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:

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:

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

Copy

Edit

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

Copy

Edit

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

Copy

Edit

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

Copy

Edit

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

Copy

Edit

node server.js

hoặc

sh

Copy

Edit

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

Copy

Edit

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! 😊