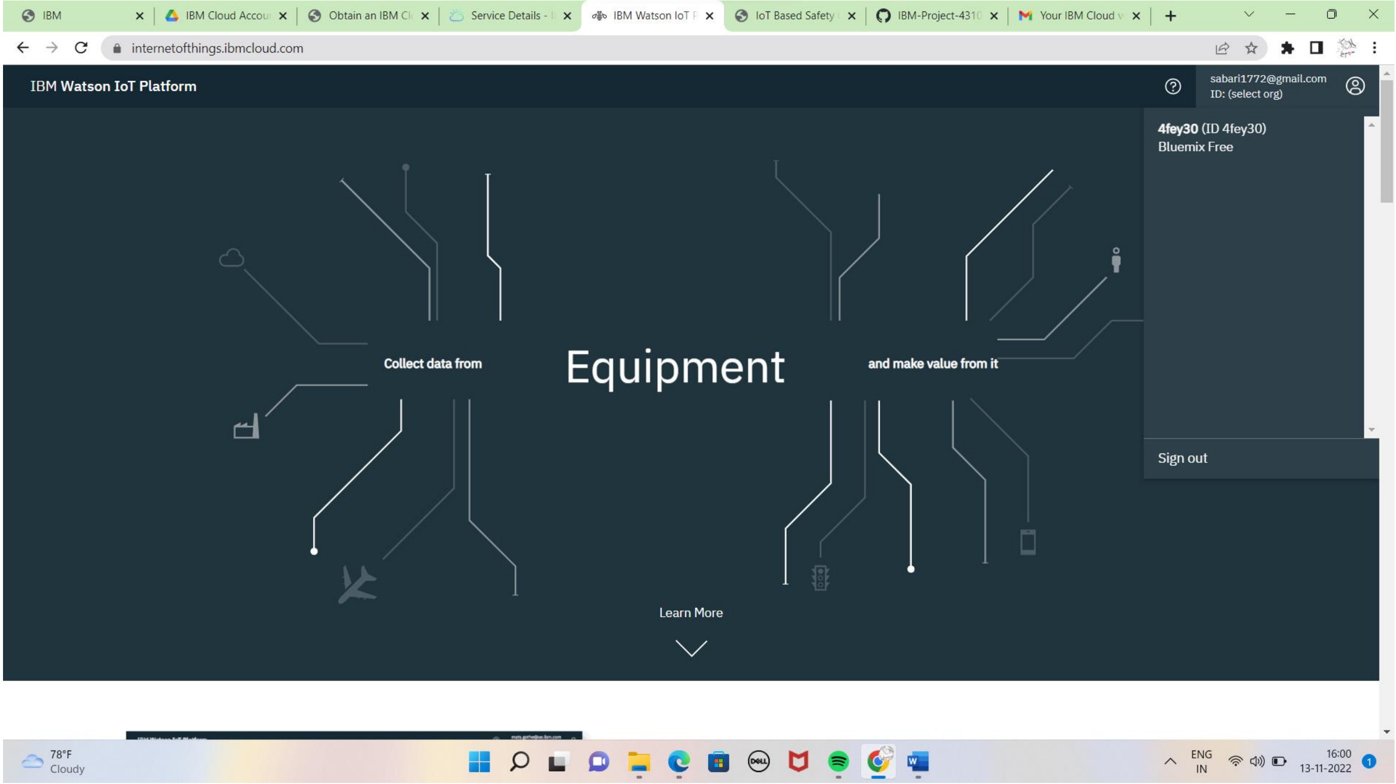
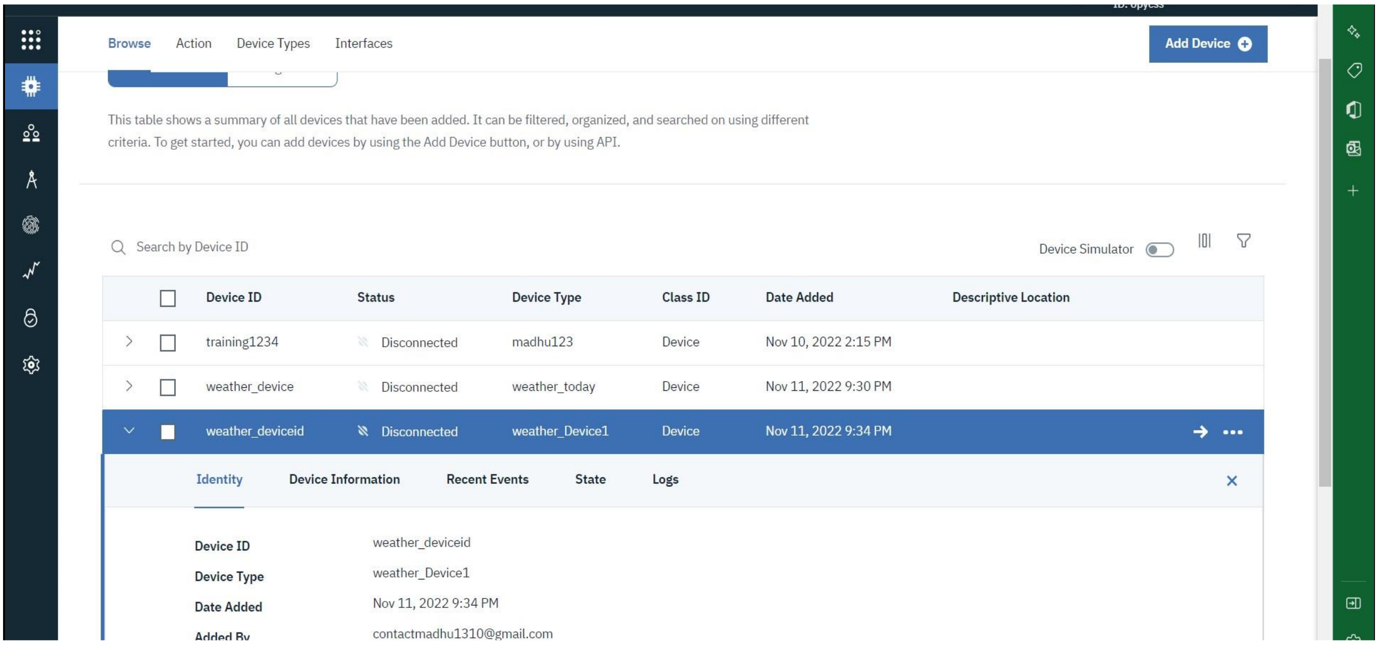
DELIVERY OF SPRINT 2

Team ID	**PNT2002TMID10533**
Project Name **Project**	***IOT Based Safety Gadget for Child Safety Monitoring&Notification***

Creating IBM Cloud Service and IBM WATSON IoT PLATFORM:





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	Descriptive Location
□	training1234	Disconnected	madhu123	Device	Nov 10, 2022 2:15 PM	
□	weather_device	Disconnected	weather_today	Device	Nov 11, 2022 9:30 PM	
□	weather_deviceid	Disconnected	weather_Device1	Device	Nov 11, 2022 9:34 PM	

Device ID	weather_deviceid
Device Type	weather_Device1
Date Added	Nov 11, 2022 9:34 PM
Added By	contactmadhu1310@gmail.com

Creating and Connecting IBM cloud for Project and Python Code

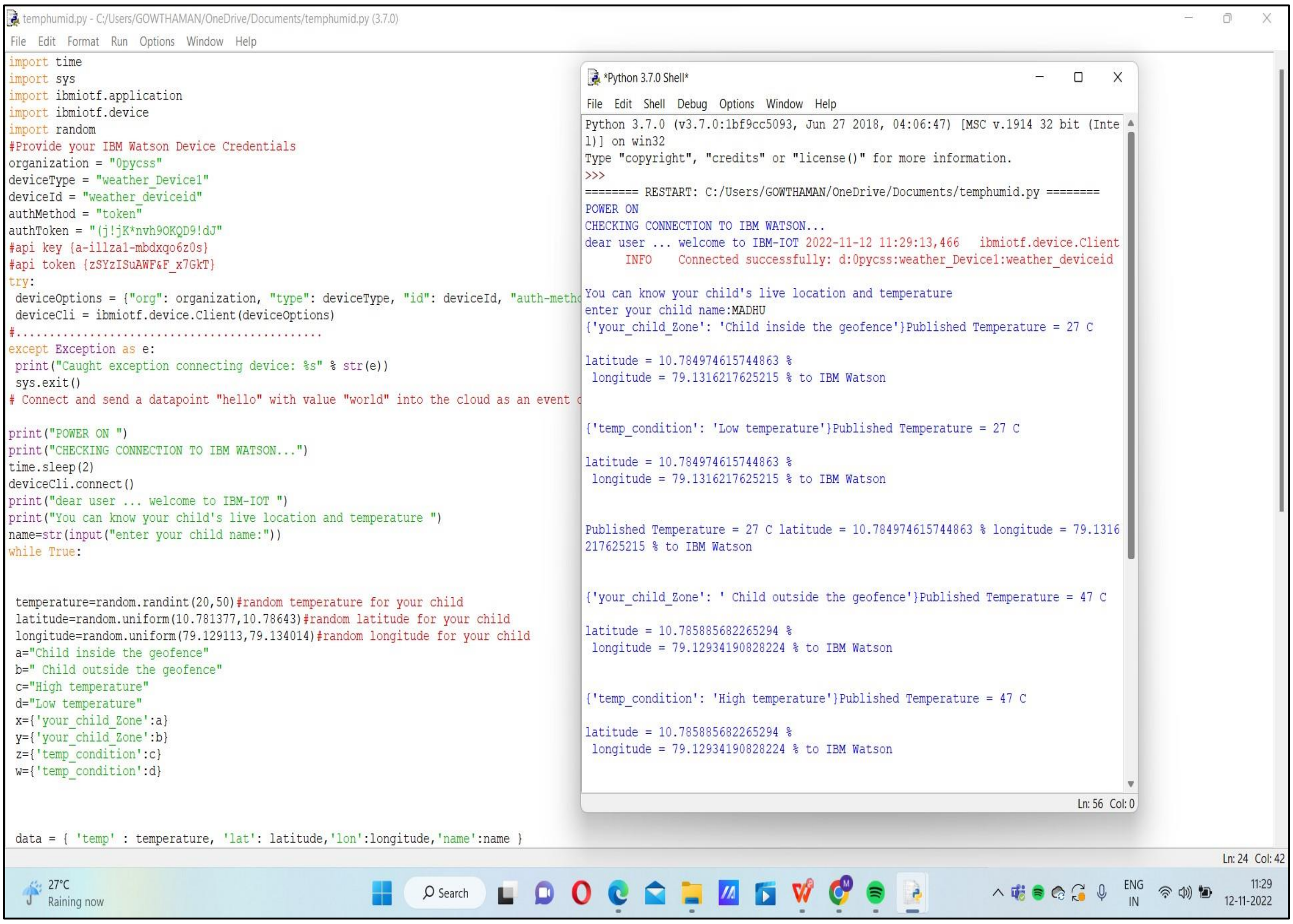
```
import time import sys

import ibmiotf.application

import ibmiotf.device

import random

#Provide your IBM Watson Device Credentials

organization = "0pycss" deviceType =

"weather_Device1" deviceId =

"weather_deviceid" authMethod = "token"


authToken = "(j!jK*nvh9OKQD9!dJ"

#api key {a-illza1-mbdxqo6z0s} #api

token {zSYzISuAWF&F_x7GkT}
```

```
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


print("POWER ON ") print("CHECKING CONNECTION

TO IBM WATSON...")

time.sleep(2) deviceCli.connect() print("dear user ... welcome to

IBM-IOT ") print("You can know your child's live location and

temperature ") name=str(input("enter your child name:")) while

True:



 temperature=random.randint(20,50)#random temperature for your child

latitude=random.uniform(10.781377,10.78643)#random latitude for your child

longitude=random.uniform(79.129113,79.134014)#random longitude for your child

a="Child inside the geofence"  b=" Child outside the geofence"  c="High temperature"

d="Low temperature"  x={'your_child_Zone':a}  y={'your_child_Zone':b}

z={'temp_condition':c}  w={'temp_condition':d}
```

```
 data = { 'temp' : temperature, 'lat': latitude,'lon':longitude,'name':name }

 #print data  def

myOnPublishCallback():

   print ("Published Temperature = %s C" % temperature, "latitude = %s %%" % latitude,
"longitude = %s %%" % longitude, "to IBM Watson")

   print("\n")

 success = deviceCli.publishEvent("IoTSensorgpsdata", "json", data, qos=0,
on_publish=myOnPublishCallback)

 if latitude>=10.78200 and latitude<=10.786000 and longitude >=79.130000 and
longitude <=79.133000:
deviceCli.publishEvent("IoTSensorgpsdata","json",data=x,qos=0,on_publish=myOnPublis
hCallback)

   print(x)

print("\n")  else:




deviceCli.publishEvent("IoTSensorgpsdata","json",data=y,qos=0,on_publish=myOnPublis
hCallback)

  print(y)

print("\n")



 if (temperature>35):
```

```
deviceCli.publishEvent("IoTSensorgpsdata","json",data=z,qos=0,on_publish=myOnPublishCallback)

     print(z)

print("\n")  else:




deviceCli.publishEvent("IoTSensorgpsdata","json",data=w,qos=0,on_publish=myOnPublishCallback)

    print(w)

print("\n")  if not

success:

print("Not connected

to IoTF")

print("\n")

time.sleep(3)

# Disconnect the device and application from the cloud deviceCli.disconnect()
```

Connecting IBM Watson and python Code:

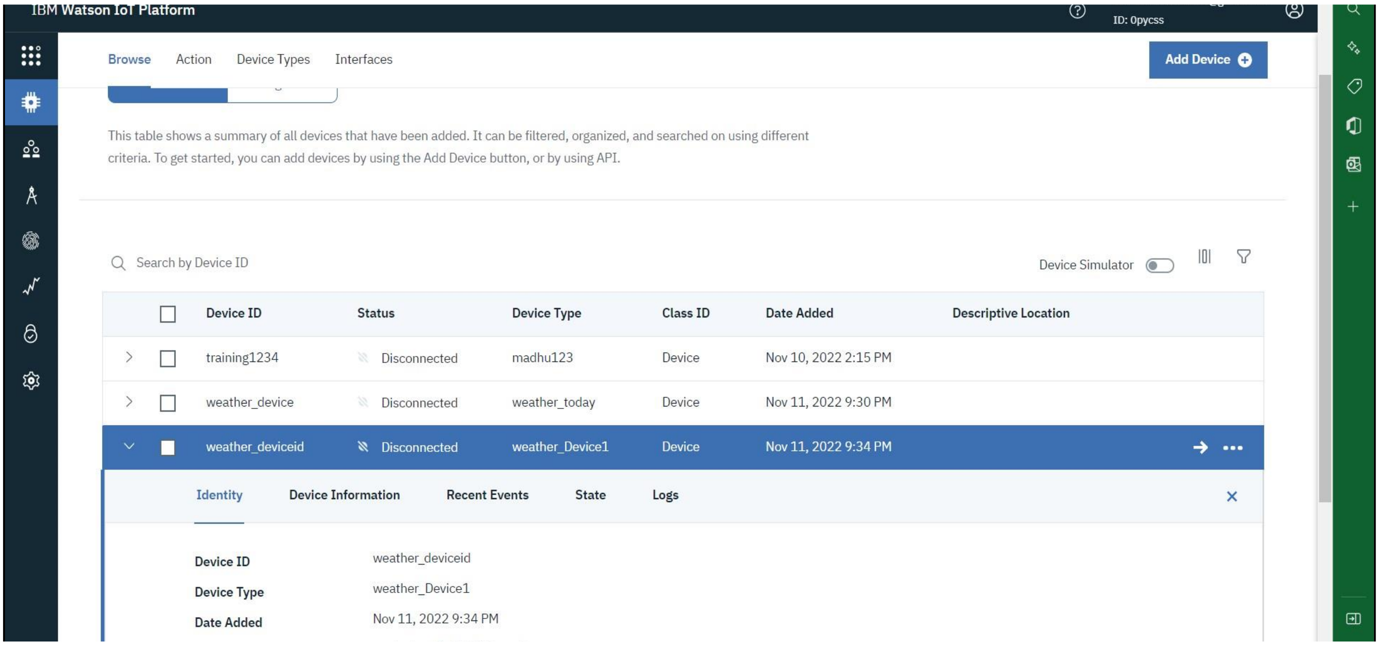


temphumid.py - C:/Users/GOWTHAMAN/OneDrive/Documents/temphumid.py (3.7.0)

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "0pycss"
deviceType = "weather_Device1"
deviceId = "weather_deviceid"
authMethod = "token"
authToken = "(j!jK*nvh9OKQD9!dJ"
#api key {a-illza1-mbdxqo6z0s}
#api token {zSYzISuAWF&F_x7GkT}
try:
 deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-metho
 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 c

print("POWER ON ")
print("CHECKING CONNECTION TO IBM WATSON...")
time.sleep(2)
deviceCli.connect()
print("dear user ... welcome to IBM-IOT ")
print("You can know your child's live location and temperature ")
name=str(input("enter your child name:"))
while True:


 temperature=random.randint(20,50)#random temperature for your child
 latitude=random.uniform(10.781377,10.78643)#random latitude for your child
 longitude=random.uniform(79.129113,79.134014)#random longitude for your child
 a="Child inside the geofence"
 b=" Child outside the geofence"
 c="High temperature"
 d="Low temperature"
 x={'your_child_Zone':a}
 y={'your_child_Zone':b}
 z={'temp_condition':c}
 w={'temp_condition':d}



 data = { 'temp' : temperature, 'lat': latitude,'lon':longitude,'name':name }
```

Python 3.7.0 Shell

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Inte
l)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:/Users/GOWTHAMAN/OneDrive/Documents/temphumid.py ========
POWER ON
CHECKING CONNECTION TO IBM WATSON...
dear user ... welcome to IBM-IOT 2022-11-12 11:29:13,466   ibmiotf.device.Client
      INFO    Connected successfully: d:0pycss:weather_Device1:weather_deviceid

You can know your child's live location and temperature
enter your child name:MADHU
{'your_child_Zone': 'Child inside the geofence'}Published Temperature = 27 C

latitude = 10.784974615744863 %
 longitude = 79.1316217625215 % to IBM Watson


{'temp_condition': 'Low temperature'}Published Temperature = 27 C

latitude = 10.784974615744863 %
 longitude = 79.1316217625215 % to IBM Watson


Published Temperature = 27 C latitude = 10.784974615744863 % longitude = 79.1316
217625215 % to IBM Watson


{'your_child_Zone': ' Child outside the geofence'}Published Temperature = 47 C

latitude = 10.785885682265294 %
 longitude = 79.12934190828224 % to IBM Watson


{'temp_condition': 'High temperature'}Published Temperature = 47 C

latitude = 10.785885682265294 %
 longitude = 79.12934190828224 % to IBM Watson
```



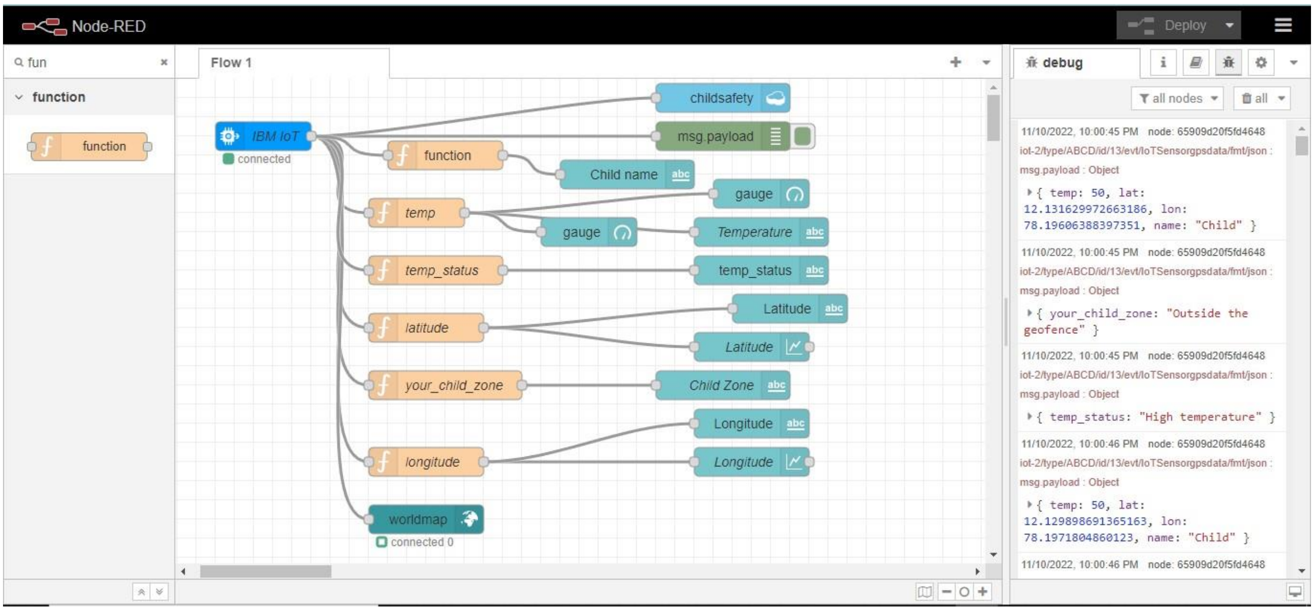
IBM Watson IoT Platform

ID: 0pycss

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.

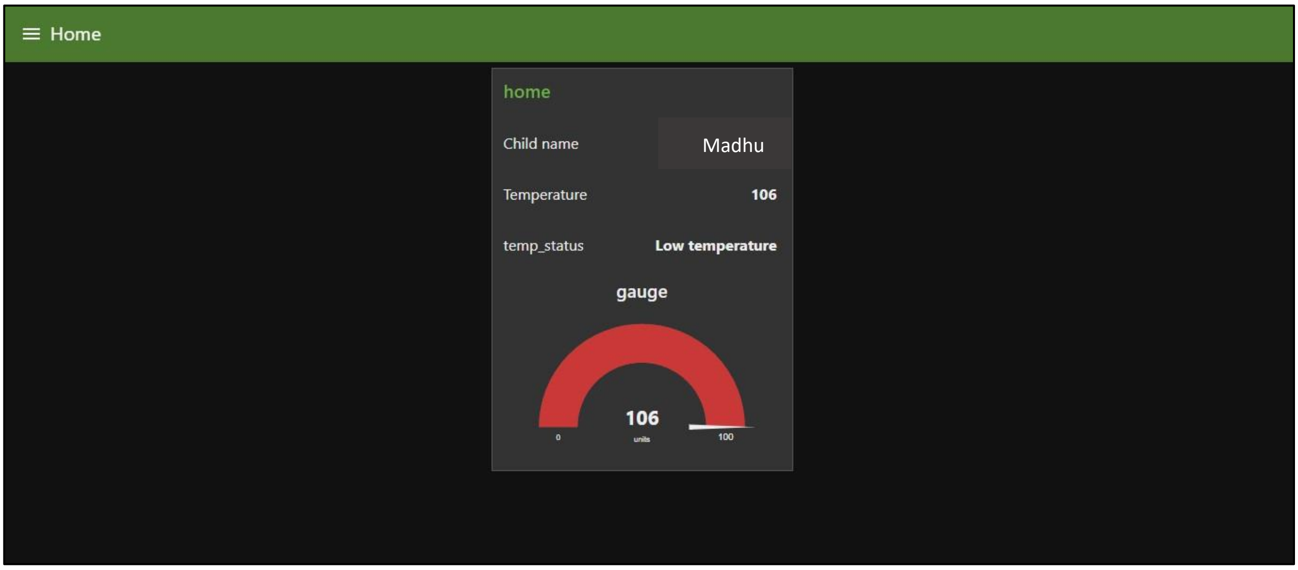
Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
training1234	Disconnected	madhu123	Device	Nov 10, 2022 2:15 PM	
weather_device	Disconnected	weather_today	Device	Nov 11, 2022 9:30 PM	
weather_deviceid	Disconnected	weather_Device1	Device	Nov 11, 2022 9:34 PM	

Device ID	weather_deviceid
Device Type	weather_Device1
Date Added	Nov 11, 2022 9:34 PM



```
11/10/2022, 10:00:45 PM   node: 65909d20f5fd4648
iot-2/type/ABCD/id/13/evt/IoTSensorgpsdata/fmt/json :
msg.payload : Object
▸{ temp: 50, lat:
12.131629972663186, lon:
78.19606388397351, name: "Child" }
11/10/2022, 10:00:45 PM   node: 65909d20f5fd4648
iot-2/type/ABCD/id/13/evt/IoTSensorgpsdata/fmt/json :
msg.payload : Object
▸{ your_child_zone: "Outside the
geofence" }
11/10/2022, 10:00:45 PM   node: 65909d20f5fd4648
iot-2/type/ABCD/id/13/evt/IoTSensorgpsdata/fmt/json :
msg.payload : Object
▸{ temp_status: "High temperature" }
11/10/2022, 10:00:46 PM   node: 65909d20f5fd4648
iot-2/type/ABCD/id/13/evt/IoTSensorgpsdata/fmt/json :
msg.payload : Object
▸{ temp: 50, lat:
12.129898691365163, lon:
78.1971804860123, name: "Child" }
11/10/2022, 10:00:46 PM   node: 65909d20f5fd4648
```

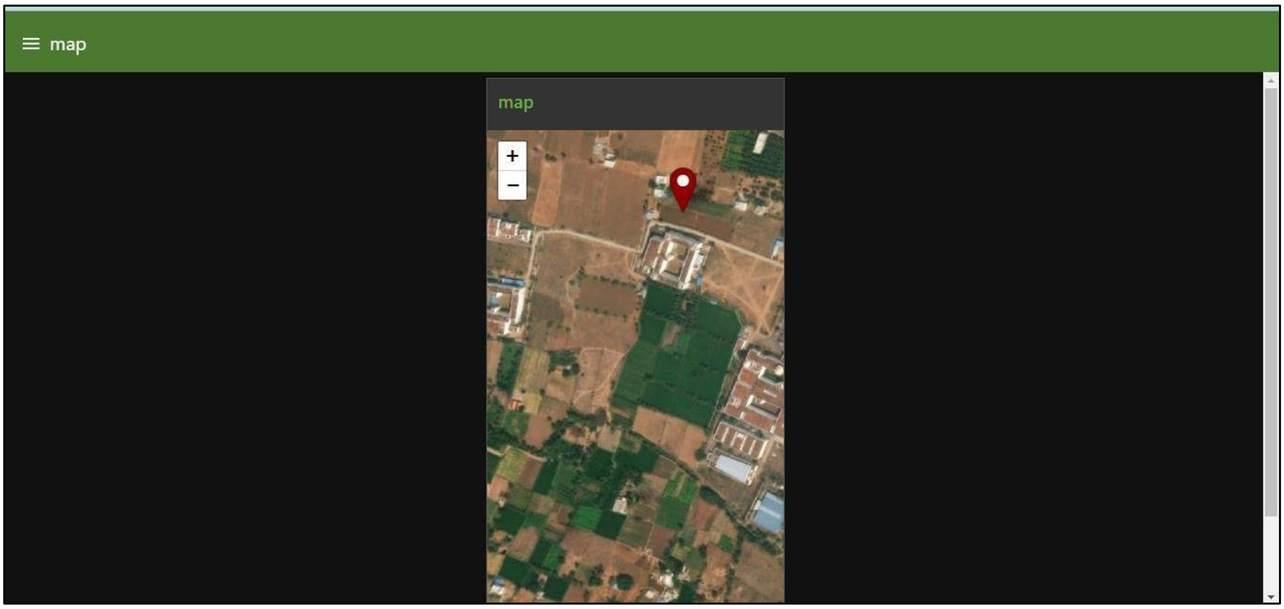
NODE-RED OUPUT:



Child name Madhu

Temperature 106

temp_status Low temperature



≡ map
map
+
−