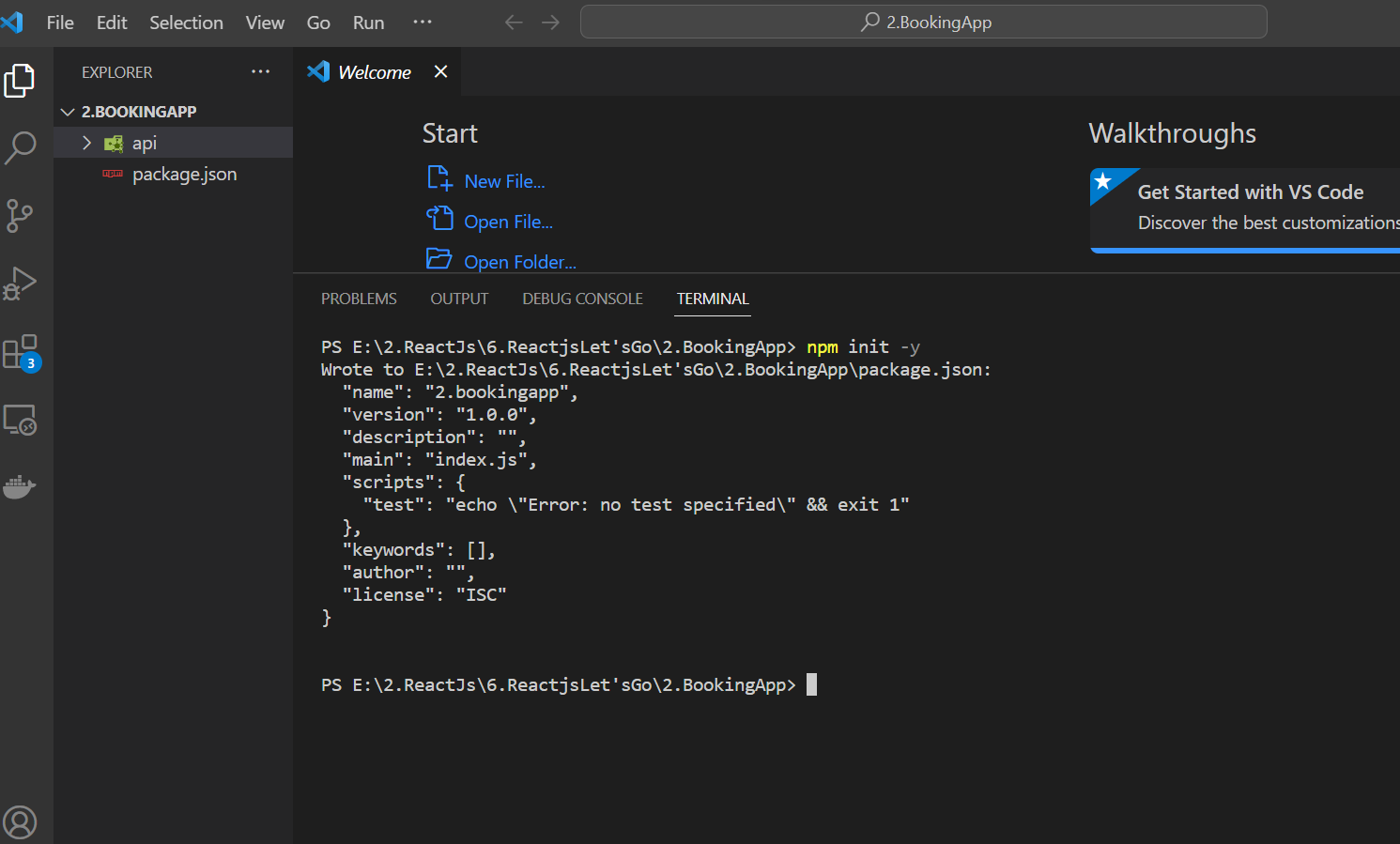
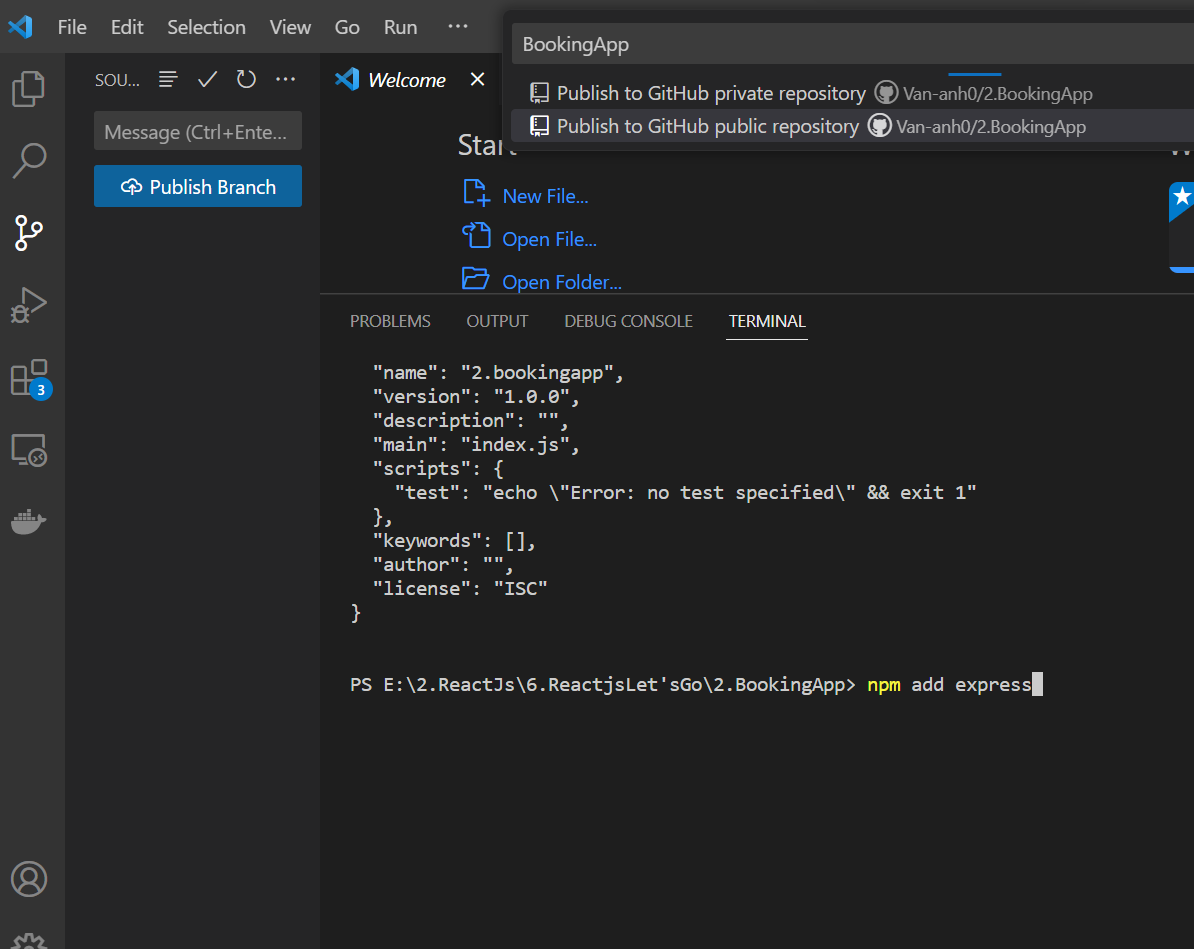
1. Tạo package.json

\*npm init -y



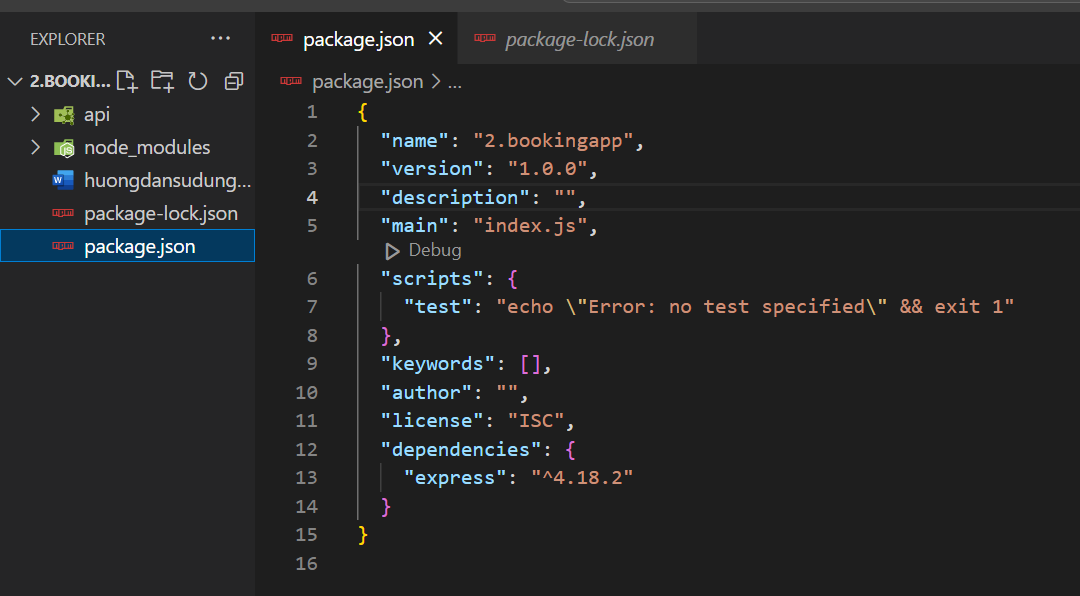
1. Thêm thư viện express

\*npm add express

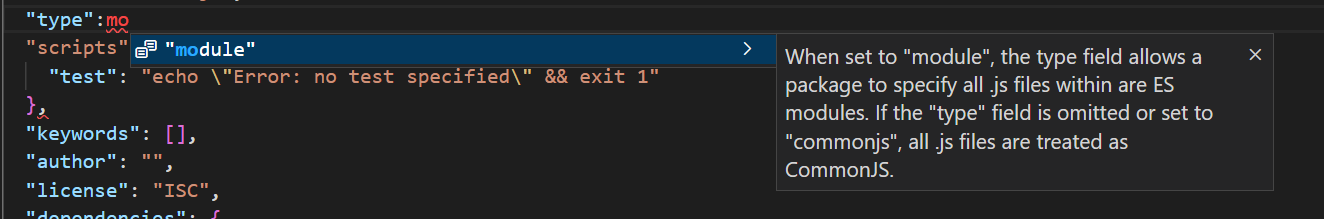


1. Chỉnh sửa package.json

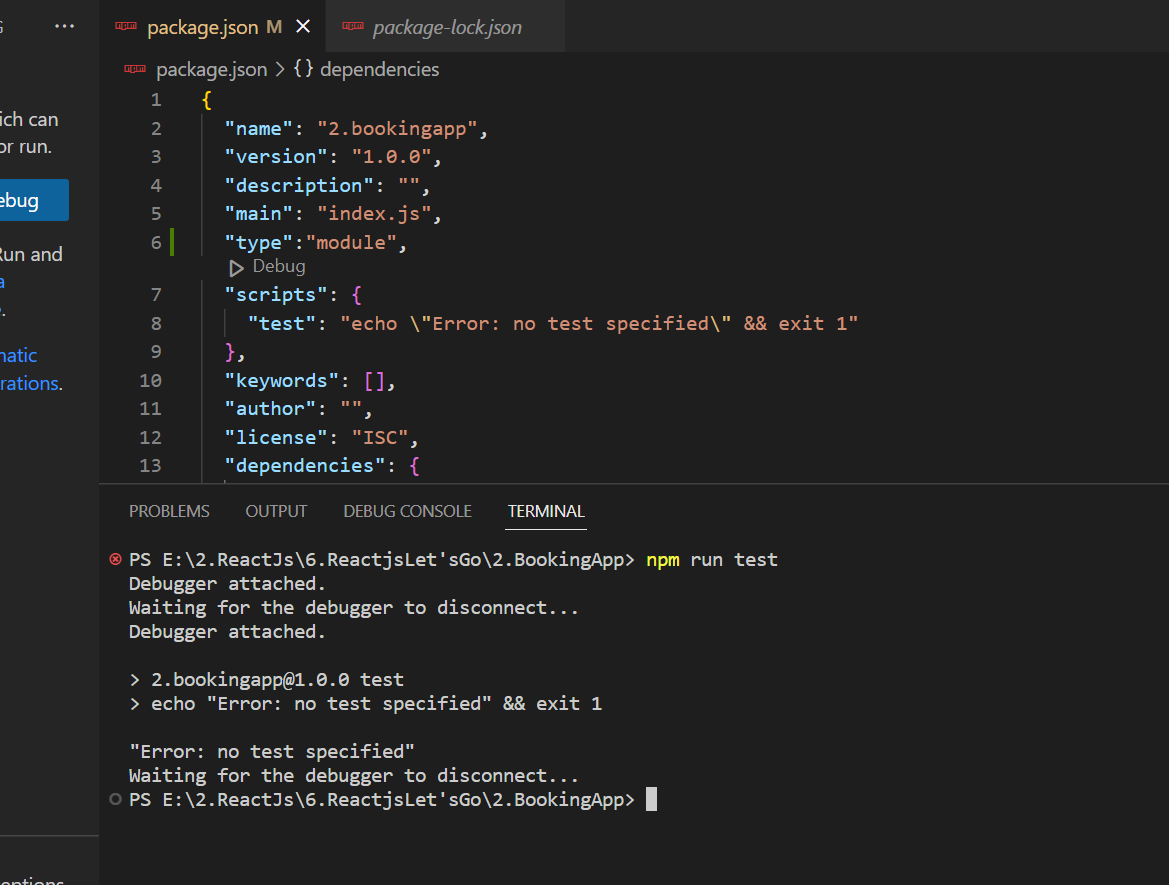
\*Ban đầu



\*thêm type để sử dụng được module

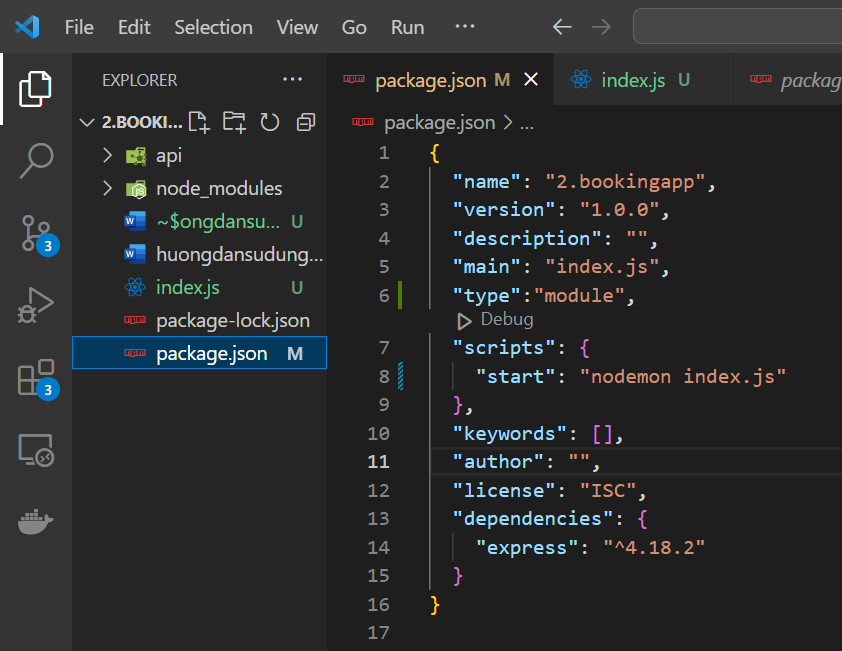


\*chỗ này nhấn vào debug để chạy thử



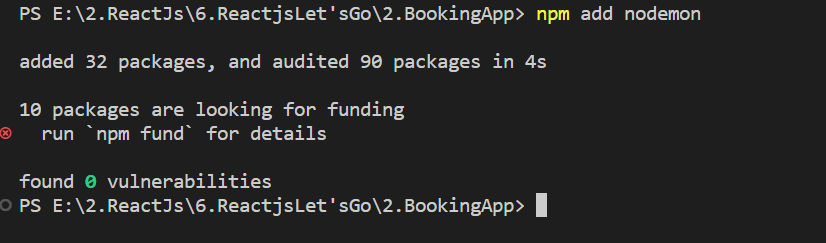
Nó ra: npm run test.

Vậy muốn chuyển nó thành npm start thì…



Tức là khi chạy project thì nó sẽ chạy node ở file index.js

Có thể dùng “start” : “node index.js”

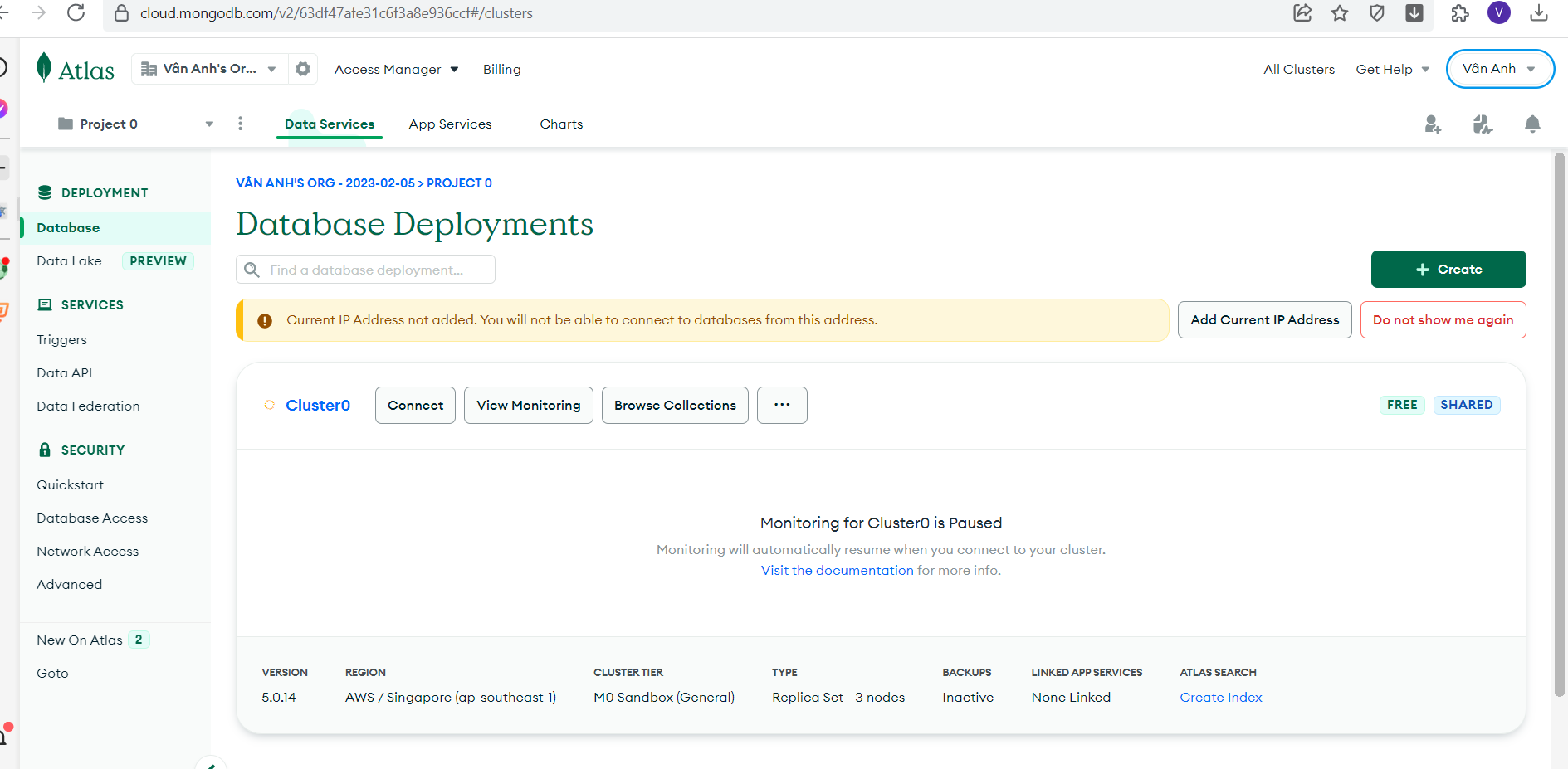


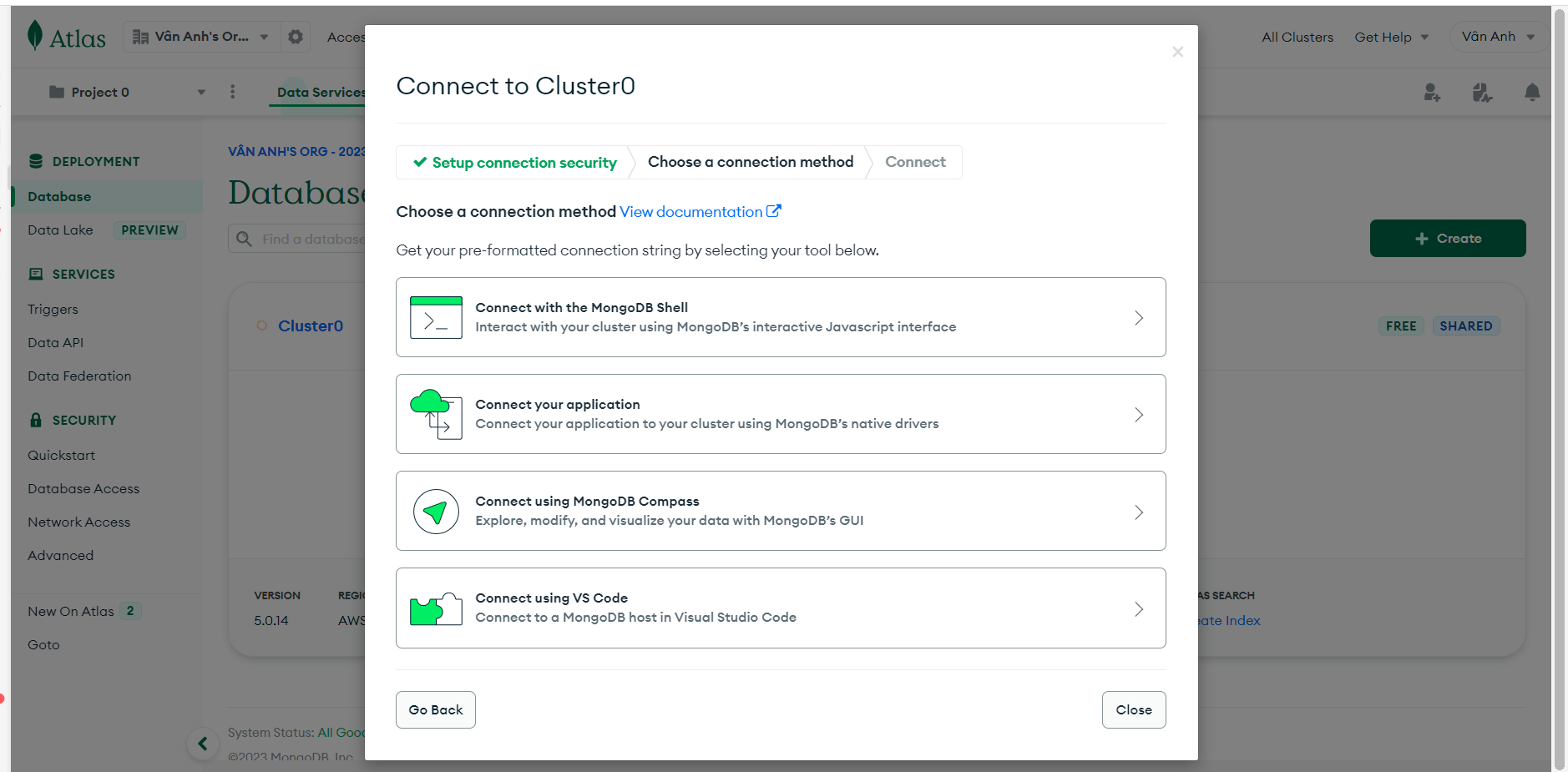
<Sài yarn cũng tương tự

\*yarn add nodemon

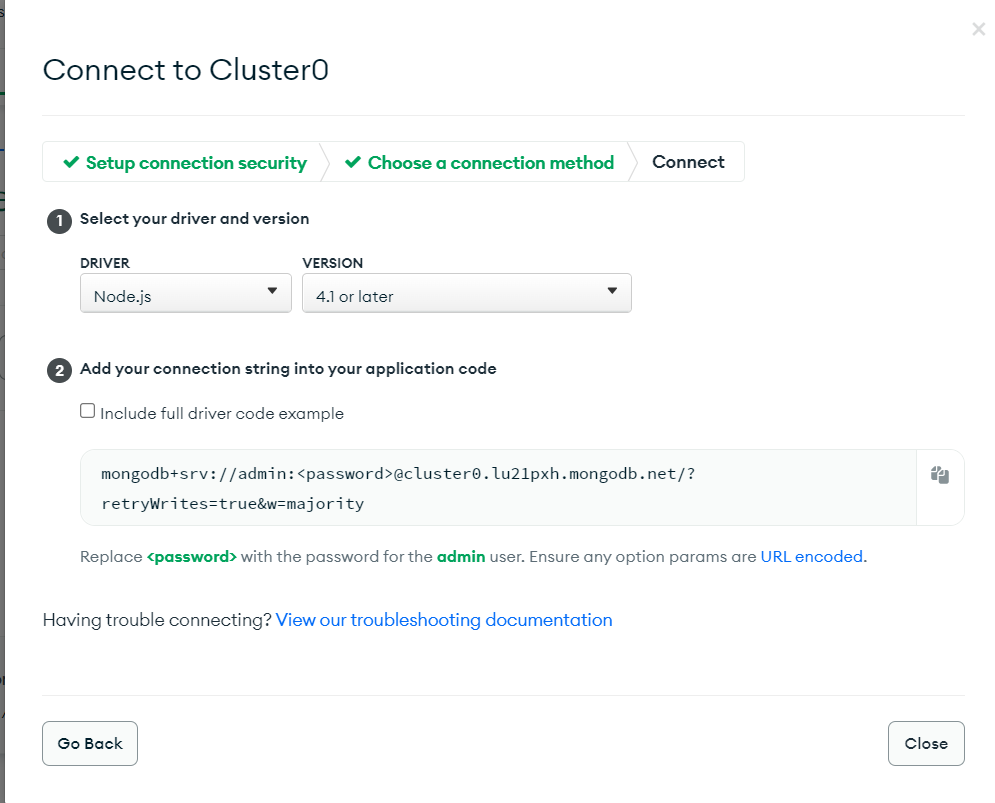
Nhưng khi sài “start” : “nodemon index.js” thì chương trình sẽ tự động reload khi code có sự thay đổi.

1. Đăng nhập vào mongodb atlas. Nếu chưa có thì tạo tài khoản và tạo db





Vô mục db chọn Connect -> Connect your application



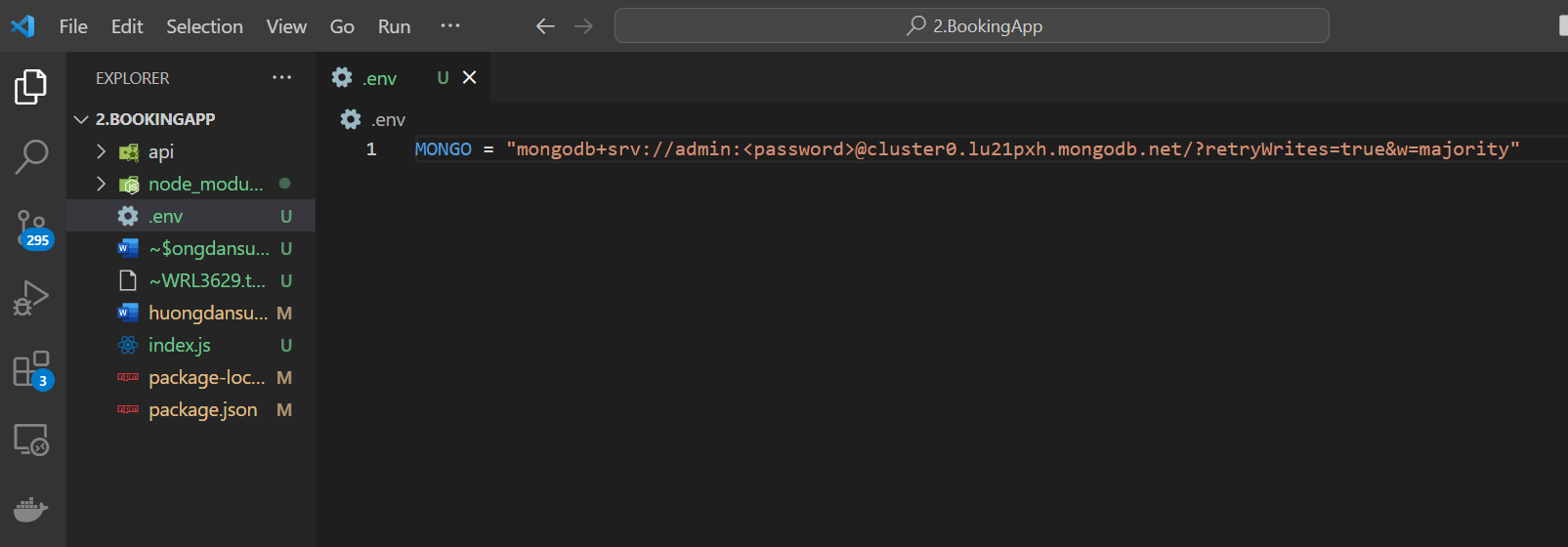
Cop lại cái dòng (2)

\*Lưu ý: chỗ chữ “admin” là tên user

\*Chỗ <password> là nơi thay thế mật khẩu của user đang sài

1. Tạo file .env

Sau đó cop cái trên

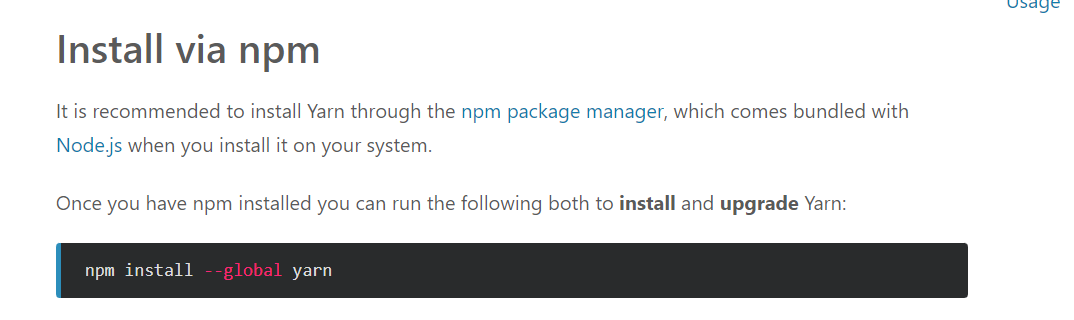


Sau đó thay đổi password cho đúng

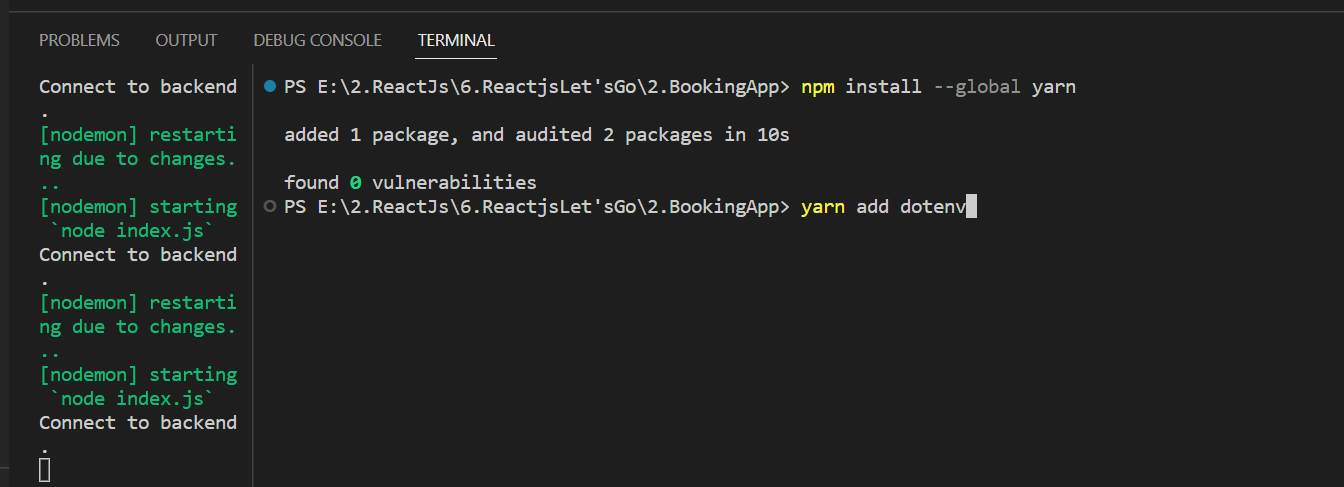
MONGO = "mongodb+srv://admin:Vananh.001@cluster0.lu21pxh.mongodb.net/booking?retryWrites=true&w=majority"

1. Dùng thử yarn nào

Nếu đã cài nodejs rồi thì có thể dùng lệnh npm để cài yarn luôn



1. Để sài được file .env cần add dotenv



Sau đó import thư viện và thêm code vào index.js

import express from "express"

import dotenv from "dotenv"

const app = express()

dotenv.config()

//lắng nghe ở port 5000: ở đây cho bất cứ port nào cũng được, miễn là chưa có ứng dụng nào khác chạy

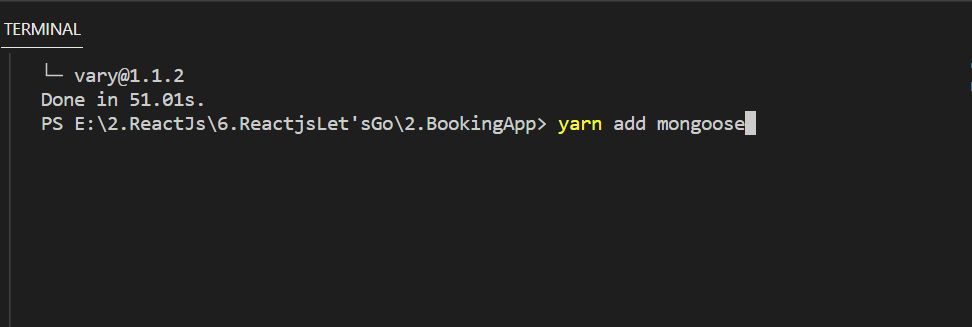
app.listen(5000, ()=> {

    console.log("Connect to backend.");

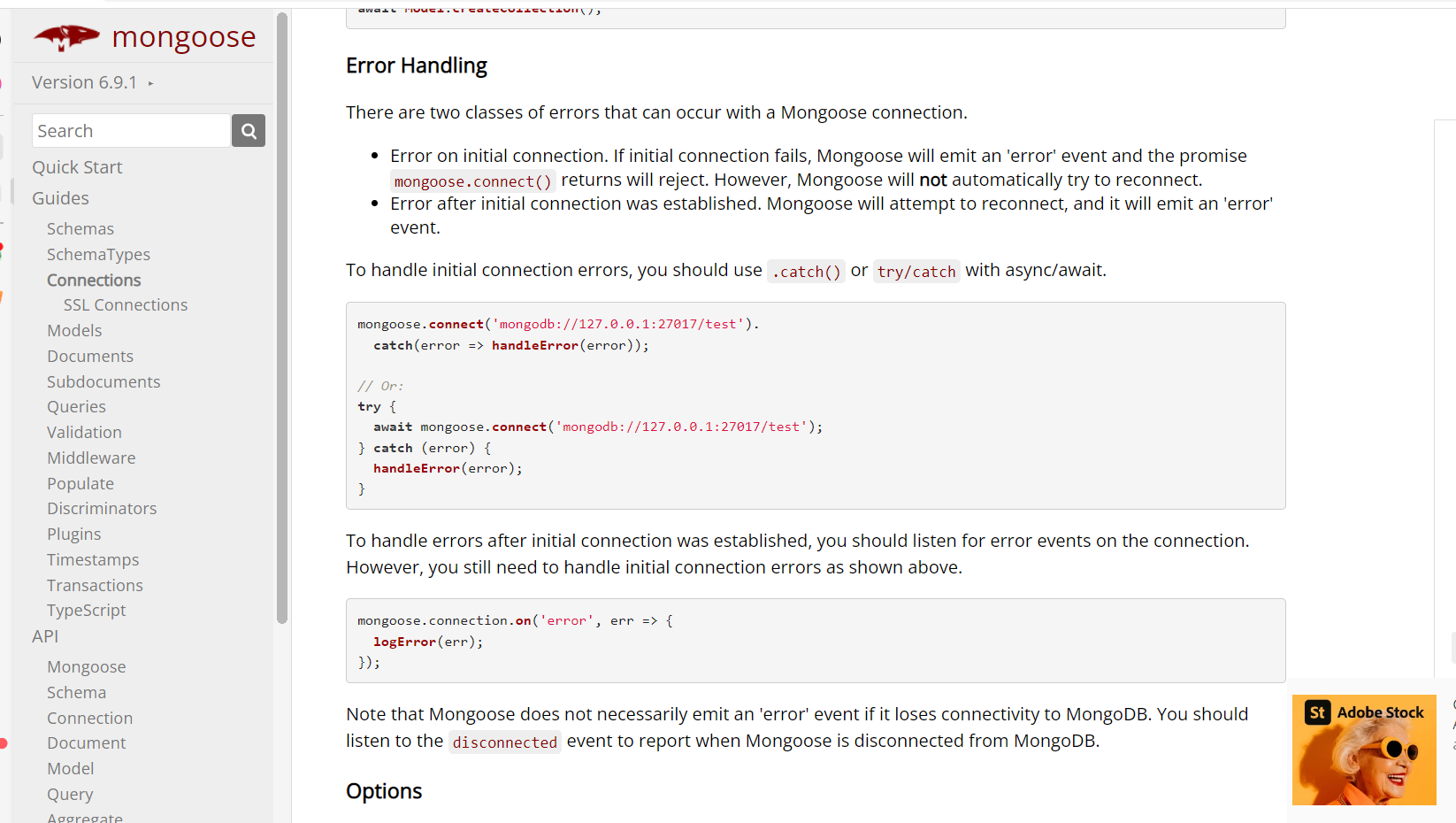
})

Đã kết nối thành công đến mongodb

Vào website này để xem cách dùng mongodb: <https://mongoosejs.com/docs/guide.html>



1. Vào website chọn vào mục connections và lụm sài



Viết hàm connect rồi gọi vào app.listen

Khi nào khởi động server thì cũng tự động kết nối đến mongodb luôn

import express from "express"

import dotenv from "dotenv"

import mongoose from "mongoose"

const app = express()

dotenv.config()

const connect = async() => {

    try {

        await mongoose.connect(process.env.MONGO);

      } catch (error) {

        throw error;

    }

}

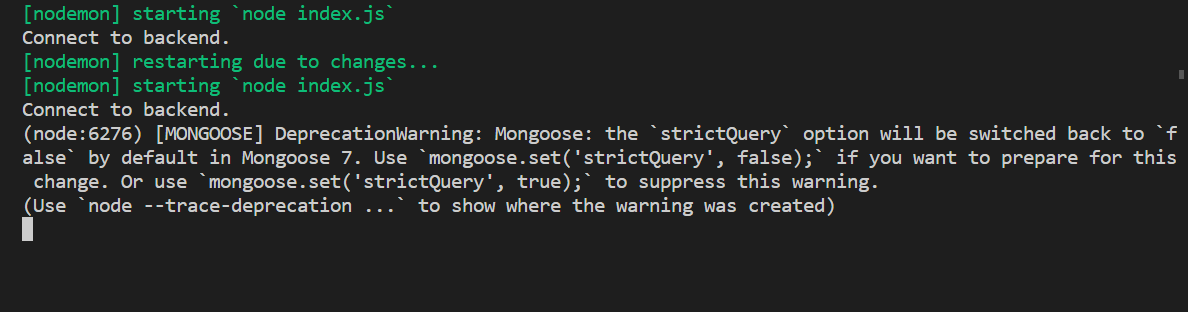
//lắng nghe ở port 5000: ở đây cho bất cứ port nào cũng được, miễn là chưa có ứng dụng nào khác chạy

app.listen(5000, ()=> {

    connect()

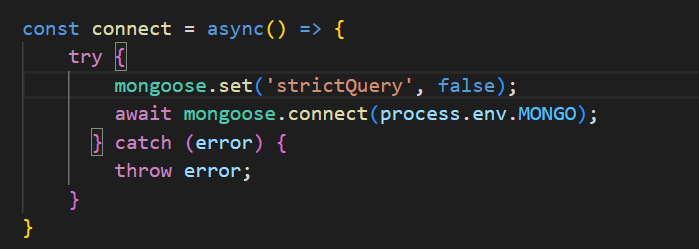
    console.log("Connect to backend.");

})

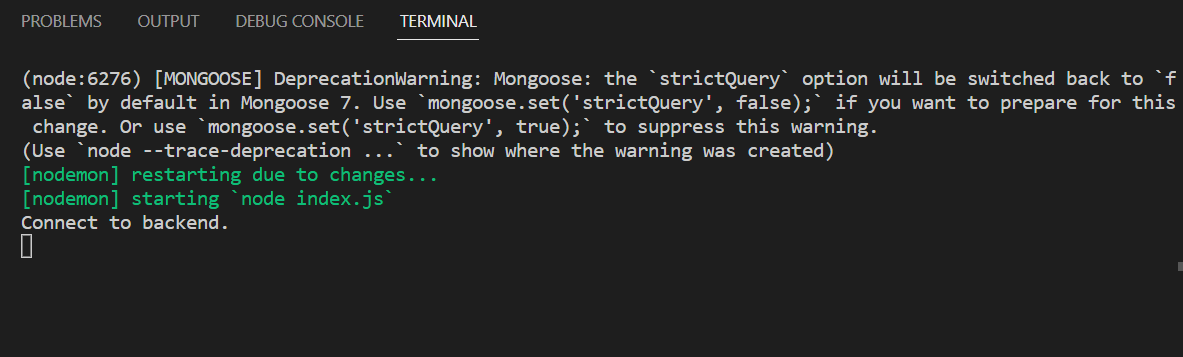


Nó có cảnh báo.

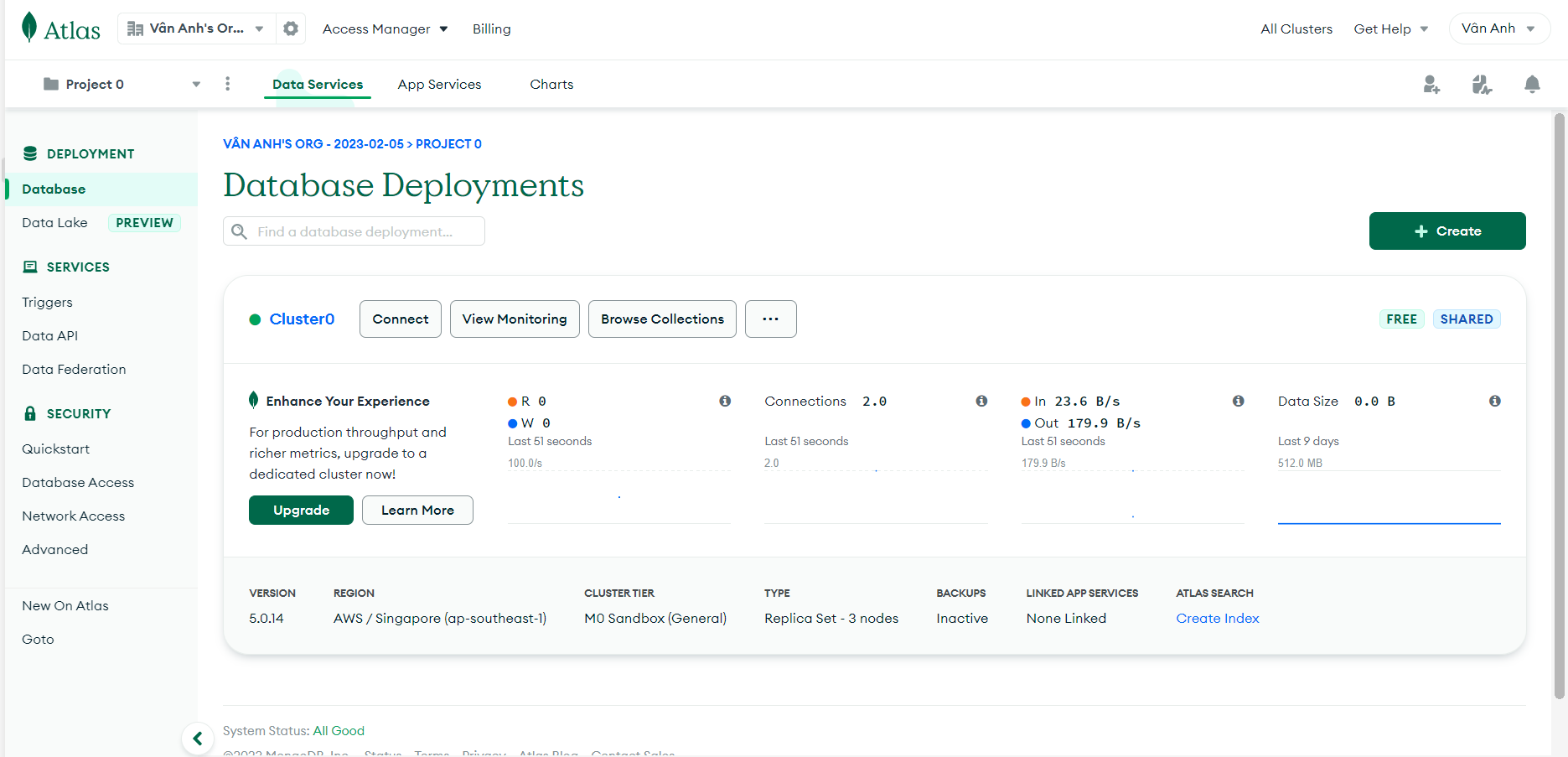
Sửa nè:



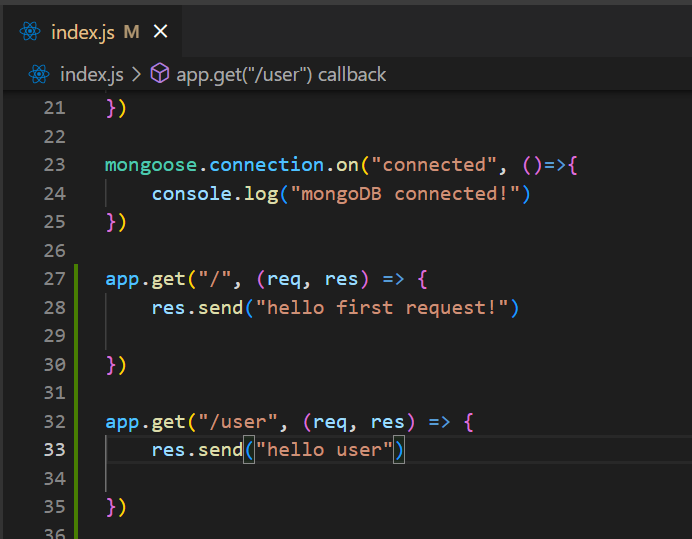
Hết cảnh báo

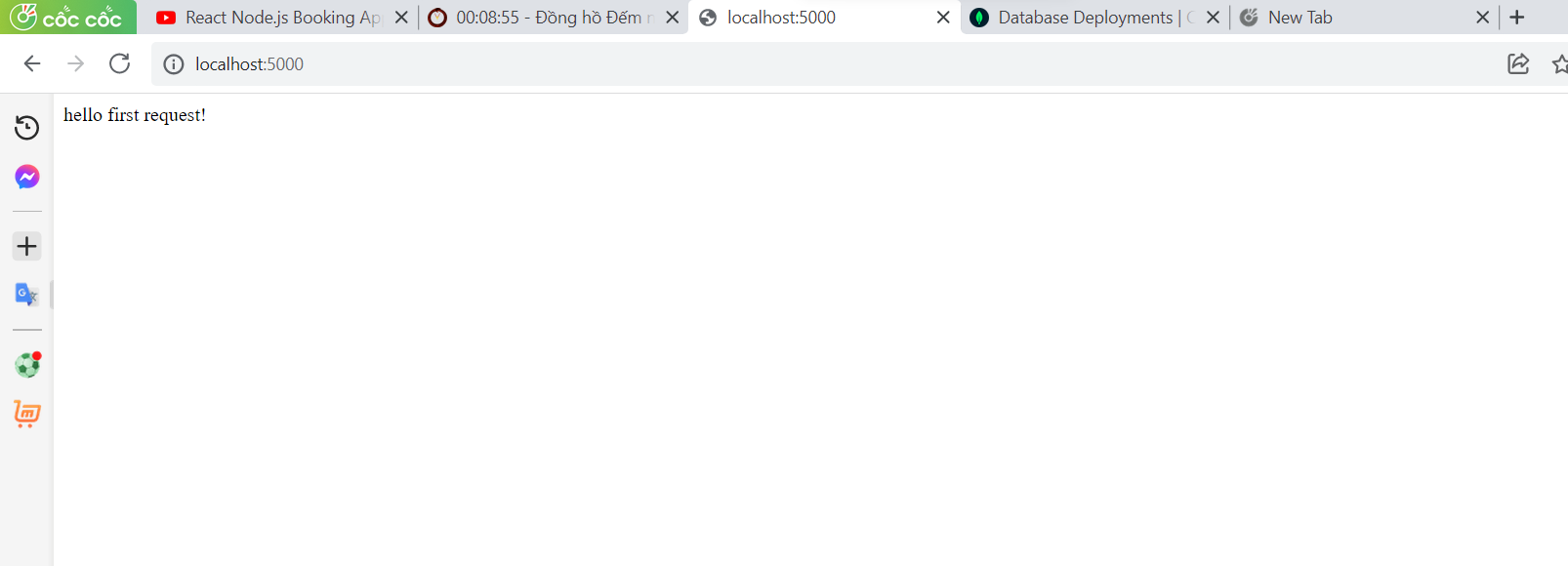


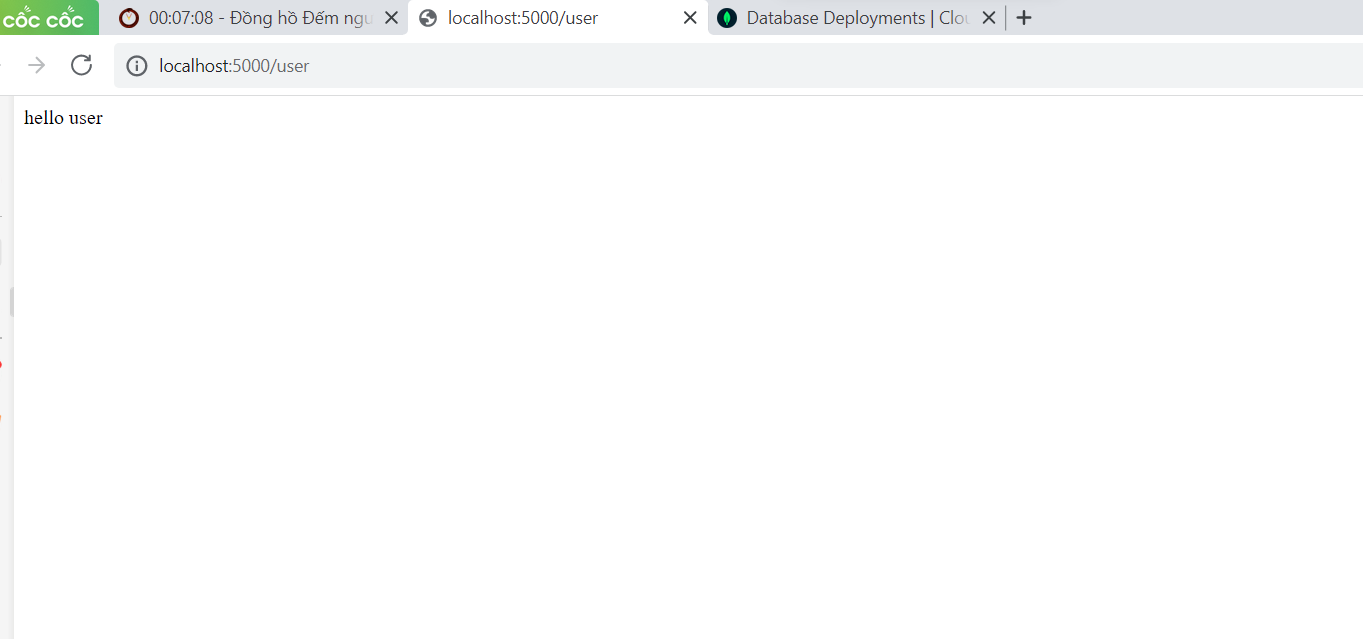
Kiểm tra lại mongo atlas, đã có kết nối đến



1. Tạo router

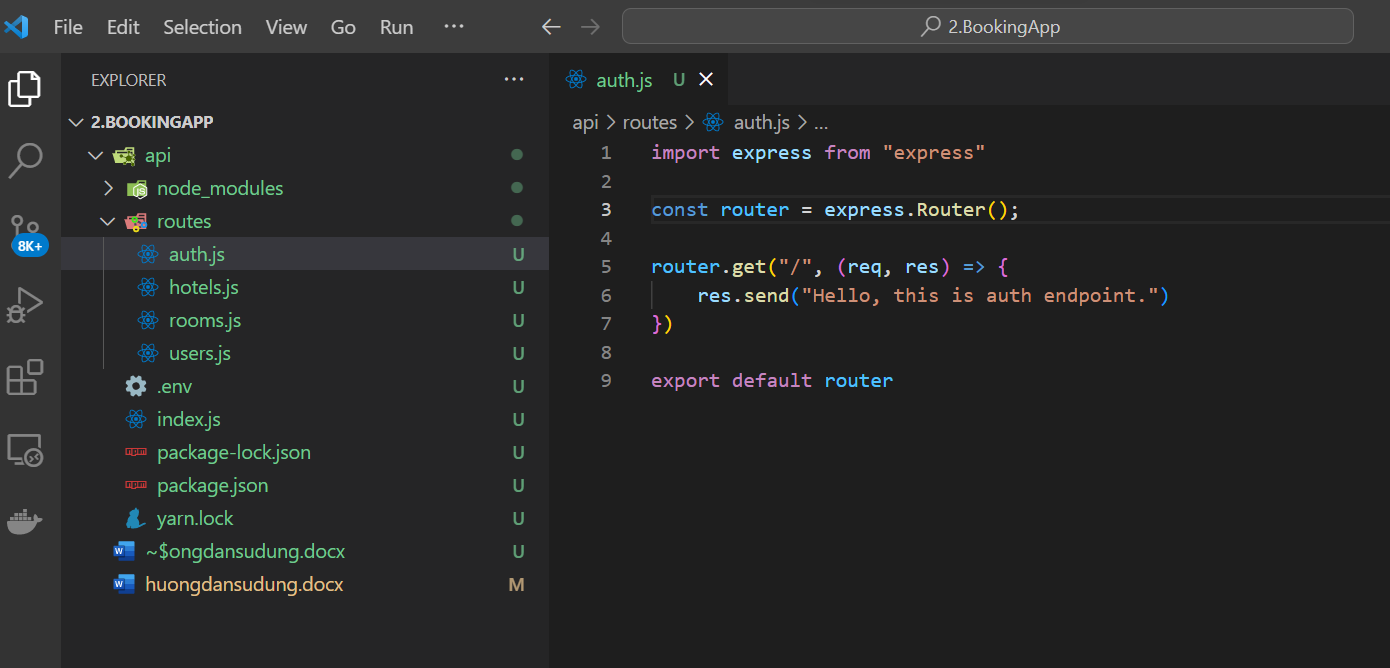






Nhưng vấn đề là chúng ta không thể viết tất cả router vào trong file index.js được

Giờ thì quăng node\_modules vô trong thư mục api và tạo thư mục routes



Quay về file index.js và import để sử dụng

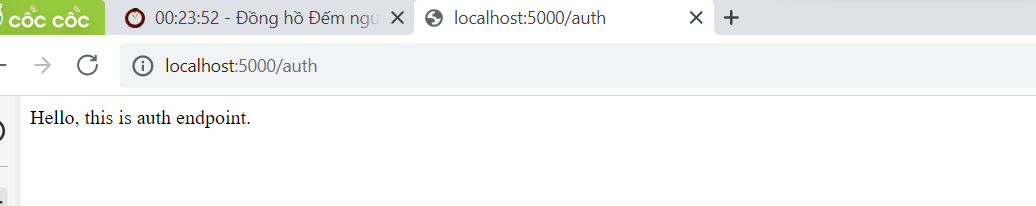
import authRoute from "./routes/auth.js"

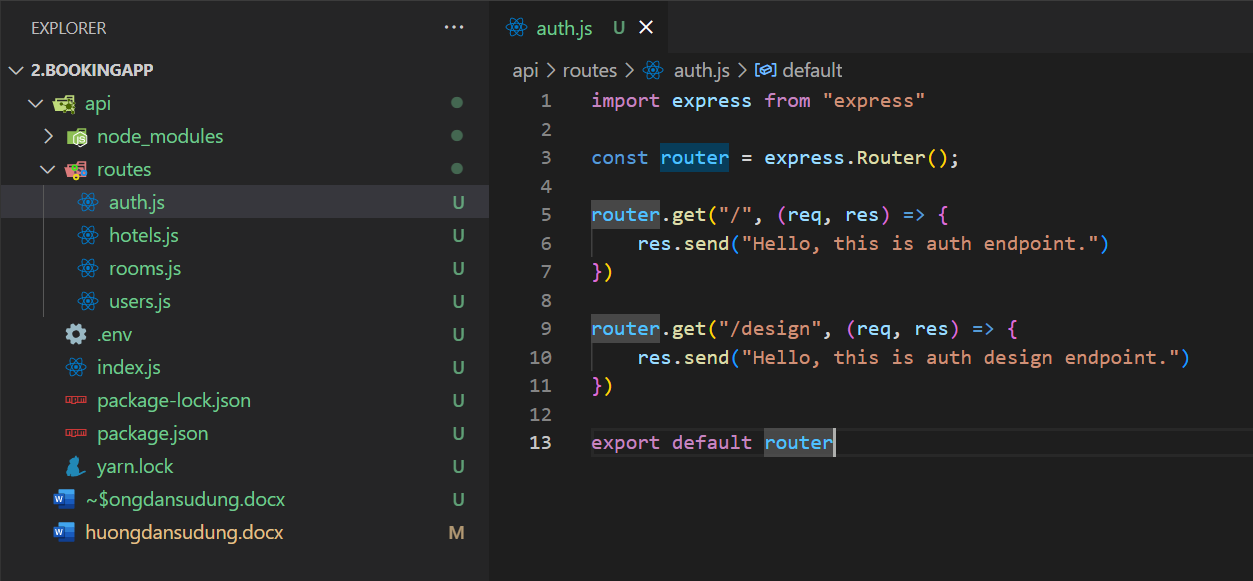
app.get("/", (req, res) => {

    res.send("hello first request!")

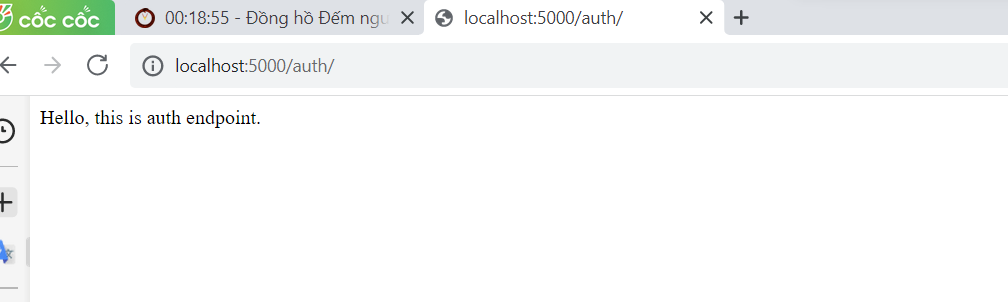
})

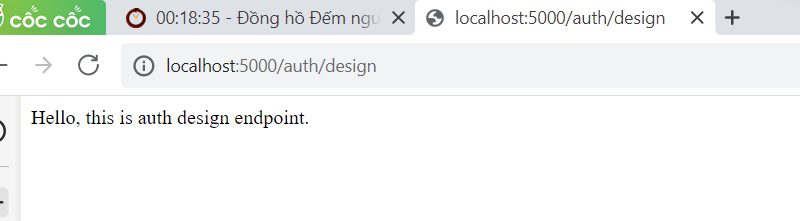
app.use("/auth", authRoute)





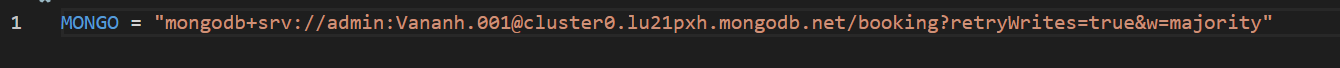
Và giữ nguyên file index.js



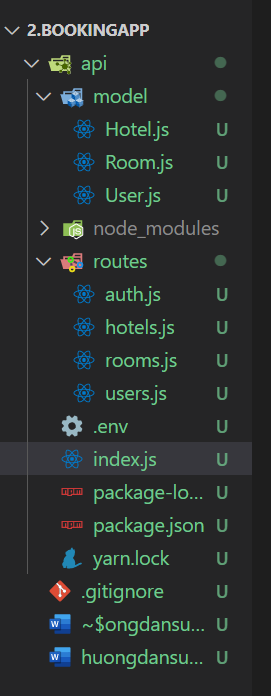


Ô kế, thử đã xong, giờ làm thiệt há

10.



Booking? : là tên cơ sở dữ liệu lưu trong mongodb, nếu chỉ để trống ? thì tự động cơ sở dữ liệu sẽ tên là test



Thêm ở file index.js

import authRoute from "./routes/auth.js"

import hotelsRoute from "./routes/hotels.js"

import roomsRoute from "./routes/rooms.js"

import usersRoute from "./routes/users.js"

app.use(express.json())

app.use("/api/auth", authRoute)

app.use("/api/users", usersRoute)

app.use("/api/hotels", hotelsRoute)

app.use("/api/rooms", roomsRoute)

Tạo model

Hotel.js

import mongoose from "mongoose"

const {Schema} = mongoose;

const HotelSchema = new mongoose.Schema({

    name:{

        type:String,

        required: true

    },

    type:{

        type: String,

        required: true

    },

    city:{

        type:String,

        required: true

    },

    address:{

        type:String,

        required:true

    },

    distance:{

        type:String,

        required:true

    },

    title:{

        type:[String],

    },

    desc:{

        type:String,

        required:true

    },

    rating:{

        type:Number,

        min:0,

        max:5

    },

    rooms:{

        type:[String],

    },

    cheapestPrice:{

        type:Number,

        required:true

    },

    featured: {

        type: Boolean,

        default:false,

    },

})

export default mongoose.model("Hotel", HotelSchema)

Trong mục router, file hotel.js

import express from "express"

import Hotel from "../model/Hotel.js"

const router = express.Router();

//CREATE

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

    const newHotel = new Hotel(req.body)

    try{

        const savedHotel = await newHotel.save()

        res.status(200).json(savedHotel)

    }catch(err){

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

    }

})

//UPDATE

//DELETE

//GET

//GET ALL

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

    try{

        const savedHotel = await Hotel.find()

        res.status(200).json(savedHotel)

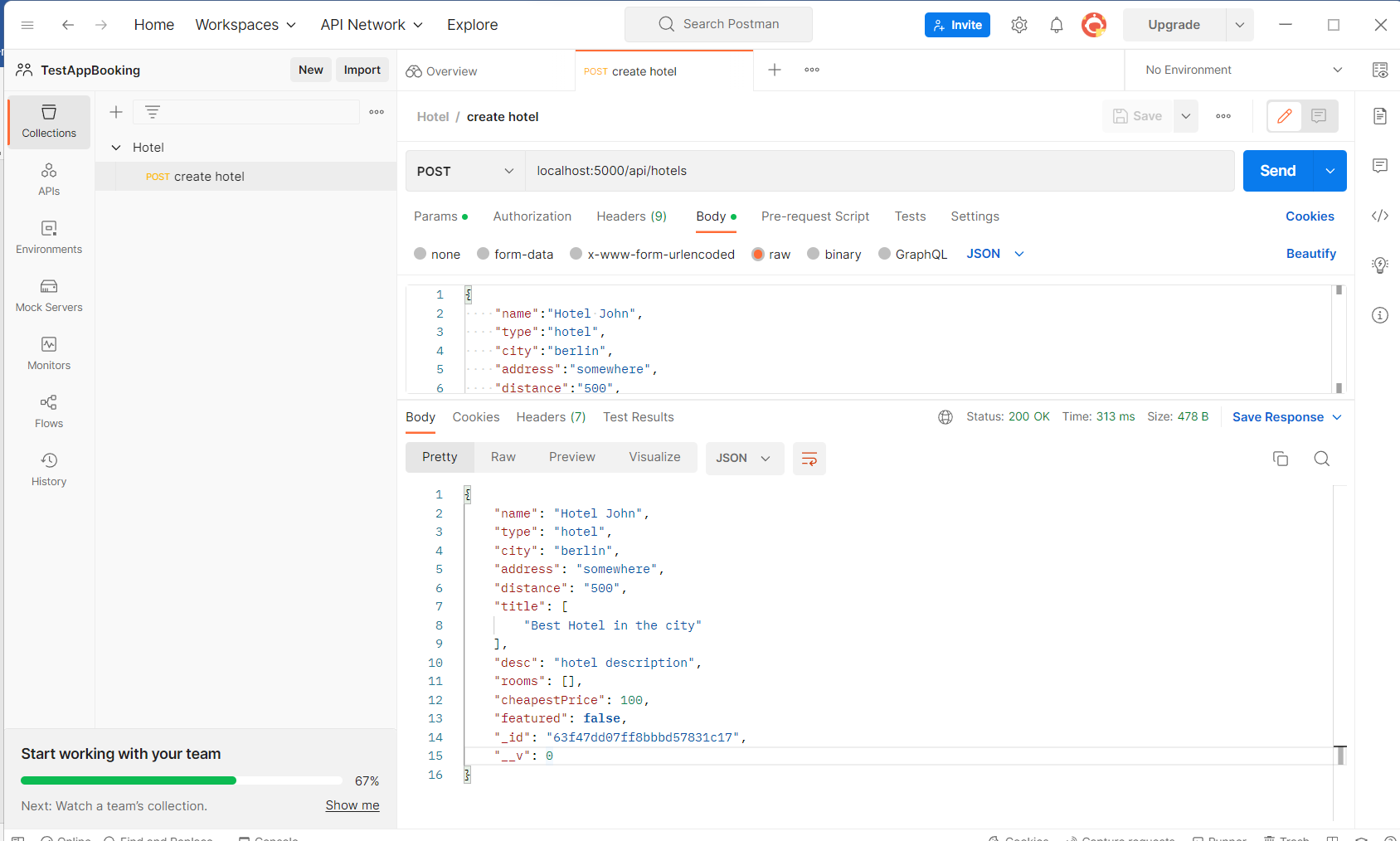
    }catch(err){

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

    }

})

export default router



Sử dụng postman để kiểm tra

catch(err){

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

    }

Khi bị lỗi thì nó sẽ xuất lỗi rõ ràng dạng json

Hàm update

//UPDATE

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

    try{

        const updatedHotel = await Hotel.findByIdAndUpdate(req.params.id, {$set: req.body})

        res.status(200).json(updatedHotel);

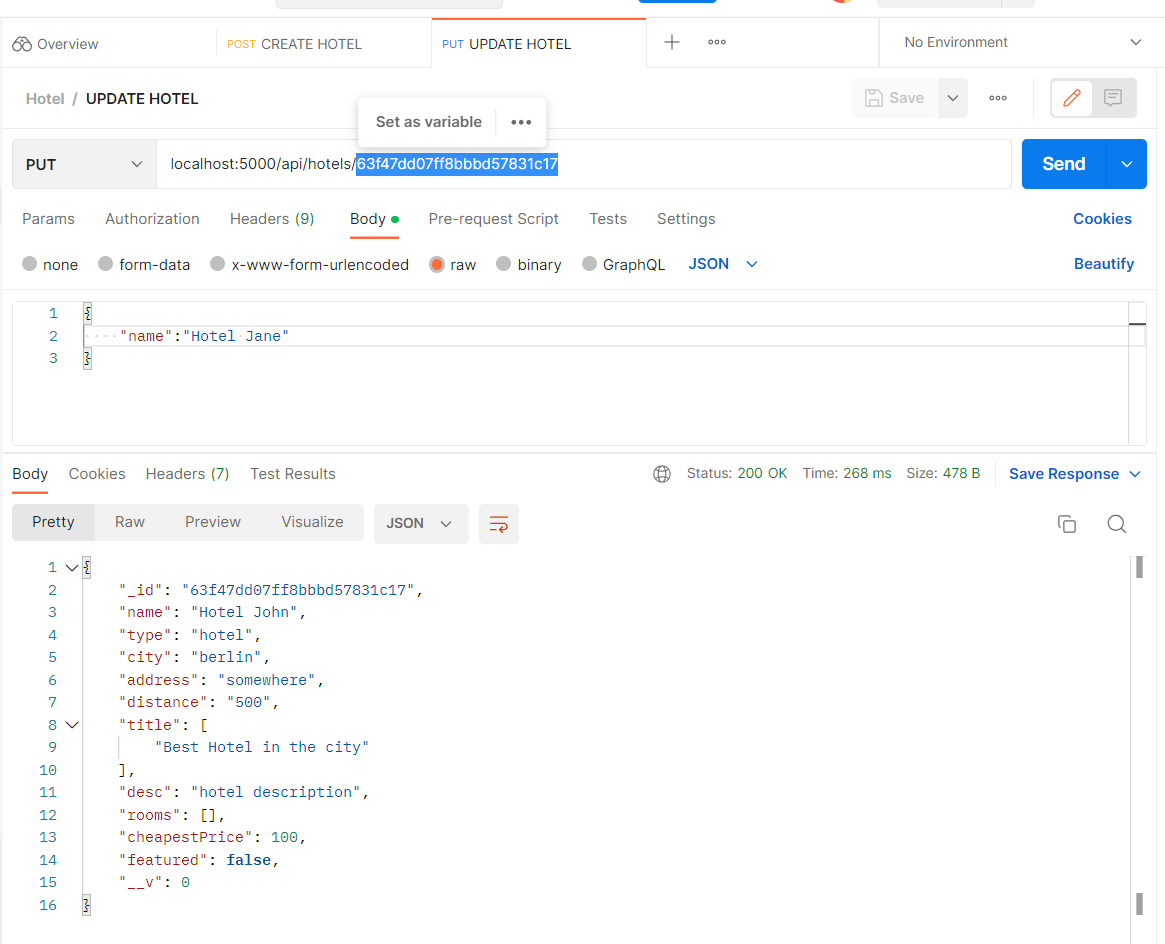
    }catch(err){

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

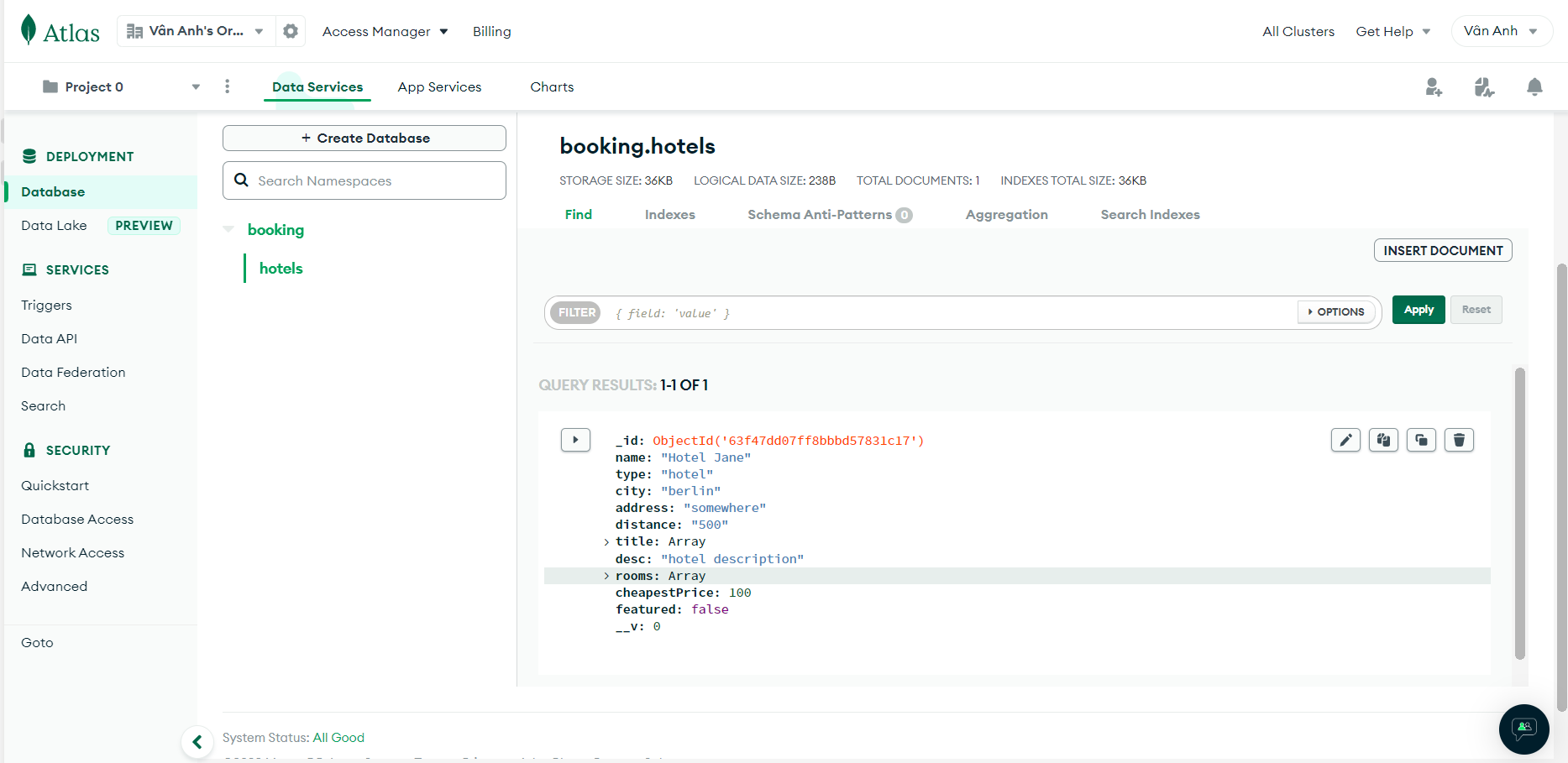
    }

})

Ta sẽ sử dụng hàm findByIdAndUpdate. Truyền vào id của hotel cần sửa, sau đó sử dụng $set: req.body để thay đổi dữ liệu theo ý muốn



Ban đầu cái name của hotel là John, sửa lại thành Jane



Nhưng có một điều là postman không thay đổi, vậy nên ta thêm new: true vào code

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

    try{

        const updatedHotel = await Hotel.findByIdAndUpdate(req.params.id, {$set: req.body},{new:true})

        res.status(200).json(updatedHotel);

    }

