HO CHI MINH CITY, UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEER

Application Based Internet of Things Report - LAB 1

Student: Nguyễn Công Thành
ID: 1915144

HỒ CHÍ MINH CITY

Content

1 Introduction

In this first LAB, students are proposed to create a simple Thingsboard backend and Dashboard for an IoT application. Students are supposed to follow steps listed in the Implementation section to finish the first Lab.

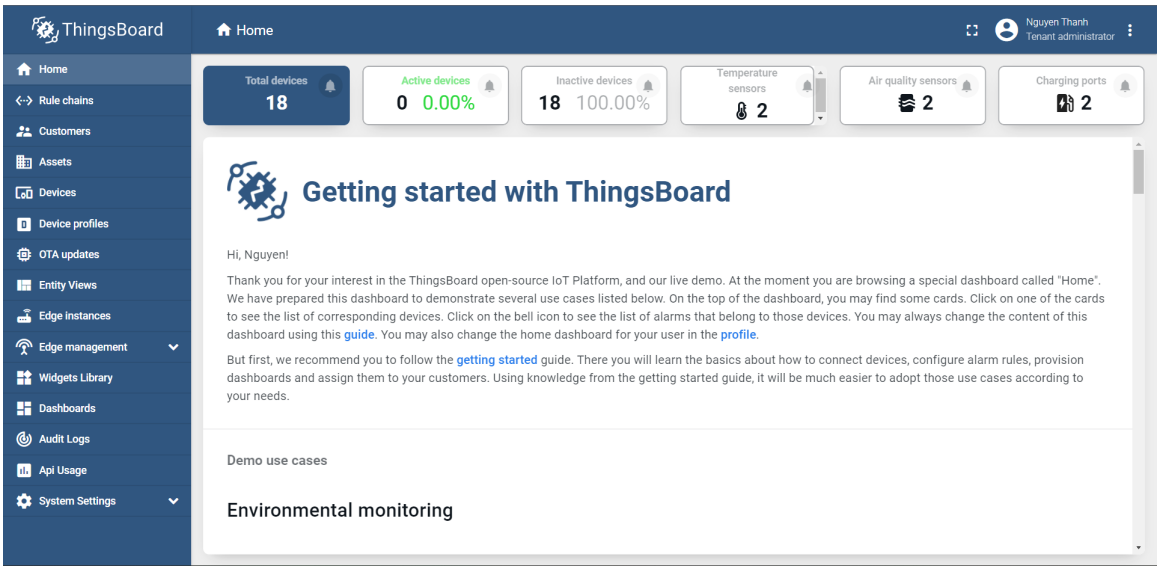
2 Implementation

2.1 Step 1: Create account and a device

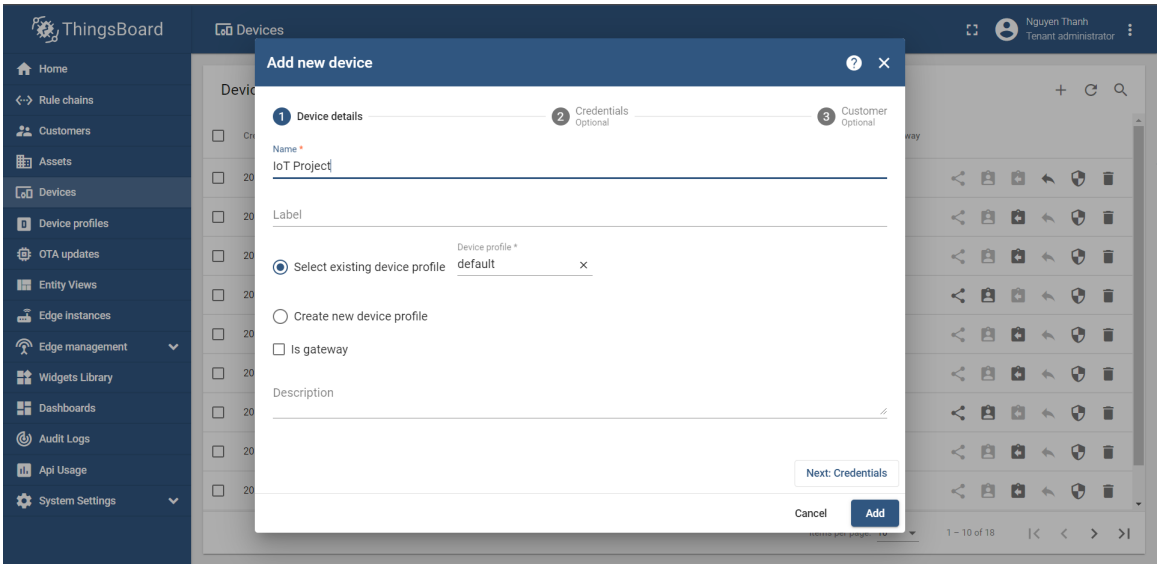
A refferent video is posted in the link bellow:

https://www.youtube.com/watch?v=kWF5ZSkXfE4

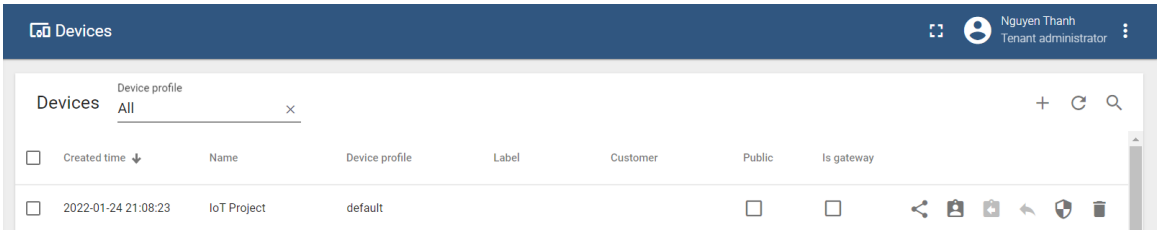
Please login to Thingsboard and create a device, named **IoT Project** for instance.



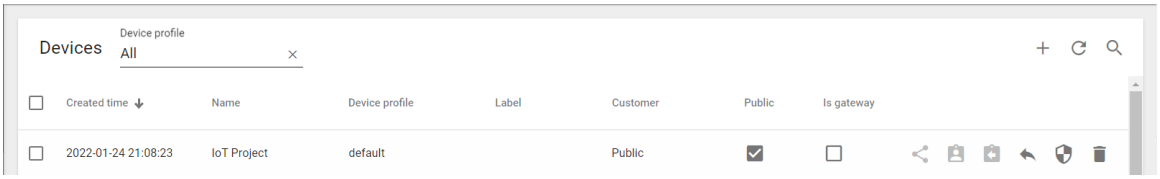
Hình 1: *Login to Thingsboard*



Hình 2: *Create a new device*



Hình 3: *Create a new device*



Hình 4: *Make device public*

2.2 Step 2: Implement python source code

In this step, please create a github account and upload your source code to github. The link of your source code is required to present in this report.

`https://github.com/nct74/IoT_Lab.git`

The manual video for this step can be found at:

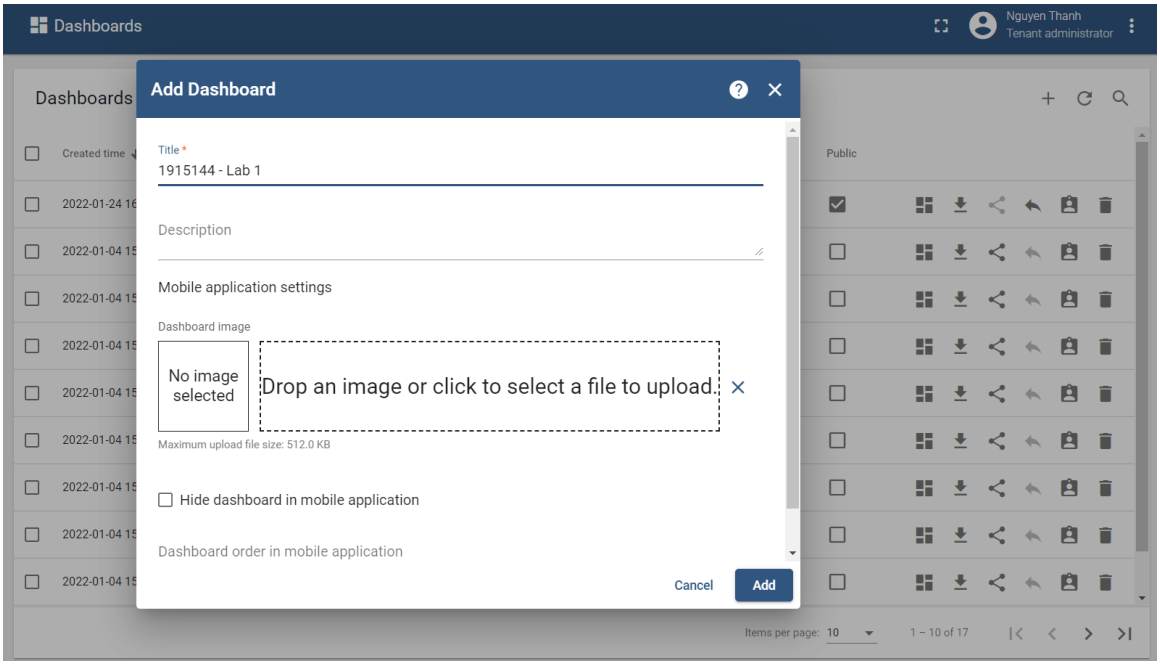
`https://www.youtube.com/watch?v=pJKTgCq_J7Y`

At this step, two random values simulated for the temperature and humidity are sent to the server every 10 seconds.

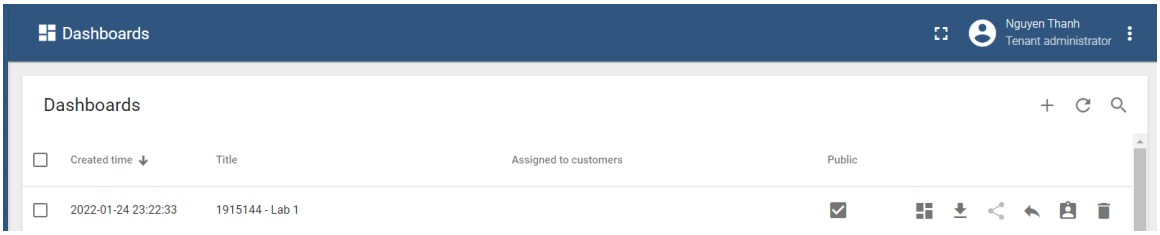
2.3 Step 3: Simple Thingsboard dashboard

Design a simple dashboard with 2 labels to display the values of temperature and humidity. The manual for this step can be found at:

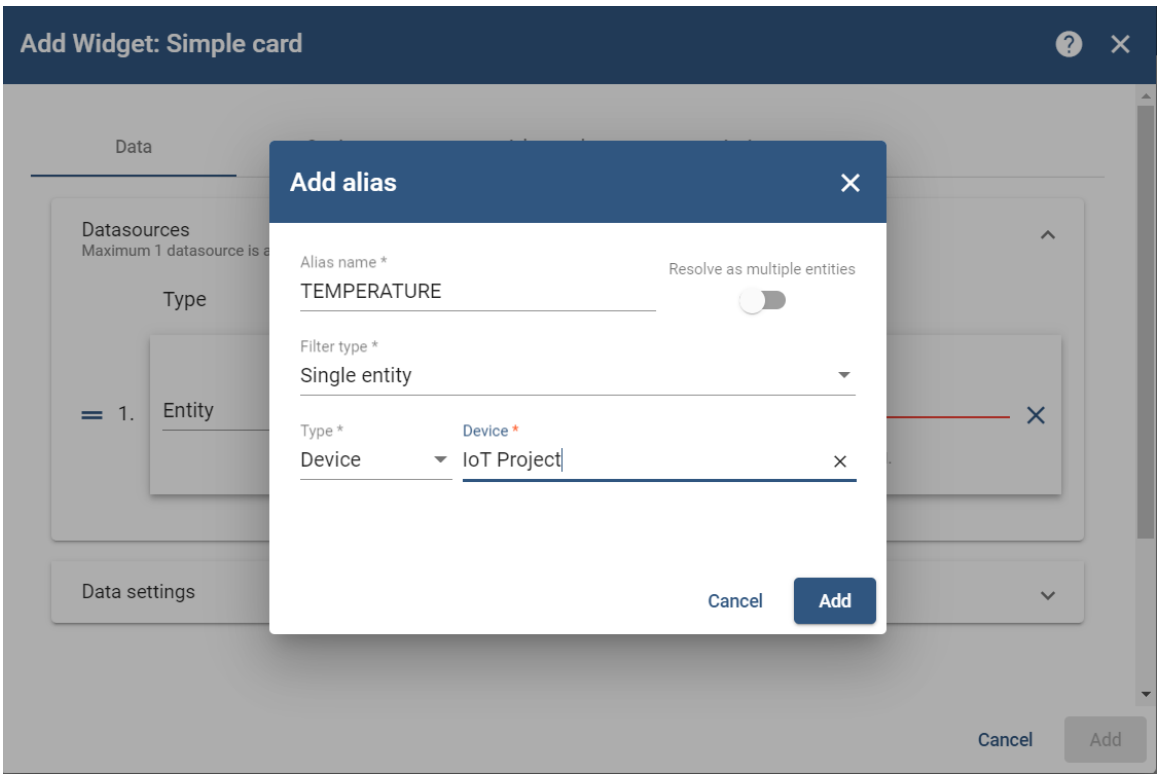
https://www.youtube.com/watch?v=8eQOag5Ymfo



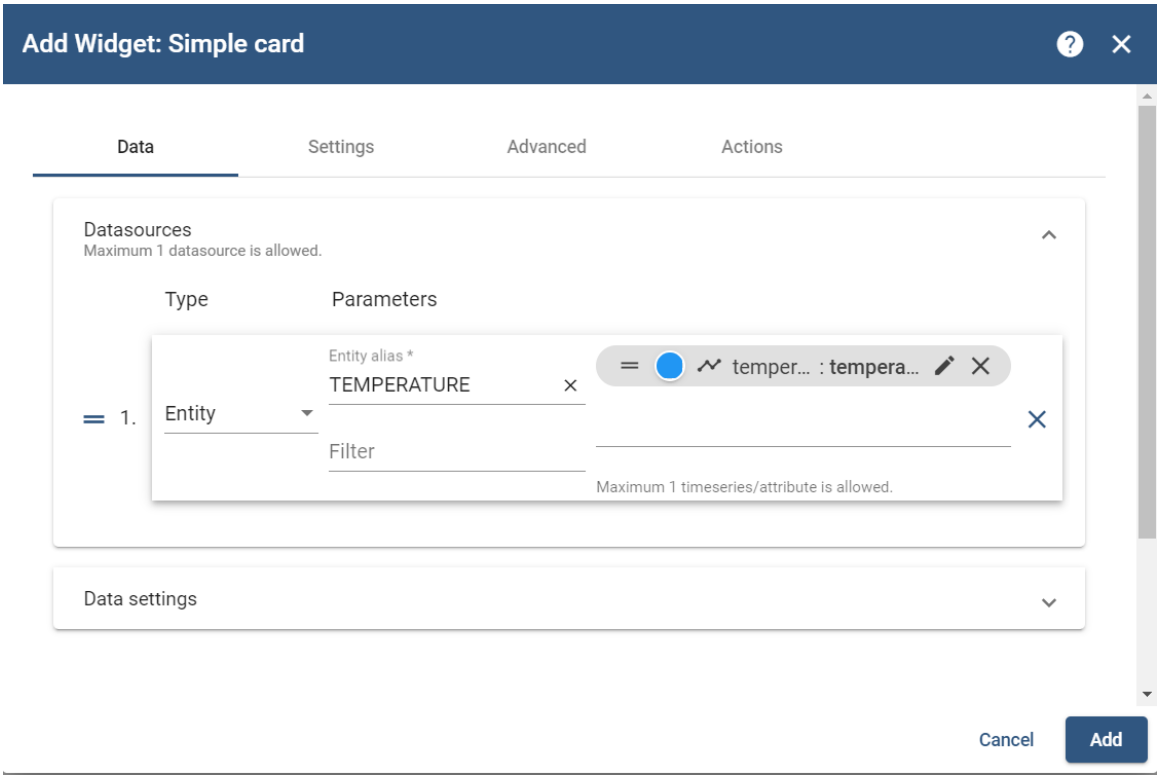
Hình 5: *Create simple dashboard*



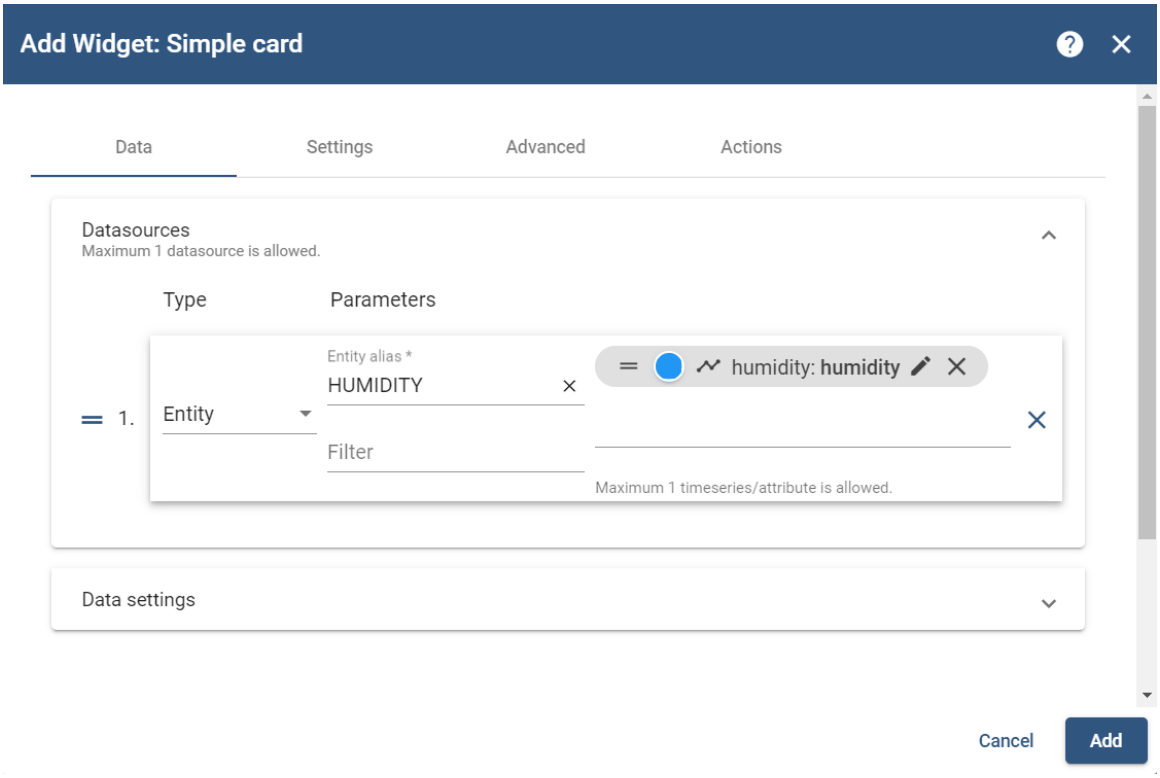
Hình 6: *Make Dashboard Public*



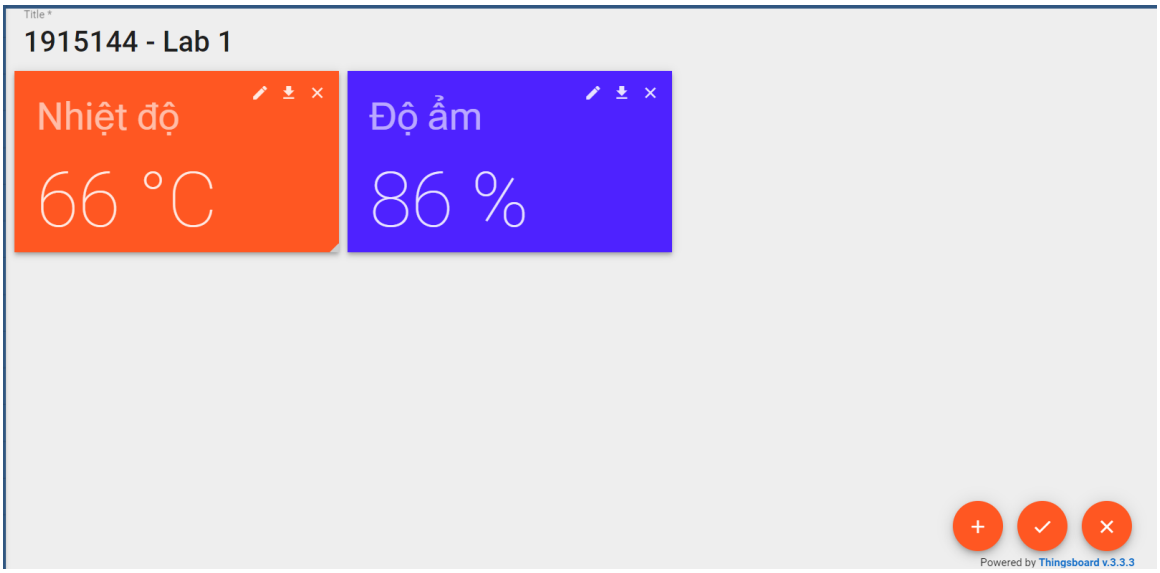
Hình 7: *Add Temperature Card*



Hình 8: *Add Temperature Card*



Hình 9: *Add Humidity Card*



Hình 10: *My dashboard*

2.4 Step 4: Use advanced UI in Thingsboard

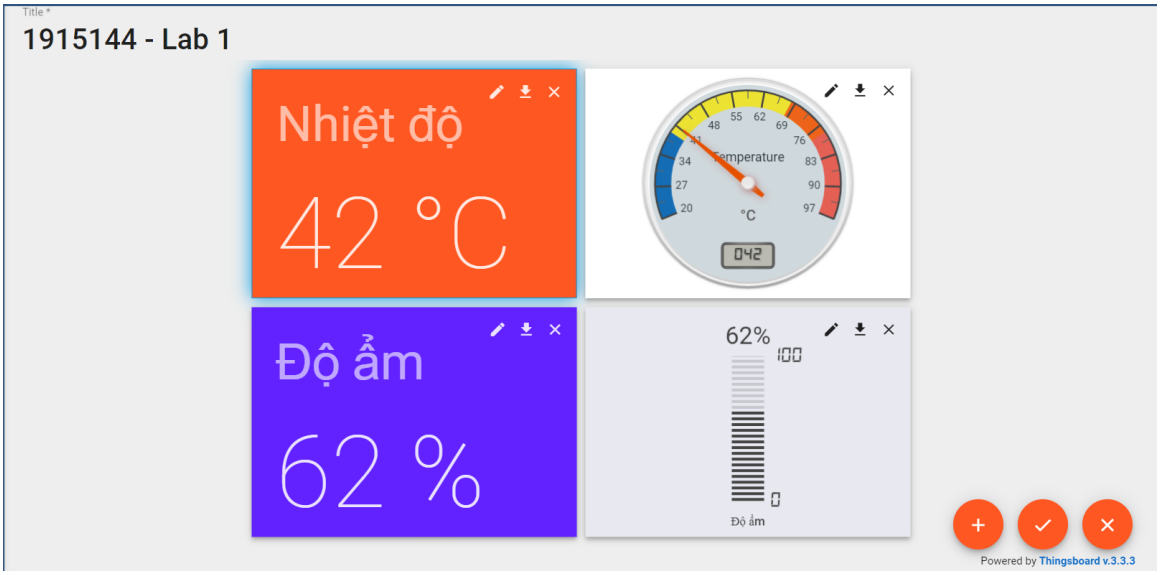
Please use a UI in the Analogue Gause and Digital Gause in your dashboard, to present the value of temperature and humidity.

Publish your dashboard and present the link in this report

https://demo.thingsboard.io/dashboard/d5e3fa40-7d31-11ec-b563-3701f12552b4?publicId=f561a5f0-7cf2-11ec-b563-3701f12552b4

A manual video is posted at:

https://www.youtube.com/watch?v=LFEllRi-5iU



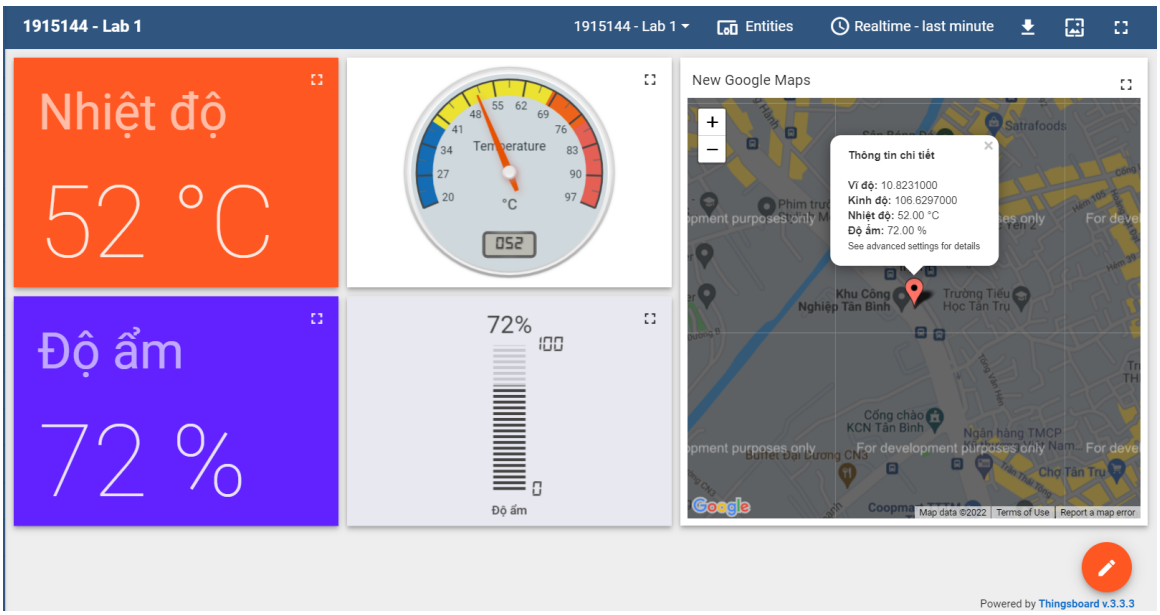
Hình 11: *My dashboard*

2.5 Step 5: Add a map to the dashboard

Finally, add a map to your dashboard. In this case, the **longitude** and **latitude** are required in your python source code. At this step, the latitude and longitude can be set to 10.8231 and 106.6297.

A manual video is posted at:

https://www.youtube.com/watch?v=0XMqH8mdWi0



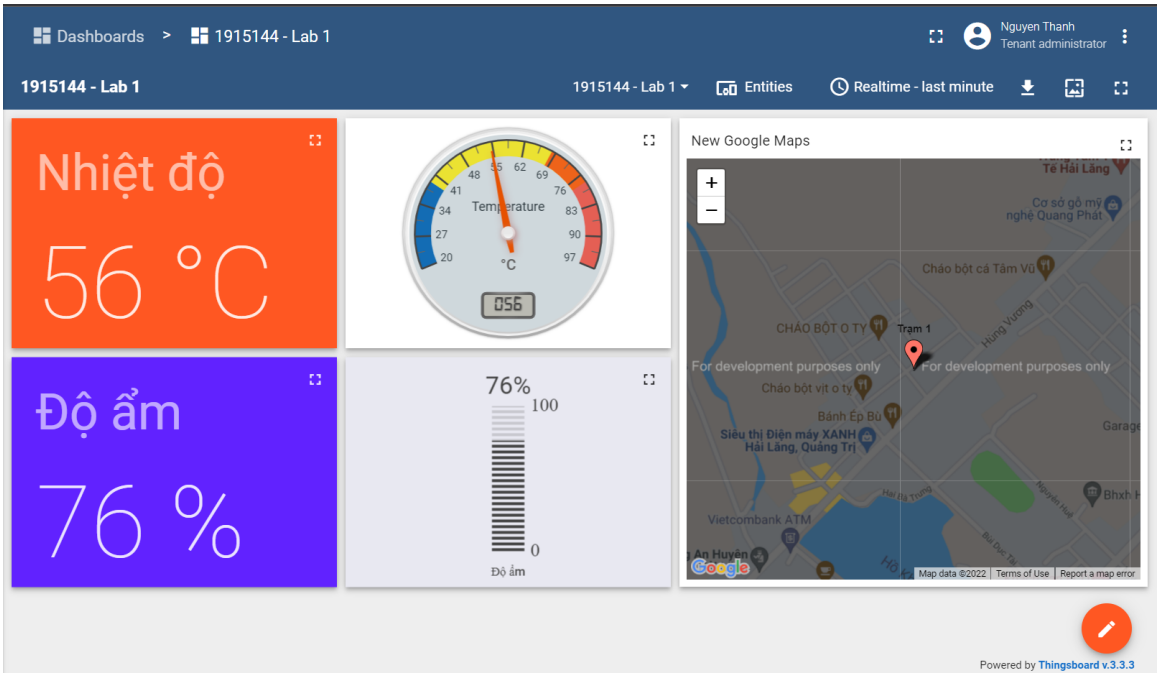
Hình 12: *My dashboard*

3 Extra point (1 point)

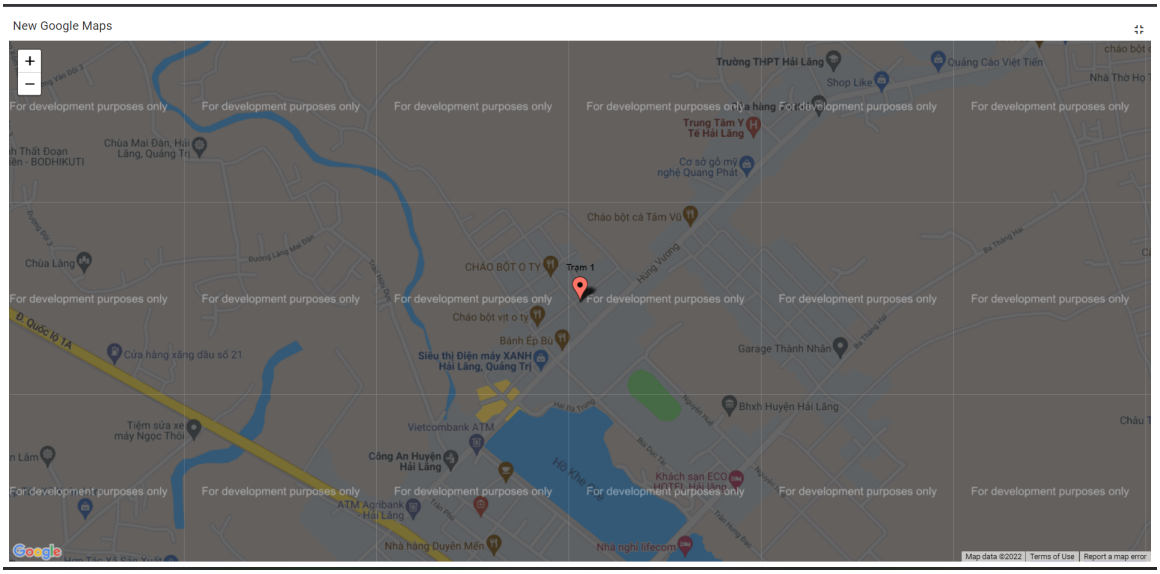
Dynamic update the current longtitude and latitude. Explain your implementation in python source code such as the library which is used, some main python source code to get the value of longtitude and latitude.

Answer

When using the Geocoder library we will locate according to the IP, it may have errors. Therefore, I made locate according to GPS to achieve the most accurate results.



Hình 13: *My dashboard*



Hình 14: *My dashboard*

Listing 1: Source code

```
1  print("1915144 - Nguyen Cong Thanh")
2  import paho.mqtt.client as mqttclient
3  import time
4  import json
5
6  # Library is used to find location
7  # Import modules subprocess, a module used to run new codes and ↩
       applications by creating new processes
8  import subprocess as sp
9  import re
10
11 # from Test import getLocateByIP
12
13 BROKER_ADDRESS = "demo.thingsboard.io"
14 PORT = 1883  # Default 1883
15 THINGS_BOARD_ACCESS_TOKEN = ("0oJj8VCAcXaoRIeWJHsV")
16
17
18 def subscribed(client, userdata, mid, granted_qos):
19     print("Subscribed...")
20
21
22 def recv_message(client, userdata, message):
23     print("Received: ", message.payload.decode("utf-8"))
24     temp_data = {"value": True}
25     try:
26         jsonobj = json.loads(message.payload)
```

```
27          if jsonobj["method"] == "setValue":
28              temp_data["value"] = jsonobj["params"]
29              client.publish("v1/devices/me/attributes", json.dumps(↩
                    temp_data), 1)
30      except:
31          pass
32
33
34  def connected(client, usedata, flags, rc):
35      if rc == 0:
36          print("Thingsboard connected successfully!!")
37          client.subscribe("v1/devices/me/rpc/request/+")
38      else:
39          print("Connection is failed")
40
41
42  client = mqttclient.Client("Gateway_Thingsboard")
43  client.username_pw_set(THINGS_BOARD_ACCESS_TOKEN)
44
45  client.on_connect = connected
46  client.connect(BROKER_ADDRESS, 1883)
47  client.loop_start()
48
49  client.on_subscribe = subscribed
50  client.on_message = recv_message
51
52  temp = 30
53  humi = 50
54  counter = 0
55  longitude = 0
56  latitude = 0
57
58  # Source: https://stackoverflow.com/questions/44400560/using-windows-↩
      gps-location-service-in-a-python-script/44462120
59  wt = 5  # Wait time -- I purposefully make it wait before the shell ↩
      command
60  accuracy = 3  # Starting desired accuracy is fine and builds at x1.5 ↩
      per loop
61
62  while True:
63
64      time.sleep(wt)  # Add Delay in the execution of program with wt ↩
          seconds
65      pshellcomm = ["powershell"]  # Run powershell in python script
66      pshellcomm.append(
```

```
67          "add-type -assemblyname system.device; "
68          "$loc = new-object system.device.location.geocoordinatewatcher↩
                ;"
69          "$loc.start(); "
70          'while(($loc.status -ne "Ready") -and ($loc.permission -ne "↩
                Denied")) '
71          "{start-sleep -milliseconds 100}; "
72          "$acc = %d; "
73          "while($loc.position.location.horizontalaccuracy -gt $acc) "
74          "{start-sleep -milliseconds 100; $acc = [math]::Round($acc↩
                *1.5)}; "
75          "$loc.position.location.latitude; "
76          "$loc.position.location.longitude; "
77          "$loc.position.location.horizontalaccuracy; "
78          "$loc.stop()" % (accuracy)
79      )
80
81      # Remove >>> $acc = [math]::Round($acc*1.5) <<< to remove accuracy↩
            builder
82      # Once removed, try setting accuracy = 10, 20, 50, 100, 1000 to ↩
            see if that affects the results
83      # Note: This code will hang if your desired accuracy is too fine ↩
            for your device
84      # Note: This code will hang if you interact with the Command ↩
            Prompt AT ALL
85      # Try pressing ESC or CTRL-C once if you interacted with the CMD,
86      # this might allow the process to continue
87
88      p = sp.Popen(pshellcomm, stdin=sp.PIPE, stdout=sp.PIPE, stderr=sp.↩
            STDOUT, text=True)
89      (out, err) = p.communicate()
90      out = re.split("\n", out)  # Split a string into a list
91
92      latitude = float(out[0])  # Assign latitude from the output
93      longitude = float(out[1])  # Assign longitude from the output
94
95      collect_data = {
96          "temperature": temp,
97          "humidity": humi,
98          "longitude": longitude,
99          "latitude": latitude,
100     }
101     temp += 1
102     humi += 1
103     client.publish(
```

```
104             "v1/devices/me/telemetry", json.dumps(collect_data), 1
105         )
106         time.sleep(10)
```