**GUILD WIRELESS SENSOR NETWORK**

1. **Cài đặt**
2. **nodeJS :**

# apt-get update

# apt-get install nodejs npm

# npm install express ejs

Note: nếu k dùng được lệnh thì cứ them sudo ở đầu.

**2.Visual studio code (hoặc bất kì 1 editor nào khác)**

sudo apt update

sudo apt install code

**3.Tạo project**

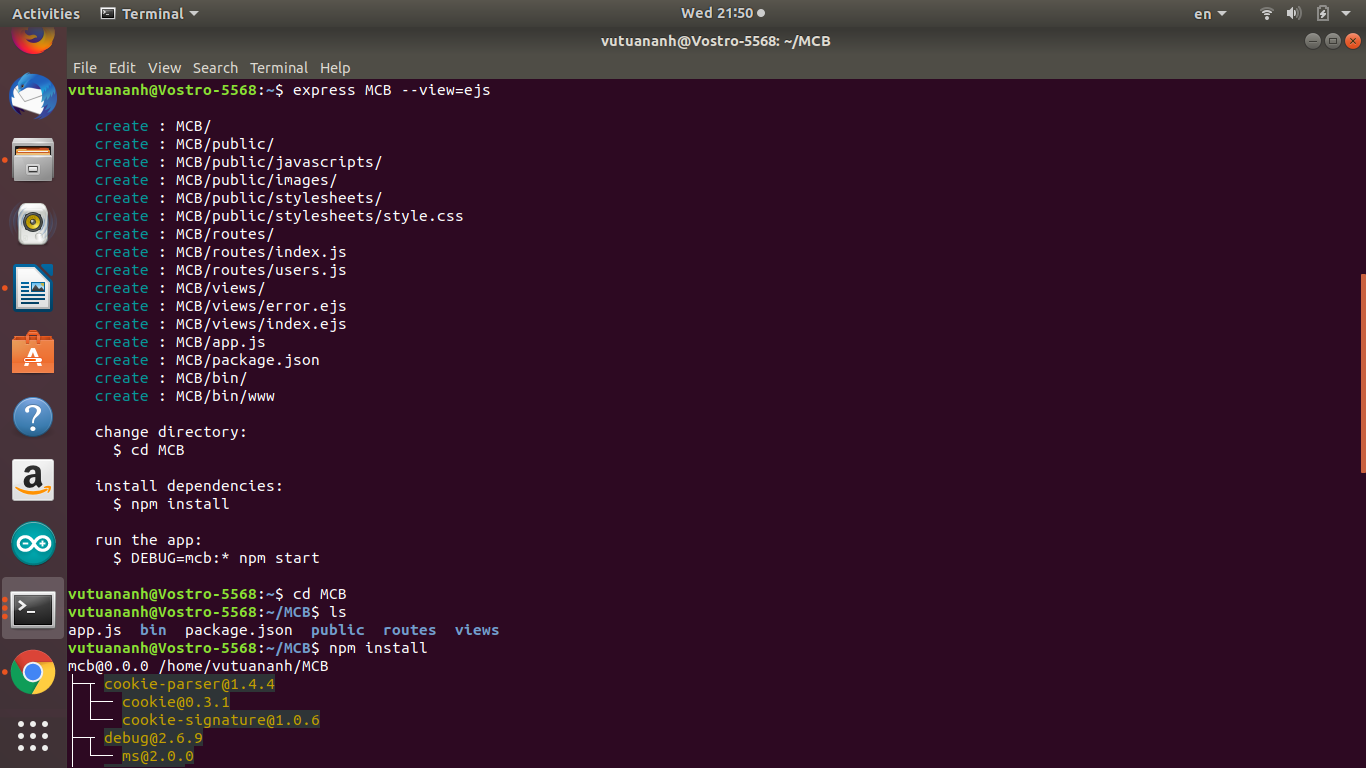
**- Nếu không dùng được lệnh dưới thì thêm sudo vào đầu.**

- Dung lenh express MCB --view=ejs để tạo project

- cd MCB để vào thư mục MCB

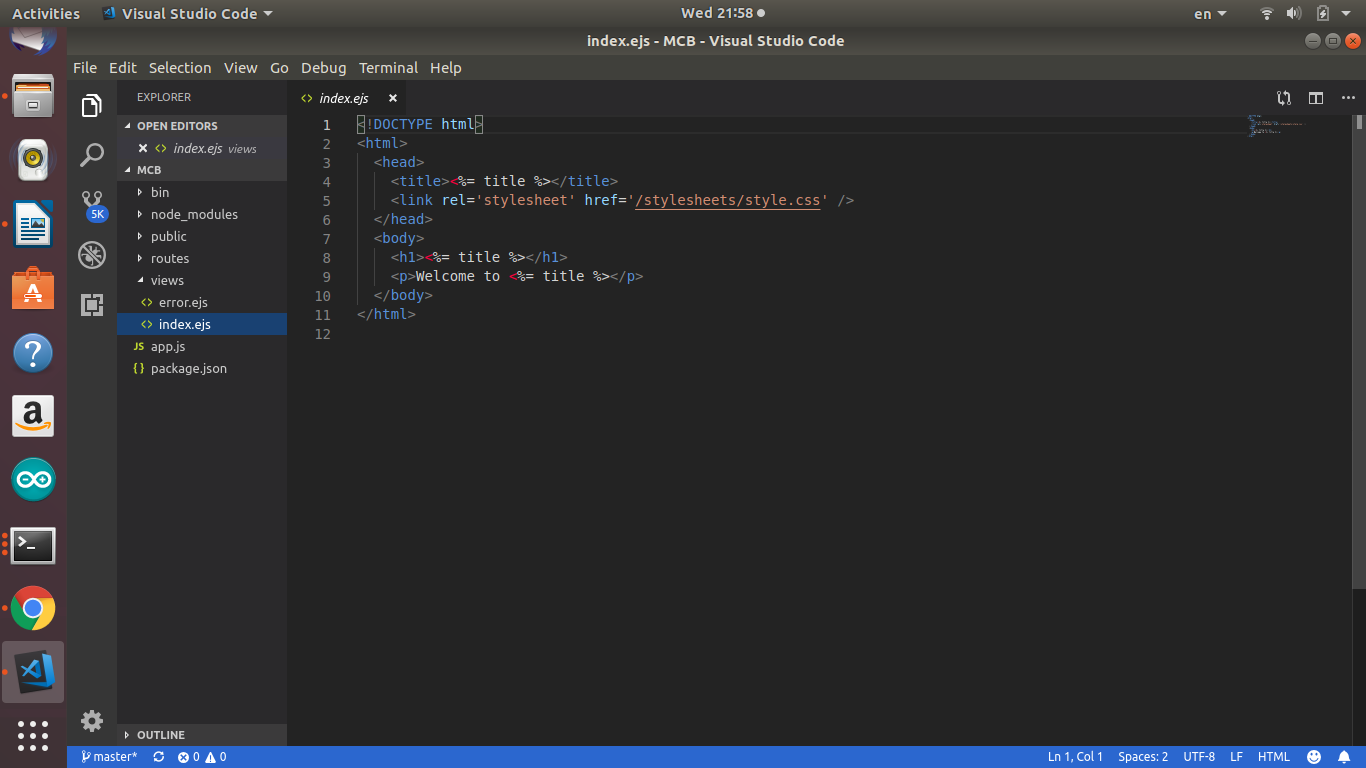
- npm install để cài đặt module cho project

- npm install mysql để cài đặt mysql



M

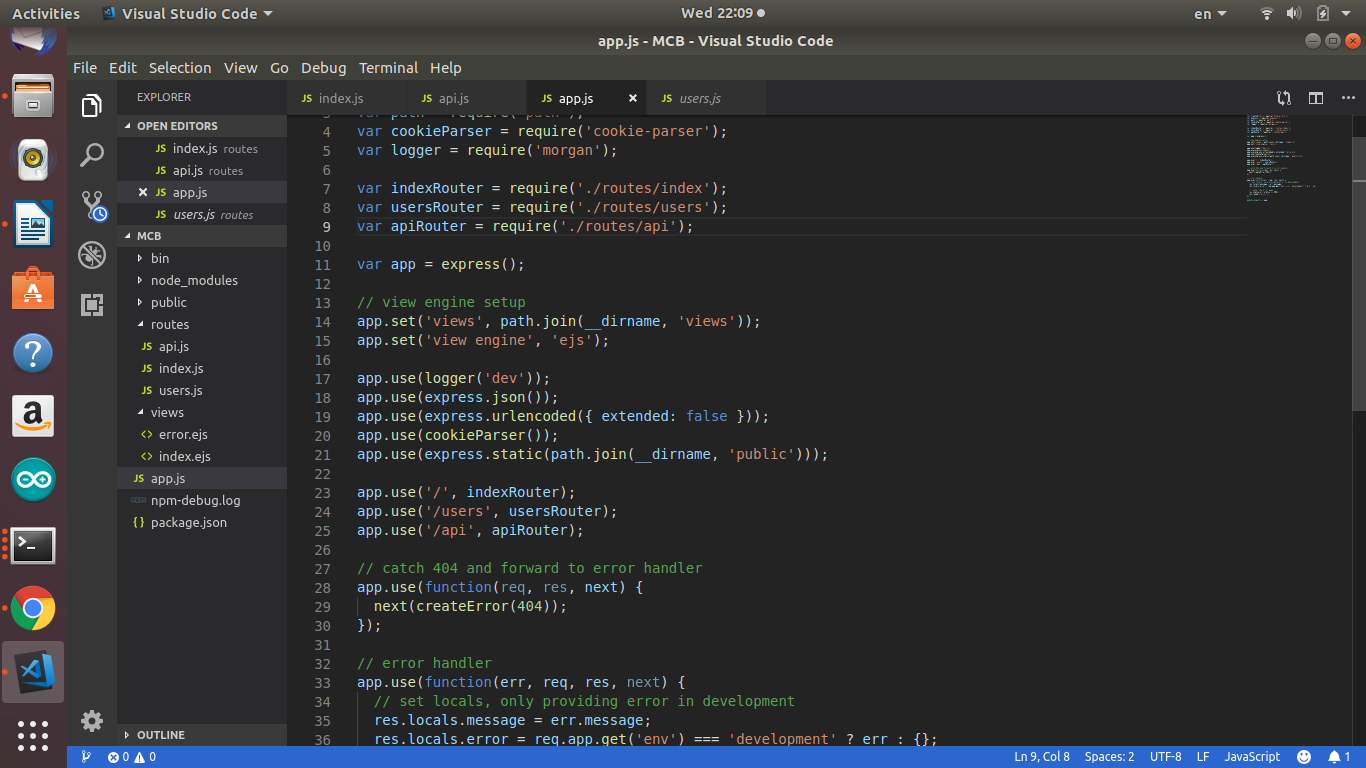
* + Mở project trên visual code bằng cách chọn File→Open Folder->MCB



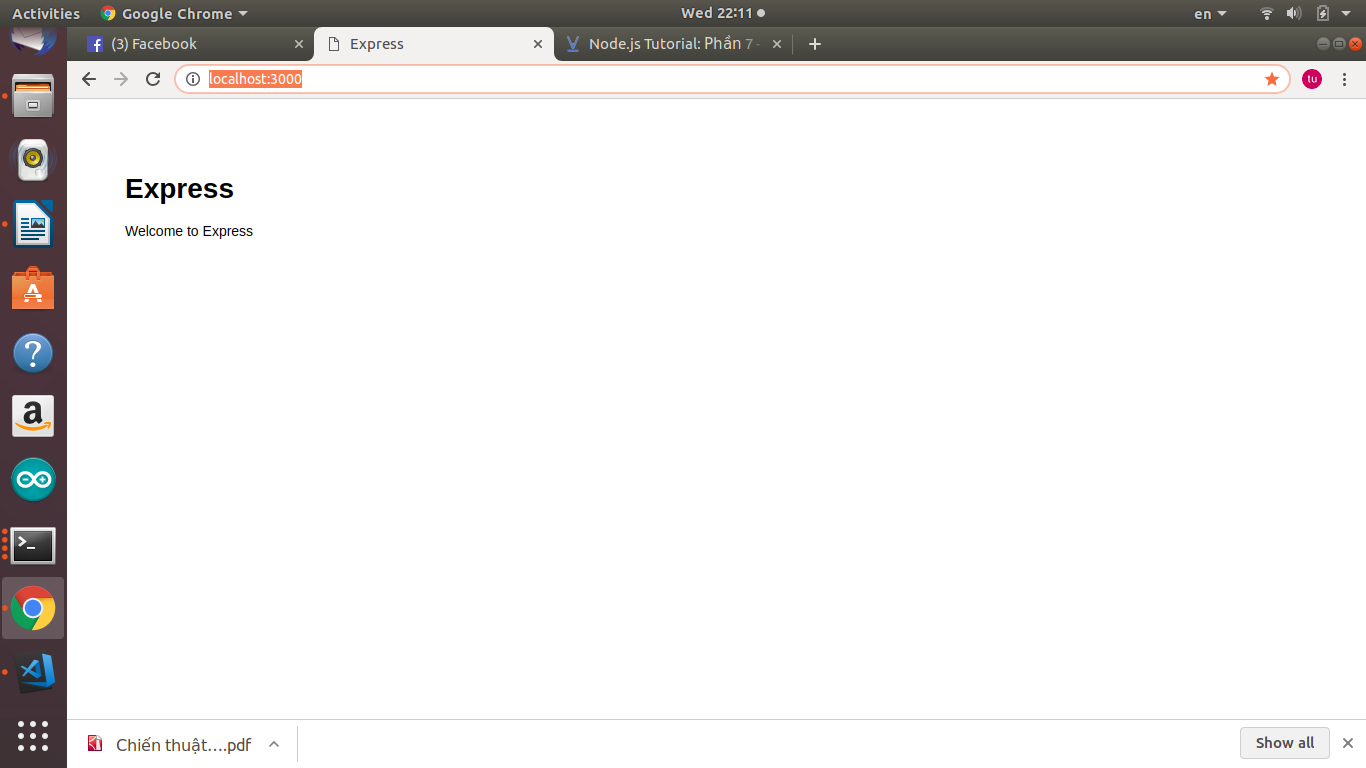
-Trong thư mục views, file index.ejs nơi chúng ta code toàn bộ html(hoặc có thể tạo nhiều file .ejs để hiển thị html, xem them tại: <https://viblo.asia/p/nodejs-tutorial-phan-7-su-dung-ejs-lam-template-engine-trong-express-Do754N74ZM6>)

-Trong thư mục routes, tạo 1 file api.js, copy code của file user.js sang api.js

Thêm khai báo và khởi tạo đường dẫn cho api trong app.js như hình



-Trên terminal gõ npm start để chạy project, mở trình duyệt và truy cập <http://localhost:3000/> ta được



1. **Code**
2. **Tạo giao diện Web**

**Note: có thể tự code , nếu có thể tự code thì bỏ qua phần này**

- Tải các bootraps template mẫu tại : <https://uicookies.com/free-bootstrap-admin-templates/>

- Xóa các phần không cần thiết đi để lại khung chính.

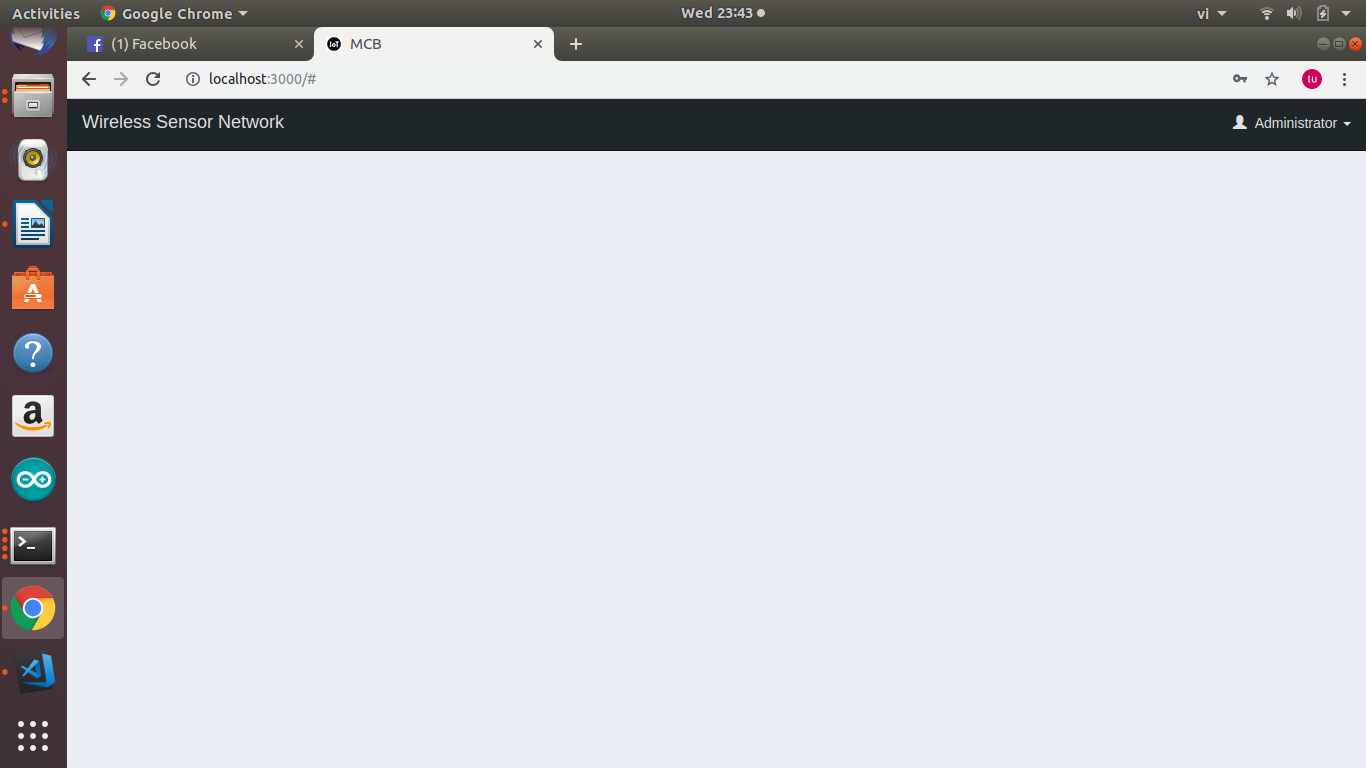
- Copy toàn bộ code html vào file index.ejs.

- Copy các folder css, bootrap, img, … vào trong folder public của project.

- day la link 1 mau bootrap: <https://drive.google.com/file/d/0B-ML1u-nYlMaWDA0cWdzV0VkX2M/view>

-Từ bây giờ phần html ta chỉ code trên index.ejs, chú ý các css sẽ phụ thuộc theo bootraps bạn tải hoặc do bạn tạo ra. Đây là mình sử dụng css theo bootraps của mình.

-Sau khi xong ta được 1 giao diện Web:



Ta them 1 list để có các lựa chọn

<div id="sidebar-wrapper">

<ul class="sidebar-nav">

<li id="li-1">

<a onClick="openTab(1);" href="#"><span class="glyphicon glyphicon-modal-window"></span>&nbsp; Room 1</a>

</li>

<li id="li-2">

<a onClick="openTab(2);" href="#"><span class="glyphicon glyphicon-modal-window"></span>&nbsp; Room 2</a>

</li>

<li id="li-3">

<a onClick="openTab(3);" href="#"><span class="glyphicon glyphicon-modal-window"></span>&nbsp; Room 3</a>

</li>

<li id="li-4" class="active">

<a onClick="openTab(4);" href="#"><span class="glyphicon glyphicon-modal-window"></span>&nbsp; Room 4</a>

</li>

<li id="li-5" class="active">

<a onClick="openTab(5);" href="#"><span class="glyphicon glyphicon-map-marker"></span>&nbsp; Location</a>

</li>

</ul>

</div>

-Script để mở tab:

<script>

function openTab(id) {

$('#tab1').hide();

$('#tab2').hide();

$('#tab3').hide();

$('#tab4').hide();

$('#tab5').hide();

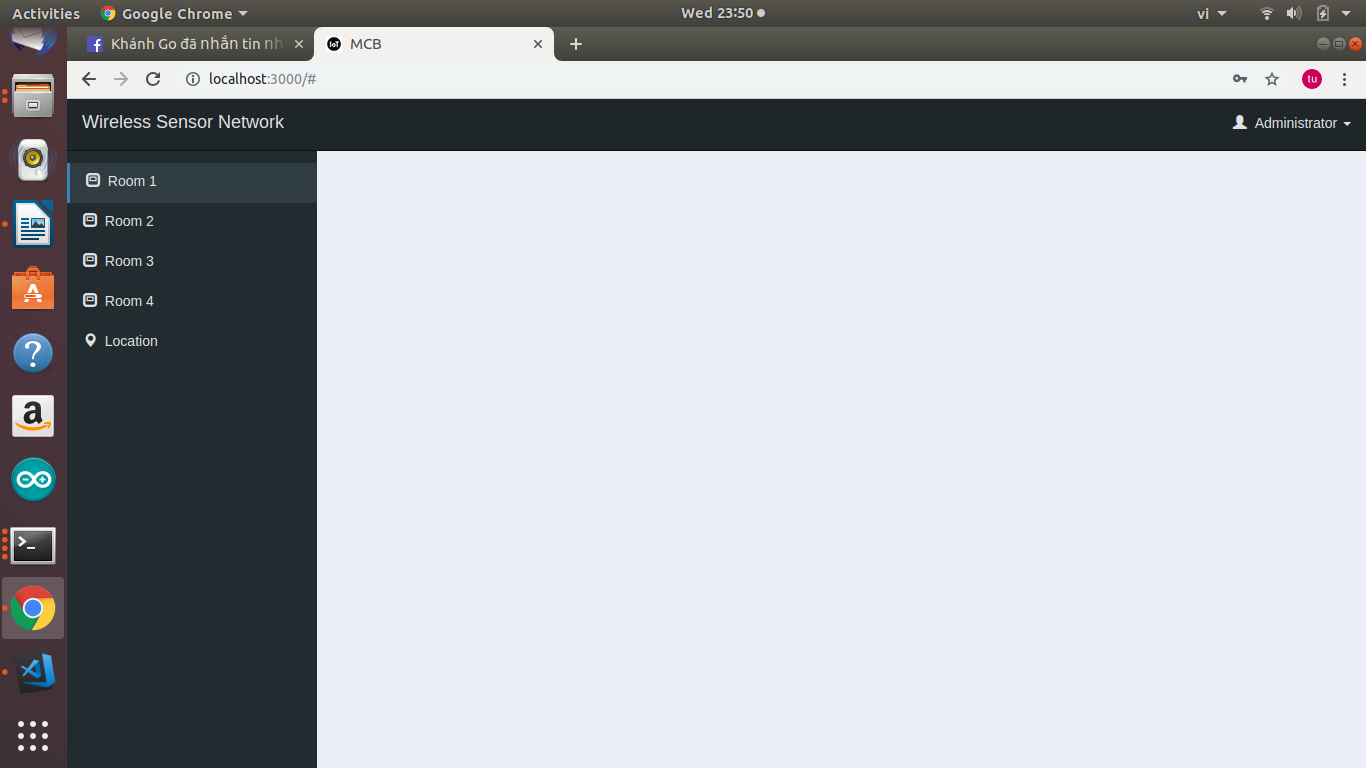
$('#tab' + id).show();

$('.sidebar-nav li').removeClass('active');

$('#li-' + id).addClass('active');

}

</script>

-Bây giờ ta sẽ code nội dung, ta có thể tra dashboard mẫu trên mạng để tải về hoặc làm theo, ở đây mình sẽ tự code nội dung của tab room1, tao tab1

<div id="tab1" class="container-fluid">

</div>

-Ta sẽ code nội dung của room1 trong thẻ div tab1

-Tạo 1 switch để điều khiển led: có thể lên mạng tìm 1 css cho switch rồi copy về trong file css của mình, sau đó code hiện thị switch trong divtab1

<label class="switch" style = "margin-left: 10px;">

<input id="check-led1" type="checkbox" checked>

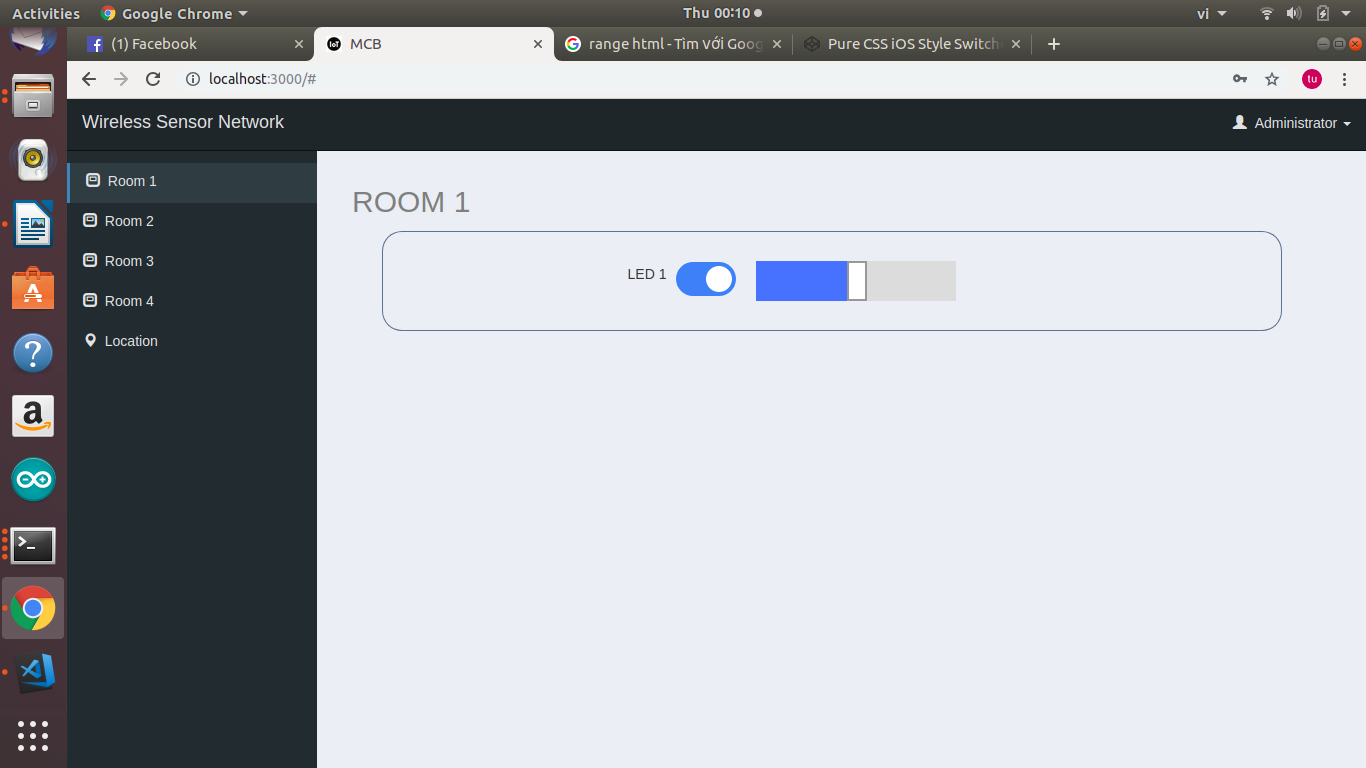
<span class="slider round"></span>

</label>

-Tạo 1 input range để điều chỉnh độ sang của đèn tương tự như swtich.

<input id="control-led1" style="margin-left:20px; margin-right:100px;" type="range" min="0" max="255">

Sau khi thêm ta được:



-Tạo 1 thẻ div để hiện thi data, style cho phần temp thì các bạn tự code

+css:

|  |
| --- |
| .temp{ |
|  | Float:left; width:200px; height: 150px; border-radius: 20px 20px 20px 20px; display: flex;align-items: center;justify-content: center; |
|  | } |

Html:

<div style="border: 1px solid rgba(57,86,118, 0.80); background: #ECF0F5; width:450px; height: 150px; border-radius: 20px 20px 20px 20px; margin-left:20px;">

<div id="temp1" class="temp" style='background: #DC143C;'>

<p style="font-size:50px; color:white; float:left;">

<span class="fas fa-thermometer-three-quarters"></span>&nbsp;

<span id="sensor-temp"></span>

</p>

</div>

<div class="col-md-6 col-md-6 col-sm-6" style="margin-left: 10px; margin-top:10px;">

<h4 style="color:#DC143C;">Temperature<small id="stttemp1" class="pull-right" style="color:#DC143C;"><span id="stt-temp"></span></small></h4>

<table class="table table-condensed">

<tbody>

<tr>

<td>Max</td>

<td><span id="maxtemp"></span></td>

</tr>

<tr>

<td>Min</td>

<td><span id="mintemp"></span></td>

</tr>

<tr>

<td>Average</td>

<td><span id="avgtemp"></span></td>

<td>Today</td>

</tr>

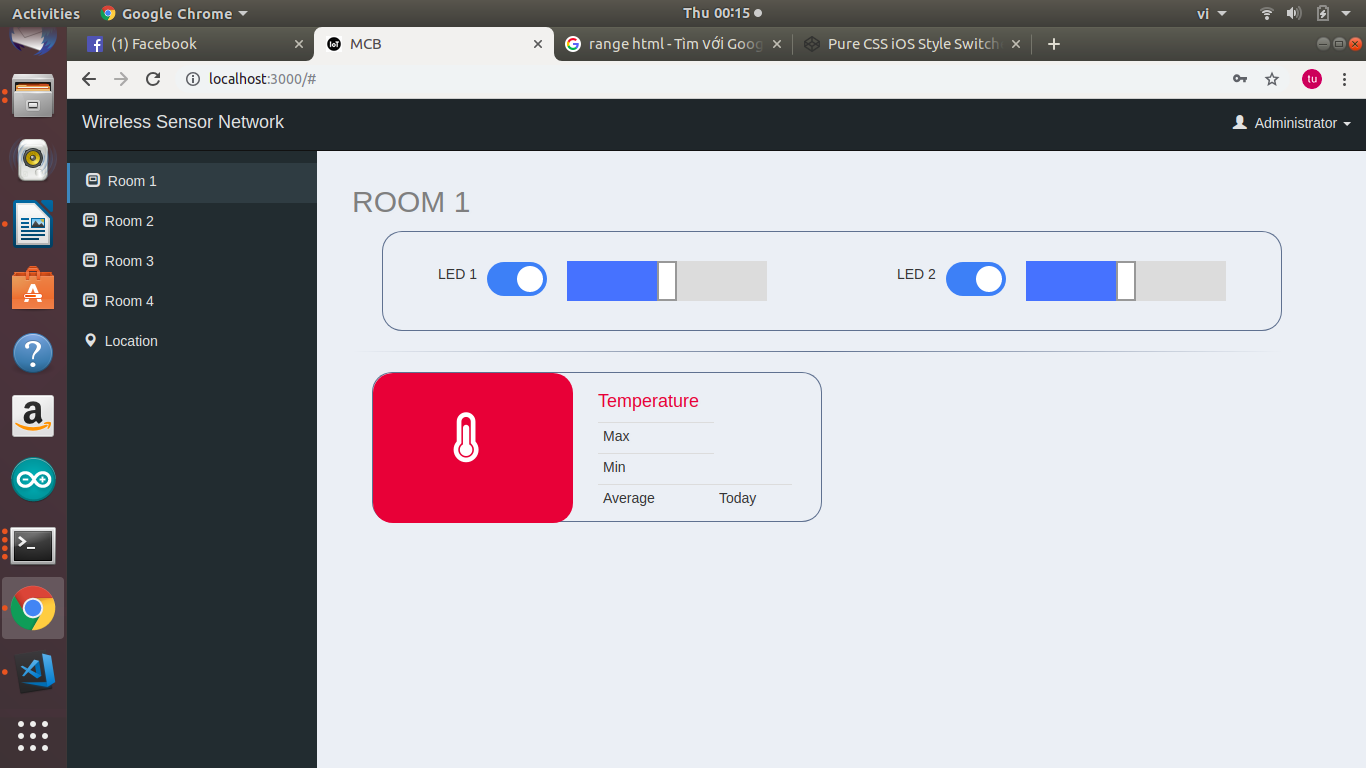
</tbody>

</table>

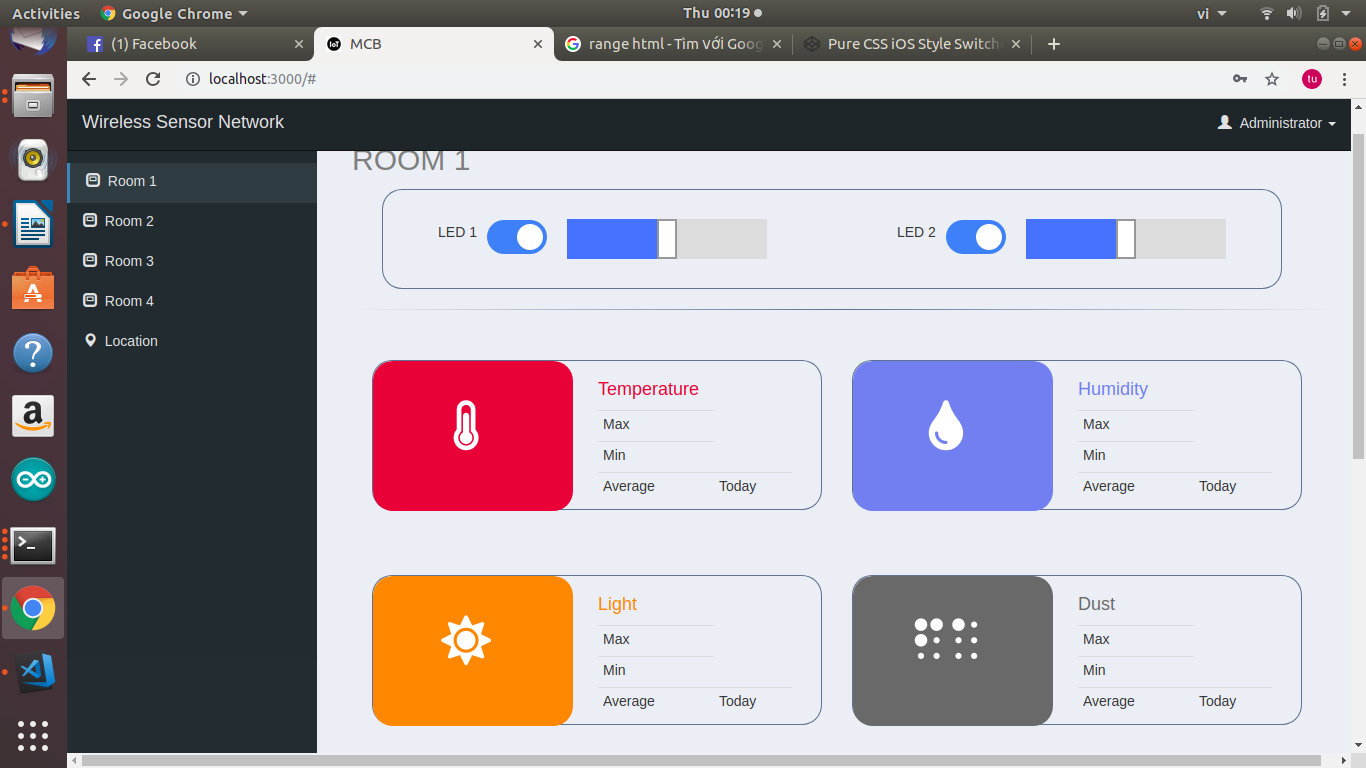
</div>

</div>

Sau khi thêm ta được:



Tương tự với các giá trị khác



-Vẽ đồ thị:

<div style = "display: flex;align-items: center;justify-content: center;">

<div id="chartlight" style="height: 200px; width: 450px; float: left;"></div>

<div id="chartdust" style="height: 200px; width: 450px;"></div>

</div>

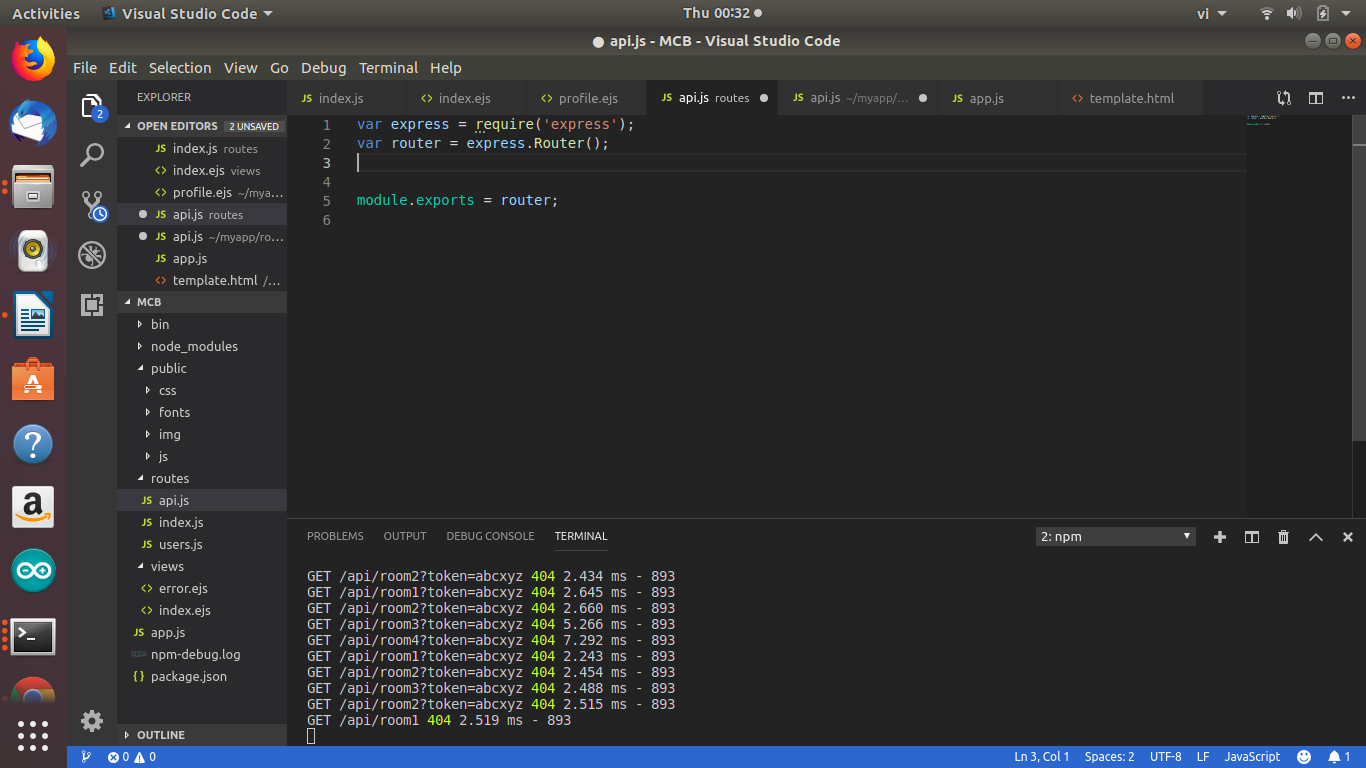
<script src="https://canvasjs.com/assets/script/canvasjs.min.js"></script>

Phần data của đồ thị thì sang phần update database

Note: có thể tự code

1. **Tạo các API**

**\*Code trong api.js**

****

-Để lấy dữ liệu từ database, hay thêm dữ liệu vào database ta cần kết nối đến mysql:

var express = require('express');

var router = express.Router();

var mysql = require('mysql');

var con = mysql.createConnection({

host: "localhost",

user: "tuananh",

password: "abc13579",

database: "wordpress"

});

con.connect();

-Để update dữ liệu thì ta sẽ lấy ra giá trị có ID lớn nhất

-Các giá trị max,min,avg tương tự

-Ta sẽ trả về 1 kiểu json cho trang web có địa chỉ <http://localhost:3000/api/room1>

router.get('/room1', function(req, res){

con.query("SELECT \* FROM room1 where id = (select max(id) from room1)", function (err, result, fields) {

if (err) throw err;

con.query("SELECT temp FROM room1 where temp = (select max(temp) from room1)", function (err, maxtemp, fields) {

if (err) throw err;

con.query("SELECT hum FROM room1 where hum = (select max(hum) from room1)", function (err, maxhum, fields) {

if (err) throw err;

con.query("SELECT light FROM room1 where light = (select max(light) from room1)", function (err, maxlight, fields) {

if (err) throw err;

con.query("SELECT dust FROM room1 where dust = (select max(dust) from room1)", function (err, maxdust, fields) {

if (err) throw err;

con.query("SELECT temp FROM room1 where temp = (select min(temp) from room1)", function (err, mintemp, fields) {

if (err) throw err;

con.query("SELECT hum FROM room1 where hum = (select min(hum) from room1)", function (err, minhum, fields) {

if (err) throw err;

con.query("SELECT light FROM room1 where light = (select min(light) from room1)", function (err, minlight, fields) {

if (err) throw err;

con.query("SELECT dust FROM room1 where dust = (select min(dust) from room1)", function (err, mindust, fields) {

if (err) throw err;

con.query("SELECT AVG(temp) AS avgtemp FROM room1", function (err, avgtemp, fields) {

if (err) throw err;

con.query("SELECT AVG(hum) AS avghum FROM room1", function (err, avghum, fields) {

if (err) throw err;

con.query("SELECT AVG(light) AS avglight FROM room1", function (err, avglight, fields) {

if (err) throw err;

con.query("SELECT AVG(dust) AS avgdust FROM room1", function (err, avgdust, fields) {

if (err) throw err;

var a = {result: result[0],

maxtemp: maxtemp[0], maxhum: maxhum[0], maxlight: maxlight[0], maxdust: maxdust[0],

mintemp: mintemp[0], minhum: minhum[0], minlight: minlight[0], mindust: mindust[0],

avgtemp: avgtemp[0], avghum: avghum[0], avglight: avglight[0], avgdust: avgdust[0]

};

res.json(a);

});

});

});

});

});

});

});

});

});

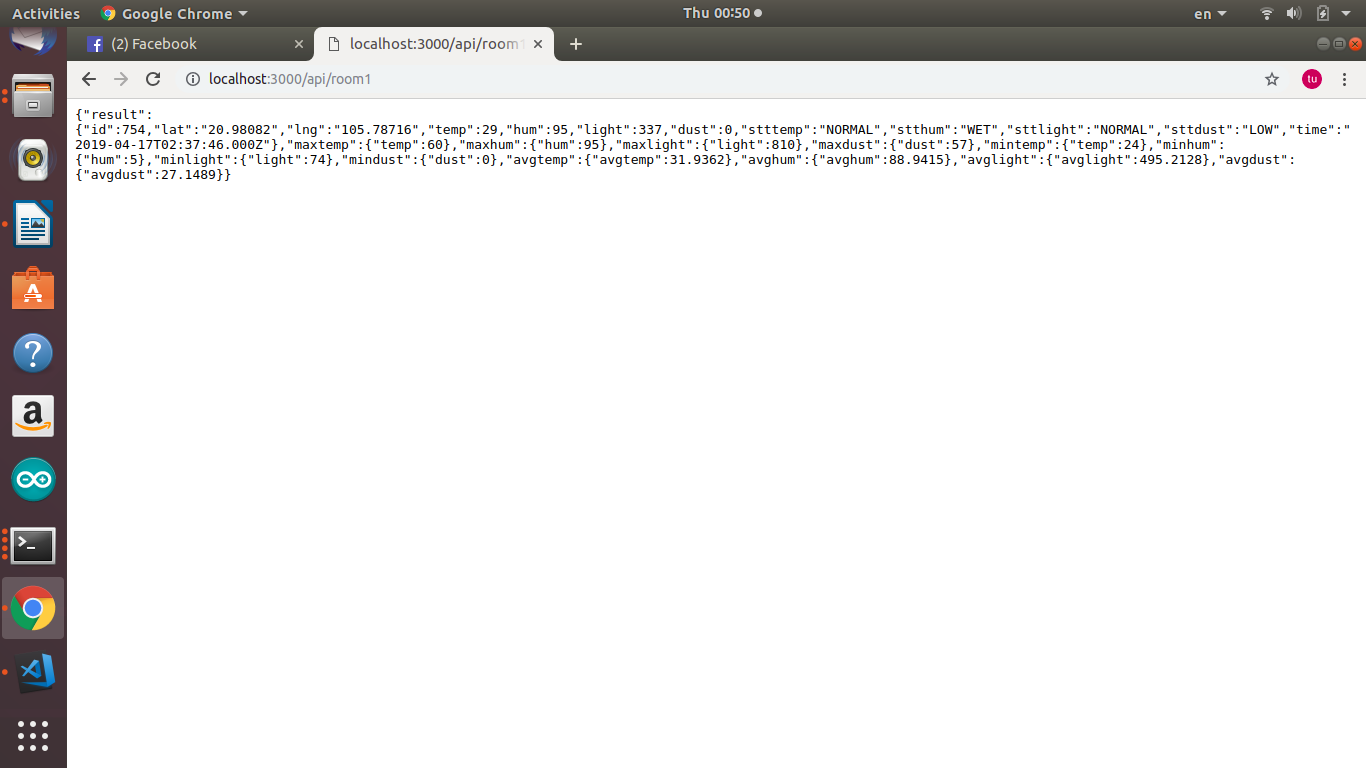
});

});

});

});

});



-Với switch và input range thì ta sẽ lấy giá trị trả về để lưu vào database

router.post('/led1', function(req, res){

con.query(`insert into sttled1(led1,time) values(${req.body.signal},now())`, function (err, result, fields) {

if (err) throw err;

res.json(result[0]);

});

});

router.post('/controlled1', function(req, res){

con.query(`insert into controlled1(led1,time) values(${req.body.pwm},now())`, function (err, result, fields) {

if (err) throw err;

res.json({

message: "success",

data: result[0]

});

});

});

1. **Update dữ liệu từ database lên web**

-Để update dữ liệu ta tạo 1 javascript yêu cầu api room1, ta thêm đoạn script sau phần hiện thị data để dễ kiểm soát

<script>

setInterval(function(){

$.getJSON( "/api/room1, function( a ) {

var temp = document.getElementById('temp1');

var stttemp = document.getElementById('stttemp1');

var hum = document.getElementById('hum1');

var stthum = document.getElementById('stthum1');

var light = document.getElementById('light1');

var sttlight = document.getElementById('sttlight1');

var dust = document.getElementById('dust1');

var sttdust = document.getElementById('sttdust1');

if(a.result.temp <= 20){

temp.style.background = "#2E9AFE";

stttemp.style.color = "#2E9AFE";

}

if(a.result.temp > 20 && a.result.temp <= 30){

temp.style.background = "#DC143C";

stttemp.style.color = "#DC143C";

}

if(a.result.temp > 30){

temp.style.background = "#FF4000";

stttemp.style.color = "#FF4000";

}

if(a.result.hum <= 40){

hum.style.background = "#2E9AFE";

stthum.style.color = "#2E9AFE";

}

if(a.result.hum > 40 && a.result.hum <= 80){

hum.style.background = "#6495ED";

stthum.style.color = "#6495ED";

}

if(a.result.hum > 80){

hum.style.background = "#084B8A";

stthum.style.color = "#084B8A";

}

if(a.result.light <= 200){

light.style.background = "#FE9A2E";

sttlight.style.color = "#FE9A2E";

}

if(a.result.light > 200 && a.result.light <= 600){

light.style.background = "#DF7401";

sttlight.style.color = "#DF7401";

}

if(a.result.light > 600){

light.style.background = "#8A4B08";

sttlight.style.color = "#8A4B08";

}

if(a.result.dust <= 20){

dust.style.background = "#696969";

sttdust.style.color = "#696969";

}

if(a.result.dust > 20 && a.result.dust <= 40){

dust.style.background = "#2E2E2E";

sttdust.style.color = "#2E2E2E";

}

if(a.result.dust > 40){

dust.style.background = "#000000";

sttdust.style.color = "#000000";

}

$('#stt-temp').text(a.result.stttemp);

$('#stt-hum').text(a.result.stthum);

$('#stt-light').text(a.result.sttlight);

$('#stt-dust').text(a.result.sttdust);

$('#sensor-light').text(a.result.light);

$('#sensor-temp').text(a.result.temp);

$('#sensor-hum').text(a.result.hum);

$('#sensor-dust').text(a.result.dust);

$('#maxtemp').text(a.maxtemp.temp);

$('#mintemp').text(a.mintemp.temp);

$('#avgtemp').text(a.avgtemp.avgtemp);

$('#maxhum').text(a.maxhum.hum);

$('#minhum').text(a.minhum.hum);

$('#avghum').text(a.avghum.avghum);

$('#maxlight').text(a.maxlight.light);

$('#minlight').text(a.minlight.light);

$('#avglight').text(a.avglight.avglight);

$('#maxdust').text(a.maxdust.dust);

$('#mindust').text(a.mindust.dust);

$('#avgdust').text(a.avgdust.avgdust);

});

}, 500);

</script>

-Ở đây, các giá trị sẽ được đưa vào các thẻ có id tương ứng

ví dụ $('#stt-temp').text(a.result.stttemp);

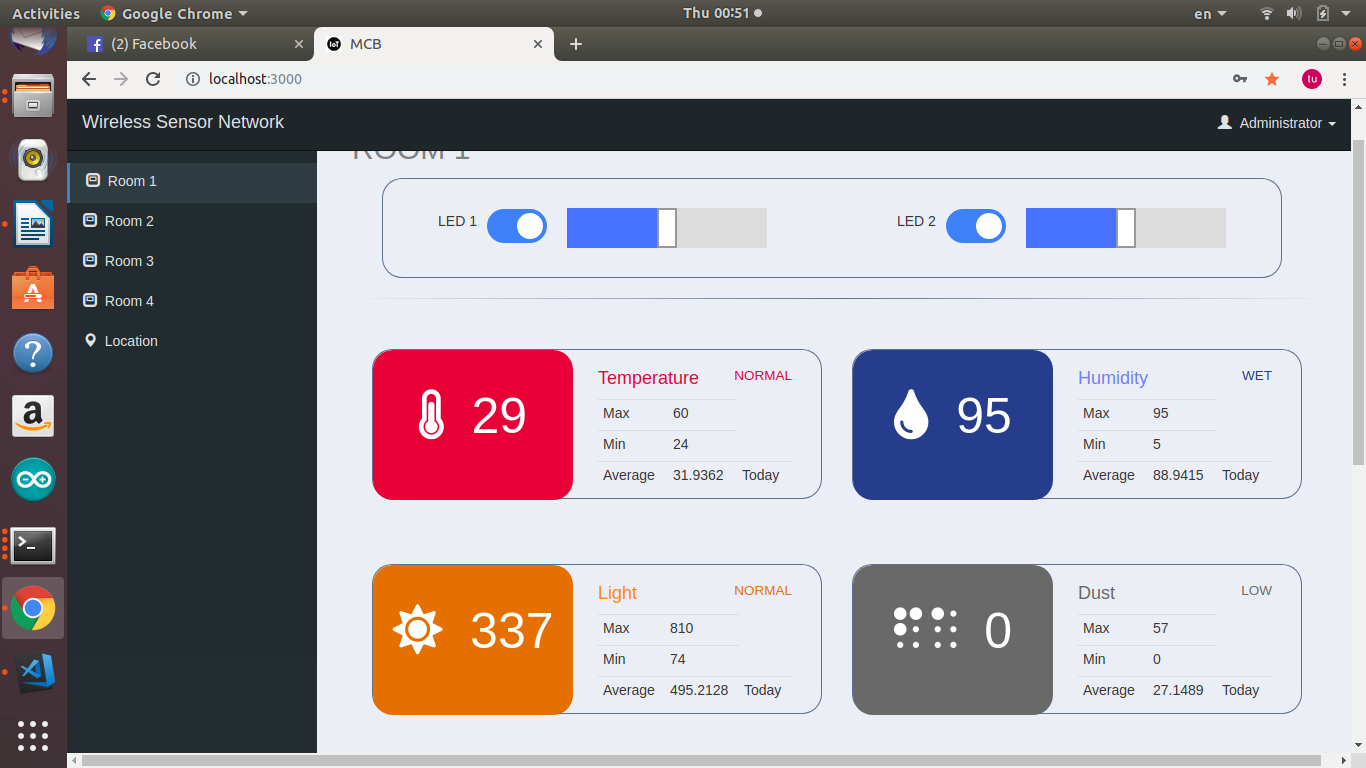
giá trị temp sẽ được đưa vào thẻ

<span id="stt-temp"></span>

-Màu sắc của data cũng được thay đổi bằng cách kiểm tra giá trị r thay đổi

thoi gian cập nhật ở đây là 500ms

-Sau khi update ta sẽ được:



-Update dữ liệu vào đồ thị, ta có thể thêm nhiều đường dữ liệu bằng cách thêm data(giống như ở trong phần comment)

<script>

window.onload = function () {

var dps = [];

var dps1 = [];

var dps2 = [];

var dps3 = [];// dataPoints

var r2dps = [];

var r2dps1 = [];

var r2dps2 = [];

var r2dps3 = [];

var r3dps = [];

var r3dps1 = [];

var r3dps2 = [];

var r3dps3 = [];

var r4dps = [];

var r4dps1 = [];

var r4dps2 = [];

var r4dps3 = [];

var chart = new CanvasJS.Chart("charttemp", {

backgroundColor: "#ECF0F5",

theme: "light2",

title :{

text: "Temperature"

},

axisY: {

includeZero: false

},

toolTip: {

shared: true

},

data: [{

type: "line",

dataPoints: dps,

xValueType: "dateTime",

xValueFormatString: "hh:mm:ss TT",

name: "Temp",

color: "#DC143C"}

// },

// {

// type: "line",

// dataPoints: dps1,

// xValueType: "dateTime",

// xValueFormatString: "hh:mm:ss TT",

// name: "Hum",

// color: "#6495ED"

// },

// {

// type: "line",

// dataPoints: dps2,

// xValueType: "dateTime",

// xValueFormatString: "hh:mm:ss TT",

// name: "Light",

// color: "#FF8C00"

// },

// {

// type: "line",

// dataPoints: dps3,

// xValueType: "dateTime",

// xValueFormatString: "hh:mm:ss TT",

// name: "Dust",

// color: "#696969"

// }

]

});

setInterval(function(){

$.getJSON( "/api/room1, function( a ) {

//time.setTime(time.getTime()+ 1000);

dps.push({

x: time.getTime(),

y: a.result.temp

});

dps1.push({

x: time.getTime(),

y: a.result.hum

});

dps2.push({

x: time.getTime(),

y: a.result.light

});

dps3.push({

x: time.getTime(),

y: a.result.dust

});

// if (dps.length > dataLength) {

// dps.shift();

// }

// if (dps1.length > dataLength) {

// dps1.shift();

// }

// if (dps2.length > dataLength) {

// dps2.shift();

// }

// if (dps3.length > dataLength) {

// dps3.shift();

// }

chart.render();

chart1.render();

chart2.render();

chart3.render();

});

}, 1000);

}

</script>

-Mỗi 1s ta lại yêu cầu api room1 để lấy giá trị r push giá trị đấy vào mảng để vẽ đồ thi

**4. Điều khiển LED**

-Ta tạo script để trả về giá trị của switch, input range để api lấy giữ liệu và lưu vào database.

<script>

$('#check-led1').click(function() {

var led1 = $(this).prop('checked');

if (led1) led1 = 1;

else led1 = 0;

$.post( "/api/led1", { signal: led1, token: token })

.done(function( data ) {

});

});

</script>

<script>

$('#control-led1').on('change', function() {

var led1 = $(this).val();

$.post( "/api/controlled1", { pwm: led1 , token: token})

.done(function( data ) {

});

console.log(led1)

});

</script>

**5.Pub sub dữ liệu giữa esp và máy tính để đọc cảm biến và điều khiển led**

**-** Ta sẽ pub dữ liệu dạng json và nhận sub giá trị để điều khiển led.

**Code esp:**

**[code]**

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#include <ArduinoJson.h>

#include "dht11.h"

#include "SharpGP2Y10.h"

#include "DHT.h"

//khai báo thư viện

// Update these with values suitable for your network.

const char\* ssid = "Delta";

const char\* password = "donttouch";

const char\* mqtt\_server = "172.20.10.9";

int pwm = 255; //biến lưu độ sáng

int pwm2 = 255;

int led1;

int led2;

#define mqtt\_port 1883

#define MQTT\_USER "tuananh"

#define MQTT\_PASSWORD "abc13579"

#define MQTT\_PUB "room1"

#define MQTT\_SUB "sttled"

#define DHTPIN 16

#define DHTTYPE DHT11

SharpGP2Y10 dustSensor(A0, 2);

WiFiClient wifiClient;

PubSubClient client(wifiClient);

DHT dht(DHTPIN, DHTTYPE);

int Read\_Dust(void){

pinMode(4, OUTPUT);

digitalWrite(4, LOW);

pinMode(5, OUTPUT);

digitalWrite(5, LOW);

int dust = dustSensor.getDustDensity()\*100;

return dust;

}

void setup\_wifi() {

pinMode(14, OUTPUT);

digitalWrite(14, HIGH);

pinMode(12, OUTPUT);

digitalWrite(12, HIGH);

delay(10);

// We start by connecting to a WiFi network

Serial.println();

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

randomSeed(micros());

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

}

void reconnect() {

// Loop until we're reconnected

while (!client.connected()) {

Serial.print("Attempting MQTT connection...");

// Create a random client ID

String clientId = "NodeMCUClient-";

clientId += String(random(0xffff), HEX);

// Attempt to connect

if (client.connect(clientId.c\_str(),MQTT\_USER,MQTT\_PASSWORD)) {

Serial.println("connected");

//Once connected, publish an announcement...

client.publish("test", "reconnected");

client.subscribe(MQTT\_SUB);

client.subscribe("led1pwm");

client.subscribe("led2pwm");

} else {

Serial.print("failed, rc=");

Serial.print(client.state());

Serial.println(" try again in 5 seconds");

// Wait 5 seconds before retrying

delay(5000);

}

}

}

//Sub dữ liệu

void callback(char\* topic, byte \*payload, unsigned int length) {

Serial.println("-------new message from broker-----");

Serial.print("channel:");

Serial.println(topic);

Serial.print("data:");

Serial.write(payload, length);

if(!strncmp((char \*)payload, "led1on", length)){

analogWrite(14, pwm); //nếu string thu được là led1on thì bật led 1

led1 = 1;

}

if(!strncmp((char \*)payload, "led1off", length)){

analogWrite(14, 0);

led1 = 0;

}

if(!strncmp((char \*)payload, "led2on", length)){

analogWrite(12, pwm2);

led2 = 1;

}

if(!strncmp((char \*)payload, "led2off", length)){

analogWrite(12, 0);

led2 = 0;

}

if(String(topic) == "led1pwm"){ // nếu topic là led1pwm thì

int i; //lưu giá trị pwm nhận được và điều chỉnh độ sáng

char message\_buff[10];

for (i = 0; i<length; i++) {

message\_buff[i] = payload[i];

}

message\_buff[i] = '\0';

const char \*p\_payload = message\_buff;

pwm = atoi(p\_payload);

if(led1 == 1){

analogWrite(14, pwm);

}

}

if(String(topic) == "led2pwm"){

int j;

char message\_buff1[10];

for (j = 0; j<length; j++) {

message\_buff1[j] = payload[j];

}

message\_buff1[j] = '\0';

const char \*p\_payload = message\_buff1;

pwm2 = atoi(p\_payload);

if(led2 == 1){

analogWrite(12, pwm2);

}

}

Serial.println();

}

void setup() {

Serial.begin(115200);

dht.begin();

Serial.setTimeout(500);// Set time out for

setup\_wifi();

client.setServer(mqtt\_server, mqtt\_port);

client.setCallback(callback);

reconnect();

}

void publishSerialData(char \*serialData){

if (!client.connected()) {

reconnect();

}

client.publish(MQTT\_PUB, serialData);

}

void loop() {

client.loop();

if (Serial.available() > 0) {

char bfr[101];

memset(bfr,0, 101);

Serial.readBytesUntil( '\n',bfr,100);

publishSerialData(bfr);

}

String temp = String(int(dht.readTemperature()));

String hum = String(int(dht.readHumidity()));

String light = String(Read\_Light());

String dust = String(Read\_Dust());

String payload = "{";

payload += "\"lat\": "; payload += "20.98082"; payload += ",";

payload += "\"lng\": "; payload += "105.78716"; payload += ",";

payload += "\"temp\": "; payload += temp; payload += ",";

payload += "\"hum\": "; payload += hum; payload += ",";

payload += "\"light\": "; payload += light; payload += ",";

payload += "\"dust\": "; payload += dust;

payload += "}";

char attributes[100];

payload.toCharArray( attributes, 100 );

client.publish( MQTT\_PUB, attributes );

delay(1000);

//pub dữ liệu dạng json

}

**Code Python:**

**+Sub dữ liệu:**

import paho.mqtt.client as mqtt

from SaveData import \*

import MySQLdb

Broker = "localhost"

Port = 1883

Wait = 45

Topic1 = "room1"

Topic2 = "room2"

Topic3 = "room3"

Topic4 = "room4"

def on\_connect(client, userdata, flags, rc):

if rc!= 0:

pass

print('Unable to connect to Broker...')

else:

print('Connected with Broker: ' + str(Broker))

client.subscribe(Topic1,0)

client.subscribe(Topic2,0)

client.subscribe(Topic3,0)

client.subscribe(Topic4,0)

def on\_message(client, userdat, msg):

print('Received data....')

print('Topic: ' + msg.topic)

if(msg.topic == 'room1'):

print('Data: ' + str(msg.payload))

Sensor(msg.payload)

if(msg.topic == 'room2'):

print('Data: ' + str(msg.payload))

Sensor2(msg.payload)

if(msg.topic == 'room3'):

print('Data: ' + str(msg.payload))

Sensor3(msg.payload)

if(msg.topic == 'room4'):

print('Data: ' + str(msg.payload))

Sensor4(msg.payload)

client = mqtt.Client()

client.username\_pw\_set(username = "tuananh", password = "abc13579")

client.on\_connect = on\_connect

client.on\_message = on\_message

client.connect(Broker, Port, Wait)

client.loop\_forever()

**+Module lưu dữ liệu:**

import MySQLdb

import json

def Sensor(jsonData):

data = json.loads(jsonData)

lat = str(data['lat'])

lng = str(data['lng'])

temp = int(data['temp'])

hum = int(data['hum'])

light = int(data['light'])

dust = int(data['dust'])

if(temp <= 20):

stttemp = "COLD"

if(temp > 20 and temp < 30):

stttemp = "NORMAL"

if(temp >= 30):

stttemp = "HOT"

if(hum <= 40):

stthum = "DRY"

if(hum > 40 and hum < 80):

stthum = "NORMAL"

if(hum >= 80):

stthum = "WET"

if(light <= 200):

sttlight = "SHINING"

if(light > 200 and light < 600):

sttlight = "NORMAL"

if(light >= 600):

sttlight = "DARK"

if(dust <= 20):

sttdust = "LOW"

if(dust > 20 and dust < 40):

sttdust = "NORMAL"

if(dust >= 40):

sttdust = "HIGH"

db = MySQLdb.connect("localhost","tuananh","abc13579","wordpress")

cursor = db.cursor()

sql = """insert into room1 (lat,lng,temp,hum,light,dust,stttemp,stthum,sttlight,sttdust,time) values ("%s","%s",%s,%s,%s,%s,"%s","%s","%s","%s",now())""" %(lat,lng,temp, hum, light, dust, stttemp,stthum,sttlight,sttdust)

cursor.execute(sql)

db.commit()

print('Data saved in room1 from database.')

db.close()

def Sensor2(jsonData):

data = json.loads(jsonData)

temp = int(data['temp'])

hum = int(data['hum'])

light = int(data['light'])

dust = int(data['dust'])

if(temp <= 20):

stttemp = "COLD"

if(temp > 20 and temp < 30):

stttemp = "NORMAL"

if(temp >= 30):

stttemp = "HOT"

if(hum <= 40):

stthum = "DRY"

if(hum > 40 and hum < 80):

stthum = "NORMAL"

if(hum >= 80):

stthum = "WET"

if(light <= 200):

sttlight = "SHINING"

if(light > 200 and light < 600):

sttlight = "NORMAL"

if(light >= 600):

sttlight = "DARK"

if(dust <= 20):

sttdust = "LOW"

if(dust > 20 and dust < 40):

sttdust = "NORMAL"

if(dust >= 40):

sttdust = "HIGH"

db = MySQLdb.connect("localhost","tuananh","abc13579","wordpress")

cursor = db.cursor()

sql = """insert into room2 (temp,hum,light,dust,stttemp,stthum,sttlight,sttdust,time) values (%s,%s,%s,%s,"%s","%s","%s","%s",now())""" %(temp, hum, light, dust, stttemp,stthum,sttlight,sttdust)

cursor.execute(sql)

db.commit()

print('Data saved in room2 from database.')

db.close()

def Sensor3(jsonData):

data = json.loads(jsonData)

temp = int(data['temp'])

hum = int(data['hum'])

light = int(data['light'])

dust = int(data['dust'])

if(temp <= 20):

stttemp = "COLD"

if(temp > 20 and temp < 30):

stttemp = "NORMAL"

if(temp >= 30):

stttemp = "HOT"

if(hum <= 40):

stthum = "DRY"

if(hum > 40 and hum < 80):

stthum = "NORMAL"

if(hum >= 80):

stthum = "WET"

if(light <= 200):

sttlight = "SHINING"

if(light > 200 and light < 600):

sttlight = "NORMAL"

if(light >= 600):

sttlight = "DARK"

if(dust <= 20):

sttdust = "LOW"

if(dust > 20 and dust < 40):

sttdust = "NORMAL"

if(dust >= 40):

sttdust = "HIGH"

db = MySQLdb.connect("localhost","tuananh","abc13579","wordpress")

cursor = db.cursor()

sql = """insert into room3 (temp,hum,light,dust,stttemp,stthum,sttlight,sttdust,time) values (%s,%s,%s,%s,"%s","%s","%s","%s",now())""" %(temp, hum, light, dust, stttemp,stthum,sttlight,sttdust)

cursor.execute(sql)

db.commit()

print('Data saved in room3 from database.')

db.close()

def Sensor4(jsonData):

data = json.loads(jsonData)

temp = int(data['temp'])

hum = int(data['hum'])

light = int(data['light'])

dust = int(data['dust'])

if(temp <= 20):

stttemp = "COLD"

if(temp > 20 and temp <= 30):

stttemp = "NORMAL"

if(temp > 30):

stttemp = "HOT"

if(hum <= 40):

stthum = "DRY"

if(hum > 40 and hum < 80):

stthum = "NORMAL"

if(hum >= 80):

stthum = "WET"

if(light <= 200):

sttlight = "SHINING"

if(light > 200 and light < 600):

sttlight = "NORMAL"

if(light >= 600):

sttlight = "DARK"

if(dust <= 20):

sttdust = "LOW"

if(dust > 20 and dust < 40):

sttdust = "NORMAL"

if(dust >= 40):

sttdust = "HIGH"

db = MySQLdb.connect("localhost","tuananh","abc13579","wordpress")

cursor = db.cursor()

sql = """insert into room4 (temp,hum,light,dust,stttemp,stthum,sttlight,sttdust,time) values (%s,%s,%s,%s,"%s","%s","%s","%s",now())""" %(temp, hum, light, dust, stttemp,stthum,sttlight,sttdust)

cursor.execute(sql)

db.commit()

print('Data saved in room4 from database.')

db.close()

**+Điều khiển led:**

import paho.mqtt.client as mqtt

import random

import json

from datetime import datetime

from time import sleep

import MySQLdb

Broker = "localhost"

Port = 1883

Wait = 65535

Topic = "sttled"

stt=3

kton = 1

ktoff = 1

kton1 = 1

ktoff1 = 1

pwmled1 = 0

pwmled2 = 0

def on\_connect(client, userdata, flags, rc):

if rc!= 0:

pass

print('Unable to connect to Broker...')

else:

print('Connected with Broker: ' + str(Broker))

def on\_publish(client, userdata, mid):

pass

def disconnect(client, userdata, rc):

if rc != 0:

pass

mqttc = mqtt.Client()

mqttc.username\_pw\_set(username = "tuananh", password = "abc13579")

mqttc.on\_connect = on\_connect

mqttc.disconnect = disconnect

mqttc.on\_publish = on\_publish

mqttc.connect(Broker, Port, Wait)

def pub2topic(topic, message):

mqttc.publish(topic,message)

print(('Published: ' + str(message) + ' ' + 'on MQTT topic: ' + str(topic)))

print('')

def pub\_data\_fake():

db = MySQLdb.connect("localhost","tuananh","abc13579","wordpress")

cursor = db.cursor()

sql = """select led1 from sttled1 where id = (select max(id) from sttled1)"""

cursor.execute(sql)

result = (cursor.fetchall())

stt = result[0][0]

return stt

def pub\_data\_fake2():

db = MySQLdb.connect("localhost","tuananh","abc13579","wordpress")

cursor = db.cursor()

sql = """select led2 from sttled2 where id = (select max(id) from sttled2)"""

cursor.execute(sql)

result = (cursor.fetchall())

stt = result[0][0]

return stt

def pub\_pwm1():

db = MySQLdb.connect("localhost","tuananh","abc13579","wordpress")

cursor = db.cursor()

sql = """select led1 from controlled1 where id = (select max(id) from controlled1)"""

cursor.execute(sql)

result = (cursor.fetchall())

stt = result[0][0]

return stt

def pub\_pwm2():

db = MySQLdb.connect("localhost","tuananh","abc13579","wordpress")

cursor = db.cursor()

sql = """select led2 from controlled2 where id = (select max(id) from controlled2)"""

cursor.execute(sql)

result = (cursor.fetchall())

stt = result[0][0]

return stt

while True:

if(pub\_data\_fake() == 1 and ktoff == 1):

mqttc.publish(Topic,"led1on")

ktoff = 0

kton = 1

print("LED1 ON")

if(pub\_data\_fake() == 0 and kton == 1):

mqttc.publish(Topic,"led1off")

ktoff = 1

kton = 0

print("LED1 OFF")

if(pub\_data\_fake2() == 1 and ktoff1 == 1):

mqttc.publish(Topic,"led2on")

ktoff1 = 0

kton1 = 1

print("LED2 ON")

if(pub\_data\_fake2() == 0 and kton1 == 1):

mqttc.publish(Topic,"led2off")

ktoff1 = 1

kton1 = 0

print("LED2 OFF")

if(pub\_pwm1() != pwmled1):

pwmled1 = pub\_pwm1()

mqttc.publish("led1pwm",str(pwmled1))

if(pub\_pwm2() != pwmled2):

pwmled2 = pub\_pwm2()

mqttc.publish("led2pwm",str(pwmled2))

**+Pub dữ liệu fake:**

import paho.mqtt.client as mqtt

import random

import json

from datetime import datetime

from time import sleep

Broker = "localhost"

Port = 1883

Wait = 45

Topic = "room2"

def on\_connect(client, userdata, flags, rc):

if rc!= 0:

pass

print('Unable to connect to Broker...')

else:

print('Connected with Broker: ' + str(Broker))

def on\_publish(client, userdata, mid):

pass

def disconnect(client, userdata, rc):

if rc != 0:

pass

mqttc = mqtt.Client()

mqttc.username\_pw\_set(username = "tuananh", password = "abc13579")

mqttc.on\_connect = on\_connect

mqttc.disconnect = disconnect

mqttc.on\_publish = on\_publish

mqttc.connect(Broker, Port, Wait)

def pub2topic(topic, message):

mqttc.publish(topic,message)

print(('Published: ' + str(message) + ' ' + 'on MQTT topic: ' + str(topic)))

print('')

def pub\_data\_fake(nameSensor):

hum = int(random.uniform(50,100))

temp = int(random.uniform(20,40))

light = int(random.uniform(0,1023))

dust = int(random.uniform(0,99))

sensor\_data = {}

sensor\_data['hum'] = hum

sensor\_data['temp'] = temp

sensor\_data['light'] = light

sensor\_data['dust'] = dust

sensor\_json\_data = json.dumps(sensor\_data)

print("Publishing data fake from %s: " %nameSensor)

pub2topic(Topic,sensor\_json\_data)

while True:

pub\_data\_fake("room2")

sleep(1)

**6.Tạo map**

**Script:**

<script>

// setInterval(function(){

var a;

var b;

var target1;

target1 = L.latLng('20.98082','105.78716');

var element = document.getElementById('osm-map');

element.style = 'height:500px; width:970px;display: flex;align-items: center;justify-content: center;';

var map = L.map(element);

L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png', {

attribution: '&copy; <a href="http://osm.org/copyright">OpenStreetMap</a> contributors'

}).addTo(map);

$.getJSON( "/api/room1?token="+token, function( result ) {

a = String(result.lat);

b = String(result.lng);

target1 = L.latLng(a,b);

});

setTimeout(function() {

// map.setView(target1, 16);

//L.marker(target1,{title:'Room 1'}).addTo(map).bindPopup("Sensor1").openPopup();

}, 1000);

var target2 = L.latLng('20.98045','105.78971');

var target3 = L.latLng('20.98148','105.79170');

var target4 = L.latLng('20.98270','105.78998');

map.setView(target2, 16);

map.setView(target3, 16);

map.setView(target4, 16);

L.marker(target1,{title:'Room 1'}).addTo(map).bindPopup("Sensor1").openPopup();

L.marker(target2,{title:'Room 2'}).addTo(map).bindPopup("Sensor2");

L.marker(target3,{title:'Room 3'}).addTo(map).bindPopup("Sensor3");

L.marker(target4,{title:'Room 4'}).addTo(map).bindPopup("Sensor4");

// }, 1000);

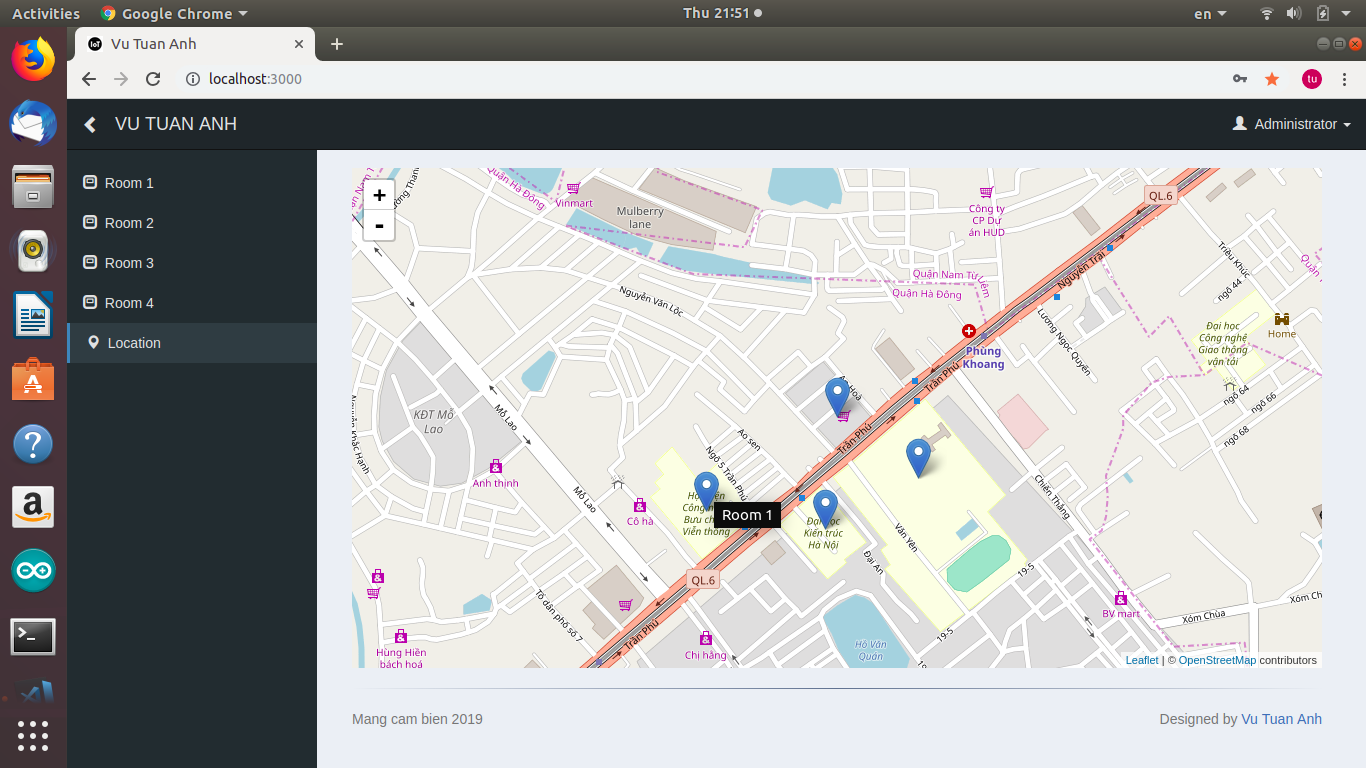
</script>

Thay đổi địa chỉ thì lên openstreetmap search địa chỉ đấy và lấy giá trị lat,lng thay vào code.

**Html:**

<div id="osm-map"></div>

**Ta được:**

****