

TranHuuHai - 22013127 - THCNTT3 - HK2233

by Hải Trần Hữu

Submission date: 02-Jul-2023 06:32PM (UTC+0700)

Submission ID: 1986145635

File name: Ba_iBa_oCa_o.docx (3.34M)

Word count: 3060

Character count: 11731

**BỘ GIÁO DỤC ĐÀO TẠO
ĐẠI HỌC HOA SEN**

**BÁO CÁO MÔN THỰC HÀNH CÔNG NGHỆ THÔNG TIN 3
ĐỒ ÁN CUỐI KỲ**

TP. HỒ CHÍ MINH, 2023

BỘ GIÁO DỤC ĐÀO TẠO
ĐẠI HỌC HOA SEN

BÁO CÁO MÔN THỰC HÀNH CÔNG NGHỆ THÔNG TIN 3
ĐỒ ÁN CUỐI KỲ

Môn: Thực Hành Công Nghệ Thông Tin 3

Lớp: IT205DV01 – 0100

DANH SÁCH THÀNH VIÊN

Trần Hữu Hải – 22013127

TP.HỒ CHÍ MINH, 2023

1
MỤC LỤC

MỤC LỤC	1
LỜI CẢM ƠN	3
NỘI DUNG	4
1. FrontEnd	4
1.1 Giới thiệu	4
1.1.1 ReactJS	4
1.1.2 Lấy dữ liệu từ BackEnd	5
1.1.3 CSS các thông số	10
1.1.4 React Native	14
1.1.5 Lấy dữ liệu từ BackEnd và thông số CSS	14
Giao diện 1:	14
Giao diện 2:	18
2. BackEnd	25
2.1 Giới thiệu về Microsoft SQL SERVER	25
2.1.1 SQL là gì?	25
2.1.2 Ưu và nhược điểm của SQL	25
a. Ưu điểm của SQL	25
b. Nhược điểm của SQL	25
2.1.3 Microsoft SQL Server là gì?	25
2.2 Code Database hệ thống đo nhiệt độ, độ ẩm	26
Hướng dẫn tải Microsoft SQL Server	26
2.3 Giới thiệu về NodeJS	30
2.3.1 Node JS là gì?	30
2.3.2 Ưu và nhược điểm của NodeJS	30
a. Ưu điểm	30
b. Nhược điểm	30
2.4 Code hệ thống đo nhiệt độ, độ ẩm bằng NodeJS	30
Các thư viện cần thiết	30
3. IoT	37
3.1 Giới thiệu sơ lược về Arduino IDE	37

3.2 Mạch và sensor sử dụng trong hệ thống FireWarning	38
3.2.1 Mạch Wemos D1	38
3.2.2 Sensor đo độ ẩm và nhiệt độ DHT1	41
3.2.3 Cảm biến gas MQ-2.....	41
3.2.4 Bluetooth ZS040.....	42
3.3 Nối dây và nạp chương trình.	43

LỜI CẢM ƠN

“Em xin gửi lời cảm ơn chân thành đến Trường Đại học Hoa Sen, môn Thực hành công nghệ thông tin 3 đã tạo điều kiện thuận lợi cho em học tập và hoàn thành đề tài nghiên cứu này. Đặc biệt, em xin bày tỏ sự biết ơn, lời cảm ơn đến cô Trần Thị Trường Thi đã dày công truyền đạt kiến thức và hướng dẫn em trong quá trình làm bài. Trong thời gian tham gia lớp học Thực hành công nghệ thông tin 3 của cô, em đã có thêm cho mình nhiều kiến thức bổ ích, tinh thần học tập hiệu quả, nghiêm túc.”

“Em đã cố gắng vận dụng những kiến thức đã học được trong học kỳ qua để hoàn thành bài Đồ án cuối kỳ này. Tuy nhiên, do vốn kiến thức của em còn hạn chế và không có nhiều kinh nghiệm thực tiễn nên khó tránh khỏi những thiếu sót trong quá trình nghiên cứu và trình bày. Rất mong cô xem xét và góp ý để bài Đồ án cuối kỳ của em được hoàn thiện hơn.”

“Em xin trân trọng cảm ơn cô !”

NỘI DUNG

1

1. FrontEnd

1.1 Giới thiệu

1.1.1 ReactJS

“ReactJS là một Opensource có thể giúp người dùng xây dựng giao diện website một cách đơn giản, nhanh chóng và hạn chế lỗi trong quá trình code. Dễ dàng lấy dữ liệu từ BackEnd và cập nhật website sau khi lấy dữ liệu một cách liên tục.”

“Đối với Project có sử dụng Sensor cảm biến thì ReactJS là một lựa chọn thích hợp, có thể cập nhật liên tục dữ liệu từ Sensor lên thẳng Website một cách nhanh chóng mà không cần reset lại trang.”

“Khi sử dụng ReactJS, ta cần phải cài đặt các thư viện sau: ”

- “Thư viện ReactJS: bản thân ReactJS là một thư viện của Javascript, để khởi tạo một dự án ReactJS ta phải dùng câu lệnh “npx create-react-app <tên thư mục chứa dự án>””
- “Thư viện Axios: Thư viện này giúp người code có thể dễ dàng lấy dữ liệu từ BackEnd, câu lệnh cài thư viện “npm install axios” ”
- “Thư viện React-Bootstrap: là thư viện bootstrap được dùng trong ReactJS, giúp người code dễ dàng thiết kế website một cách nhanh chóng, câu lệnh cài thư viện “npm install react-bootstrap bootstrap” ”

1.1.2 Lấy dữ liệu từ BackEnd

```
You, 6 seconds ago | 2 authors (You and others)
1 import "./App.css";
2 import { useEffect, useState } from "react";
3 import axios from "axios";
4 import Formtable from "./components/Formtable";
5
6 function App() {
7   const [addSection, setAddSection] = useState(false);
8   const [editSection, setEditSection] = useState(false);
9   const [formData, setFormData] = useState({
10     id: "",
11     name: "",
12     imageURL: "",
13   });
14   const [formDataEdit, setFormDataEdit] = useState({
15     id: "",
16     name: "",
17     imageURL: "",
18   });
19   const [dataList, setDataList] = useState([]);
20
21   const handleOnChange = (e) => {
22     const { value, name } = e.target;
23     setFormData((preve) => {
24       return {
25         ...preve,
26         [name]: value,
27       };
28     });
29   };
30 }
```

```
31 | const handleSubmit = async (e) => {
32 |   e.preventDefault();
33 |   try {
34 |     const response = await axios.post(
35 |       "http://localhost:5555/addRooms",
36 |       formData
37 |     );
38 |     console.log(response.data);
39 |
40 |     if (response.data === "Room added successfully") {
41 |       setAddSection(false);
42 |       alert("Room added successfully");
43 |       getFetchData();
44 |       setFormData({
45 |         name: "",
46 |         imageURL: "",
47 |       });
48 |     } else {
49 |       alert("Error adding room");
50 |     }
51 |   } catch (error) {
52 |     console.error(error);
53 |     alert("Error adding room");
54 |   }
55 | };
56 | const getFetchData = async () => {
57 |   try {
58 |     const response = await axios.get("http://localhost:5555/getRooms");
59 |     console.log(response.data);
60 |     setDataTable(response.data);
61 |   } catch (error) {
62 |     console.error(error);
63 |   }
64 | };
```

You, 23 seconds ago • Uncommitted changes

```
useEffect(() => {
  getFetchData();
}, []);

const handleDelete = async (id) => {
  try {
    const response = await fetch(`http://localhost:5555/deleteRooms/${id}`, {
      method: 'DELETE',
    });

    if (!response.ok) {
      throw new Error('Error deleting room');
    }

    const data = await response.json();
    getFetchData();
    alert(data.message);
  } catch (error) {
    console.error("Error deleting room:", error);
    alert("Error deleting room");
  }
};
```

```
89  const handleUpdate = async (e, id) => {
90    e.preventDefault();
91
92    try {
93      if (formDataEdit.name === "" || id === "") {
94        // Xử lý thông báo lỗi hoặc sẽ không gọi axios.put()
95      } else {
96        const response = await axios.put("http://localhost:5555/updateRooms/" + id, formDataEdit);
97        console.log(response.data) ;
98        if(response.data==="Room updated successfully")
99        {
100          setEditSection(false);
101          alert("Room updated successfully");
102          getFetchData();
103        }
104      } else{
105        alert("Error updating room");
106      }
107    }catch (error) {
108      console.error(error);
109      alert("Error updating room");
110      // Xử lý lỗi tại đây
111    }
112  };
113
```

```
115  const handleEditOnChange = (e) => {
116    const { value, name } = e.target;
117    setFormDataEdit((prev) => ({
118      ...prev,
119      [name]: value,
120    }));
121  };
122  const handleEdit = (el) => {
123    setFormDataEdit(el);
124    setEditSection(true);
125  };
126  return (
127    <>
128      <div className="container">
129        <button className="btn btn-add" onClick={() => setAddSection(true)}>
130          Add
131        </button>
132
133        {addSection && (
134          <Formtable
135            handleSubmit={handleSubmit}
136            handleChange={handleOnChange}
137            handleClose={() => setAddSection(false)}
138            rest={formData}
139          />
140        )}
141        {editSection && (
142          <Formtable
143            handleSubmit={handleUpdate}
144            handleChange={handleEditOnChange}
145            handleClose={() => setEditSection(false)}
146            rest={formDataEdit}
147          />
148        )}

```

```
150     <div className="tableContainer">
151       <table>
152         <thead>
153           <tr>
154             <th>ID</th>
155             <th>Name</th>
156             <th>imageURL</th>
157             <th></th>
158           </tr>
159         </thead>
160         <tbody>
161           {dataList[0] ? (
162             dataList.map((el) => {
163               console.log(el);
164               return (
165                 <tr key={el.id}>
166                   <td>{el.id}</td>
167                   <td>{el.name}</td>
168                   <td>{el.imageURL}</td>
169                   <td>
170                     <button className="btn btn-edit" onClick={() => handleEdit(el)}>Edit</button>
171                     <button className="btn btn-delete" onClick={() => handleDelete(el.id)}>Delete</button>
172                   </td>
173                 </tr>
174               );
175             )
176           ) : (
177             <tr>
178               <td style={{ textAlign: "center" }}>No data</td>
179             </tr>
180           )
181         </tbody>
182       </table>
```

```
183       </div>
184     </div>
185   </>
186 };
187 }
188
189 export default App;
190
```

1.1.3 CSS các thông số

```
You, 3 seconds ago | 2 authors (Amit prajapati and others)
1 *{           Amit prajapati, 2 weeks ago • initial comm
2   box-sizing: border-box;
3 }
4 body{
5   background-color: #f5f6ff;
6 }
7 .container{
8   padding: 10px;
9   /* background-color: red; */
10  max-width: fit-content;
11  margin: 50px auto;
12 }
13 .btn{
14   border: none;
15   padding: 7px 15px;
16   font-size: 18px;
17   border-radius: 5px;
18   cursor: pointer;
19 }
20 .btn-add{
21   background: #194064;
22   color: white;
23 }
24 }
```

```
.addContainer{
  position: absolute;
  background-color: #rgba(0,0,0,0.2);
  left: 0;
  right: 0;
  bottom: 0;
  top: 0;
  display: flex;
  justify-content: center;
  align-items: center;
}

.addContainer form{
  width: 420px;
  background-color: white;
  display: flex;
  flex-direction: column;
  padding: 50px 40px;
  box-shadow: 5px 5px 10px #rgba(0,0,0,0.2) ;
  border-radius: 5px;
}

.addContainer form label{
  font-size: 18px;
}

.addContainer form input{
  font-size: 18px;
  padding: 5px;
  margin-top: 5px;
  margin-bottom: 10px;
}
```

```
55 .addContainer form button{
56   background-color: #royalblue;
57   color: white;
58   font-weight: 500;
59   margin-top: 20px;
60 }
61 .addContainer form .close-btn{
62   margin-left: auto;
63   font-size: 18px;
64   width: 20px;
65   height: 20px;
66   display: flex;
67   justify-content: center;
68   align-items: center;
69   border-radius: 20px;
70   border: 1px solid #000;
71   cursor: pointer;
72 }
73 .tableContainer{
74   margin-top: 50px;
75   box-shadow: 5px 5px 10px #rgba(0,0,0,0.2);
76   border-radius: 5px;
77 }
78 .tableContainer table thead tr{
79   background-color: #f0eaea;
80 }
81 .tableContainer table thead tr th,
82 .tableContainer table tbody tr td{
83   min-width: 200px;
84   padding: 7px;
85   text-align: center;
86 }
```

```
87 .tableContainer table tbody tr{  
88 | border : 1px solid #c9c5c2;  
89 }  
90 .tableContainer table tbody tr td{  
91 | border-bottom: 1px solid #c9c5c2;  
92 }  
93 .btn-edit,  
94 .btn-delete{  
95 | font-size: 16px;  
96 | padding: 5px 10px;  
97 | margin:0px 10px;  
98 }  
99 .btn-edit{  
100 | background-color: #e2f413;  
101 }  
102 .btn-delete{  
103 | background-color: #fa1717;  
104 | color:#white;  
105 }
```

The screenshot shows a user interface for managing room data. At the top left is a blue 'Add' button. Below it is a table with three rows of data:

ID	Name	imageURL	
1	Living Room	living.png	<button>Edit</button> <button>Delete</button>
2	Kitchen	kitchen.png	<button>Edit</button> <button>Delete</button>
3	BedRoom	bedroom.png	<button>Edit</button> <button>Delete</button>

1.1.4 React Native

“React Native là một framework phát triển ứng dụng di động sử dụng ngôn ngữ JavaScript và sử dụng cùng một cơ sở mã nguồn với ReactJS. Nó cho phép bạn xây dựng ứng dụng di động nhanh chóng và hiệu quả bằng cách sử dụng các thành phần UI tái sử dụng được xây dựng trên nền tảng JavaScript.”

“Một số điểm nổi bật của React Native:

- “Cross-platform: React Native cho phép bạn xây dựng ứng dụng cho cả iOS và Android từ một cơ sở mã nguồn duy nhất. Điều này giúp tiết kiệm thời gian và công sức lập trình so với việc phát triển riêng biệt trên hai nền tảng.”
- “Tích hợp với các thư viện và khung công cụ: React Native đi kèm với một loạt các thư viện và khung công cụ sẵn có, giúp phát triển ứng dụng dễ dàng hơn. Ví dụ, Expo cung cấp một tập hợp các API và công cụ phát triển để tăng cường trải nghiệm phát triển.”
- “Hiệu suất cao: React Native sử dụng kiến trúc “Native Bridge” để giao tiếp giữa mã JavaScript và các thành phần UI native. Điều này cho phép ứng dụng chạy mượt mà và có hiệu suất tương tự như các ứng dụng được viết bằng Java hoặc Objective-C.”
- “Tái sử dụng mã nguồn: Với React Native, bạn có thể tái sử dụng lại mã nguồn giữa các nền tảng khác nhau. Điều này giúp giảm thiểu công sức lập trình và duy trì ứng dụng.”
- “Cộng đồng mạnh mẽ: React Native có một cộng đồng lớn và tích cực, với nhiều tài liệu, thư viện và ví dụ sẵn có. Bạn có thể tìm thấy hỗ trợ từ cộng đồng khi gặp phải vấn đề hoặc cần kiến thức mới.”

“React Native đã được sử dụng rộng rãi trong việc phát triển ứng dụng di động, cho phép các nhà phát triển xây dựng ứng dụng mạnh mẽ, đáng tin cậy và đa nền tảng với ít công sức hơn.”

1.1.5 Lấy dữ liệu từ BackEnd và thông số CSS

Giao diện 1:

```
1 import { StyleSheet, Text, View, Switch, TouchableOpacity } from 'react-native';
2 import Ionicons from '@expo/vector-icons/Ionicons';
3 import { Image } from 'react-native'
4 import React, { useEffect, useState } from 'react';
5 import { useNavigation } from '@react-navigation/native';
6
7 const ShowListRoom = () => {
8   const [room, setRooms] = useState([])
9   useEffect(() => {
10     console.log("here")
11     getRooms().then((data) => setRooms(data))
12   }, [])
13
14   const getRooms = async () => {
15     const result = await fetch("http://192.168.1.7:5555/getRooms");
16     const data = await result.json();
17     console.log('data', data)
18     return data;
19   }
20 }
```

```
21  const navigation = useNavigation();
22  return (
23    <View style={styles.container}>
24      <Vtew style={styles.header}>
25        <Ionicons name="home-outline" size={70} color="orange" />
26        <Text style={styles.txtTitle}>Smart Home</Text>
27    </View>
28    <View style={styles.main}>
29      <View style={styles.row}>
30        {room.map((item) => (
31          <TouchableOpacity key={item.id} onPress={() => navigation.navigate('List Devices', { roomId: item.id })} style={styles.btn} >
32            <View style={styles.item}>
33              <Image source={require('../assets/kitchen.png')} style={{ width: 120, height: 120 }} />
34              <View style={styles.item1}>
35                <Text style={styles.txtTitle}> {item.name} </Text>
36                <Text style={styles.txtSubTitle}> Lights on </Text>
37                <Text style={styles.txtSubTitle}> Fans on </Text>
38                <Text style={styles.txtSubTitle}> TVs on </Text>
39              </View>
40            </View>
41          </TouchableOpacity>
42        )
43      ))
44    </View>
45  </View>
46)
47)
48}
49export default function ListRooms() {
50  return (
51    <ShowListRoom />
52  );
53}
```

```
55 v const styles = StyleSheet.create({
56   v container: {
57     flex: 1,
58     justifyContent: 'center',
59     alignItems: 'center',
60     backgroundColor: '#ECEFF1',
61   },
62   v row: {
63     flexDirection: 'row',
64     flexWrap: 'wrap',
65     alignItems: "center",
66     justifyContent: 'center',
67     flexGrow: 1,
68   },
69   v header: {
70     flexDirection: 'row',
71     alignItems: 'center',
72     marginBottom: 20,
73   },
74   v txtTitle: {
75     fontSize: 30,
76     fontWeight: 'bold',
77     marginLeft: 10,
78   },
79   v main: {
80     flex: 1,
81     justifyContent: 'center',
82     alignItems: 'center',
83     marginRight: 10,
84     justifyContent: 'space-between',
85   },
}
```

```
86  ↴  btn: {
87    |   marginVertical: 10,
88  },
89  ↴  item: {
90    |   flexDirection: 'row',
91    |   alignItems: 'center',
92    |   backgroundColor: 'lightblue',
93    |   padding: 10,
94    |   margin: 10,
95    |   borderRadius: 10,
96  },
97  ↴  image: {
98    |   width: 120,
99    |   height: 120,
100 },
101 ↴  item1: {
102   |   marginLeft: 10,
103 },
104 ↴  txtSubTitle: {
105   |   fontSize: 16,
106   |   marginLeft: 10,
107 },
108 ↴  footer: {
109   |   marginTop: 20,
110 },
111});
```

List Rooms

 Smart Home

**Living Room**
Lights on
Fans on
TVs on

**Kitchen**
Lights on
Fans on
TVs on

**BedRoom**
Lights on
Fans on
TVs on

[▼ List Rooms](#) [▼ List Devices](#)

Giao diện 2:

```
1 import React, { useEffect, useState } from 'react';
2 import { StyleSheet, Text, View, Switch, TouchableOpacity } from 'react-native';
3 import { Image } from 'react-native'
4 import Ionicons from '@expo/vector-icons/Ionicons';
5 import axios from 'axios';
6 import { useRoute } from '@react-navigation/native';
7 import AsyncStorage from "@react-native-async-storage/async-storage";
8 import init from "react_native_mqtt";
9
10 //MQTT
11 init({
12   size: 10000,
13   storageBackend: AsyncStorage,
14   defaultExpires: 1000 * 3600 * 24,
15   enableCache: true,
16   sync: {},
17 });
18 const options = {
19   host: "broker.emqx.io",
20   port: 8083,
21   path: "/thh1206_thh1206",
22   id: "id_" + parseInt(Math.random() * 100000),
23 };
24 const client = new Paho.MQTT.Client(options.host, options.port, options.path);
25 //
26 const OnOffLed = ({ navigation }) => {
27   const route = useRoute();
28   const roomId = route.params?.roomId;
29   const [devices, setDevices] = useState([]);
30   const [isLightEnabled, setLightValue] = useState([]);
31   const [isDevicesLoaded, setIsDevicesLoaded] = useState(false);
32   const toggleLightSwitch = (index, value, topic) => {
33     let initialStatus;
34     console.log("Topic toggle " + topic)
35     const updatedLightValues = [...isLightEnabled];
36     updatedLightValues[index] = value;
37     if (value == false) {
38       console.log("Switch off")
39       publishTopic("0", topic)
40     } else {
41       console.log("Switch on")
42       publishTopic("1", topic)
43     }
44     setLightValue(updatedLightValues);
45   };
46   const getDevices = async () => {
47     const result = await fetch(`http://192.168.1.7:5555/getDevices/${roomId}`);
48     const data = await result.json();
49     console.log('data', data);
50     return data;
51 }
```

```
52  useEffect(() => {
53    console.log(roomId);
54    console.log("here");
55  <div>v</div>  getDevices().then((data) => {
56    setDevices(data);
57    const initialToggleValues = data.map((device) => device.status === 1);
58    setLightValue(initialToggleValues);
59    setIsDevicesLoaded(true);
60  });
61 }, []);
62 //----- MQTT
63 const [msg, setMsg] = useState("No message");
64 const [statusLed, setStatusLed] = useState("off");
65 <div>v</div>  useEffect(() => {
66  <div>v</div>  if (isDevicesLoaded) {
67    connect();
68    // step 3 handling when message arrived
69    client.onMessageArrived = onMessageArrived;
70  }
71  //step 1 connect Mqtt broker
72 }, [isDevicesLoaded]);
```

```
73  const connect = () => {
74    client.connect({
75      onSuccess: () => [
76        console.log("connect MQTT broker ok!"),
77        //step 2 subscribe topic
78        subscribeTopic() // ledstatus
79      ],
80      useSSL: false,
81      timeout: 5,
82      onFailure: () => {
83        console.log("connect fail");
84        connect();
85        console.log("reconnect ...");
86      },
87    });
88  };
89  const publishTopic = (deviceStatus, topic) => {
90    console.log("Publish topic " + topic)
91    const s = '{"status":"' + deviceStatus + '","mqtt_topic_sub":"' + topic + '"}';
92    var message = new Paho.MQTT.Message(s);
93    message.destinationName = topic;
94    client.send(message);
95  };
96  const subscribeTopic = () => {
97    console.log("asd");
98    devices.forEach(device => {
99      console.log(device.mqtt_topic_sub)
100     client.subscribe(device.mqtt_topic_sub, { qos: 0 });
101     console.log(device.mqtt_topic_sub)
102   })
103 };
104
```

```

  105  const sendData = async (topic, status) => {
  106    if (topic != "undefined") {
  107      try {
  108        console.log("topic :", topic);
  109        console.log("topic :", status);
  110        const response = await axios.put(`http://192.168.1.7:5555/updateDevice/${topic}`, { status: status });
  111        console.log('PUT request successful');
  112        console.log(response.data);
  113      } catch (error) {
  114        console.error('Error making PUT request:', error);
  115      }
  116    }
  117  };
  118  const onMessageArrived = async (message) => {
  119    console.log("onMessageArrived:" + message.payloadString);
  120    setMsg(message.payloadString);
  121    const jsondata = JSON.parse(message.payloadString);
  122    const topic = jsondata.topic;
  123    const status = jsondata.status;
  124
  125    sendData(topic, status);
  126    setStatusLed(jsondata.status);
  127  };
  128  const handleButtonOn = async () => {
  129    //connect();
  130    console.log("turn on led...");
  131    publishTopic("on");
  132  };
  133  const handleButtonOff = async () => {
  134    console.log("turn off led...");
  135    publishTopic("off");
  136  };
  137
  138  return (
  139    <View style={styles.container}>
  140      <View style={styles.header}>
  141        <Ionicons name="home-outline" size={64} color="orange"/>
  142        <Text style={styles.txtTitle}>Smart Home</Text>
  143      </View>
  144      <View style={styles.main}>
  145        {devices.map((item, index) => [
  146          item.name === 'led' ?
  147            <View style={styles.itemDevice} key={item.name}>
  148              <Image source={{ uri: require("../assets/bulb.jpg") }} style={{ width: 50, height: 50 }} />
  149              <View style={styles.itemDevice1}>
  150                <Text style={styles.txtItem}>{item.name}</Text>
  151                <Switch style={styles.toggleBtn} value={isLightEnabled[index]} onValueChange={(value) => toggleLightSwitch(index, value, item.mqtt_topic_pn)} />
  152              </View>
  153            </View>
  154          :
  155            <View style={styles.itemDevice} key={item.name}>
  156              <Image source={{ uri: require("../assets/bulb.jpg") }} style={{ width: 50, height: 50 }} />
  157              <View style={styles.itemDevice1}>
  158                <Text style={styles.txtItem}>{item.name}</Text>
  159                <Text style={styles.txtItem}>{item.status} °C</Text>
  160              </View>
  161            </View>
  162          ]
  163        )}
  164      </View>
  165      <View style={styles.footer}>
  166        <Text style={styles.txtSubTitle}> THCN113 </Text>
  167      </View>
  168    </View>
  169  )

```

```
165      </View>
166      <View style={styles.footer}>
167        <Text style={styles.txtSubTitle}> THCNTT3 </Text>
168      </View>
169    </View>
170  )
171 }
172 export default function ListDevices() {
173   return (
174     <OnOffLed />
175   );
176 }
177
178 const styles = StyleSheet.create({
179   container: {
180     flex: 1,
181     backgroundColor: '#ECEFF1',
182     justifyContent: 'flex-start',
183   },
184   txtHelloWorld: {
185     fontSize: 50
186   },
187   header: {
188     flex: 1,
189     flexDirection: 'row',
190     alignItems: 'center',
191     justifyContent: 'center',
192     marginTop: 30,
193   },
```

```
194  },
195  footer: {
196    display: 'flex',
197    alignItems: 'flex-end',
198    alignItems: 'center',
199    paddingBottom: 20
200  },
201  txtTitle: {
202    fontSize: 40,
203    fontWeight: 'bold',
204    color: 'black',
205    marginLeft: 10,
206  },
207  txtSubTitle: {
208    fontSize: 20,
209    fontWeight: 'bold',
210    color: '#424242',
211  },
212  txtItem: {
213    fontSize: 20,
214    fontWeight: 'bold',
215    color: '#424242',
216    flex: 1,
217  },
218  main: {
219    flex: 3,
220    alignItems: 'center',
221  },
```

```
222     itemDevice: {
223       borderRadius: 5,
224       height: 60,
225       width: '100%',
226       backgroundColor: '#FFC107',
227       marginTop: 15,
228       flexDirection: "row",
229       alignItems: 'center',
230       paddingHorizontal: 10,
231     },
232
233     itemDevice1: {
234       height: 60,
235       width: '90%',
236       marginTop: 10,
237       flexDirection: "row",
238       alignItems: 'center',
239     },
240
241     btnText: {
242       color: 'auto',
243       fontWeight: 'bold'
244     },
245
246     toggleBtn: {
247       width: 40,
248       height: 25,
249     }
250   });
251
```

2. BackEnd

2.1 Giới thiệu về Microsoft SQL SERVER

2.1.1 SQL là gì?

“Nó được viết tắt bởi cụm từ Structured Query Language, là ngôn ngữ truy vấn có cấu trúc. Đây là một ngôn ngữ, là một tập hợp các lệnh tương tác với cơ sở dữ liệu. Dùng để lưu trữ, thao tác và truy xuất dữ liệu được lưu trữ trong một cơ sở dữ liệu quan hệ. Trong thực tế, SQL là ngôn ngữ chuẩn được sử dụng cho hệ cơ sở dữ liệu quan hệ. Tất cả các hệ thống quản lý cơ sở dữ liệu như MS Access, Oracle, MySQL, Postgres và SQL Server... đều sử dụng ngôn ngữ này làm cơ sở dữ liệu chuẩn.”

2.1.2 Ưu và nhược điểm của SQL

a. “Ưu điểm của SQL”

- “**Xử lý truy vấn nhanh hơn:** Một lượng lớn dữ liệu được truy xuất nhanh chóng và hiệu quả. Các thao tác như chèn, xóa, thao tác dữ liệu cũng được thực hiện gần như là không tồn thời gian.”
- “**Không có kỹ năng mã hóa:** Để truy xuất dữ liệu, không cần số lượng lớn dòng mã. Tất cả các từ khóa cơ bản như INTO, SELECT, INSERT, UPDATE,... đều được dùng và các quy tắc cú pháp trong SQL cũng không phức tạp, điều này làm cho nó trở thành một ngôn ngữ thân thiện với người dùng.”
- “**Ngôn ngữ chuẩn hóa:** Do lịch sử thành lập lâu đời trong nhiều năm và có hệ thống tài liệu hướng dẫn đầy đủ, SQL cung cấp cho người dùng một nền tảng thống nhất trên toàn thế giới.”
- “**Portable:** Được dùng trong PC, máy chủ, máy tính xách tay, bất kể hệ điều hành nào nó cũng có thể được nhúng với các ứng dụng khác.”
- “**Ngôn ngữ dễ hiểu:** Có thể nhận được câu trả lời cho câu truy vấn phức tạp chỉ trong vài giây.”

b. “Nhược điểm của SQL”

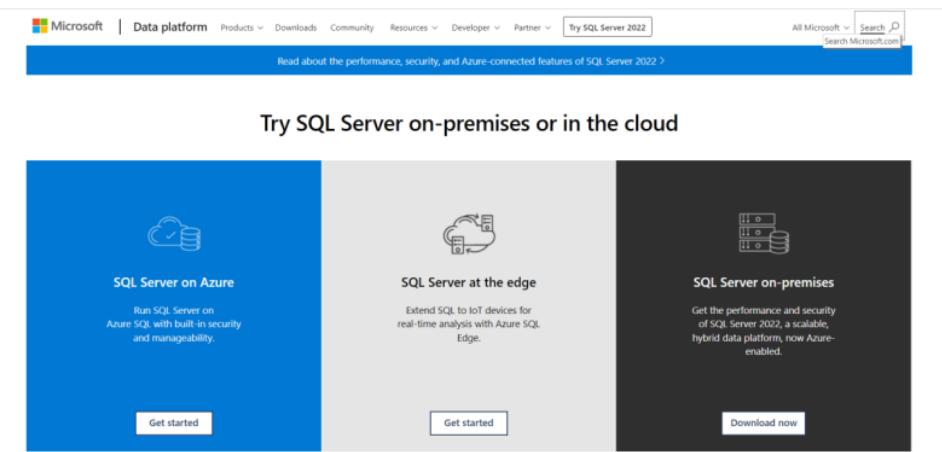
- “**Giao diện phức tạp:** SQL có một giao diện vô cùng phức tạp, khiến một số người dùng cảm thấy khó khăn trong việc xử lý cơ sở dữ liệu.”
- “**Chi phí:** Một số phiên bản đắt tiền, khiến các lập trình viên không thể truy cập được”
- “**Chỉ được kiểm soát một phần:** Do các nguyên tắc nghiệp vụ ẩn, cơ sở dữ liệu không thể được kiểm soát hoàn toàn.”

2.1.3 Microsoft SQL Server là gì?

“SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System (RDBMS)) sử dụng câu lệnh SQL (**Transact-SQL**) để trao đổi dữ liệu giữa máy Client và máy cài SQL Server. Một RDBMS bao gồm databases, database engine và các ứng dụng dùng để quản lý dữ liệu và các bộ phận khác nhau trong RDBMS. SQL Server được phát triển và tiếp thị bởi Microsoft.”

2.2 Code Database hệ thống đo nhiệt độ, độ ẩm

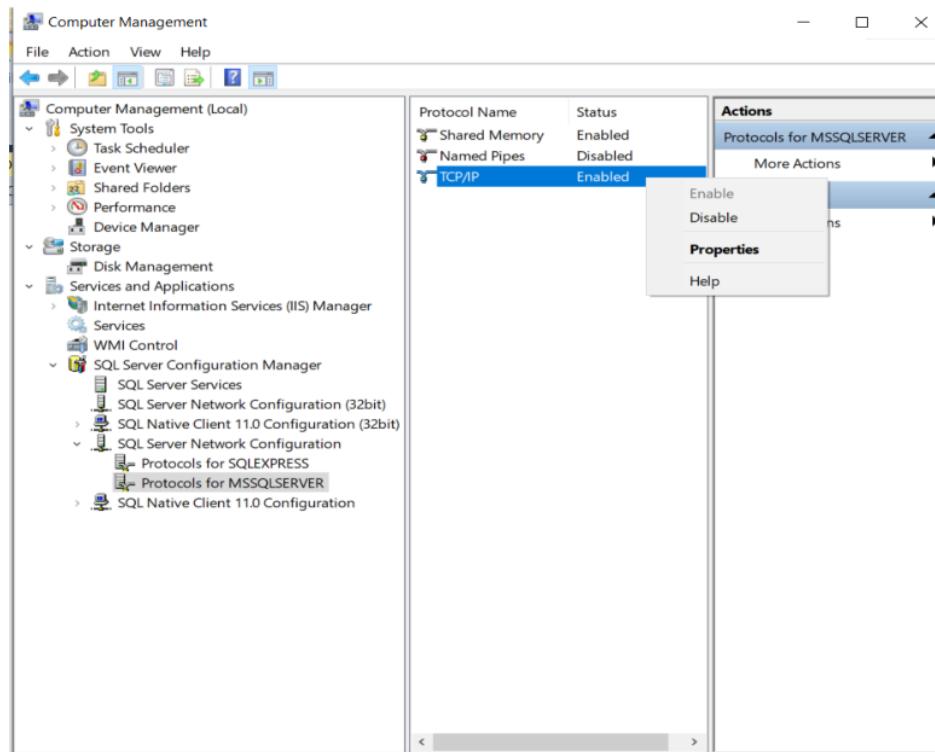
Hướng dẫn tải Microsoft SQL Server



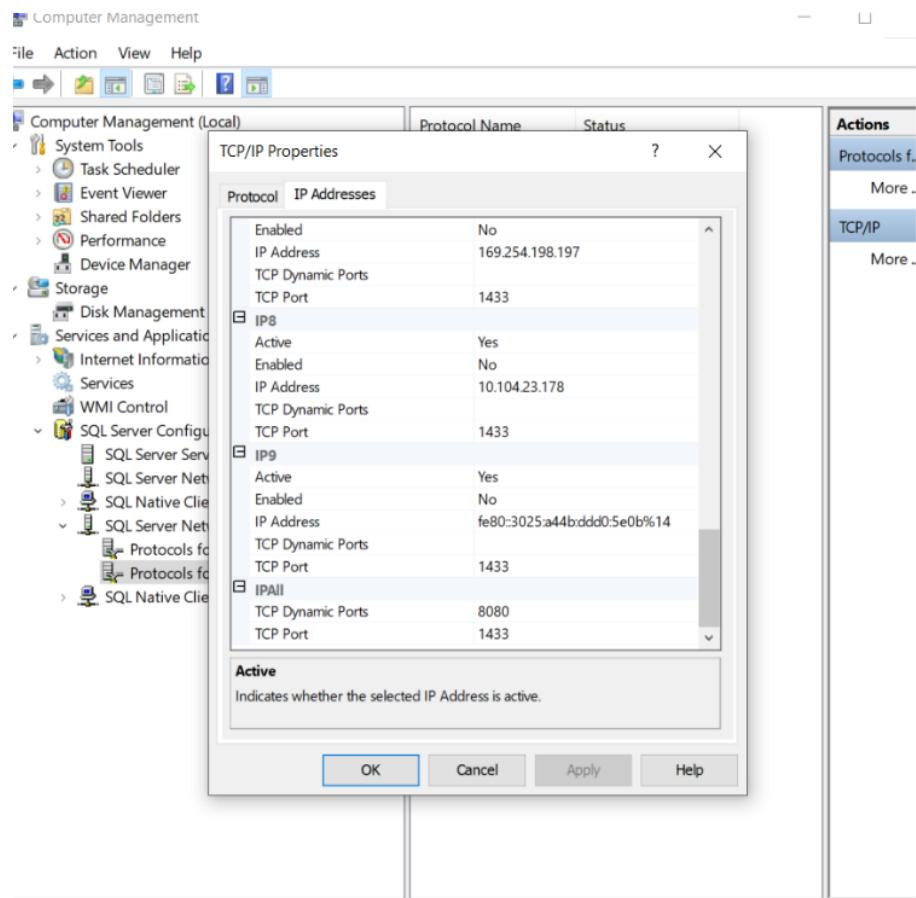
Bước 1: Vào trang chính thức của Microsoft SQL Server để tải và install

Bước 2: Bật phần window của máy lên và tìm kiếm Computer Management

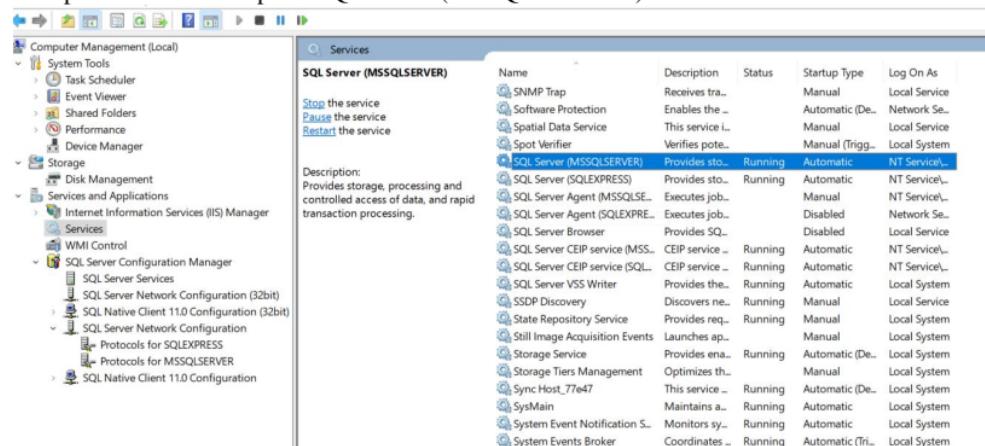
“Vào phần SQL Server Network Configuration và enable TCP/IP của Protocols for MSSQLSERVER”



“Vào phần Properties của TCP/IP vào phần cuối cùng và điều chỉnh TCP Dynamic Ports thành 8080”



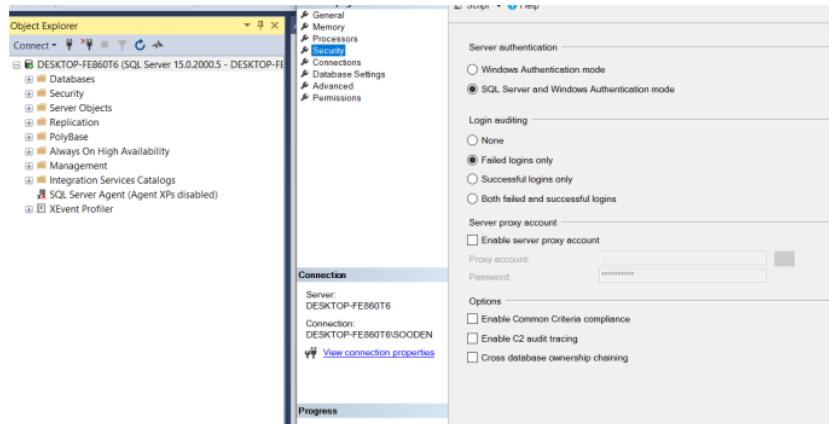
Vào phần Services tìm phần SQL Server(MSSQLSERVER) và restart



Bước 3: Vào Microsoft SQL Server kết nối với server và New 1 Query sau đó chép những đoạn code sau đây rồi execute từng đoạn code

```
1 Create database thcqtt3
2
3 use thcqtt3
4 create table Rooms(
5     ID int not null primary key,
6     name nvarchar(50) null,
7     imageURL nvarchar(50) null
8 )
9
10 use thcqtt3
11 create table Devices(
12     ID int not null primary key,
13     name nvarchar(50) null,
14     imageURL nvarchar(50) null,
15     status int null,
16     room_id int not null,
17     mqtt_topic_sub nvarchar(50) null,
18     mqtt_topic_pub nvarchar(50) null
19 )
20
21 //Đoạn code này để tạo mật khẩu cho Server
22 USE [master]
23 GO
24 ALTER LOGIN [sa] WITH PASSWORD='password'
25 GO
26 ALTER LOGIN [sa] ENABLE
27 GO
```

Bước 4: Vào phần Server Properties -> chọn Security -> chọn SQL Server and Windows Authentication mode rồi disconnect Server



Bước 5: Connect lại server với Authentication là SQL Server Authentication

Login: sa

Password: password

“Bước 6: Mở Database FireWarning -> mở phần design của Table dbo.SensorData -> set khóa chính cho companyID và đổi identity Specification thành Yes”

Column Name	Data Type	Allow Nulls
id	int	<input type="checkbox"/>
name	nvarchar(50)	<input checked="" type="checkbox"/>
imageURL	nvarchar(50)	<input checked="" type="checkbox"/>
status	int	<input checked="" type="checkbox"/>
room_id	int	<input type="checkbox"/>
▶ mqtt_topic_sub	nvarchar(50)	<input checked="" type="checkbox"/>
mqtt_topic_pub	nvarchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
▶ id	int	<input type="checkbox"/>
name	nvarchar(50)	<input checked="" type="checkbox"/>
imageURL	nvarchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

2.3 Giới thiệu về NodeJS

2.3.1 Node JS là gì?

“NodeJS là một môi trường runtime chạy JavaScript sử dụng để chạy ứng dụng web bên ngoài trình duyệt của client. Ryan Dahl tạo ra NodeJS, đây được xem là một giải pháp giúp cho các ứng dụng sử dụng nhiều dữ liệu nhờ vào mô hình hướng sự kiện (event-driven) không đồng bộ.”

2.3.2 Ưu và nhược điểm của NodeJS

a. “Ưu điểm”

- “IO hướng sự kiện không đồng bộ và cho phép xử lý nhiều yêu cầu đồng thời”
- “JavaScript - một ngôn ngữ lập trình dễ tiếp cận”
- “Dễ dàng hơn trong việc giao tiếp giữa client và server”
- “Cộng đồng hỗ trợ tích cực”

b. “Nhược điểm”

- “Không có khả năng mở rộng, vì vậy không thể tận dụng lợi thế mô hình đa lõi trong các phần cứng cấp server hiện nay”
- “Khó thao tác với cơ sở dữ liệu quan hệ”
- “Mỗi call back sẽ đi kèm với rất nhiều callback lồng nhau khác”
- “Cần có kiến thức tốt về JavaScript”
- “Không phù hợp với các tác vụ đòi hỏi nhiều CPU”

2.4 Code hệ thống đo nhiệt độ, độ ẩm bằng NodeJS

Các thư viện cần thiết



Bước 1: Vào page chính thức tải NodeJS về máy và chạy

Bước 2: Mở Visual Studio Code và tạo một project

Bước 3: Mở phần terminal trong view và download các thư viện dưới đây
“npm init -y”

“npm install express mssql cors body-parser”

“npm install fs-extra”

“npm install nodemon --save-dev”

Bước 4: Vào phần package.json

Thêm “start”: “nodemon app.js” bên dưới dòng “test” trong “scripts”

Bước 5:

Tạo file server và sao chép đoạn code bên dưới

```
1  var mysql = require('mysql');
2  var express = require('express');
3  var bodyParser = require("body-parser");
4  const cors = require('cors');
5  var sql = require("mssql");
6  var mysql = require('mysql');
7  var app = express();
8  var sql1 = require("msnodesqlv8")
9  app.use(cors());
10 app.use(express.static('public'));
11 app.use(bodyParser.json());
12 app.use(bodyParser.urlencoded({ extended: true }));
13
14 app.get('/getDevices', function (req, res) {
15   const connect = "server=.;Database=thcntt3;Trusted_Connection=Yes;Driver={SQL Server Native Client 11.0}";
16   const query = " Select * from [thcntt3].[dbo].[Devices]";
17   sql1.query(connect, query, (err, rows) => {
18     res.send(rows);
19   })
20 });
21
22 app.put('/updateDevice/:id', function (req, res) {
23   const connect = "server=.;Database=thcntt3;Trusted_Connection=Yes;Driver={SQL Server Native Client 11.0}";
24   const deviceId = req.params.id; // Lấy giá trị id từ URL params
25   const { status } = req.body; // Lấy dữ liệu cập nhật từ body request
26   console.log(status);
27   // Câu truy vấn UPDATE cơ sở dữ liệu
28   const query = `UPDATE [thcntt3].[dbo].[Devices] SET status='${status}' WHERE mqtt_topic_sub='${deviceId}'`;
29   sql1.query(connect, query, (err, result) => {
30     if (err) {
31       console.error("Error updating device:", err);
32       res.status(500).send("Error updating device");
33     } else {
34       console.log("Device updated successfully");
35       res.send("Device updated successfully");
36     }
37   });
38 });

39
```

```

40  v app.get('/getDevices/:id', function (req, res) {
41    const roomId = req.params.id;
42    // Kiểm tra nếu roomId không hợp lệ (ví dụ: không phải số nguyên dương)
43    v if (!Number.isInteger(+roomId) || +roomId <= 0) {
44      res.status(400).send('Invalid roomId');
45      return;
46    }
47    const connect = "server=.;Database=thcnett3;Trusted_Connection=Yes;Driver={SQL Server Native Client 11.0}";
48    const query = `SELECT * FROM [thcnett3].[dbo].[Devices] WHERE Room_id = ${roomId}`;
49    v sql1.query(connect, query, (err, rows) => {
50      v if (err) {
51        res.status(500).send('Internal Server Error');
52        return;
53      }
54      res.send(rows);
55    });
56  });
57

58 app.get('/getRooms', function (req, res) {
59   const connect = "server=.;Database=thcnett3;Trusted_Connection=Yes;Driver={SQL Server Native Client 11.0}";
60   const query = " Select * from [thcnett3].[dbo].[Rooms]";
61   sql1.query(connect, query, (err, rows) => {
62     res.send(rows);
63   })
64 });
65

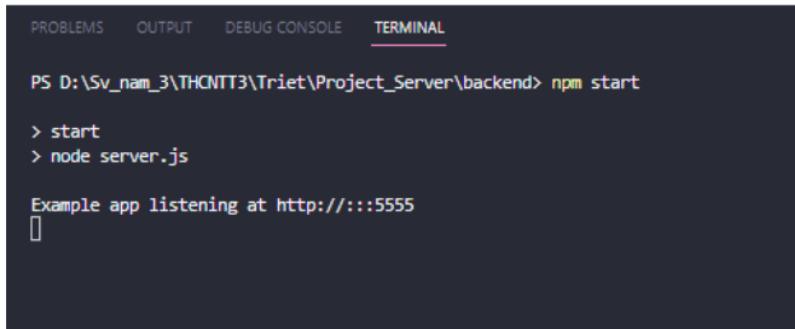
66 // Add a new room
67 app.post('/addRooms', function (req, res) {
68   const connect = "server=.;Database=thcnett3;Trusted_Connection=Yes;Driver={SQL Server Native Client 11.0}";
69   const name = req.body.name; // Get room name from request body
70
71   // SQL query to add room
72   const query = `INSERT INTO [thcnett3].[dbo].[Rooms] (name) VALUES ('${name}')`;
73
74   sql1.query(connect, query, (err, result) => {
75     if (err) {
76       console.error("Error adding room:", err);
77       res.status(500).send("Error adding room");
78     } else {
79       console.log("Room added successfully");
80       res.send("Room added successfully");
81     }
82   });
83 });
84

85 // Update room by ID
86 v app.put('/updateRooms/:id', function (req, res) {
87   const connect = "server=.;Database=thcnett3;Trusted_Connection=Yes;Driver={SQL Server Native Client 11.0}";
88   const id = req.params.id;
89   const name = req.body.name;// Get updated room name from request body
90   console.log(name);
91
92   // SQL query to update room
93   const query = `UPDATE [thcnett3].[dbo].[Rooms] SET name='${name}' WHERE id='${id}'`;
94
95   v sql1.query(connect, query, (err, result) => {
96     if (err) {
97       console.error("Error updating room:", err);
98       res.status(500).send("Error updating room");
99     } else {
100       console.log("Room updated successfully");
101       res.send("Room updated successfully");
102     }
103   });
104 });
105

```

```
106 // Delete room by ID
107 app.delete('/deleteRooms/:id', function (req, res) {
108   const connect = "server=.;Database=thcnett3;Trusted_Connection=True;Driver={SQL Server Native Client 11.0}";
109   const id = req.params.id; // Get room ID from URL params
110
111   // SQL query to delete room
112   const query = `DELETE FROM [thcnett3].[dbo].[Rooms] WHERE id='${id}'`;
113
114   sql1.query(connect, query, (err, result) => {
115     if (err) {
116       console.error("Error deleting room:", err);
117       res.status(500).json({ success: false, message: "Error deleting room" });
118     } else {
119       console.log("Room deleted successfully");
120       res.json({ success: true, message: "Room deleted successfully" });
121     }
122   });
123 });
124 var server = app.listen(5555, function () {
125   var host = server.address().address
126   var port = server.address().port
127   console.log("Example app listening at http://%s:%s", host, port)
128 })
```

Bước 6: Mở terminal của vscode và chạy lệnh npm start



The screenshot shows the VS Code interface with the terminal tab selected. The terminal window displays the following text:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\Sv_nam_3\THCNETT3\Triet\Project_Server\backend> npm start
> start
> node server.js

Example app listening at http://:::5555
[]
```

Bước 7: Test chức năng bằng Postman

Chức năng 1: xuất danh sách

GET Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (8) Test Results Status: 200 OK Time: 517 ms Size: 768 B Save as Example

Pretty Raw Preview Visualize JSON

```

1 [
2   {
3     "id": 1,
4     "name": "gas",
5     "imageURL": "gas.jpg",
6     "status": 180,
7     "room_id": 1,
8     "mqtt_topic_sub": "gas_room1",
9     "mqtt_topic_pub": "gas_room1"
10 },
11 {
12   "id": 2,
13   "name": "gas",
14   "imageURL": "gas.jpg",
15   "status": 0,
16   "room_id": 2,
17   "mqtt_topic_sub": "gas_room2",
18   "mqtt_topic_pub": "gas_room2"
19 },
20 {
21   "id": 3,
22   "name": "led",

```

GET Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (8) Test Results Status: 200 OK Time: 65 ms Size: 425 B Save as Example

Pretty Raw Preview Visualize JSON

```

1 [
2   {
3     "id": 1,
4     "name": "Living Room",
5     "imageURL": "living.png"
6   },
7   {
8     "id": 2,
9     "name": "Kitchen",
10    "imageURL": "kitchen.png"
11   },
12   {
13     "id": 3,
14     "name": "BedRoom",
15     "imageURL": "bedroom.png"
16   }
17 ]

```

Chức năng 2: Thêm dữ liệu

Tạo một tab mới trong Postman đổi từ GET -> POST

Vào phần Body -> raw và đổi định dạng thành JSON

The screenshot shows the Postman interface with a POST request to `http://localhost:8090/api/addHouse`. The Body tab is selected, containing the following JSON payload:

```
1
2   "RoomCode": "16",
3   "Name": "ClassRoom",
4   "Temperature": "31K",
5   "FireStatus": "Yes",
6   "GasStatus": "Yes"
7
```

The response status is 200 OK, with a message: "Data save successfully".

Chức năng 3: Cập nhật dữ liệu

Tạo một tab mới trong Postman đổi từ GET -> PUT

Vào phần Body -> raw và đổi định dạng thành JSON

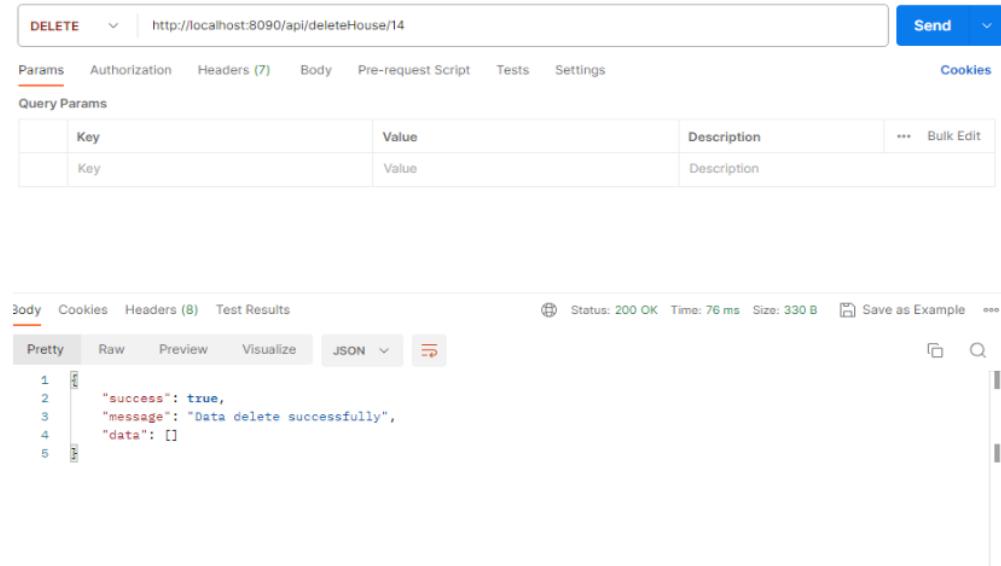
The screenshot shows the Postman interface with a PUT request to `http://localhost:8090/api/updateHouse/15`. The Body tab is selected, containing the same JSON payload as the previous screenshot:

```
1
2   "RoomCode": "16",
3   "Name": "ClassRoom",
4   "Temperature": "31K",
5   "FireStatus": "Yes",
6   "GasStatus": "Yes"
7
```

The response status is 200 OK, with a message: "Data update successfully".

Chức năng 4: Xóa dữ liệu

Tạo một tab mới trong Postman đổi từ GET -> DELETE



The screenshot shows the Postman application interface. At the top, there is a header bar with a dropdown menu set to "DELETE", a URL field containing "http://localhost:8090/api/deleteHouse/14", and a "Send" button. Below the header are tabs for "Params", "Authorization", "Headers (7)", "Body", "Pre-request Script", "Tests", and "Settings". The "Body" tab is selected. Under "Body", there is a "Query Params" table with one row containing "Key" and "Value". The main content area displays the response from the API. At the top of the response view, there are tabs for "Body", "Cookies", "Headers (8)", and "Test Results". The "Body" tab is selected and shows the JSON response:

```
1
2   "success": true,
3   "message": "Data delete successfully",
4   "data": []
5
```

Below the JSON, there are buttons for "Pretty", "Raw", "Preview", "Visualize", and "JSON". To the right of the JSON content, there are status indicators: "Status: 200 OK", "Time: 76 ms", "Size: 330 B", and a "Save as Example" button. There are also search and refresh icons.

3. IoT

3.1 Giới thiệu sơ lược về Arduino IDE

- “IDE trong Arduino IDE là phần có nghĩa là mã nguồn mở, nghĩa là phần mềm này miễn phí cả về phân tài và lấn phần bản quyền: Người dùng có quyền sửa đổi, cải tiến, phát triển, nâng cấp theo một số nguyên tắc chung được nhà phát hành cho phép mà không cần xin phép ai, điều mà họ không được phép làm đối với các phần mềm nguồn đóng..”
- “Sử dụng ngôn ngữ lập trình C/C++ thân thiện với các lập trình viên”
- “Hỗ trợ lập trình tốt cho bo mạch Arduino. **Arduino** có một module quản lý bo mạch, nơi người dùng có thể chọn bo mạch mà họ muốn làm việc cùng và có thể thay đổi bo mạch thông qua Menu. Quá trình sửa đổi lựa chọn cũng liên tục tự động cập nhật để các dữ liệu có sẵn trong bo mạch và dữ liệu sửa đổi đồng nhất với nhau. Bên cạnh đó, **Arduino IDE** cũng giúp bạn tìm ra lỗi từ code mà bạn biết giúp bạn sửa lỗi kịp thời tránh tình trạng bo mạch Arduino làm việc với code lỗi quá lâu dẫn đến hư hỏng hoặc tốc độ xử lý bị giảm sút.”
- “Thư viện hỗ trợ phong phú tích hợp với hơn 700 thư viện, được viết và chia sẻ bởi nhà phát hành **Arduino Software** và thành viên trong cộng đồng Arduino. Mọi người có thể tận dụng chúng cho dự án của riêng mình mà không cần phải bỏ ra bất kỳ chi phí nào.”
- “Hỗ trợ đa nền tảng như Windows, MacOS, Linux. Ngoài ra, người dùng có thể truy cập vào công cụ từ đám mây. Điều này cho phép các nhà lập trình lựa chọn tạo và lưu dự án của mình trên đám mây hoặc xây dựng chương trình trên máy tính và upload nó lên bo mạch Arduino.”

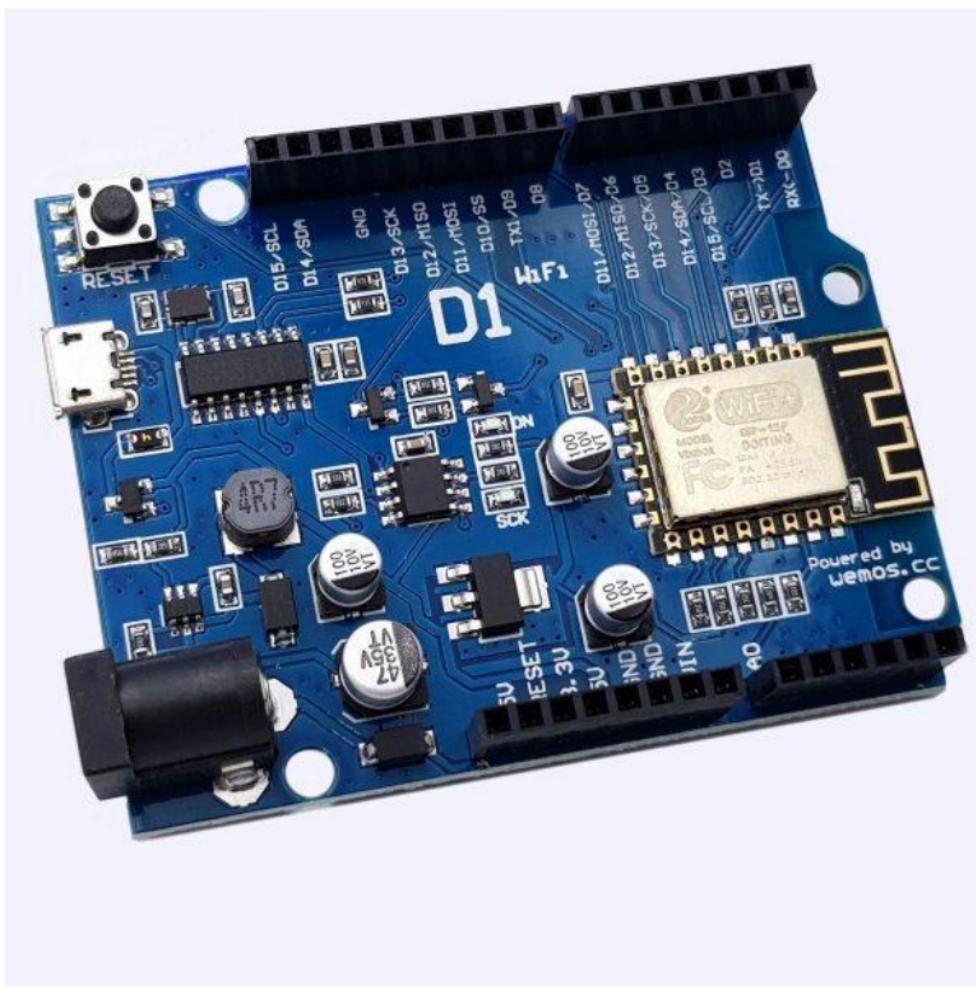
1

3.2 Mạch và sensor sử dụng trong hệ thống FireWarning

3.2.1 Mạch Wemos D1

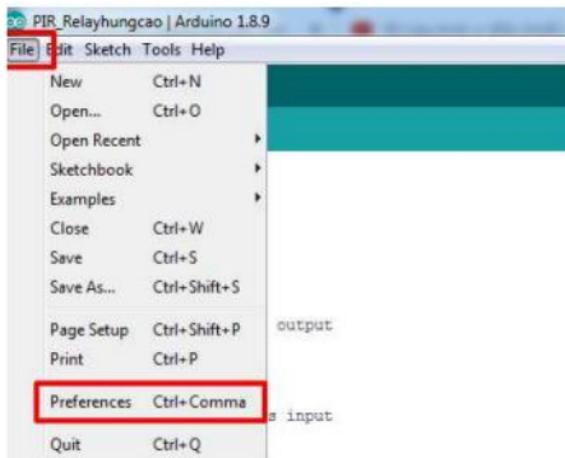
Sơ lược về mạch

“Mạch WeMos D1 là board mạch được phát triển dựa trên Module Esp8266-12E và được thiết kế theo tiêu chuẩn của board mạch Arduino UNO, tương thích với Arduino IDE và NodeMCU. Mạch Wemos D1 được tích hợp Wifi, dễ dàng thực hiện các ứng dụng thu thập dữ liệu và điều khiển thiết bị thông qua Wifi. Mạch Wemos D1 có khả năng chuyển đổi điện áp trên board, cho phép cấp 1 điện áp DC 9-24V để chuyển đổi thành 5V với dòng tối đa 1A.”

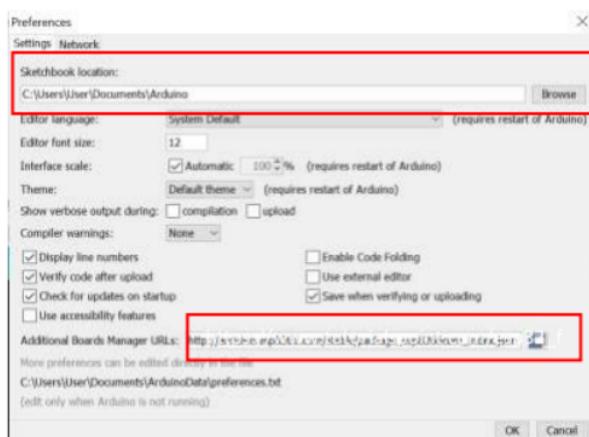


Cách cài đặt thư viện

“Cài đặt board WEMOS D1 và thư viện ArduinoJson”



Bước 1: Chọn File → Preferences

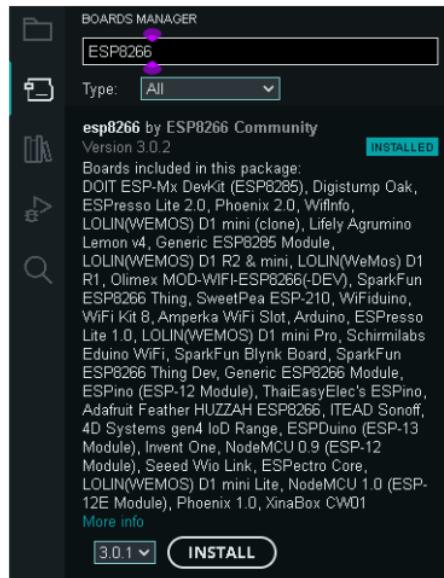


Đường dẫn đến nơi chứa thư mục cài đặt của Arduino

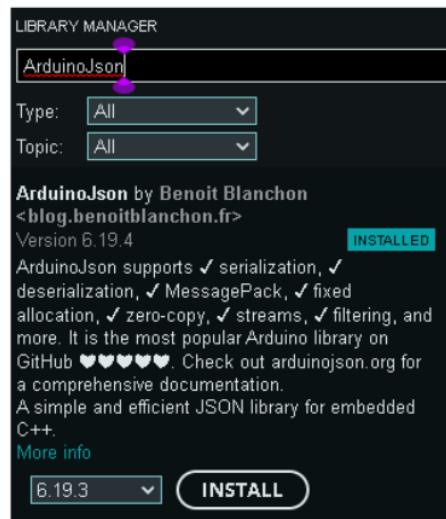
Vd: C:\Program Files (x86)\Arduino

“Bước 2: Copy link http://arduino.esp8266.com/stable/package_esp8266com_index.json”

“Bước 3: Cài đặt board ESP8266 Tools → Board → Boards Manager → ESP8266 → Install”



“Bước 4: Cài đặt thư viện Arduino JSON Chọn Tools → Manage Libraries → ArduinoJson → Install”



3.2.2 Sensor đo độ ẩm và nhiệt độ DHT1

Sơ lược về mạch

“Cảm biến phát hiện lửa (flame sensor) thường được sử dụng cho các ứng dụng phát hiện lửa như: xe robot chữa cháy, cảm biến lửa,... Tầm phát hiện của cảm biến trong khoảng 80cm, góc quét là 60 độ, có thể phát hiện lửa tốt nhất là loại có bước sóng từ 760nm - 1100nm.”

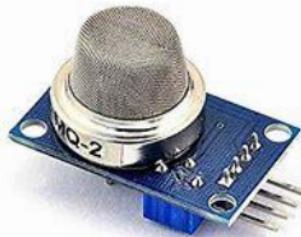
“Cảm biến phát hiện lửa (flame sensor) có hai ngõ ra tín hiệu là Digital và Analog rất dễ sử dụng.”



3.2.3 “Cảm biến gas MQ-2”

“Cảm biến gas MQ-2 là một loại cảm biến dùng để phát hiện sự có mặt của các loại khí trong không khí như khí CO (carbon monoxide), khí LPG (Liquefied Petroleum Gas), khí metan (CH₄), và nhiều hợp chất hữu cơ khác. Đây là một trong những loại cảm biến gas phổ biến và được sử dụng rộng rãi trong các ứng dụng như báo động cháy, thiết bị an toàn gia đình, chuỗi cung ứng gas tự động, và hệ thống điều khiển tự động.”

“Cảm biến gas MQ-2 hoạt động dựa trên nguyên lý trao đổi điện trở của chất tạo thành dòng điện. Khi có khí trong không khí, nó sẽ tương tác với chất tạo điện trở trong cảm biến, làm thay đổi điện trở và tạo ra một tín hiệu thuật ngữ mô phỏng mức độ của khí có mặt.”



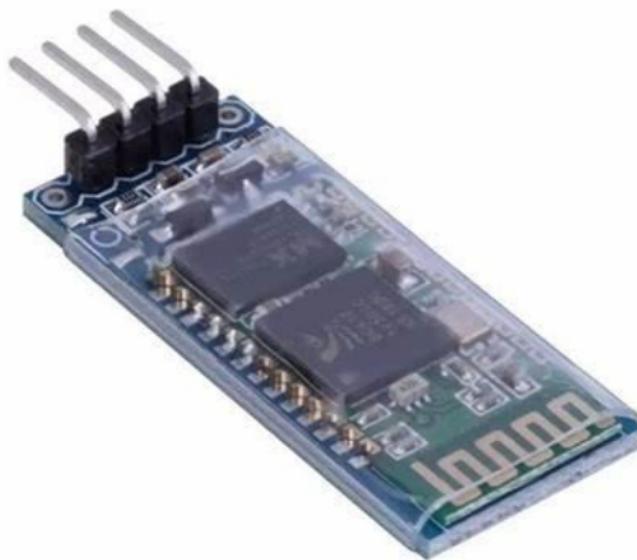
3.2.4 “Bluetooth ZS040”

“Bluetooth ZS040 là một mô-đun Bluetooth hỗ trợ chuẩn Bluetooth 2.0 + EDR (Enhanced Data Rate). Đây là một mô-đun Bluetooth dựa trên chip CSR BC417, có kích thước nhỏ gọn và sử dụng giao thức UART để giao tiếp với các thiết bị ngoại vi khác.”

“Mô-đun Bluetooth ZS040 có các đặc điểm kỹ thuật chính sau: ”

- “Khoảng cách hoạt động: khoảng 10m.”
- “Tốc độ truyền dữ liệu tối đa: 3Mbps.”
- “Giao diện giao tiếp: UART TTL (3.3V).”
- “Điện áp hoạt động: 3.6 - 6VDC.”
- “Công suất tiêu thụ: khoảng 30mA.”
- “Hỗ trợ chế độ Master và Slave.”
- “Hỗ trợ mã hóa và xác thực bảo mật.”

“Mô-đun Bluetooth ZS040 được sử dụng rộng rãi trong các dự án DIY (làm đồ tự chế) và ứng dụng IoT nhỏ gọn do tính đơn giản và linh hoạt của nó trong việc thiết lập kết nối không dây qua Bluetooth.”

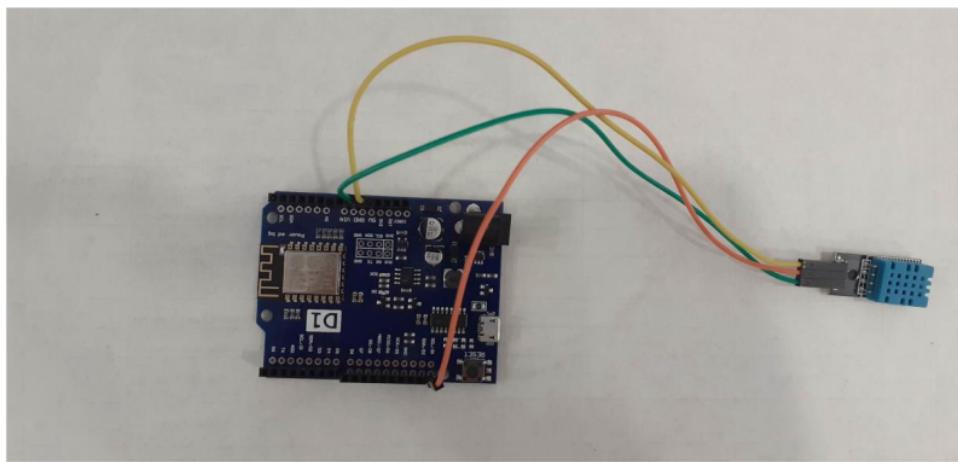


1

3.3 Nối dây và nạp chương trình.

Bảng nối dây theo hình

Mạch	Flame Sensor	Wemos D1
Chân	-	GND
	+	5V
	OUT	SCL/D1 (D15)



Code chạy chương trình được sử dụng trong hệ thống

```

1 #include <ESP8266WiFi.h>
2 #include <PubSubClient.h>
3 #include <ArduinoJson.h>
4 #define ssid "SNOW"
5 #define password "0906474978"
6 // Thông tin về MQTT Broker
7 #define mqtt_server "broker.emqx.io"
8 const uint16_t mqtt_port = 1883; //Port của MQTT broker
9 #define mqtt_topic_pub_led1 "led_room1"
10 #define mqtt_topic_sub_led1 "led_room1"
11 #define mqtt_topic_pub_gas1 "gas_room1"
12 #define mqtt_topic_sub_gas1 "gas_room1"
13 #define mqtt_topic_pub_led2 "led_room2"
14 #define mqtt_topic_sub_led2 "led_room2"
15 #define mqtt_topic_pub_gas2 "gas_room2"
16 #define mqtt_topic_sub_gas2 "gas_room2"
17 int cambien = A0;
18 int giatri;
19 WiFiClient espClient;
20 PubSubClient client(espClient);
21 StaticJsonDocument<256> doc; //PubSubClient limits the message size to 256 bytes (including
22 char ledstatus[32]="on";
23 char topicmqtt[32]="none";
24 void setup() {
25   pinMode(cambien,INPUT);
26   pinMode(D6, OUTPUT);
27   digitalWrite(D6, HIGH);
28   Serial.begin(115200);
29   // hàm thực hiện chức năng kết nối Wifi và in ra địa chỉ IP của ESP8266
30   setup_wifi();
31   // cài đặt server eclipspe mosquitto / mqtx và lắng nghe client ở port 1883
32   client.setServer(mqtt_server, mqtt_port);
33   // gọi hàm callback để thực hiện các chức năng publish/subscribe
34   client.setCallback(callback);
35   // gọi hàm reconnect() để thực hiện kết nối lại với server khi bị mất kết nối
36   reconnect();
37 }
38 void setup_wifi() {
39   delay(10);
40   Serial.println();
41   Serial.print("Connecting to ");
42   Serial.println(ssid);
43   // kết nối đến mạng Wifi
44   WiFi.begin(ssid, password);
45   // in ra dấu . nếu chưa kết nối được đến mạng Wifi
46   while (WiFi.status() != WL_CONNECTED) {
47     delay(500);
48     Serial.print(".");
49   }
50   // in ra thông báo đã kết nối và địa chỉ IP của ESP8266
51   Serial.println("");
52   Serial.println("WiFi connected");
53   Serial.println("IP address: ");
54   Serial.println(WiFi.localIP());
55 }
56 void callback(char* topic, byte* payload, unsigned int length) {
57   Serial.println("callback");
58   //chuyen doi *byte sang json
59   deserializeJson(doc, payload, length);
60   //doc thong tin status tu chuỗi json trả về
61   strcpy(ledstatus, doc["status"] | "on", sizeof(ledstatus));
62   String mystring(ledstatus);

```

```
63  strlcpy(topicmqtt, doc["mqtt_topic_sub"] | "on", sizeof(topicmqtt));
64  String mytopic(topicmqtt);
65  //in ra tên của topic và nội dung nhận được từ kênh MQTT lens đã publish
66  Serial.print("Message arrived [");
67  Serial.print(topic);
68  Serial.print("] ");
69  // in trạng thái của led
70  Serial.println(ledstatus);
71
72  if(mytopic == "led_room1")
73  {
74      StaticJsonDocument<256> doc2;
75      char buffer[256];
76      doc2["topic"] = "led_room1";
77      if(mystring == "1")
78      {
79          Serial.print("turn on");
80          doc2["status"] = "1";
81          digitalWrite(D6, HIGH);
82      }else if(mystring == "0")
83      {
84          doc2["status"] = "0";
85          Serial.print("turn off");
86          digitalWrite(D6, LOW);
87      }
88      size_t n = serializeJson(doc2, buffer);
89      client.publish(mqtt_topic_pub_led1, buffer, n);
90  }
91
92  if(mytopic == "led_room2")
93  {
```

```
94     StaticJsonDocument<256> doc1;
95     char buffer[256];
96     doc1["topic"] = "led_room2";
97     if(mystring == "1")
98     {
99         //on
100        doc1["status"] = "1";
101        Serial.print("turn on");
102        digitalWrite(D2, HIGH);
103    }  
104    else if(mystring == "0")
105    {
106        doc1["status"] = "0";
107        Serial.print("turn off");
108        digitalWrite(D2, LOW);
109    }
110    size_t n = serializeJson(doc1, buffer);
111    client.publish(mqtt_topic_pub_led2, buffer, n);
112 }
113 StaticJsonDocument<256> doc3;
114 giatri = analogRead(cambien);
115     Serial.println("GAS : ");
116     Serial.print(giatri);
117     char buffer[256];
118     doc3["status"] = String(giatri);
119     doc3["topic"] = "gas_room1";
120     size_t n = serializeJson(doc3, buffer);
121     client.publish(mqtt_topic_pub_gas1, buffer, n);
122     delay(2000);
123 }
124 void reconnect() {
125     Serial.println("reconnect");
```

```
125 // lặp cho đến khi được kết nối trở lại
126 while (!client.connected()) {
127     Serial.print("Attempting MQTT connection...");
128     // hàm connect có đối số thứ 1 là id đại diện cho mqtt client, đối số thứ 2 là username và đối
129     if (client.connect("thcntt3_thcntt3")) {
130         Serial.println("connected");
131         // publish gói tin "Hello esp8266!" đến topic mqtt_topic_pub_test
132
133     char buffer[256];
134     doc["message"] = "Hello esp8266!";
135     size_t n = serializeJson(doc, buffer);
136     client.publish(mqtt_topic_pub_led1, buffer, n);
137     // publish gói tin "{\"message\":\"turn on led\",\"name\":\"led\",\"status\":\"on\"}" đến topic mqtt_topic_pub_led
138     doc["name"] = "led";
139     doc["status"] = "on";
140     doc["message"] = "turn on led";
141     n=serializeJson(doc, buffer);
142     // đăng ký nhận gói tin tại topic wemos/ledstatus
143     client.subscribe(mqtt_topic_sub_led1);
144     client.subscribe(mqtt_topic_sub_led2);
145     client.subscribe(mqtt_topic_sub_gas1);
146     client.subscribe(mqtt_topic_sub_gas2);
147
148 } else {
149     // in ra màn hình trạng thái của client khi không kết nối được với MQTT broker
150     Serial.print("failed, rc=");
151     Serial.print(client.state());
152     Serial.println(" try again in 5 seconds");
153     // delay 5s trước khi thử lại
154     delay(5000);
155 }
156 }
157 }
158 void loop() {
159     // kiểm tra nếu ESP8266 chưa kết nối được thì sẽ thực hiện kết nối lại
160     if (!client.connected()) {
161         reconnect();
162     }
163     client.loop();
164 }
165 }
```

ORIGINALITY REPORT

11 %
SIMILARITY INDEX

3%
INTERNET SOURCES

2%
PUBLICATIONS

11 %
STUDENT PAPERS

PRIMARY SOURCES

1 Submitted to Hoa Sen University 11 %
Student Paper

Exclude quotes On

Exclude bibliography On

Exclude matches Off