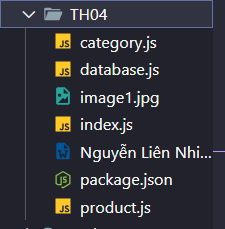
Nguyễn Liên Nhi - 2274802010612

**Cấu trúc:**



**Category.js**

import mongoose from "mongoose";

// Định nghĩa Schema cho Category

const categorySchema = **new** *mongoose*.Schema({

  \_id: { type: *String*, required: true },

  name: { type: *String*, required: true },

  parent\_category\_id: { type: *mongoose*.Schema.*Types*.ObjectId, ref: "categories", default: null },

});

// Tạo model cho Category

const Category = *mongoose*.model("categories", categorySchema);

// Thêm Category

async function addCategory(*\_id*, *name*, *parent\_category\_id* = null) {

  const category = **new** Category({\_id, name, parent\_category\_id });

  await category.save();

  console.log("Thêm category:", category);

}

// Cập nhật Category

async function updateCategory(*\_id*, *newName*, *newParent* = null) {

  const updated = await Category.findByIdAndUpdate(*\_id*, { name: *newName*, parent\_category\_id: *newParent* }, { new: true });

  console.log("Cập nhật category:", updated);

}

// Xóa Category

async function deleteCategory(*\_id*) {

  const hasProducts = await *mongoose*.model("products").exists({ categories\_id: *\_id* });

  const hasSubCategories = await Category.exists({ parent\_category\_id: *\_id* });

  if (hasProducts || hasSubCategories) {

    console.log("Không thể xóa category có sản phẩm hoặc category con!");

    return;

  }

  await Category.findByIdAndDelete(id);

  console.log("Xóa category thành công!");

}

// Lấy danh sách Categories

async function getCategories() {

  const categories = await Category.find().populate("parent\_category\_id");

  console.log("Danh sách categories:", categories);

}

export { addCategory, updateCategory, deleteCategory, getCategories };

**Product.js**

import mongoose from "mongoose";

// Định nghĩa Schema cho Product

const productSchema = **new** *mongoose*.Schema({

  \_id: { type: *String*, required: true },

  name: { type: *String*, required: true },

  description: { type: *String* },

  price: { type: *Number*, required: true },

  categories\_id: [{ type: *String*, ref: "categories" }],

  images: [{ type: *String* }],

  show: { type: *Boolean*, default: true },

});

// Tạo model cho Product

const Product = *mongoose*.model("products", productSchema);

// Thêm Product

async function addProduct(*\_id*, *name*, *description*, *price*, *categories*, *images* = []) {

  const product = **new** Product({\_id, name, description, price, categories\_id: *categories*, images });

  await product.save();

  console.log("Thêm sản phẩm:", product);

}

// Cập nhật Product

async function updateProduct(*id*, *newData*) {

  const updated = await Product.findByIdAndUpdate(*id*, *newData*, { new: true });

  console.log("Cập nhật sản phẩm:", updated);

}

// Xóa Product

async function deleteProduct(*id*) {

  await Product.findByIdAndDelete(*id*);

  console.log("Xóa sản phẩm thành công!");

}

// Lấy danh sách Products

async function getProducts() {

  const products = await Product.find().populate("categories\_id");

  console.log("Danh sách sản phẩm:", products);

}

// Tìm kiếm Product theo tên

async function searchProductByName(*name*) {

  const products = await Product.find({ name: **new** *RegExp*(*name*, "i") }).populate("categories\_id");

  console.log("Kết quả tìm kiếm:", products);

}

// Lấy sản phẩm theo Category

async function getProductsByCategory(*categoryId*) {

  const products = await Product.find({ categories\_id: *categoryId* }).populate("categories\_id");

  console.log("Sản phẩm theo danh mục:", products);

}

export { addProduct, updateProduct, deleteProduct, getProducts, searchProductByName, getProductsByCategory };

**Database.js**

import mongoose from "mongoose";

const MONGO\_URI = "mongodb://127.0.0.1:27017/Lab3";

async function connectDB() {

  try {

    await *mongoose*.connect(MONGO\_URI);

    console.log("Kết nối MongoDB thành công!");

  } catch (err) {

    console.error("Lỗi kết nối MongoDB:", err);

  }

}

export default connectDB;

Index.js

import connectDB from "./database.js";

import { addCategory, updateCategory, deleteCategory, getCategories } from "./category.js";

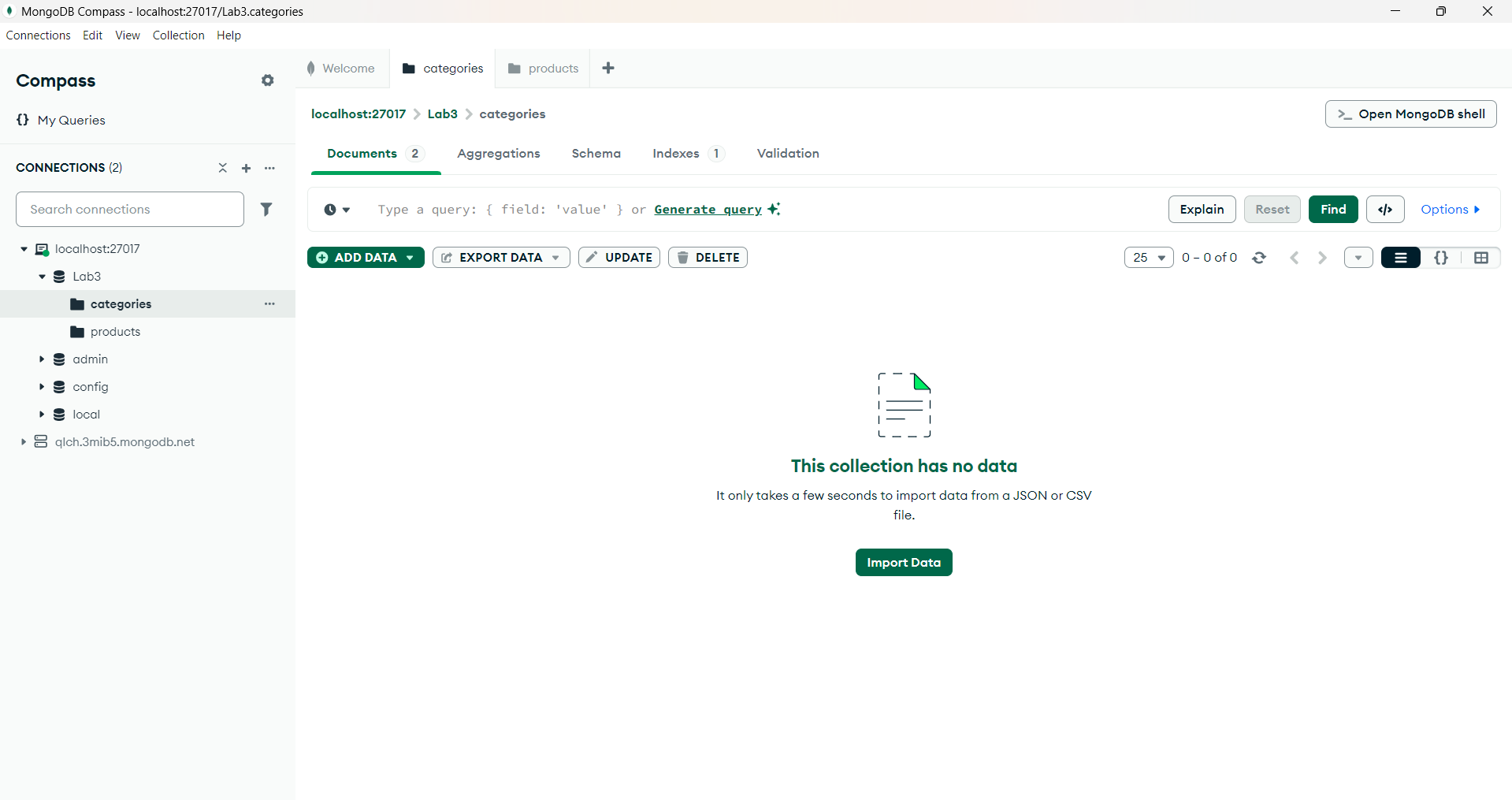
import { addProduct, updateProduct, deleteProduct, getProducts, searchProductByName, getProductsByCategory } from "./product.js";

await connectDB();

console.log("Chương trình đã hoàn thành! Đang thoát...");

process.exit(0);

**MongoDB sau khởi tạo**

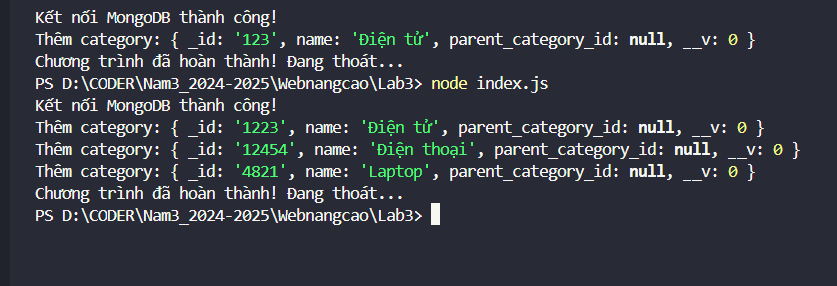


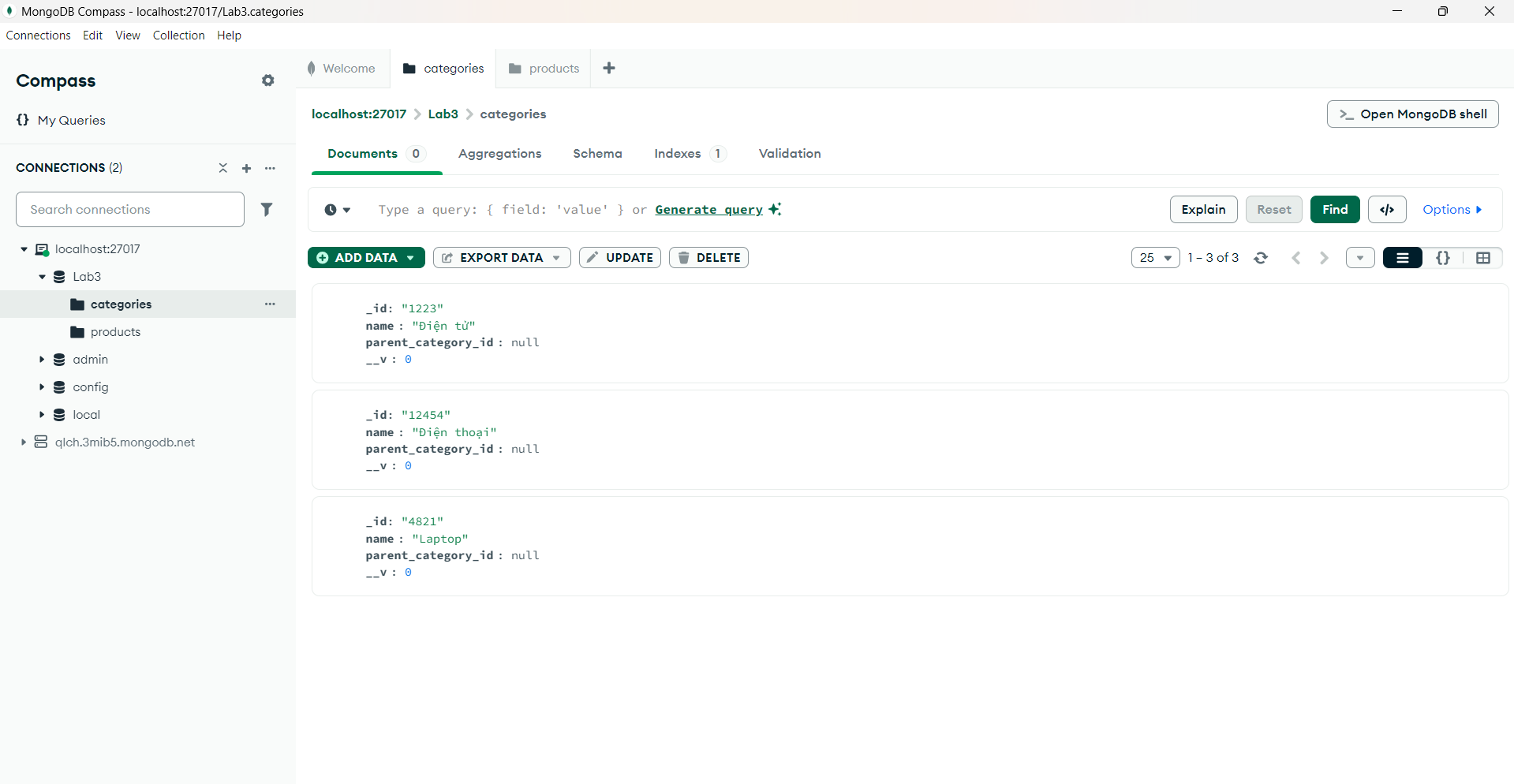
1. Thêm 1 category mới

await addCategory("1223","Điện tử");

await addCategory("12454","Điện thoại");

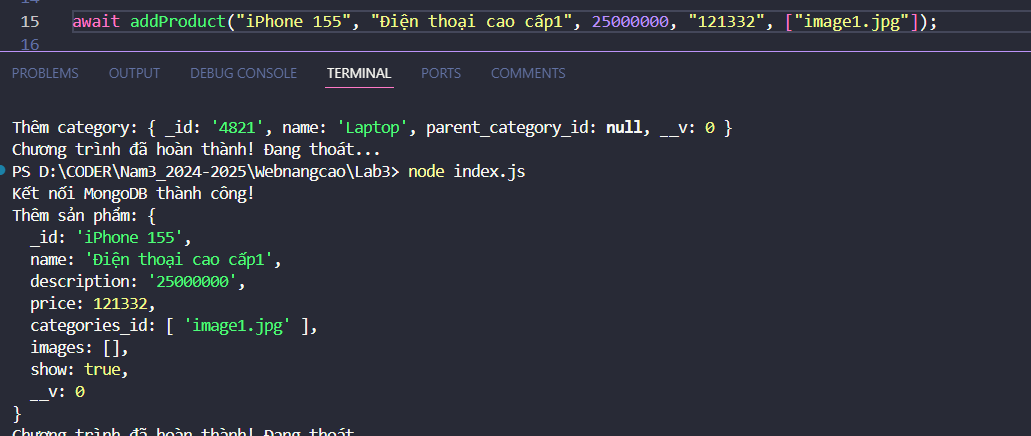
await addCategory("4821","Laptop");

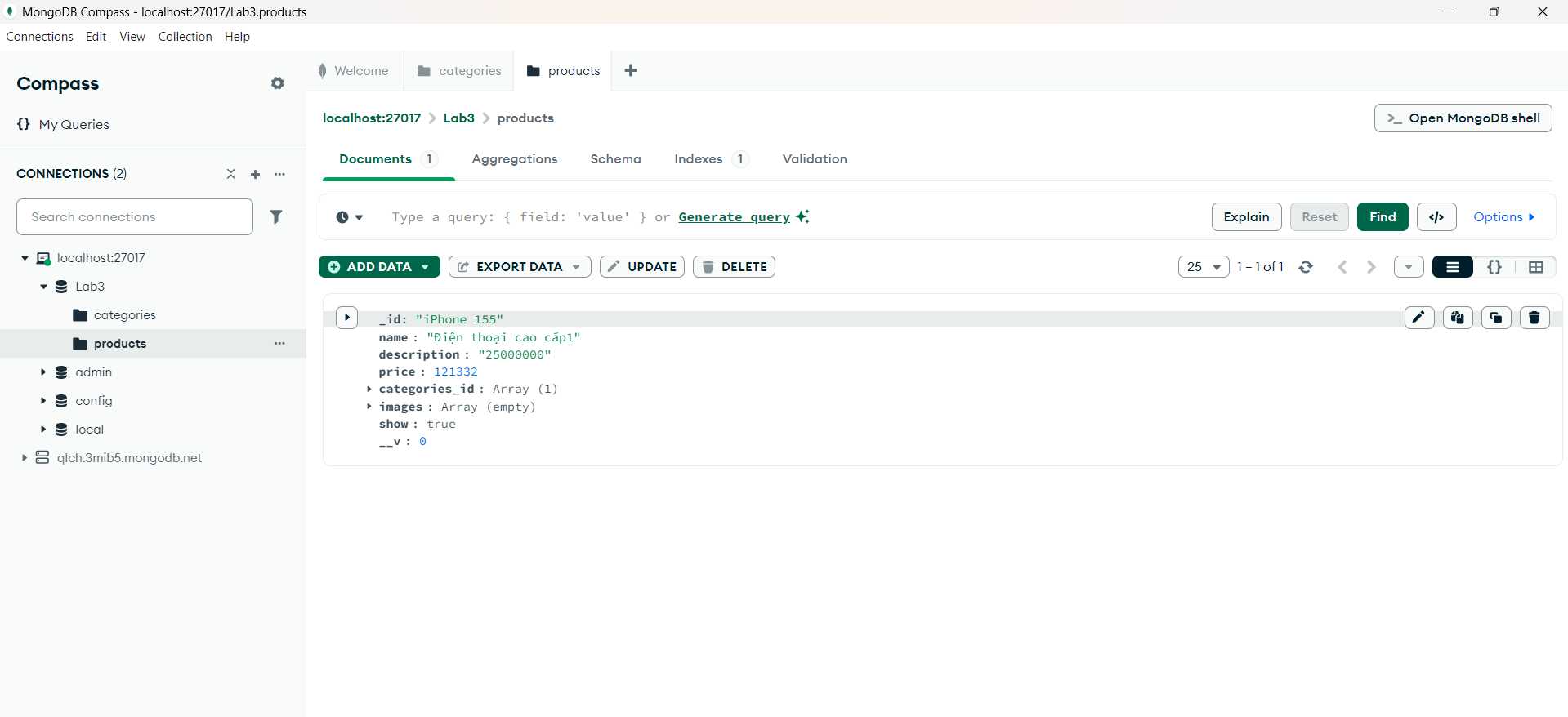




1. Thêm 1 product mới

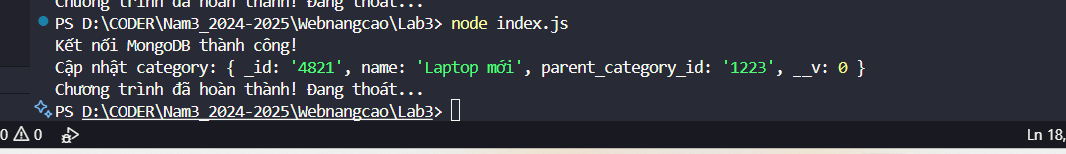
await addProduct("iPhone 155", "Điện thoại cao cấp1", 25000000, "121332", ["image1.jpg"]);

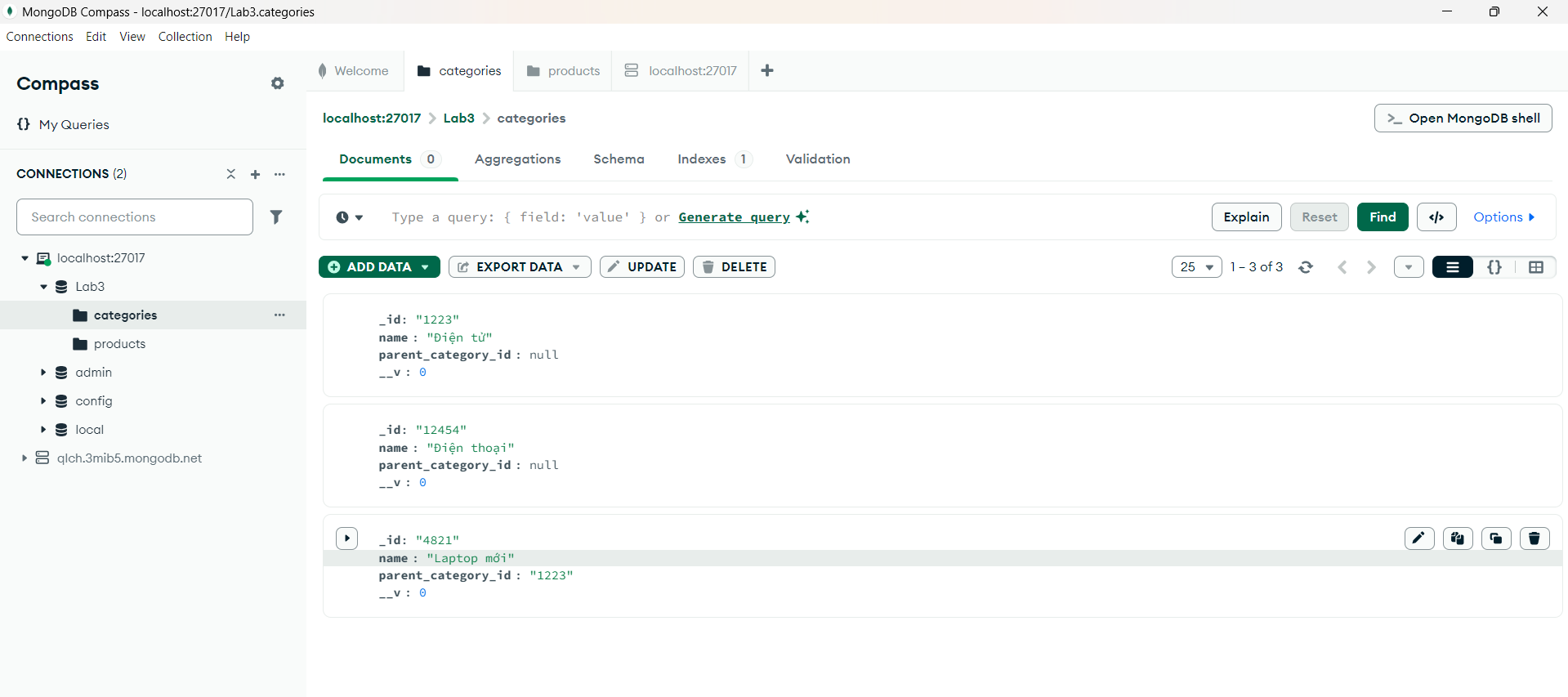




1. Sửa 1 category theo id

await updateCategory("4821", "Laptop mới", "1223");

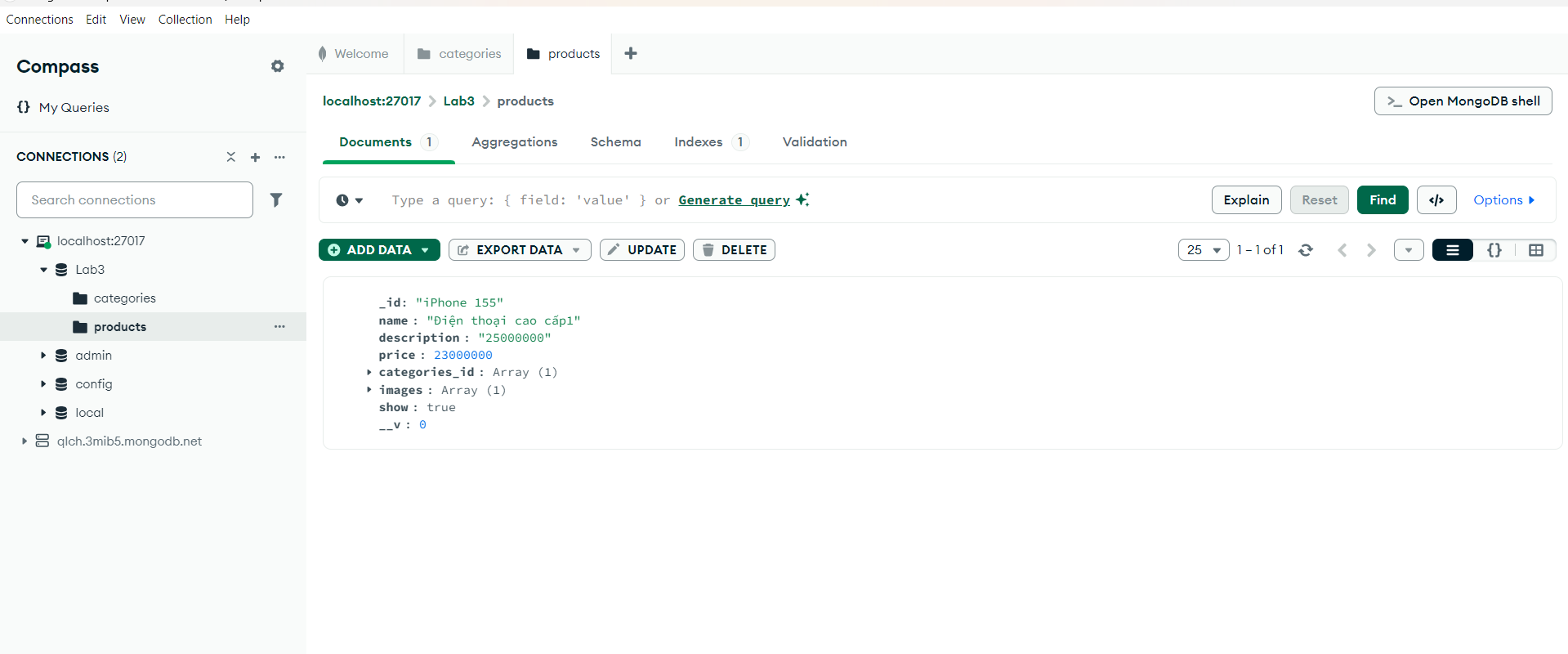




1. Sửa 1 product theo id

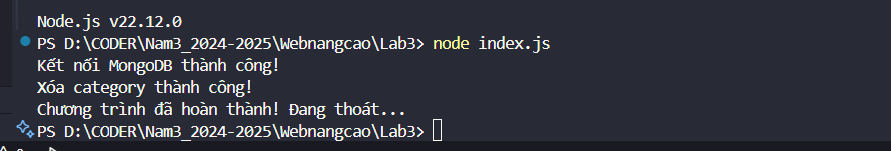
await updateProduct("iPhone 155", { price: 23000000, images: ["iphone15\_new.jpg"] });

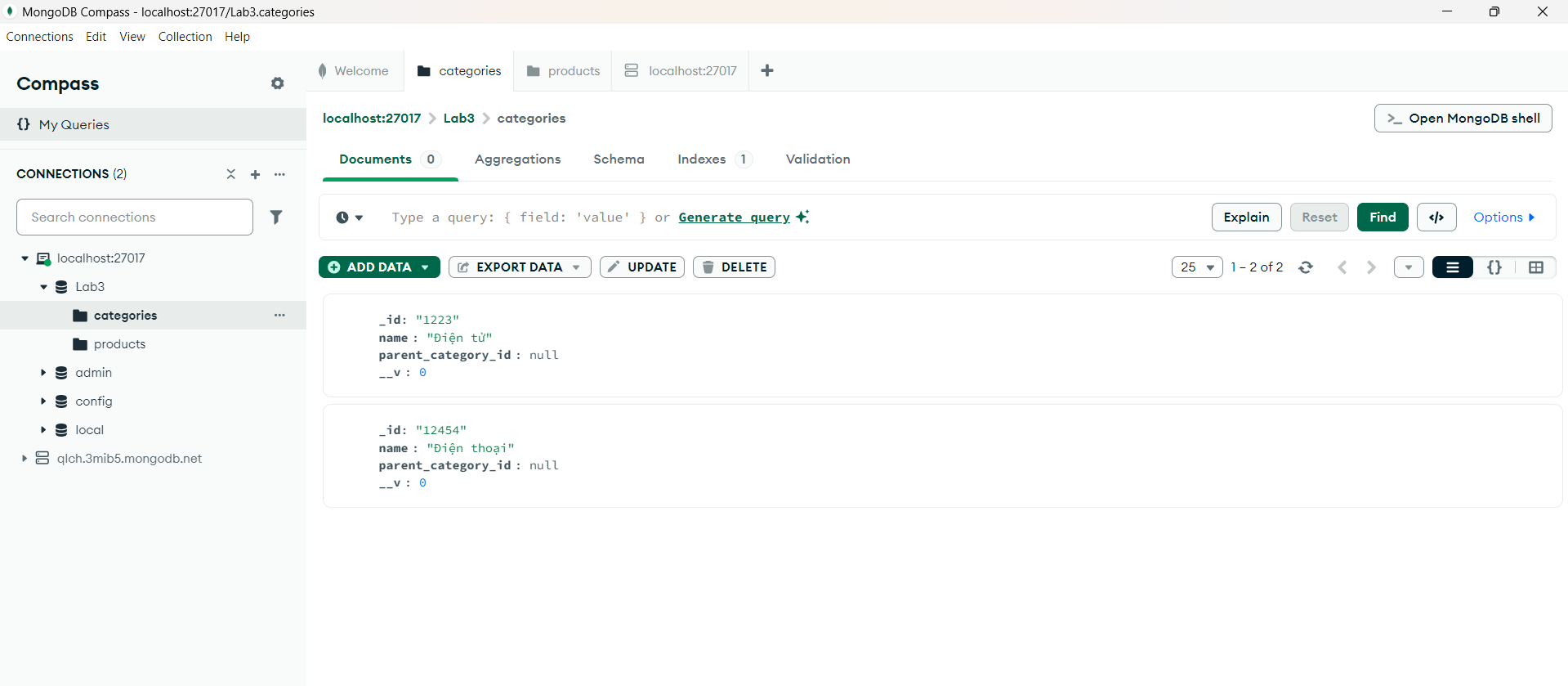




1. Xóa 1 product theo id

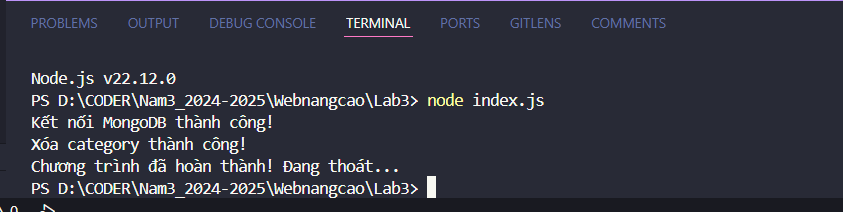
await deleteCategory("4821");

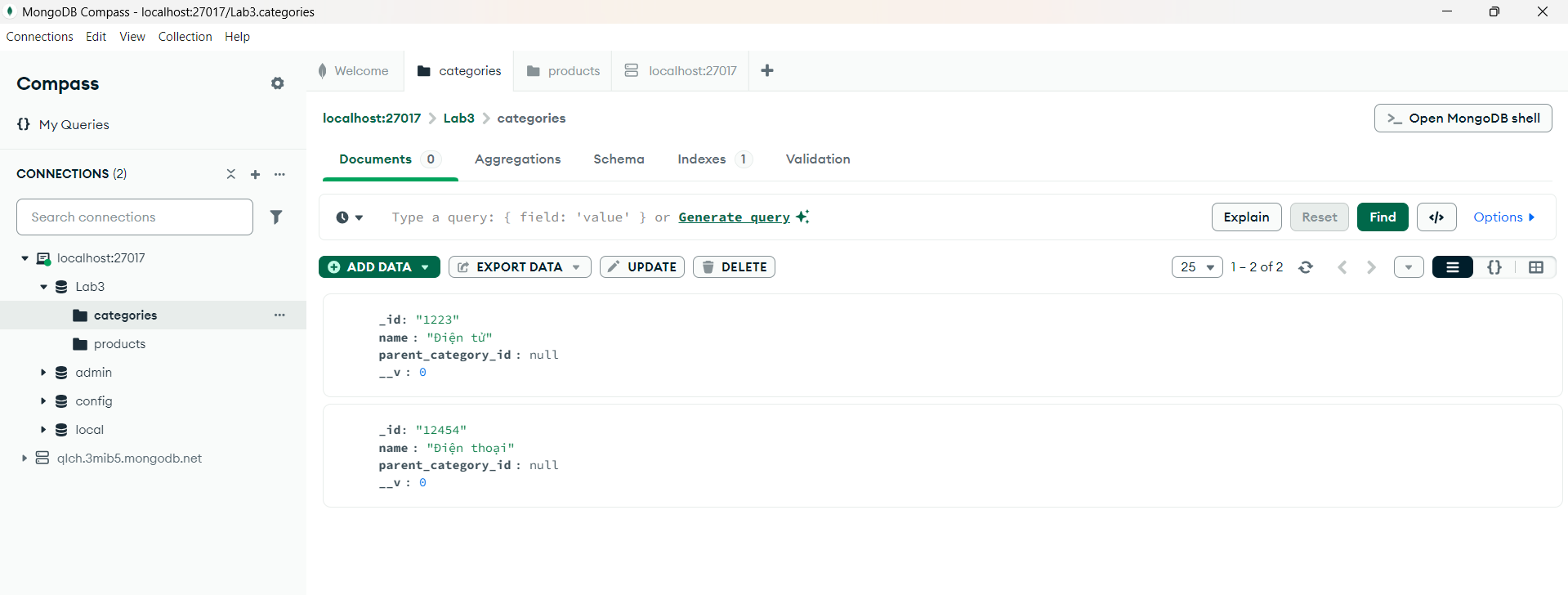




1. Xóa 1 category theo id (chỉ xóa khi category đó không có sản phẩm nào và không có category con nào).

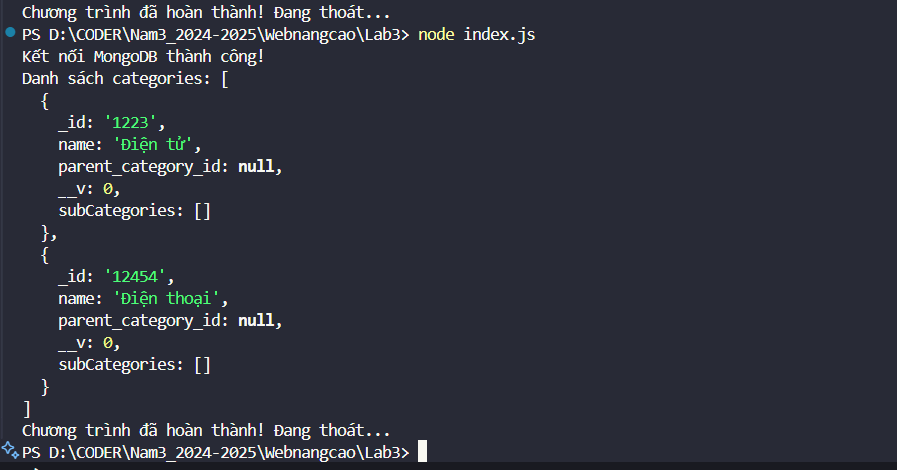
await deleteCategory("4821");

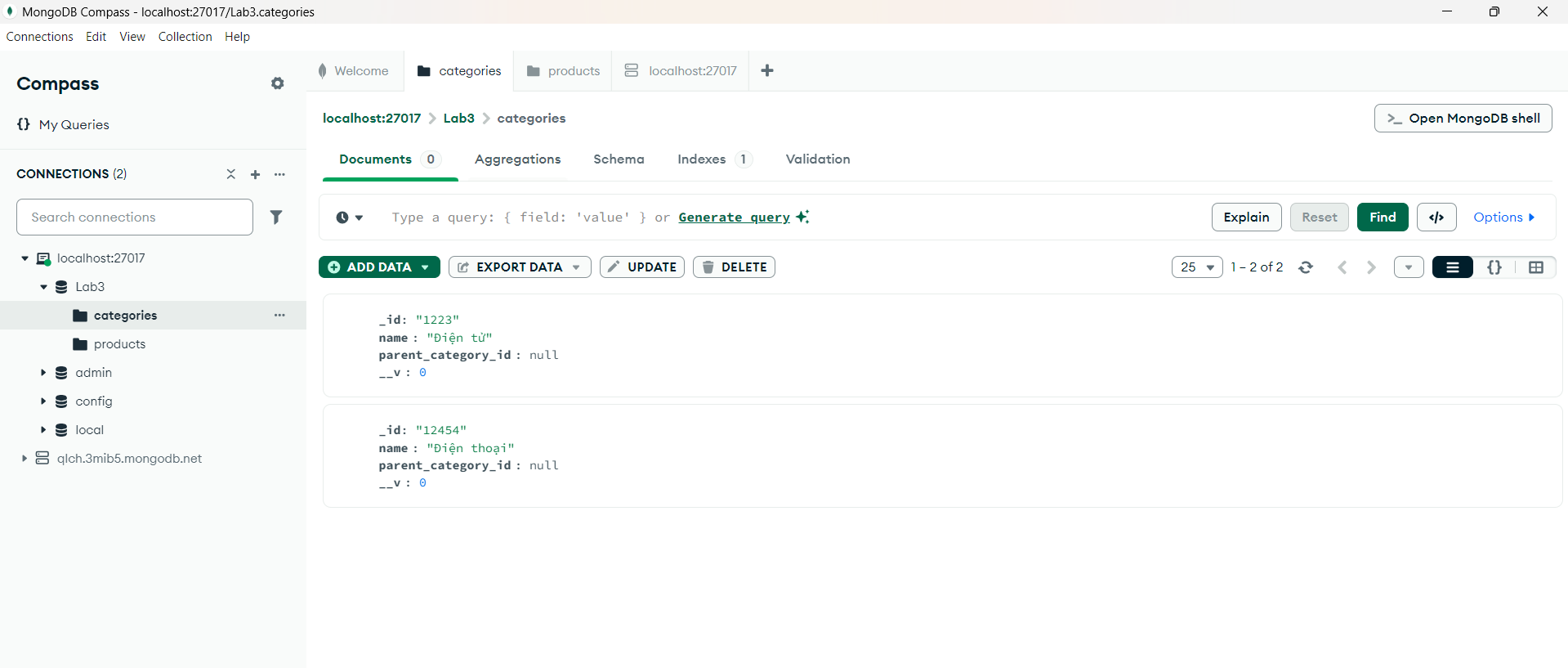




1. Lấy danh sách các category (mỗi category trả về cần có thông tin đầy đủ của category con của nó nếu có).

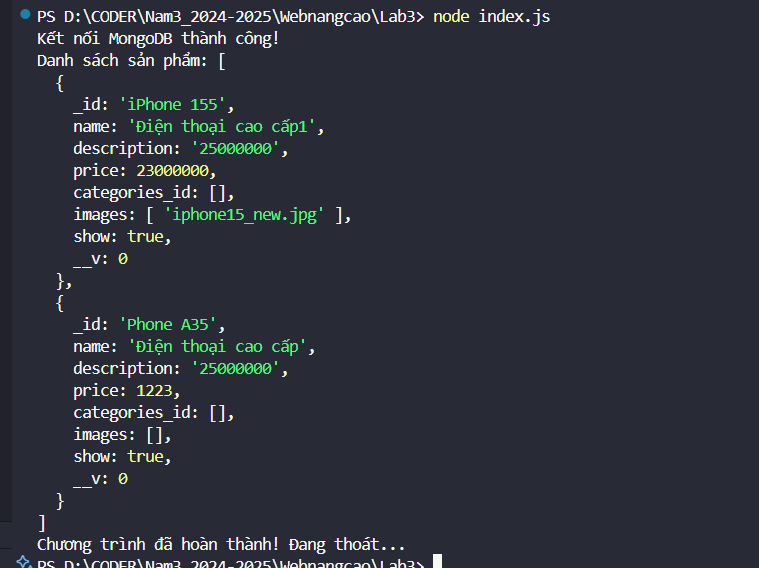
await getCategories();

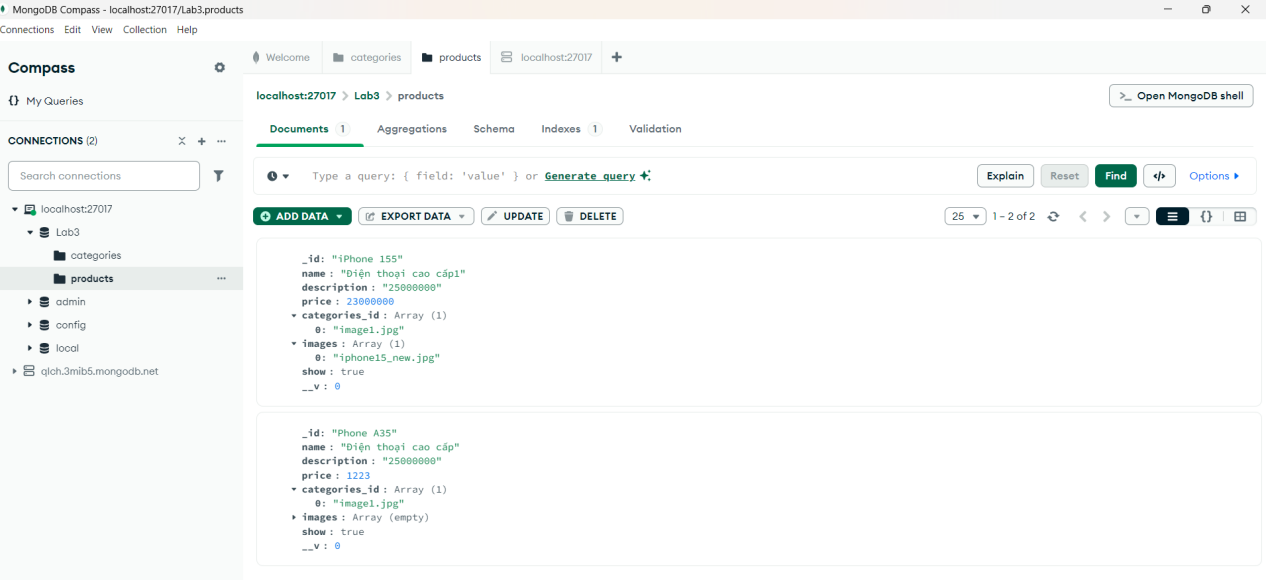




1. Lấy danh sách các product

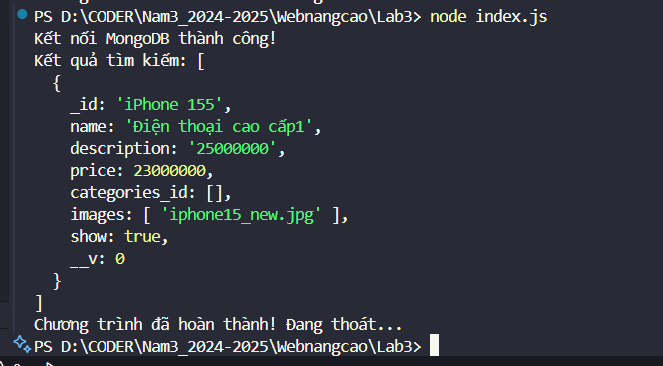
await getProducts();





1. Tìm kiếm product theo tên

await searchProductByName("Điện thoại cao cấp1");



1. Tìm danh sách product theo category

+ Thêm sản phẩm theo danh mục

await addProduct(

    "PhoneX100",

    "Điện thoại thông minh",

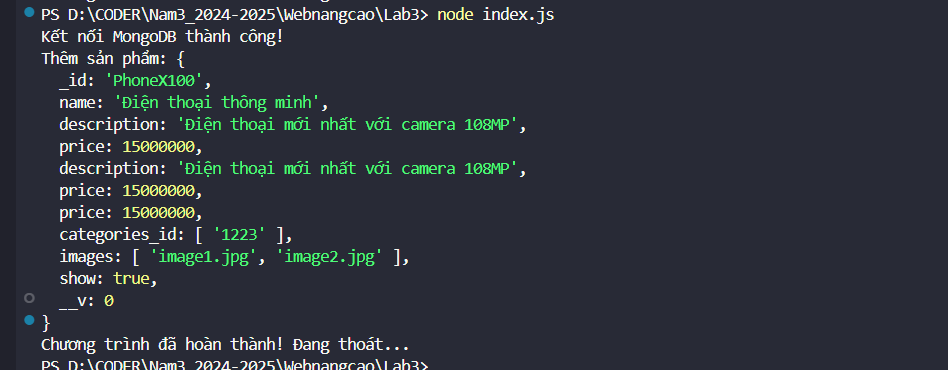
    "Điện thoại mới nhất với camera 108MP",

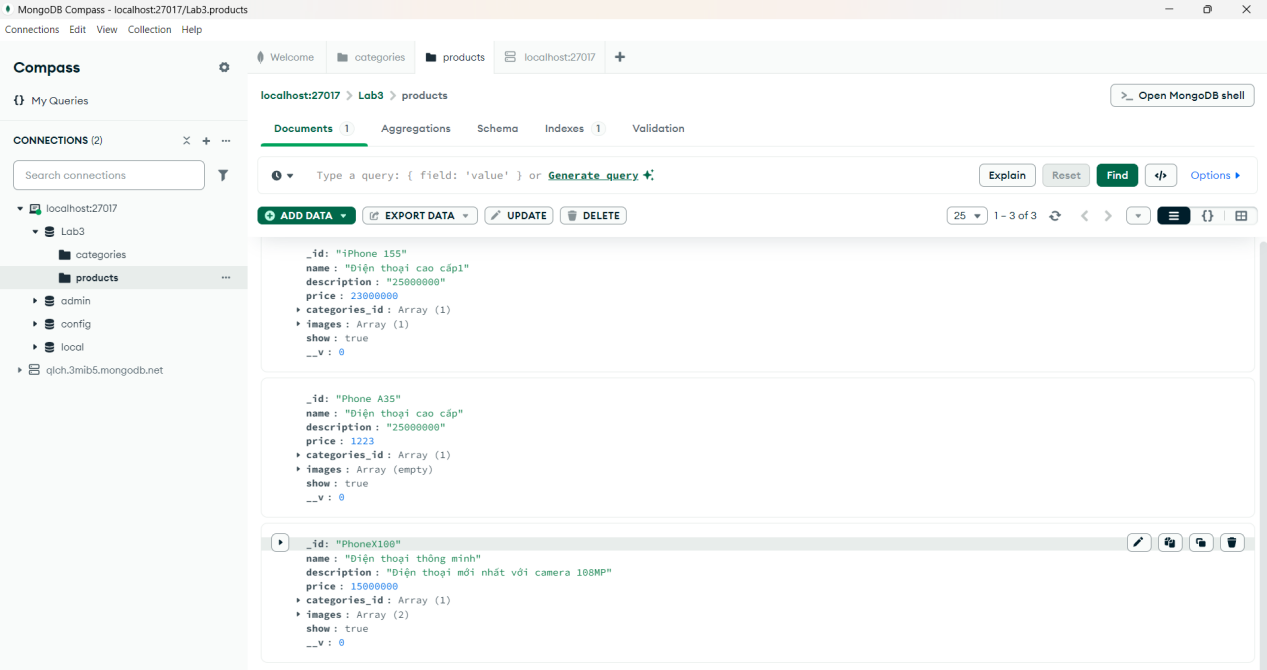
    15000000,

    "1223",

    ["image1.jpg", "image2.jpg"]

  );





+ Tìm sản phẩm theo danh mục bằng id

await getProductsByCategory("1223");

