

# BaiBaoCaoCuoiKy\_THCNTT2

*by* Bảo Phạm Thái

---

**Submission date:** 07-Dec-2023 10:02PM (UTC+0700)

**Submission ID:** 2251327318

**File name:** BaoCaoCuoiKy\_PhamThaiBao\_22114919.docx (2.49M)

**Word count:** 1739

**Character count:** 7815



ĐẠI HỌC  
HOA SEN

“BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC HOA SEN

KHOA CÔNG NGHỆ THÔNG TIN”

**BÁO CÁO MÔN HỌC**

**ĐỀ TÀI:**

**Thiết Bị Báo Cháy**

**Môn học**

: Thực hành Công nghệ thông tin 2

**Lớp môn học**

: IT 204DV01 - 0100

**HỌ TÊN**

**PHẠM THÁI BẢO - 22114919**

**Giảng viên hướng dẫn : ThS. Trần Thị Trương Thi**

“TP.HCM, tháng 12 năm 2023”

Submission date:

Submiss ID:

File name:

Word count:

Character count:

# <sup>1</sup> MỤC LỤC

<b>MỤC LỤC .....</b>	<b>1</b>
<b>DANH MỤC HÌNH ẢNH .....</b>	<b>2</b>
<b>LỜI MỞ ĐẦU .....</b>	<b>4</b>
<b>NỘI DUNG .....</b>	<b>5</b>
<b>I.Thiết bị.....</b>	<b>5</b>
1. Yêu cầu thiết bị .....	5
2. Cách cắm thiết bị .....	7
3. Yêu cầu phần mềm và tải thư viện .....	8
4. Code .....	9
<b>II. Back-end.....</b>	<b>15</b>
1) TẢI NODEJS NẾU CHƯA CÓ .....	15
2) TẠO PROJECT BẰNG NODEJS .....	15
3) TẠO PROJECT VÀ DATABASE TRÊN MYSQL .....	17
4) CODE .....	19
5) TEST API BẰNG POSTMAN .....	21
<b>III. Front-end .....</b>	<b>22</b>
<b>LỜI KẾT .....</b>	<b>41</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>42</b>

## DANH MỤC HÌNH ẢNH

Hình 1: Wemos D1 R1 .....	5
Hình 2: DHT11 .....	5
Hình 3: Relay 5VDC .....	6
Hình 4: Còi báo -Buzzer .....	6
Hình 5: Máy bơm nước.....	7
Hình 6: Cách cắm dây Wemos và DHT11 và Relay .....	7
Hình 7: Thêm link driver của Wemos D1 R1 .....	8
Hình 8: Tìm board phù hợp với Wemos .....	8
Hình 9: Check Version của Node.....	15
Hình 10: Phần mềm Visual Studio Code.....	15
Hình 11: Tạo Folder back-end .....	16
Hình 12: Các thư viện cần thiết của Folder back-end .....	16
Hình 13: Thư viện để liên kết back-end với database.....	17
Hình 14: Giao diện của MySQL Workbench .....	17
Hình 15: Tạo database bằng New Queryry trên MySQL .....	18
Hình 16: Viết code để tạo database và dữ liệu.....	18
Hình 17: Dữ liệu trong database .....	19
Hình 18: Tạo file backend.js trong folder back-end .....	19
Hình 19: Chạy file backend.js và dữ liệu xuất ra từ database .....	22
Hình 20: Test các phương thức trên Postman .....	22
Hình 21: Tạo folder Front-end.....	22
Hình 22: Giao diện của phần mềm ReactJS .....	23
Hình 23: Tải và cài các thư viện cần thiết cho folder Frontend.....	23

**Hình 24: Giao diện của ứng dụng báo cháy .....38**

## **LỜI MỞ ĐẦU**

“Lời nói đầu tiên em muốn cảm ơn giảng viên Trần Thị Trường Thi vì đã cho em cơ hội để thử sức với hệ thống này. Tuy nó chỉ là “hệ thống nhỏ” nhưng đã cho em rất nhiều kinh nghiệm thực tiễn.

Đồ án về ứng dụng quản lý, cảnh báo cháy khi trong nhà có xuất hiện lửa, thiết bị sẽ báo và chữa cháy ngay lập tức để đảm bảo an toàn cho người và tài sản. Và đi xa hơn dù đây chỉ là mô hình nhỏ nhưng có thể phát triển thành một ứng dụng với quy mô lớn nếu muốn.

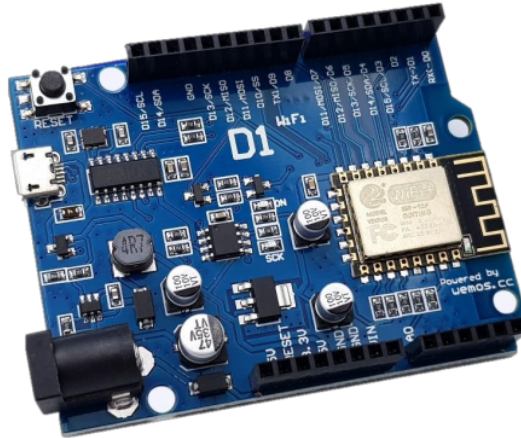
Em chân thành cảm ơn cô”

## <sup>1</sup> NỘI DUNG

### I. Thiết bị

#### 1. Yêu cầu thiết bị

Wemos D1 R1, DHT11



<sup>1</sup>  
*Hình 1: Wemos D1 R1*



*Hình 2: DHT11*





*Hình 6: máy bơm nước*

## 2. Cách cắm thiết bị

*Hình 7: Cách cắm dây Wemos với DHT11 và Relay*

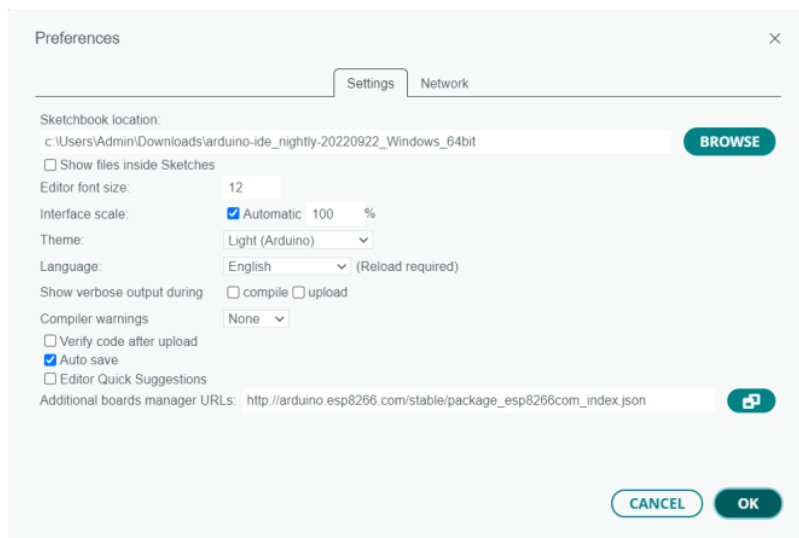
## 3. Yêu cầu chương trình, thư viện

Arduino IDE

Tải thư viện board của wemos D1 R1:

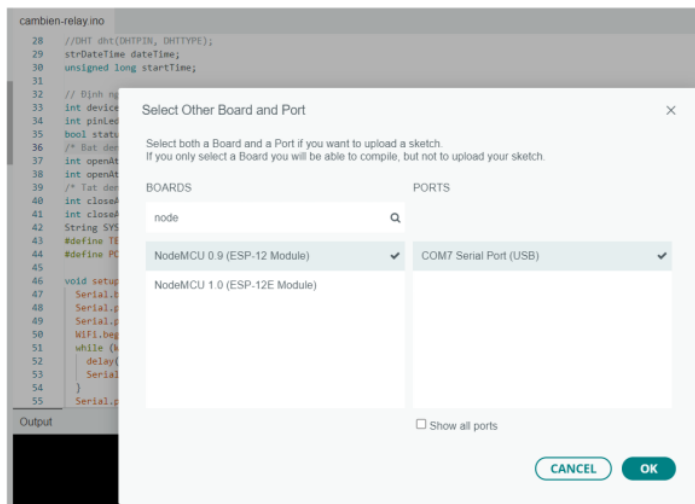
[https://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](https://arduino.esp8266.com/stable/package_esp8266com_index.json)

Ta lấy link đó, sau đó vào arduino IDE, “File -> References -> dán link vào Additional boards managers URLs”. Sau đó nhấn OK



**Hình 8: Thêm link driver của Wemos D1 R1**

Với file DHT11, ta tìm board



**Hình 9: Tìm board và cổng phù hợp với Wemos**

#### 4. Code

1

## II. Back-end

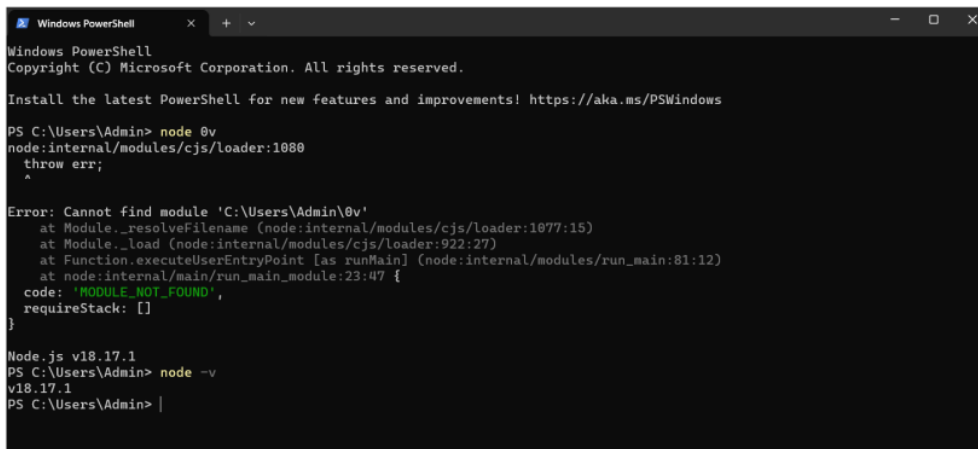
### 1) TẢI NODEJS NẾU CHƯA CÓ

Trước khi ta vào phần back-end hoặc front-end, ta cần phải tải NodeJS trước.

Ta lên "<https://nodejs.org/en/download/>" để tải về

Khi tải về xong, ta mở file vừa tải lên và cho chạy cài đặt

“Để kiểm tra ta đã có nodejs hay chưa, ta nhấn nút window và gõ PowerShell. Sau đó ta gõ “node -v” để kiểm tra. Nếu có hình dưới đây thì ta đã cài đặt thành công”



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Admin> node -v
node:internal/modules/cjs/loader:1080
  throw err;
  ^

Error: Cannot find module 'C:\Users\Admin\@v'
    at Module._resolveFilename (node:internal/modules/cjs/loader:1077:15)
    at Module._load (node:internal/modules/cjs/loader:922:27)
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:81:12)
    at node:internal/main/run_main_module:23:47 {
  code: 'MODULE_NOT_FOUND',
  requireStack: []
}

Node.js v18.17.1
PS C:\Users\Admin> node -v
v18.17.1
PS C:\Users\Admin> |
```

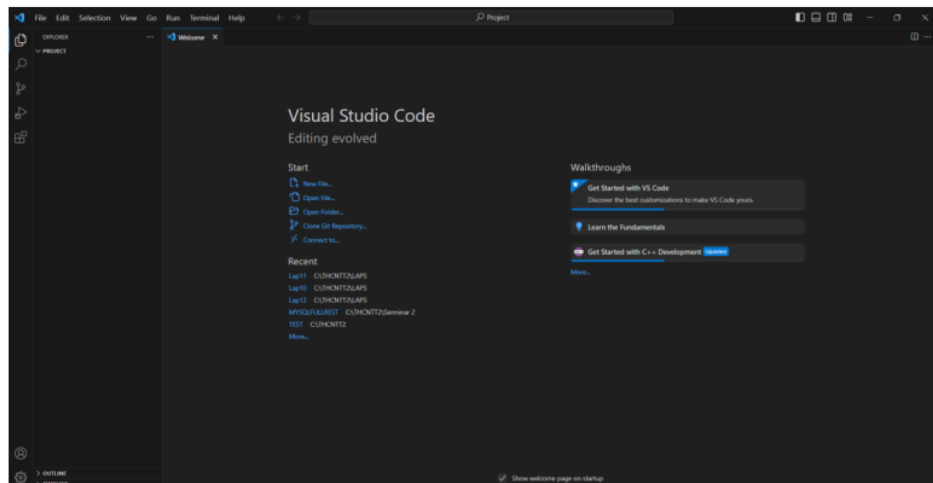
**Hình 10: Check Version của Node**

Trong bài này, ta sử dụng NodeJS cho phần back-end và MySQL cho phần database

### 2) TẠO “PROJECT” BẰNG PHẦN MỀM VISUAL STUDIO CODE

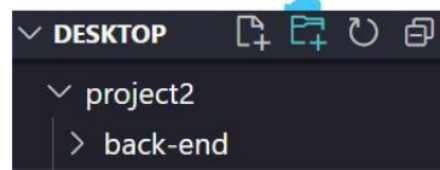
Đầu tiên ta cần mở phần mềm Visual Studio để tiến hành tạo thư mục để chứa server

Sau đó ta chọn chỗ để lưu thư mục



**Hình 11: Phần mềm Visual Studio Code**

Để tạo được file back-end ta vào phần mềm và tạo folder back-end



**Hình 12: Tạo Folder back-end**

Tiếp đến để có thể sử dụng được project này ta tiến hành cài các thư viện cần thiết

“npm install express”, “npm install cors”, “npm install body-parser”, “npm install nodemon”

```

PS C:\Users\ADMIN\OneDrive\Desktop\project2\back-end> npm install express

up to date, audited 119 packages in 9s

15 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\ADMIN\OneDrive\Desktop\project2\back-end> npm install cors

up to date, audited 119 packages in 2s

15 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\ADMIN\OneDrive\Desktop\project2\back-end> npm install body-parser

up to date, audited 119 packages in 2s

15 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

```

```

PS C:\Users\ADMIN\OneDrive\Desktop\project2\back-end> npm install nodemon

up to date, audited 119 packages in 1s

15 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

```

**Hình 13: Các thư viện cần thiết của Folder back-end**

2 Tiếp theo để sử dụng được MySQL thì ta cần cài thêm thư viện bằng lệnh “npm install mysql” hoặc “npm install mysql2” tùy phiên bản.

```

PS C:\Users\ADMIN\OneDrive\Desktop\project2\back-end> npm install nodemon

up to date, audited 119 packages in 1s

15 packages are looking for funding
  run `npm fund` for details

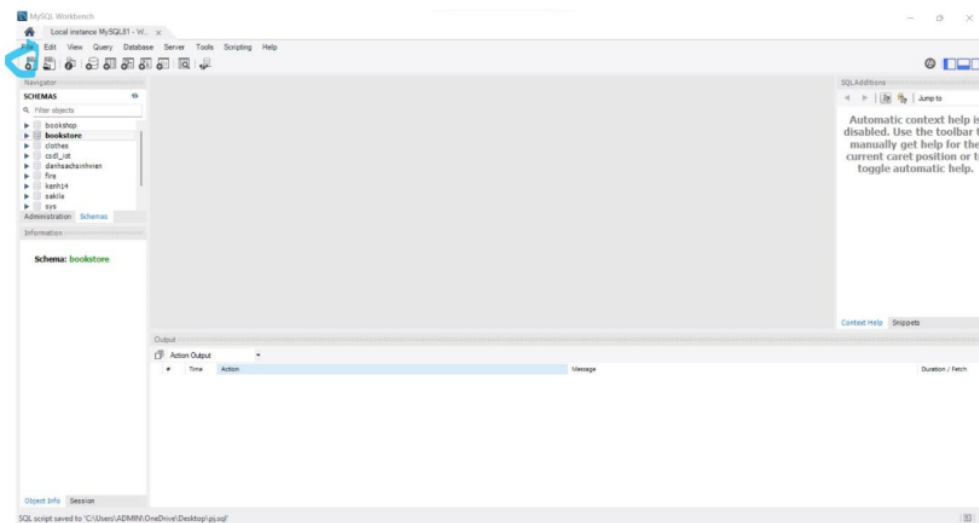
found 0 vulnerabilities

```

**Hình 14: Thư viện để liên kết back-end với database**

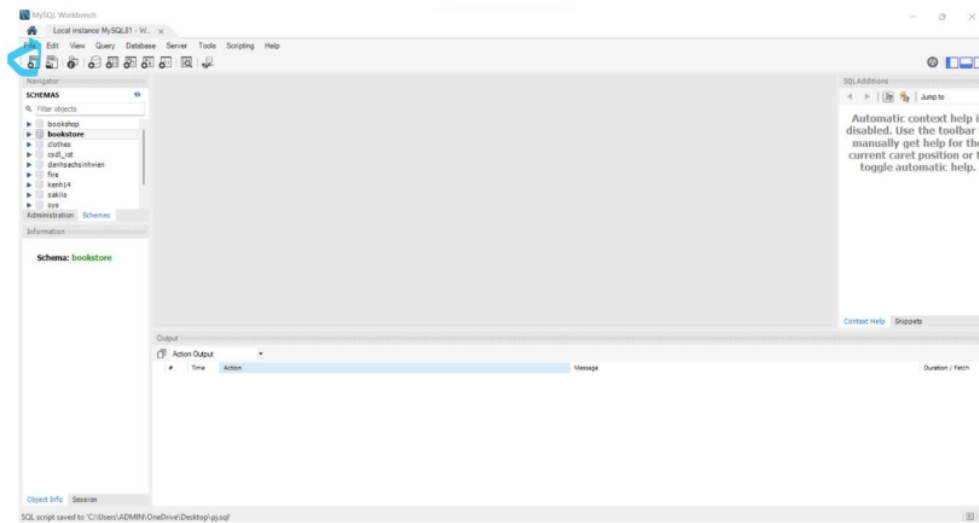
### 3) TẠO PROJECT VÀ DATABASE TRÊN MYSQL

Đầu tiên ta cần truy cập vào phần mềm MySQL Workbench và có thể vào truy cập web này để tải phần mềm <https://dev.mysql.com/downloads/mysql/>  
Sau khi truy cập vào phần mềm sẽ có giao diện như thế này



**Hình 15: Giao diện của MySQL Workbench**

Để tạo database mới mọi người nhấn vào “New Query” để tiến hành viết chương trình tạo database và dữ liệu



**Hình 16: Tạo database bằng New Query trên MySQL**

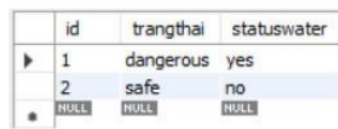
Ta tiến hành viết code để tạo ra database



```
Query 1 x  tbl_nhietdo  tbl_nhietdo  tbl_nhietdo  tbl_nhietdo  tbl_status  tbl_status
Limit to 1000 rows
1 • use csdl_iot;
2 • create table tbl_status
3   ( id int primary key auto_increment, trangthai varchar(255), statuswater varchar(255));
4 • insert into tbl_status(trangthai,statuswater) values( 'dangerous', 'yes' ) ;
5 • insert into tbl_status(trangthai,statuswater) values( 'safe', 'no' ) ;
6 • select * from tbl_status
```

**Hình 17: Viết code để tạo database và dữ liệu**

Sau khi viết xong mọi người tiến hành chạy để có thể thấy được dữ liệu trong database của mình



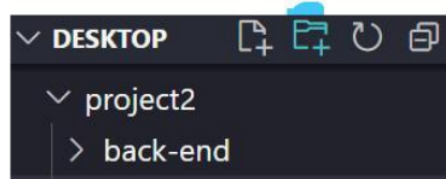
	id	trangthai	statuswater
▶	1	dangerous	yes
	2	safe	no
•	NULL	NULL	NULL

**Hình 18: Dữ liệu trong database**

#### 4) CODE

Đầu tiên ta cần phải tạo thêm folder để code backend

Ta tạo file “backend.js” bằng cách tạo trực tiếp trong folder



**Hình 19: Tạo file backend.js trong folder back-end**

Ta vào file backend.js và tiến hành ghi code

```
“var express = require('express');
var bodyParser = require("body-parser");
var cors = require('cors');
var mysql = require('mysql'); //npm install mysql
var app = express();
app.use(cors());
app.use(bodyParser.json());
});
```



```

// connect
var con = mysql.createConnection({
  host: "localhost",
  port: "3306",
  user: "root",
  password: "123",
  insecureAuth : true,
  database: "csdl_iot"
con.connect(function(err) {
  if (err) {
    console.error('Lỗi khi kết nối cơ sở dữ liệu:', err);
  } else {
    console.log('Kết nối cơ sở dữ liệu thành công'); } }));

//viết api
app.get('/nhietdo', function (req, res) {
var sql = "SELECT * FROM tbl_status ORDER BY id DESC LIMIT 1";
  con.query(sql, function(err, results) {
    if (err) throw err;
    res.send(results); }); })
app.post('/add', (req, res) => {
  const { status,statuswater } = req.body;
  var sql = "INSERT INTO tbl_status (status,statuswater ) VALUES (?,?)";
  con.query(sql, [status,statuswater], function(err, results) { if (err) throw err;
    var sqlGetTemperature = "Select * from tbl_status "
    con.query(sqlGetTemperature,function(err,tbl_status){ if(err) throw err;
    res.send(tbl_status); } ) }); });

// server đang chạy ở cổng
var server = app.listen(8081, function () {
var host = server.address().address

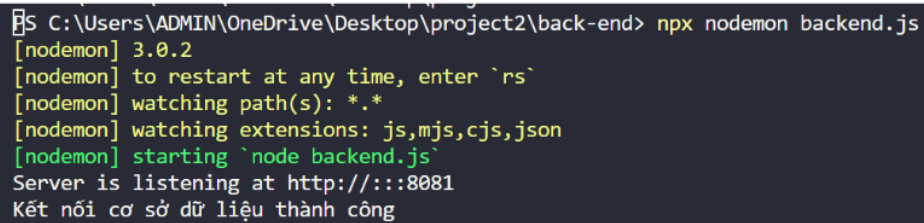
```

```
var port = server.address().port
```

```
console.log("Server is listening at http://%s:%s", host, port) })"
```

## 5) TEST API MÌNH VIẾT BẰNG POSTMAN

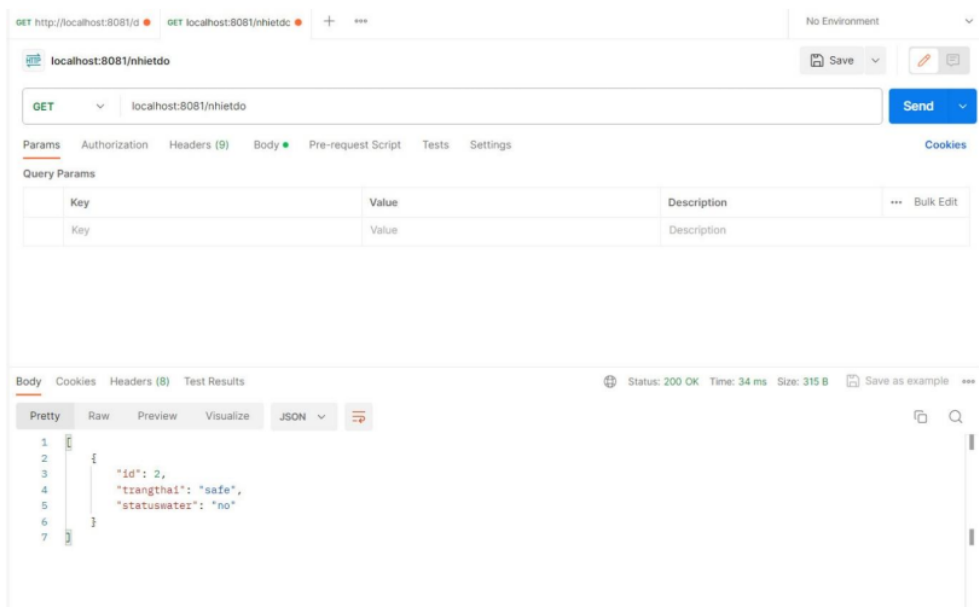
Ta khởi động server bằng cú pháp “npx nodemon backend.js” ở trong PowerShell



```
PS C:\Users\ADMIN\OneDrive\Desktop\project2\back-end> npx nodemon backend.js
[nodemon] 3.0.2
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node backend.js`
Server is listening at http://:::8081
Kết nối cơ sở dữ liệu thành công
```

*Hình 20: Chạy file nhietdo.js và dữ liệu xuất ra từ database*

Cuối cùng, ta test các phương thức như get, post,...



**Hình 21: Test các phương thức trên Postman**

### III. Front-end

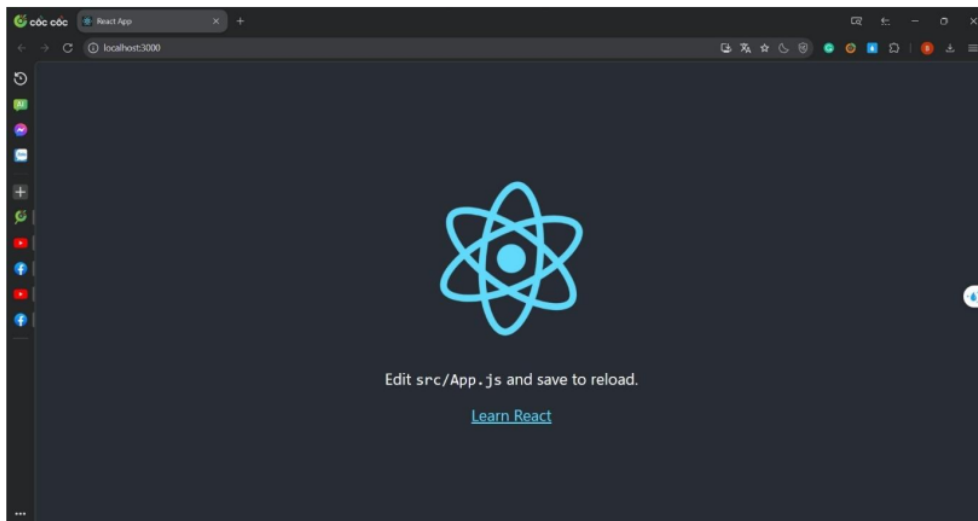
Còn Front-end, ta dùng ReactJS để khởi tạo

Đầu tiên ta truy cập vào folder chứa file back-end và cài đặt reactJS vào folder Front-end bằng lệnh

“npm install -g create-react-app”

Sau đó ta tạo project bằng lệnh “npx create-react-app <projectname>”

Để khởi động file Front-end ta vào file Front-end vừa tạo và chạy lệnh “npm start” và sẽ hiện giao diện như thế này



**Hình 22: Giao diện của phần mềm ReactJS**

Và để có thể chạy được và liên kết với back-end và database ta sẽ tải thêm những thư viện cần thiết như

“npm install axios”, “npm install express”, “npm body-parser”

```
PS C:\Users\ADMIN\OneDrive\Desktop\project2\front-end> npm install axios
up to date, audited 119 packages in 2s
15 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
PS C:\Users\ADMIN\OneDrive\Desktop\project2\front-end> npm install express
up to date, audited 119 packages in 1s
15 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
PS C:\Users\ADMIN\OneDrive\Desktop\project2\front-end> npm install body-parser
up to date, audited 119 packages in 2s
15 packages are looking for funding
  run `npm fund` for details
```

**Hình 23: Tải và cài các thư viện cần thiết cho folder Frontend**

Ta vô file **App.js** để tiến hành ghi code để nhận dữ liệu từ back-end và database

```
“import './App.css';
import React, { useState,useEffect } from "react";
import axios from 'axios';
```

```

function App() {
  const [items,setItems] = useState([])

  useEffect(()=>{
    axios.get("http://localhost:8081/nhietdo")

    .then((response) => {
      console.log(response.data)
      //luu
      setItems(response.data)
    })
  },[]) // Bỏ data ra khỏi mảng phụ thuộc

  return (
    <div className='container' >
      <div className='row col-md-12 text-center'>
        <div className='wrap'>
          <h1>Bảng thông tin trạng thái hiện tại</h1>
        </div>
        <div className='body'>
          {items.map((items)=> (
            <div className='item'>
              <table className='item-info'>
                <tr>
                  <th>Status</th>
                  <th>Statuswater</th>
                </tr>
                <tr>
                  <th>{items.trangthai}</th>

```

```

        <th>{items.statuswater}</th>
      </tr>
    </table>
  </div>
)}}
</div>

</div>
</div>
})

```

export default App;”

Và sau đó ta sẽ trang trí dữ liệu database của mình trong file **App.css**

```

“.wrap {
  text-align: center;
  background-color: grey;
  min-height: 50vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  font-size: calc(10px + 2vmin);
  color: turquoise;
  table-layout: fixed;
  border: 2px;
}
.body{
  background-color: white;
  min-height: 50vh;
  display: flex;

```

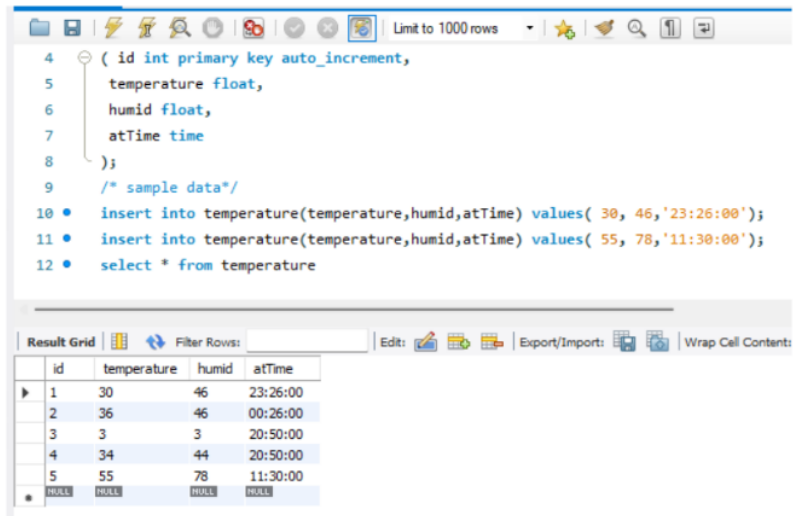
```
flex-direction: column;
align-items: center;
justify-content: center;
font-size: calc(10px + 2vmin);
color: turquoise;
table-layout: fixed;
border: 5px solid black;

}
th, td {
padding: 8px;
text-align: center;
border-bottom: 1px solid #ddd;
}”
```

Và sau khi tạo xong file Front-end và trang trí ta sẽ có website như thế này

**Hình 26: Giao diện của ứng dụng cảnh báo lửa**

Hoặc add trực tiếp dữ liệu trên database



The screenshot shows a database management interface. The top section displays SQL code for creating a table and inserting data. The bottom section shows a 'Result Grid' with 5 rows of data.

```
4 ( id int primary key auto_increment,  
5   temperature float,  
6   humid float,  
7   atTime time  
8 );  
9 /* sample data*/  
10 • insert into temperature(temperature,humid,atTime) values( 30, 46,'23:26:00');  
11 • insert into temperature(temperature,humid,atTime) values( 55, 78,'11:30:00');  
12 • select * from temperature
```

	id	temperature	humid	atTime
▶	1	30	46	23:26:00
	2	36	46	00:26:00
	3	3	3	20:50:00
	4	34	44	20:50:00
	5	55	78	11:30:00
*	NULL	NULL	NULL	NULL

**Hình 28: Thêm thông tin cảm biến từ database**



## LỜI KẾT

“Trong xã hội phát triển công nghệ 4.0 hiện nay chúng ta không thể không thừa nhận tầm quan trọng của các thiết bị công nghệ”. “Ngày có nhiều thứ mới mẽ xuất hiện nhằm mục đích chính là phục vụ cho đời sống của con người. Như những gì đã nói nên em đã quyết định thực thành lắp ráp những mô hình nhỏ từ những gì đã học được. Thông qua bài thực hành này bản thân em đã gom góp được nhiều bài học, em có thể tự tay làm các thiết bị nhỏ có ích cho gia đình, bạn bè của mình.

Em cảm ơn cô vì đã đồng hành với em suốt 15 tuần qua, em mong rằng có thể trong tương lai em sẽ đồng hành với cô trong những môn học khác hoặc có thể làm việc với nhau

Em chân thành cảm ơn cô Thi”

## <sup>1</sup> **TÀI LIỆU THAM KHẢO**

- 1) [https://arduinogetstarted.com/tutorials/arduino-relay#google\\_vignette](https://arduinogetstarted.com/tutorials/arduino-relay#google_vignette)

# BaiBaoCaoCuoiky\_THCNTT2

## ORIGINALITY REPORT

15%  
SIMILARITY INDEX

3%  
INTERNET SOURCES

0%  
PUBLICATIONS

15%  
STUDENT PAPERS

## PRIMARY SOURCES

1 Submitted to Hoa Sen University 15%  
Student Paper

2 hocwebchuan.com 1%  
Internet Source

Exclude quotes On  
Exclude bibliography On

Exclude matches Off