**You said:**

import React from "react"; import "./Tang1Beta.css"; import image from "../assets/T1.png"; import { Wifi } from "lucide-react"; // Import icon Wi-Fi export function Tang1Beta() { const wifiLocations = [ { name: "AP-BT-Phong AI", top: "17%", left: "15%" }, { name: "AP-BT-IT-ACP", top: "17%", left: "37%" }, { name: "Phòng họp", top: "15%", left: "62%" }, { name: "AP-BT-Sales-U6P", top: "15%", left: "75%" }, { name: "AP-BT-CTSV-U6", top: "13%", left: "88%" }, { name: "AP-BT-SanTruong-01-U6", top: "28%", left: "55%" }, { name: "AP-BT-SanTruong-03-ACP", top: "50%", left: "67%" }, { name: "AP-BT-FU-U6P", top: "86%", left: "80%" }, { name: "AP-BT-SanTruong-02-U6P", top: "78%", left: "40%" }, { name: "Thư viện-01", top: "83%", left: "14%" }, { name: "Thư viện-02", top: "53%", left: "14%" }, { name: "Thư viện-03", top: "33%", left: "14%" }, ]; return ( <div className="tang1beta"> <div className="map-container"> <img src={image} alt="Tang 1 Beta" className="map-image" /> {wifiLocations.map((wifi, index) => ( <div key={index} className="wifi-marker" style={{ top: wifi.top, left: wifi.left }} > <Wifi className="wifi-icon" size={28} color="green" /> <div className="wifi-name">{wifi.name}</div> </div> ))} </div> </div> ); } cách kết nối tới backend và cơ sở dữ liệu để làm chức năng thêm wifi, sửa wifi, xoá wifi

**ChatGPT said:**

Bạn cần thiết lập backend để lưu trữ thông tin Wi-Fi và kết nối frontend với API để thêm, sửa, xóa điểm Wi-Fi. Dưới đây là cách thực hiện:

**1. Backend (Node.js + Express + MongoDB)**

**Cài đặt backend**

Trước tiên, bạn cần cài đặt Node.js và MongoDB. Sau đó, khởi tạo một dự án Node.js:

sh

CopyEdit

mkdir backend

cd backend

npm init -y

Cài đặt các thư viện cần thiết:

sh

CopyEdit

npm install express mongoose cors dotenv

Tạo file .env để cấu hình kết nối MongoDB:

ini

CopyEdit

MONGO\_URI=mongodb://localhost:27017/wifi\_locations

PORT=5000

**Tạo server với Express**

Tạo file server.js:

javascript

CopyEdit

require("dotenv").config();

const express = require("express");

const mongoose = require("mongoose");

const cors = require("cors");

const app = express();

app.use(cors());

app.use(express.json());

mongoose.connect(process.env.MONGO\_URI, { useNewUrlParser: true, useUnifiedTopology: true })

.then(() => console.log("MongoDB Connected"))

.catch(err => console.log(err));

const wifiSchema = new mongoose.Schema({

name: String,

top: String,

left: String

});

const Wifi = mongoose.model("Wifi", wifiSchema);

// Lấy danh sách Wi-Fi

app.get("/api/wifi", async (req, res) => {

const wifiList = await Wifi.find();

res.json(wifiList);

});

// Thêm Wi-Fi

app.post("/api/wifi", async (req, res) => {

const newWifi = new Wifi(req.body);

await newWifi.save();

res.json(newWifi);

});

// Cập nhật Wi-Fi

app.put("/api/wifi/:id", async (req, res) => {

const updatedWifi = await Wifi.findByIdAndUpdate(req.params.id, req.body, { new: true });

res.json(updatedWifi);

});

// Xóa Wi-Fi

app.delete("/api/wifi/:id", async (req, res) => {

await Wifi.findByIdAndDelete(req.params.id);

res.json({ message: "Deleted successfully" });

});

const PORT = process.env.PORT || 5000;

app.listen(PORT, () => console.log(`Server running on port ${PORT}`));

Chạy server:

sh

CopyEdit

node server.js

**2. Frontend (React)**

**Cài đặt Axios để gọi API**

sh

CopyEdit

npm install axios

**Tích hợp API vào React**

Cập nhật Tang1Beta.js để gọi dữ liệu từ backend:

javascript

CopyEdit

import React, { useEffect, useState } from "react";

import axios from "axios";

import "./Tang1Beta.css";

import image from "../assets/T1.png";

import { Wifi } from "lucide-react";

export function Tang1Beta() {

const [wifiLocations, setWifiLocations] = useState([]);

// Lấy dữ liệu từ backend

useEffect(() => {

axios.get("http://localhost:5000/api/wifi")

.then(response => setWifiLocations(response.data))

.catch(error => console.error("Error fetching WiFi data:", error));

}, []);

// Hàm thêm WiFi

const addWifi = () => {

const newWifi = { name: "AP Mới", top: "50%", left: "50%" };

axios.post("http://localhost:5000/api/wifi", newWifi)

.then(response => setWifiLocations([...wifiLocations, response.data]));

};

// Hàm xóa WiFi

const deleteWifi = (id) => {

axios.delete(`http://localhost:5000/api/wifi/${id}`)

.then(() => setWifiLocations(wifiLocations.filter(w => w.\_id !== id)));

};

return (

<div className="tang1beta">

<button onClick={addWifi}>Thêm WiFi</button>

<div className="map-container">

<img src={image} alt="Tang 1 Beta" className="map-image" />

{wifiLocations.map((wifi) => (

<div

key={wifi.\_id}

className="wifi-marker"

style={{ top: wifi.top, left: wifi.left }}

>

<Wifi className="wifi-icon" size={28} color="green" />

<div className="wifi-name">

{wifi.name}

<button onClick={() => deleteWifi(wifi.\_id)}>Xóa</button>

</div>

</div>

))}

</div>

</div>

);

}

**3. Chạy ứng dụng**

**Chạy backend**

sh

CopyEdit

node server.js

**Chạy frontend**

sh

CopyEdit

npm start

Giờ đây, bạn có thể thêm, sửa, xóa điểm Wi-Fi trên bản đồ! 🚀

**You said:**

MONGO\_URI=mongodb://localhost:27017/wifi\_locations MONGO\_URI=mongodb://localhost:27017/wifi\_locations : The term 'MONGO\_URI=mongodb://localhost:27017/wifi\_locations' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try again. At line:1 char:1 + MONGO\_URI=mongodb://localhost:27017/wifi\_locations + ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ + CategoryInfo : ObjectNotFound: (MONGO\_URI=mongo.../wifi\_locations:String) [], CommandNotFoundException + FullyQualifiedErrorId : CommandNotFoundException PS C:\ReactJS\mapthietbi\backend> MONGO\_URI=mongodb://localhost:27017/wifi\_locations MONGO\_URI=mongodb://localhost:27017/wifi\_locations : The term 'MONGO\_URI=mongodb://localhost:27017/wifi\_locations' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try again. At line:1 char:1 + MONGO\_URI=mongodb://localhost:27017/wifi\_locations + ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ + CategoryInfo : ObjectNotFound: (MONGO\_URI=mongo.../wifi\_locations:String) [], CommandNotFoundException + FullyQualifiedErrorId : CommandNotFoundException

**ChatGPT said:**

Lỗi này xảy ra do bạn đang sử dụng **PowerShell** và cố gắng thiết lập biến môi trường bằng cú pháp của **Linux/MacOS**. Trong **PowerShell**, bạn cần thiết lập biến môi trường theo cú pháp khác.

**✅ Cách đúng để thiết lập biến môi trường trong PowerShell**

Thay vì viết:

sh

CopyEdit

MONGO\_URI=mongodb://localhost:27017/wifi\_locations

Bạn hãy dùng lệnh sau trong **PowerShell**:

powershell

CopyEdit

$env:MONGO\_URI="mongodb://localhost:27017/wifi\_locations"

Lệnh này sẽ đặt giá trị MONGO\_URI cho phiên làm việc hiện tại của PowerShell.

Nếu bạn muốn thiết lập biến này trong file .env, hãy đảm bảo rằng nội dung file .env đúng như sau:

ini

CopyEdit

MONGO\_URI=mongodb://localhost:27017/wifi\_locations

PORT=5000

Sau đó, trong file server.js, sử dụng **dotenv** để đọc biến này:

javascript

CopyEdit

require("dotenv").config();

const mongoose = require("mongoose");

const mongoURI = process.env.MONGO\_URI;

mongoose.connect(mongoURI, { useNewUrlParser: true, useUnifiedTopology: true })

.then(() => console.log("MongoDB Connected"))

.catch(err => console.log("MongoDB Connection Error:", err));

**🔄 Nếu vẫn lỗi, hãy kiểm tra**

1. **Đảm bảo đã cài đặt MongoDB**  
   Chạy lệnh sau trong PowerShell để kiểm tra:

powershell

CopyEdit

mongod --version

1. **Kiểm tra MongoDB có đang chạy không**  
   Nếu chưa chạy, hãy khởi động MongoDB bằng lệnh:

powershell

CopyEdit

net start MongoDB

Hoặc nếu bạn dùng MongoDB Community với mongod, chạy:

powershell

CopyEdit

mongod

1. **Sử dụng Git Bash thay vì PowerShell**  
   Nếu bạn muốn dùng cú pháp Linux, hãy mở **Git Bash** (nếu đã cài đặt Git) và thử lệnh:

sh

CopyEdit

export MONGO\_URI="mongodb://localhost:27017/wifi\_locations"

Sau khi thiết lập đúng, bạn có thể chạy backend bằng:

powershell

CopyEdit

node server.js

Hoặc nếu dùng **nodemon**:

powershell

CopyEdit

npx nodemon server.js

🚀 **Giờ bạn có thể kết nối MongoDB mà không gặp lỗi!** Nếu còn vấn đề, hãy cho mình biết lỗi cụ thể nhé! 😊

**You said:**

PS C:\ReactJS\mapthietbi\backend> $env:MONGO\_URI="mongodb://localhost:27017/wifi\_locations" PS C:\ReactJS\mapthietbi\backend> làm gì tiếp

**ChatGPT said:**

Sau khi thiết lập biến môi trường thành công, bạn có thể tiếp tục các bước sau để chạy backend:

**1. Kiểm tra MongoDB đã chạy chưa**

Trước khi chạy server, bạn cần đảm bảo MongoDB đang hoạt động. Mở **PowerShell** hoặc **Command Prompt** và chạy lệnh:

powershell

CopyEdit

mongod

Nếu MongoDB đã được cài đặt dưới dạng dịch vụ trên Windows, bạn có thể chạy:

powershell

CopyEdit

net start MongoDB

Nếu MongoDB đang chạy, bạn có thể tiếp tục.

**2. Chạy server Node.js**

Di chuyển vào thư mục backend:

powershell

CopyEdit

cd C:\ReactJS\mapthietbi\backend

Sau đó, chạy server:

powershell

CopyEdit

node server.js

Nếu bạn muốn server tự động reload khi có thay đổi, hãy sử dụng nodemon (nếu chưa cài, bạn có thể cài bằng npm install -g nodemon):

powershell

CopyEdit

npx nodemon server.js

**3. Kiểm tra API có hoạt động không**

Mở trình duyệt hoặc sử dụng Postman để kiểm tra API:

* **Lấy danh sách Wi-Fi:**  
  Truy cập: http://localhost:5000/api/wifi  
  Nếu API hoạt động, bạn sẽ thấy danh sách các điểm Wi-Fi được lưu trong MongoDB.
* **Thêm Wi-Fi mới:**  
  Gửi yêu cầu POST đến http://localhost:5000/api/wifi với nội dung JSON như sau:

json

CopyEdit

{

"name": "AP Test",

"top": "50%",

"left": "50%"

}

* **Xóa Wi-Fi:**  
  Gửi yêu cầu DELETE đến http://localhost:5000/api/wifi/:id (thay :id bằng ID thực tế của một điểm Wi-Fi trong cơ sở dữ liệu).

**4. Chạy Frontend**

Nếu backend hoạt động ổn, bạn có thể chạy frontend để hiển thị dữ liệu:

Di chuyển vào thư mục frontend:

powershell

CopyEdit

cd C:\ReactJS\mapthietbi\frontend

Chạy ứng dụng React:

CopyEdit

**You said:**

mongod mongod : The term 'mongod' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try again. At line:1 char:1 + mongod + ~~~~~~ + CategoryInfo : ObjectNotFound: (mongod:String) [], CommandNotFoundException + FullyQualifiedErrorId : CommandNotFoundException

**ChatGPT said:**

Lỗi này xảy ra vì PowerShell không tìm thấy MongoDB. Bạn có thể kiểm tra và khắc phục bằng các bước sau:

**1. Kiểm tra MongoDB đã được cài đặt chưa**

Chạy lệnh sau trong PowerShell:

powershell

CopyEdit

Get-Command mongod

Nếu không có kết quả, có thể MongoDB chưa được cài đặt hoặc chưa được thêm vào **PATH**.

**2. Cài đặt MongoDB (nếu chưa có)**

Nếu chưa cài, bạn có thể tải **MongoDB Community Edition** tại:  
🔗 <https://www.mongodb.com/try/download/community>

* Chọn phiên bản dành cho **Windows**
* Trong quá trình cài đặt, **chọn "Complete Installation"**
* Đảm bảo đã chọn tùy chọn **"Install MongoDB as a Service"** để MongoDB tự động chạy

Sau khi cài đặt xong, thử chạy lệnh:

powershell

CopyEdit

mongod --version

Nếu thấy phiên bản MongoDB, bạn có thể tiếp tục.

**3. Thêm MongoDB vào PATH (nếu đã cài nhưng không tìm thấy lệnh)**

Nếu bạn đã cài MongoDB nhưng vẫn không thể chạy mongod, có thể nó chưa được thêm vào **PATH**.

**Thêm thủ công vào PATH**

1. Mở **File Explorer**
2. Điều hướng đến thư mục cài đặt MongoDB (thường là):

arduino

CopyEdit

C:\Program Files\MongoDB\Server\6.0\bin

1. Sao chép đường dẫn này
2. Mở PowerShell và chạy:

powershell

CopyEdit

[System.Environment]::SetEnvironmentVariable("Path", $env:Path + ";C:\Program Files\MongoDB\Server\6.0\bin", [System.EnvironmentVariableTarget]::Machine)

1. Khởi động lại máy tính hoặc mở cửa sổ PowerShell mới và thử lại:

powershell

CopyEdit

mongod --version

**4. Chạy MongoDB**

Nếu MongoDB đã được cài dưới dạng **service**, bạn có thể khởi động nó bằng:

powershell

CopyEdit

net start MongoDB

Hoặc, nếu muốn chạy thủ công:

powershell

CopyEdit

mongod --dbpath C:\data\db

📌 **Lưu ý:**

* Nếu C:\data\db chưa tồn tại, bạn cần tạo thư mục này trước khi chạy:

powershell

CopyEdit

mkdir C:\data\db

**5. Kiểm tra MongoDB có chạy không**

Chạy lệnh:

powershell

CopyEdit

mongo

Nếu vào được shell của MongoDB (>) thì đã thành công.

✅ **Sau khi MongoDB chạy, bạn có thể tiếp tục với backend:**

powershell

CopyEdit

cd C:\ReactJS\mapthietbi\backend

node server.js

Nếu vẫn gặp lỗi, hãy báo lại mình nhé! 🚀

**You said:**

import React from "react"; import "./Tang1Beta.css"; import image from "../assets/T1.png"; import { Wifi } from "lucide-react"; // Import icon Wi-Fi export function Tang1Beta() { const wifiLocations = [ { name: "AP-BT-Phong AI", top: "17%", left: "15%" }, { name: "AP-BT-IT-ACP", top: "17%", left: "37%" }, { name: "Phòng họp", top: "15%", left: "62%" }, { name: "AP-BT-Sales-U6P", top: "15%", left: "75%" }, { name: "AP-BT-CTSV-U6", top: "13%", left: "88%" }, { name: "AP-BT-SanTruong-01-U6", top: "28%", left: "55%" }, { name: "AP-BT-SanTruong-03-ACP", top: "50%", left: "67%" }, { name: "AP-BT-FU-U6P", top: "86%", left: "80%" }, { name: "AP-BT-SanTruong-02-U6P", top: "78%", left: "40%" }, { name: "Thư viện-01", top: "83%", left: "14%" }, { name: "Thư viện-02", top: "53%", left: "14%" }, { name: "Thư viện-03", top: "33%", left: "14%" }, ]; return ( <div className="tang1beta"> <div className="map-container"> <img src={image} alt="Tang 1 Beta" className="map-image" /> {wifiLocations.map((wifi, index) => ( <div key={index} className="wifi-marker" style={{ top: wifi.top, left: wifi.left }} > <Wifi className="wifi-icon" size={28} color="green" /> <div className="wifi-name">{wifi.name}</div> </div> ))} </div> </div> ); } cách kết nối tới backend và cơ sở dữ liệu sql server để làm chức năng thêm wifi, sửa wifi, xoá wifi

**ChatGPT said:**

Để kết nối frontend React với backend sử dụng **SQL Server** để quản lý dữ liệu Wi-Fi (thêm, sửa, xóa), bạn có thể làm theo các bước sau:

**1. Thiết lập Backend (Node.js + Express + mssql)**

**Bước 1: Cài đặt Node.js và các thư viện cần thiết**

Mở terminal và chạy:

sh

CopyEdit

npm init -y

npm install express mssql dotenv cors body-parser

**Bước 2: Cấu hình kết nối SQL Server**

Tạo file .env để lưu thông tin kết nối:

ini

CopyEdit

DB\_USER=sa

DB\_PASSWORD=your\_password

DB\_SERVER=localhost

DB\_DATABASE=wifi\_locations

DB\_PORT=1433

Tạo file db.js để kết nối SQL Server:

javascript

CopyEdit

require("dotenv").config();

const sql = require("mssql");

const config = {

user: process.env.DB\_USER,

password: process.env.DB\_PASSWORD,

server: process.env.DB\_SERVER,

database: process.env.DB\_DATABASE,

port: parseInt(process.env.DB\_PORT),

options: {

encrypt: false, // Nếu dùng Azure thì đổi thành true

trustServerCertificate: true,

},

};

const poolPromise = new sql.ConnectionPool(config)

.connect()

.then(pool => {

console.log("✅ Connected to SQL Server");

return pool;

})

.catch(err => console.error("❌ Database Connection Failed: ", err));

module.exports = { sql, poolPromise };

**2. Xây dựng API cho Wi-Fi**

Tạo server.js và thêm API cho Wi-Fi:

javascript

CopyEdit

const express = require("express");

const cors = require("cors");

const bodyParser = require("body-parser");

const { sql, poolPromise } = require("./db");

const app = express();

const PORT = 5000;

app.use(cors());

app.use(bodyParser.json());

// Lấy danh sách Wi-Fi

app.get("/api/wifi", async (req, res) => {

try {

const pool = await poolPromise;

const result = await pool.request().query("SELECT \* FROM WifiLocations");

res.json(result.recordset);

} catch (error) {

res.status(500).json({ error: error.message });

}

});

// Thêm Wi-Fi mới

app.post("/api/wifi", async (req, res) => {

try {

const { name, top, left } = req.body;

const pool = await poolPromise;

await pool

.request()

.input("name", sql.NVarChar, name)

.input("top", sql.VarChar, top)

.input("left", sql.VarChar, left)

.query("INSERT INTO WifiLocations (name, top, left) VALUES (@name, @top, @left)");

res.json({ message: "Wi-Fi added successfully!" });

} catch (error) {

res.status(500).json({ error: error.message });

}

});

// Sửa Wi-Fi

app.put("/api/wifi/:id", async (req, res) => {

try {

const { id } = req.params;

const { name, top, left } = req.body;

const pool = await poolPromise;

await pool

.request()

.input("id", sql.Int, id)

.input("name", sql.NVarChar, name)

.input("top", sql.VarChar, top)

.input("left", sql.VarChar, left)

.query("UPDATE WifiLocations SET name = @name, top = @top, left = @left WHERE id = @id");

res.json({ message: "Wi-Fi updated successfully!" });

} catch (error) {

res.status(500).json({ error: error.message });

}

});

// Xóa Wi-Fi

app.delete("/api/wifi/:id", async (req, res) => {

try {

const { id } = req.params;

const pool = await poolPromise;

await pool.request().input("id", sql.Int, id).query("DELETE FROM WifiLocations WHERE id = @id");

res.json({ message: "Wi-Fi deleted successfully!" });

} catch (error) {

res.status(500).json({ error: error.message });

}

});

app.listen(PORT, () => console.log(`🚀 Server is running on port ${PORT}`));

**3. Tạo Database & Bảng Wi-Fi trong SQL Server**

Mở **SQL Server Management Studio (SSMS)** hoặc dùng **Azure Data Studio**, chạy lệnh sau:

sql

CopyEdit

CREATE DATABASE wifi\_locations;

USE wifi\_locations;

CREATE TABLE WifiLocations (

id INT IDENTITY(1,1) PRIMARY KEY,

name NVARCHAR(255) NOT NULL,

top VARCHAR(10) NOT NULL,

left VARCHAR(10) NOT NULL

);

**4. Kết nối Frontend React**

**Cập nhật Tang1Beta.js để lấy dữ liệu từ API**

Thay vì dữ liệu tĩnh, sử dụng **fetch API** để lấy danh sách từ backend:

javascript

CopyEdit

import React, { useEffect, useState } from "react";

import "./Tang1Beta.css";

import image from "../assets/T1.png";

import { Wifi } from "lucide-react";

export function Tang1Beta() {

const [wifiLocations, setWifiLocations] = useState([]);

useEffect(() => {

fetch("http://localhost:5000/api/wifi")

.then(response => response.json())

.then(data => setWifiLocations(data))

.catch(error => console.error("Error fetching Wi-Fi data:", error));

}, []);

return (

<div className="tang1beta">

<div className="map-container">

<img src={image} alt="Tang 1 Beta" className="map-image" />

{wifiLocations.map((wifi, index) => (

<div

key={index}

className="wifi-marker"

style={{ top: wifi.top, left: wifi.left }}

>

<Wifi className="wifi-icon" size={28} color="green" />

<div className="wifi-name">{wifi.name}</div>

</div>

))}

</div>

</div>

);

}

**5. Kiểm tra & Chạy dự án**

**Chạy Backend**

sh

CopyEdit

node server.js

hoặc

sh

CopyEdit

npx nodemon server.js

Kiểm tra API bằng Postman hoặc truy cập:  
🔗 **http://localhost:5000/api/wifi**

**Chạy Frontend**

Di chuyển đến thư mục React và chạy:

sh

CopyEdit

npm start

Trang React sẽ hiển thị sơ đồ với dữ liệu Wi-Fi từ SQL Server.

**✅ Tóm tắt**

* **Backend**: Node.js + Express + mssql để kết nối SQL Server
* **Database**: Tạo bảng WifiLocations
* **Frontend**: Fetch API để lấy danh sách Wi-Fi từ backend

🔥 **Giờ bạn có thể thêm, sửa, xóa Wi-Fi từ SQL Server và hiển thị trên React!** 🚀  
Nếu có lỗi, gửi log để mình hỗ trợ nhé! 😊

**You said:**

trong sql server tạo sao;

**ChatGPT said:**

Bạn muốn tạo cơ sở dữ liệu và bảng trong SQL Server đúng không?  
Chạy các lệnh sau trong **SQL Server Management Studio (SSMS)** hoặc **Azure Data Studio**:

**1. Tạo Cơ Sở Dữ Liệu**

sql

CopyEdit

CREATE DATABASE wifi\_locations;

GO

USE wifi\_locations;

GO

**2. Tạo Bảng Lưu Thông Tin Wi-Fi**

sql

CopyEdit

CREATE TABLE WifiLocations (

id INT IDENTITY(1,1) PRIMARY KEY,

name NVARCHAR(255) NOT NULL,

top VARCHAR(10) NOT NULL,

left VARCHAR(10) NOT NULL

);

GO

**3. Thêm Dữ Liệu Mẫu**

sql

CopyEdit

INSERT INTO WifiLocations (name, top, left) VALUES

('AP-BT-Phong AI', '17%', '15%'),

('AP-BT-IT-ACP', '17%', '37%'),

('Phòng họp', '15%', '62%');

GO

Sau khi tạo xong, bạn có thể chạy:

sql

CopyEdit

SELECT \* FROM WifiLocations;

để kiểm tra dữ liệu.

Nếu gặp lỗi hoặc cần hỗ trợ gì thêm, cứ báo mình nhé! 🚀

**You said:**

tạo cái này trong sql server Tạo file .env để lưu thông tin kết nối: ini Copy Edit DB\_USER=sa DB\_PASSWORD=your\_password DB\_SERVER=localhost DB\_DATABASE=wifi\_locations DB\_PORT=1433

**ChatGPT said:**

File .env không được tạo trực tiếp trong SQL Server mà bạn phải tạo nó trong thư mục **backend** của dự án Node.js.

**Cách tạo file .env trong backend**

1. Mở thư mục **backend** của bạn.
2. Tạo một file mới tên là .env.
3. Thêm nội dung sau vào file .env:

ini

CopyEdit

DB\_USER=sa

DB\_PASSWORD=your\_password

DB\_SERVER=localhost

DB\_DATABASE=wifi\_locations

DB\_PORT=1433

**Giải thích các thông số:**

* DB\_USER: Tài khoản SQL Server (mặc định là sa nếu bạn dùng SQL Server Authentication).
* DB\_PASSWORD: Mật khẩu của tài khoản sa.
* DB\_SERVER: Địa chỉ của SQL Server, nếu chạy cục bộ thì để localhost.
* DB\_DATABASE: Tên database, ở đây là wifi\_locations.
* DB\_PORT: Cổng kết nối, mặc định của SQL Server là **1433**.

Sau khi tạo .env, trong code Node.js, bạn sử dụng thư viện **dotenv** để đọc thông tin kết nối:

javascript

CopyEdit

require("dotenv").config();

const sql = require("mssql");

const config = {

user: process.env.DB\_USER,

password: process.env.DB\_PASSWORD,

server: process.env.DB\_SERVER,

database: process.env.DB\_DATABASE,

port: parseInt(process.env.DB\_PORT),

options: {

encrypt: false,

trustServerCertificate: true,

},

};

const poolPromise = new sql.ConnectionPool(config)

.connect()

.then(pool => {

console.log("✅ Connected to SQL Server");

return pool;

})

.catch(err => console.error("❌ Database Connection Failed: ", err));

module.exports = { sql, poolPromise };

Bạn có thể chạy backend bằng:

sh

CopyEdit

node server.js

hoặc dùng **nodemon**:

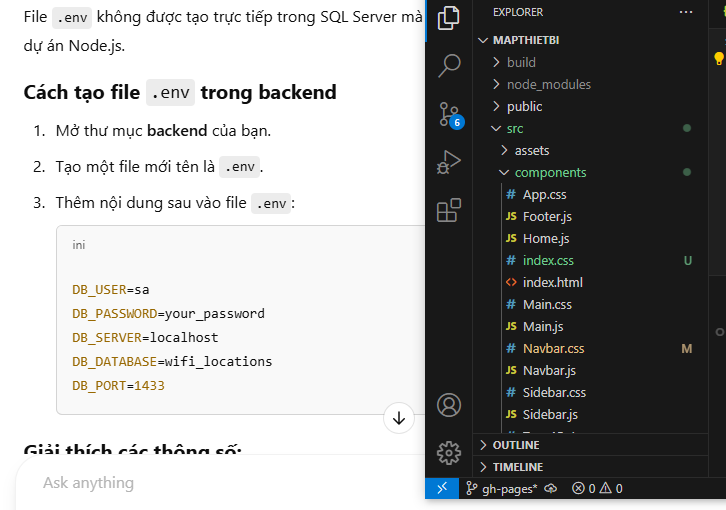
sh

CopyEdit

npx nodemon server.js

Nếu có lỗi kết nối, mình có thể giúp bạn debug! 🚀

**You said:**



nên đặt file .env ở đâu

**ChatGPT said:**

Bạn nên đặt file .env trong thư mục **backend** của dự án.

Hiện tại, trong hình của bạn chỉ hiển thị thư mục **frontend** (src, components, public), nhưng chưa thấy thư mục backend. Nếu bạn chưa có thư mục **backend**, hãy làm như sau:

**1. Tạo thư mục backend**

* Trong VS Code, nhấn **Ctrl + Shift + P**, chọn **New Folder**, đặt tên là backend.

**2. Thêm file .env vào thư mục backend**

* Mở thư mục backend và tạo file mới tên .env.
* Thêm nội dung:

ini

CopyEdit

DB\_USER=sa

DB\_PASSWORD=your\_password

DB\_SERVER=localhost

DB\_DATABASE=wifi\_locations

DB\_PORT=1433

**3. Cấu trúc thư mục đúng**

pgsql

CopyEdit

mapthietbi/

│── backend/ <-- Đặt file .env ở đây

│ ├── server.js

│ ├── .env

│ ├── package.json

│── src/ <-- Đây là frontend (React)

│── public/

│── node\_modules/

Sau khi thêm .env, kiểm tra lại bằng cách chạy:

sh

CopyEdit

cd backend

node server.js

Nếu bạn gặp lỗi gì, cứ báo mình nhé! 🚀

**You said:**

PS C:\ReactJS\mapthietbi> cd backend PS C:\ReactJS\mapthietbi\backend> node server.js node:internal/modules/cjs/loader:1228 throw err; ^ Error: Cannot find module 'C:\ReactJS\mapthietbi\backend\server.js' at Function.\_resolveFilename (node:internal/modules/cjs/loader:1225:15) at Function.\_load (node:internal/modules/cjs/loader:1055:27) at TracingChannel.traceSync (node:diagnostics\_channel:322:14) at wrapModuleLoad (node:internal/modules/cjs/loader:220:24) at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run\_main:170:5) at node:internal/main/run\_main\_module:36:49 { code: 'MODULE\_NOT\_FOUND', requireStack: [] } Node.js v22.14.0

**ChatGPT said:**

Lỗi này báo rằng Node.js không tìm thấy file server.js trong thư mục backend. Bạn hãy kiểm tra và thực hiện các bước sau:

**1. Kiểm tra thư mục backend**

* Mở **VS Code** hoặc **File Explorer**, vào thư mục C:\ReactJS\mapthietbi\backend
* Kiểm tra xem có file server.js hay không.
* Nếu không có, bạn cần tạo nó.

**2. Tạo file server.js**

* Nếu chưa có file, hãy tạo file **server.js** trong thư mục **backend** và dán code sau:

javascript

CopyEdit

require("dotenv").config();

const express = require("express");

const sql = require("mssql");

const app = express();

app.use(express.json());

const config = {

user: process.env.DB\_USER,

password: process.env.DB\_PASSWORD,

server: process.env.DB\_SERVER,

database: process.env.DB\_DATABASE,

port: parseInt(process.env.DB\_PORT),

options: {

encrypt: false,

trustServerCertificate: true,

},

};

// Kết nối SQL Server

const poolPromise = new sql.ConnectionPool(config)

.connect()

.then(pool => {

console.log("✅ Connected to SQL Server");

return pool;

})

.catch(err => console.error("❌ Database Connection Failed: ", err));

// API lấy danh sách Wi-Fi

app.get("/wifi", async (req, res) => {

try {

const pool = await poolPromise;

const result = await pool.request().query("SELECT \* FROM WifiLocations");

res.json(result.recordset);

} catch (err) {

res.status(500).json({ error: err.message });

}

});

// Chạy server

const PORT = 5000;

app.listen(PORT, () => {

console.log(`🚀 Server is running on http://localhost:${PORT}`);

});

**3. Cài đặt thư viện**

Chạy lệnh sau trong thư mục backend để cài đặt các thư viện cần thiết:

sh

CopyEdit

npm init -y

npm install express mssql dotenv cors

**4. Kiểm tra lại file .env**

File .env trong thư mục backend phải đúng như sau:

ini

CopyEdit

DB\_USER=sa

DB\_PASSWORD=your\_password

DB\_SERVER=localhost

DB\_DATABASE=wifi\_locations

DB\_PORT=1433

**5. Chạy lại server**

Sau khi tạo xong file server.js, chạy lệnh:

sh

CopyEdit

node server.js

Hoặc nếu bạn dùng **nodemon**:

sh

CopyEdit

npx nodemon server.js

Nếu có lỗi gì, cứ gửi mình xem nhé! 🚀

**You said:**

Tạo file server.js từ .env sau DB\_USER=sa DB\_PASSWORD=D@ihocfpt2025 DB\_SERVER=localhost DB\_DATABASE=tang1beta DB\_PORT=1433

**ChatGPT said:**

Dưới đây là file server.js sử dụng thông tin từ file .env mà bạn đã cung cấp.

**📌 Các bước thực hiện**

1. **Tạo file server.js** trong thư mục backend.
2. **Cài đặt thư viện**:

sh

CopyEdit

npm init -y

npm install express mssql dotenv cors

1. **Chạy server**:

sh

CopyEdit

node server.js

**📄 server.js**

javascript

CopyEdit

require("dotenv").config();

const express = require("express");

const sql = require("mssql");

const cors = require("cors");

const app = express();

app.use(express.json());

app.use(cors());

const config = {

user: process.env.DB\_USER,

password: process.env.DB\_PASSWORD,

server: process.env.DB\_SERVER,

database: process.env.DB\_DATABASE,

port: parseInt(process.env.DB\_PORT),

options: {

encrypt: false, // Nếu bạn dùng Azure, hãy đặt là true

trustServerCertificate: true,

},

};

// Kết nối SQL Server

const poolPromise = new sql.ConnectionPool(config)

.connect()

.then(pool => {

console.log("✅ Kết nối SQL Server thành công!");

return pool;

})

.catch(err => console.error("❌ Lỗi kết nối SQL Server: ", err));

// Lấy danh sách Wi-Fi

app.get("/wifi", async (req, res) => {

try {

const pool = await poolPromise;

const result = await pool.request().query("SELECT \* FROM WifiLocations");

res.json(result.recordset);

} catch (err) {

res.status(500).json({ error: err.message });

}

});

// Thêm một Wi-Fi mới

app.post("/wifi", async (req, res) => {

try {

const { name, top, left } = req.body;

const pool = await poolPromise;

await pool.request()

.input("name", sql.NVarChar, name)

.input("top", sql.NVarChar, top)

.input("left", sql.NVarChar, left)

.query("INSERT INTO WifiLocations (name, top, left) VALUES (@name, @top, @left)");

res.status(201).json({ message: "Wi-Fi đã được thêm!" });

} catch (err) {

res.status(500).json({ error: err.message });

}

});

// Cập nhật Wi-Fi

app.put("/wifi/:id", async (req, res) => {

try {

const { name, top, left } = req.body;

const { id } = req.params;

const pool = await poolPromise;

await pool.request()

.input("id", sql.Int, id)

.input("name", sql.NVarChar, name)

.input("top", sql.NVarChar, top)

.input("left", sql.NVarChar, left)

.query("UPDATE WifiLocations SET name = @name, top = @top, left = @left WHERE id = @id");

res.json({ message: "Wi-Fi đã được cập nhật!" });

} catch (err) {

res.status(500).json({ error: err.message });

}

});

// Xóa Wi-Fi

app.delete("/wifi/:id", async (req, res) => {

try {

const { id } = req.params;

const pool = await poolPromise;

await pool.request().input("id", sql.Int, id).query("DELETE FROM WifiLocations WHERE id = @id");

res.json({ message: "Wi-Fi đã được xóa!" });

} catch (err) {

res.status(500).json({ error: err.message });

}

});

// Chạy server

const PORT = 5000;

app.listen(PORT, () => {

console.log(`🚀 Server đang chạy tại http://localhost:${PORT}`);

});

**🎯 Kiểm tra API**

Sau khi chạy node server.js, bạn có thể kiểm tra API bằng Postman hoặc trình duyệt:

* **Lấy danh sách Wi-Fi**
  + URL: http://localhost:5000/wifi
  + Method: GET
* **Thêm Wi-Fi mới**
  + URL: http://localhost:5000/wifi
  + Method: POST
  + Body (JSON):

json

CopyEdit

{

"name": "AP-Phong AI",

"top": "17%",

"left": "15%"

}

* **Cập nhật Wi-Fi**
  + URL: http://localhost:5000/wifi/1
  + Method: PUT
  + Body (JSON):

json

CopyEdit

{

"name": "AP-Phong AI - Updated",

"top": "18%",

"left": "16%"

}

* **Xóa Wi-Fi**
  + URL: http://localhost:5000/wifi/1
  + Method: DELETE

Nếu có lỗi gì thì gửi mình xem nhé! 🚀

**You said:**

require("dotenv").config(); const express = require("express"); const sql = require("mssql"); const cors = require("cors"); const app = express(); app.use(express.json()); app.use(cors()); const config = { user: process.env.DB\_USER, password: process.env.DB\_PASSWORD, server: process.env.DB\_SERVER, database: process.env.DB\_DATABASE, port: parseInt(process.env.DB\_PORT), options: { encrypt: false, // Nếu bạn dùng Azure, hãy đặt là true trustServerCertificate: true, }, }; // Kết nối SQL Server const poolPromise = new sql.ConnectionPool(config) .connect() .then(pool => { console.log("✅ Kết nối SQL Server thành công!"); return pool; }) .catch(err => console.error("❌ Lỗi kết nối SQL Server: ", err)); // Lấy danh sách Wi-Fi app.get("/wifi", async (req, res) => { try { const pool = await poolPromise; const result = await pool.request().query("SELECT \* FROM WifiLocations"); res.json(result.recordset); } catch (err) { res.status(500).json({ error: err.message }); } }); // Thêm một Wi-Fi mới app.post("/wifi", async (req, res) => { try { const { name, top, left } = req.body; const pool = await poolPromise; await pool.request() .input("name", sql.NVarChar, name) .input("top", sql.NVarChar, top) .input("left", sql.NVarChar, left) .query("INSERT INTO WifiLocations (name, top, left) VALUES (@name, @top, @left)"); res.status(201).json({ message: "Wi-Fi đã được thêm!" }); } catch (err) { res.status(500).json({ error: err.message }); } }); // Cập nhật Wi-Fi app.put("/wifi/:id", async (req, res) => { try { const { name, top, left } = req.body; const { id } = req.params; const pool = await poolPromise; await pool.request() .input("id", sql.Int, id) .input("name", sql.NVarChar, name) .input("top", sql.NVarChar, top) .input("left", sql.NVarChar, left) .query("UPDATE WifiLocations SET name = @name, top = @top, left = @left WHERE id = @id"); res.json({ message: "Wi-Fi đã được cập nhật!" }); } catch (err) { res.status(500).json({ error: err.message }); } }); // Xóa Wi-Fi app.delete("/wifi/:id", async (req, res) => { try { const { id } = req.params; const pool = await poolPromise; await pool.request().input("id", sql.Int, id).query("DELETE FROM WifiLocations WHERE id = @id"); res.json({ message: "Wi-Fi đã được xóa!" }); } catch (err) { res.status(500).json({ error: err.message }); } }); // Chạy server const PORT = 5000; app.listen(PORT, () => { console.log(🚀 Server đang chạy tại http://localhost:${PORT}); });

**ChatGPT said:**

Mã server.js của bạn đã đúng về mặt cấu trúc. Nếu có lỗi khi chạy, hãy kiểm tra các bước sau:

**1️⃣ Kiểm tra thư mục backend**

Bạn phải chắc chắn rằng file .env đã được tạo trong thư mục backend cùng với server.js.

pgsql

CopyEdit

mapthietbi/

│── backend/

│ │── .env

│ │── server.js

│ │── package.json

│ └── node\_modules/

└── src/

**2️⃣ Cài đặt các thư viện cần thiết**

Chạy lệnh sau trong thư mục backend:

sh

CopyEdit

npm install express mssql dotenv cors

**3️⃣ Kiểm tra nội dung file .env**

Mở file .env trong thư mục backend và kiểm tra lại nội dung:

ini

CopyEdit

DB\_USER=sa

DB\_PASSWORD=D@ihocfpt2025

DB\_SERVER=localhost

DB\_DATABASE=tang1beta

DB\_PORT=1433

**4️⃣ Chạy server**

Sau khi đã kiểm tra đầy đủ, bạn chạy:

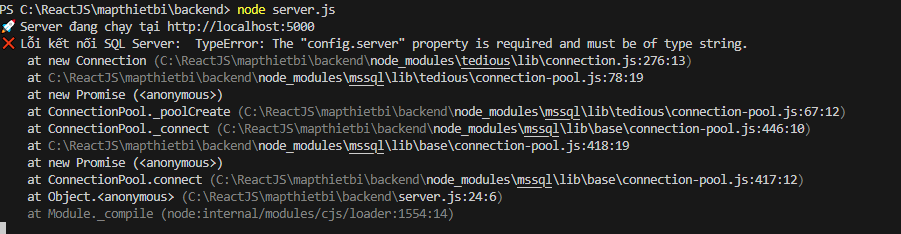
sh

CopyEdit

node server.js

Nếu vẫn lỗi, hãy gửi thông báo lỗi mình kiểm tra giúp bạn! 🚀

**You said:**



**ChatGPT said:**

Lỗi này xuất phát từ việc cấu hình server trong file .env hoặc cách đọc giá trị từ file .env. Cách khắc phục:

**1️⃣ Kiểm tra file .env**

Mở file .env trong thư mục backend và kiểm tra lại:

ini

CopyEdit

DB\_USER=sa

DB\_PASSWORD=D@ihocfpt2025

DB\_SERVER=localhost

DB\_DATABASE=tang1beta

DB\_PORT=1433

✅ **Lưu ý:** Không có khoảng trắng hoặc ký tự đặc biệt trong tên biến.

**2️⃣ Kiểm tra cách đọc biến môi trường trong server.js**

Thêm dòng này vào đầu file server.js để đảm bảo dotenv được tải trước:

js

CopyEdit

require("dotenv").config();

console.log("Database Config:", process.env.DB\_SERVER); // Kiểm tra giá trị

Chạy lại server:

sh

CopyEdit

node server.js

Nếu undefined, nghĩa là file .env chưa được tải đúng.

**3️⃣ Cập nhật lại cấu hình SQL**

Sửa lại config trong server.js:

js

CopyEdit

const config = {

user: process.env.DB\_USER,

password: process.env.DB\_PASSWORD,

server: process.env.DB\_SERVER.toString(), // Ép kiểu về string

database: process.env.DB\_DATABASE,

port: Number(process.env.DB\_PORT), // Đảm bảo cổng là số

options: {

encrypt: false,

trustServerCertificate: true,

},

};

**4️⃣ Xóa cache và chạy lại**

Sau khi sửa xong, chạy các lệnh sau:

sh

CopyEdit

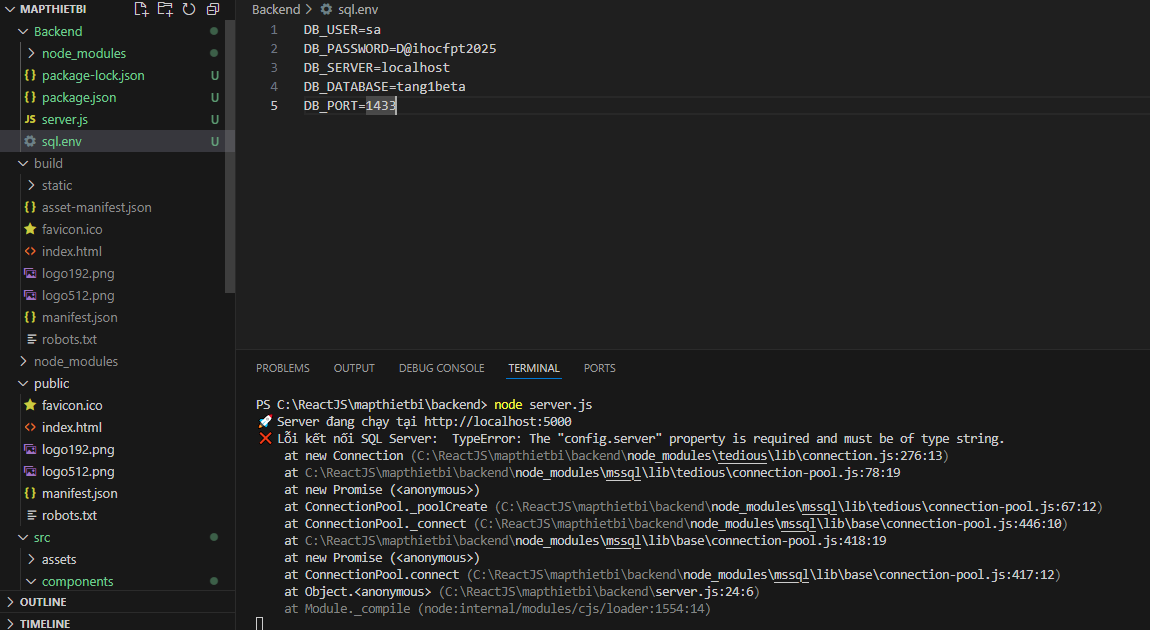
rm -rf node\_modules package-lock.json

npm install

node server.js

Nếu vẫn lỗi, bạn gửi lại thông báo lỗi mới nhé! 🚀

**You said:**



**ChatGPT said:**

Bạn đang đặt file .env thành sql.env, nhưng mặc định thư viện dotenv chỉ tìm file có tên là .env.

**Cách khắc phục:**

**🔹 1. Đổi tên file**

Bạn cần đổi tên sql.env thành .env trong thư mục backend.

**🔹 2. Hoặc cập nhật code để đọc sql.env**

Nếu bạn muốn giữ nguyên tên sql.env, hãy cập nhật server.js:

js

CopyEdit

require("dotenv").config({ path: "sql.env" });

Sau đó, khởi động lại server:

sh

CopyEdit

node server.js

**Lưu ý:** Sau khi sửa, nếu vẫn bị lỗi, hãy kiểm tra lại bằng:

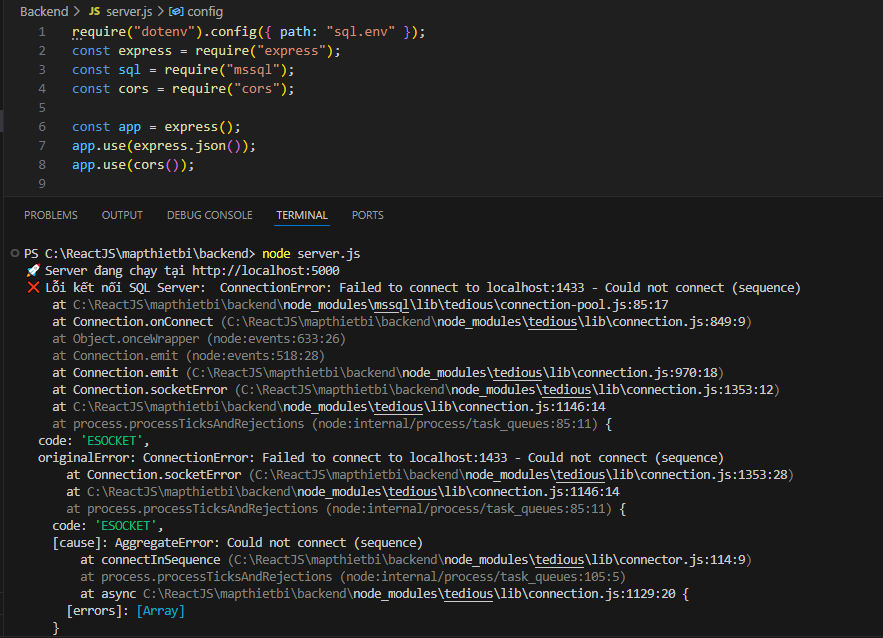
js

CopyEdit

console.log("DB\_SERVER:", process.env.DB\_SERVER);

Nếu vẫn undefined, tức là file .env chưa được đọc đúng.

**You said:**



require("dotenv").config({ path: "sql.env" }); const express = require("express"); const sql = require("mssql"); const cors = require("cors"); const app = express(); app.use(express.json()); app.use(cors()); const config = { user: process.env.DB\_USER, password: process.env.DB\_PASSWORD, server: process.env.DB\_SERVER, database: process.env.DB\_DATABASE, port: parseInt(process.env.DB\_PORT), options: { encrypt: false, // Nếu bạn dùng Azure, hãy đặt là true trustServerCertificate: true, }, }; // Kết nối SQL Server const poolPromise = new sql.ConnectionPool(config) .connect() .then(pool => { console.log("✅ Kết nối SQL Server thành công!"); return pool; }) .catch(err => console.error("❌ Lỗi kết nối SQL Server: ", err)); // Lấy danh sách Wi-Fi app.get("/wifi", async (req, res) => { try { const pool = await poolPromise; const result = await pool.request().query("SELECT \* FROM WifiLocations"); res.json(result.recordset); } catch (err) { res.status(500).json({ error: err.message }); } }); // Thêm một Wi-Fi mới app.post("/wifi", async (req, res) => { try { const { name, top, left } = req.body; const pool = await poolPromise; await pool.request() .input("name", sql.NVarChar, name) .input("top", sql.NVarChar, top) .input("left", sql.NVarChar, left) .query("INSERT INTO WifiLocations (name, top, left) VALUES (@name, @top, @left)"); res.status(201).json({ message: "Wi-Fi đã được thêm!" }); } catch (err) { res.status(500).json({ error: err.message }); } }); // Cập nhật Wi-Fi app.put("/wifi/:id", async (req, res) => { try { const { name, top, left } = req.body; const { id } = req.params; const pool = await poolPromise; await pool.request() .input("id", sql.Int, id) .input("name", sql.NVarChar, name) .input("top", sql.NVarChar, top) .input("left", sql.NVarChar, left) .query("UPDATE WifiLocations SET name = @name, top = @top, left = @left WHERE id = @id"); res.json({ message: "Wi-Fi đã được cập nhật!" }); } catch (err) { res.status(500).json({ error: err.message }); } }); // Xóa Wi-Fi app.delete("/wifi/:id", async (req, res) => { try { const { id } = req.params; const pool = await poolPromise; await pool.request().input("id", sql.Int, id).query("DELETE FROM WifiLocations WHERE id = @id"); res.json({ message: "Wi-Fi đã được xóa!" }); } catch (err) { res.status(500).json({ error: err.message }); } }); // Chạy server const PORT = 5000; app.listen(PORT, () => { console.log(🚀 Server đang chạy tại http://localhost:${PORT}); });

**ChatGPT said:**

Lỗi **"Failed to connect to localhost:1433 - Could not connect (sequence)"** cho thấy Node.js không thể kết nối đến SQL Server. Hãy thử các cách khắc phục sau:

**🔹 1. Kiểm tra SQL Server có đang chạy không**

Chạy **SQL Server Configuration Manager** → **SQL Server Services**  
→ Đảm bảo **SQL Server (MSSQLSERVER)** và **SQL Server Browser** đang **Running**.  
Nếu không, nhấp chuột phải và chọn **Start**.

**🔹 2. Kiểm tra cổng lắng nghe của SQL Server**

Mặc định, SQL Server sử dụng cổng 1433, nhưng có thể bị thay đổi.

**Cách kiểm tra:**

1. Mở **SQL Server Configuration Manager**.
2. Điều hướng đến **SQL Server Network Configuration** → **Protocols for MSSQLSERVER**.
3. Đảm bảo **TCP/IP** đã được **Enabled**.
4. Nhấp đúp vào **TCP/IP**, vào tab **IP Addresses**, kéo xuống **IPAll**.
5. Kiểm tra cổng trong **TCP Dynamic Ports** và **TCP Port**.
   * Nếu **TCP Dynamic Ports** có giá trị (ví dụ: 49172), hãy dùng nó thay vì 1433 trong .env.

Cập nhật .env:

ini

CopyEdit

DB\_PORT=49172

Rồi chạy lại server.

**🔹 3. Kiểm tra Firewall**

Windows Firewall có thể chặn kết nối **TCP 1433**.  
Thử **Tắt tạm thời Firewall** hoặc **mở cổng 1433**.

**🔹 4. Kiểm tra đăng nhập SQL Server**

Bạn đang sử dụng tài khoản **sa** (user: sa). Nếu tài khoản này bị vô hiệu hóa hoặc mật khẩu sai, kết nối sẽ thất bại.

**Cách kiểm tra:**

1. Mở **SQL Server Management Studio (SSMS)**.
2. Đăng nhập với:
   * **Server name:** localhost
   * **Authentication:** SQL Server Authentication
   * **Login:** sa
   * **Password:** [mật khẩu trong .env]
3. Nếu đăng nhập thất bại:
   * Chạy lệnh sau để bật tài khoản **sa**:

sql

CopyEdit

ALTER LOGIN sa ENABLE;

ALTER LOGIN sa WITH PASSWORD = 'MậtKhẩuMới';

* + Đảm bảo **Authentication Mode** là SQL Server and Windows Authentication mode.  
    Vào **Security** → **Right-click SQL Server** → **Properties** → **Security**.

Sau khi thực hiện các bước trên, chạy lại server.js:

sh

CopyEdit

node server.js

Nếu vẫn lỗi, báo lại mình để hỗ trợ tiếp nhé! 🚀

**You said:**

chưa tạo và chưa kết nối đến sql server, hãy hướng dẫn tôi

**ChatGPT said:**

OK! Mình sẽ hướng dẫn bạn tạo và kết nối SQL Server từ đầu. 🚀

**🔹 1. Cài đặt SQL Server & SQL Server Management Studio (SSMS)**

Trước tiên, nếu bạn chưa cài đặt **SQL Server** và **SSMS**, hãy tải về từ:

* **SQL Server**: [Tải tại đây](https://www.microsoft.com/en-us/sql-server/sql-server-downloads)
* **SSMS (SQL Server Management Studio)**: Tải tại đây

Sau khi cài đặt xong, mở **SQL Server Configuration Manager** và đảm bảo SQL Server đã **chạy**.

**🔹 2. Tạo Cơ Sở Dữ Liệu (Database)**

**Cách 1: Dùng SSMS**

1. Mở **SQL Server Management Studio (SSMS)**.
2. Chọn **Authentication: Windows Authentication** → **Connect**.
3. Trong Object Explorer, chuột phải vào **Databases** → **New Database...**.
4. Đặt tên Database là tang1beta (theo file .env của bạn) → **OK**.

**Cách 2: Dùng SQL Query**

Nếu bạn thích dùng lệnh SQL, hãy mở **New Query** và chạy:

sql

CopyEdit

CREATE DATABASE tang1beta;

GO

**🔹 3. Tạo Bảng Lưu Thông Tin Wi-Fi**

Chạy đoạn SQL sau để tạo bảng WifiLocations:

sql

CopyEdit

USE tang1beta;

GO

CREATE TABLE WifiLocations (

id INT IDENTITY(1,1) PRIMARY KEY,

name NVARCHAR(255) NOT NULL,

top NVARCHAR(50),

left NVARCHAR(50)

);

GO

**🔹 4. Kích Hoạt Đăng Nhập SQL Authentication**

Nếu bạn muốn dùng tài khoản sa, hãy bật chế độ **SQL Server Authentication**:

1. Trong SSMS, click chuột phải vào **Server** → **Properties**.
2. Chọn tab **Security** → **Authentication Mode: SQL Server and Windows Authentication mode** → **OK**.
3. Trong Object Explorer, mở **Security** → **Logins** → Chuột phải vào **sa** → **Properties**.
4. Đặt mật khẩu (D@ihocfpt2025 theo file .env của bạn).
5. Bật quyền truy cập:

sql

CopyEdit

ALTER LOGIN sa ENABLE;

ALTER LOGIN sa WITH PASSWORD = 'D@ihocfpt2025';

ALTER LOGIN sa WITH CHECK\_POLICY = OFF;

1. **Restart SQL Server**.

**🔹 5. Bật TCP/IP & Cấu Hình Cổng Kết Nối**

**Kiểm tra cổng 1433**

1. Mở **SQL Server Configuration Manager**.
2. Mở **SQL Server Network Configuration** → **Protocols for MSSQLSERVER**.
3. Đảm bảo **TCP/IP** đã **Enabled**.
4. Click đúp vào **TCP/IP**, vào tab **IP Addresses**, kéo xuống **IPAll**.
5. Đảm bảo:
   * **TCP Dynamic Ports** = *Trống* (xóa nếu có).
   * **TCP Port** = 1433.
6. **Restart SQL Server**.

**🔹 6. Kiểm Tra Kết Nối Từ Node.js**

Giờ bạn có thể chạy lại **server.js** với đoạn mã này:

javascript

CopyEdit

require("dotenv").config({ path: "sql.env" });

const sql = require("mssql");

const config = {

user: process.env.DB\_USER,

password: process.env.DB\_PASSWORD,

server: process.env.DB\_SERVER,

database: process.env.DB\_DATABASE,

port: parseInt(process.env.DB\_PORT),

options: {

encrypt: false,

trustServerCertificate: true,

},

};

// Kết nối SQL Server

const poolPromise = new sql.ConnectionPool(config)

.connect()

.then(pool => {

console.log("✅ Kết nối SQL Server thành công!");

return pool;

})

.catch(err => console.error("❌ Lỗi kết nối SQL Server: ", err));

Chạy thử:

sh

CopyEdit

node server.js

Nếu thấy **"✅ Kết nối SQL Server thành công!"**, bạn đã làm đúng! 🎉

**🔹 7. Kiểm Tra API**

Mở trình duyệt và truy cập:

bash

CopyEdit

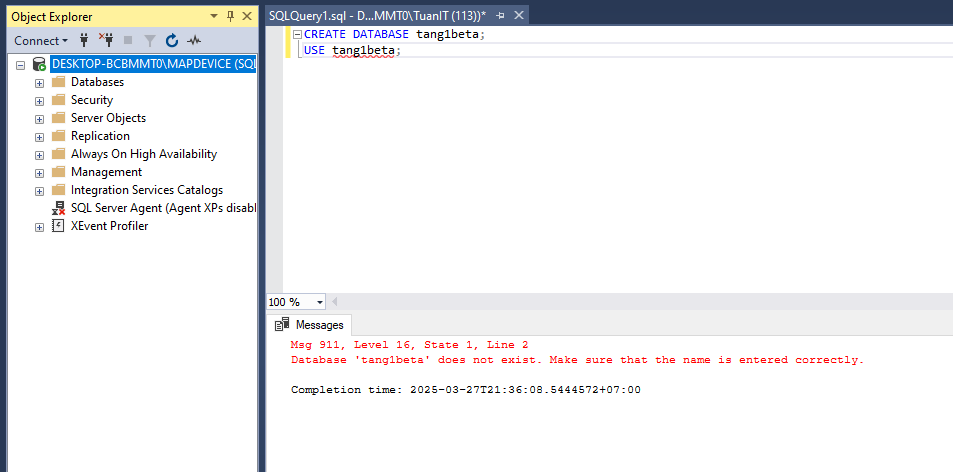
http://localhost:5000/wifi

Nếu trả về danh sách Wi-Fi, bạn đã kết nối thành công! 🎯

Cần hỗ trợ thêm thì cứ hỏi nhé! 😃

4o

**You said:**



**ChatGPT said:**

Bạn đang gặp lỗi **"Database 'tang1beta' does not exist."** khi chạy lệnh USE tang1beta;. Điều này có thể do:

1. **Database chưa được tạo** (câu lệnh CREATE DATABASE tang1beta; chưa chạy thành công).
2. **Database tạo bị lỗi hoặc không lưu**.
3. **Bạn đang kết nối sai Server hoặc User không có quyền**.

**🔹 Cách khắc phục**

**Bước 1: Kiểm tra lại danh sách Database**

Trong SSMS:

1. Click vào **Databases** → Nhấn **Refresh**.
2. Nếu **tang1beta** không xuất hiện, nghĩa là database chưa được tạo.

**Bước 2: Chạy lại lệnh tạo Database**

Mở **New Query** và chạy từng dòng một:

sql

CopyEdit

-- Kiểm tra xem database có tồn tại không

SELECT name FROM sys.databases WHERE name = 'tang1beta';

-- Nếu không thấy trong danh sách, chạy lệnh này để tạo database

CREATE DATABASE tang1beta;

GO

Sau khi chạy xong, nhấn **Refresh** Databases để kiểm tra.

**Bước 3: Kiểm tra quyền truy cập**

Nếu vẫn không chạy được, hãy đảm bảo bạn đang đăng nhập với quyền **sa** hoặc tài khoản có quyền quản lý database.

Chạy lệnh này để kiểm tra user:

sql

CopyEdit

SELECT SUSER\_NAME();

Nếu không phải sa, hãy đăng xuất và đăng nhập lại bằng tài khoản **sa**.

**Bước 4: Restart SQL Server**

Nếu sau khi tạo vẫn không thấy database, hãy thử:

1. Mở **SQL Server Configuration Manager**.
2. Restart **SQL Server (MSSQLSERVER)**.

Sau đó, thử lại lệnh:

sql

CopyEdit

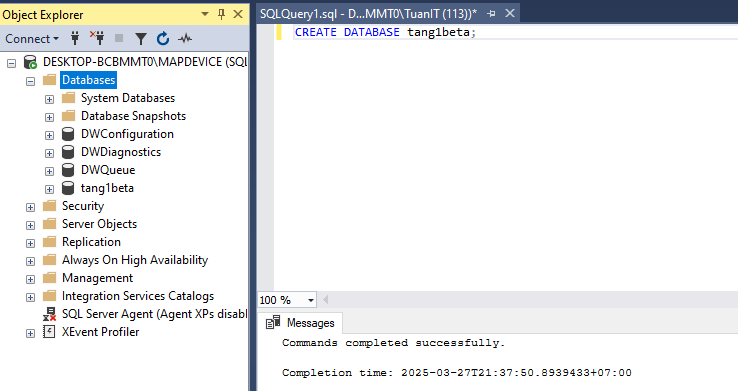
USE tang1beta;

GO

Nếu vẫn lỗi, gửi lại ảnh màn hình **Object Explorer** sau khi làm các bước trên để mình kiểm tra giúp bạn nhé! 🚀

4o

**You said:**



đã tạo nhưng ko thấy database của tang1beta

**ChatGPT said:**

Database **tang1beta** đã được tạo thành công nhưng chưa hiển thị trong **Object Explorer**. Bạn có thể thử các cách sau:

**🔹 Cách 1: Refresh lại danh sách Database**

1. Click chuột phải vào **Databases**.
2. Chọn **Refresh**.
3. Kiểm tra xem **tang1beta** đã xuất hiện chưa.

**🔹 Cách 2: Đóng và mở lại SSMS**

1. Đóng SQL Server Management Studio (SSMS).
2. Mở lại SSMS và kết nối lại với SQL Server.
3. Kiểm tra lại danh sách Databases.

**🔹 Cách 3: Kiểm tra Database có tồn tại không**

Mở **New Query** và chạy lệnh:

sql

CopyEdit

SELECT name FROM sys.databases WHERE name = 'tang1beta';

* Nếu kết quả có **tang1beta**, nghĩa là database đã tồn tại.
* Nếu không có, bạn cần chạy lại lệnh **CREATE DATABASE tang1beta**.

**🔹 Cách 4: Kiểm tra SQL Server Instance**

Nếu vẫn không thấy **tang1beta**, có thể bạn đang kết nối nhầm instance của SQL Server.

1. Nhìn trên **Object Explorer**, kiểm tra xem bạn đang kết nối đến **DESKTOP-BCBMMT0\MAPDEVICE**.
2. Nếu không chắc, thử chạy:

sql

CopyEdit

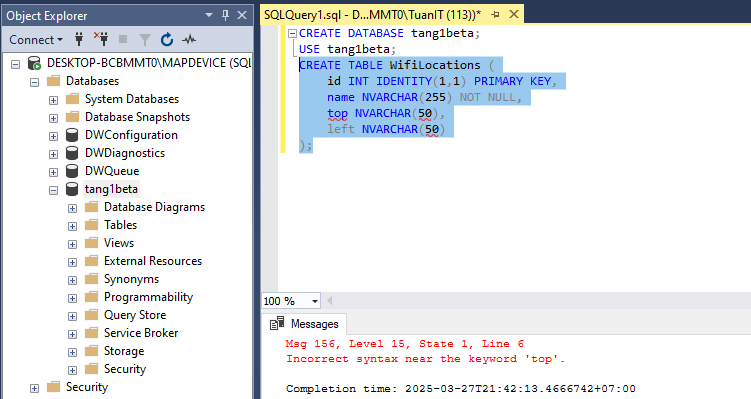
SELECT @@SERVERNAME;

* Nếu tên server không đúng với **DESKTOP-BCBMMT0\MAPDEVICE**, bạn cần kết nối lại đúng server.

Sau khi thử các bước trên, nếu vẫn không thấy **tang1beta**, hãy gửi ảnh màn hình **Object Explorer** sau khi đã refresh. 🚀

4o

**You said:**



**ChatGPT said:**

Lỗi **"Incorrect syntax near the keyword 'top'"** xảy ra do **top** là một từ khóa trong SQL Server. Để khắc phục, bạn cần đổi tên cột hoặc đặt nó trong dấu ngoặc vuông **[]**.

**🔹 Cách sửa lỗi**

Thử chạy lại đoạn SQL sau:

sql

CopyEdit

USE tang1beta;

CREATE TABLE WifiLocations (

id INT IDENTITY(1,1) PRIMARY KEY,

name NVARCHAR(255) NOT NULL,

[top] NVARCHAR(50),

[left] NVARCHAR(50)

);

✅ **Giải thích**:

* [top] và [left] được đặt trong **ngoặc vuông** để tránh xung đột với từ khóa SQL.
* Giữ nguyên kiểu dữ liệu và ràng buộc **PRIMARY KEY**.

Sau khi sửa, chạy lại lệnh trên trong SQL Server Management Studio (SSMS) và kiểm tra lại. 🚀

4o

**You said:**

hướng dẫn tôi 4. Kích Hoạt Đăng Nhập SQL Authentication trong sql server

**ChatGPT said:**

Để kích hoạt **SQL Server Authentication**, bạn cần làm theo các bước sau:

**🔹 Bước 1: Mở SQL Server Management Studio (SSMS)**

1. Mở **SQL Server Management Studio (SSMS)**.
2. Kết nối vào SQL Server bằng **Windows Authentication**.

**🔹 Bước 2: Bật Chế Độ Đăng Nhập SQL Authentication**

1. Trong **Object Explorer**, nhấp chuột phải vào **server** (VD: DESKTOP-XXXXX).
2. Chọn **Properties**.
3. Chuyển sang tab **Security**.
4. Trong phần **Server Authentication**, chọn **SQL Server and Windows Authentication mode**.
5. Nhấn **OK** để lưu lại.

**🔹 Bước 3: Bật Login cho tài khoản sa**

1. Mở **Object Explorer** > **Security** > **Logins**.
2. Nhấp chuột phải vào tài khoản **sa** > **Properties**.
3. Trong tab **General**:
   * Đặt **password** mới.
   * Xác nhận lại **password**.
4. Trong tab **Status**:
   * Đặt **Login** thành **Enabled**.
5. Nhấn **OK**.

**🔹 Bước 4: Khởi Động Lại SQL Server**

Bạn cần **restart SQL Server** để áp dụng thay đổi:

1. Mở **SQL Server Configuration Manager**.
2. Tìm **SQL Server (MSSQLSERVER)**.
3. Nhấp chuột phải, chọn **Restart**.

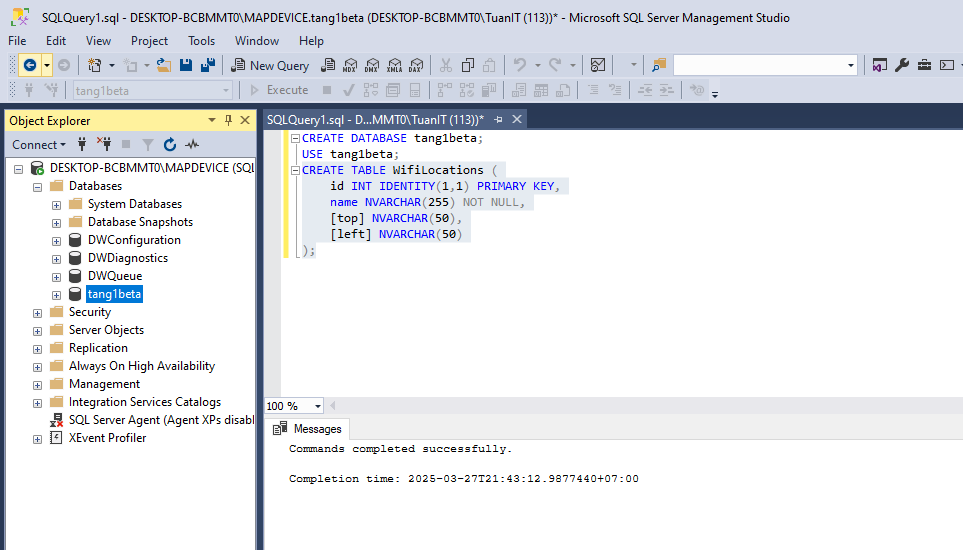
**🔹 Bước 5: Kiểm Tra Đăng Nhập Bằng SQL Authentication**

1. Mở **SSMS**.
2. Chọn **Authentication: SQL Server Authentication**.
3. Nhập:
   * **Login**: sa
   * **Password**: (mật khẩu bạn đã đặt ở Bước 3)
4. Nhấn **Connect**.

Nếu kết nối thành công, bạn đã bật thành công **SQL Server Authentication**! 🎉

4o

**You said:**



hướng dẫn tôi 4. Kích Hoạt Đăng Nhập SQL Authentication trong sql server

**ChatGPT said:**

Bạn làm theo các bước sau để kích hoạt **SQL Server Authentication**:

**Bước 1: Mở SQL Server Management Studio (SSMS)**

1. Mở **SQL Server Management Studio (SSMS)**.
2. Kết nối vào SQL Server bằng **Windows Authentication**.

**Bước 2: Chuyển Đổi Chế Độ Đăng Nhập SQL Authentication**

1. Trong **Object Explorer**, nhấp chuột phải vào **server** của bạn (ví dụ: DESKTOP-BCBMMTO).
2. Chọn **Properties**.
3. Trong cửa sổ **Server Properties**, chọn tab **Security**.
4. Ở phần **Server authentication**, chọn **SQL Server and Windows Authentication mode**.
5. Nhấn **OK** để lưu lại.

**Bước 3: Bật Login Cho Tài Khoản sa**

1. Trong **Object Explorer**, mở rộng **Security** > **Logins**.
2. Nhấp chuột phải vào **sa**, chọn **Properties**.
3. Trong tab **General**:
   * Đặt **password** mới.
   * Xác nhận lại **password**.
4. Trong tab **Status**:
   * Ở phần **Login**, chọn **Enabled**.
5. Nhấn **OK** để lưu.

**Bước 4: Khởi Động Lại SQL Server**

Bạn cần **restart SQL Server** để thay đổi có hiệu lực:

1. Mở **SQL Server Configuration Manager**.
2. Chọn **SQL Server Services**.
3. Tìm **SQL Server (MSSQLSERVER)**.
4. Nhấp chuột phải vào **SQL Server (MSSQLSERVER)**, chọn **Restart**.

**Bước 5: Đăng Nhập Bằng SQL Authentication**

1. Mở **SSMS**.
2. Chọn **Authentication: SQL Server Authentication**.
3. Nhập:
   * **Login**: sa
   * **Password**: (mật khẩu bạn đã đặt ở Bước 3).
4. Nhấn **Connect**.

✔ **Nếu kết nối thành công**, bạn đã bật SQL Server Authentication thành công! 🎉

4o

**You said:**

tôi ko thấy Kết nối vào SQL Server bằng Windows Authentication

**ChatGPT said:**

Bạn có thể thử các bước sau để kết nối vào SQL Server bằng **Windows Authentication** trong SQL Server Management Studio (SSMS):

**1. Mở SQL Server Management Studio (SSMS)**

* Mở **SQL Server Management Studio (SSMS)** trên máy của bạn.

**2. Chọn Kết Nối với SQL Server**

* Khi cửa sổ **Connect to Server** hiện ra:
  + **Server type**: Chọn **Database Engine**.
  + **Server name**:
    - Nếu bạn cài đặt SQL Server trên máy của mình, nhập **(local)** hoặc **.** hoặc **DESKTOP-BCBMMTO** (tên server của bạn).
    - Nếu bạn kết nối từ xa, nhập đúng **tên máy chủ SQL Server**.
  + **Authentication**: Chọn **Windows Authentication**.
  + Nhấn **Connect**.

**3. Nếu Không Thấy Cửa Sổ Kết Nối**

* Nếu bạn đã mở SSMS nhưng không thấy cửa sổ **Connect to Server**, hãy làm như sau:
  1. Vào **File** > **Connect Object Explorer**.
  2. Làm theo hướng dẫn ở **Bước 2**.

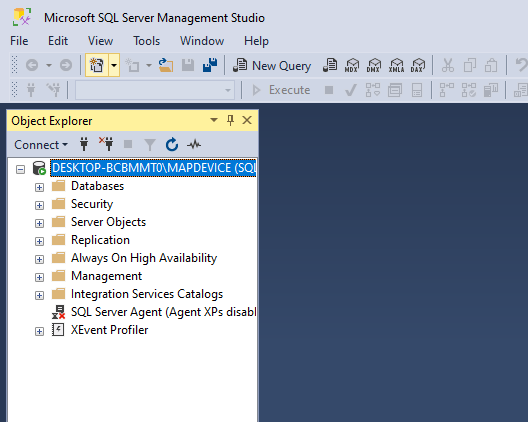
**4. Nếu Không Kết Nối Được**

Nếu **Windows Authentication** không hoạt động:

* **Kiểm tra SQL Server có đang chạy không?**
  1. Mở **SQL Server Configuration Manager**.
  2. Kiểm tra **SQL Server (MSSQLSERVER)** có đang **Running** không.
  3. Nếu không, nhấp chuột phải vào nó và chọn **Start**.
* **Kiểm tra Tên Server**
  1. Nhấn Win + R, nhập services.msc và tìm **SQL Server (MSSQLSERVER)** để xem tên chính xác của server.
* **Nếu vấn đề vẫn chưa giải quyết**, gửi ảnh lỗi để mình hỗ trợ tiếp. 🚀

4o

**You said:**



làm gì tiếp theo

**ChatGPT said:**

Bạn đã kết nối vào SQL Server thành công bằng **Windows Authentication**. Giờ bạn có thể thực hiện các bước sau để kích hoạt **SQL Server Authentication**:

**Bước 1: Mở Cấu Hình Server**

1. Nhấp chuột phải vào tên server **DESKTOP-BCBMMTO\MAPDEVICE**.
2. Chọn **Properties**.

**Bước 2: Kích Hoạt SQL Server Authentication**

1. Trong cửa sổ **Server Properties**, chọn mục **Security**.
2. Ở phần **Server authentication**, chọn **SQL Server and Windows Authentication mode**.
3. Nhấn **OK** để lưu cài đặt.

**Bước 3: Khởi Động Lại SQL Server**

1. Nhấp chuột phải vào tên server **DESKTOP-BCBMMTO\MAPDEVICE**.
2. Chọn **Restart** để áp dụng thay đổi.

**Bước 4: Kiểm Tra Và Tạo Tài Khoản Đăng Nhập SQL Authentication**

1. Mở mục **Security** > **Logins**.
2. Nếu chưa có tài khoản **sa**, bạn có thể tạo mới bằng cách:
   * Nhấp chuột phải vào **Logins** > **New Login**.
   * Chọn **SQL Server Authentication**.
   * Nhập **Login name** (VD: admin).
   * Nhập **Password** và xác nhận lại.
   * Ở mục **Default database**, chọn **master** hoặc database mong muốn.
   * Nhấn **OK** để tạo tài khoản.

**Bước 5: Kích Hoạt Tài Khoản sa (nếu cần)**

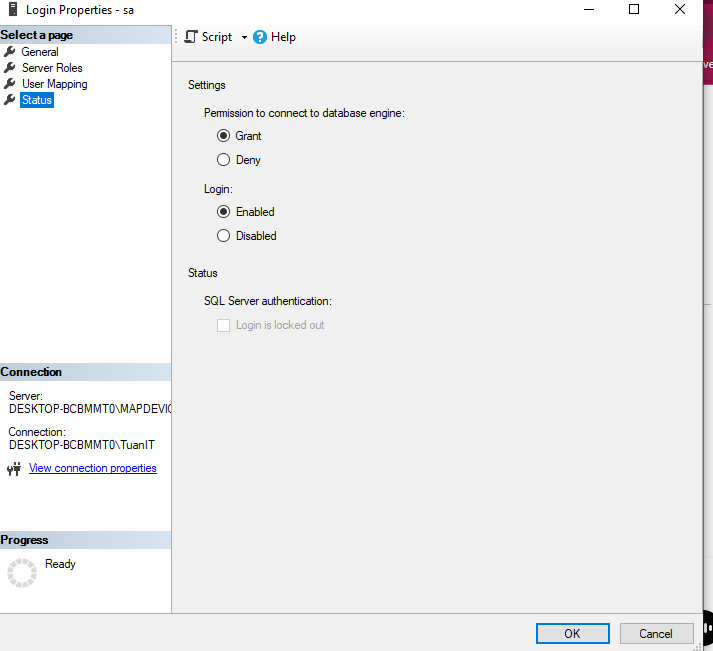
1. Mở **Logins** > Nhấp chuột phải vào **sa** > **Properties**.
2. Đặt **password mới** nếu cần.
3. Trong tab **Status**, đảm bảo **Login** đang ở trạng thái **Enabled**.
4. Nhấn **OK** để lưu.

Sau khi làm xong các bước này, bạn có thể đăng nhập vào SQL Server bằng **SQL Authentication** bằng cách chọn **SQL Server Authentication** và nhập tài khoản/mật khẩu vừa tạo.

Bạn làm thử đi, nếu gặp lỗi gì cứ gửi ảnh mình hỗ trợ tiếp! 🚀

4o

**You said:**



làm gì tiếp theo, cách đăng nhập như thế nào

**ChatGPT said:**

Bạn đã kích hoạt tài khoản sa, bây giờ làm tiếp các bước sau để đăng nhập bằng **SQL Server Authentication**:

**Bước 1: Đặt mật khẩu cho tài khoản sa**

1. Trong cửa sổ **Login Properties - sa**, chọn tab **General**.
2. Nhập mật khẩu mới vào các ô **Password** và **Confirm Password**.
3. Nhấn **OK** để lưu.

**Bước 2: Đăng nhập bằng SQL Authentication**

1. **Đóng SQL Server Management Studio (SSMS) và mở lại**.
2. Trong cửa sổ **Connect to Server**, nhập thông tin:
   * **Server name**: DESKTOP-BCBMMTO\MAPDEVICE (hoặc tên server của bạn).
   * **Authentication**: Chọn **SQL Server Authentication**.
   * **Login**: sa
   * **Password**: Nhập mật khẩu bạn đã đặt.
3. Nhấn **Connect** để đăng nhập.

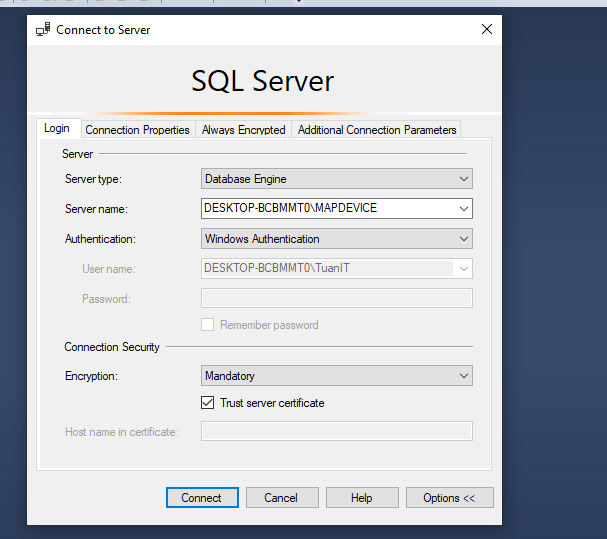
**Bước 3: Kiểm tra lại nếu không đăng nhập được**

* Nếu có lỗi, hãy kiểm tra lại:
  1. Bạn đã chọn **SQL Server and Windows Authentication Mode** chưa?
  2. SQL Server đã được **Restart** sau khi đổi Authentication chưa?
  3. Nhập đúng mật khẩu của tài khoản sa.

Nếu vẫn lỗi, chụp màn hình gửi mình xem nhé! 🚀

4o

**You said:**



đăng nhập bằng tài khoản sa như thế nào

**ChatGPT said:**

Bạn đang đăng nhập bằng **Windows Authentication**, nhưng để đăng nhập bằng **SQL Server Authentication** với tài khoản sa, hãy làm như sau:

**Bước 1: Chuyển sang SQL Server Authentication**

1. Trong cửa sổ **Connect to Server**, tìm mục **Authentication**.
2. Chọn **SQL Server Authentication** thay vì **Windows Authentication**.

**Bước 2: Nhập thông tin tài khoản**

1. **Login**: Nhập sa.
2. **Password**: Nhập mật khẩu bạn đã đặt cho tài khoản sa.

**Bước 3: Kết nối**

1. Nhấn **Connect** để đăng nhập vào SQL Server bằng tài khoản sa.

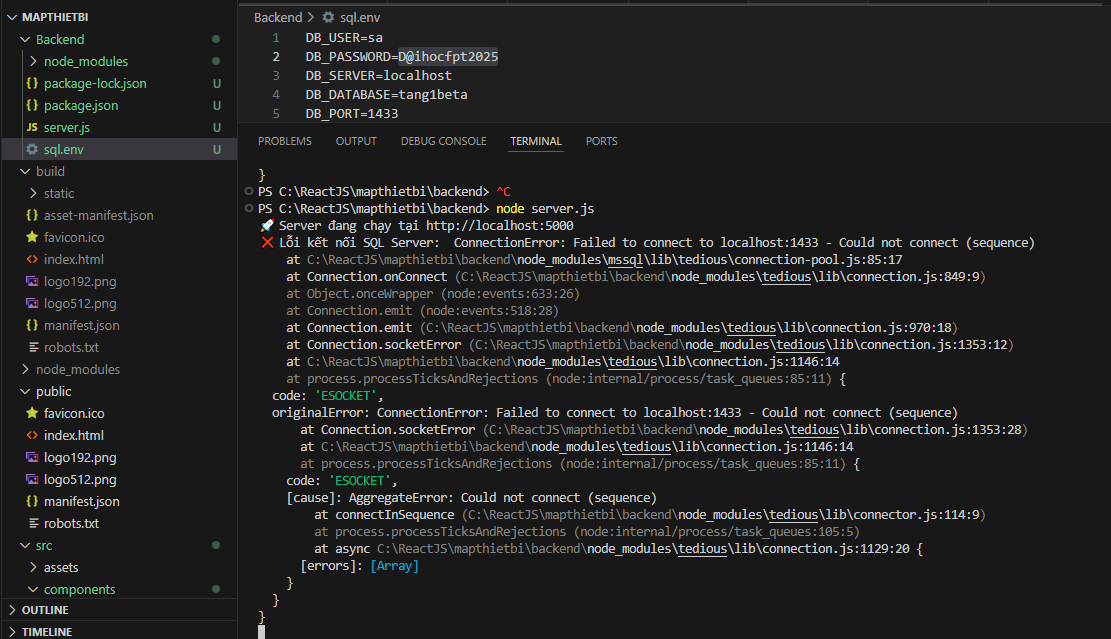
Nếu có lỗi khi đăng nhập, bạn có thể cần:

* Kiểm tra lại mật khẩu.
* Đảm bảo chế độ **SQL Server and Windows Authentication Mode** đã được bật.
* Khởi động lại SQL Server để áp dụng thay đổi.

Hãy thử và báo lại nếu gặp vấn đề nhé! 🚀

4o

**You said:**



không kết nối được đến database trong sql server vừa tạo

**ChatGPT said:**

Lỗi Failed to connect to localhost:1433 - Could not connect (sequence) có thể do một số nguyên nhân sau:

**1. Kiểm tra SQL Server có đang chạy không**

* Mở **SQL Server Configuration Manager**.
* Vào **SQL Server Services**.
* Kiểm tra xem **SQL Server (MSSQLSERVER)** có đang chạy không. Nếu chưa, hãy nhấp chuột phải và chọn **Start**.

**2. Kiểm tra SQL Server có lắng nghe trên cổng 1433 không**

* Vào **SQL Server Configuration Manager**.
* Chọn **SQL Server Network Configuration** > **Protocols for MSSQLSERVER**.
* Bật **TCP/IP** (nhấp chuột phải chọn **Enable**).
* Nhấp đúp vào **TCP/IP**, kéo xuống **IPAll**, kiểm tra **TCP Port** phải là **1433**.
* Khởi động lại SQL Server.

**3. Kiểm tra tường lửa có chặn cổng 1433 không**

* Mở **Windows Defender Firewall**.
* Chọn **Advanced settings** > **Inbound Rules**.
* Kiểm tra có **Rule** nào cho SQL Server trên cổng **1433** chưa. Nếu chưa, thêm mới với các thông số:
  + **Protocol**: TCP
  + **Port**: 1433
  + **Action**: Allow

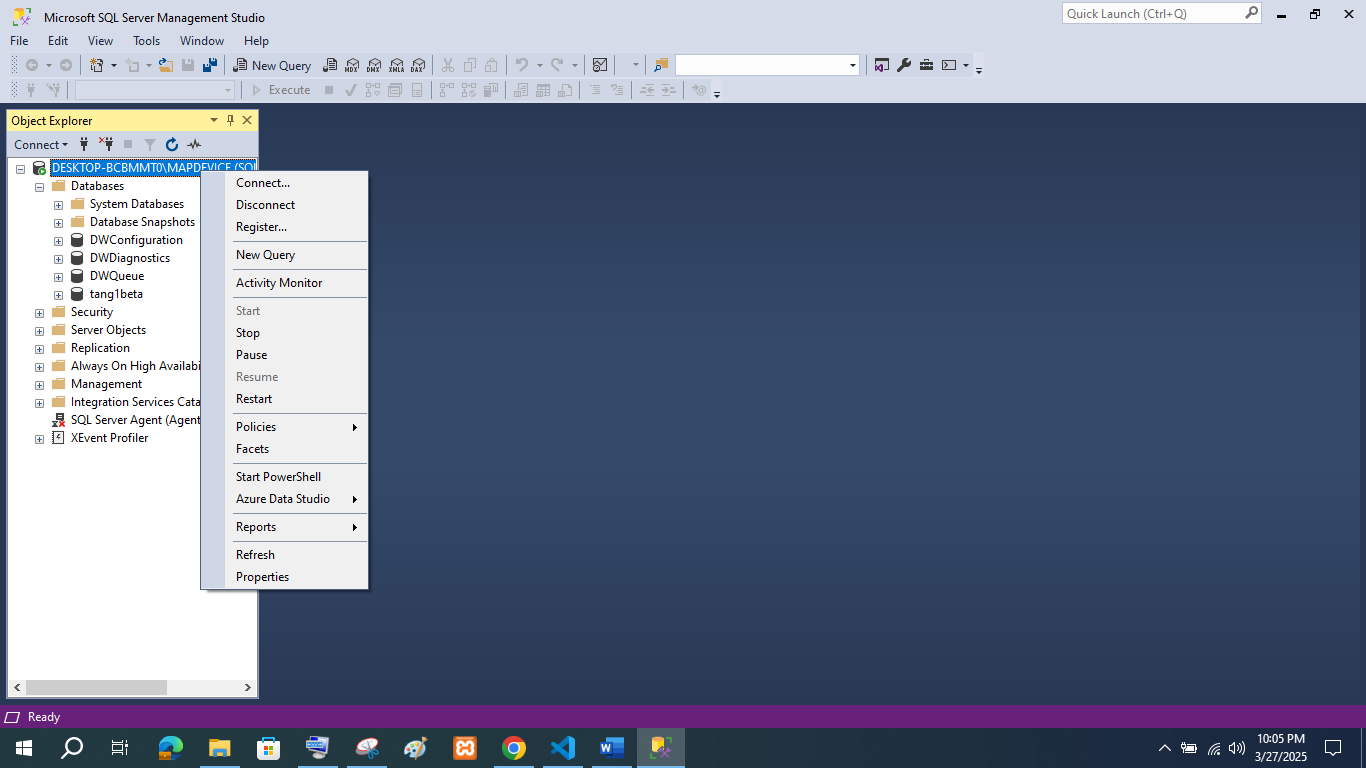
**4. Kiểm tra tài khoản và xác thực**

* Đảm bảo tài khoản sa đã bật và có mật khẩu đúng.
* Kiểm tra chế độ xác thực:
  1. Mở **SQL Server Management Studio (SSMS)**.
  2. Nhấp chuột phải vào **server** > **Properties**.
  3. Chọn **Security** > Đảm bảo **SQL Server and Windows Authentication mode** đã được chọn.

Sau khi thực hiện các bước trên, hãy thử khởi động lại server và kiểm tra lại kết nối. Nếu vẫn lỗi, bạn có thể kiểm tra log chi tiết trong **SSMS** hoặc gửi lỗi cụ thể hơn để mình hỗ trợ tiếp. 🚀

4o

**You said:**



ktra giùm tôi sql server chạy chưa ?

**ChatGPT said:**

SQL Server của bạn đang chạy, vì bạn có thể truy cập vào **Object Explorer** trong **SQL Server Management Studio (SSMS)** và thấy danh sách cơ sở dữ liệu.

Tuy nhiên, để chắc chắn SQL Server đang lắng nghe trên cổng 1433 và có thể kết nối từ ứng dụng bên ngoài, hãy kiểm tra thêm:

**1. Kiểm tra SQL Server đang chạy**

* Trong **SQL Server Configuration Manager**:
  + Chọn **SQL Server Services**.
  + Kiểm tra **SQL Server (MSSQLSERVER)** hoặc **SQL Server (<Tên instance>)** có đang ở trạng thái **Running** không.

**2. Kiểm tra TCP/IP và cổng 1433**

* Trong **SQL Server Configuration Manager**:
  + Vào **SQL Server Network Configuration** > **Protocols for MSSQLSERVER**.
  + Đảm bảo **TCP/IP** đã **Enabled**.
  + Nhấp đúp vào **TCP/IP**, kéo xuống **IPAll**, kiểm tra:
    - **TCP Dynamic Ports**: Để trống.
    - **TCP Port**: Phải là **1433**.

**3. Kiểm tra SQL Server đang lắng nghe cổng 1433**

* Mở **Command Prompt** và chạy:

sh

CopyEdit

netstat -an | findstr 1433

Nếu thấy dòng như 0.0.0.0:1433 hoặc 127.0.0.1:1433 nghĩa là SQL Server đang lắng nghe.

**4. Kiểm tra user sa có hoạt động không**

* Trong SSMS, chạy:

sql

CopyEdit

ALTER LOGIN sa ENABLE;

ALTER LOGIN sa WITH PASSWORD = 'Mật\_khẩu\_mới';

* Kiểm tra chế độ xác thực:
  + Nhấp chuột phải vào **server** > **Properties**.
  + Chọn **Security**, đảm bảo **SQL Server and Windows Authentication mode** được chọn.

Sau khi kiểm tra và sửa lỗi nếu cần, khởi động lại SQL Server và thử kết nối lại từ ứng dụng Node.js. 🚀

4o

**You said:**

tôi ko thấy Chọn SQL Server Services.

**ChatGPT said:**

Bạn có thể làm theo hướng dẫn dưới đây để mở **SQL Server Configuration Manager** và kiểm tra **SQL Server Services**:

**Cách mở SQL Server Configuration Manager**

**Cách 1: Mở bằng Run**

1. Nhấn Win + R, nhập:

CopyEdit

SQLServerManager16.msc

(với **SQL Server 2022**)

Nếu dùng phiên bản khác, thử:

pgsql

CopyEdit

SQLServerManager15.msc (SQL Server 2019)

SQLServerManager14.msc (SQL Server 2017)

SQLServerManager13.msc (SQL Server 2016)

1. Nhấn Enter để mở.

**Cách 2: Mở bằng Start Menu**

1. Nhấn Win + S, tìm **SQL Server Configuration Manager**.
2. Nhấp vào ứng dụng để mở.

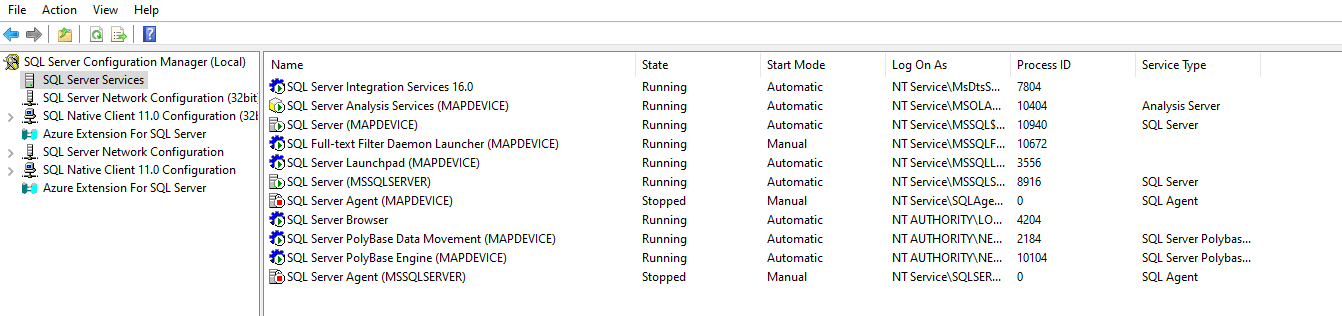
**Kiểm tra SQL Server Services**

* Trong **SQL Server Configuration Manager**, chọn **SQL Server Services** ở bên trái.
* Kiểm tra trạng thái của **SQL Server (MSSQLSERVER)** hoặc **SQL Server (Tên\_Instance)**.
  + Nếu **Running**, nghĩa là SQL Server đang hoạt động.
  + Nếu **Stopped**, nhấp chuột phải > **Start** để khởi động.

Nếu bạn vẫn không thấy **SQL Server Services**, hãy kiểm tra xem bạn có cài đúng phiên bản SQL Server không. Nếu cần, mình sẽ hướng dẫn cách kiểm tra và cài đặt lại! 🔍🚀

4o

**You said:**



**ChatGPT said:**

Dựa vào ảnh bạn gửi, **SQL Server đang chạy**, nhưng có thể bạn đang kết nối sai instance.

**Những gì đang chạy ✅:**

* SQL Server (MAPDEVICE) **(Đang chạy - Running)**
* SQL Server Browser **(Đang chạy - Running)**

**Những gì đang dừng ❌:**

* SQL Server (MSSQLSERVER) **(Dừng - Stopped)**  
  → Nếu bạn đang kết nối với **MSSQLSERVER**, hãy thử **Start** nó.

**Cách sửa lỗi kết nối SQL Server từ Node.js**

**1️⃣ Kiểm tra Instance đang dùng**

* Mở **SQL Server Management Studio (SSMS)**.
* Trong **Object Explorer**, kiểm tra xem bạn đang kết nối bằng **MAPDEVICE** hay **MSSQLSERVER**.

**2️⃣ Kiểm tra lại DB\_SERVER trong file .env**

Mở file sql.env, sửa lại:

env

CopyEdit

DB\_USER=sa

DB\_PASSWORD=D@ihocfpt2025

DB\_SERVER=MAPDEVICE # Hoặc localhost\MAPDEVICE

DB\_DATABASE=tang1beta

DB\_PORT=1433

Lưu ý:

* Nếu bạn đang kết nối **SQL Server Authentication**, đảm bảo user sa đã được kích hoạt.
* Nếu không kết nối được, thử DB\_SERVER=localhost\MAPDEVICE.

**3️⃣ Kiểm tra cổng 1433 có mở không**

* Mở **SQL Server Configuration Manager**.
* Chọn **SQL Server Network Configuration** > **Protocols for MAPDEVICE**.
* **Bật TCP/IP** (Nếu đang Disabled thì Enable lại).
* Khởi động lại SQL Server.

Sau khi thử các bước trên, bạn hãy thử chạy lại Node.js (node server.js) xem có kết nối được chưa! 🚀

4o

**You said:**

Object Explorer ở đâu

**ChatGPT said:**

**Object Explorer** là cửa sổ hiển thị danh sách các database và server trong **SQL Server Management Studio (SSMS)**.

**Cách mở Object Explorer**

**Cách 1: Sử dụng thanh menu**

1. Mở **SQL Server Management Studio (SSMS)**.
2. Vào **View** trên thanh menu.
3. Chọn **Object Explorer** (hoặc nhấn **F8**).

**Cách 2: Hiện nếu bị ẩn**

* Nếu Object Explorer không hiển thị, bạn có thể mở nó bằng cách:
  + **Nhấn Ctrl + Alt + G** hoặc
  + **Chọn View > Object Explorer trong menu**.

Sau khi mở **Object Explorer**, bạn sẽ thấy danh sách các cơ sở dữ liệu (Databases) và server đang kết nối. Nếu chưa kết nối, bạn cần nhấn **Connect** và nhập thông tin đăng nhập SQL Server của bạn.

4o

**You said:**

đã nhấn Chọn Object Explorer (hoặc nhấn F8). mà không ra thông tin

**ChatGPT said:**

Nếu bạn đã nhấn **F8** hoặc chọn **Object Explorer** trong **SSMS** mà vẫn không thấy cửa sổ Object Explorer, hãy thử các cách sau:

**1. Kiểm tra xem Object Explorer có bị thu nhỏ không**

* Nhìn vào **góc trái hoặc phải** của SSMS xem có **cửa sổ nhỏ bị thu gọn** không.
* Nếu có, nhấn vào đó để mở rộng lại.

**2. Mở lại Object Explorer bằng lệnh**

1. Nhấn **Ctrl + Shift + N** để mở cửa sổ **Connect to Server**.
2. Chọn SQL Server của bạn và nhấn **Connect**.
3. Object Explorer sẽ tự động xuất hiện sau khi kết nối thành công.

**3. Kiểm tra cửa sổ SSMS có bị lỗi hiển thị không**

* Đóng **SSMS** và mở lại.
* Nếu vẫn không có Object Explorer, thử **reset lại layout mặc định**:
  1. Vào **Window** trên thanh menu.
  2. Chọn **Reset Window Layout**.
  3. Thử mở lại **Object Explorer** bằng **F8**.

**4. Cập nhật hoặc cài lại SSMS**

* Nếu tất cả cách trên không hoạt động, có thể **SSMS bị lỗi**. Hãy thử:
  + Cập nhật **SQL Server Management Studio** lên phiên bản mới nhất.
  + Nếu cần, gỡ cài đặt và cài lại **SSMS**.

Bạn thử các cách trên và cho mình biết kết quả nhé! 🚀

4o

**You said:**

PS C:\ReactJS\mapthietbi\backend> node server.js 🚀 Server đang chạy tại http://localhost:5000 ❌ Lỗi kết nối SQL Server: ConnectionError: Failed to connect to MAPDEVICE:1433 - getaddrinfo ENOTFOUND mapdevice at C:\ReactJS\mapthietbi\backend\node\_modules\mssql\lib\tedious\connection-pool.js:85:17 at Connection.onConnect (C:\ReactJS\mapthietbi\backend\node\_modules\tedious\lib\connection.js:849:9) at Object.onceWrapper (node:events:633:26) at Connection.emit (node:events:518:28) at Connection.emit (C:\ReactJS\mapthietbi\backend\node\_modules\tedious\lib\connection.js:970:18) at Connection.socketError (C:\ReactJS\mapthietbi\backend\node\_modules\tedious\lib\connection.js:1353:12) at C:\ReactJS\mapthietbi\backend\node\_modules\tedious\lib\connection.js:1146:14 at process.processTicksAndRejections (node:internal/process/task\_queues:85:11) { code: 'ESOCKET', originalError: ConnectionError: Failed to connect to MAPDEVICE:1433 - getaddrinfo ENOTFOUND mapdevice at Connection.socketError (C:\ReactJS\mapthietbi\backend\node\_modules\tedious\lib\connection.js:1353:28) at C:\ReactJS\mapthietbi\backend\node\_modules\tedious\lib\connection.js:1146:14 at process.processTicksAndRejections (node:internal/process/task\_queues:85:11) { code: 'ESOCKET', [cause]: Error: getaddrinfo ENOTFOUND mapdevice at GetAddrInfoReqWrap.onlookupall [as oncomplete] (node:dns:120:26) { errno: -3008, code: 'ENOTFOUND', syscall: 'getaddrinfo', hostname: 'mapdevice' } } }

**ChatGPT said:**

Lỗi này cho thấy **Node.js không thể kết nối đến SQL Server** vì hostname MAPDEVICE không được tìm thấy. Dưới đây là cách khắc phục:

**🔍 1. Kiểm tra SQL Server có đang chạy không**

📌 **Từ ảnh bạn gửi**, SQL Server (MAPDEVICE) đang **chạy** nên không cần bật lại. Nhưng cần kiểm tra **kết nối**.

Thử kết nối từ SSMS:

1. Mở **SQL Server Management Studio (SSMS)**.
2. Chọn **Connect** → **Database Engine**.
3. Nhập thông tin:
   * **Server name**: localhost hoặc MAPDEVICE
   * **Authentication**: SQL Server Authentication
   * **Username**: sa
   * **Password**: D@ihocfpt2025 (từ file sql.env của bạn)
4. Nhấn **Connect**.
   * **Nếu kết nối được** → Tiếp tục bước 2.
   * **Nếu không kết nối được** → SQL Server chưa cấu hình đúng.

**🔥 2. Sửa lại DB\_SERVER trong sql.env**

Hiện tại, bạn đang dùng:

env

CopyEdit

DB\_SERVER=MAPDEVICE

🚨 **Nhưng lỗi "getaddrinfo ENOTFOUND" cho thấy Node.js không nhận diện được MAPDEVICE**.

Thử sửa thành:

env

CopyEdit

DB\_SERVER=localhost

hoặc:

env

CopyEdit

DB\_SERVER=127.0.0.1

Rồi **lưu lại** và khởi động lại server:

sh

CopyEdit

node server.js

**🔑 3. Kiểm tra SQL Server có chấp nhận kết nối từ bên ngoài không**

Mở **SQL Server Configuration Manager**:

1. **Bật TCP/IP**:
   * Chọn **SQL Server Network Configuration** → **Protocols for MSSQLSERVER**.
   * Đảm bảo **TCP/IP** đang **Enabled**.
2. **Mở Port 1433**:
   * Chọn **IP Addresses** → Ở phần **IPAll**, đặt **TCP Port = 1433**.
3. **Khởi động lại SQL Server**.

**🖥️ 4. Kiểm tra SQL Server Browser**

* SQL Server Browser giúp nhận diện instance, nếu tắt thì có thể không kết nối được.
* Trong **SQL Server Configuration Manager**, kiểm tra **SQL Server Browser** có đang chạy không.

Nếu không chạy:

1. Nhấp chuột phải → **Start**.
2. Đặt **Start Mode** thành **Automatic**.

**🚀 5. Khởi động lại Server & Thử lại**

Sau khi sửa xong, thử chạy lại:

sh

CopyEdit

node server.js

Nếu vẫn lỗi, gửi lại ảnh màn hình SSMS của bạn khi kết nối nhé! 🔍

**You said:**

servername là:DESKTOP-BCBMMT0\MAPDEVICE

**ChatGPT said:**

Cảm ơn bạn đã cung cấp Server Name:

📌 **Hiện tại, cấu hình trong .env của bạn có thể sai**. Thử thay đổi:

**1️⃣ Sửa lại DB\_SERVER trong file sql.env**

Mở file .env của bạn và sửa:

env

CopyEdit

DB\_SERVER=DESKTOP-BCBMMT0\MAPDEVICE

Hoặc nếu vẫn không kết nối được, thử:

env

CopyEdit

DB\_SERVER=localhost\MAPDEVICE

hoặc:

env

CopyEdit

DB\_SERVER=127.0.0.1\MAPDEVICE

**2️⃣ Kiểm tra SQL Server có hỗ trợ kết nối qua TCP/IP không**

**Mở SQL Server Configuration Manager**:

1. **Chọn SQL Server Network Configuration** → **Protocols for MAPDEVICE**.
2. Bật **TCP/IP** (nếu chưa).
3. **Mở Properties của TCP/IP** → Ở phần **IPAll**, kiểm tra:
   * **TCP Port = 1433**
   * **TCP Dynamic Ports**: Xóa giá trị nếu có.
4. **Lưu lại & khởi động lại SQL Server**.

**3️⃣ Thử Kết Nối Bằng IP (Thay vì Tên Máy)**

Chạy lệnh sau trong **cmd**:

sh

CopyEdit

ipconfig

Tìm **IPv4 Address** (ví dụ 192.168.1.100), rồi thử sửa .env:

env

CopyEdit

DB\_SERVER=192.168.1.100\MAPDEVICE

**4️⃣ Restart Server & Chạy lại Backend**

Sau khi chỉnh sửa, chạy lại:

sh

CopyEdit

node server.js

Nếu vẫn bị lỗi, gửi lại ảnh lỗi mới nhất để mình hỗ trợ tiếp nhé! 🚀

4o

**You said:**

khong ping tới DB\_SERVER=192.168.1.100\MAPDEVICE

**ChatGPT said:**

Nếu không ping được 192.168.1.100\MAPDEVICE, bạn thử các bước sau để kiểm tra kết nối:

**1️⃣ Kiểm tra cổng 1433 có mở không?**

Chạy lệnh sau trong **Command Prompt (cmd)** trên máy chứa SQL Server:

sh

CopyEdit

netstat -an | findstr 1433

* Nếu thấy dòng chứa 0.0.0.0:1433 hoặc 192.168.1.100:1433 → **SQL Server đang mở cổng** ✅.
* Nếu không thấy gì → **Cần bật cổng 1433**.

**2️⃣ Bật tính năng lắng nghe TCP/IP trên SQL Server**

1. **Mở SQL Server Configuration Manager**.
2. Vào **SQL Server Network Configuration** → **Protocols for MAPDEVICE**.
3. Đảm bảo **TCP/IP** đang **Enabled**.
4. Nhấp đúp vào **TCP/IP**:
   * Tại tab **IP Addresses**, tìm **IPAll**:
     + **TCP Port**: Đặt là **1433**.
     + **TCP Dynamic Ports**: XÓA nếu có giá trị.
5. Nhấn **OK** và **Restart SQL Server**.

**3️⃣ Kiểm tra firewall (tường lửa)**

SQL Server có thể bị chặn bởi firewall:

1. **Mở Windows Defender Firewall**.
2. Chọn **Advanced Settings**.
3. Chọn **Inbound Rules** → **New Rule**.
4. Chọn **Port** → **Next**.
5. Chọn **TCP** và nhập **1433** → **Next**.
6. Chọn **Allow the connection** → **Next**.
7. Đặt tên rule là **SQL Server 1433** và nhấn **Finish**.

**4️⃣ Kiểm tra SQL Browser Service**

SQL Server có thể không lắng nghe đúng cổng:

1. **Mở SQL Server Configuration Manager**.
2. Chọn **SQL Server Services**.
3. Đảm bảo **SQL Server Browser** đang **Running** (Nếu **Stopped**, nhấp chuột phải → **Start**).

**5️⃣ Kiểm tra kết nối từ máy khác**

Trên máy client (máy chạy Node.js backend), thử lệnh:

sh

CopyEdit

telnet 192.168.1.100 1433

* Nếu màn hình đen (không lỗi) → **Kết nối OK** ✅.
* Nếu lỗi **Could not open connection** → **Cổng 1433 bị chặn**.

**6️⃣ Cập nhật lại kết nối trong backend**

Nếu SQL Server chạy trên **localhost**, thử thay đổi .env:

env

CopyEdit

DB\_SERVER=localhost\MAPDEVICE

Hoặc:

env

CopyEdit

DB\_SERVER=DESKTOP-BCBMMT0\MAPDEVICE

Và khởi động lại backend:

sh

CopyEdit

node server.js

Nếu vẫn không được, gửi lại ảnh lỗi mới nhất hoặc mô tả chi tiết mình sẽ hỗ trợ tiếp! 🚀

4o

Top of Form

Search

Reason

Bottom of Form

ChatGPT can make mistakes. Check important info.

?