LAPORAN PRAKTIKUM
MATA KULIAH PEMROGRAMAN BERORIENTASI OBJEK

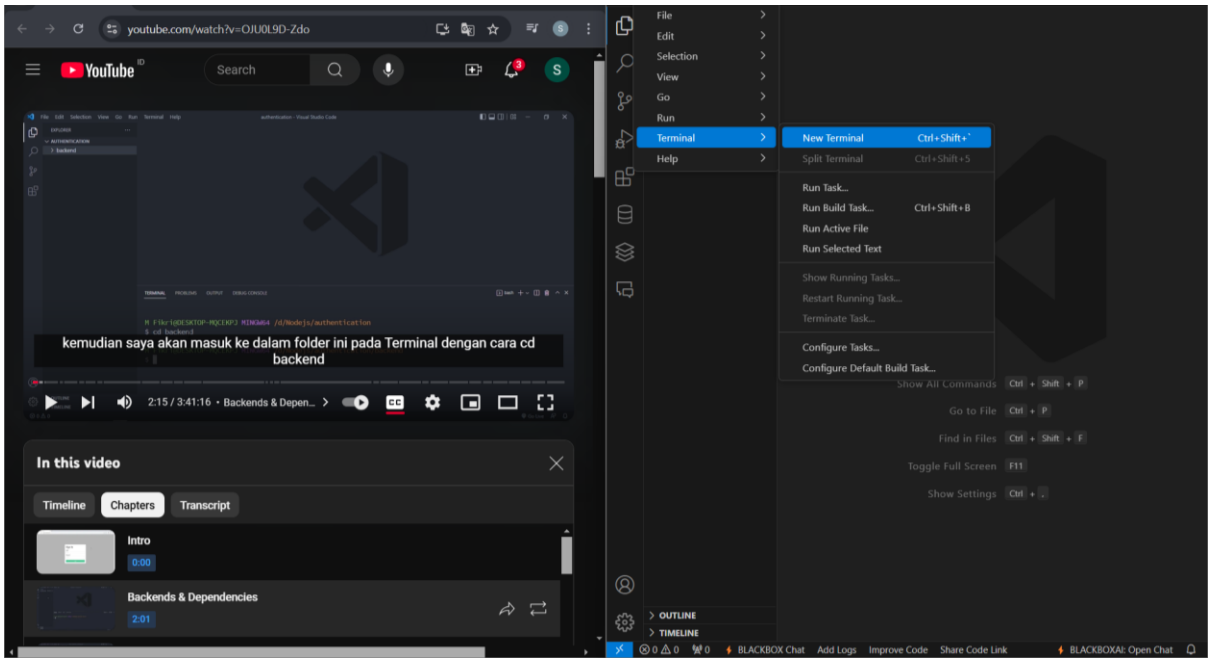
Multi Role Login using Node JS, Express, MySQL, and React JS

Dosen Pengampu :
Willdan Aprizal Arifin, S.Pd., M.Kom.

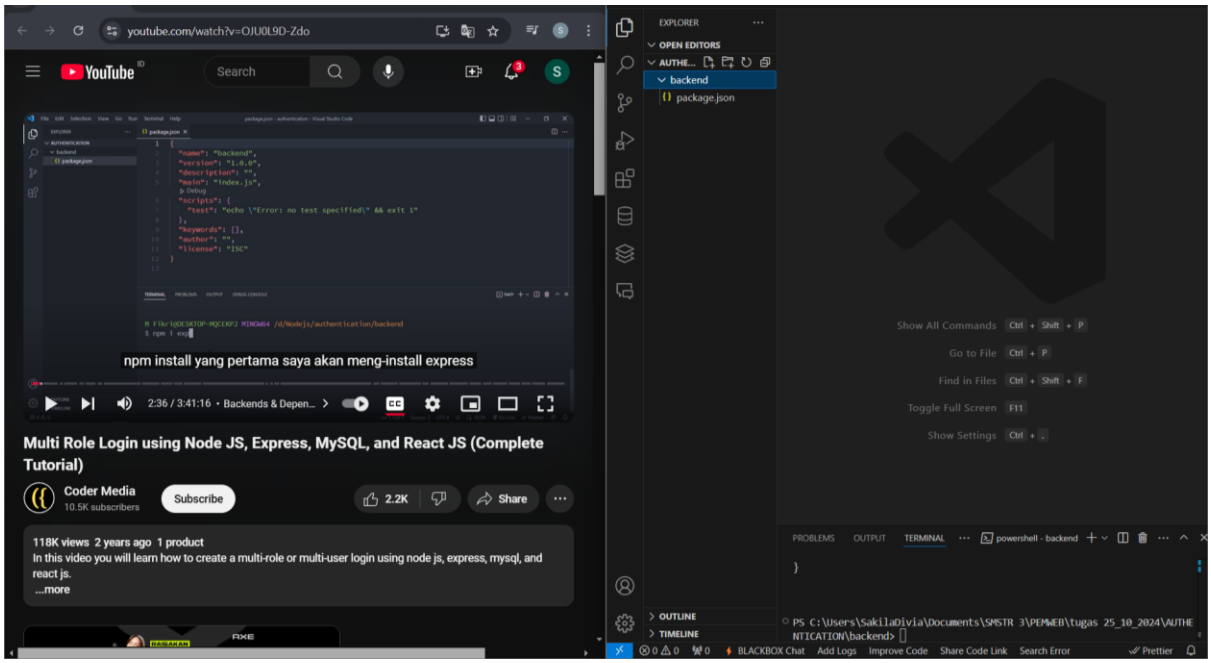
Disusun Oleh :
Sakila Divia F
2302023

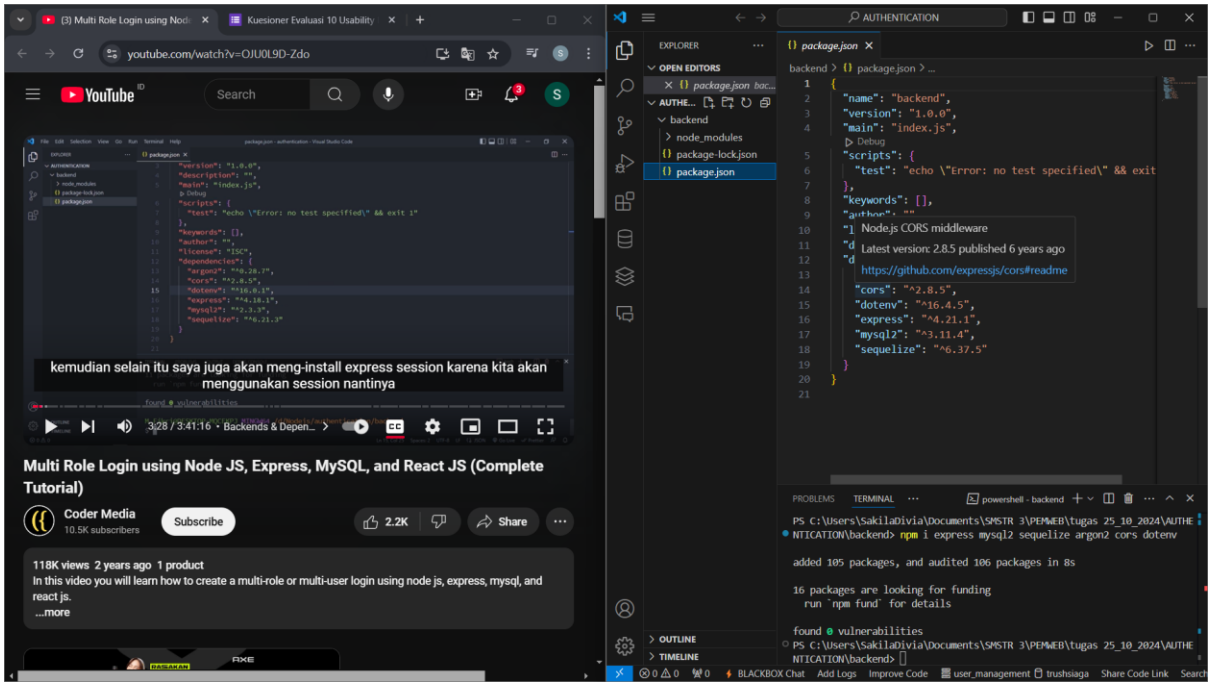
PROGRAM STUDI SISTEM INFORMASI KELAUTAN
UNIVERSITAS PENDIDIKAN INDONESIA
2024

Buka forder dan buka terminal baru

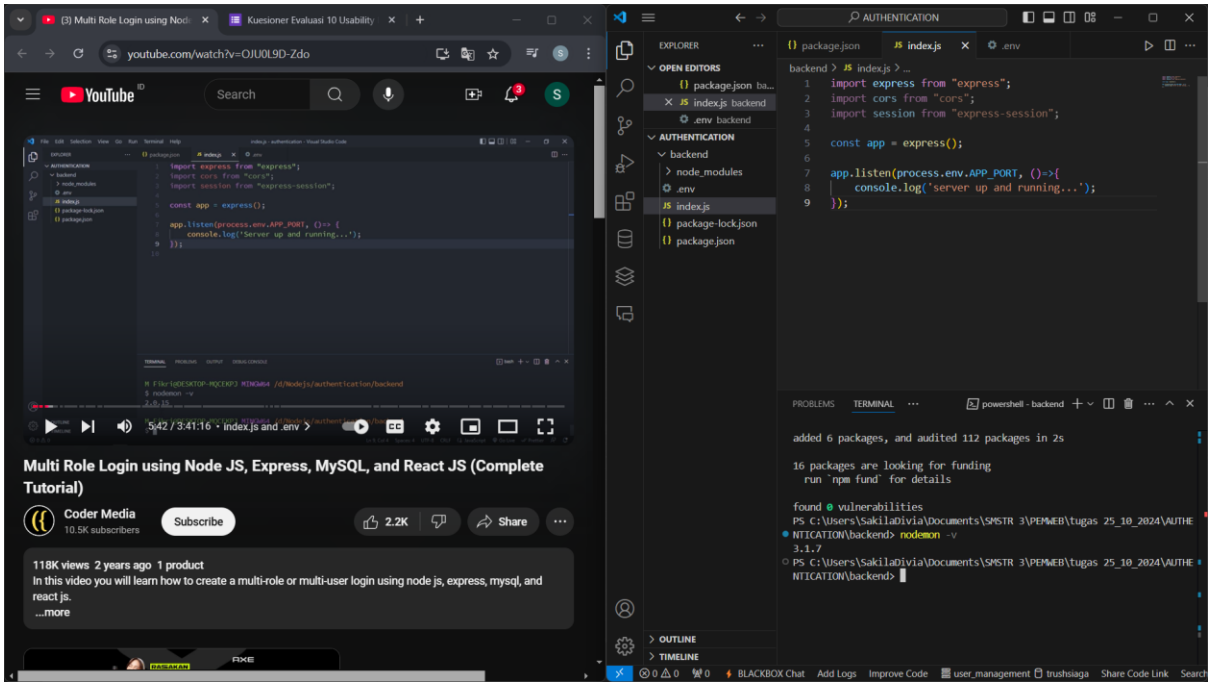


Lakukan npm install

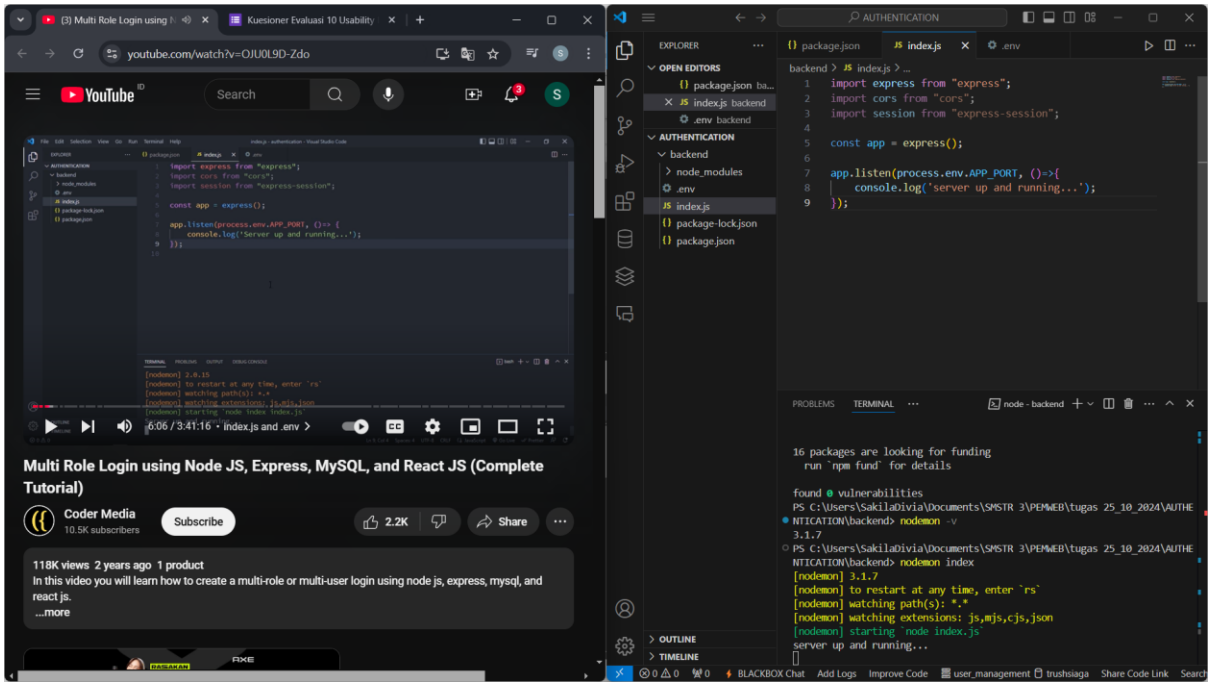




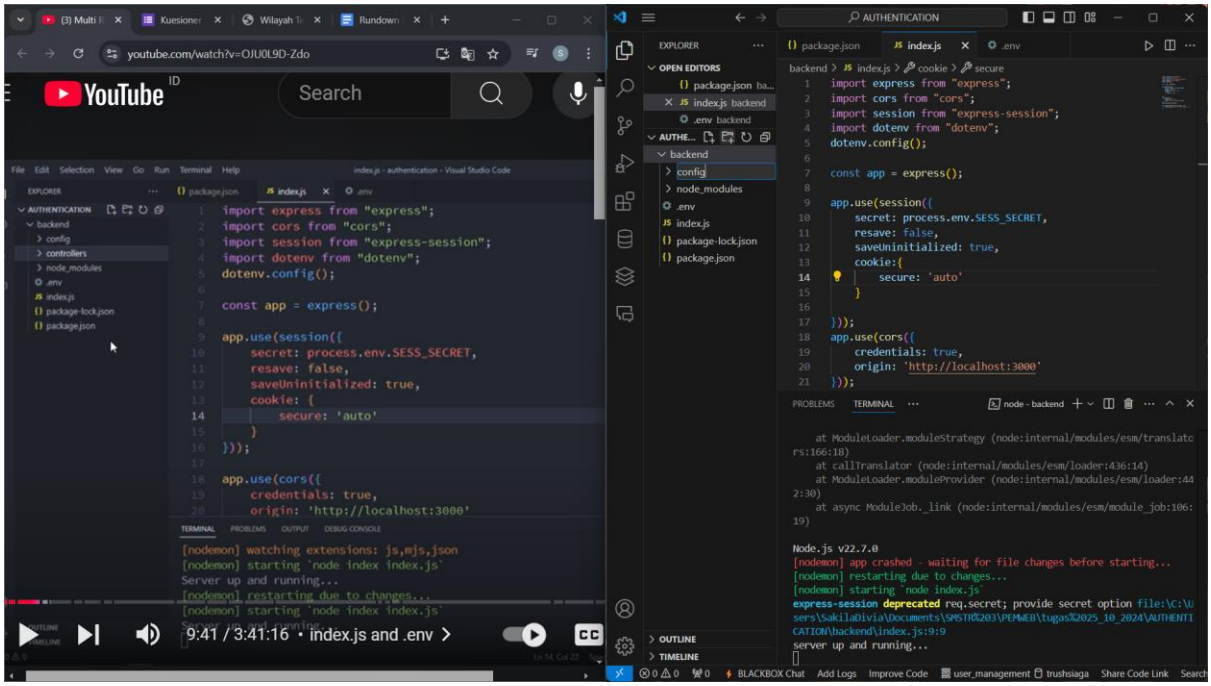
Install nodemon -v untuk

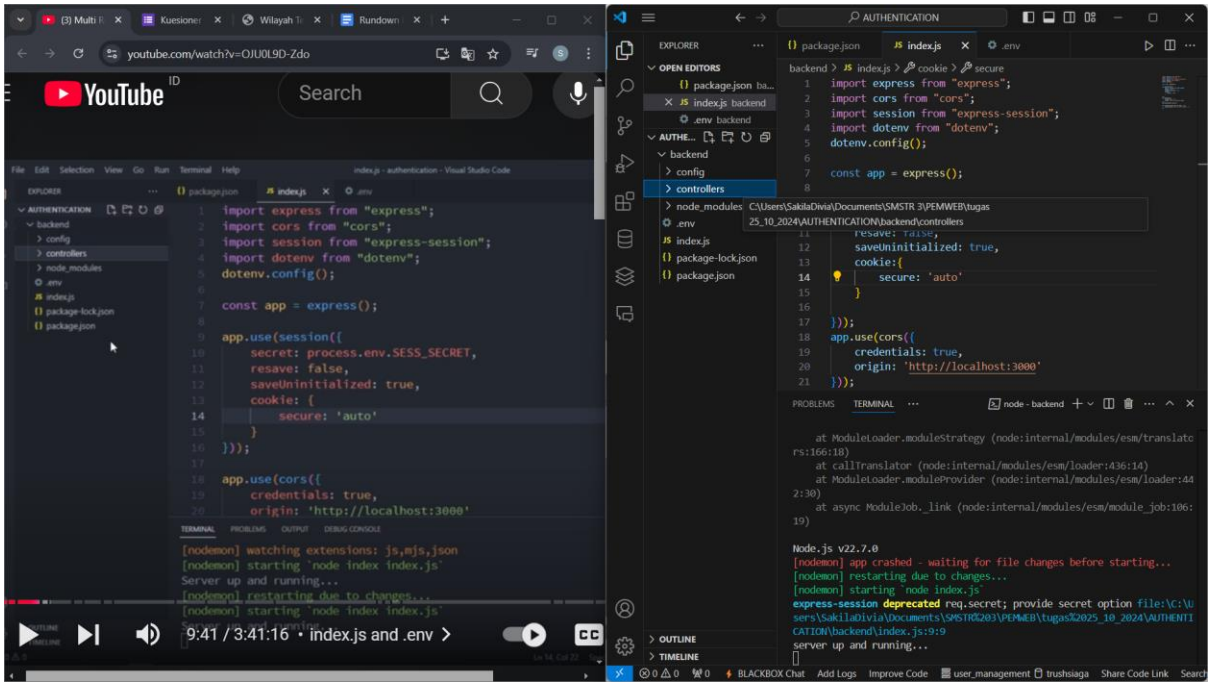


Selanjutnya untuk mengecek apakah aplikasi kita dapat berjalan dengan baik bisa klik nodemon indeks, jika bertuliskan server up and running maka aplikasinya bisa berjalan dengan baik. Seperti gambar dibawah ini.

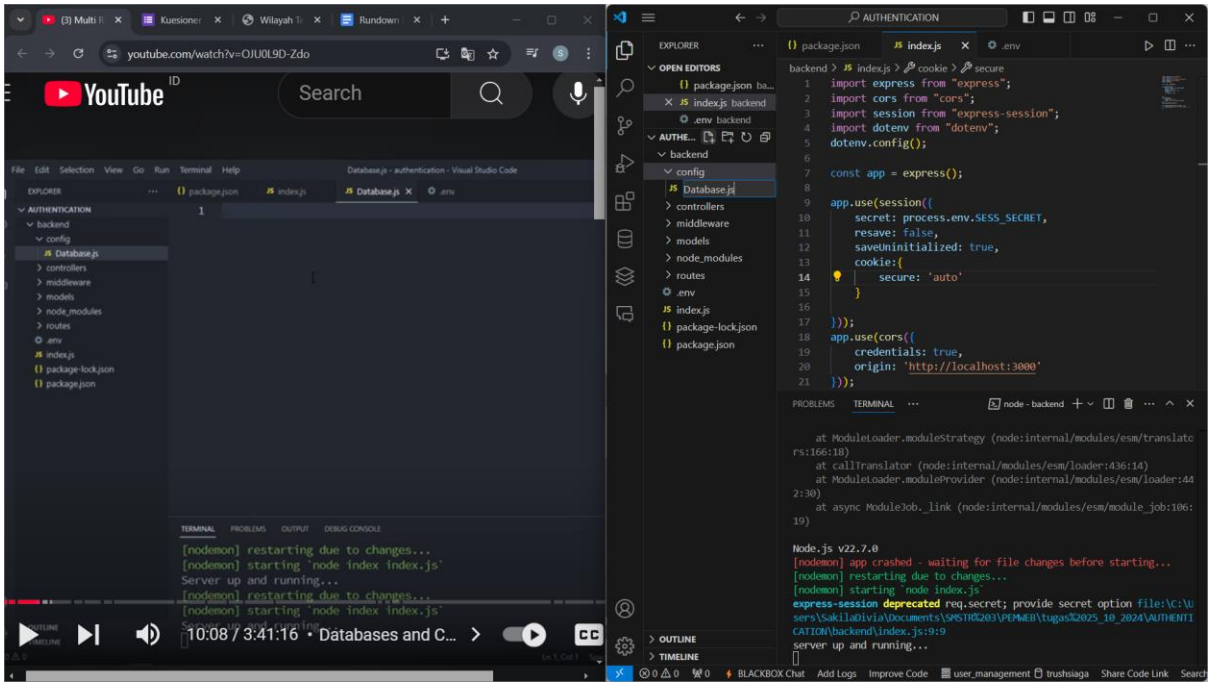


Membuat folder dalam backend

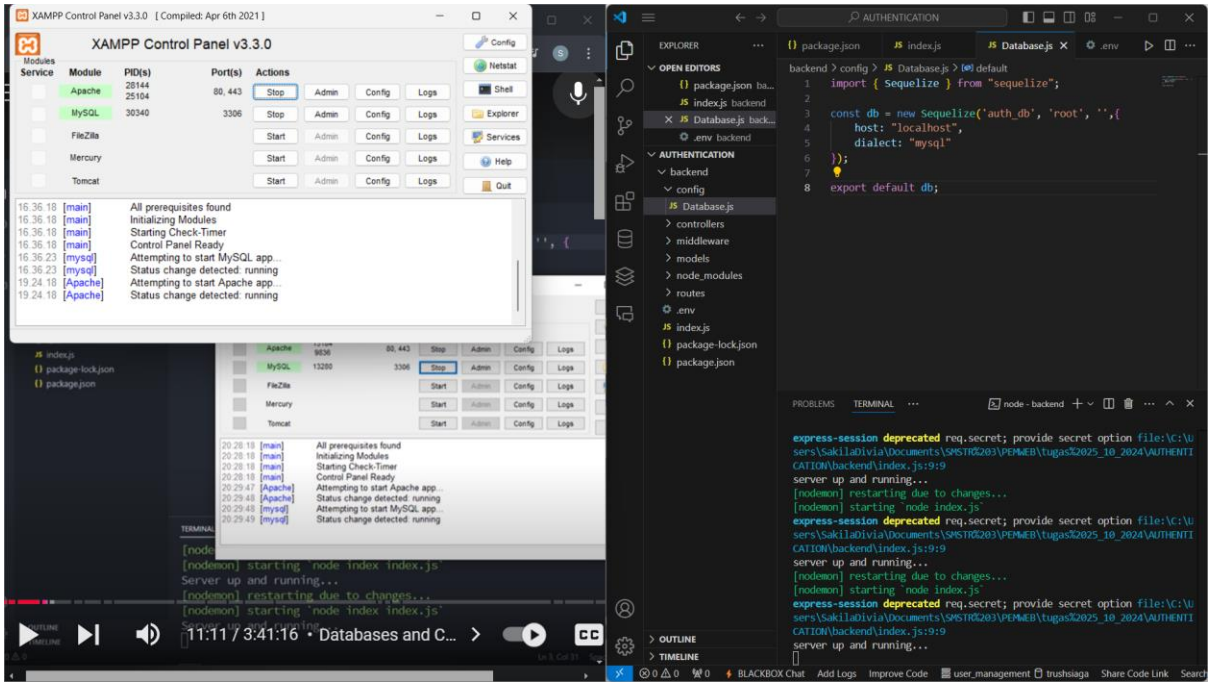




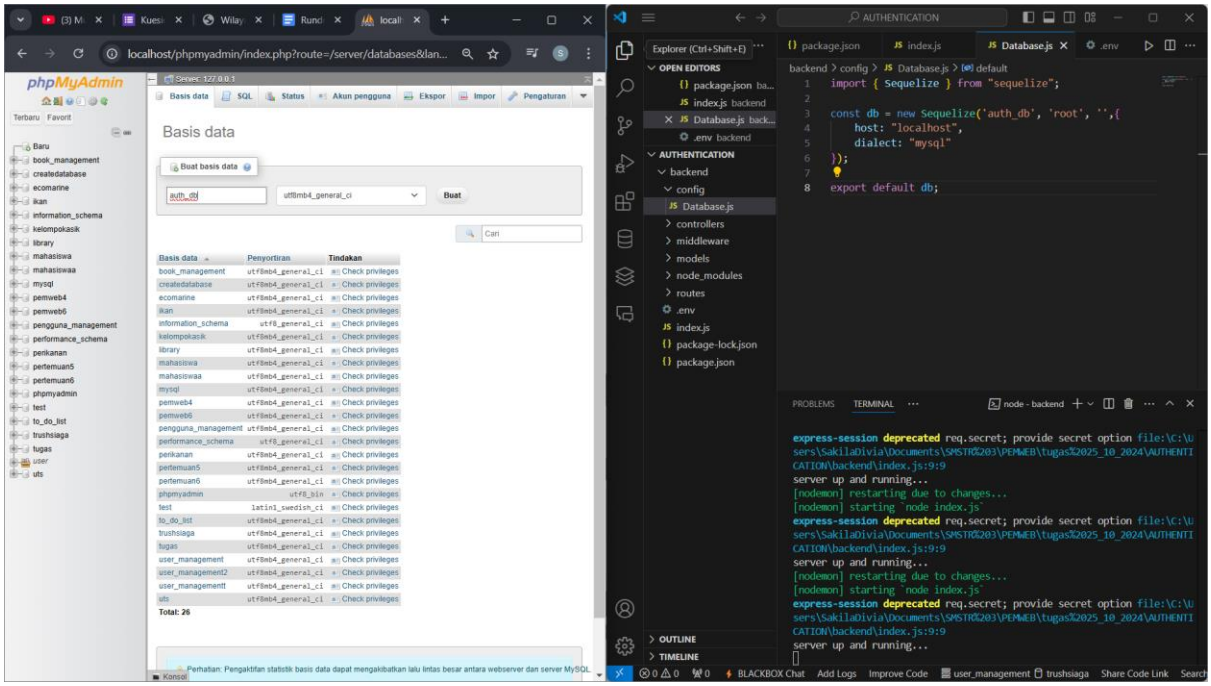
Membuat file dalam configq berupa database.js



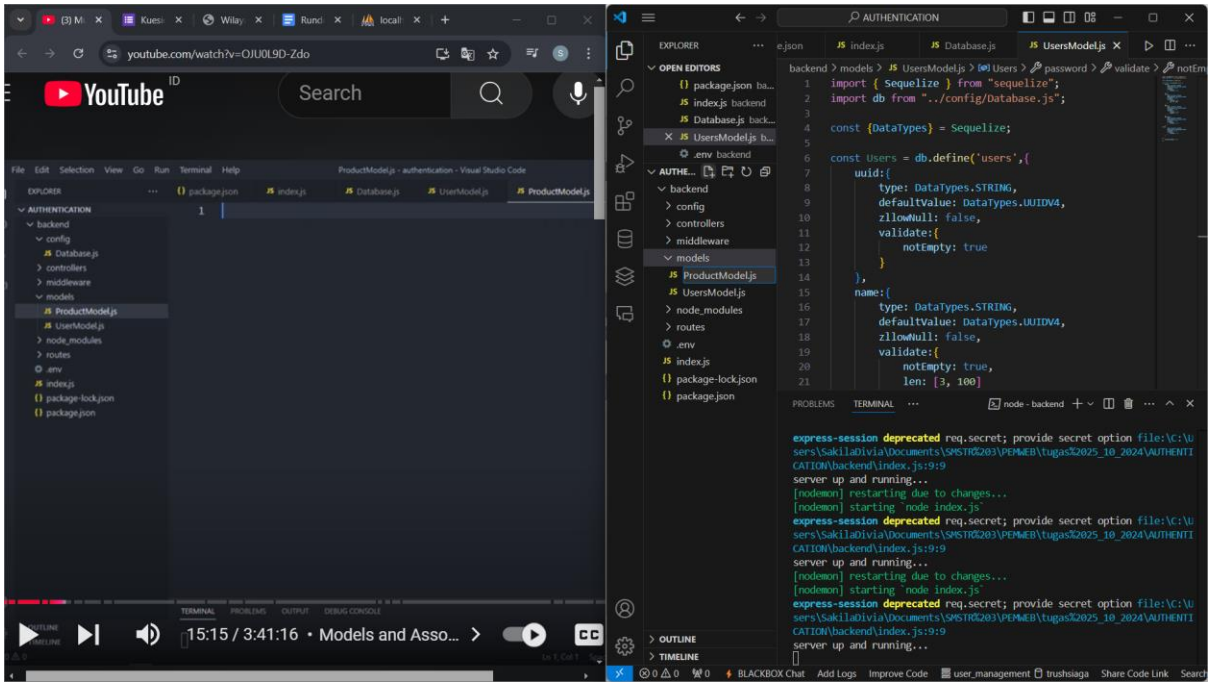
Setelah itu ke localhost pastikan mysql dan apace berjalan dengan baik



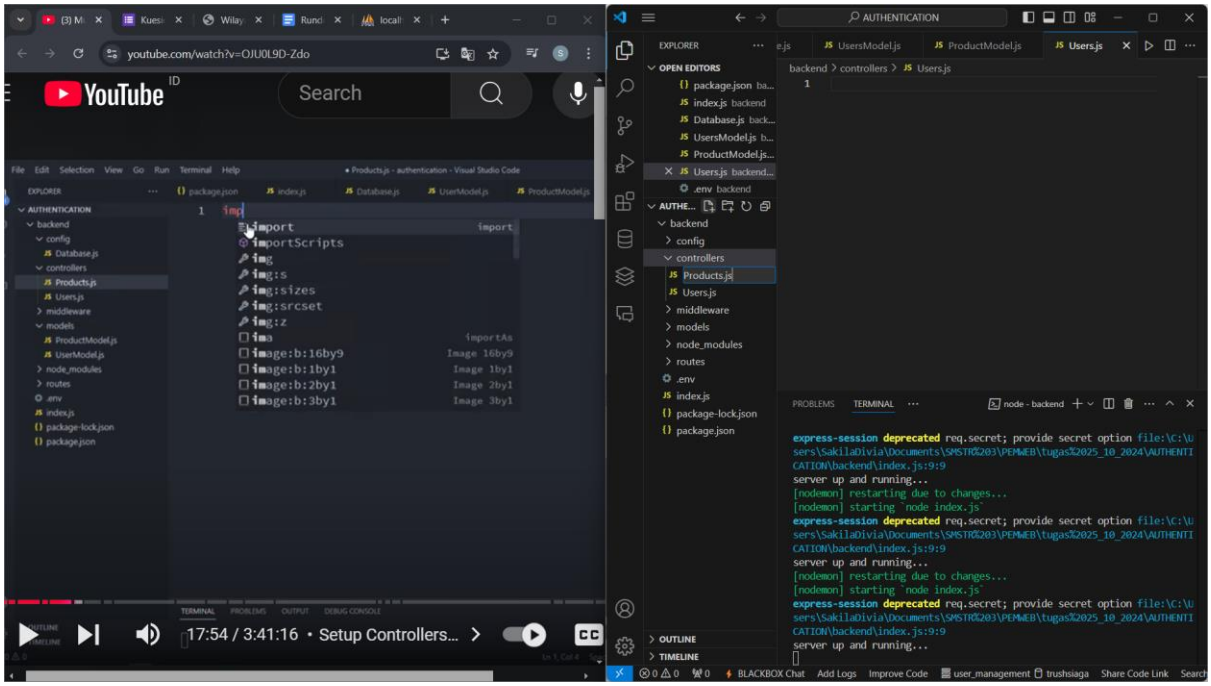
Membuat database



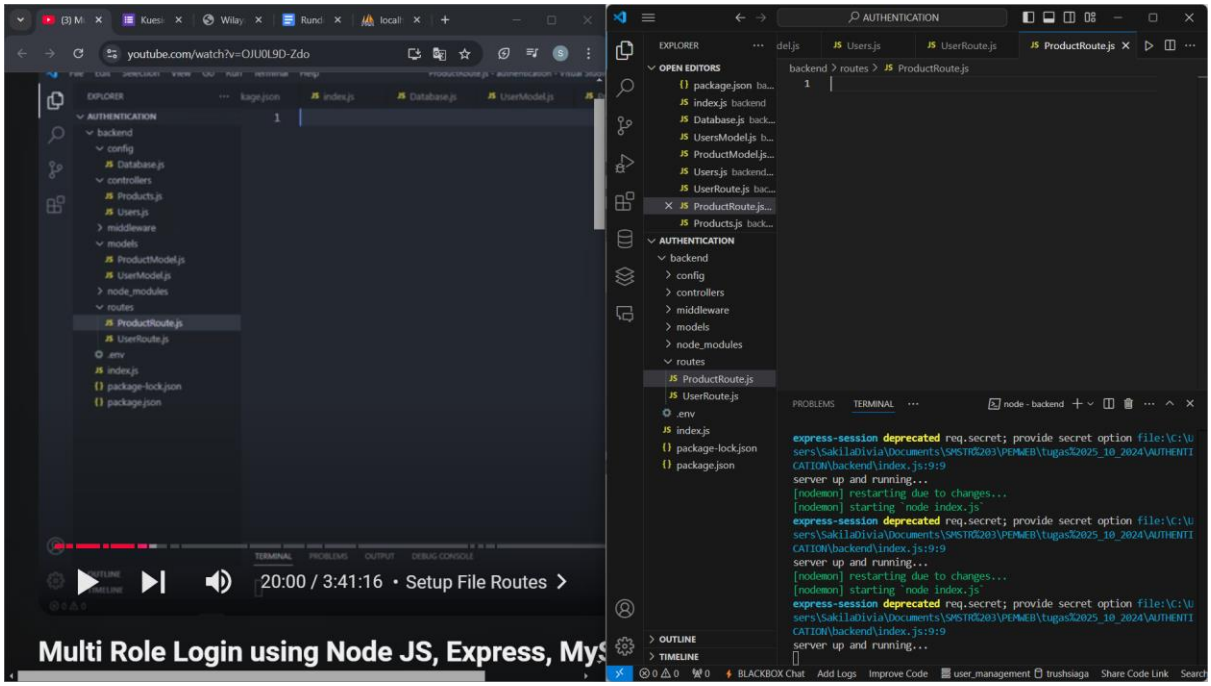
Membuat file dalam model “ProductModel.js



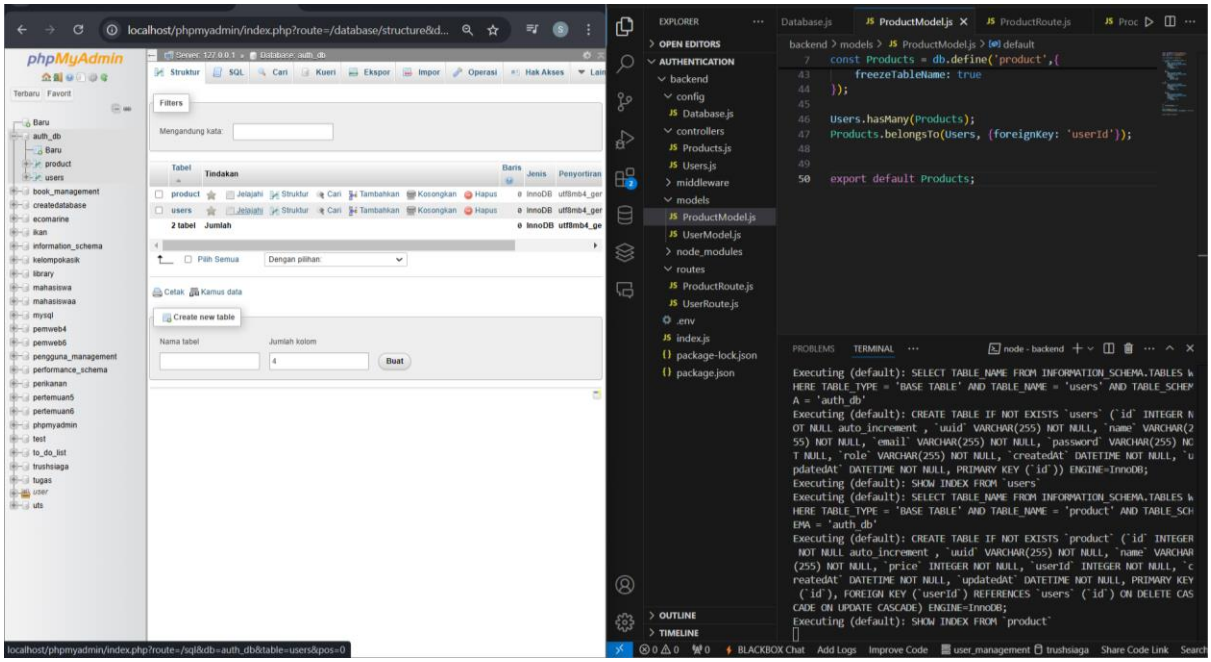
Membuat file users dan products.js dalam controllers



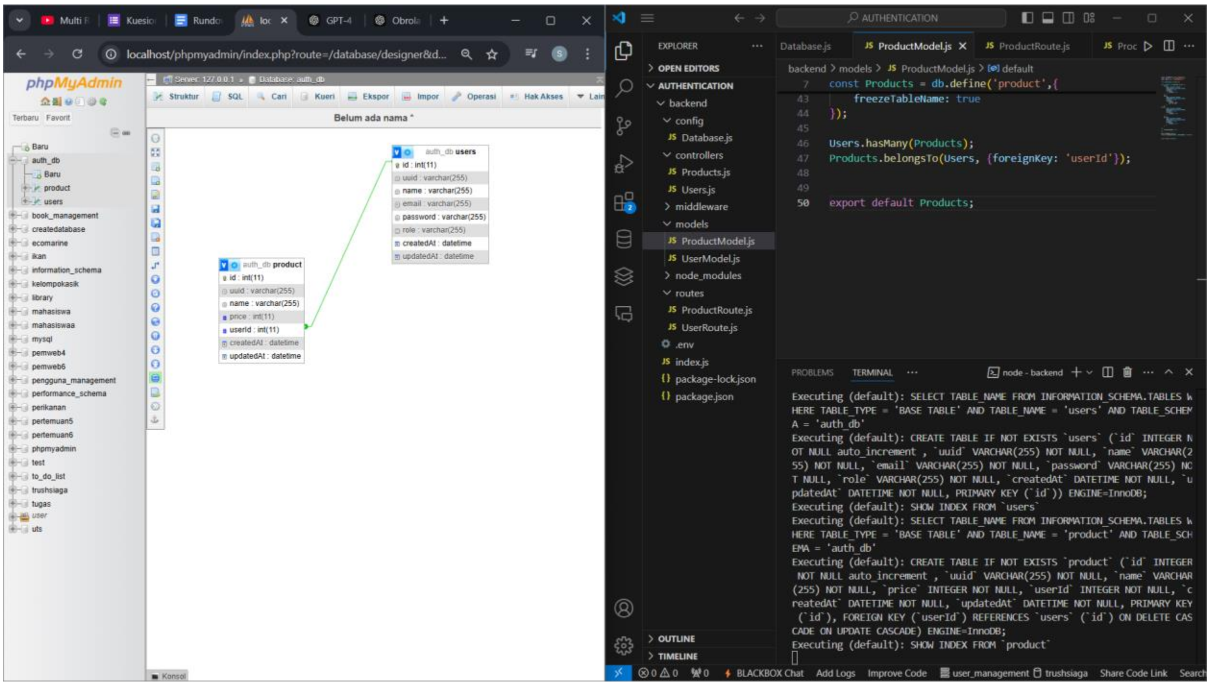
Membuat file dalam routes user dan product.js



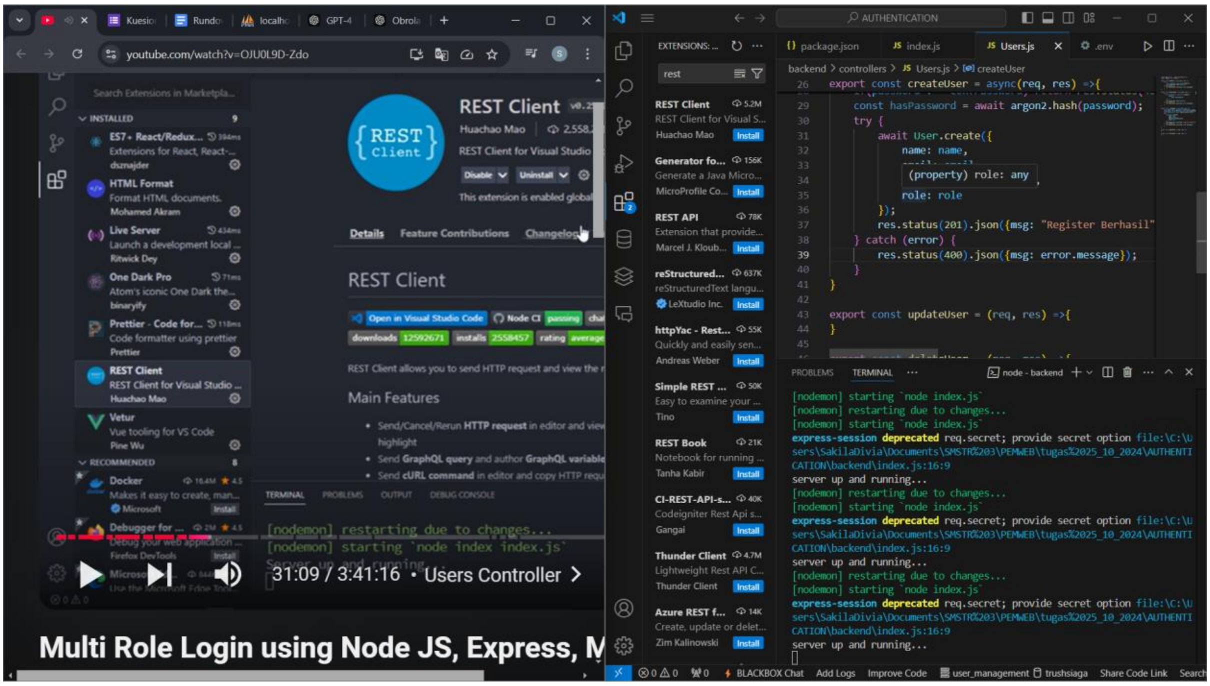
Selanjutnya mengecek apakah tabel sudah ada atau belum di mysql



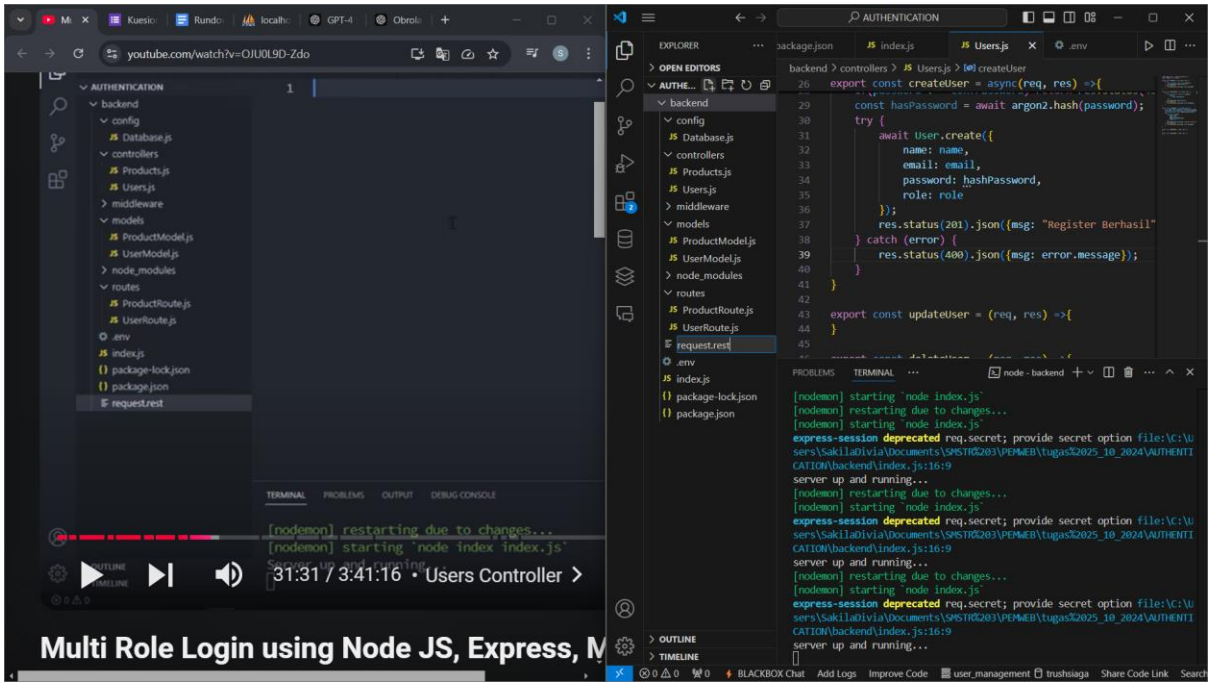
Tampilan jika klik more>desain



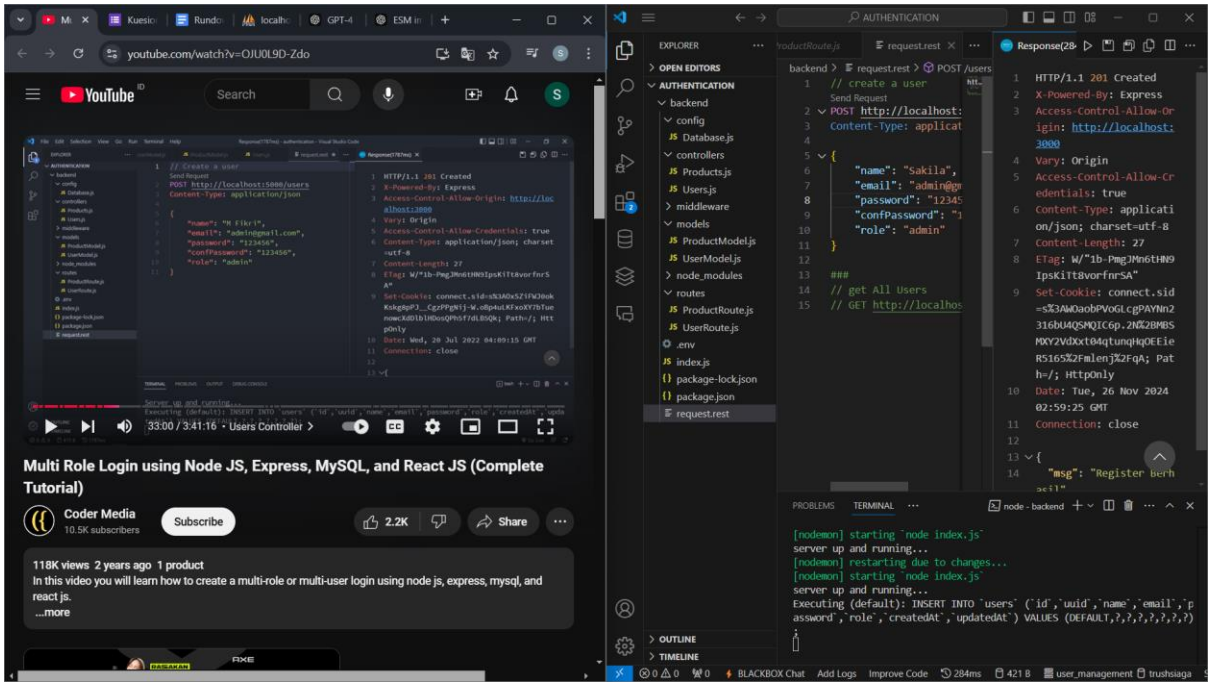
Melakukan install rest client atau bisa juga menggunakan post man untuk melakukan pengujian



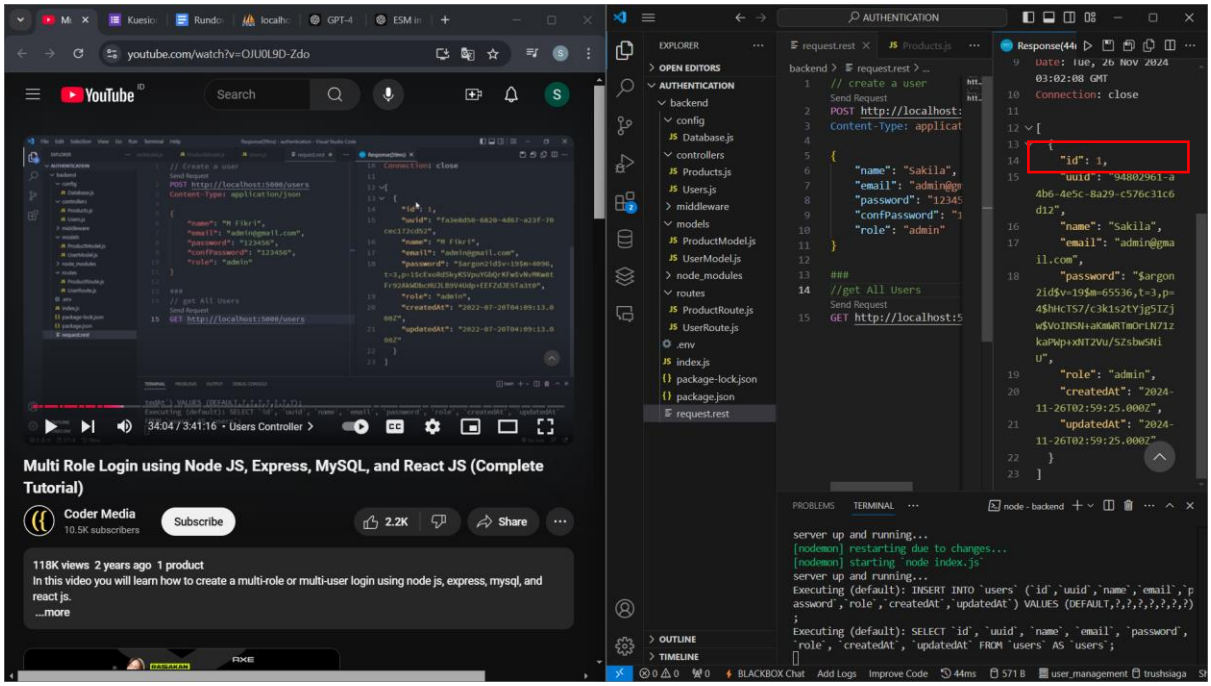
Membuat file baru untuk pengujian



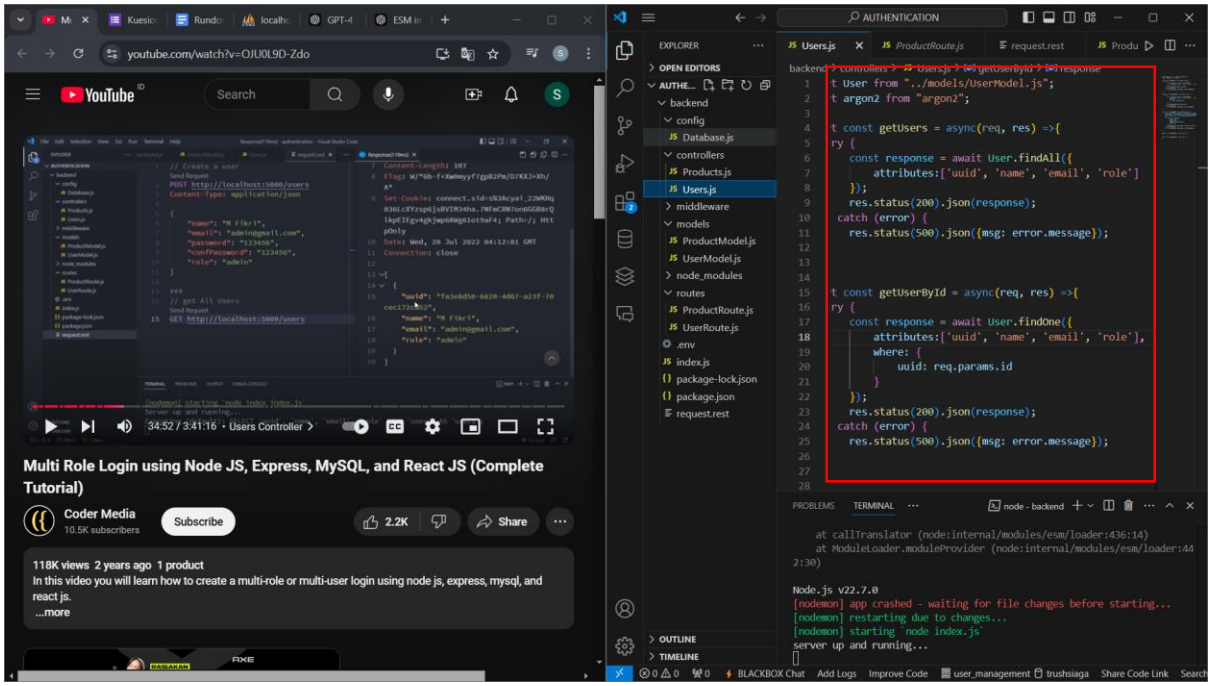
Selanjutnya melakukan penggunaan terhadap rest client dengan klink send request



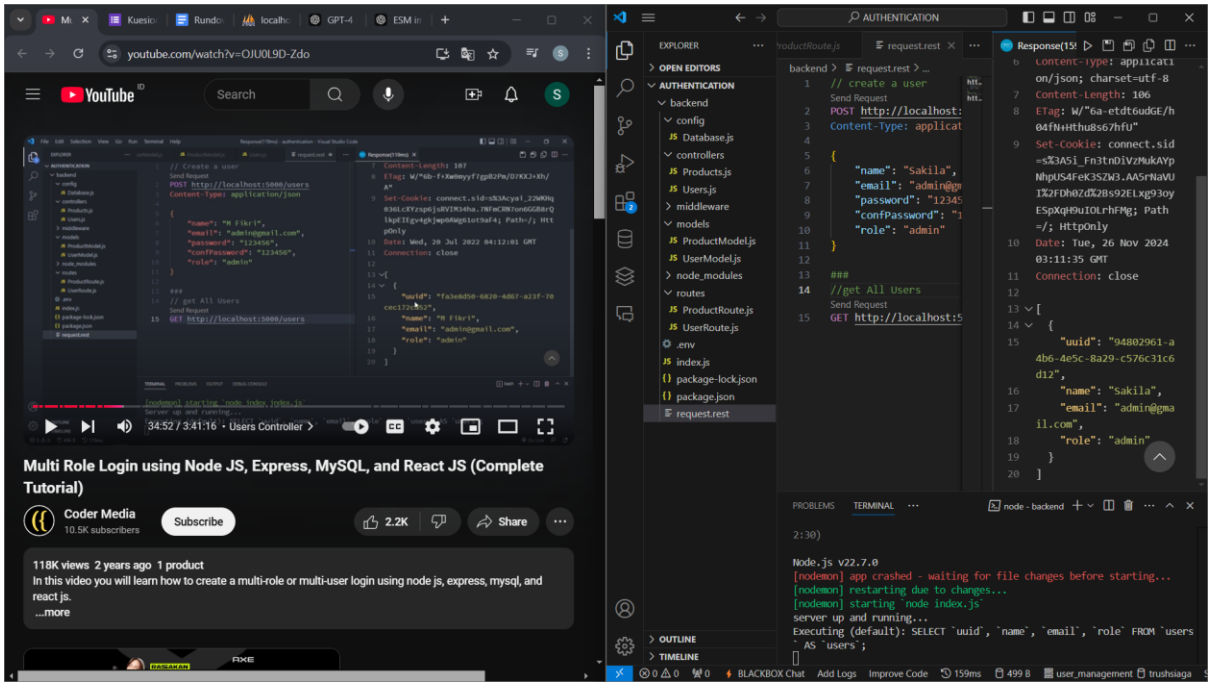
Langkah selanjutnya menghilangkan id pada gambar di bawah ini



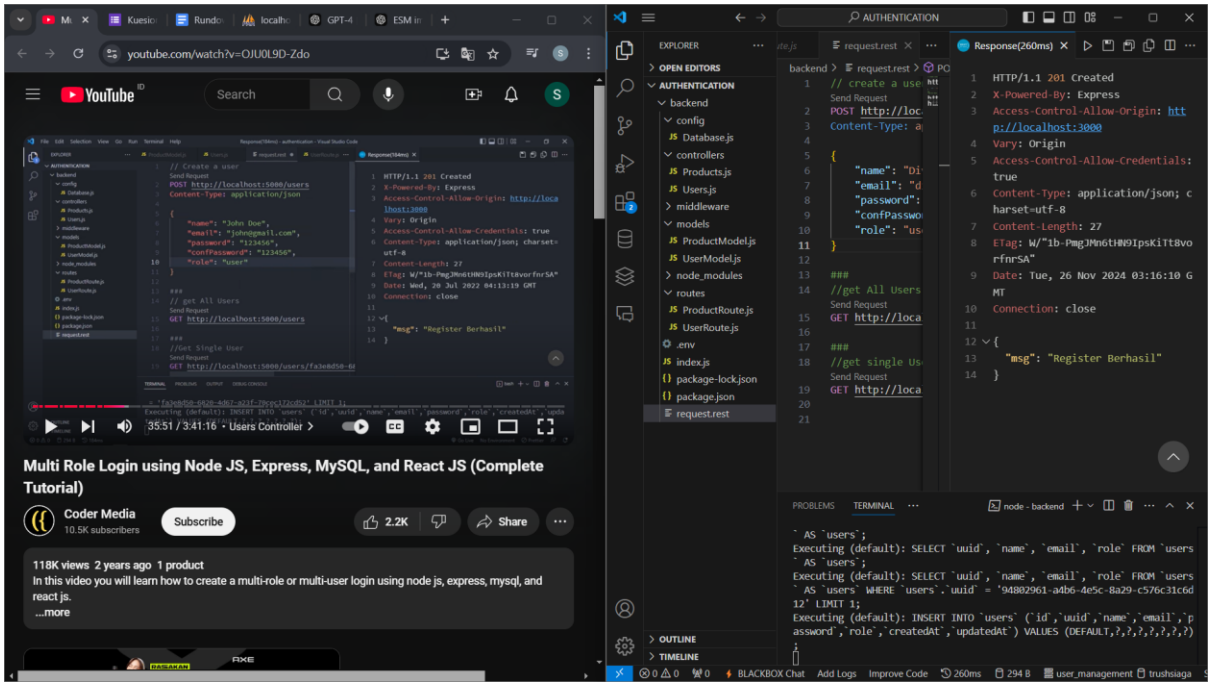
Lakukan penambahan pada users controller dengan menambakna atribut pada getusers dan getuserbyid



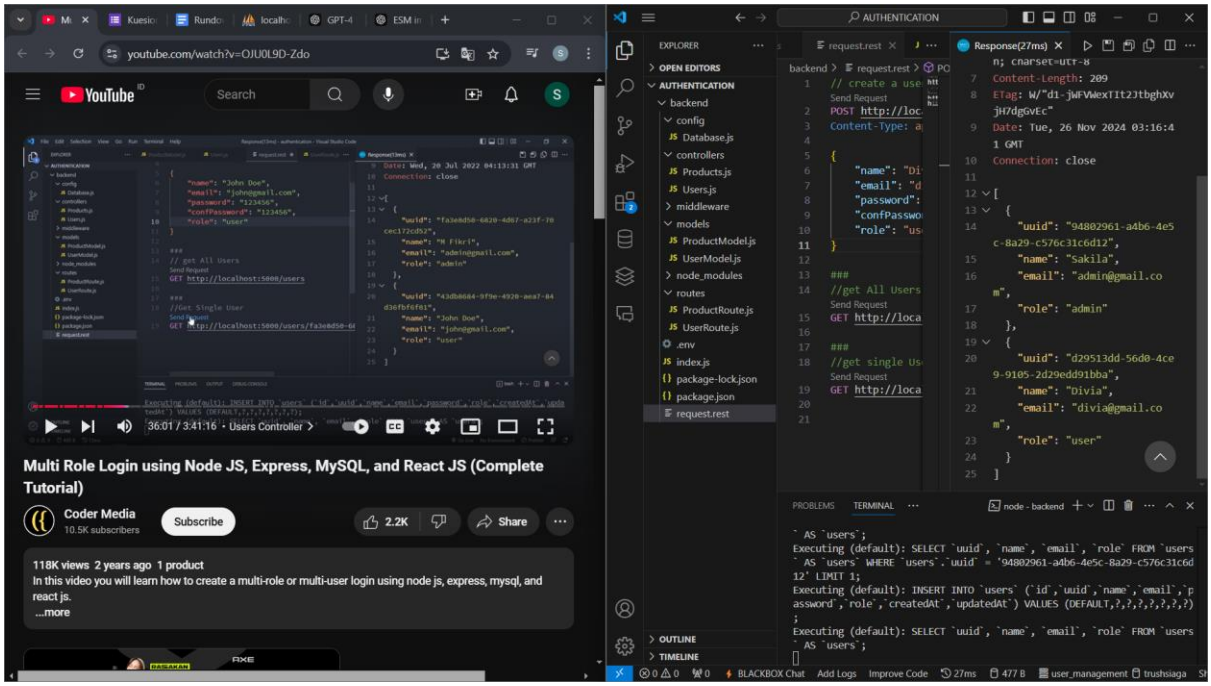
Selanjutnya kembali ke file request rest dan send request. Maka id sudah tidak ada



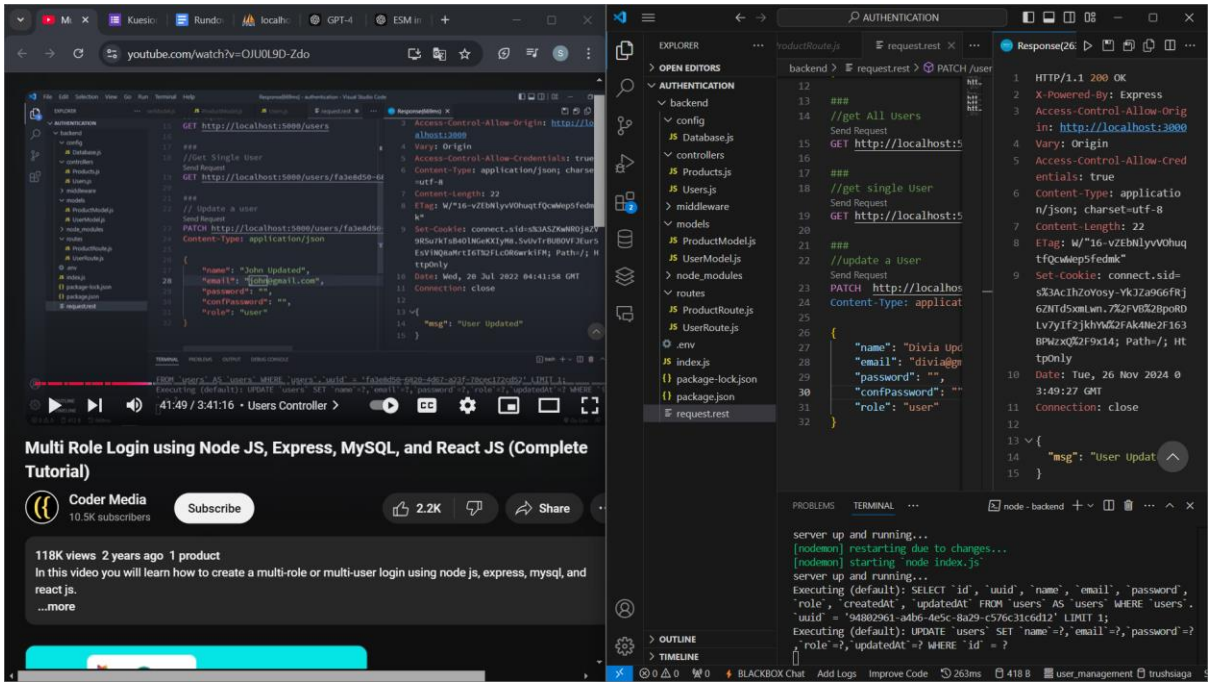
Membuat 1 data lagi yaitu untuk user, lalu send request



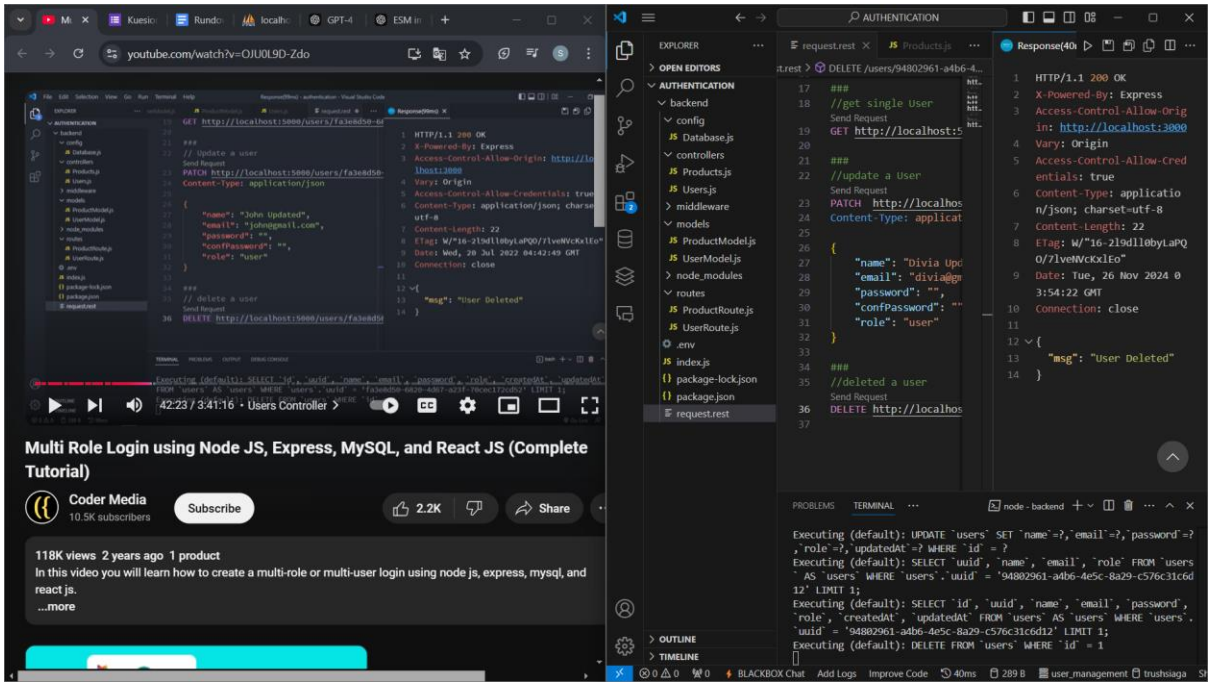
Ketika klik send request yang al users maka dapat terlihat kita memiliki 2 data yaitu user dan admin



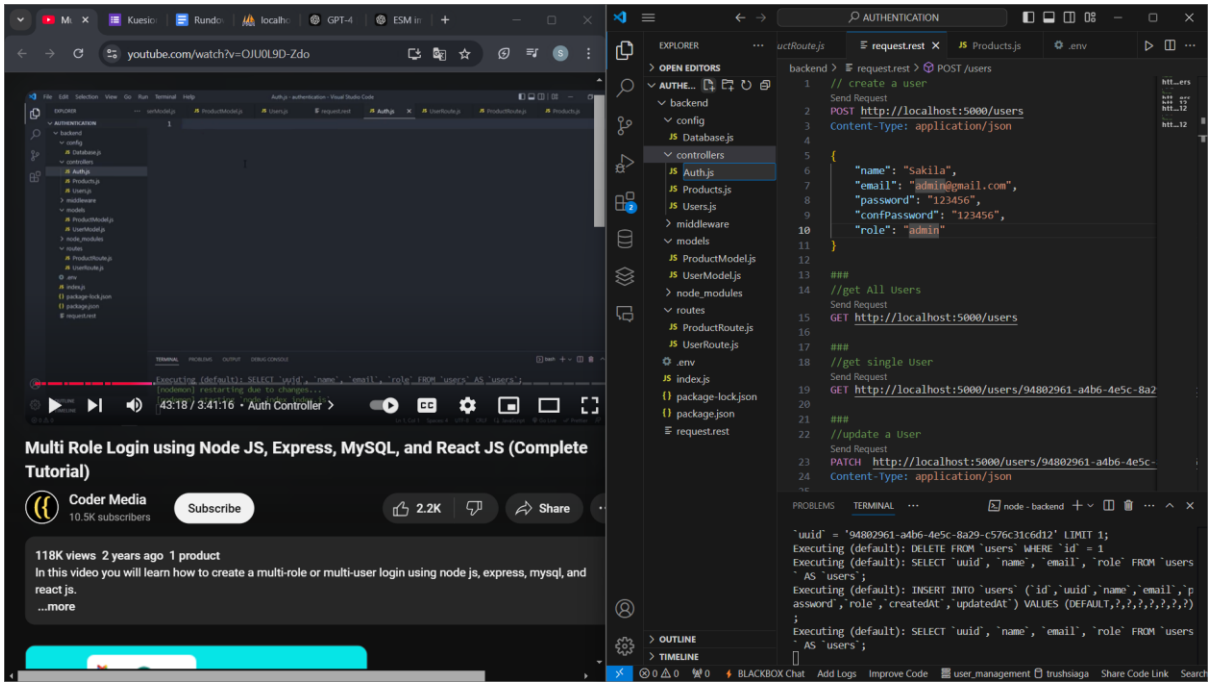
Selanjutnya merupakan melakukan user updated

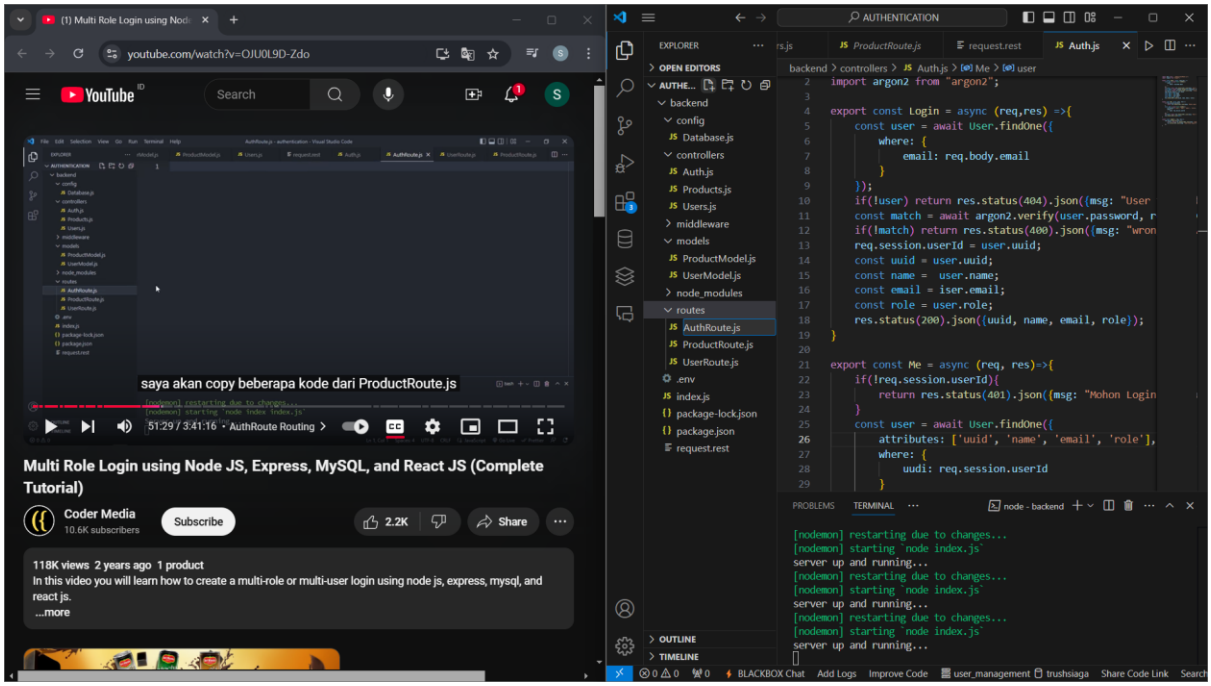


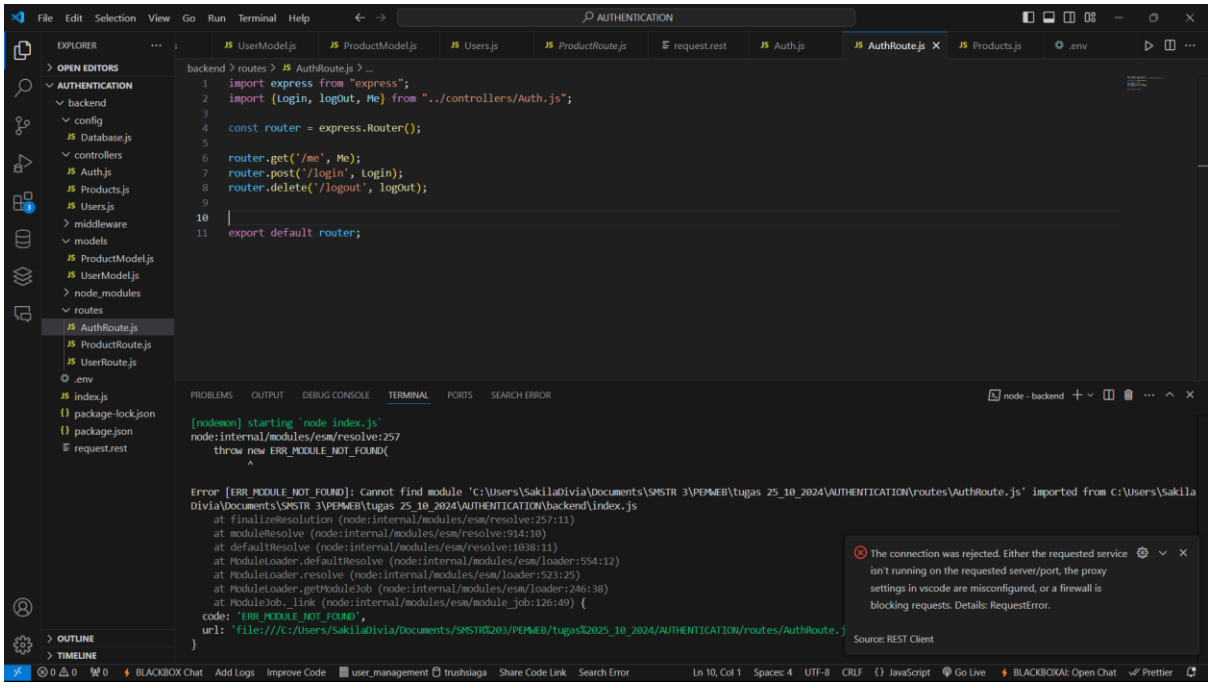
Selanjutnya merupakan melakukan user deleted

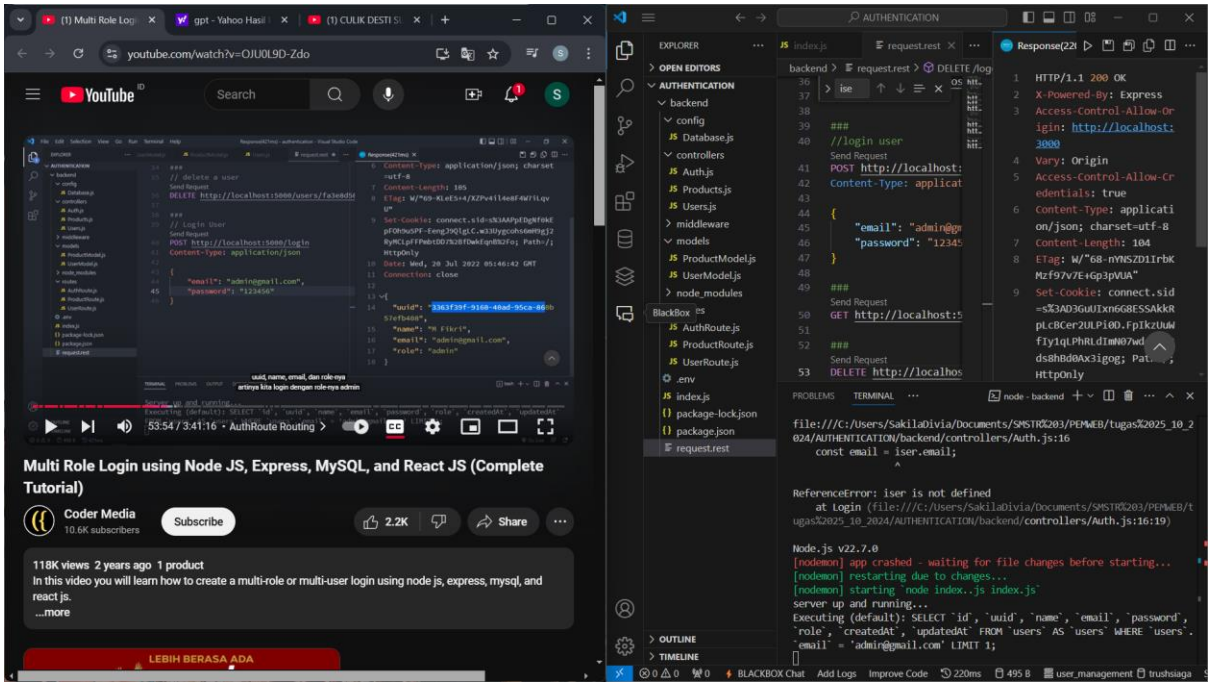


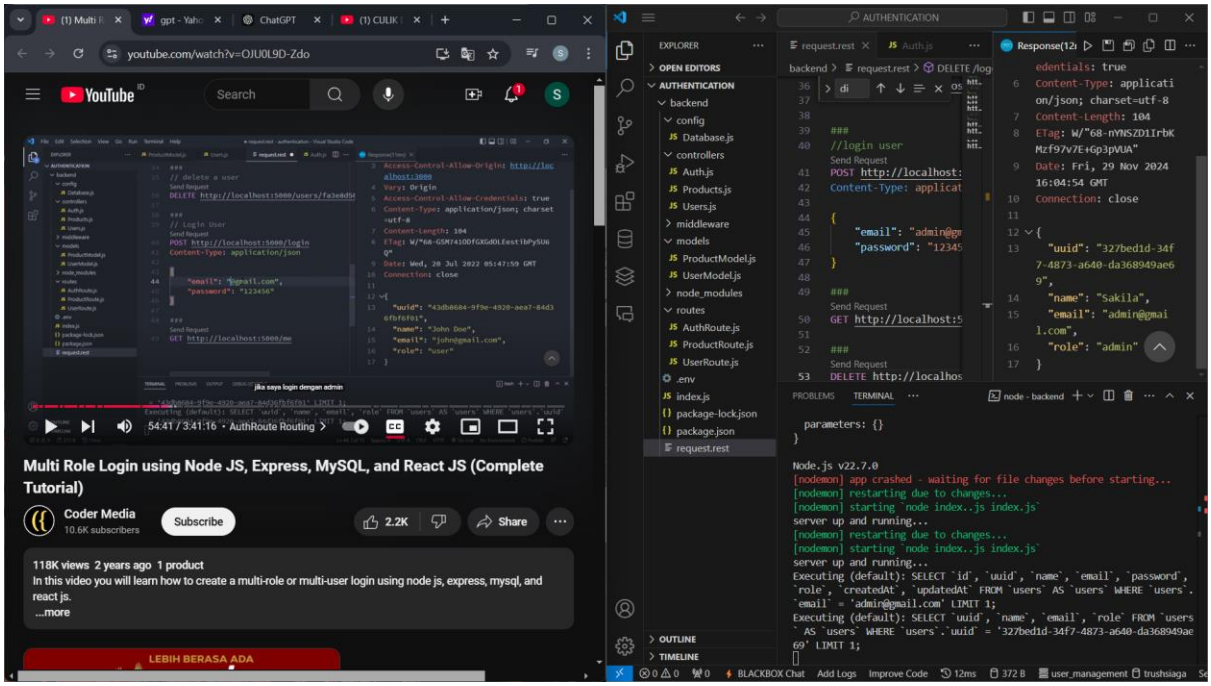
Membuat file bernama auth.js di dalam controllers

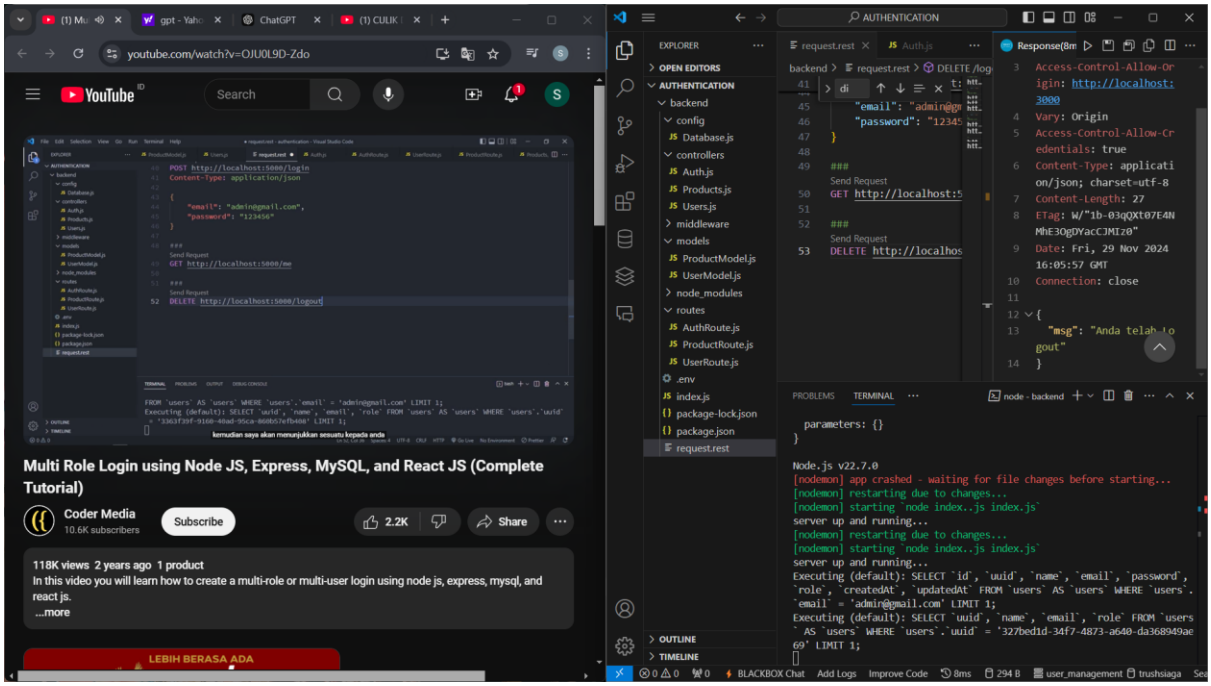



youtube.com/watch?v=OJU0L9D-Zdo
saya akan copy beberapa kode dari ProductRoute.js
Multi Role Login using Node JS, Express, MySQL, and React JS (Complete Tutorial)
Coder Media
10.6K subscribers
Subscribe
118K views 2 years ago 1 product
In this video you will learn how to create a multi-role or multi-user login using node js, express, mysql, and react js.
...more
AUTHENTICATION
backend > controllers > Auth.js > Me > user
import argon2 from "argon2";
export const Login = async (req,res) =>{
const user = await User.findOne({
where: {
email: req.body.email
}
});
if(!user) return res.status(404).json({msg: "User
const match = await argon2.verify(user.password, r
if(!match) return res.status(400).json({msg: "wron
req.session.userId = user.uuid;
const uuid = user.uuid;
const name = user.name;
const email = user.email;
const role = user.role;
res.status(200).json({uuid, name, email, role});
}
export const Me = async (req, res)=>{
if(!req.session.userId){
return res.status(401).json({msg: "Mohon Login
}
const user = await User.findOne({
attributes: ['uuid', 'name', 'email', 'role'],
where: {
uuid: req.session.userId
}
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
server up and running...
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
server up and running...
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
server up and running...

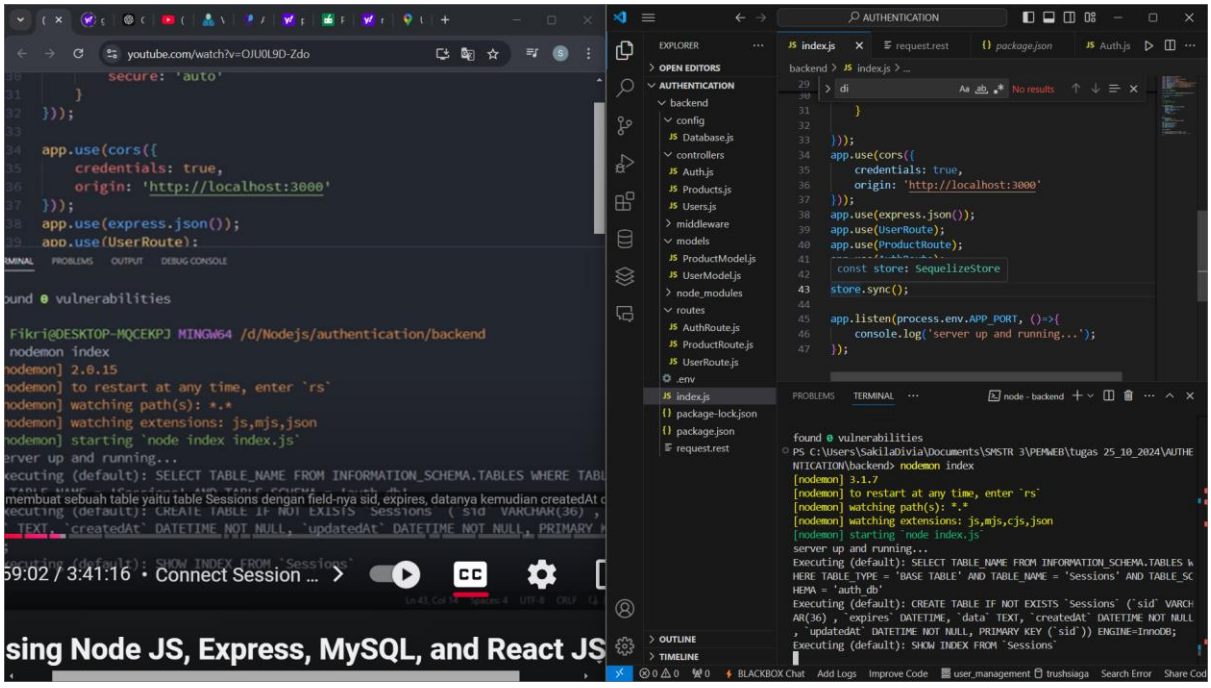

AUTHENTICATION
backend > routes > AuthRoute.js > ...
import express from "express";
import {Login, logOut, Me} from "../controllers/Auth.js";
const router = express.Router();
router.get('/me', Me);
router.post('/login', Login);
router.delete('/logout', logOut);
export default router;
[nodemon] starting `node index.js`
node:internal/modules/esm/resolve:257
throw new ERR_MODULE_NOT_FOUND(
Error [ERR_MODULE_NOT_FOUND]: Cannot find module 'C:\Users\SakilaDivia\Documents\SMSTR 3\PEMWEB\tugas 25_10_2024\AUTHENTICATION\routes\AuthRoute.js' imported from C:\Users\SakilaDivia\Documents\SMSTR 3\PEMWEB\tugas 25_10_2024\AUTHENTICATION\backend\index.js
at finalizeResolution (node:internal/modules/esm/resolve:257:11)
at moduleResolve (node:internal/modules/esm/resolve:914:10)
at defaultResolve (node:internal/modules/esm/resolve:1038:11)
at ModuleLoader.defaultResolve (node:internal/modules/esm/loader:554:12)
at ModuleLoader.resolve (node:internal/modules/esm/loader:523:25)
at ModuleLoader.getModuleJob (node:internal/modules/esm/loader:246:38)
at ModuleJob._link (node:internal/modules/esm/module_job:126:49) {
code: 'ERR_MODULE_NOT_FOUND',
url: 'file:///C:/Users/SakilaDivia/Documents/SMSTR%203/PEMWEB/tugas%2025_10_2024/AUTHENTICATION/routes/AuthRoute.j
}
The connection was rejected. Either the requested service isn't running on the requested server/port, the proxy settings in vscode are misconfigured, or a firewall is blocking requests. Details: RequestError.
Source: REST Client

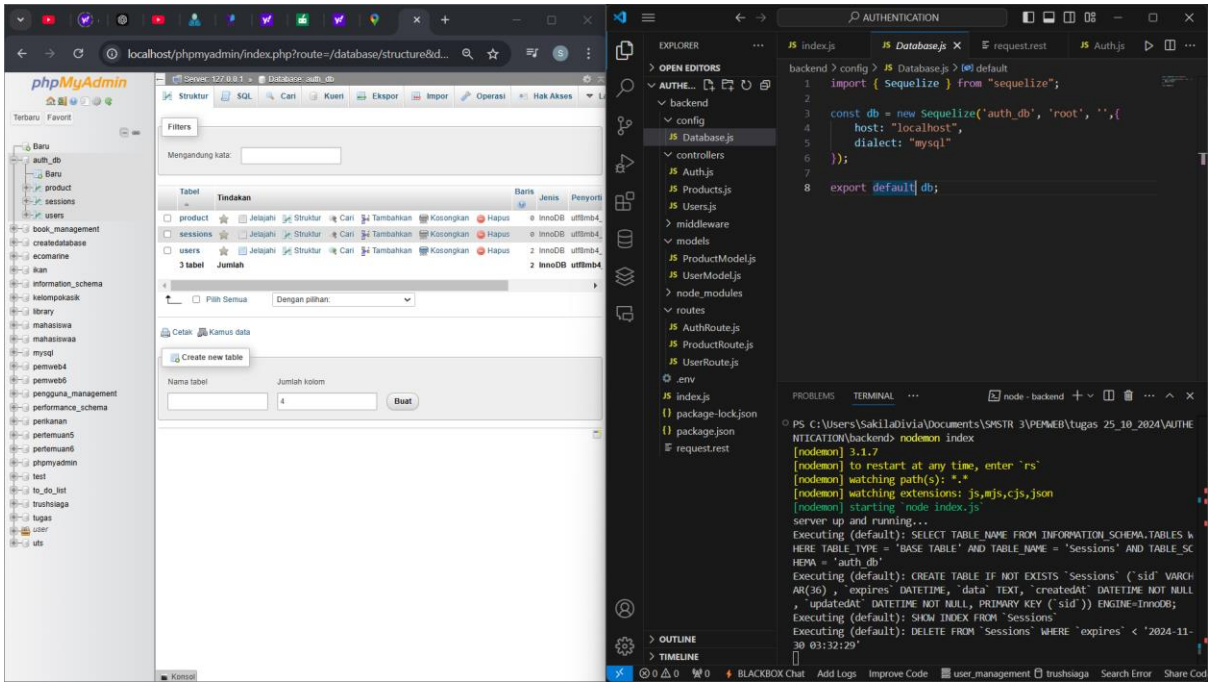

youtube.com/watch?v=OJU0L9D-Zdo
Search
53:54 / 3:41:16 • AuthRoute Routing
Multi Role Login using Node JS, Express, MySQL, and React JS (Complete Tutorial)
Coder Media
10.6K subscribers
Subscribe
2.2K
Share
118K views 2 years ago 1 product
In this video you will learn how to create a multi-role or multi-user login using node js, express, mysql, and react js.
...more
LEBIH BERASA ADA
AUTHENTICATION
EXPLORER
OPEN EDITORS
backend
config
Database.js
controllers
Auth.js
Products.js
Users.js
middleware
models
ProductModel.js
UserModel.js
node_modules
routes
AuthRoute.js
ProductRoute.js
UserRoute.js
.env
index.js
package-lock.json
package.json
request.rest
OUTLINE
TIMELINE
PROBLEMS
TERMINAL
node - backend
file:///C:/Users/SakilaDivia/Documents/SMSTR%203/PEMWEB/tugas%2025_10_2024/AUTHENTICATION/backend/controllers/Auth.js:16
const email = iser.email;
ReferenceError: iser is not defined
at Login (file:///C:/Users/SakilaDivia/Documents/SMSTR%203/PEMWEB/tugas%2025_10_2024/AUTHENTICATION/backend/controllers/Auth.js:16:19)
Node.js v22.7.0
[nodemon] app crashed - waiting for file changes before starting...
[nodemon] restarting due to changes...
[nodemon] starting `node index.js index.js`
server up and running...
Executing (default): SELECT `id`, `uuid`, `name`, `email`, `password`, `role`, `createdAt`, `updatedAt` FROM `users` AS `users` WHERE `users`.`email` = 'admin@gmail.com' LIMIT 1;
BLACKBOX Chat
Add Logs
Improve Code
220ms
495 B
user_management
trushsiaga


youtube.com/watch?v=OJU0L9D-Zdo
Search
54:41 / 3:41:16 • AuthRoute Routing
Multi Role Login using Node JS, Express, MySQL, and React JS (Complete Tutorial)
Coder Media
10.6K subscribers
Subscribe
2.2K
Share
118K views 2 years ago 1 product
In this video you will learn how to create a multi-role or multi-user login using node js, express, mysql, and react js.
...more
LEBIH BERASA ADA
AUTHENTICATION
EXPLORER
OPEN EDITORS
backend
config
Database.js
controllers
Auth.js
Products.js
Users.js
middleware
models
ProductModel.js
UserModel.js
node_modules
routes
AuthRoute.js
ProductRoute.js
UserRoute.js
.env
index.js
package-lock.json
package.json
request.rest
OUTLINE
TIMELINE
PROBLEMS
TERMINAL
node - backend
parameters: {}
Node.js v22.7.0
[nodemon] app crashed - waiting for file changes before starting...
[nodemon] restarting due to changes...
[nodemon] starting `node index.js index.js`
server up and running...
[nodemon] restarting due to changes...
[nodemon] starting `node index.js index.js`
server up and running...
Executing (default): SELECT `id`, `uuid`, `name`, `email`, `password`, `role`, `createdAt`, `updatedAt` FROM `users` AS `users` WHERE `users`.`email` = 'admin@gmail.com' LIMIT 1;
Executing (default): SELECT `uuid`, `name`, `email`, `role` FROM `users` AS `users` WHERE `users`.`uuid` = '327bed1d-34f7-4873-a640-da368949ae69' LIMIT 1;
BLACKBOX Chat
Add Logs
Improve Code
12ms
372 B
user_management
trushsiaga

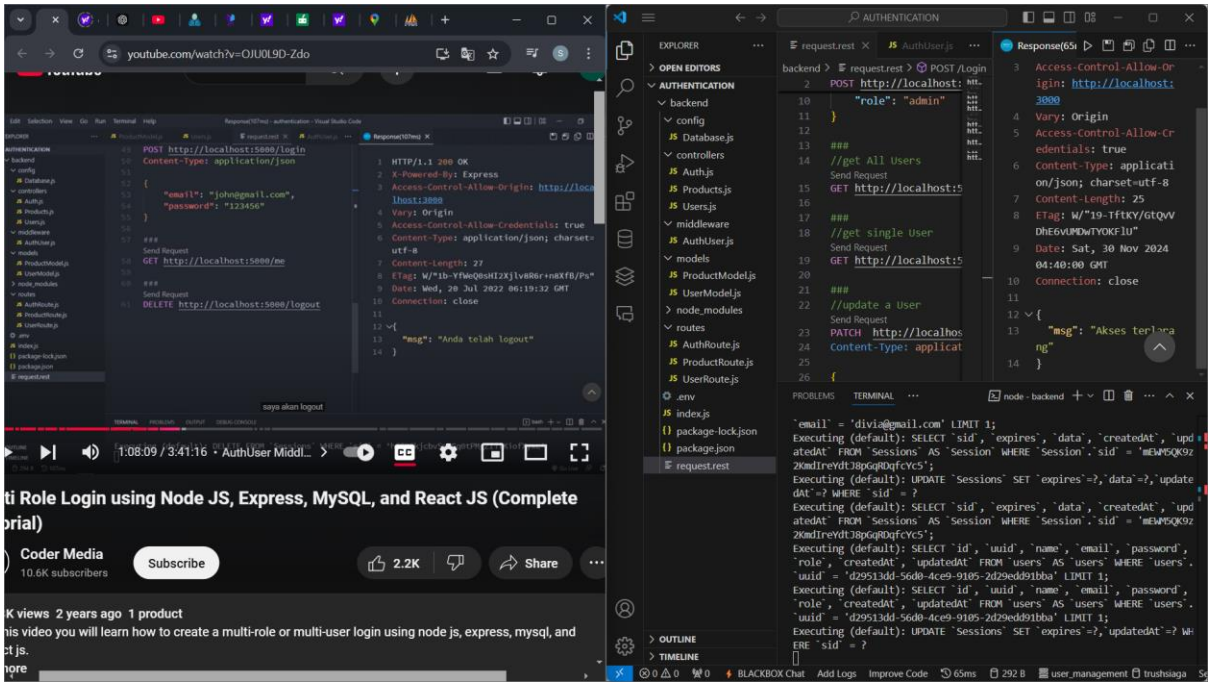


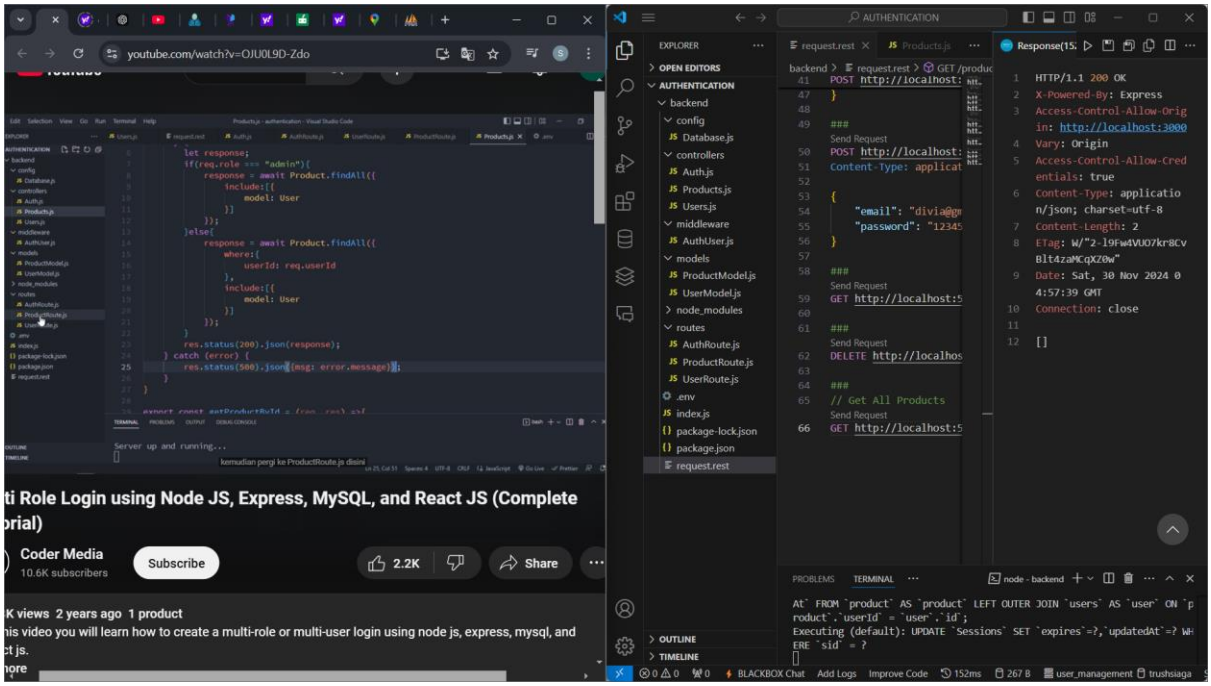
Setiap kali server di reastart maka kita akan di arahkan untuk login kembali, maka dari itu kita melakukan penyimpanan di database agar tidak login" terlebih dahulu yaitu dengan menginstal "npm install connect-session-sequelize". Selanjutnya jalankan kembali terminalnya



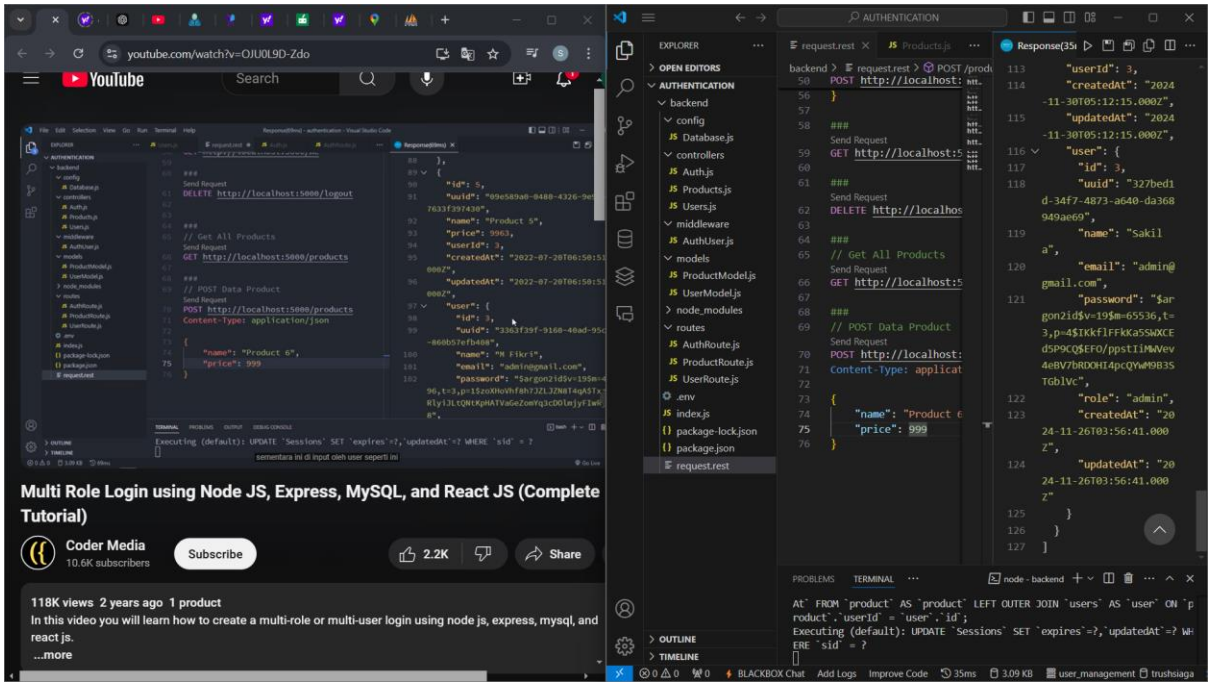


Membuat akses admin only, ketika klik get all user hnya admin yang bisa akses

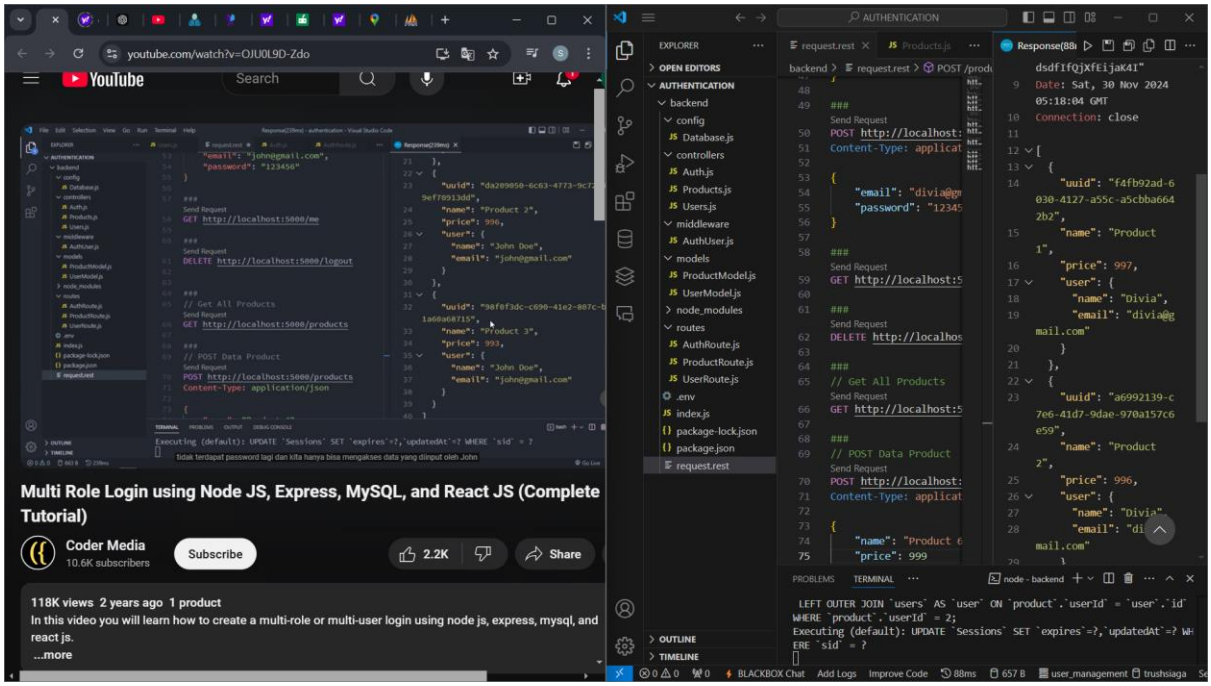




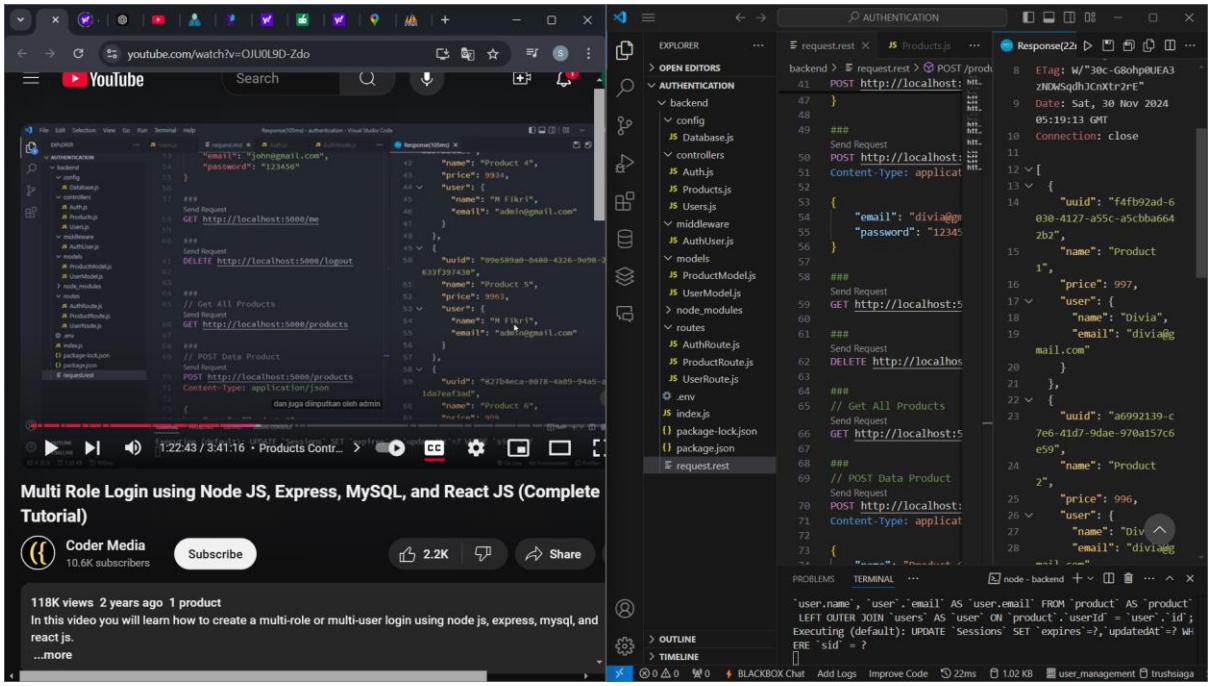
Membuat penginputan product oleh user dan admin

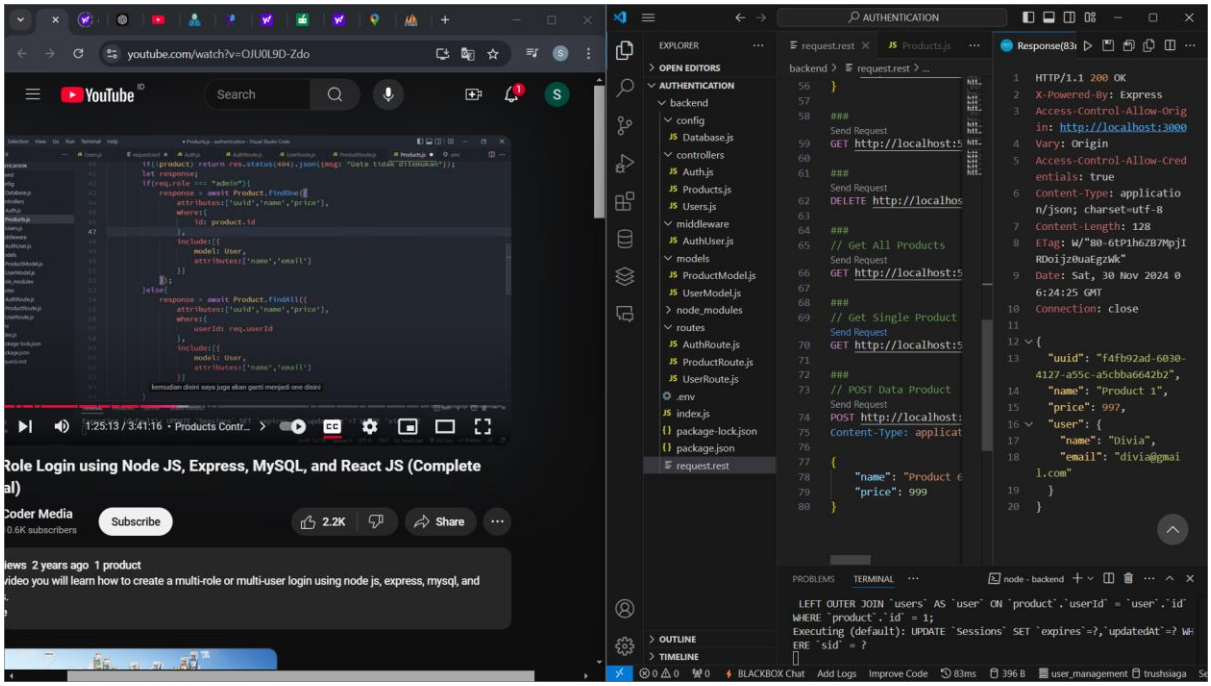


Menghilangkan data yang kita tidak inginkan

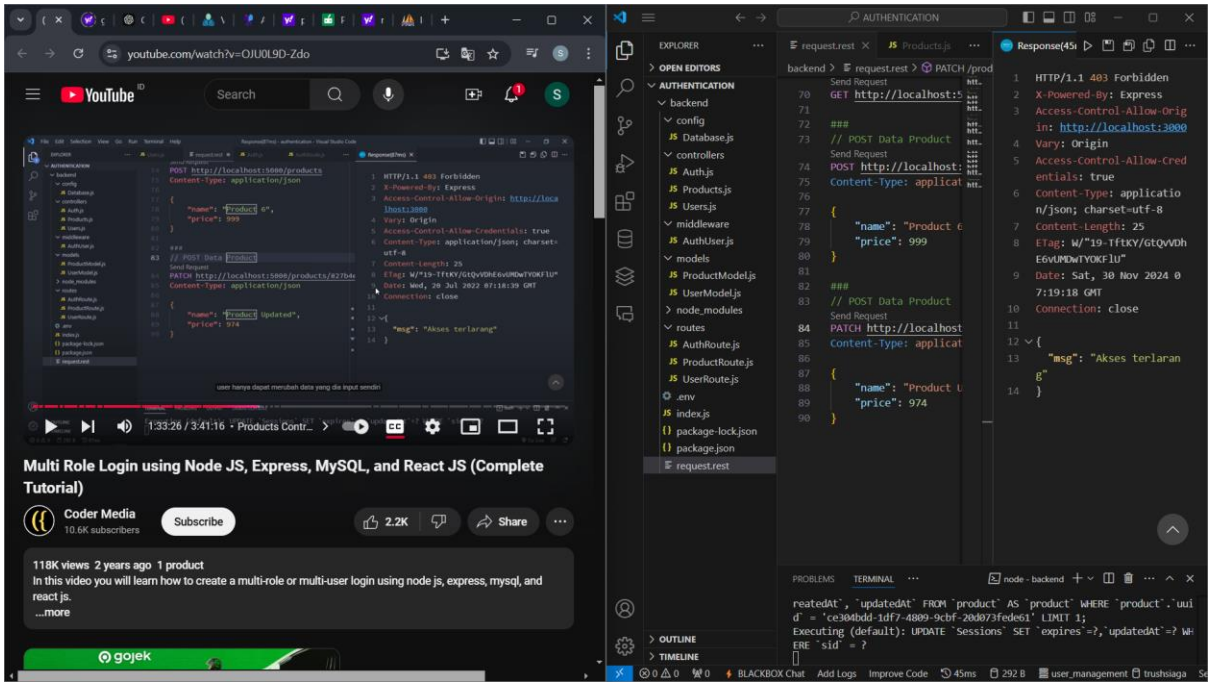


Jika login sebagai admin sekarang sudah bisa melihat data yang di inputkan oleh user

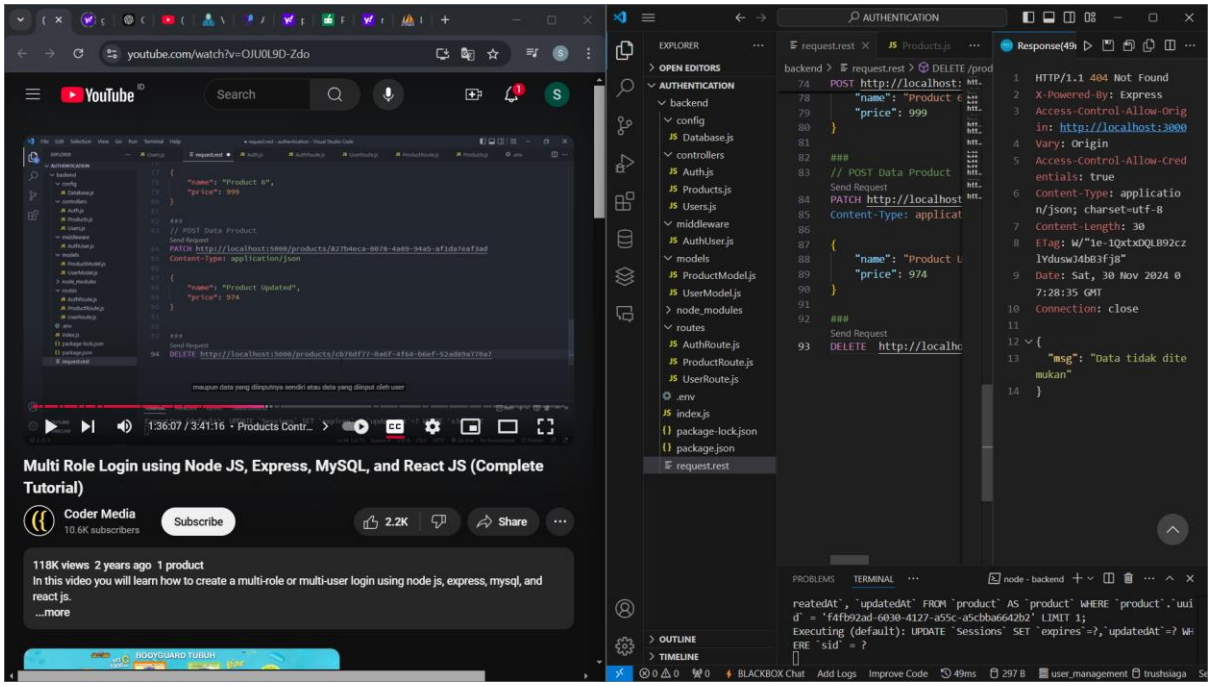




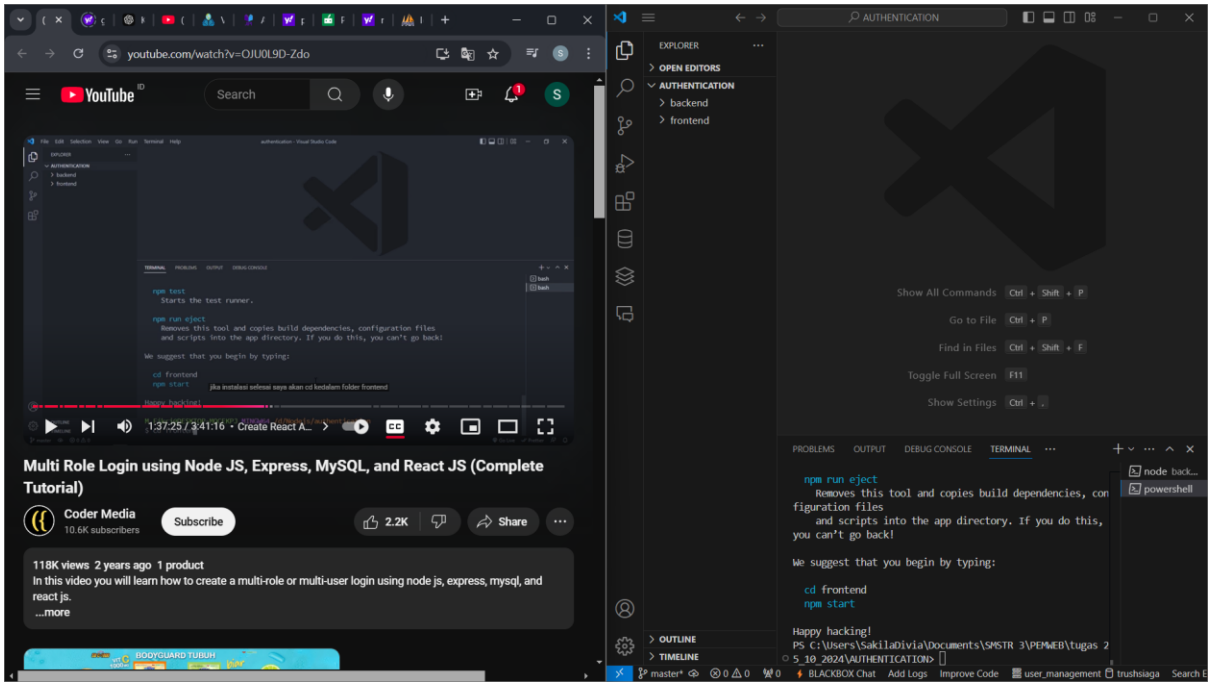
Memberikan pesan akses terlarang karena user tidak dapat menginput id yang di inputkan oleh admin



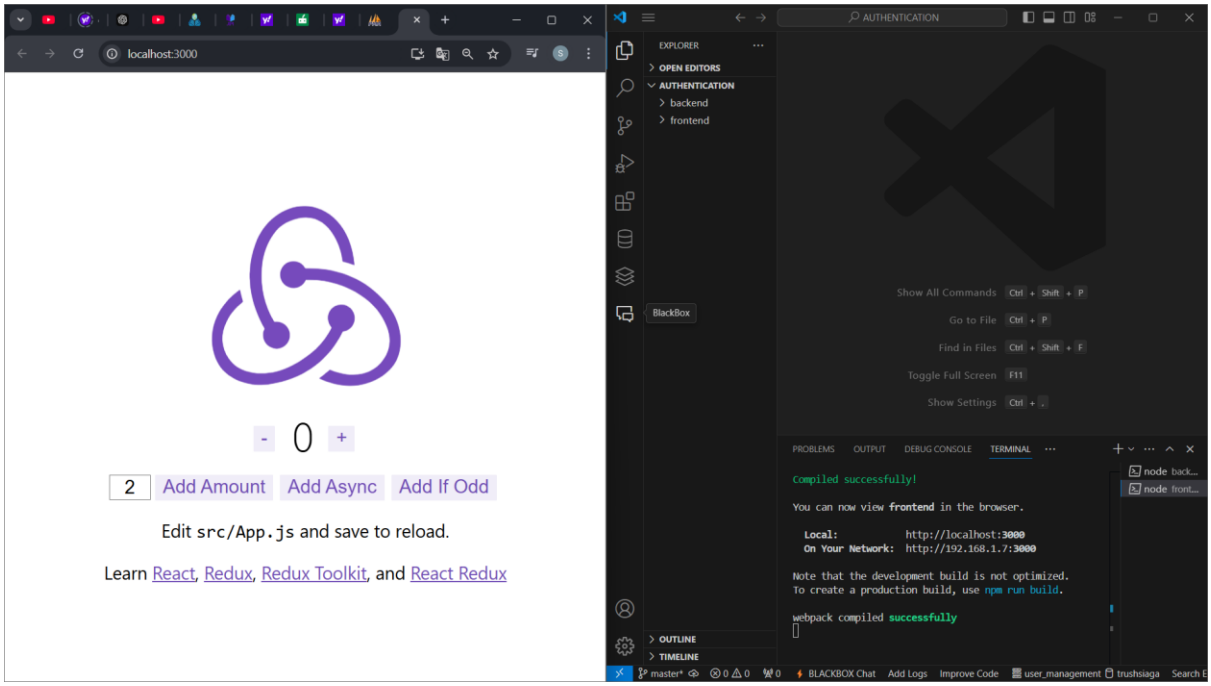
User dapat menghapus data yang sudah di inputkan oleh user sendiri tetapi admin bisa menghapus semua data yang di inputkan oleh user maupun admin

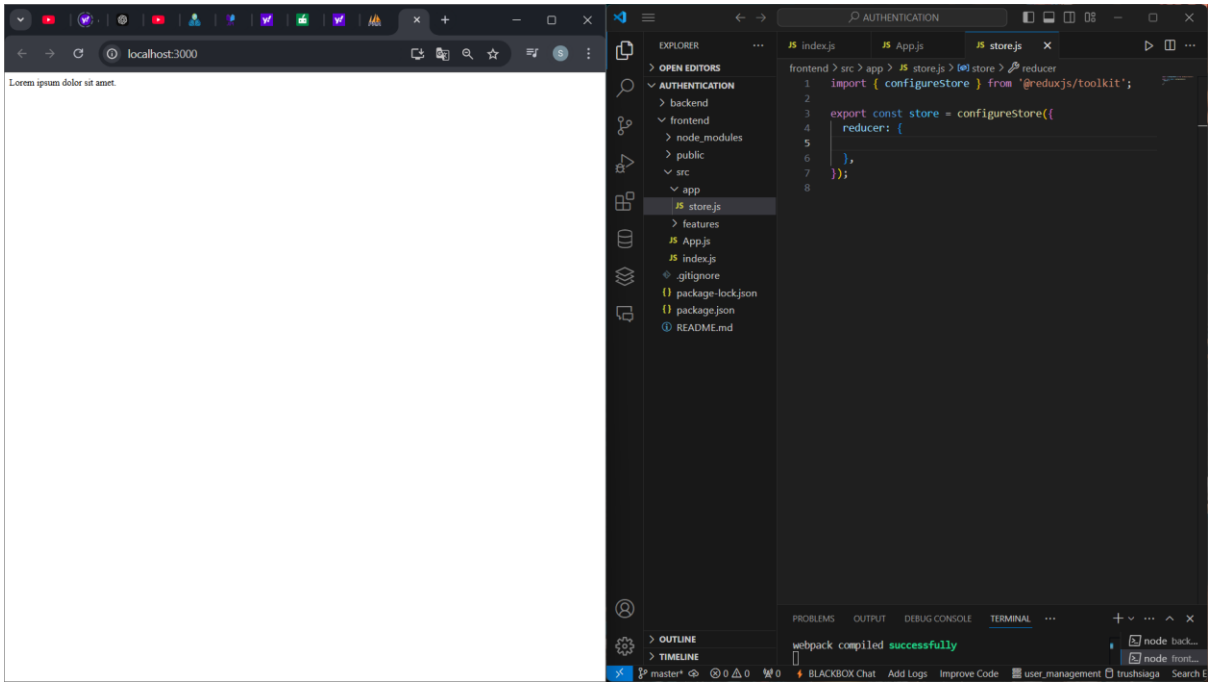


Melakukan intalisasi frontend

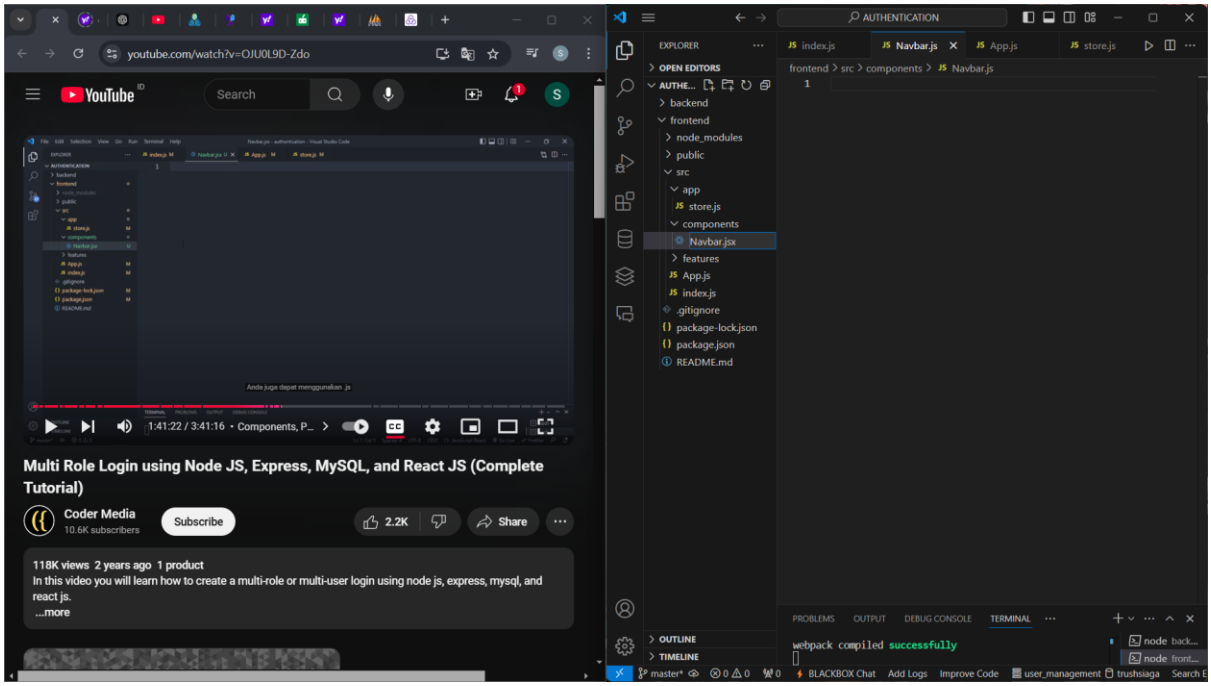


Setelah melakukan instalisasi yang di butuhkan

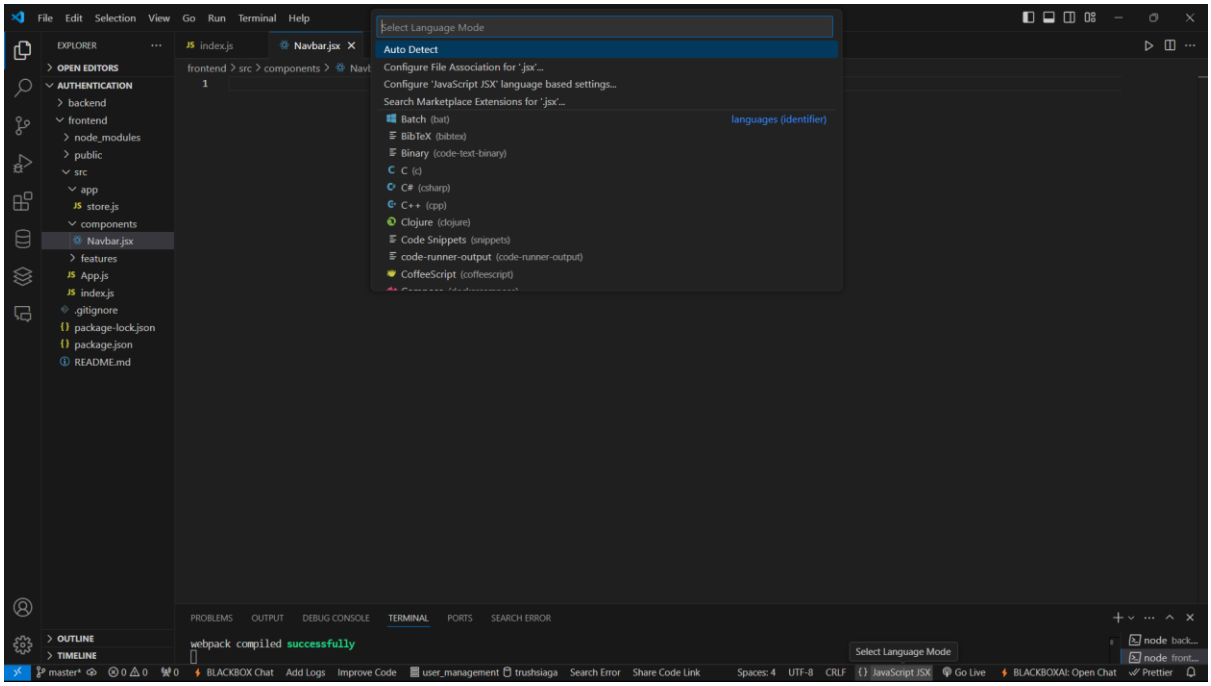




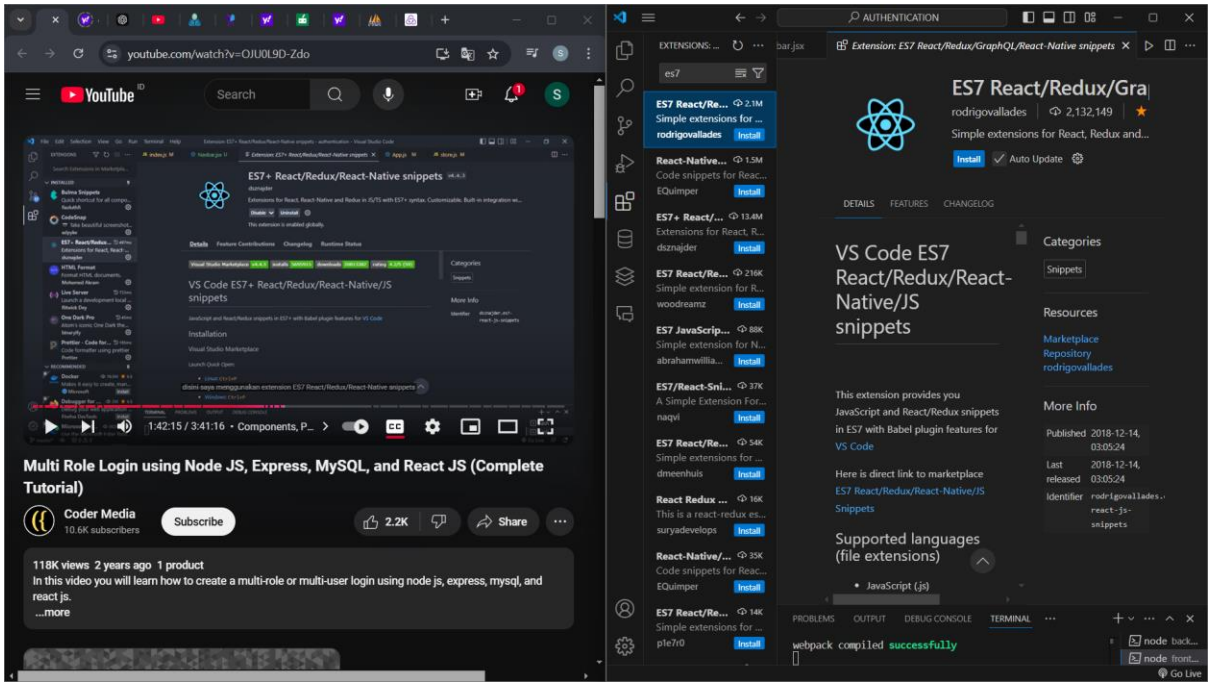
Membuat file navbar.jsx dalam folder components yang ada dalam src

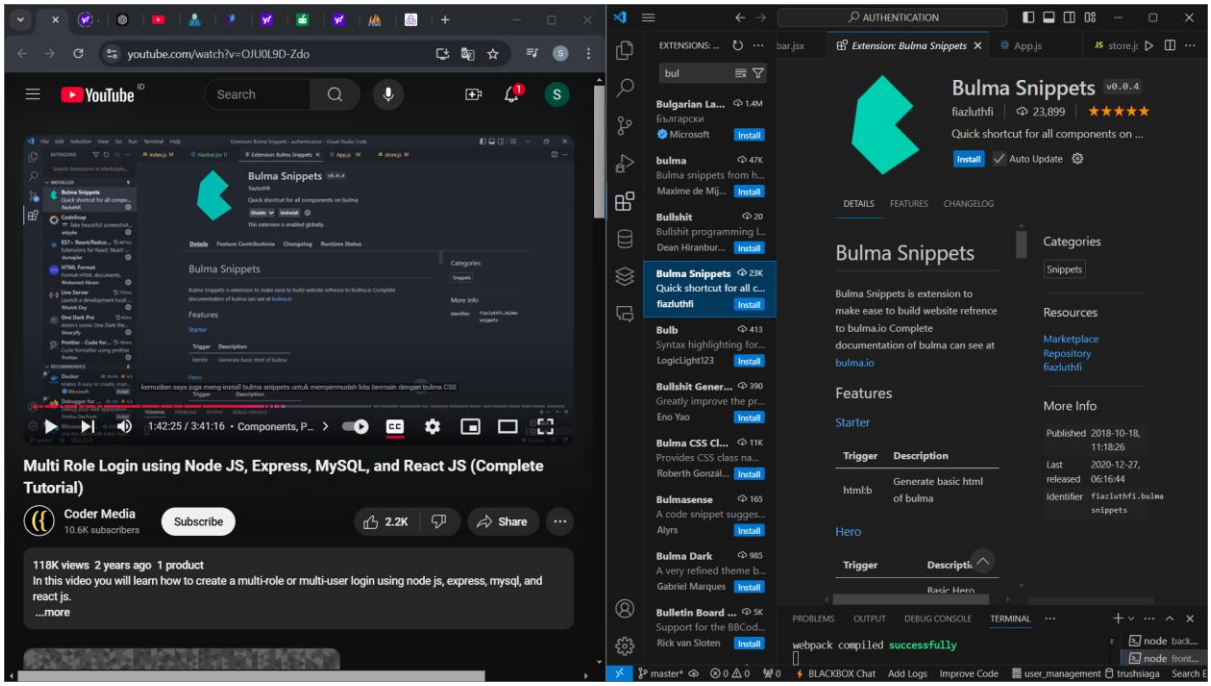


Agar snippetnya bekerja dengan baik maka kita melakukan pengaturan terlebih dahulu

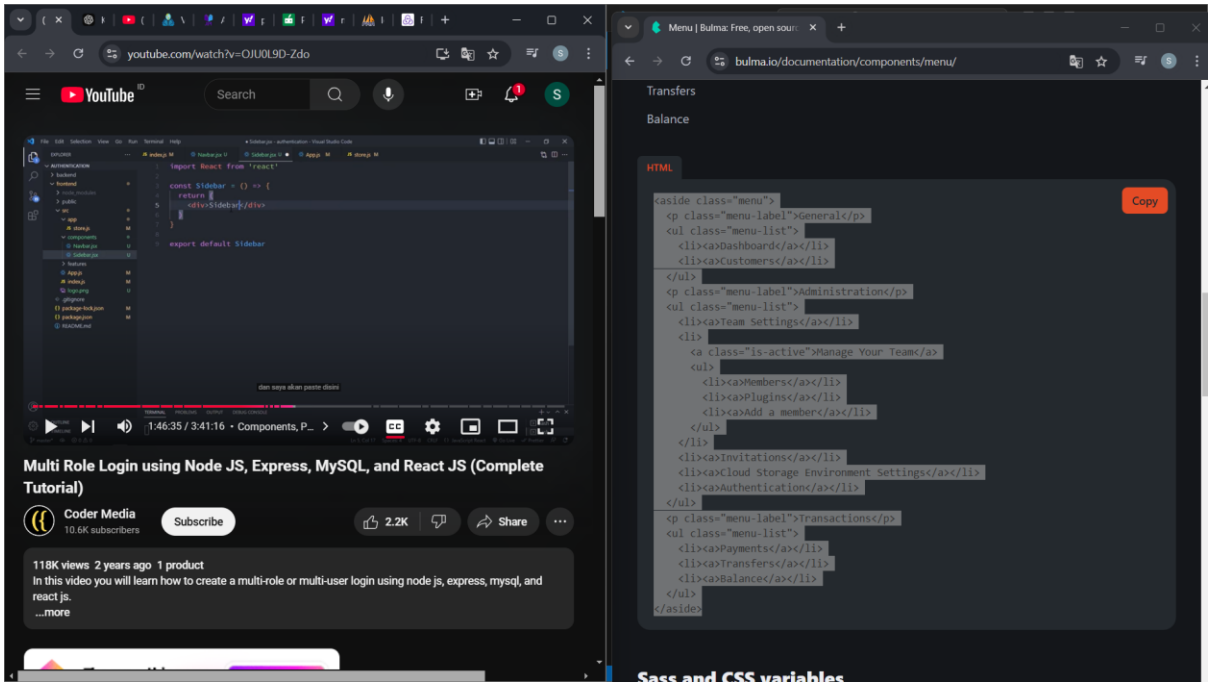


Mendownload extention es7 react dan bulma(untuk mempermudah bermain bulma css)



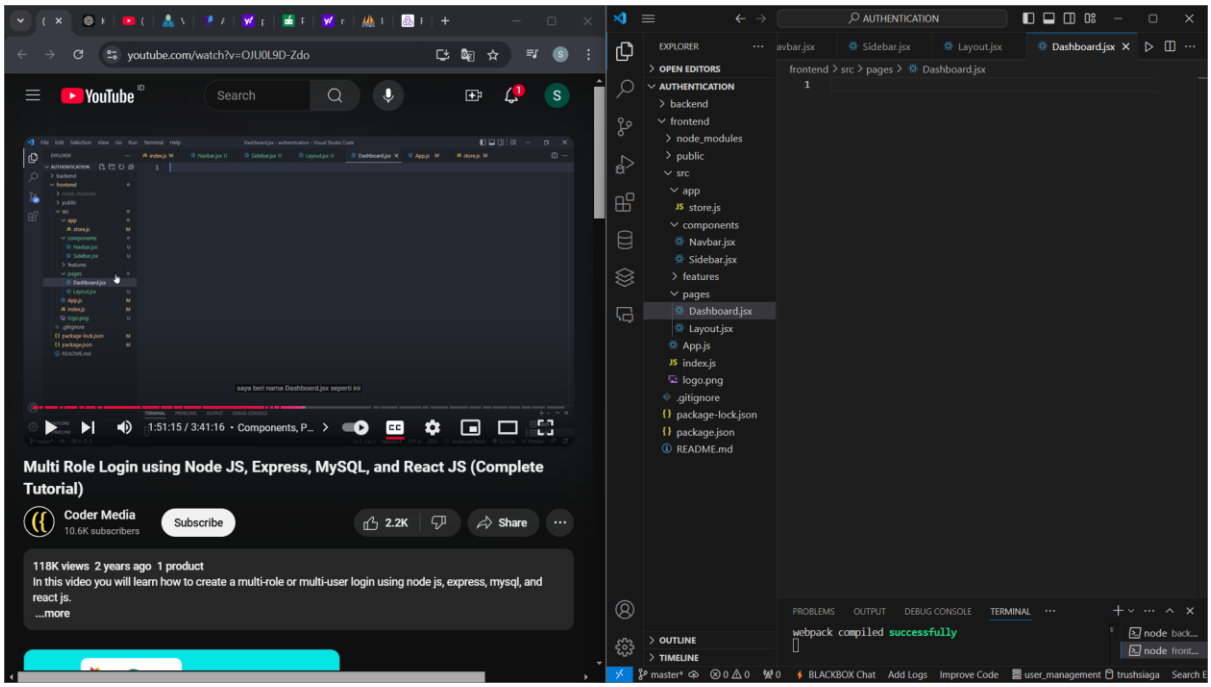


Membuat file sidebar.jsx, lalu ke bulma.io dan pilih yang menu dan copy

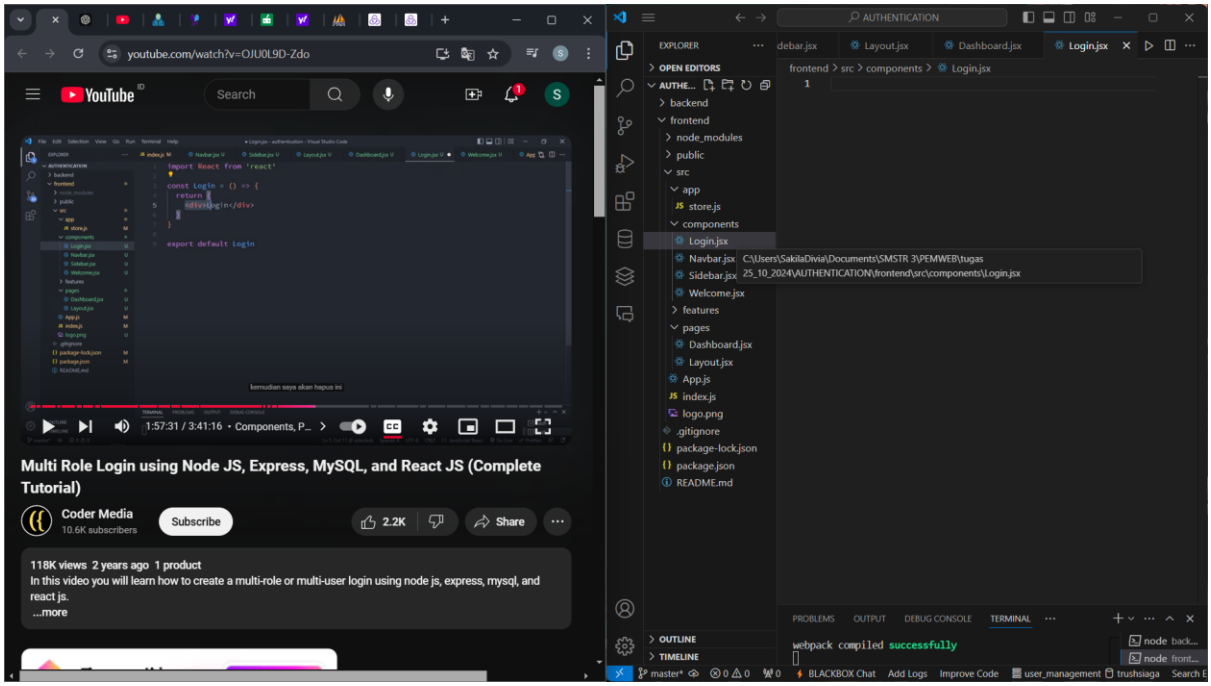


Membuat folder pages>file layout.jsx

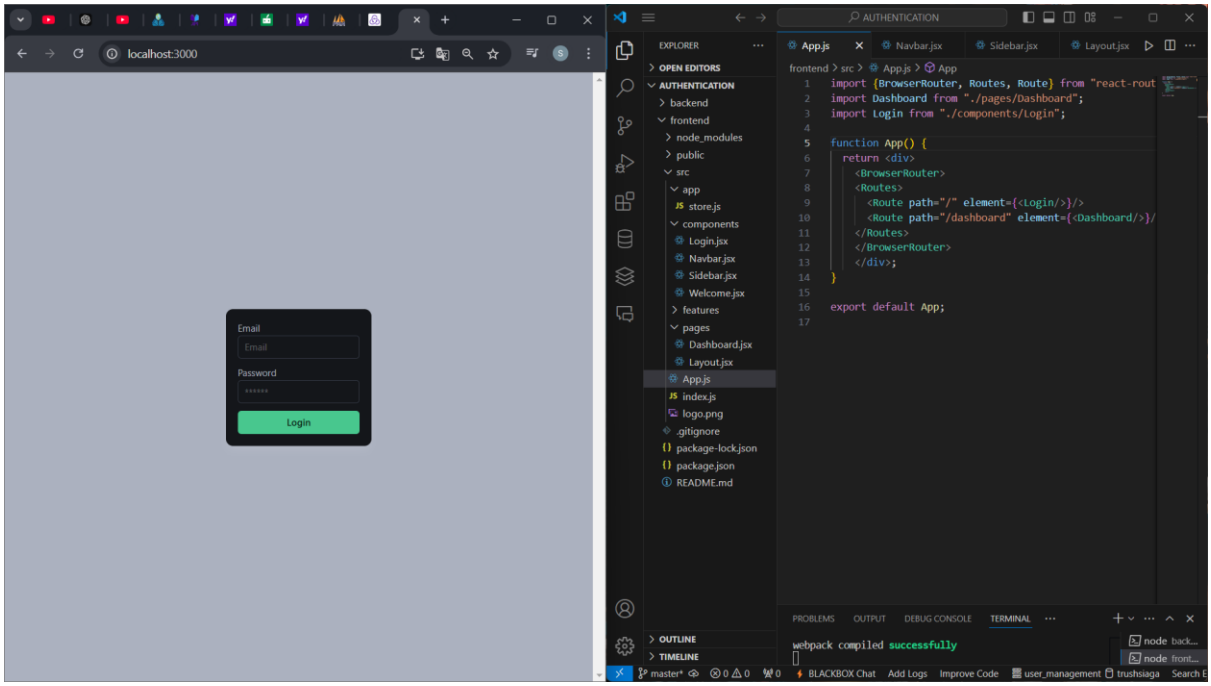
Membuat file dashboard.jsx



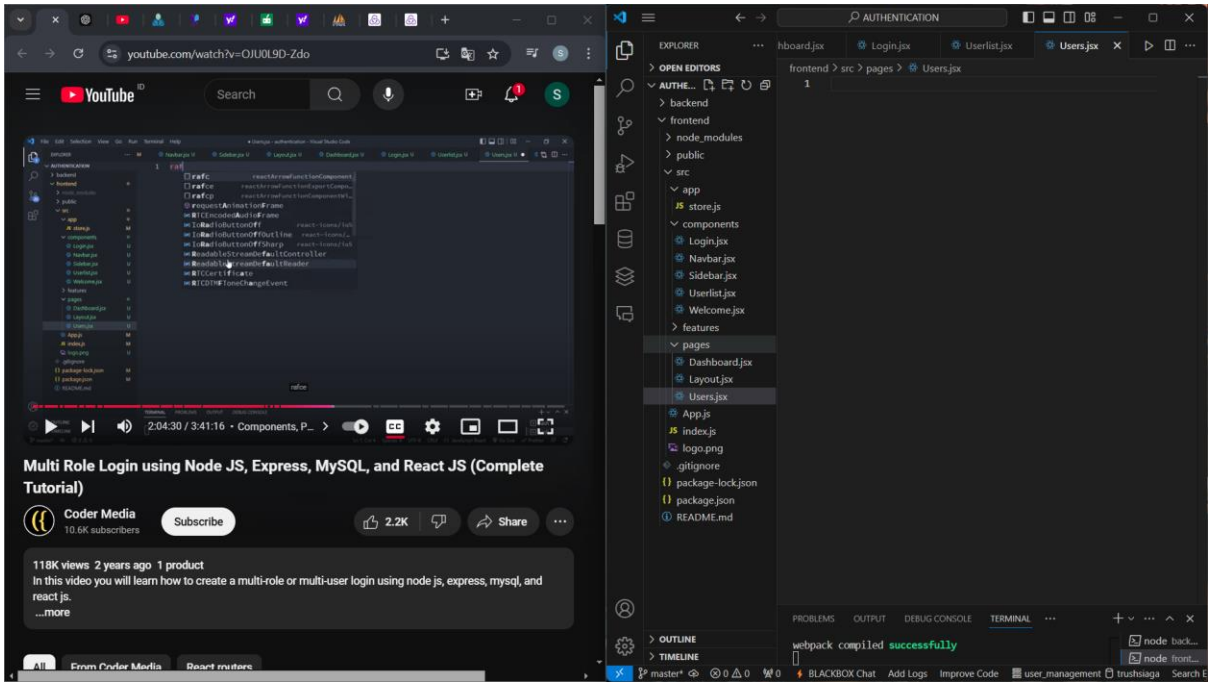
Membuat file login



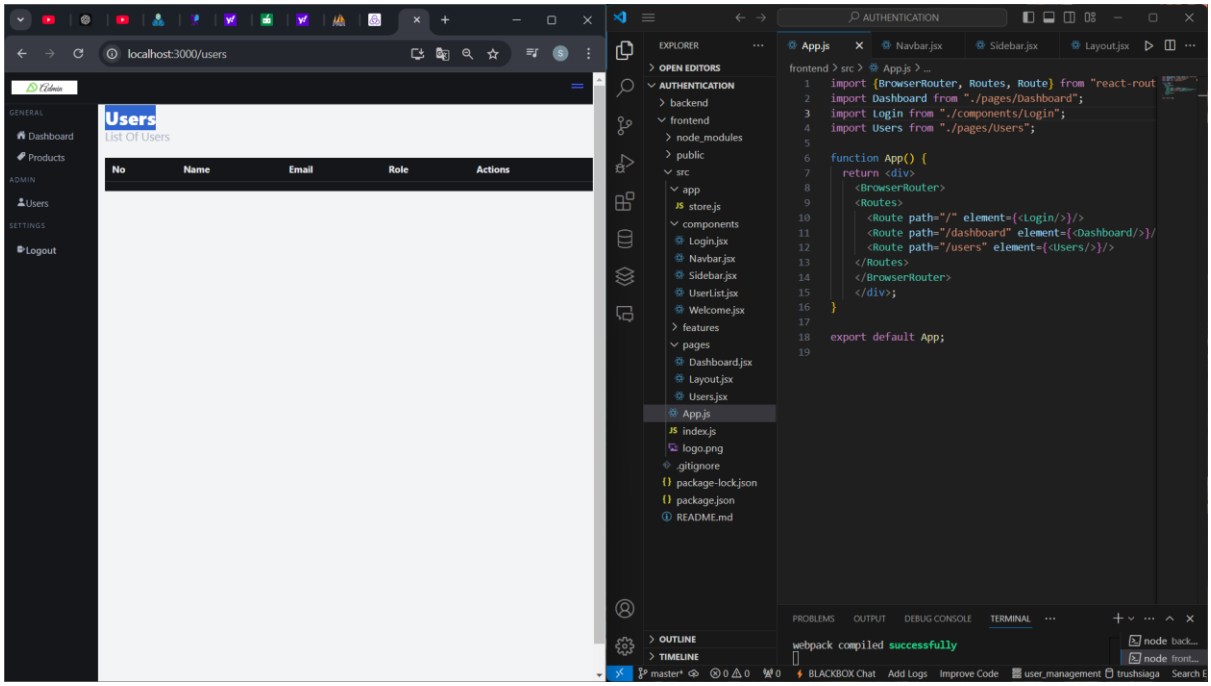
Tampilan login



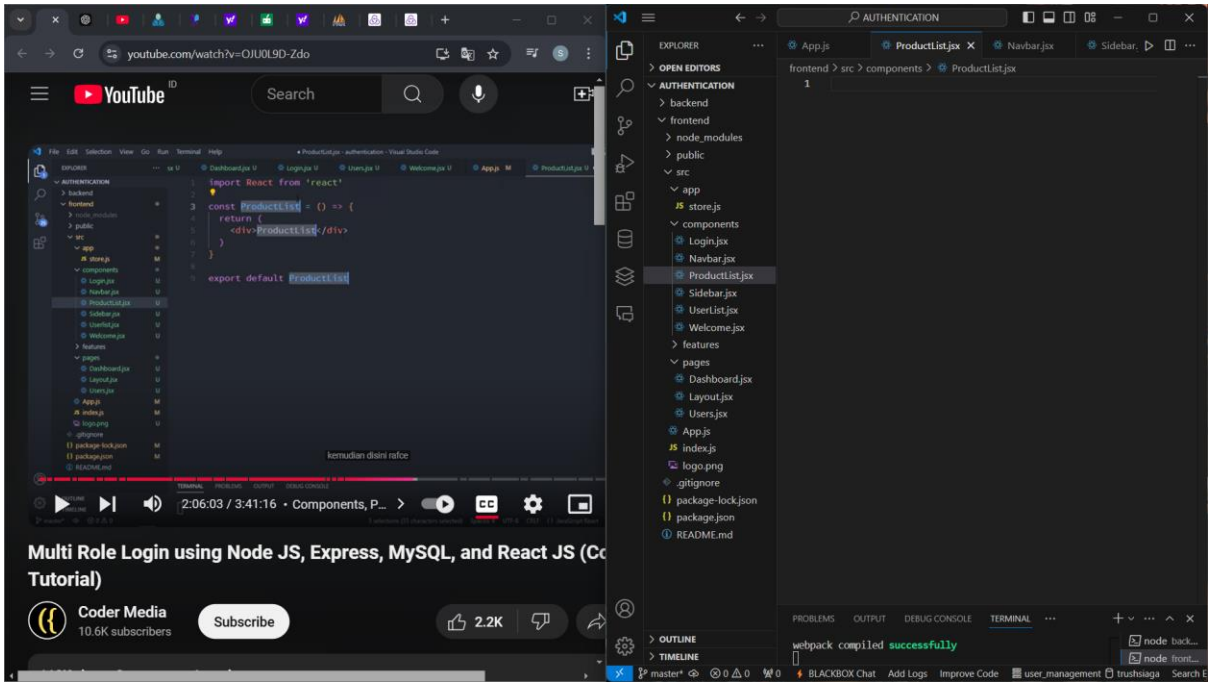
membuT file Userlist .jsx dan users.jsx



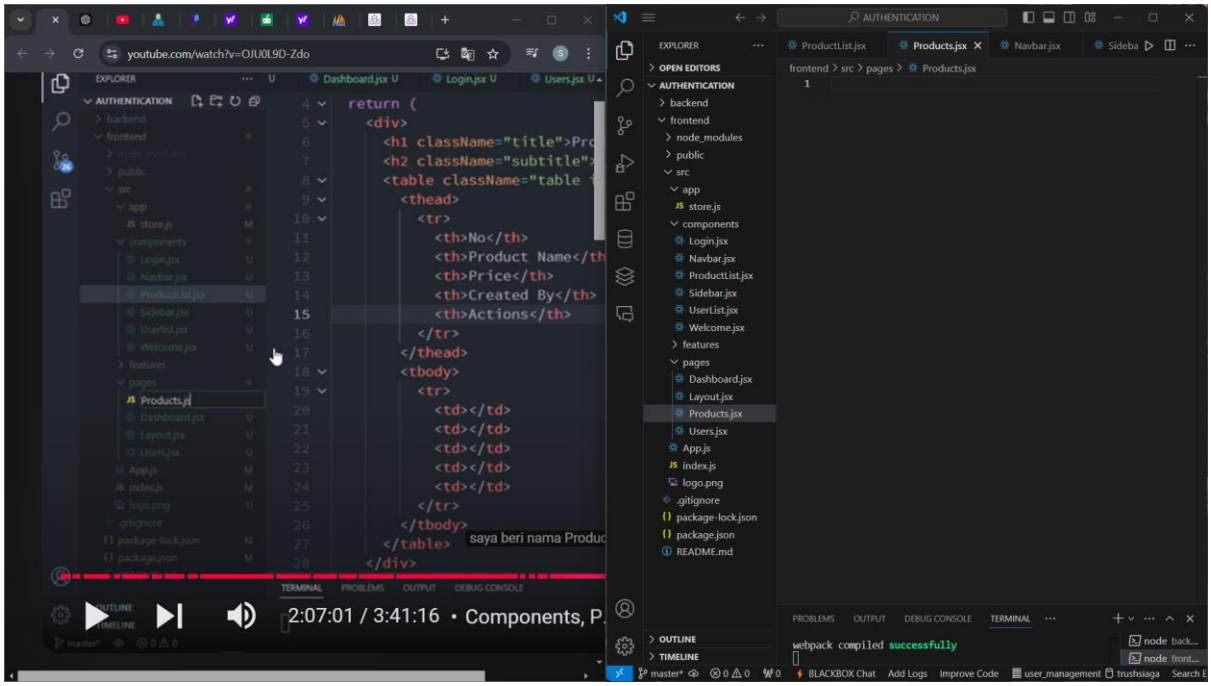
Tampilan users



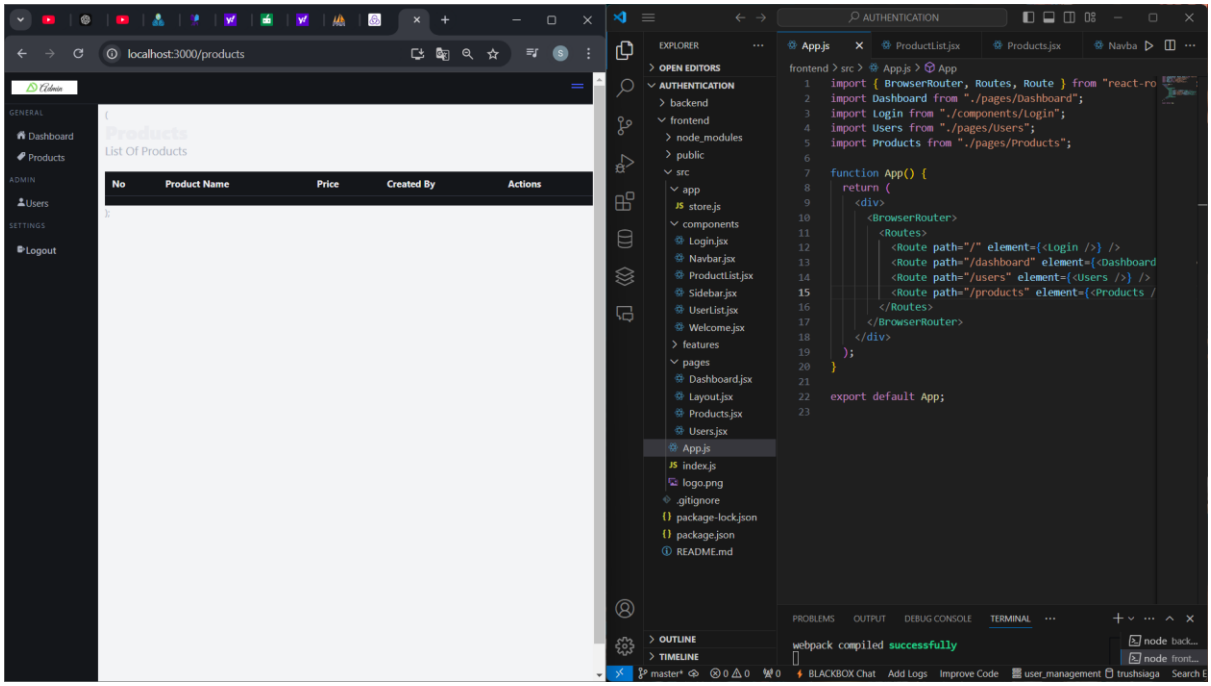
Membuat components productlist.jsx



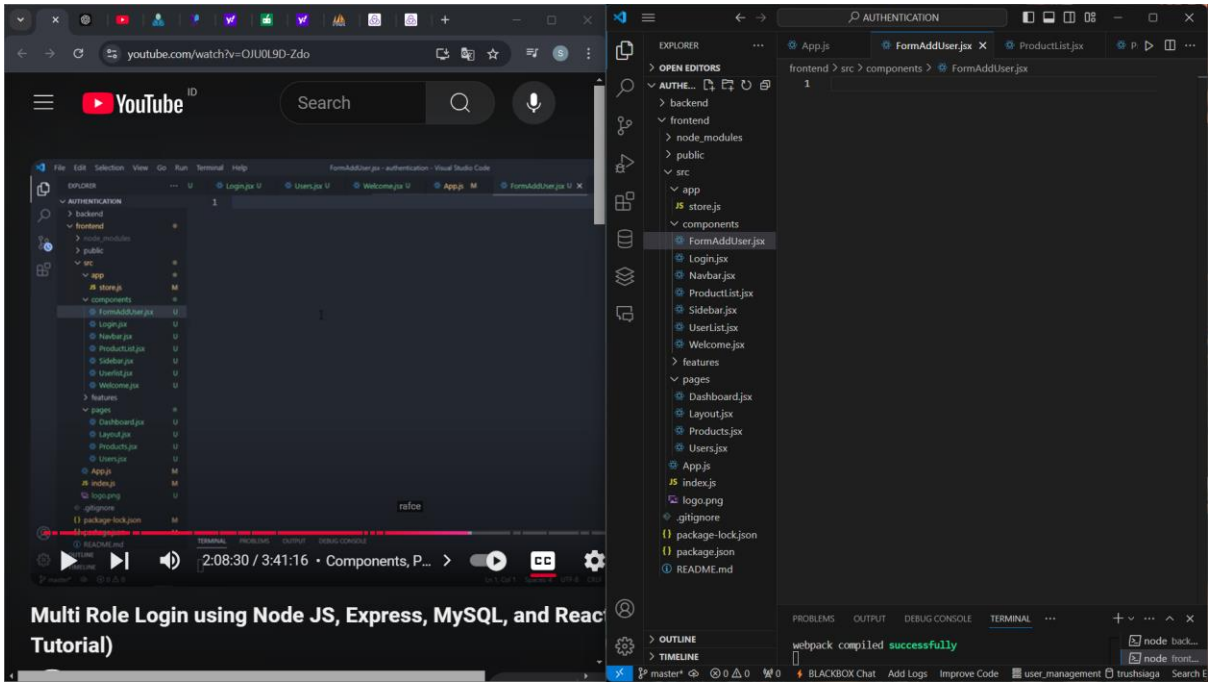
Membuat pages products.jsx



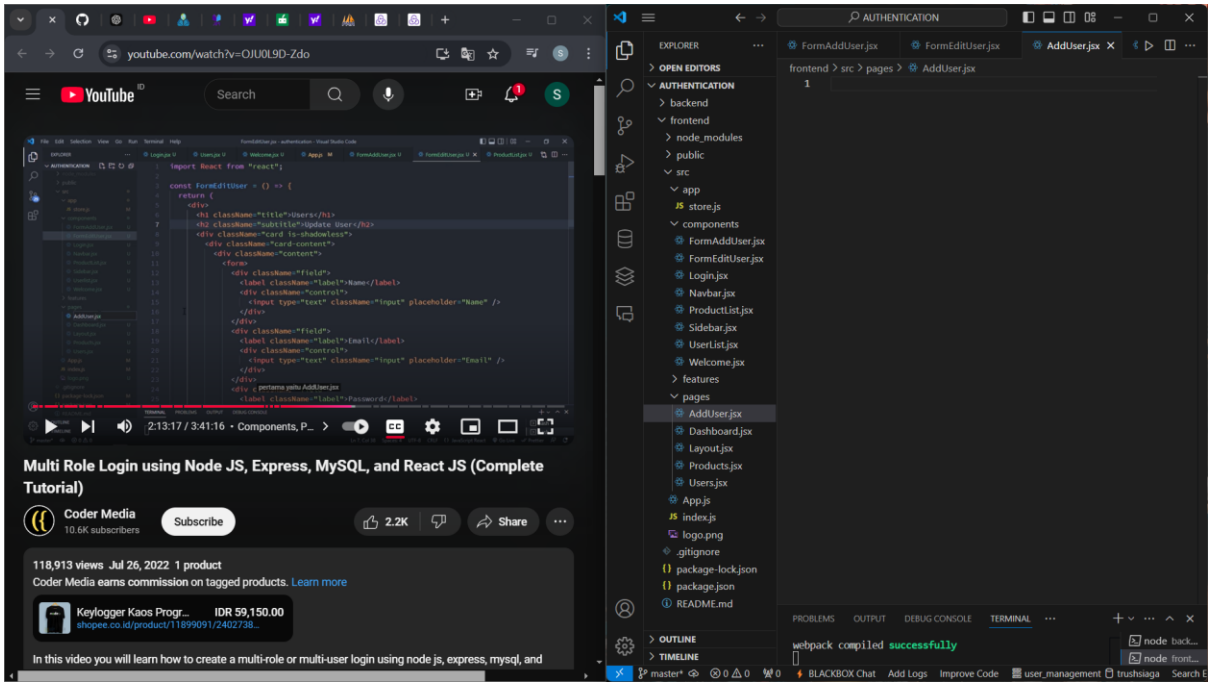
Menampilkan products



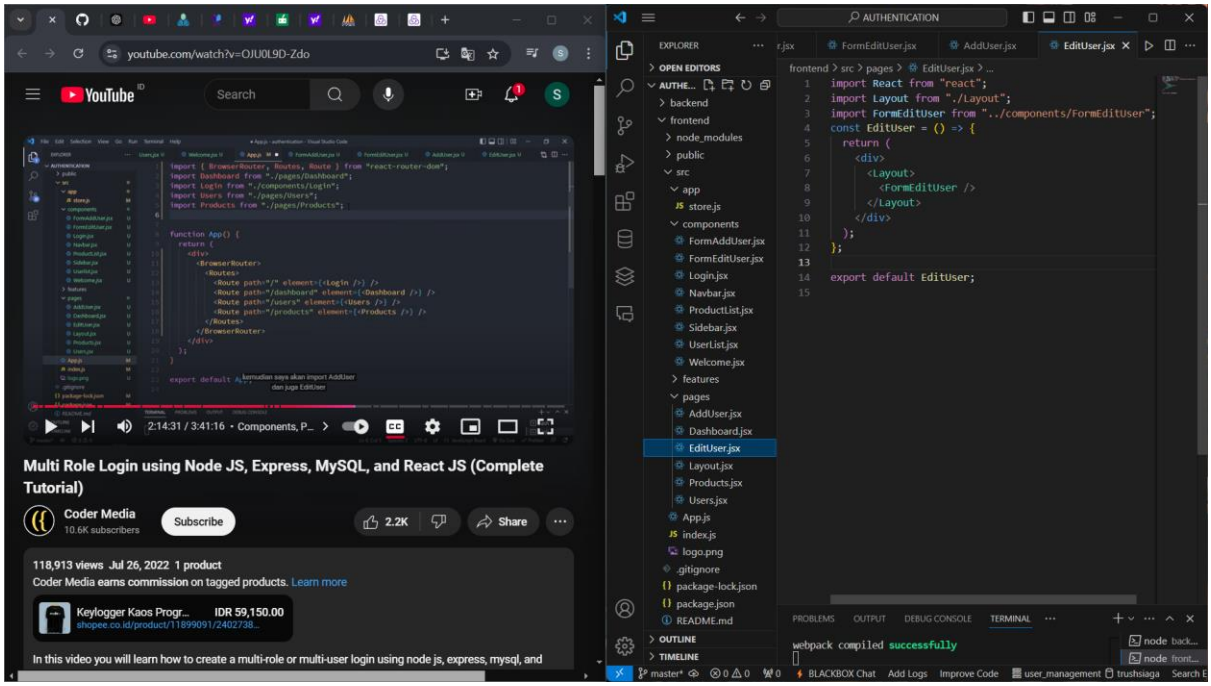
Membuat components formadduser.jsx



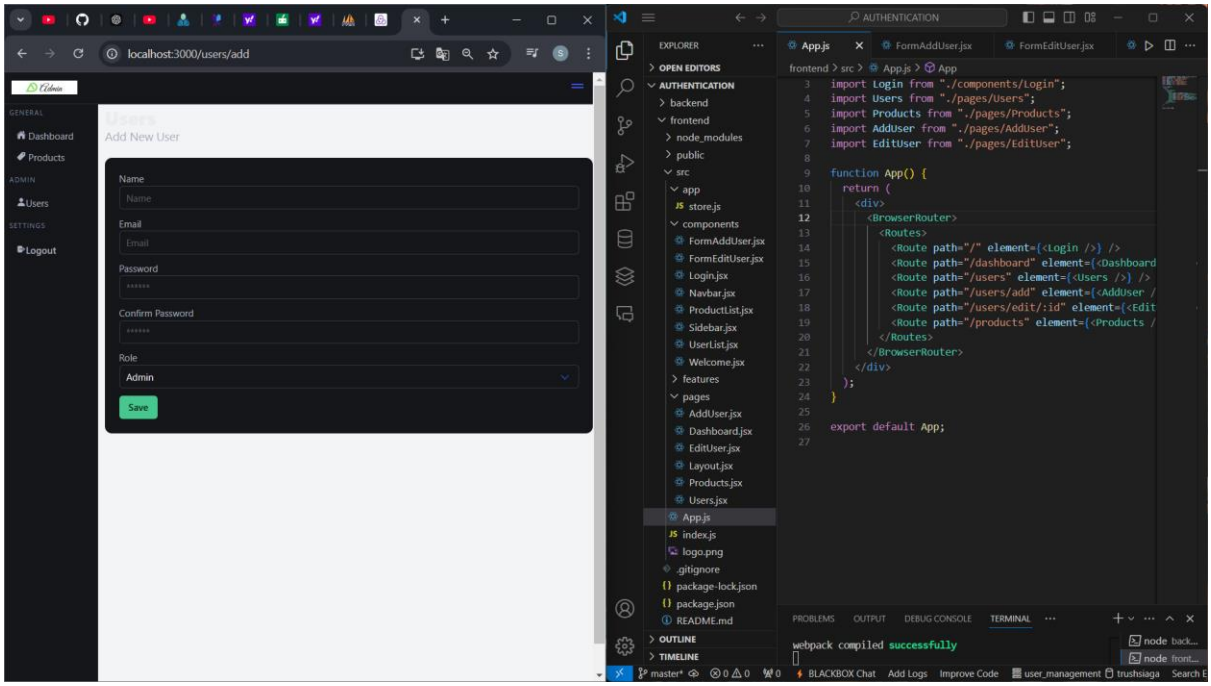
Membuat page file adduser.jsx



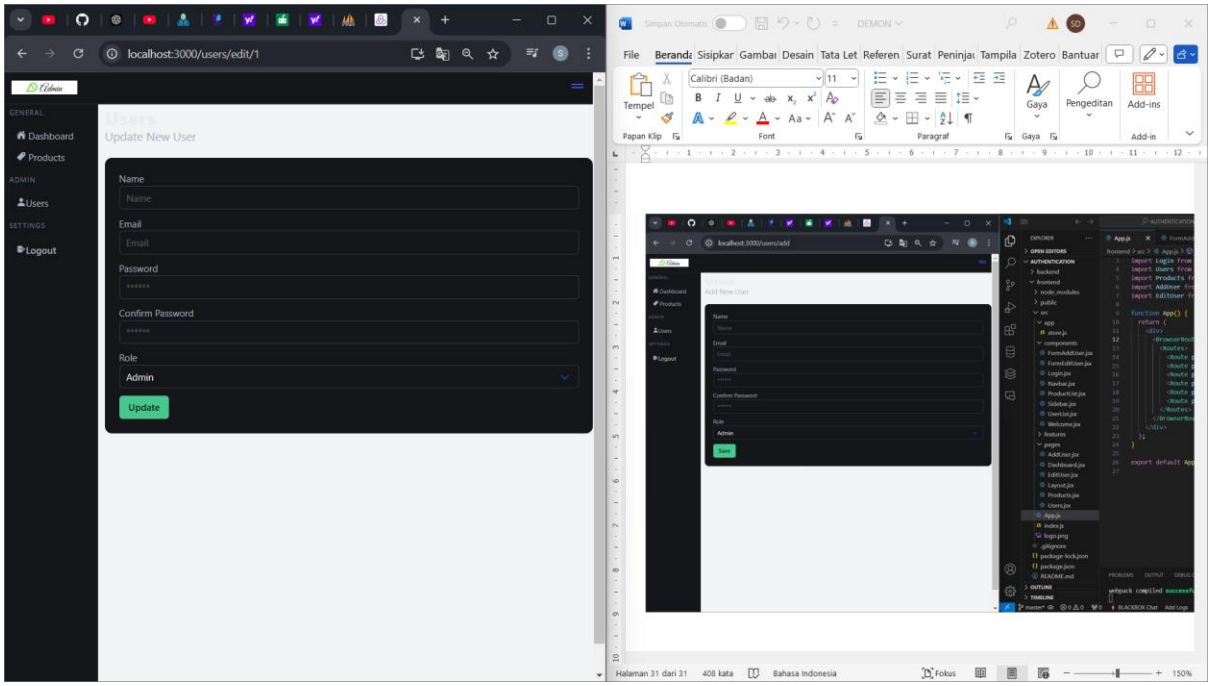
Membuat pages edituser.jsx



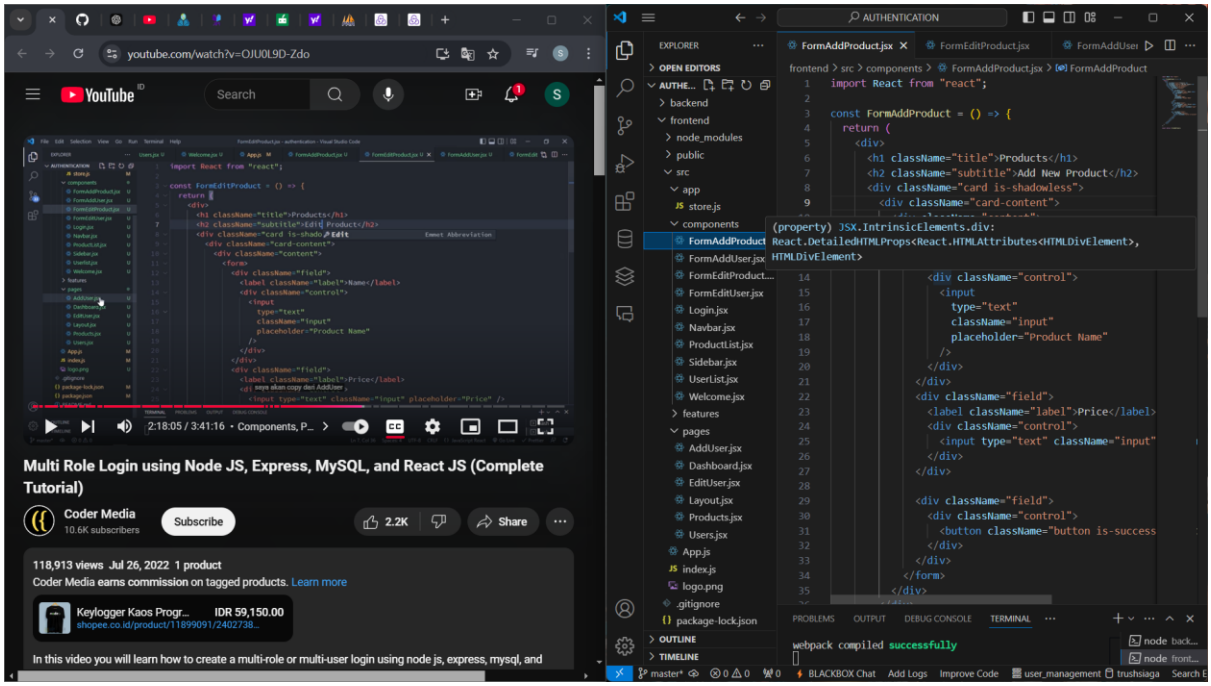
Tampilan localhost users/add



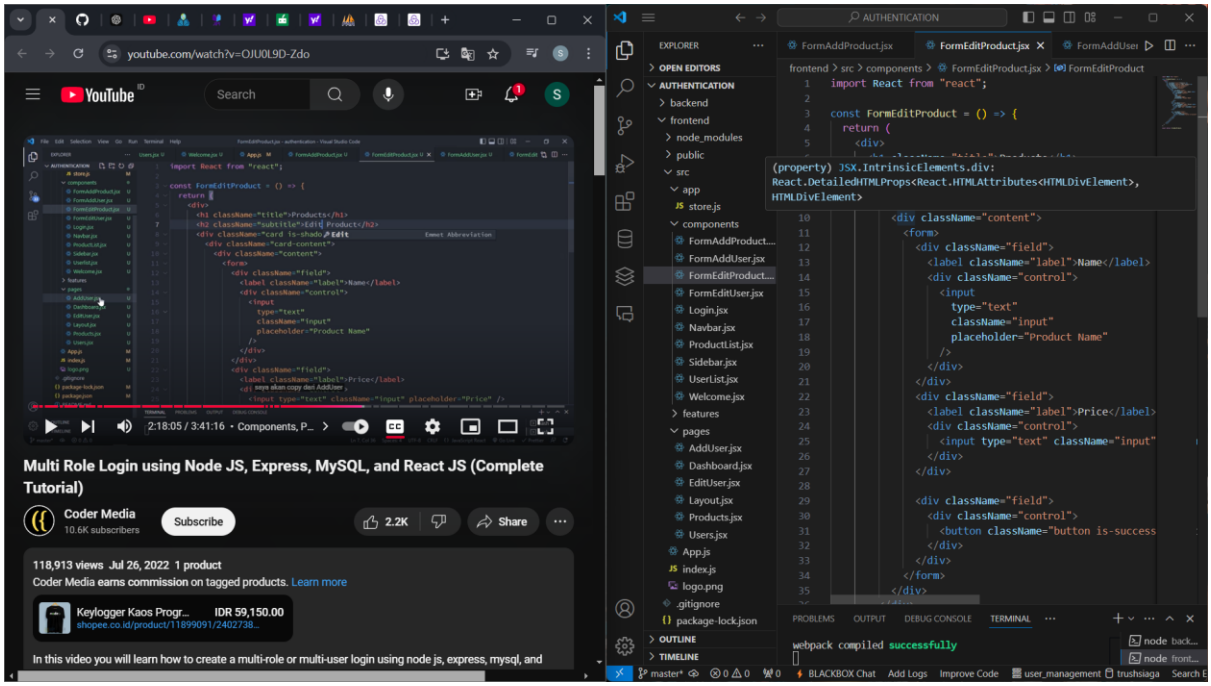
Tampilan jika brows http://localhost:3000/users/edit/1



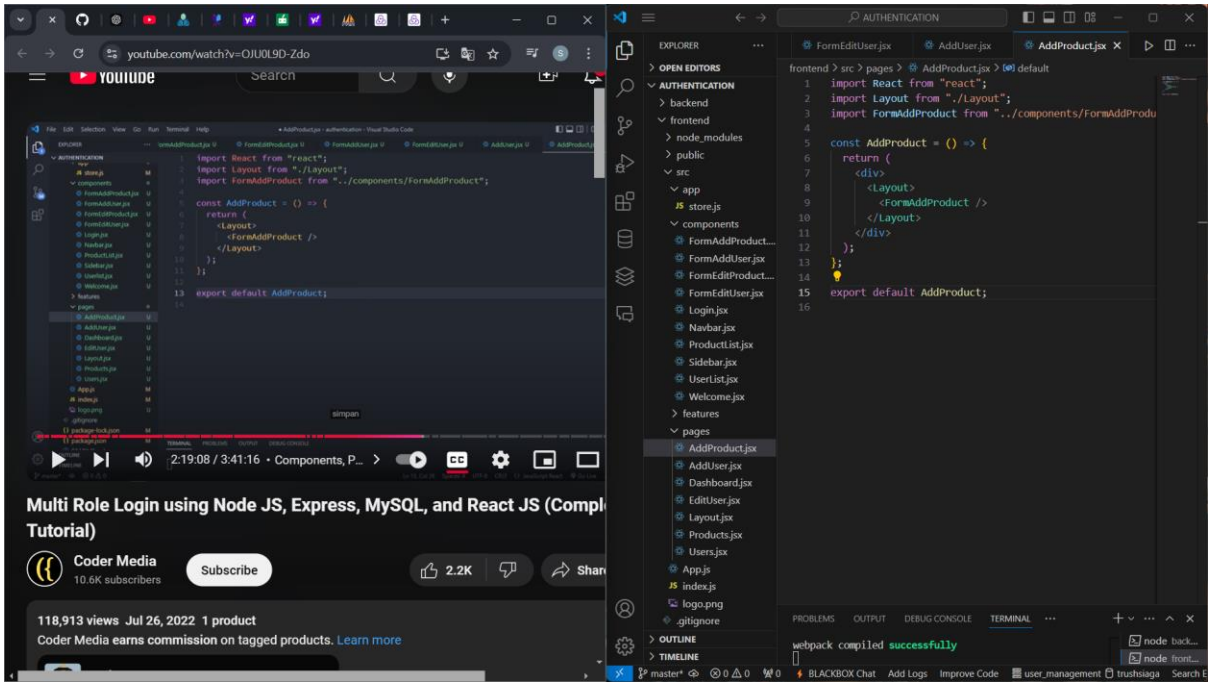
Membuat components formaddproduct



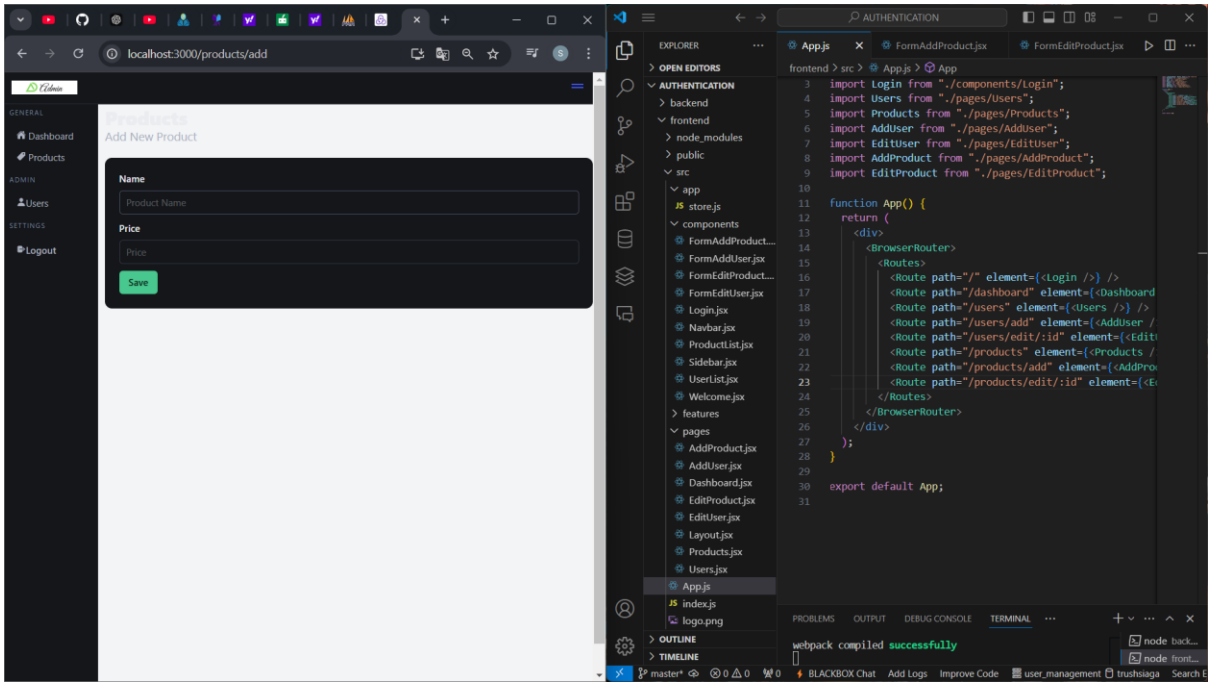
Membuat components formeditproduct



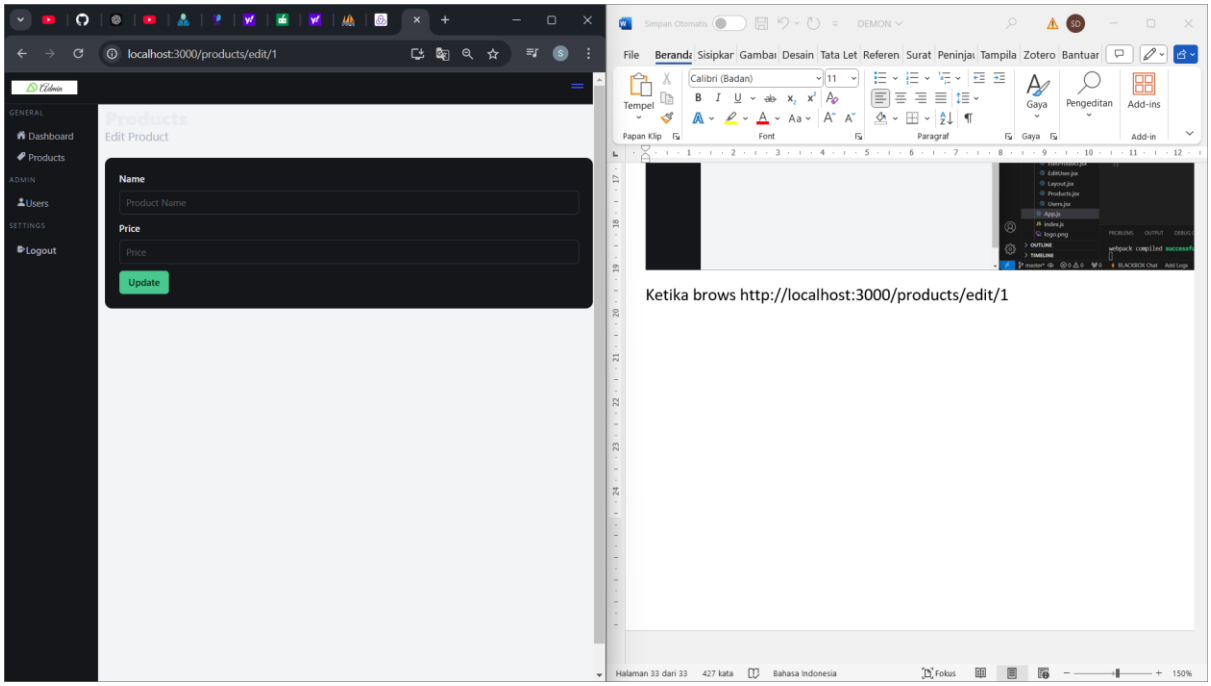
Membuat pages addproduct.jsx



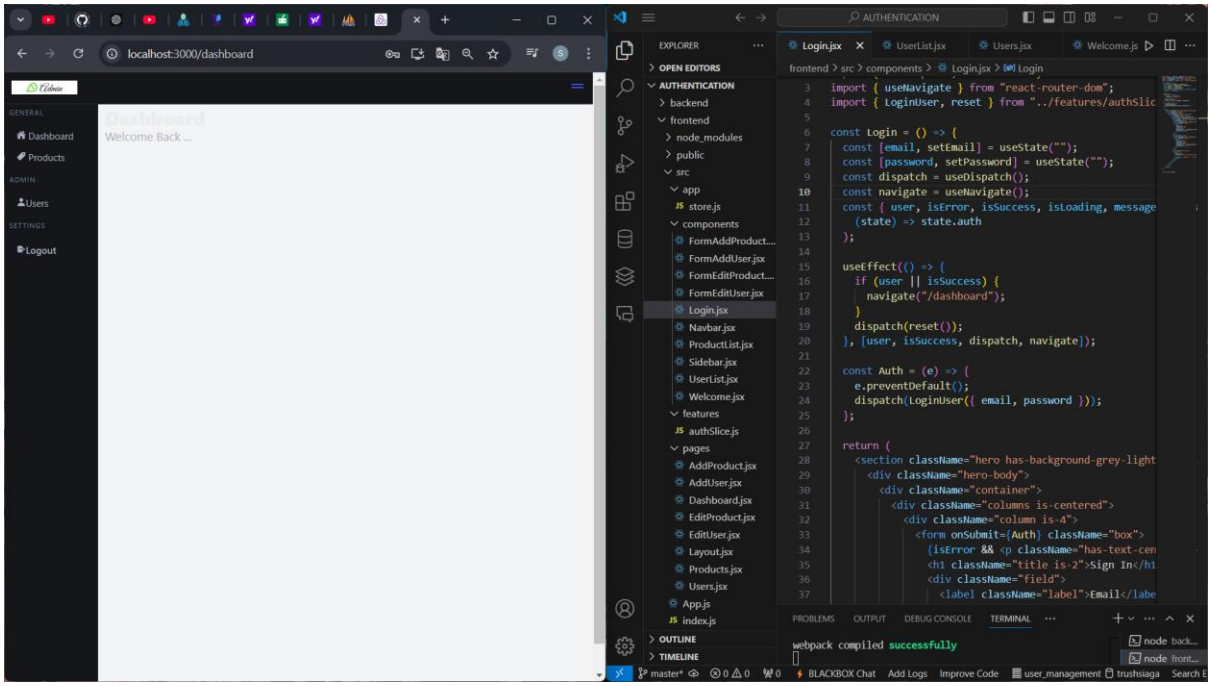
Jika brows http://localhost:3000/products/add



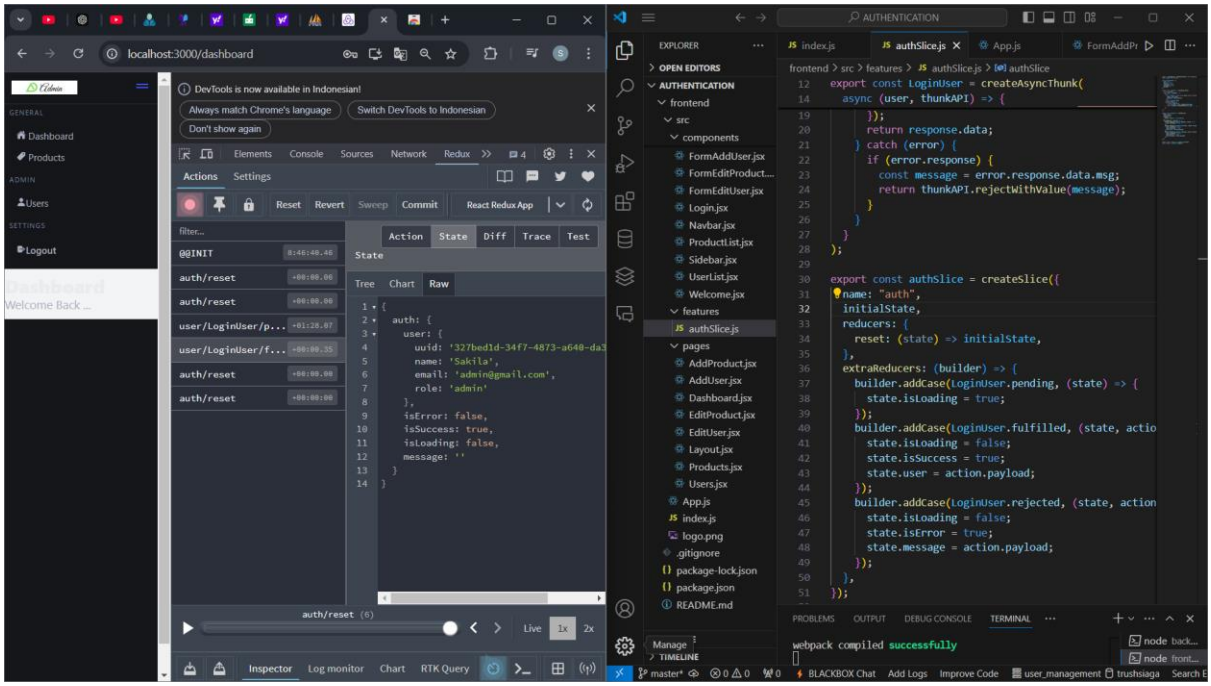
Ketika brows http://localhost:3000/products/edit/1



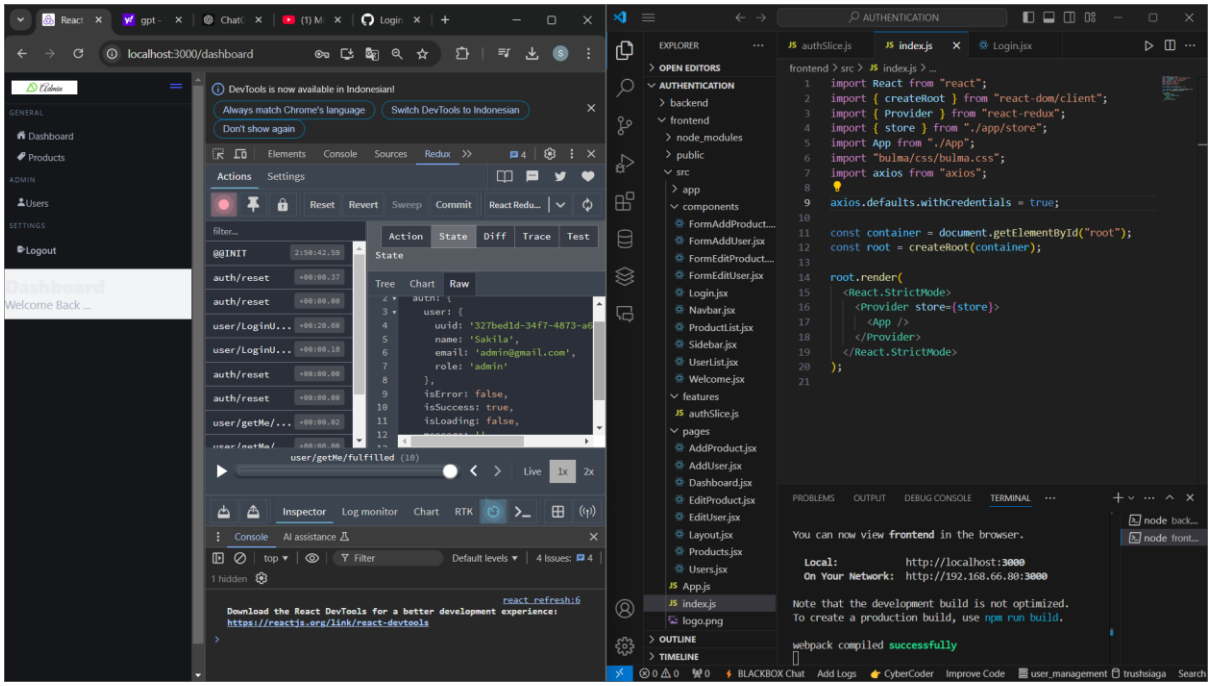
Tampilan login admin



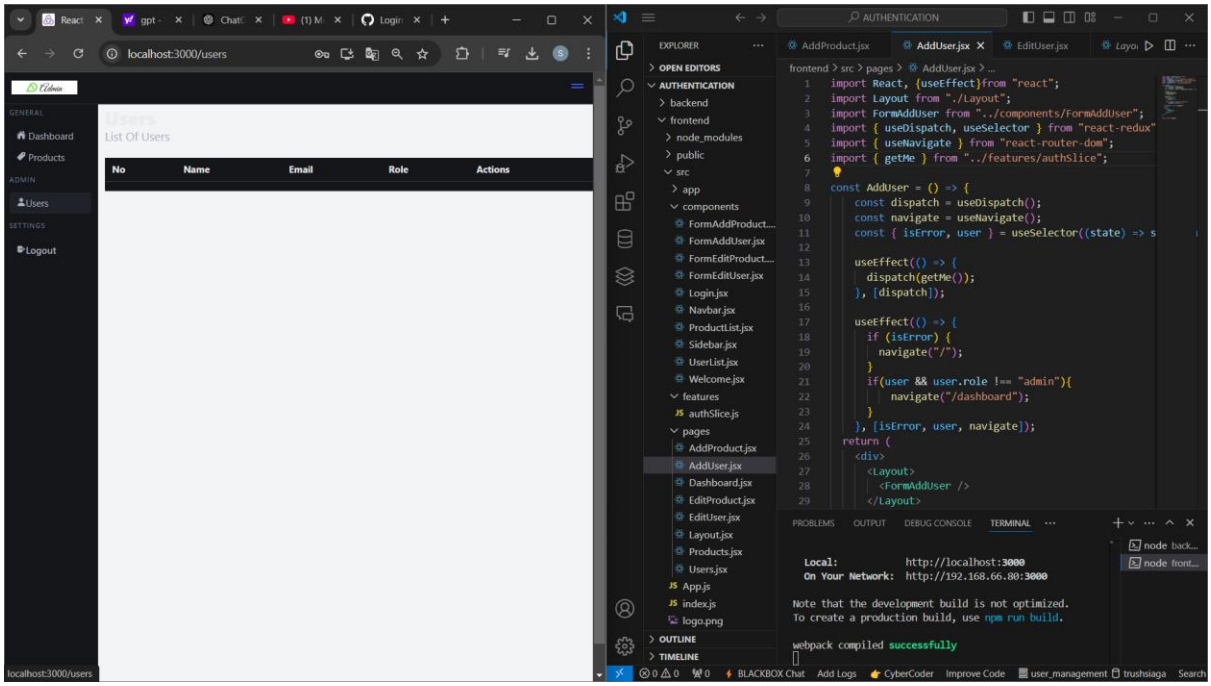
Melakukan pengecekan login berhasil atau tidak pada chrome>redux. Jika berhasil maka akan tampil seperti gambar dibawah ini.



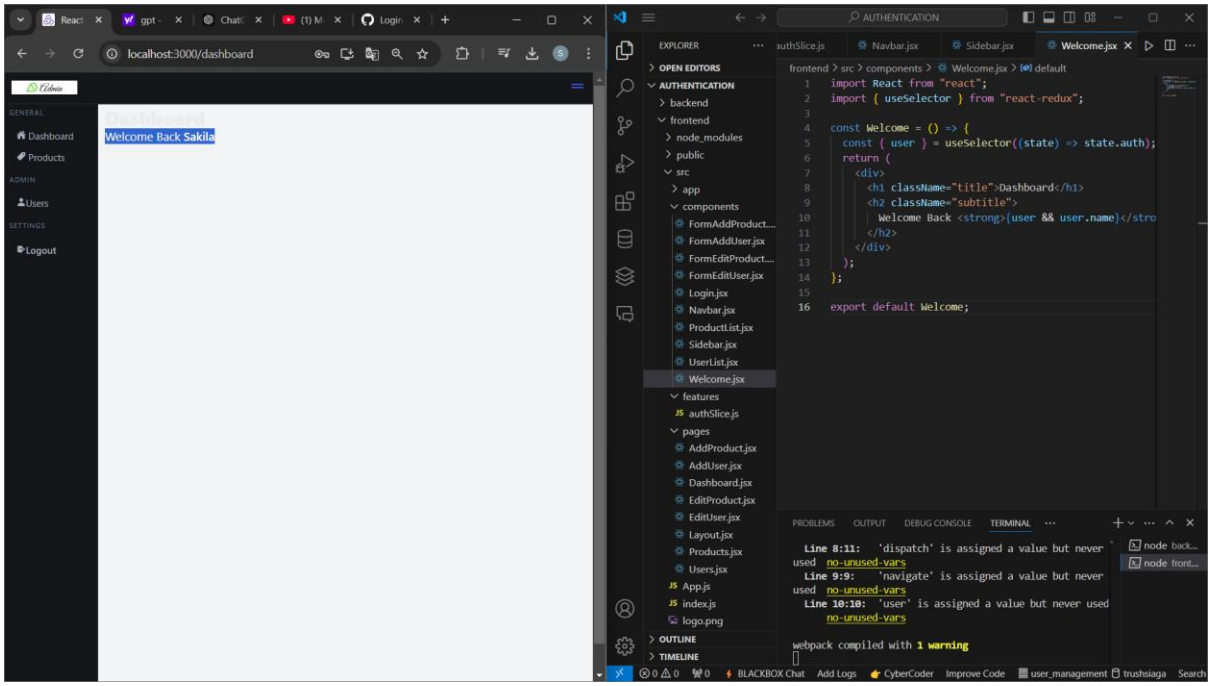
Melakukan proteksi terhadap halaman dashboard



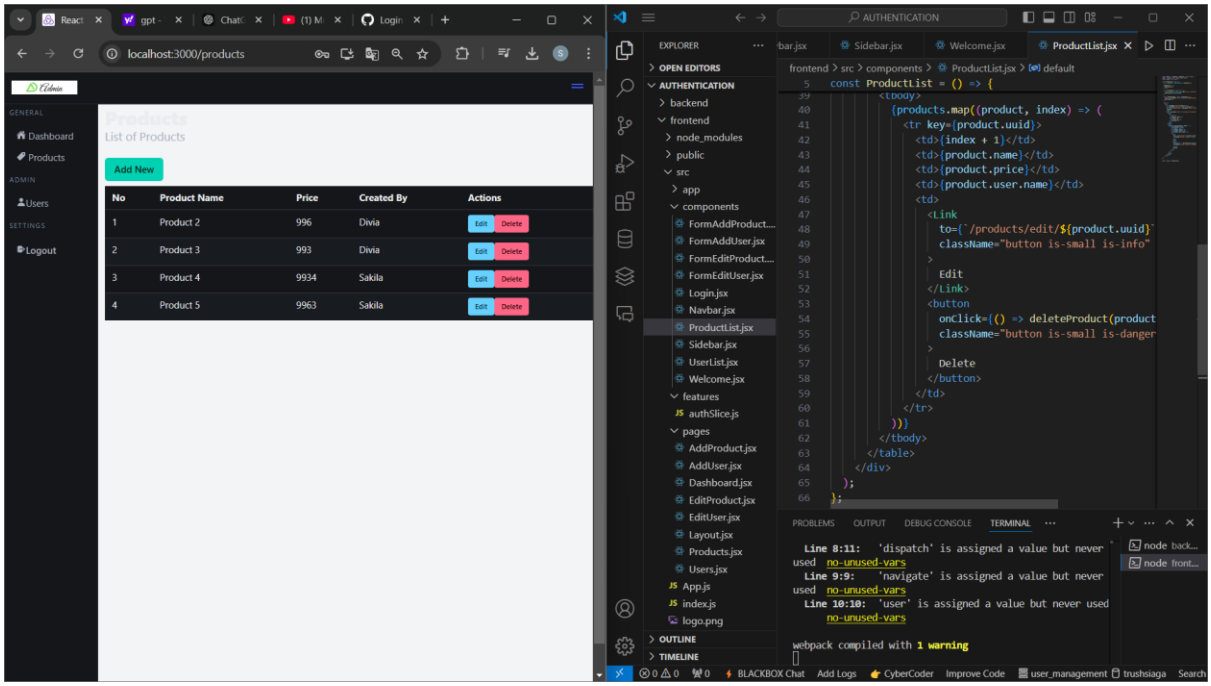
Protected all page



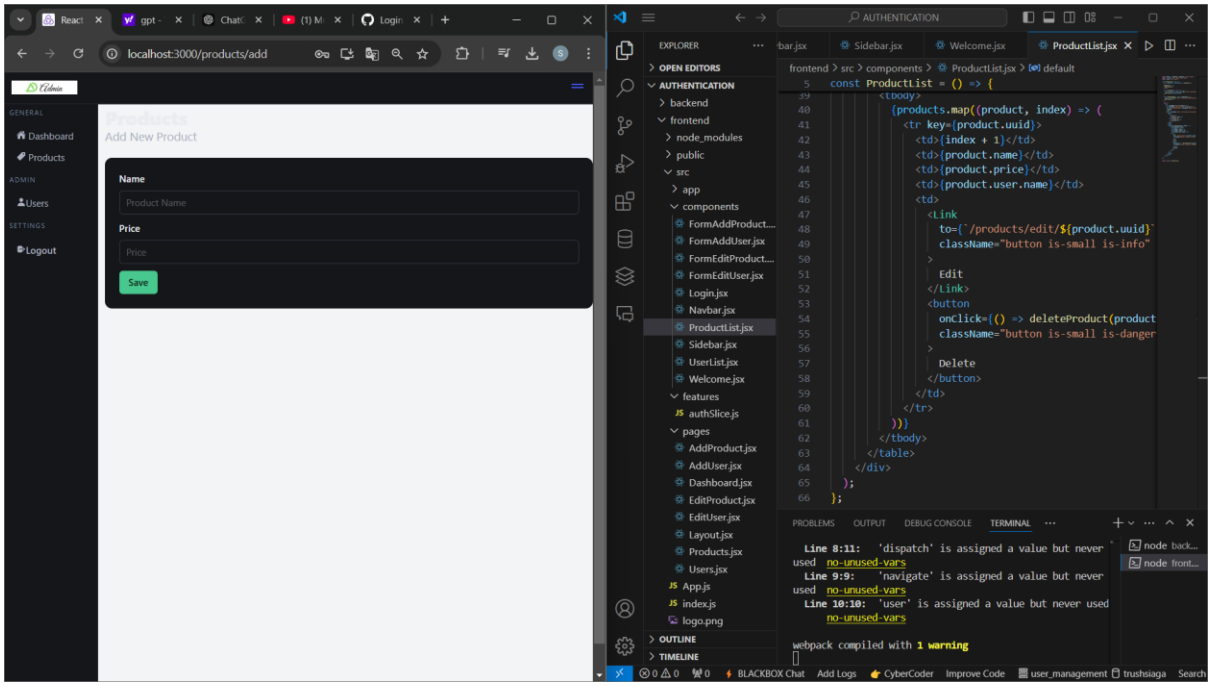
Menampilkan nama admin



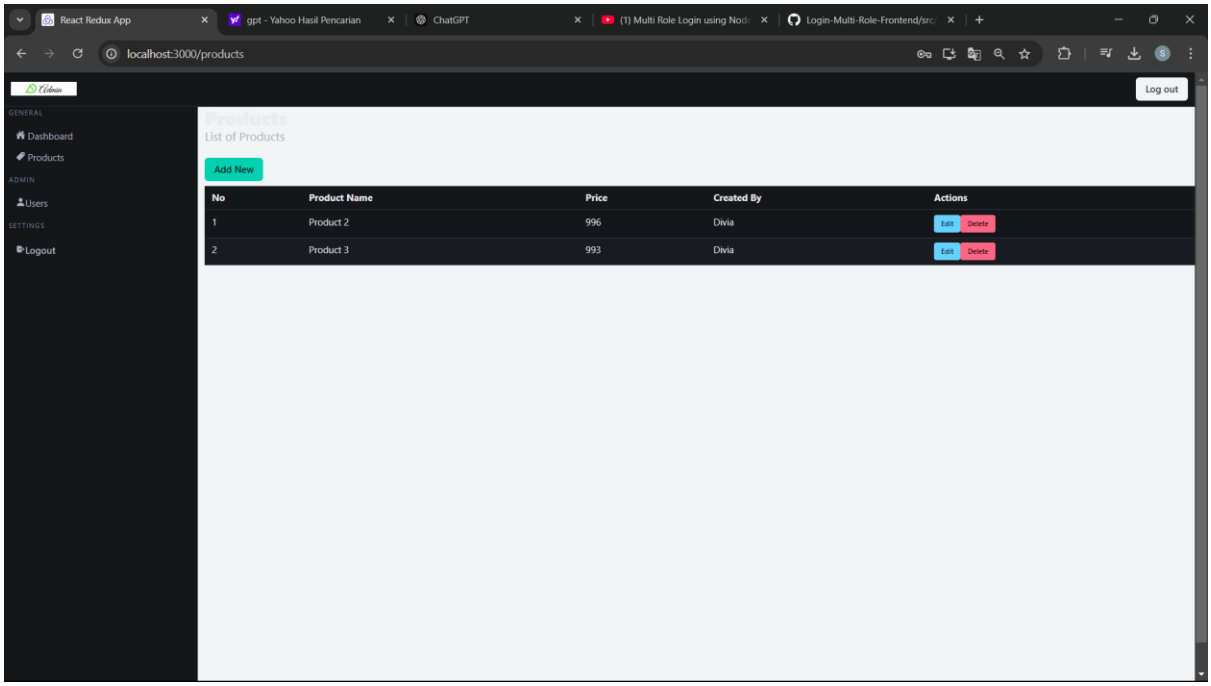
Menampilkan product list



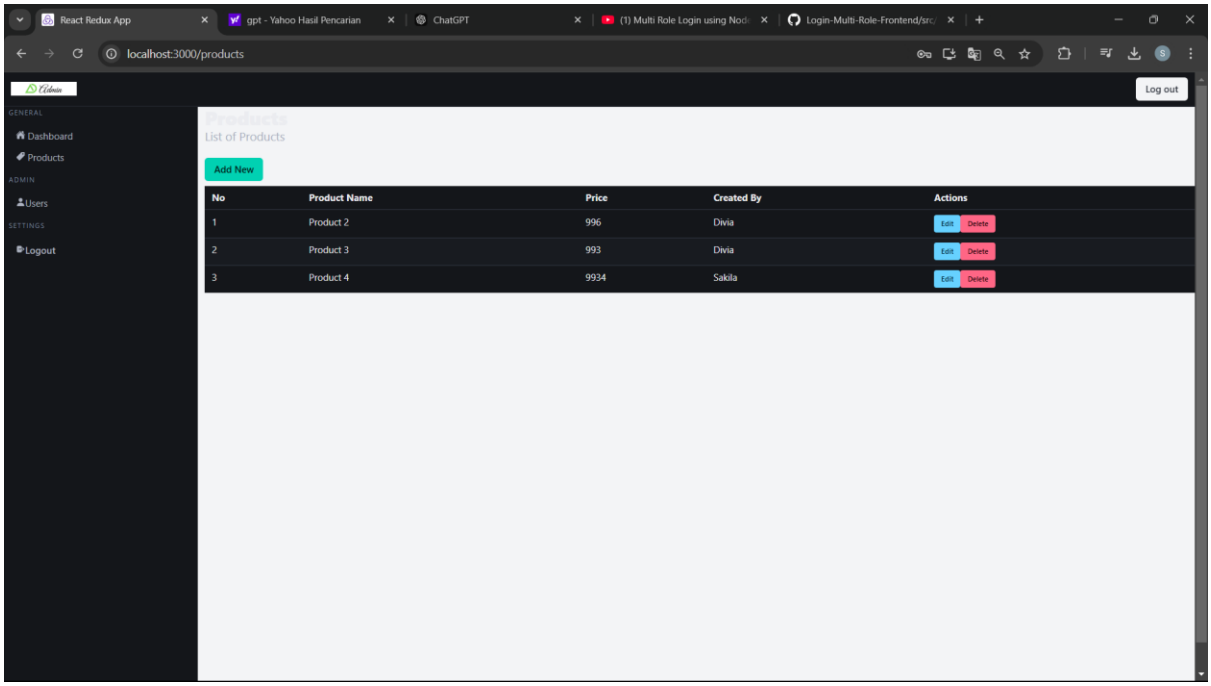
Tampilan add new



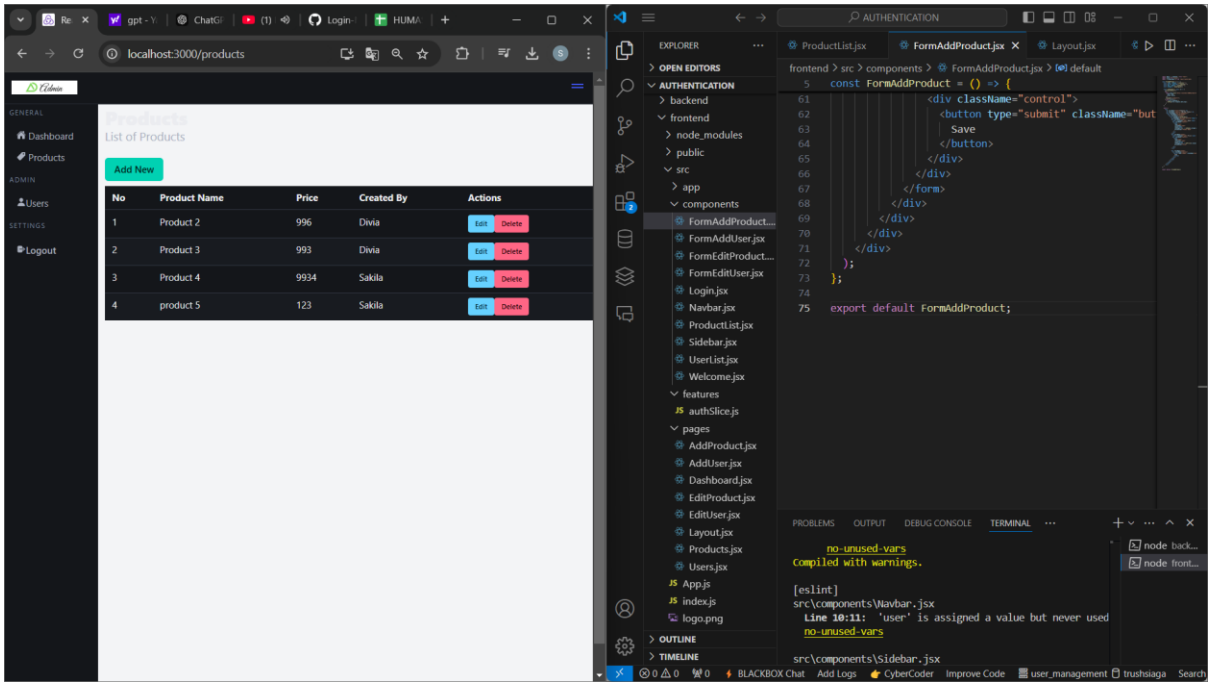
Jika user login maka hanya akan tampil product user



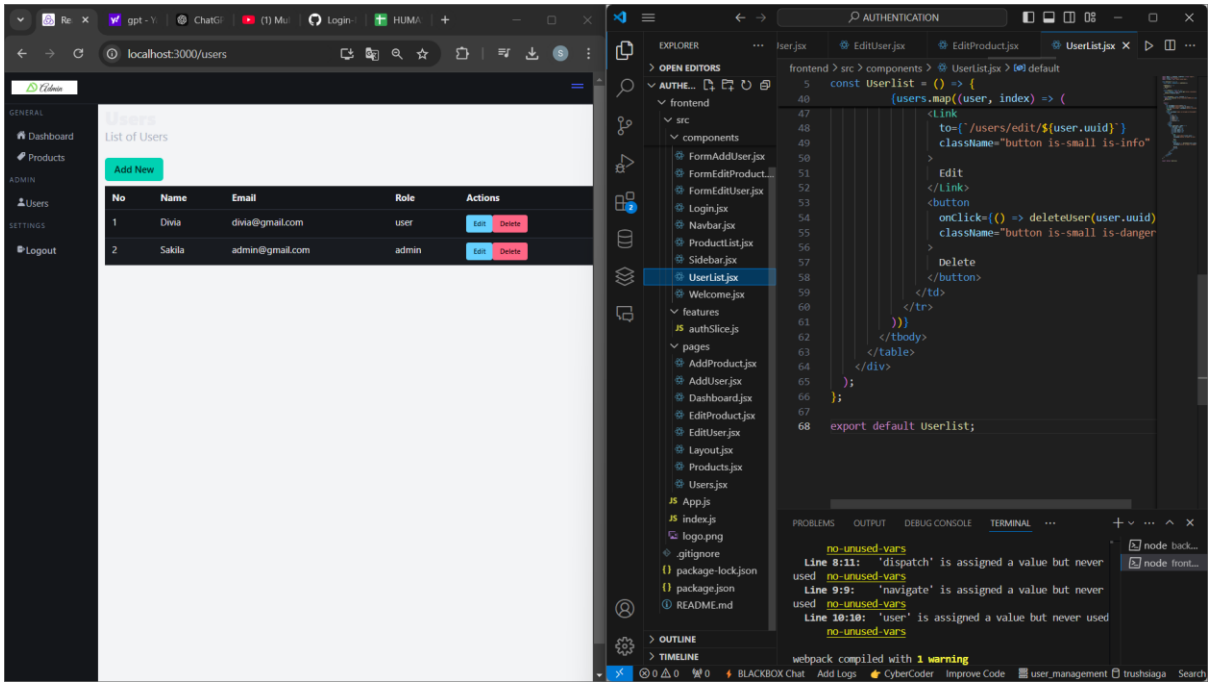
Jika login kembali sebagai admin



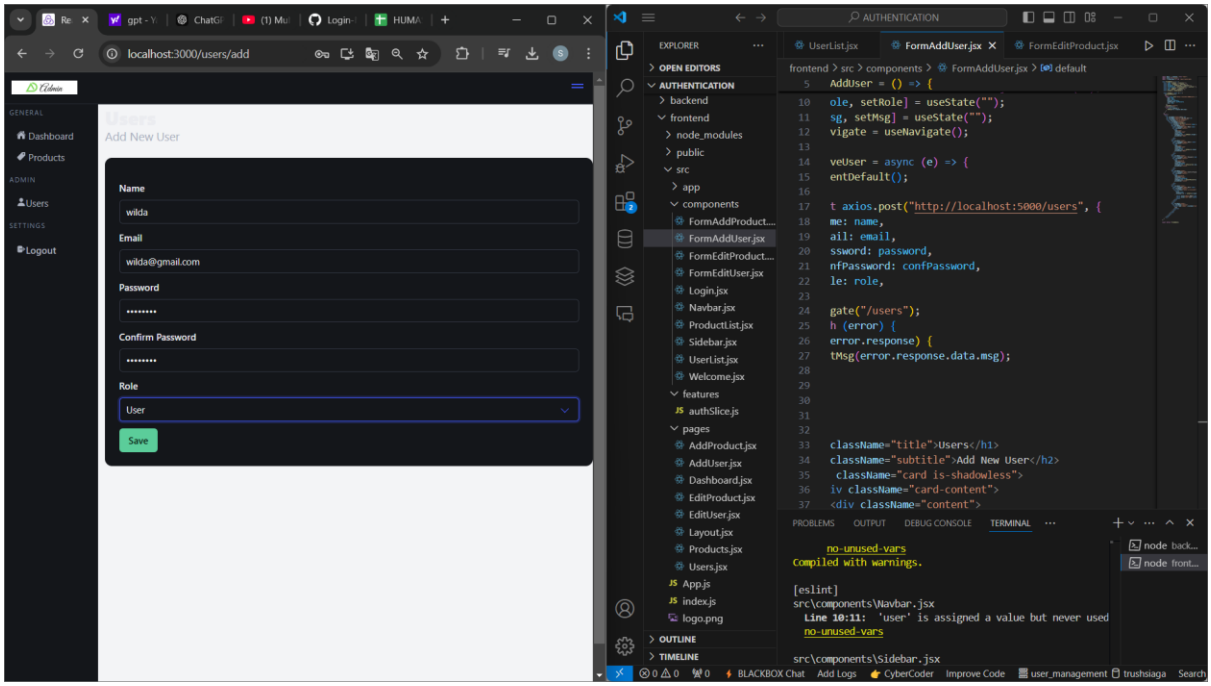
Tampilan setelah menambah data product

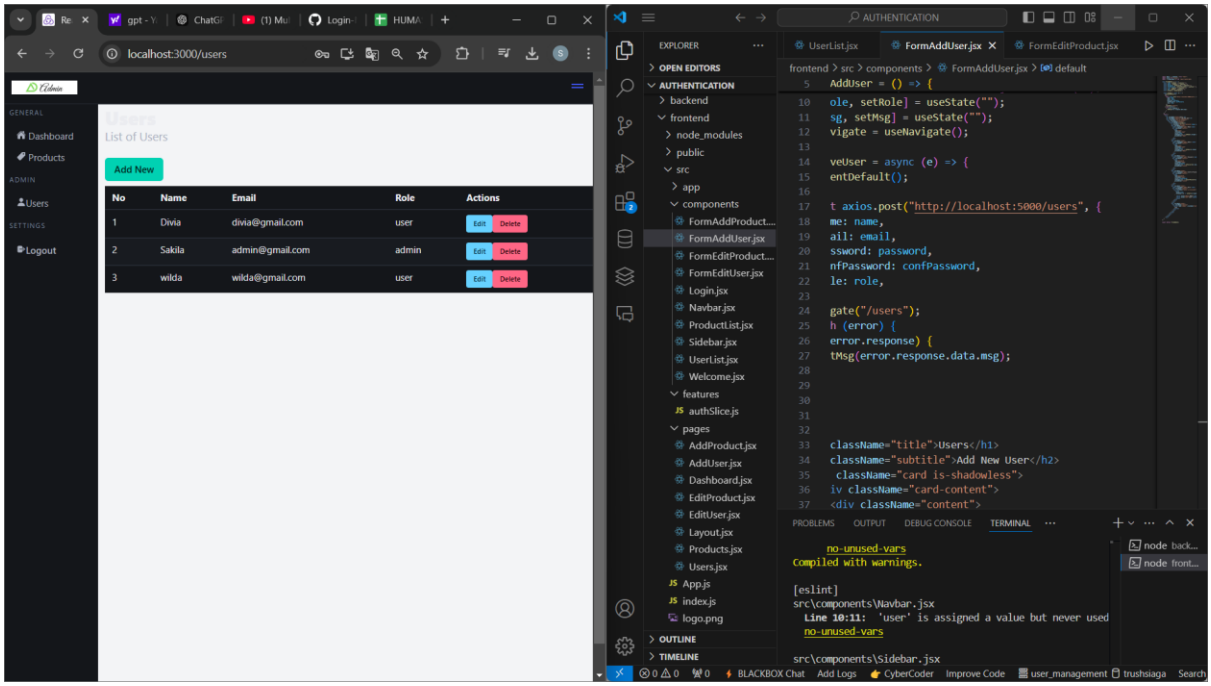


Tampilan list of users

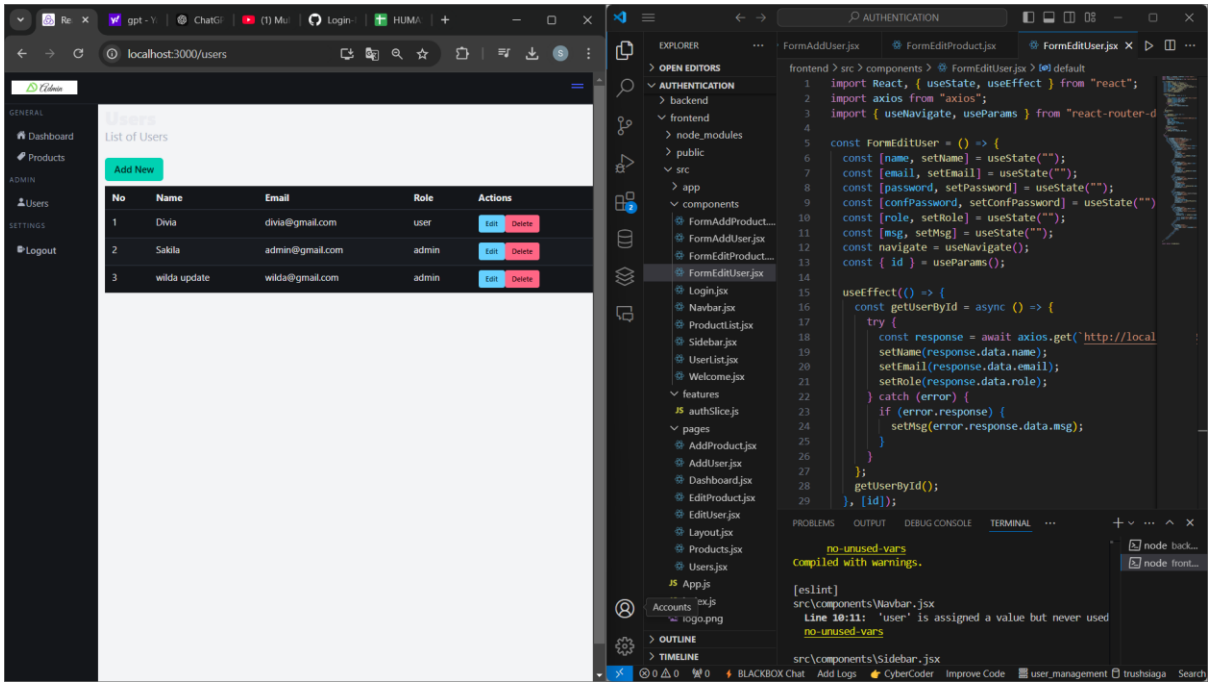


Menjalankan fungsi adduser





Update users


localhost:3000/users
Admin
GENERAL
Dashboard
Products
ADMIN
Users
SETTINGS
Logout
Users
List of Users
Add New
No
Name
Email
Role
Actions
1
Divia
divia@gmail.com
user
Edit
Delete
2
Sakila
admin@gmail.com
admin
Edit
Delete
3
wilda update
wilda@gmail.com
admin
Edit
Delete
AUTHENTICATION
EXPLORER
OPEN EDITORS
AUTHENTICATION
backend
frontend
node_modules
public
src
app
components
FormAddProduct....
FormAddUser.jsx
FormEditProduct....
FormEditUser.jsx
Login.jsx
Navbar.jsx
ProductList.jsx
Sidebar.jsx
UserList.jsx
Welcome.jsx
features
authSlice.js
pages
AddProduct.jsx
AddUser.jsx
Dashboard.jsx
EditProduct.jsx
EditUser.jsx
Layout.jsx
Products.jsx
Users.jsx
App.js
logo.png
OUTLINE
TIMELINE
FormAddUser.jsx
FormEditProduct.jsx
FormEditUser.jsx
frontend > src > components > FormEditUser.jsx > default
import React, { useState, useEffect } from "react";
import axios from "axios";
import { useNavigate, useParams } from "react-router-d
const FormEditUser = () => {
const [name, setName] = useState("");
const [email, setEmail] = useState("");
const [password, setPassword] = useState("");
const [confPassword, setConfPassword] = useState("")
const [role, setRole] = useState("");
const [msg, setMsg] = useState("");
const navigate = useNavigate();
const { id } = useParams();
useEffect(() => {
const getUserById = async () => {
try {
const response = await axios.get(`http://local
setName(response.data.name);
setEmail(response.data.email);
setRole(response.data.role);
} catch (error) {
if (error.response) {
setMsg(error.response.data.msg);
}
}
};
getUserById();
}, [id]);
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
no-unused-vars
Compiled with warnings.
[eslint]
src\components\Navbar.jsx
Line 10:11: 'user' is assigned a value but never used
no-unused-vars
src\components\Sidebar.jsx
node back...
node front...
BLACKBOX Chat
Add Logs
CyberCoder
Improve Code
user_management
trushsiaga
Search