Laporan Praktikum

Mata Kuliah Pemrograman Berorientasi Objek

AUTHENTICATION (Login Multi-Level)

Dosen Pengampu :

Willdan Aprizal Arifin S.pd M.kom

Disusun Oleh :

RIYADI TRI WALUYA BUDI

2308065

PROGRAM STUDI SISTEM INFORMASI KELAUTAN

UNIVERSITAS PENDIDIKAN INDONESIA

2024

I.PENDAHULUAN

Authentication adalah proses penting dalam aplikasi web untuk memastikan keamanan akses ke data dan fitur aplikasi. Dalam praktik ini, akan dibuat sistem autentikasi login berbasis peran menggunakan Node.js, Express, MySQL, dan React.js. Sistem ini memungkinkan pengguna dengan berbagai peran, seperti Admin, User, atau Manager, untuk mengakses fitur atau halaman yang sesuai dengan hak akses mereka.

Tujuan Praktikum:

1. Memahami Autentikasi dan Otorisasi:
    - Memahami perbedaan antara autentikasi (proses memverifikasi identitas pengguna) dan otorisasi (proses memberikan hak akses berdasarkan peran pengguna).
    - Mengimplementasikan autentikasi berbasis peran.
2. Integrasi Teknologi Full-Stack:
    - Backend: Menggunakan Node.js dan Express untuk menangani login serta manajemen pengguna.
    - Database: Menggunakan MySQL untuk menyimpan data pengguna dan peran.
    - Frontend: Menggunakan React.js untuk antarmuka login dan tampilan berbasis peran.
3. Penerapan Keamanan:
    - Mengenkripsi kata sandi menggunakan hashing (contoh: bcrypt).
    - Menggunakan JSON Web Token (JWT) untuk mengelola sesi pengguna secara aman.

Pendekatan ini menekankan praktik pengembangan aman dan modern untuk membangun sistem autentikasi yang andal dan berbasis peran.

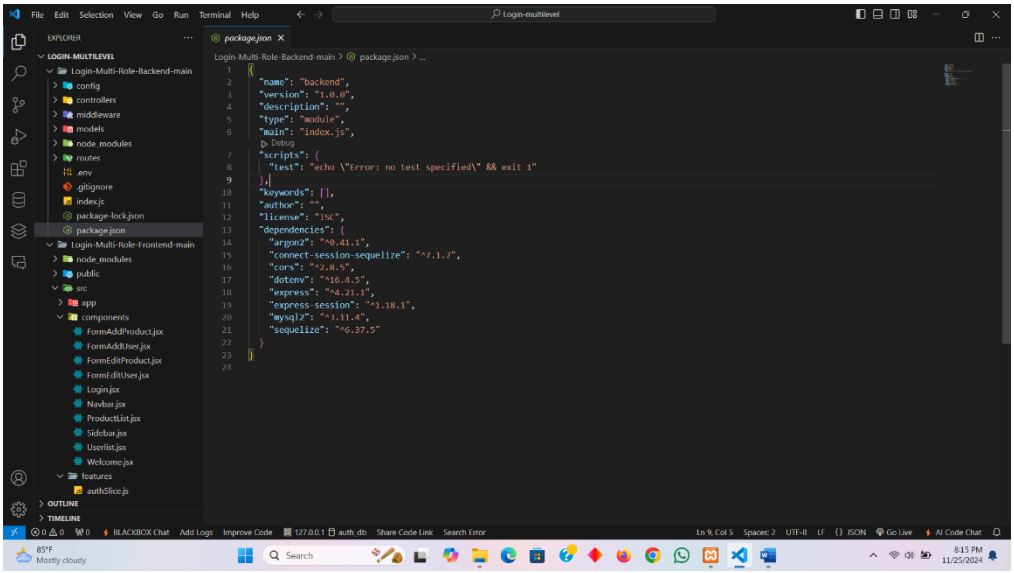
II. ALAT DAN BAHAN

- Node.js: Untuk membangun backend server.
- Express.js: Framework backend untuk menangani API dan middleware.
- MySQL: Database relasional untuk menyimpan data pengguna.
- React.js: Library frontend untuk membangun antarmuka pengguna.
- JWT (JSON Web Token): Untuk mengelola otorisasi berbasis token.
- bcrypt: Untuk mengenkripsi kata sandi.

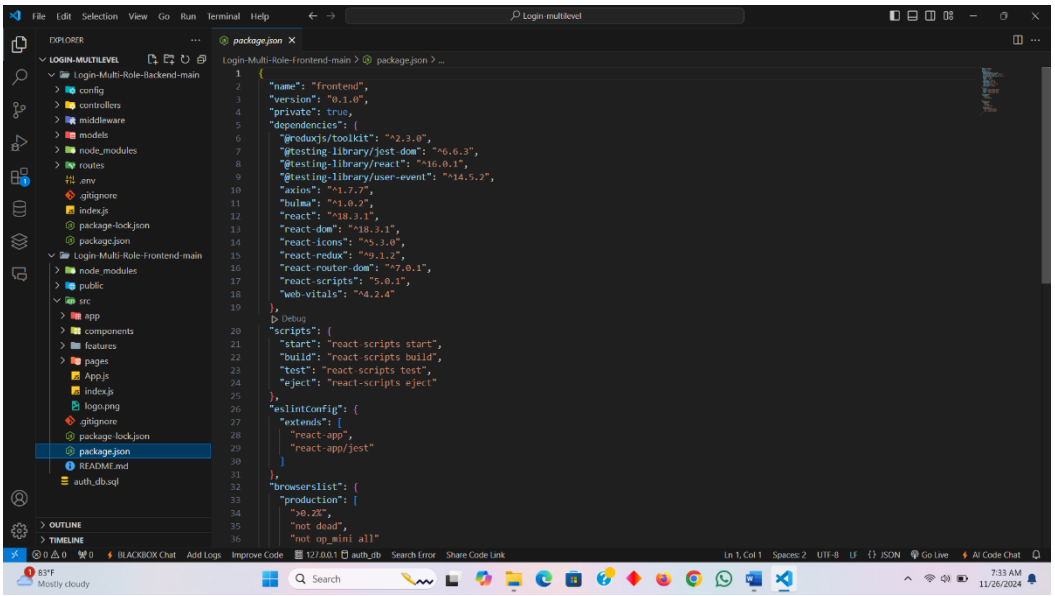
III. LANGKAH KERJA

1.Pertama ,Buka Visual Studio Code setelah itu install package.json dan hasilnya seperti di bawah ini

-Backend

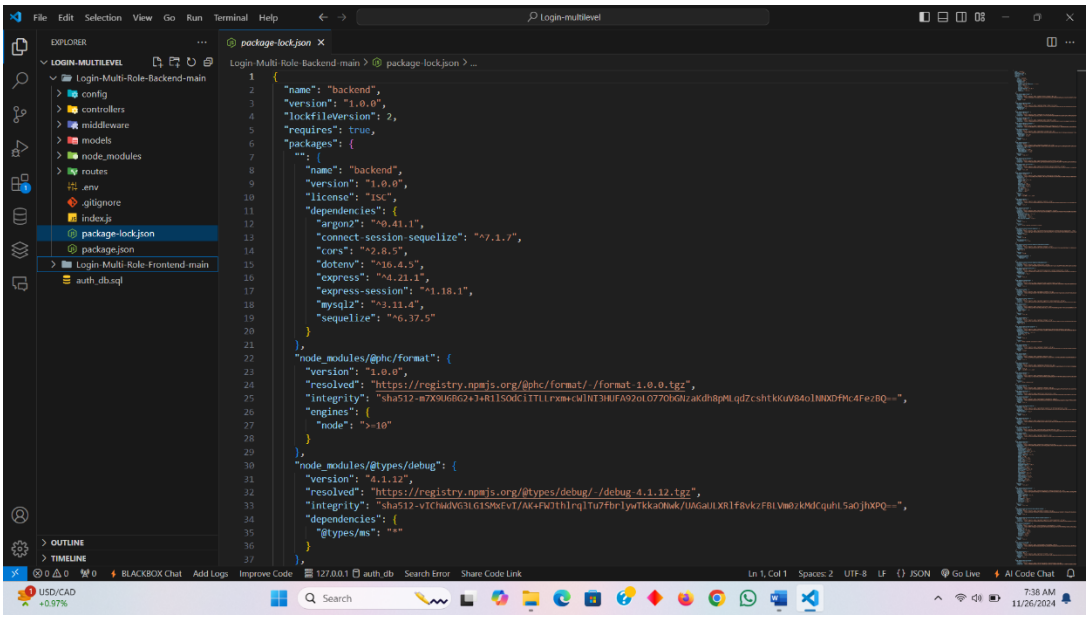


-Frontend

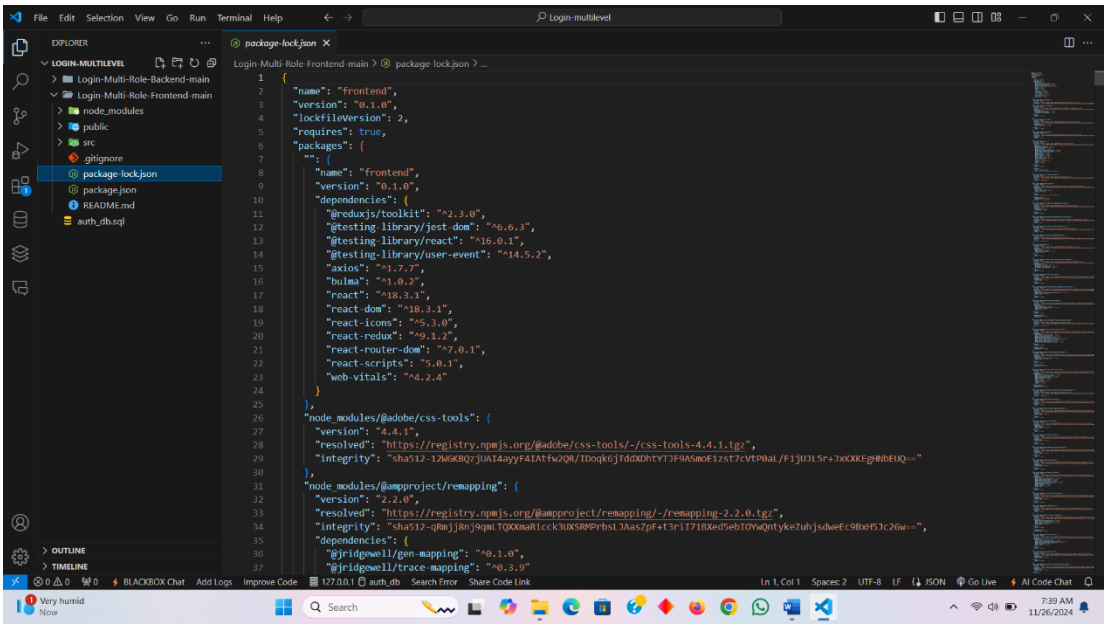


2. Kedua, Install package-lock.json untuk backend dan frontend dan hasilnya seperti di bawah ini

-Backend

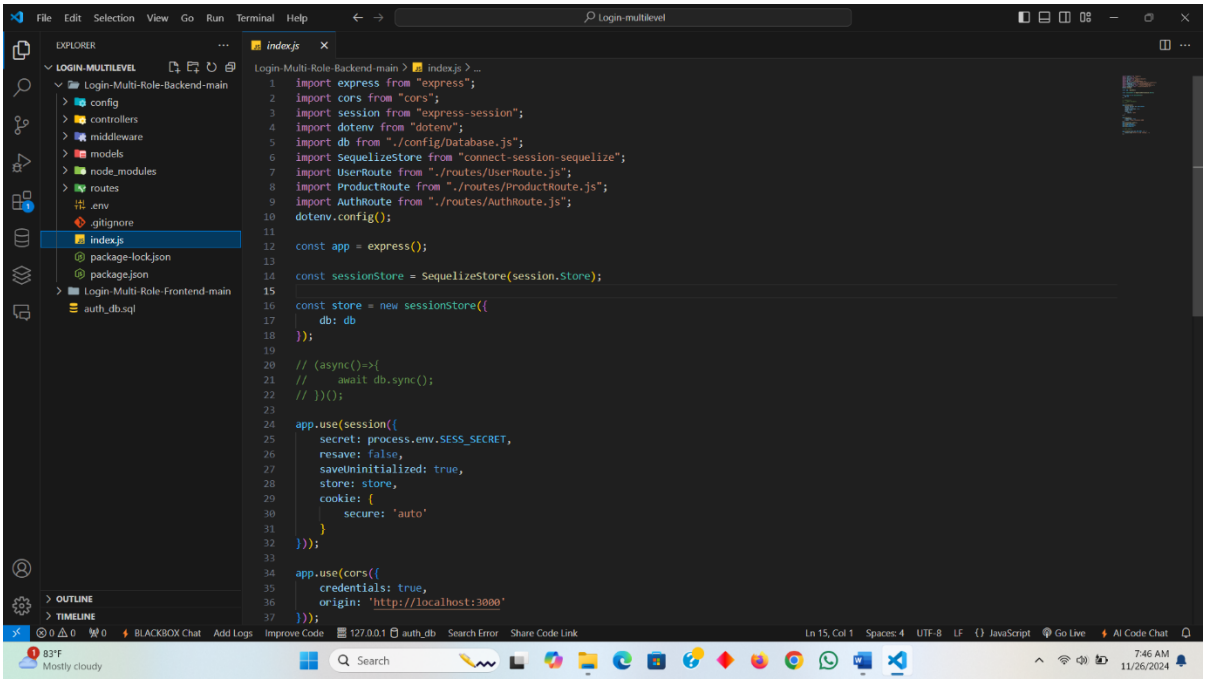


-Frontend

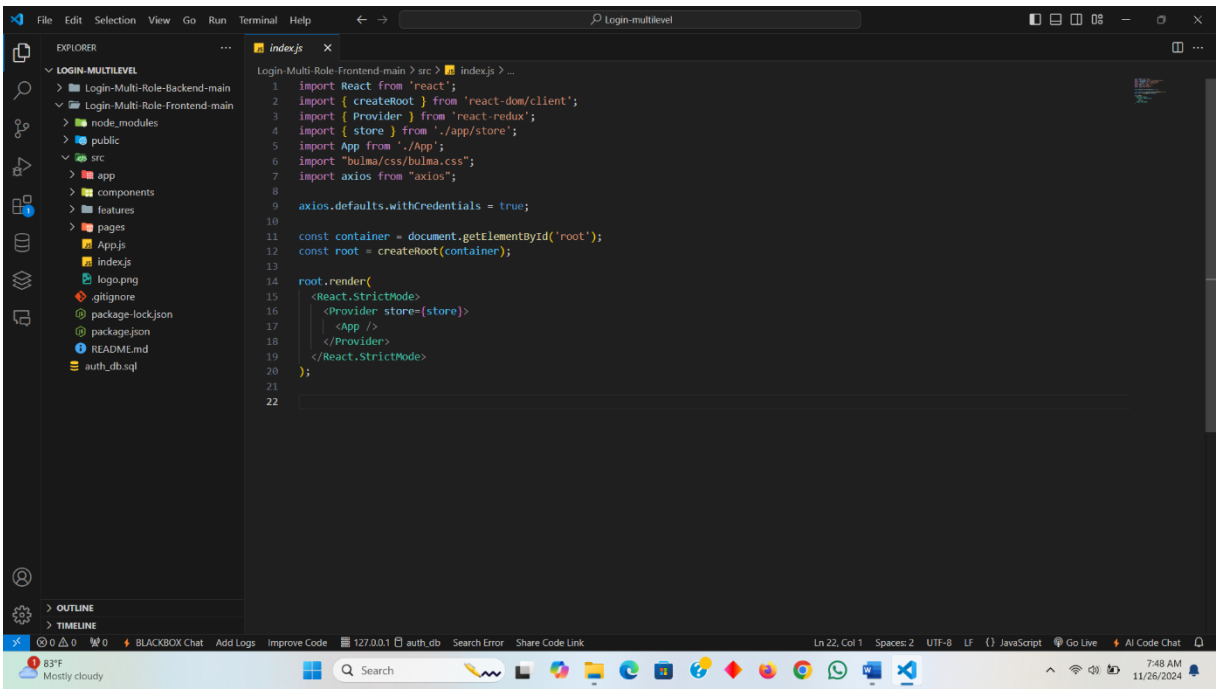


3.Ketiga, membuat file baru “index.js” untuk backend dan frontend dan isinya seperti gambar di bawah ini

-Backend



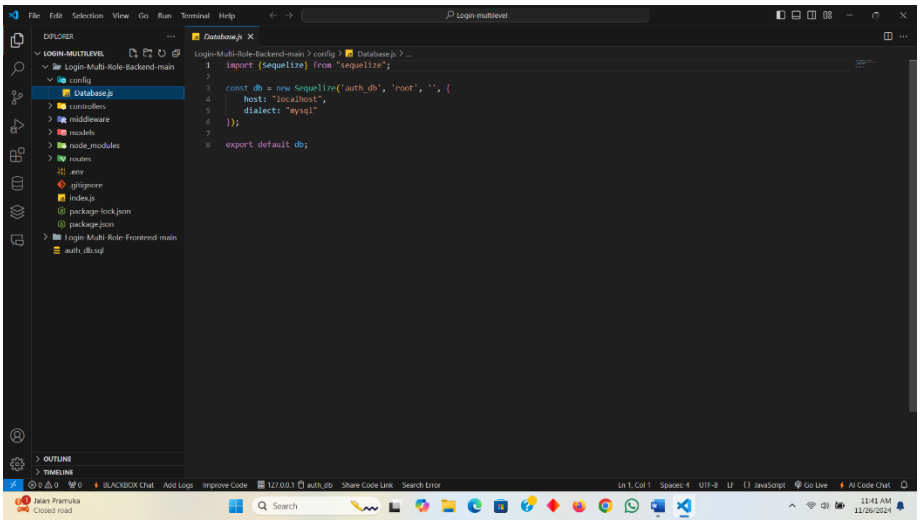
-Frontend



4.Keempat, Di dalam folder “Login-Multi-Role-Backend-main” terdapat folder config,controllers,middleware,models,node_modules,routes,env dan gitignore yang isinya seperti gambar di bawah ini.

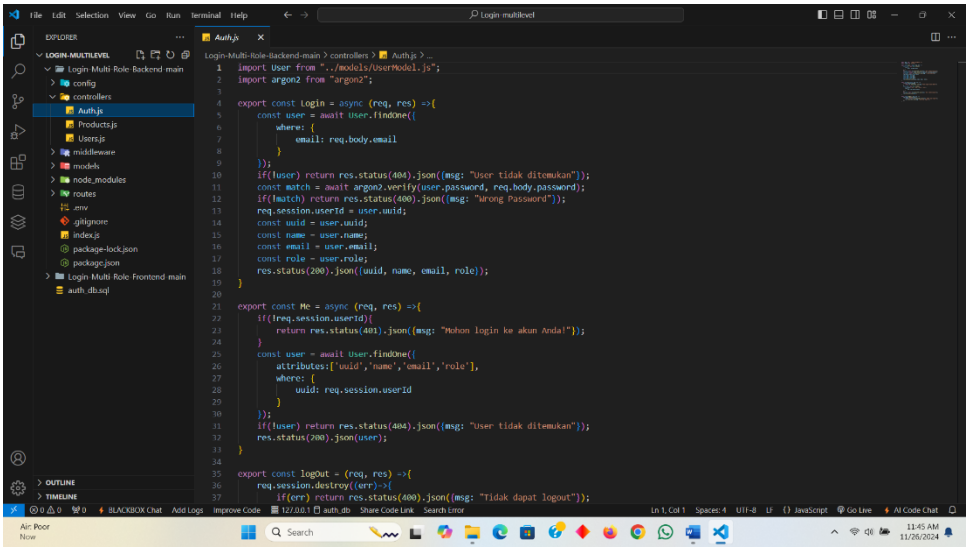
*Config

-Database.js

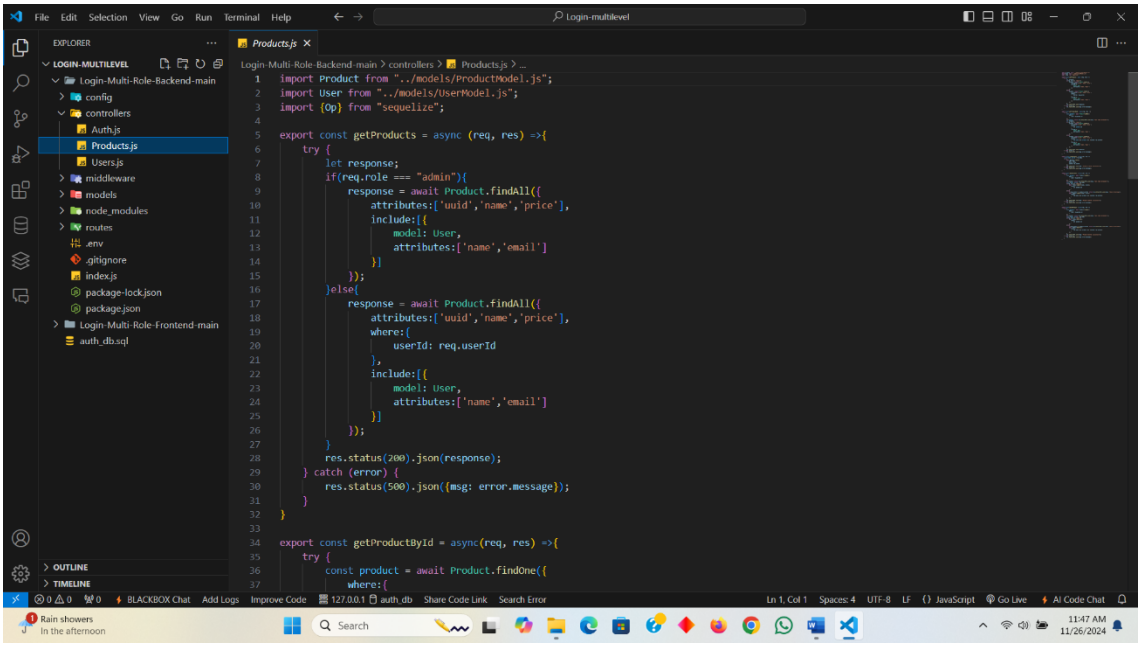


*Controllers

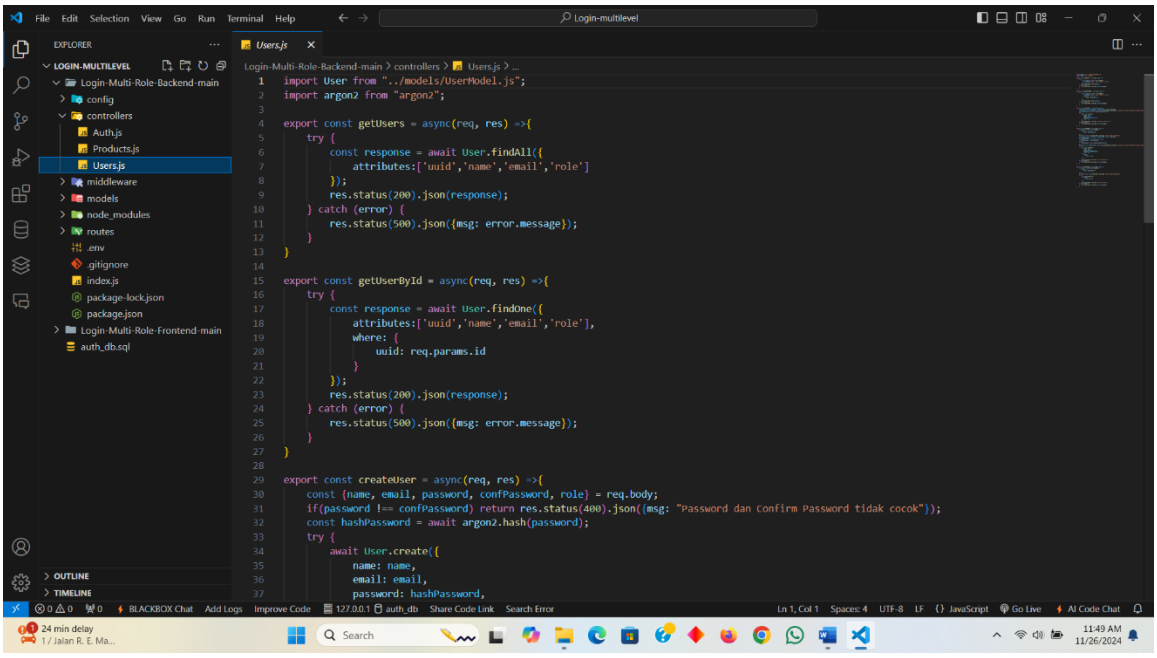
-Auth.js



-Products.js

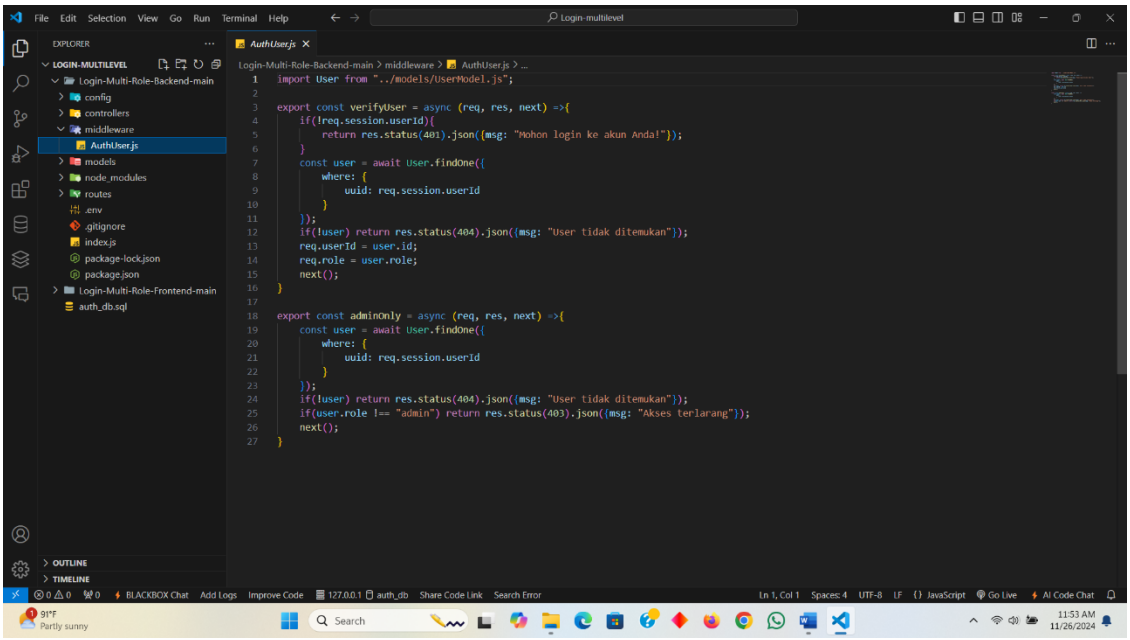


-Users.js



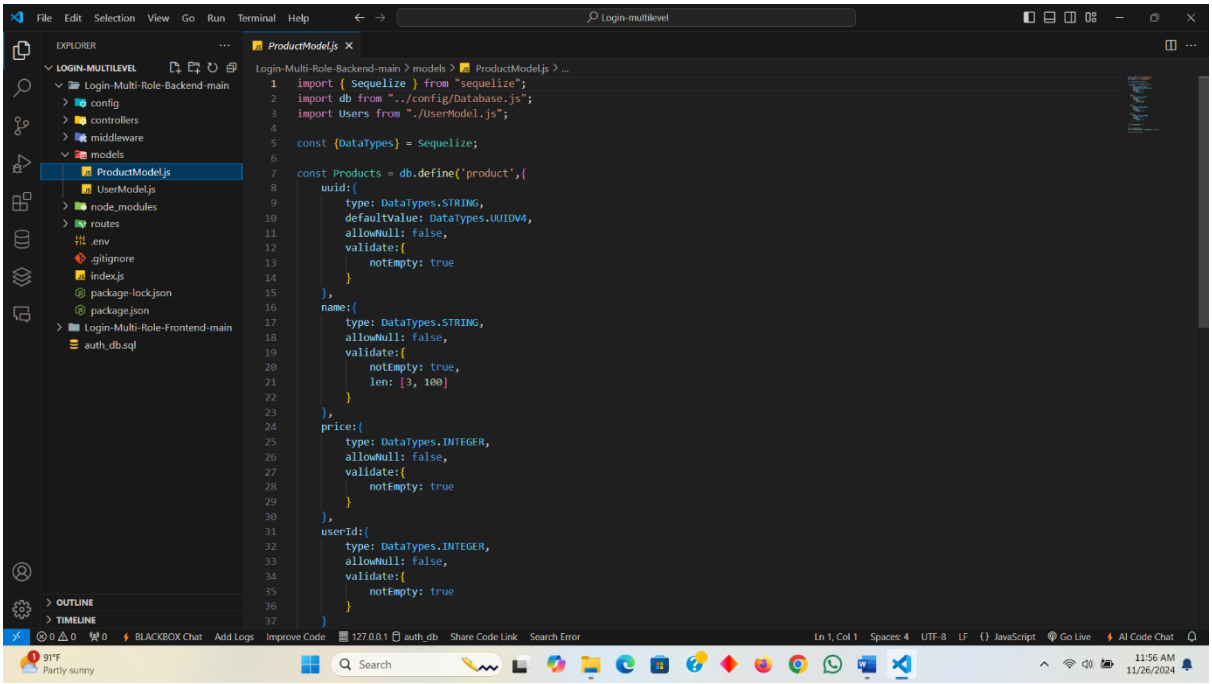
*Middleware

-AuthUser.js

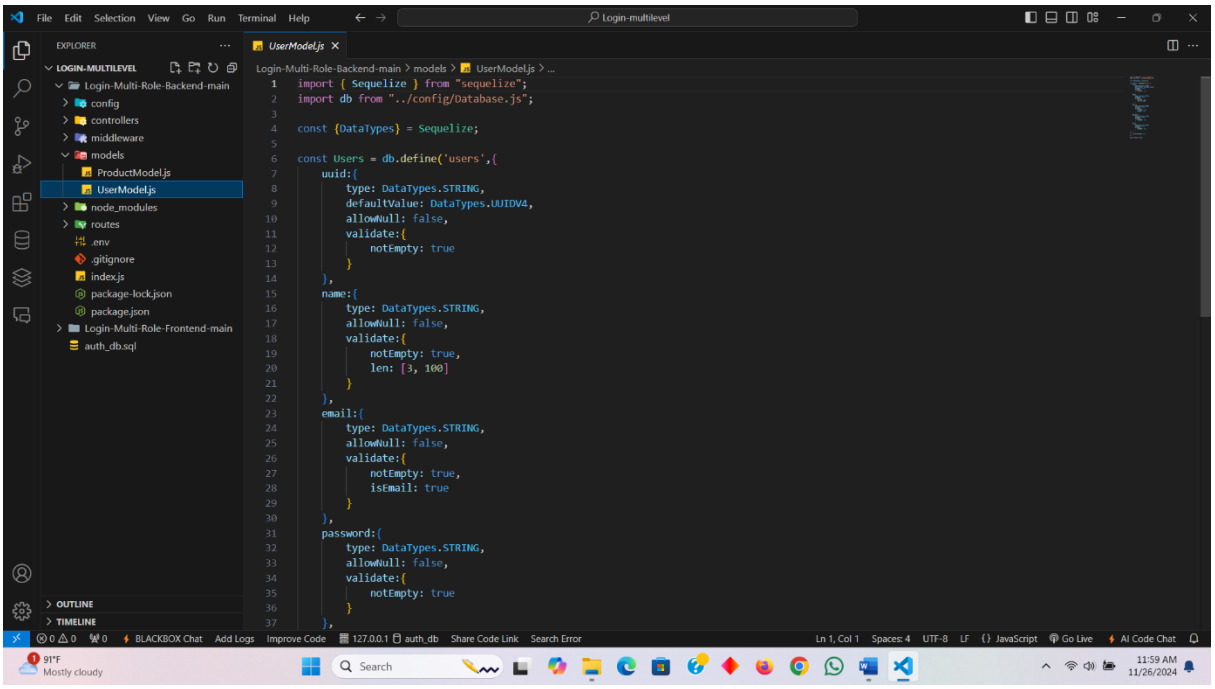


*Models

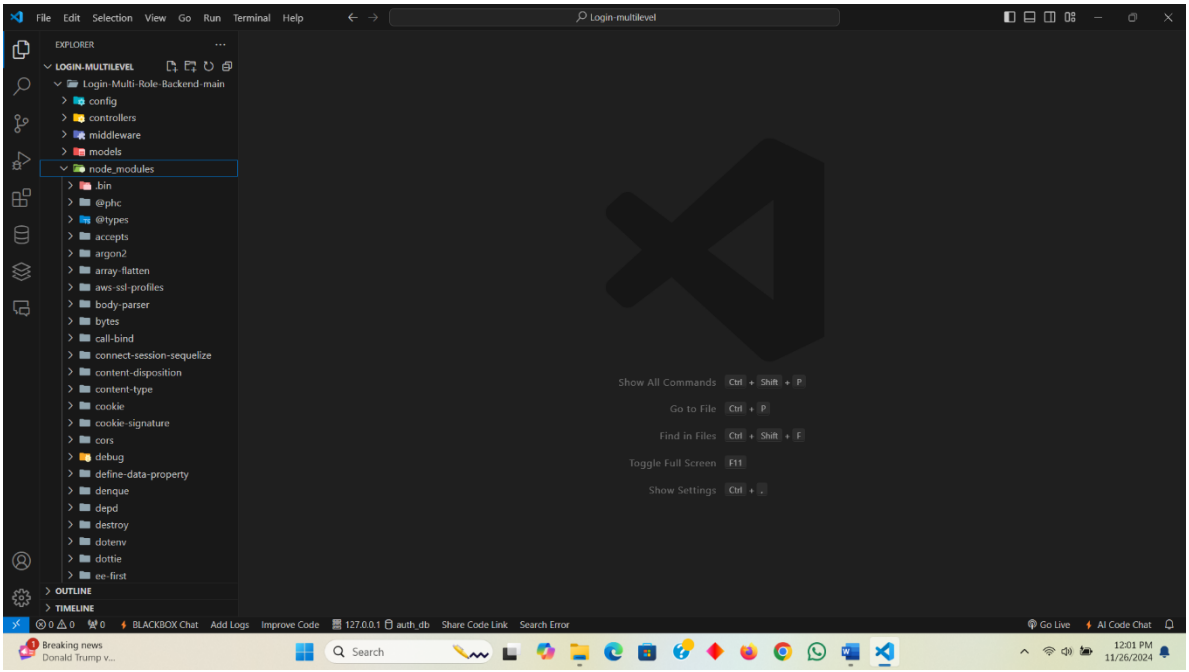
-ProductModel.js



-UserModel.js

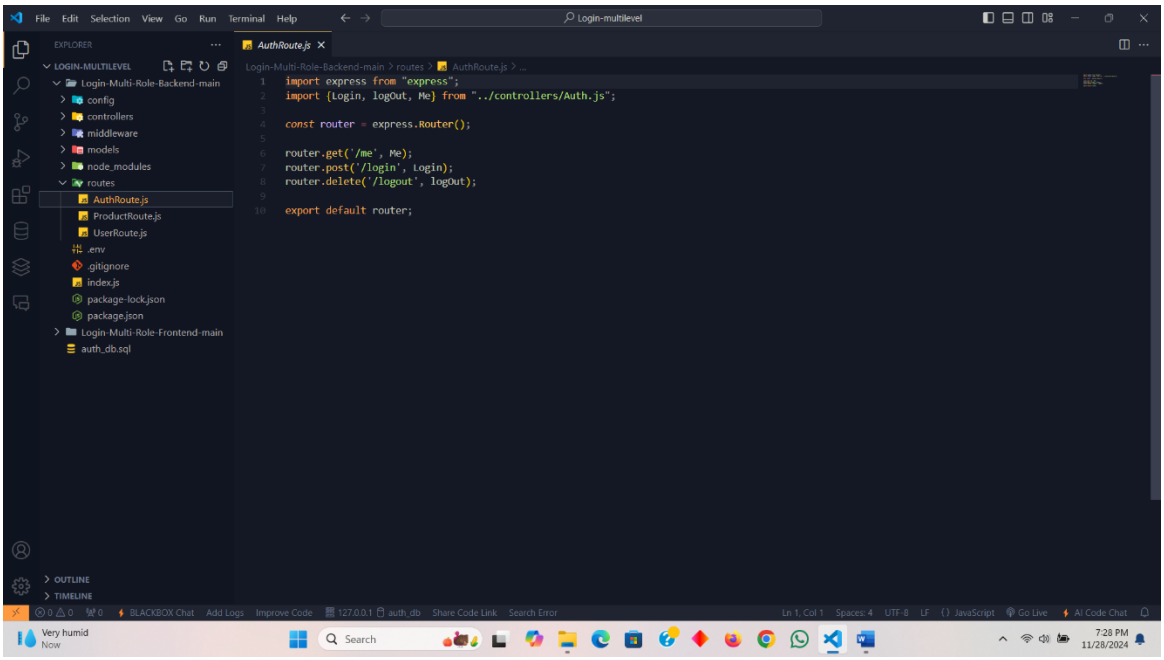


*Node_modules

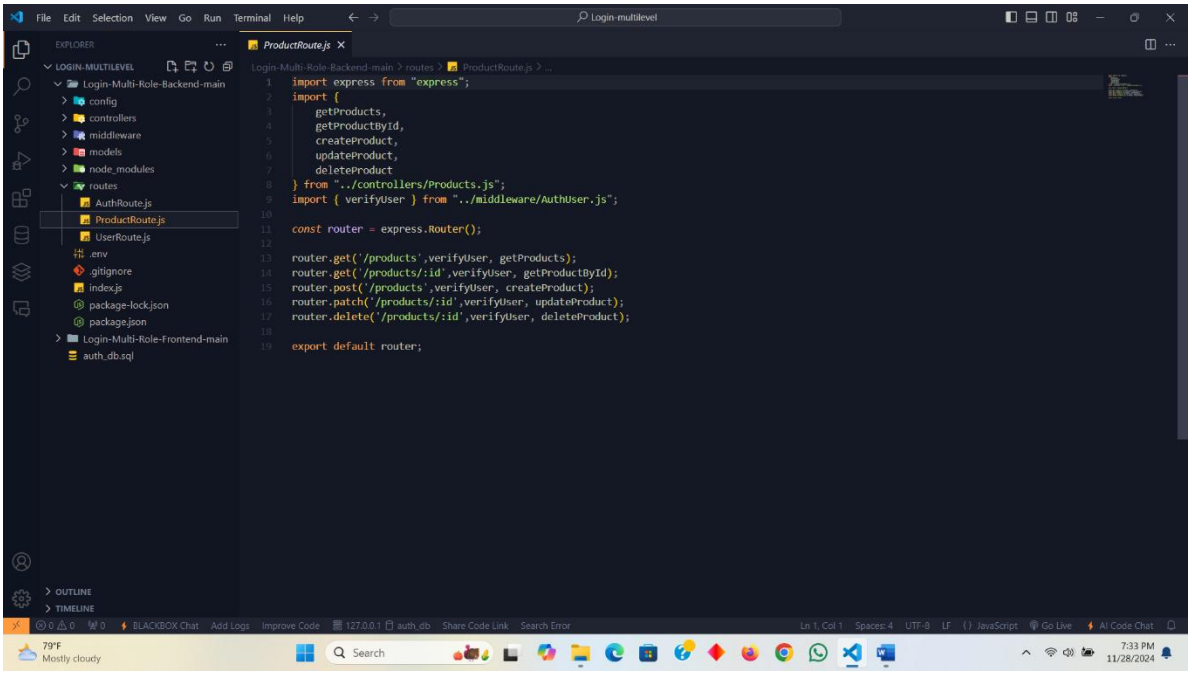


*Routes

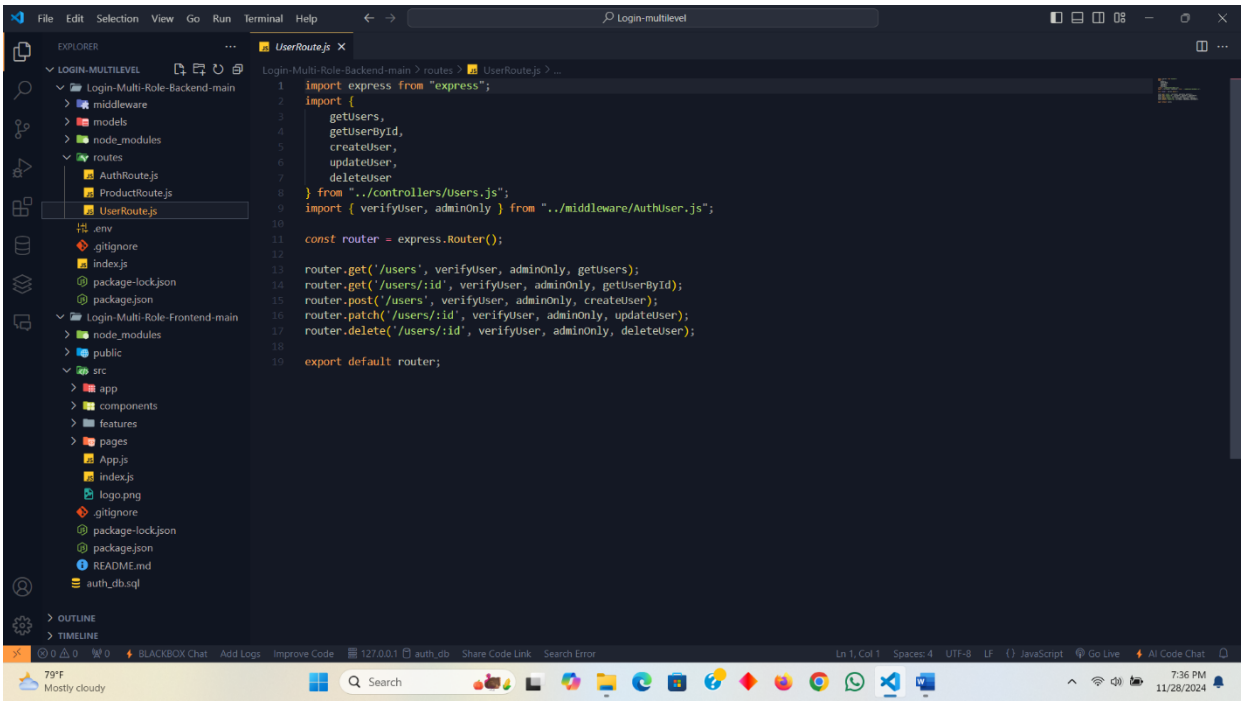
-AuthRoute.js



-ProductRoute.js

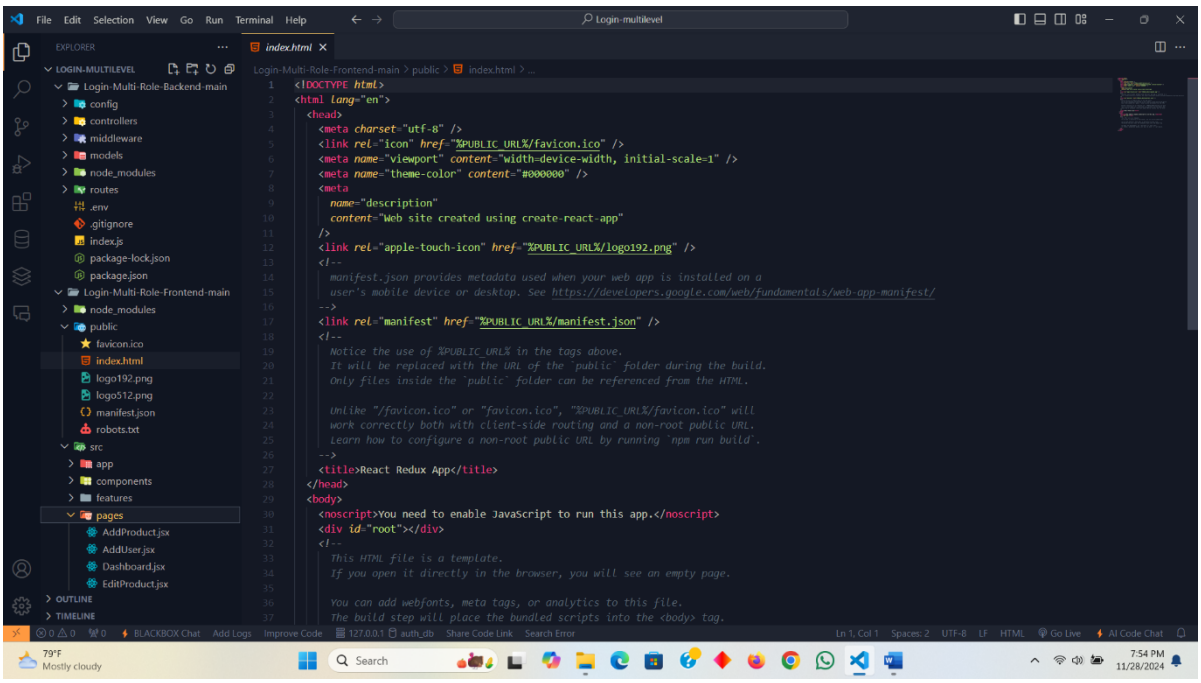


-UserRoute.js



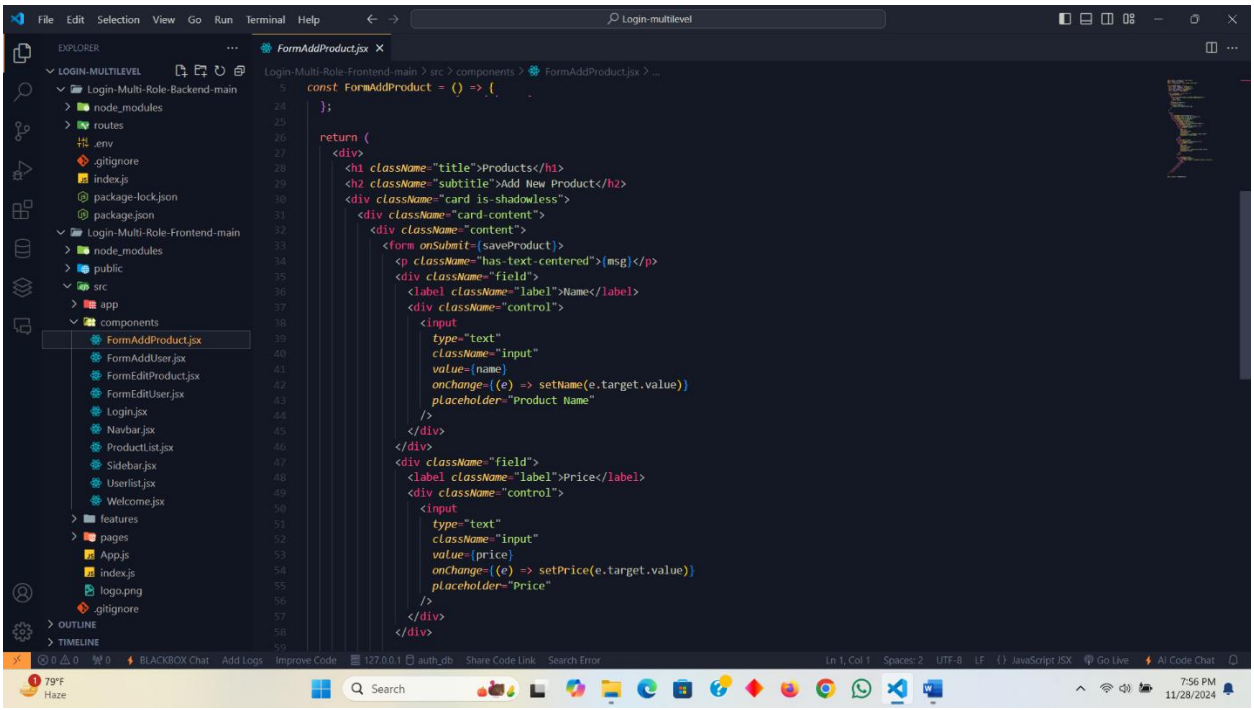
*Public

-index.html



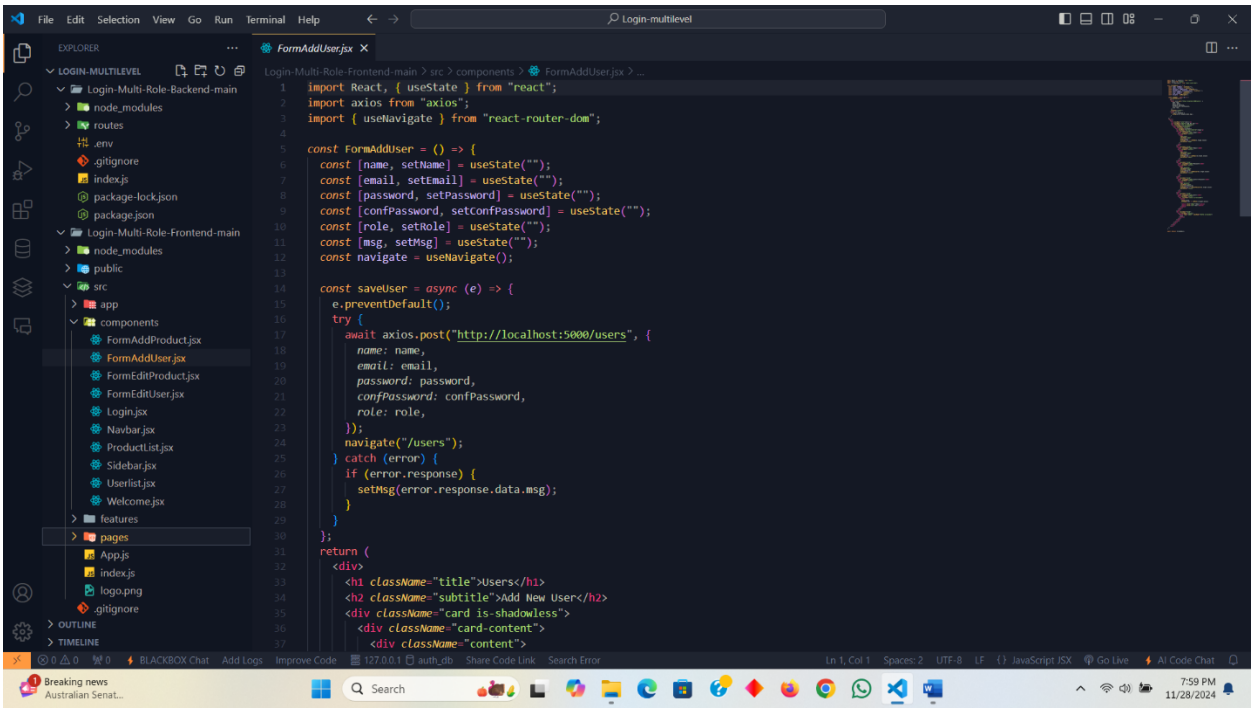
*Components

-FormAddProduct.jsx



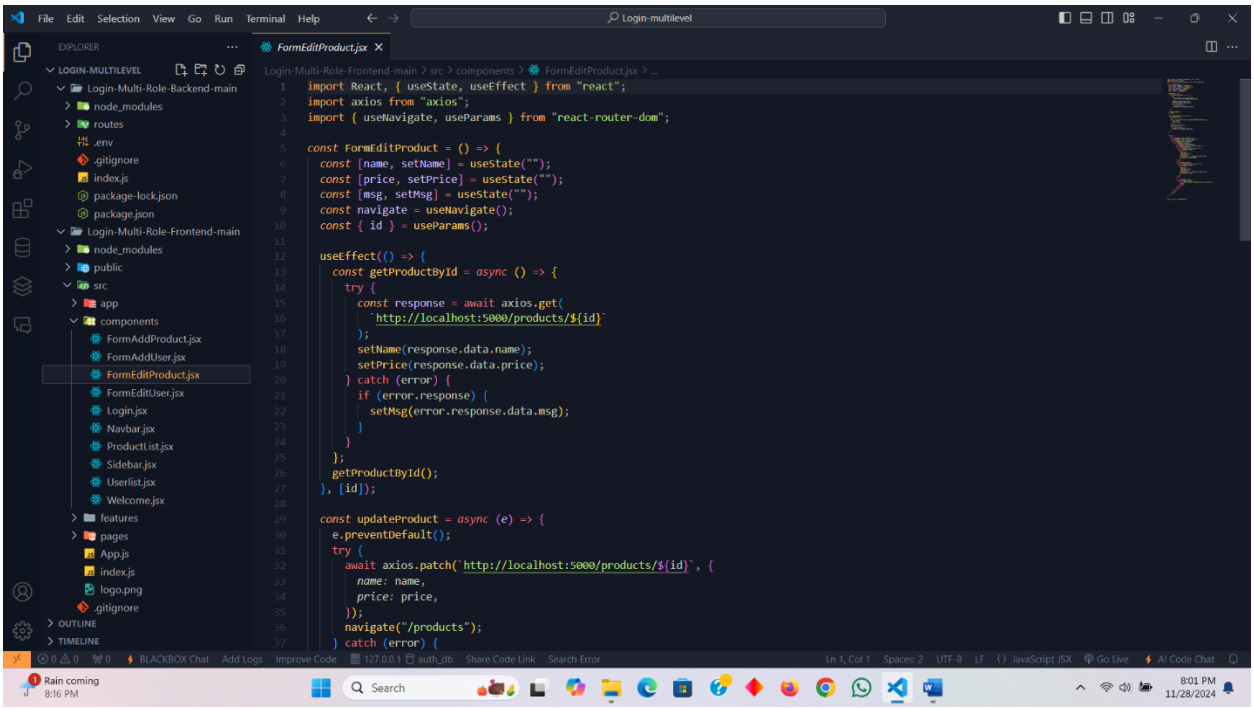
```jsx
const FormAddProduct = () => {
  };

  return (
    <div>
      <h1 className="title">Products</h1>
      <h2 className="subtitle">Add New Product</h2>
      <div className="card is-shadowless">
        <div className="card-content">
          <div className="content">
            <form onSubmit={saveProduct}>
              <p className="has-text-centered">{msg}</p>
              <div className="field">
                <label className="label">Name</label>
                <div className="control">
                  <input
                    type="text"
                    className="input"
                    value={name}
                    onChange={(e) => setName(e.target.value)}
                    placeholder="Product Name"
                  />
                </div>
              </div>
              <div className="field">
                <label className="label">Price</label>
                <div className="control">
                  <input
                    type="text"
                    className="input"
                    value={price}
                    onChange={(e) => setPrice(e.target.value)}
                    placeholder="Price"
                  />
                </div>
              </div>
```

-FormAddUser.jsx

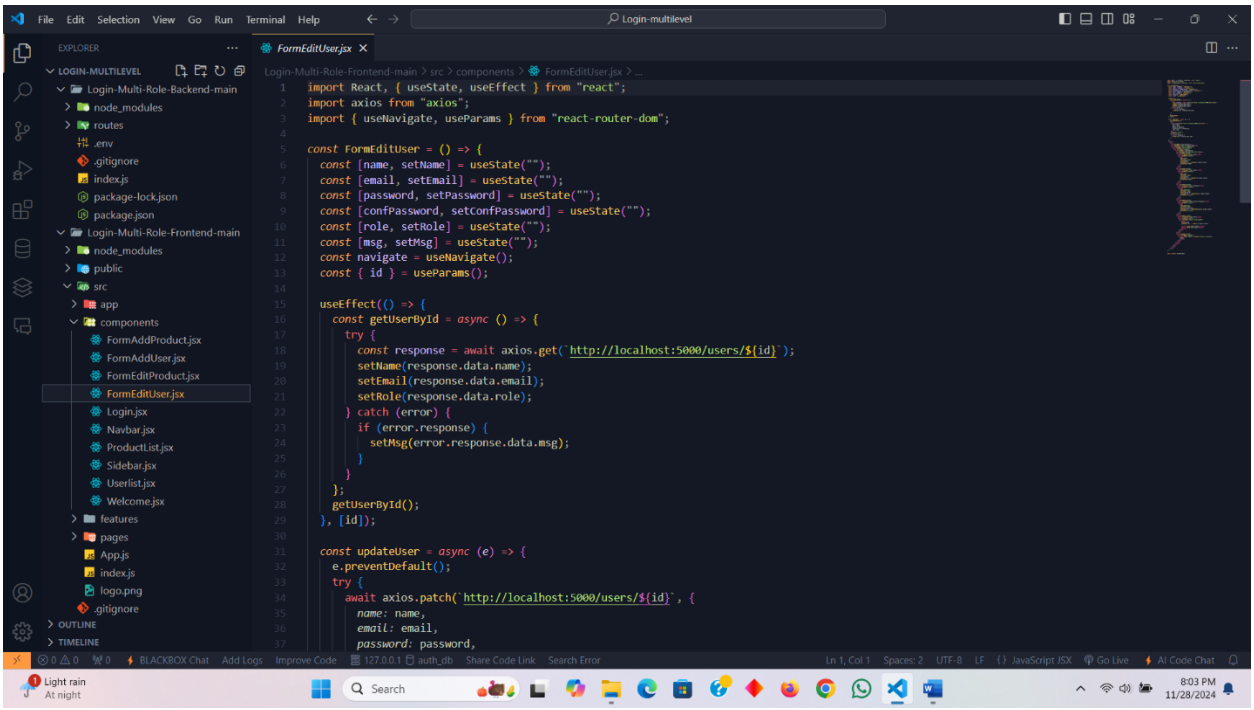


```jsx
import React, { useState } from "react";
import axios from "axios";
import { useNavigate } from "react-router-dom";

const FormAddUser = () => {
  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 saveUser = async (e) => {
    e.preventDefault();
    try {
      await axios.post("http://localhost:5000/users", {
        name: name,
        email: email,
        password: password,
        confPassword: confPassword,
        role: role,
      });
      navigate("/users");
    } catch (error) {
      if (error.response) {
        setMsg(error.response.data.msg);
      }
    }
  };
  return (
    <div>
      <h1 className="title">Users</h1>
      <h2 className="subtitle">Add New User</h2>
      <div className="card is-shadowless">
        <div className="card-content">
          <div className="content">
```

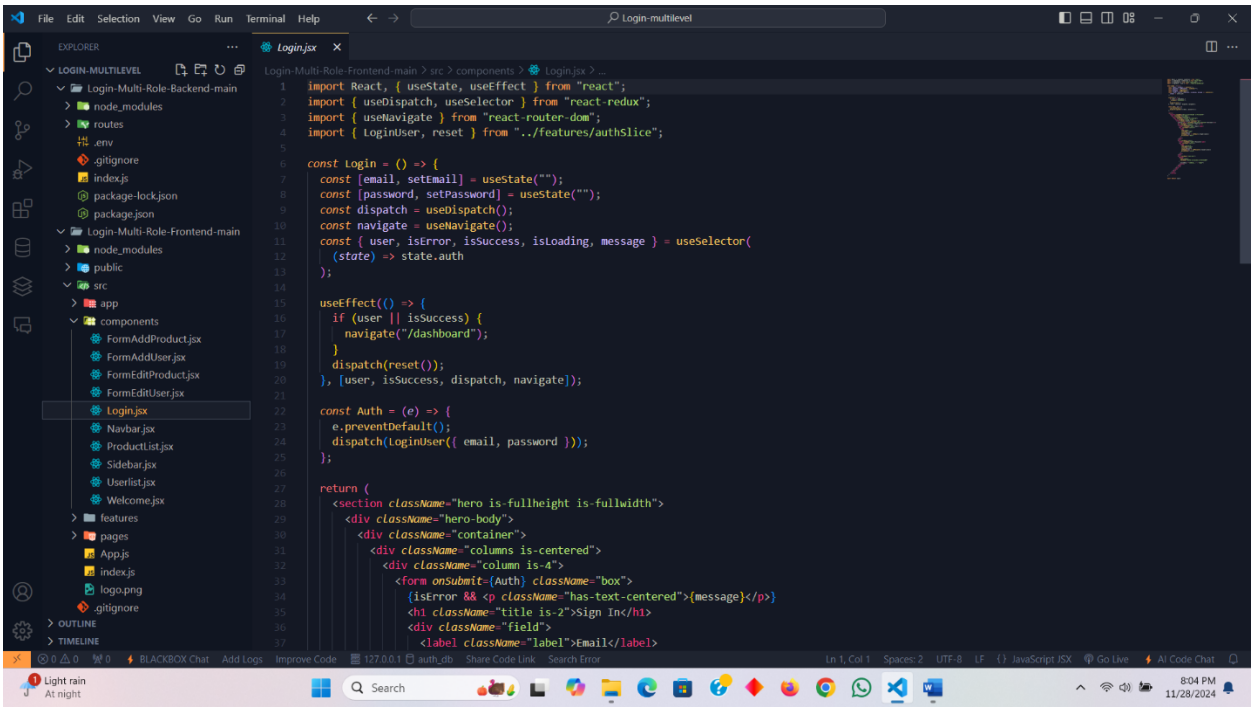
-FormEditProduct.jsx



-FormEditUser.jsx

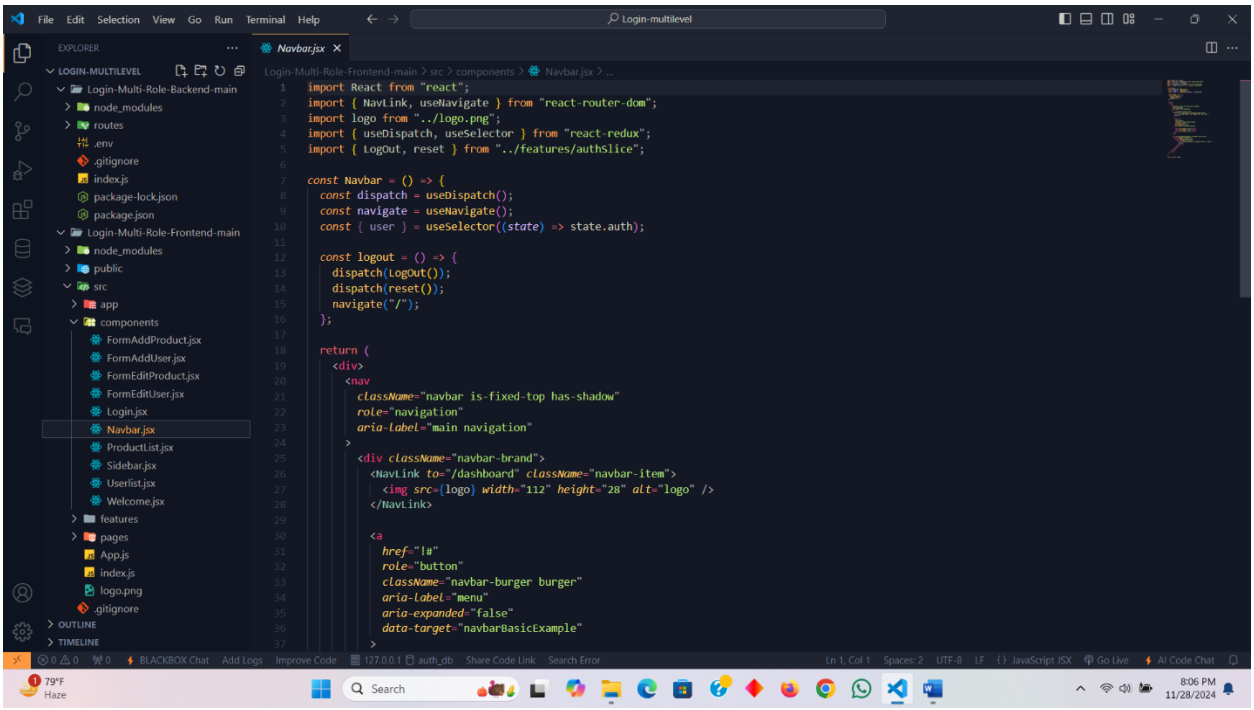


-Login.jsx



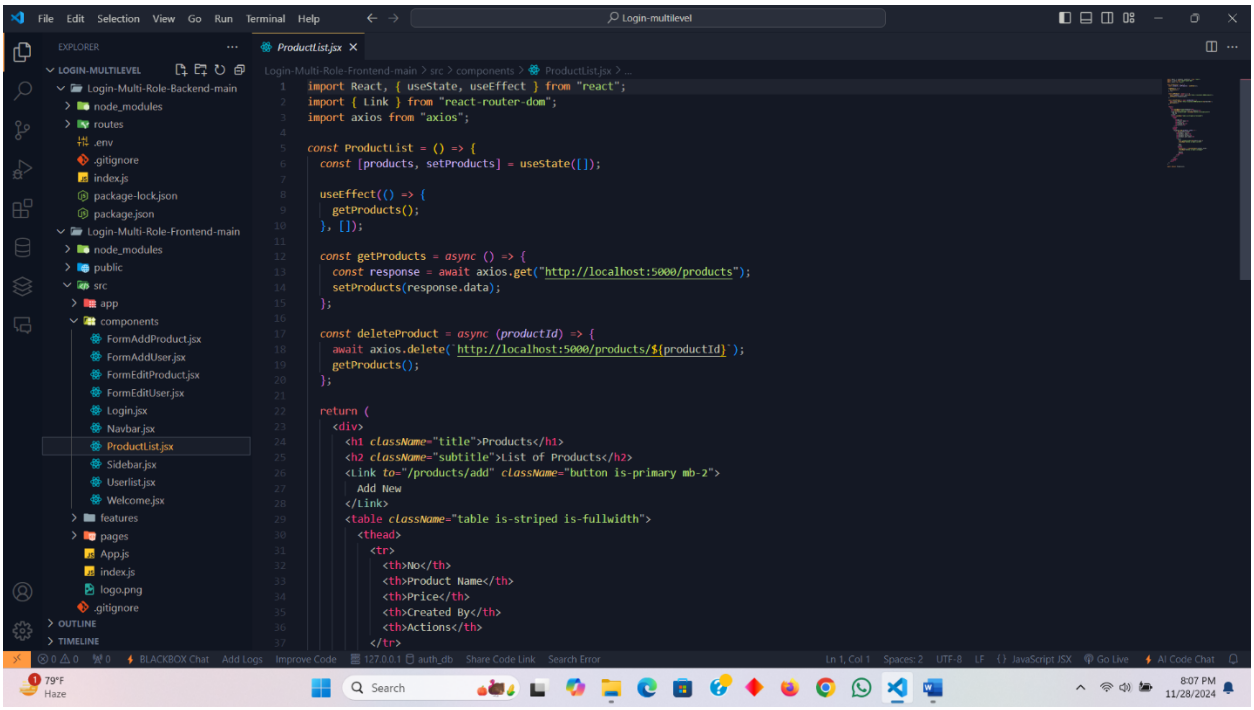
```jsx
import React, { useState, useEffect } from "react";
import { useDispatch, useSelector } from "react-redux";
import { useNavigate } from "react-router-dom";
import { LoginUser, reset } from "../features/authSlice";

const Login = () => {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const dispatch = useDispatch();
  const navigate = useNavigate();
  const { user, isError, isSuccess, isLoading, message } = useSelector(
    (state) => state.auth
  );

  useEffect(() => {
    if (user || isSuccess) {
      navigate("/dashboard");
    }
    dispatch(reset());
  }, [user, isSuccess, dispatch, navigate]);

  const Auth = (e) => {
    e.preventDefault();
    dispatch(LoginUser({ email, password }));
  };

  return (
    <section className="hero is-fullheight is-fullwidth">
      <div className="hero-body">
        <div className="container">
          <div className="columns is-centered">
            <div className="column is-4">
              <form onSubmit={Auth} className="box">
                {isError && <p className="has-text-centered">{message}</p>}
                <h1 className="title is-2">Sign In</h1>
                <div className="field">
                  <label className="label">Email</label>
```

-Navbar.jsx



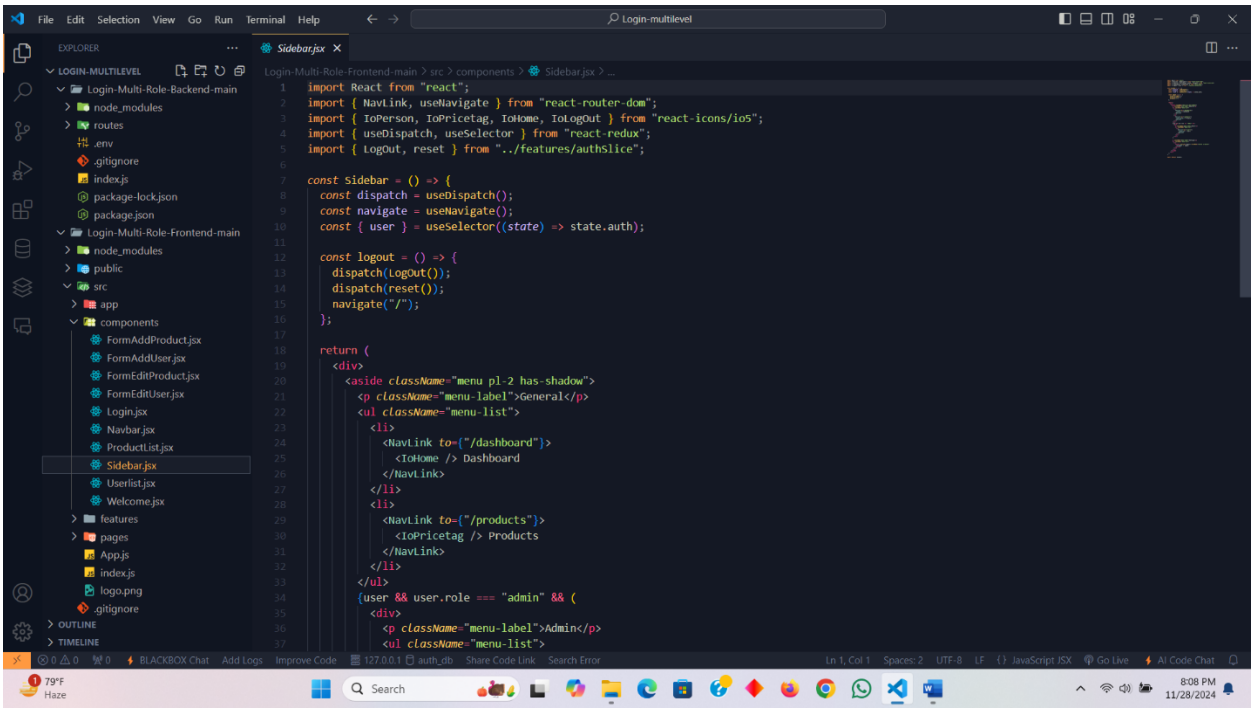
```jsx
import React from "react";
import { NavLink, useNavigate } from "react-router-dom";
import logo from "../logo.png";
import { useDispatch, useSelector } from "react-redux";
import { LogOut, reset } from "../features/authSlice";

const Navbar = () => {
  const dispatch = useDispatch();
  const navigate = useNavigate();
  const { user } = useSelector((state) => state.auth);

  const logout = () => {
    dispatch(LogOut());
    dispatch(reset());
    navigate("/");
  };

  return (
    <div>
      <nav
        className="navbar is-fixed-top has-shadow"
        role="navigation"
        aria-label="main navigation"
      >
        <div className="navbar-brand">
          <NavLink to="/dashboard" className="navbar-item">
            <img src={logo} width="112" height="28" alt="logo" />
          </NavLink>

          <a
            href="!#"
            role="button"
            className="navbar-burger burger"
            aria-label="menu"
            aria-expanded="false"
            data-target="navbarBasicExample"
          >
```

-ProductList.jsx



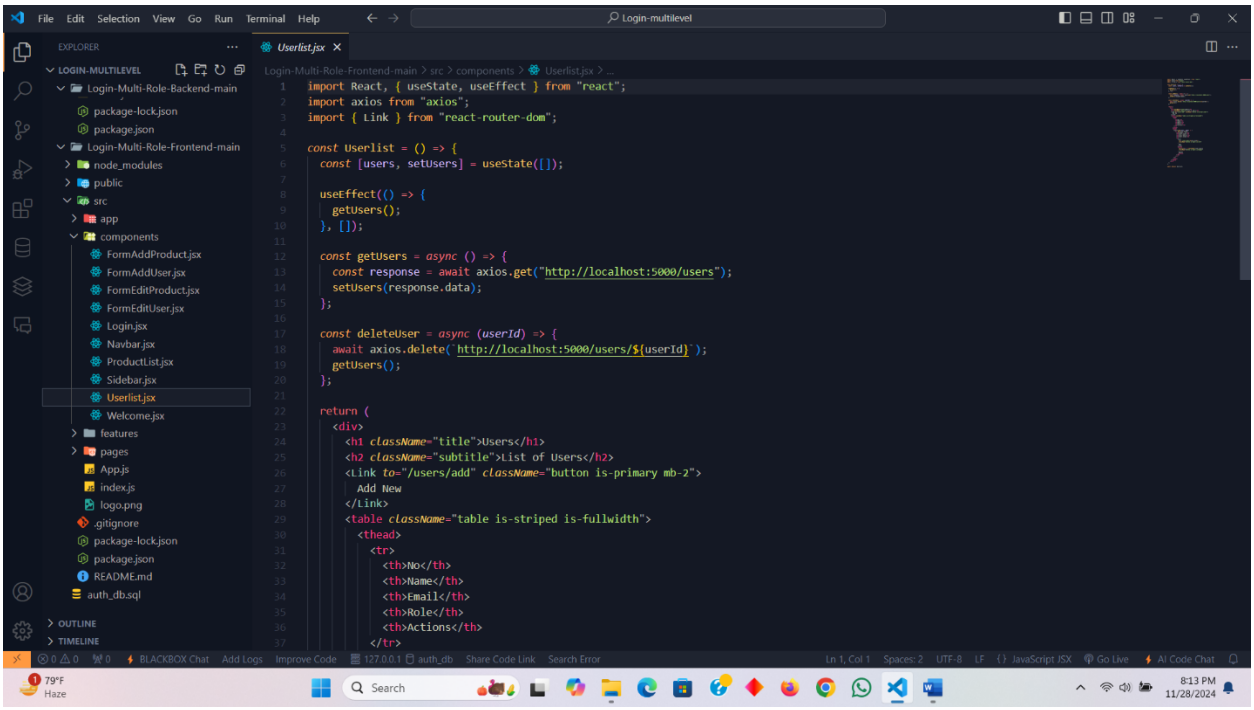
```jsx
import React, { useState, useEffect } from "react";
import { Link } from "react-router-dom";
import axios from "axios";

const ProductList = () => {
  const [products, setProducts] = useState([]);

  useEffect(() => {
    getProducts();
  }, []);

  const getProducts = async () => {
    const response = await axios.get("http://localhost:5000/products");
    setProducts(response.data);
  };

  const deleteProduct = async (productId) => {
    await axios.delete(`http://localhost:5000/products/${productId}`);
    getProducts();
  };

  return (
    <div>
      <h1 className="title">Products</h1>
      <h2 className="subtitle">List of Products</h2>
      <Link to="/products/add" className="button is-primary mb-2">
        Add New
      </Link>
      <table className="table is-striped is-fullwidth">
        <thead>
          <tr>
            <th>No</th>
            <th>Product Name</th>
            <th>Price</th>
            <th>Created By</th>
            <th>Actions</th>
          </tr>
```

-Sidebar.jsx

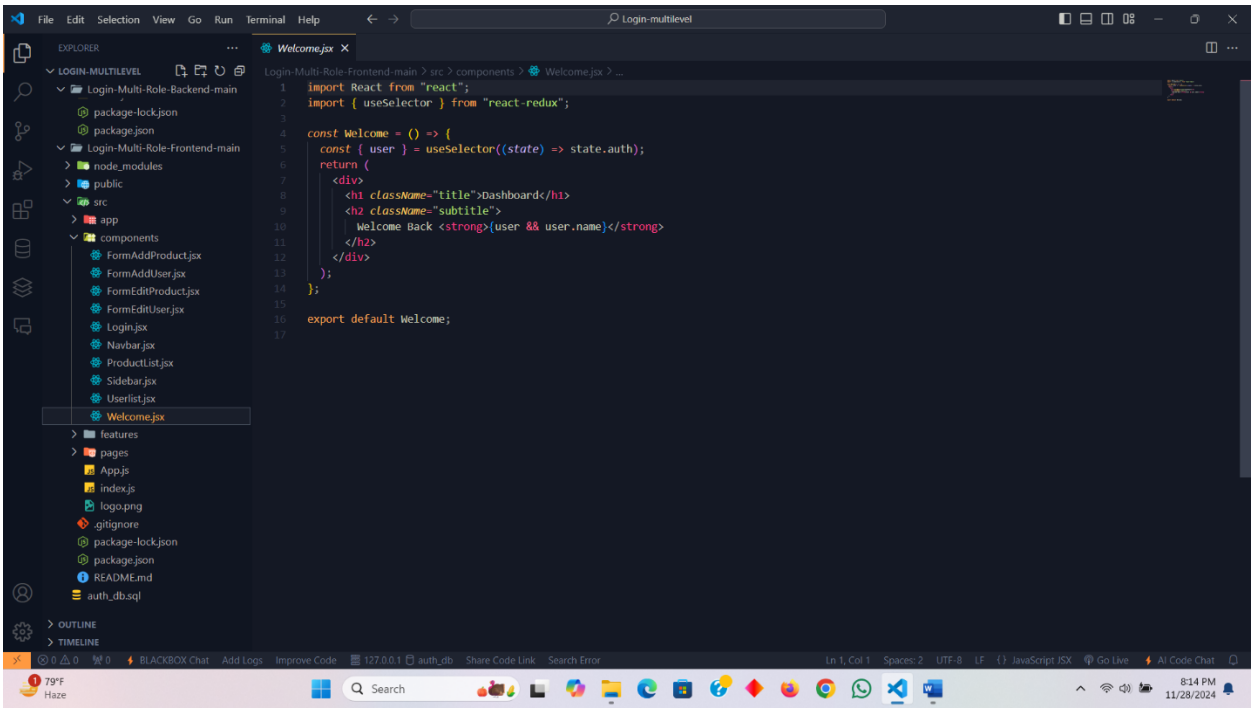


```jsx
import React from "react";
import { NavLink, useNavigate } from "react-router-dom";
import { IoPerson, IoPricetag, IoHome, IoLogOut } from "react-icons/io5";
import { useDispatch, useSelector } from "react-redux";
import { LogOut, reset } from "../features/authSlice";

const Sidebar = () => {
  const dispatch = useDispatch();
  const navigate = useNavigate();
  const { user } = useSelector((state) => state.auth);

  const logout = () => {
    dispatch(LogOut());
    dispatch(reset());
    navigate("/");
  };

  return (
    <div>
      <aside className="menu pl-2 has-shadow">
        <p className="menu-label">General</p>
        <ul className="menu-list">
          <li>
            <NavLink to={"/dashboard"}>
              <IoHome /> Dashboard
            </NavLink>
          </li>
          <li>
            <NavLink to={"/products"}>
              <IoPricetag /> Products
            </NavLink>
          </li>
        </ul>
        {user && user.role === "admin" && (
          <div>
            <p className="menu-label">Admin</p>
            <ul className="menu-list">
```

-Userlist.jsx

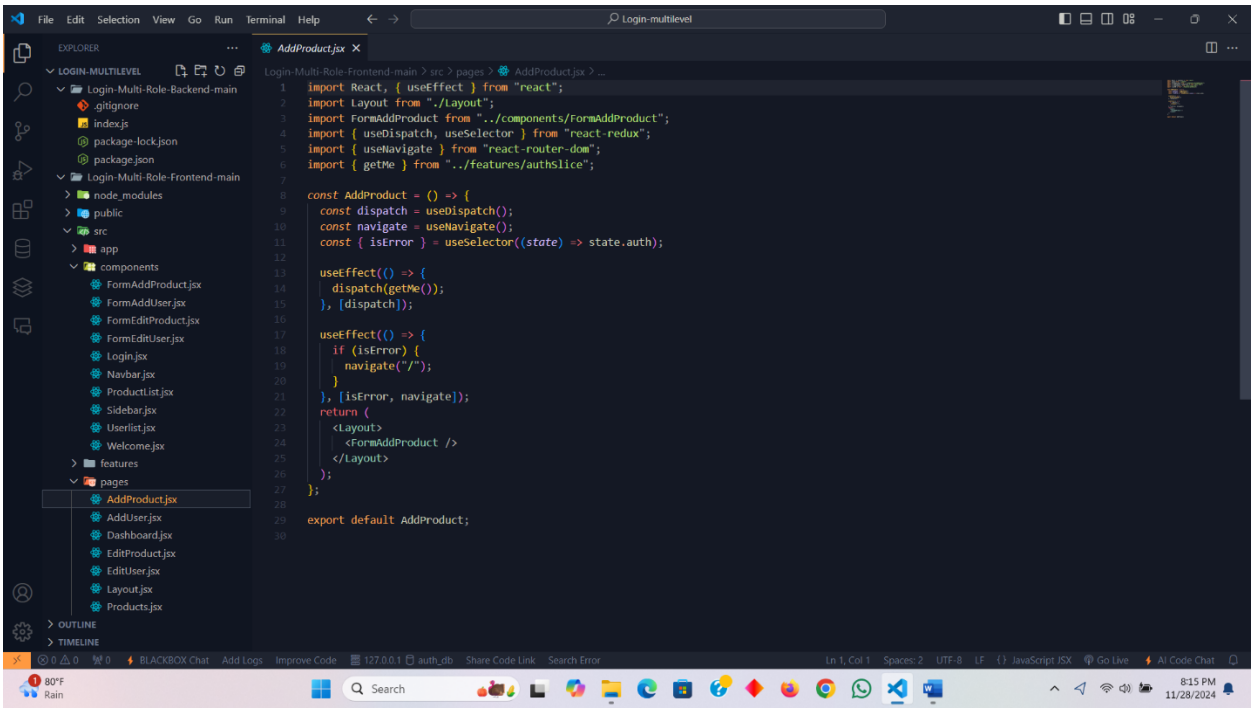


-Welcome.jsx



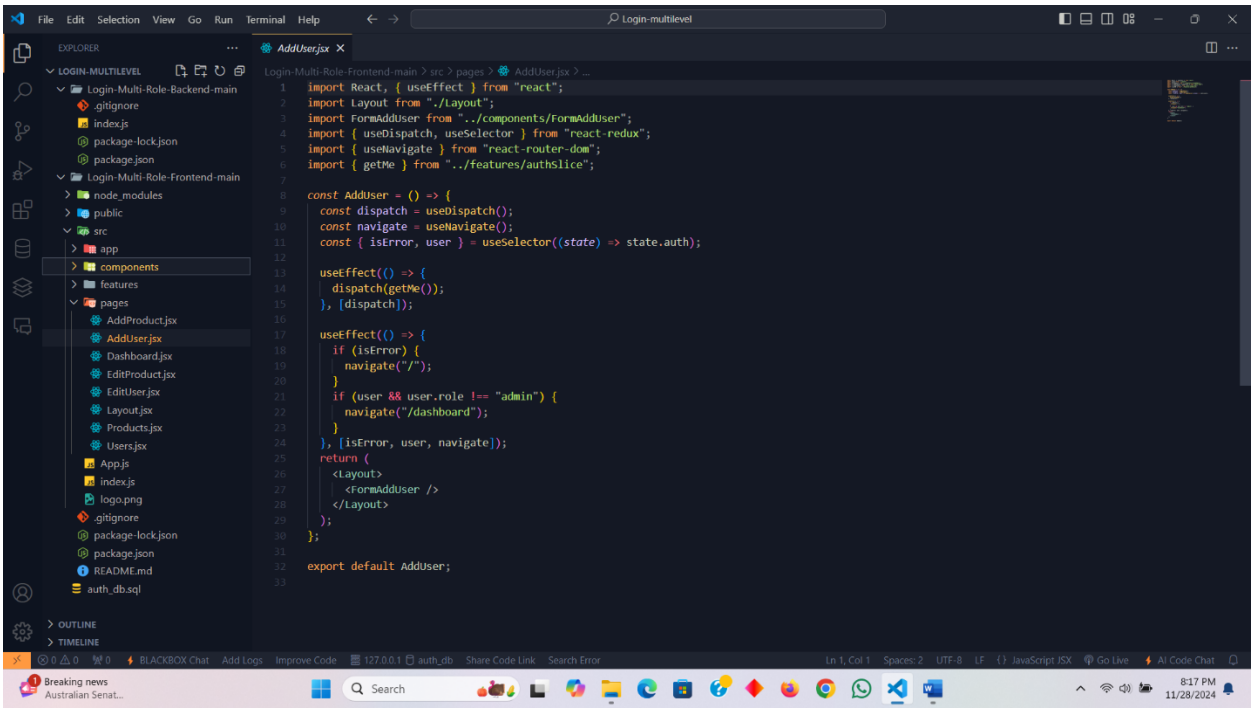
*Pages

-AddProduct.jsx



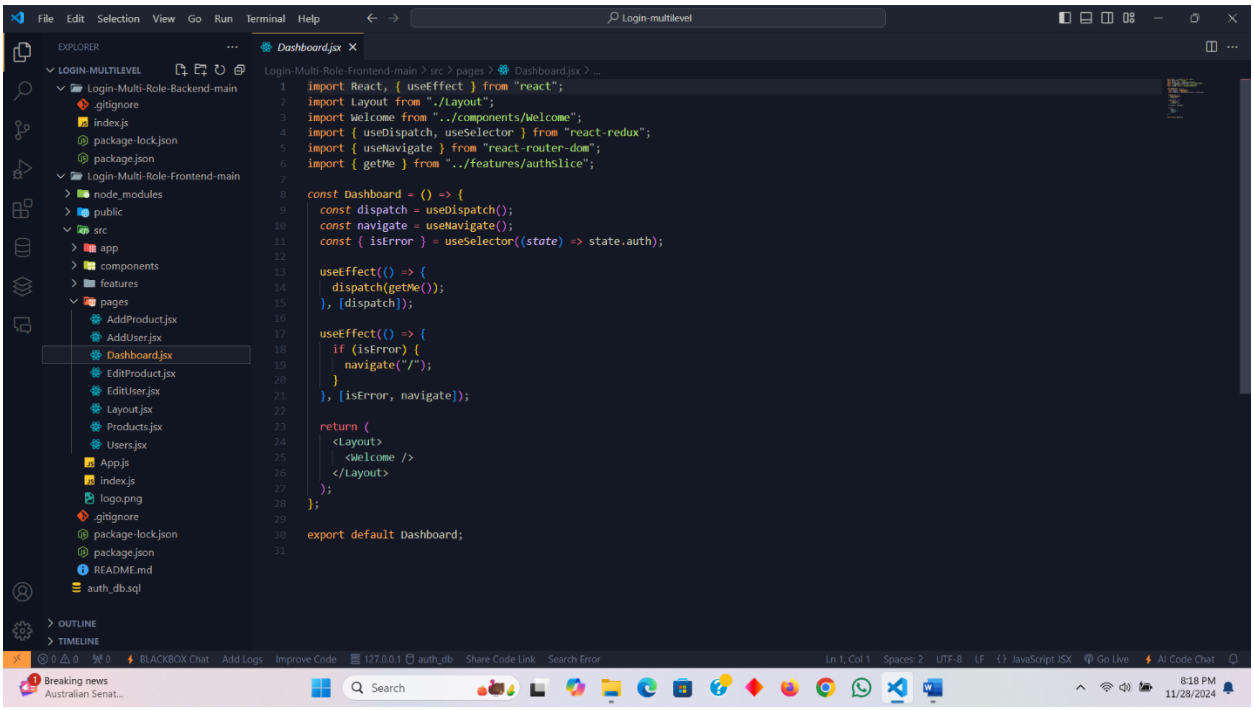
```jsx
import React, { useEffect } from "react";
import Layout from "./Layout";
import FormAddProduct from "../components/FormAddProduct";
import { useDispatch, useSelector } from "react-redux";
import { useNavigate } from "react-router-dom";
import { getMe } from "../features/authSlice";

const AddProduct = () => {
  const dispatch = useDispatch();
  const navigate = useNavigate();
  const { isError } = useSelector((state) => state.auth);

  useEffect(() => {
    dispatch(getMe());
  }, [dispatch]);

  useEffect(() => {
    if (isError) {
      navigate("/");
    }
  }, [isError, navigate]);
  return (
    <Layout>
      <FormAddProduct />
    </Layout>
  );
};

export default AddProduct;
```

-AddUser.jsx



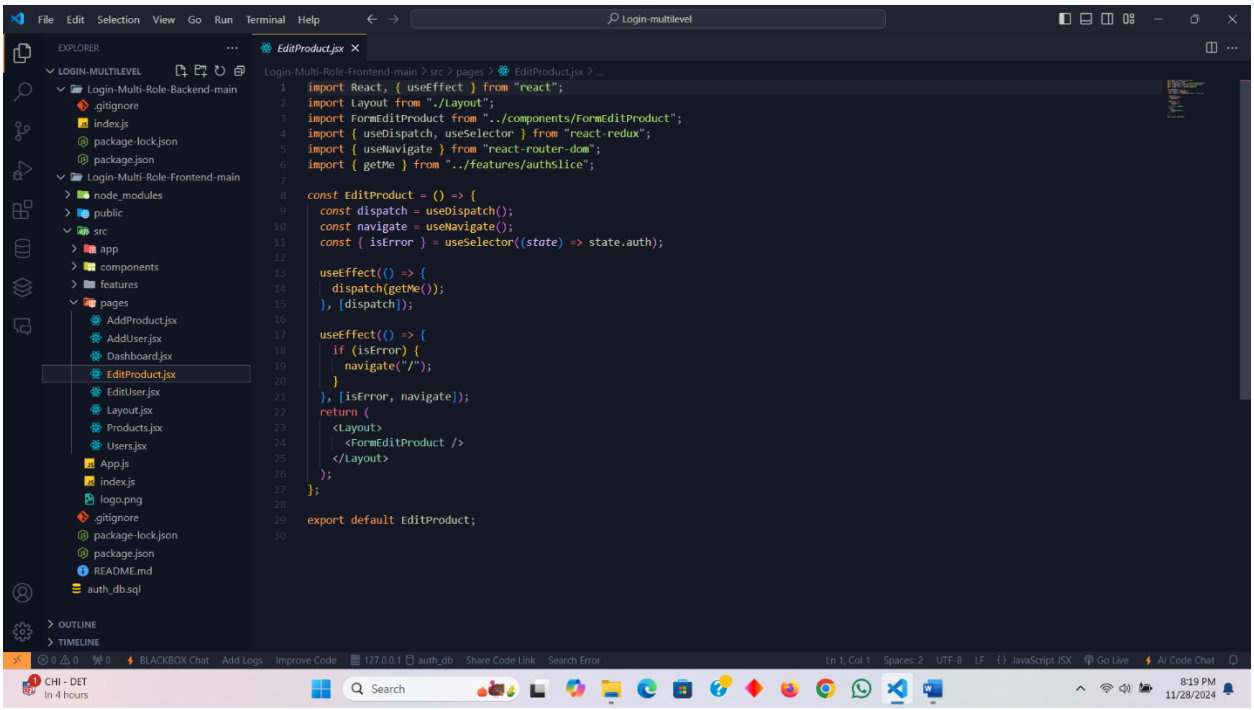
```jsx
import React, { useEffect } from "react";
import Layout from "./Layout";
import FormAddUser from "../components/FormAddUser";
import { useDispatch, useSelector } from "react-redux";
import { useNavigate } from "react-router-dom";
import { getMe } from "../features/authSlice";

const AddUser = () => {
  const dispatch = useDispatch();
  const navigate = useNavigate();
  const { isError, user } = useSelector((state) => state.auth);

  useEffect(() => {
    dispatch(getMe());
  }, [dispatch]);

  useEffect(() => {
    if (isError) {
      navigate("/");
    }
    if (user && user.role !== "admin") {
      navigate("/dashboard");
    }
  }, [isError, user, navigate]);
  return (
    <Layout>
      <FormAddUser />
    </Layout>
  );
};

export default AddUser;
```

-Dashboard.jsx



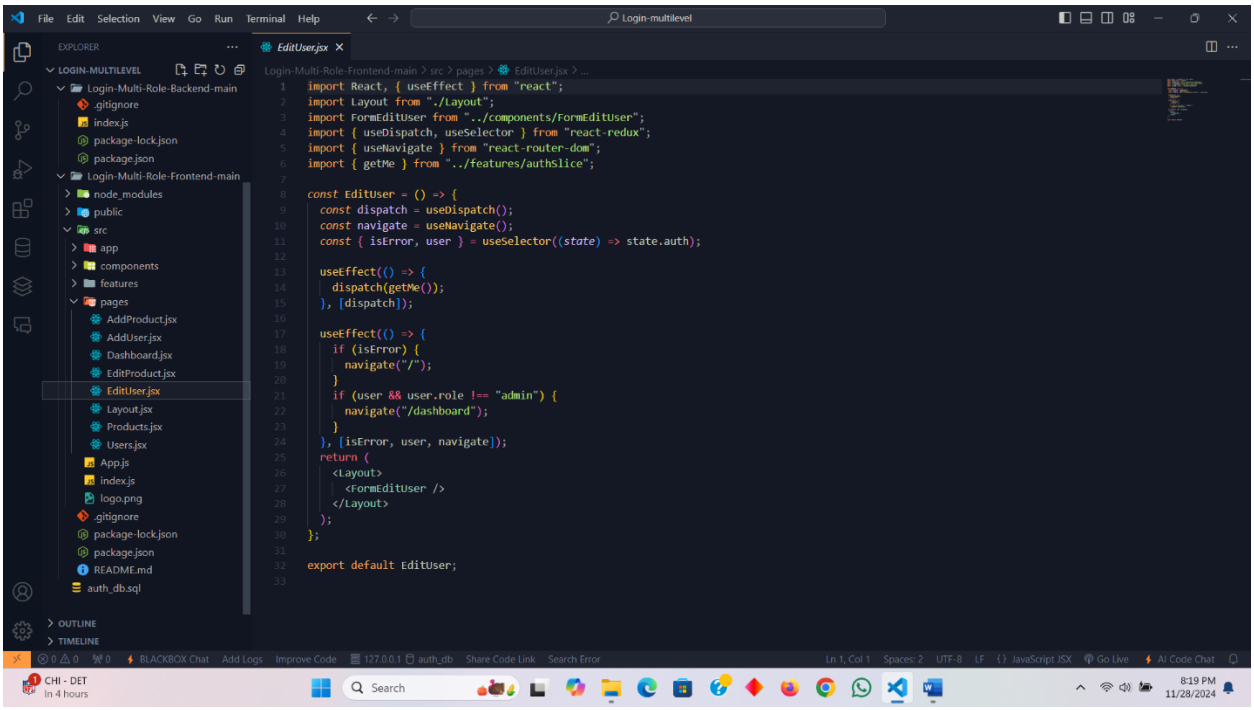
```jsx
import React, { useEffect } from "react";
import Layout from "./Layout";
import Welcome from "../components/Welcome";
import { useDispatch, useSelector } from "react-redux";
import { useNavigate } from "react-router-dom";
import { getMe } from "../features/authSlice";

const Dashboard = () => {
  const dispatch = useDispatch();
  const navigate = useNavigate();
  const { isError } = useSelector((state) => state.auth);

  useEffect(() => {
    dispatch(getMe());
  }, [dispatch]);

  useEffect(() => {
    if (isError) {
      navigate("/");
    }
  }, [isError, navigate]);

  return (
    <Layout>
      <Welcome />
    </Layout>
  );
};

export default Dashboard;
```

-EditProduct.jsx

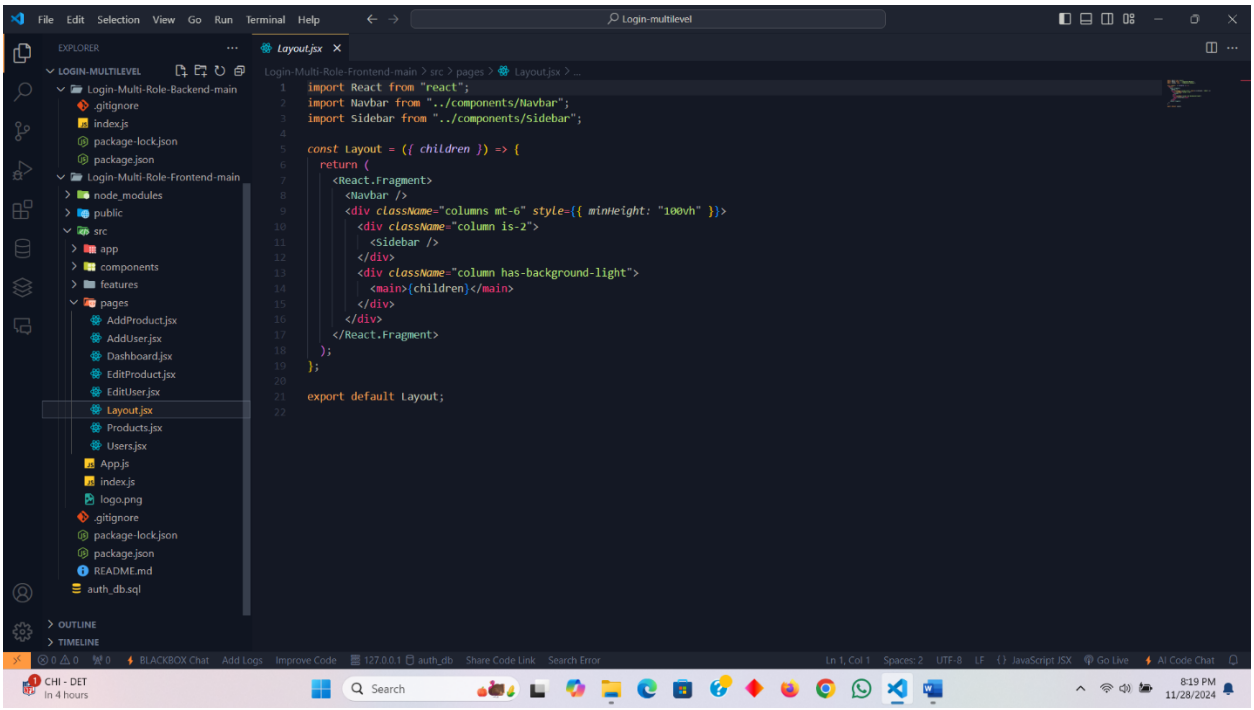


```jsx
import React, { useEffect } from "react";
import Layout from "./Layout";
import FormEditProduct from "../components/FormEditProduct";
import { useDispatch, useSelector } from "react-redux";
import { useNavigate } from "react-router-dom";
import { getMe } from "../features/authSlice";

const EditProduct = () => {
  const dispatch = useDispatch();
  const navigate = useNavigate();
  const { isError } = useSelector((state) => state.auth);

  useEffect(() => {
    dispatch(getMe());
  }, [dispatch]);

  useEffect(() => {
    if (isError) {
      navigate("/");
    }
  }, [isError, navigate]);
  return (
    <Layout>
      <FormEditProduct />
    </Layout>
  );
};

export default EditProduct;
```

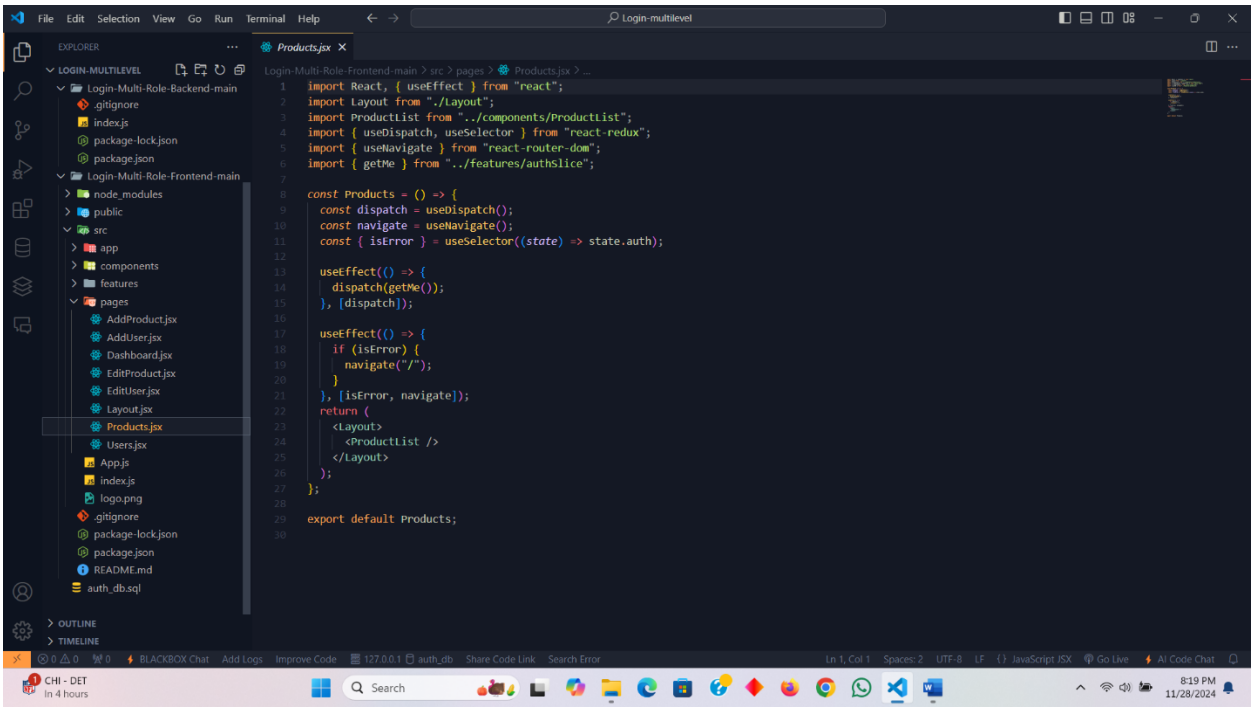
-EditUser.jsx



-Layout.jsx

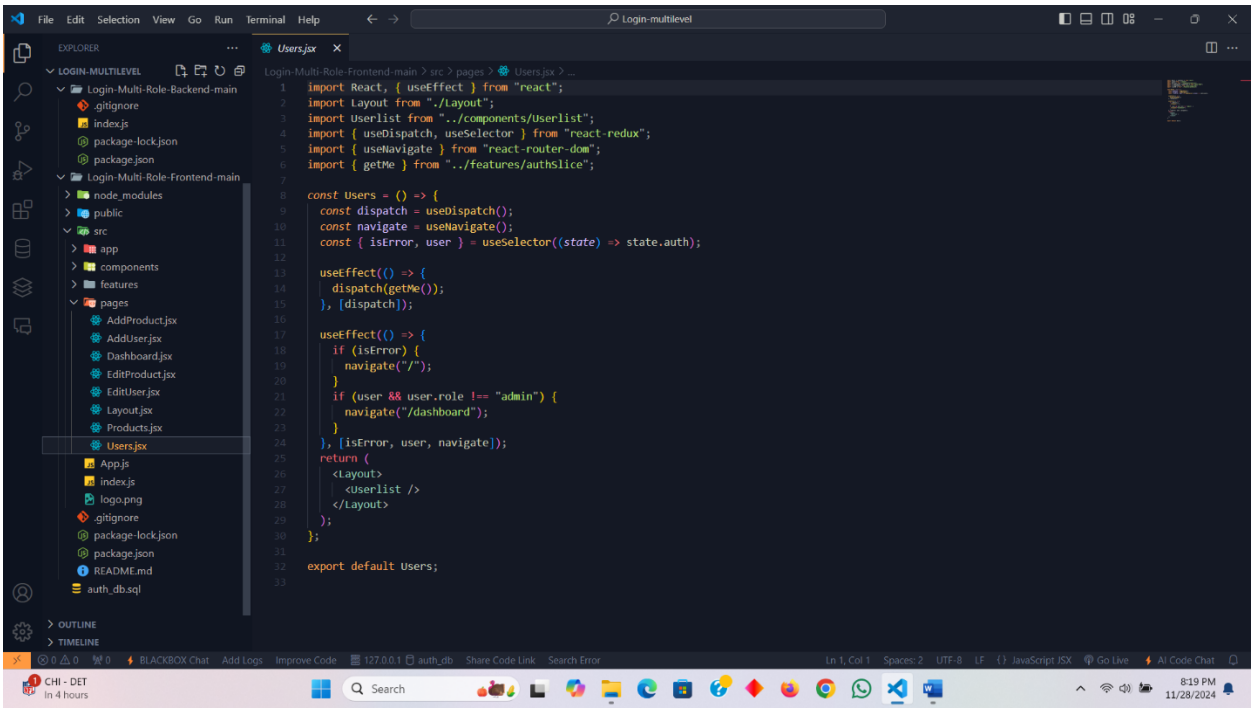


-Product.jsx



```jsx
import React, { useEffect } from "react";
import Layout from "./Layout";
import ProductList from "../components/ProductList";
import { useDispatch, useSelector } from "react-redux";
import { useNavigate } from "react-router-dom";
import { getMe } from "../features/authSlice";

const Products = () => {
  const dispatch = useDispatch();
  const navigate = useNavigate();
  const { isError } = useSelector((state) => state.auth);

  useEffect(() => {
    dispatch(getMe());
  }, [dispatch]);

  useEffect(() => {
    if (isError) {
      navigate("/");
    }
  }, [isError, navigate]);
  return (
    <Layout>
      <ProductList />
    </Layout>
  );
};

export default Products;
```

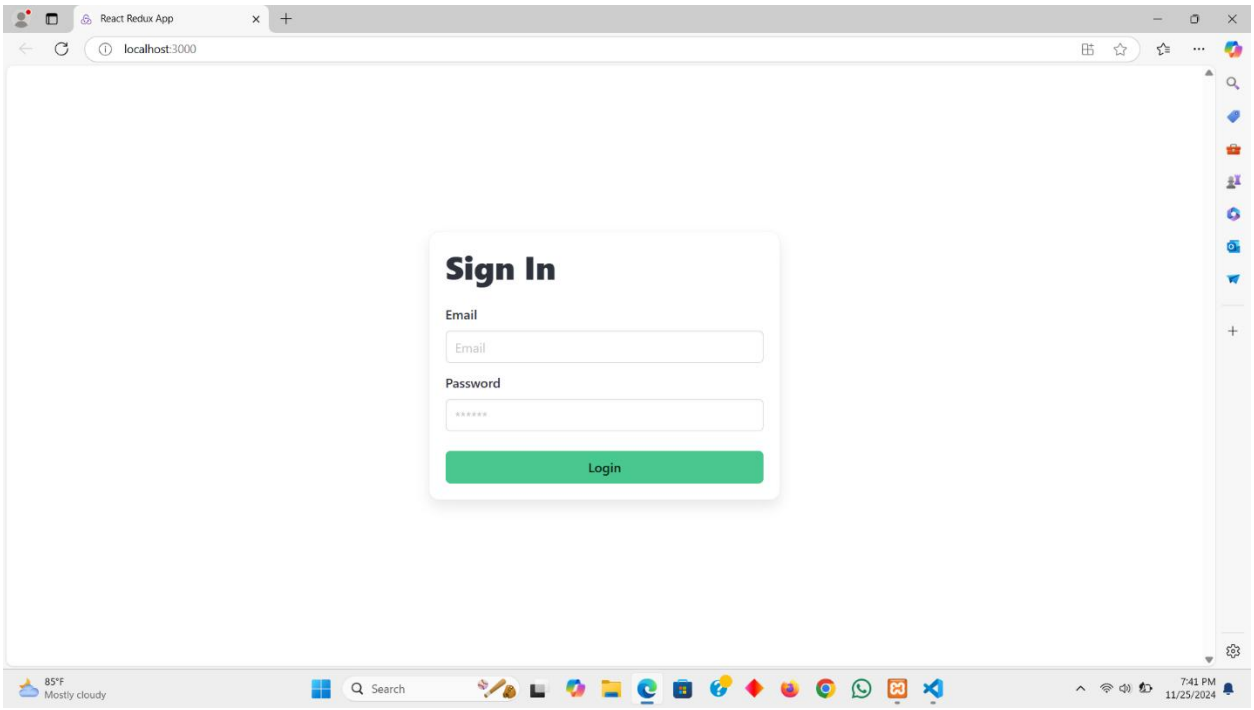
-Users.jsx



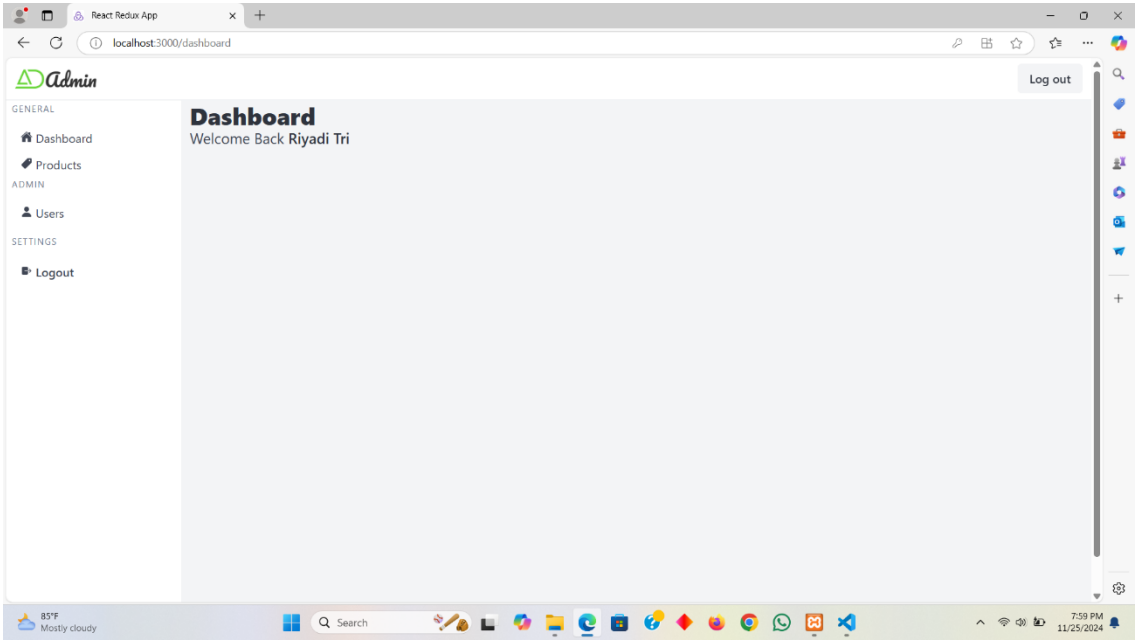
```jsx
import React, { useEffect } from "react";
import Layout from "./Layout";
import Userlist from "../components/Userlist";
import { useDispatch, useSelector } from "react-redux";
import { useNavigate } from "react-router-dom";
import { getMe } from "../features/authSlice";

const Users = () => {
  const dispatch = useDispatch();
  const navigate = useNavigate();
  const { isError, user } = useSelector((state) => state.auth);

  useEffect(() => {
    dispatch(getMe());
  }, [dispatch]);

  useEffect(() => {
    if (isError) {
      navigate("/");
    }
    if (user && user.role !== "admin") {
      navigate("/dashboard");
    }
  }, [isError, user, navigate]);
  return (
    <Layout>
      <Userlist />
    </Layout>
  );
};

export default Users;
```

*Tampilan Web

-Sign in

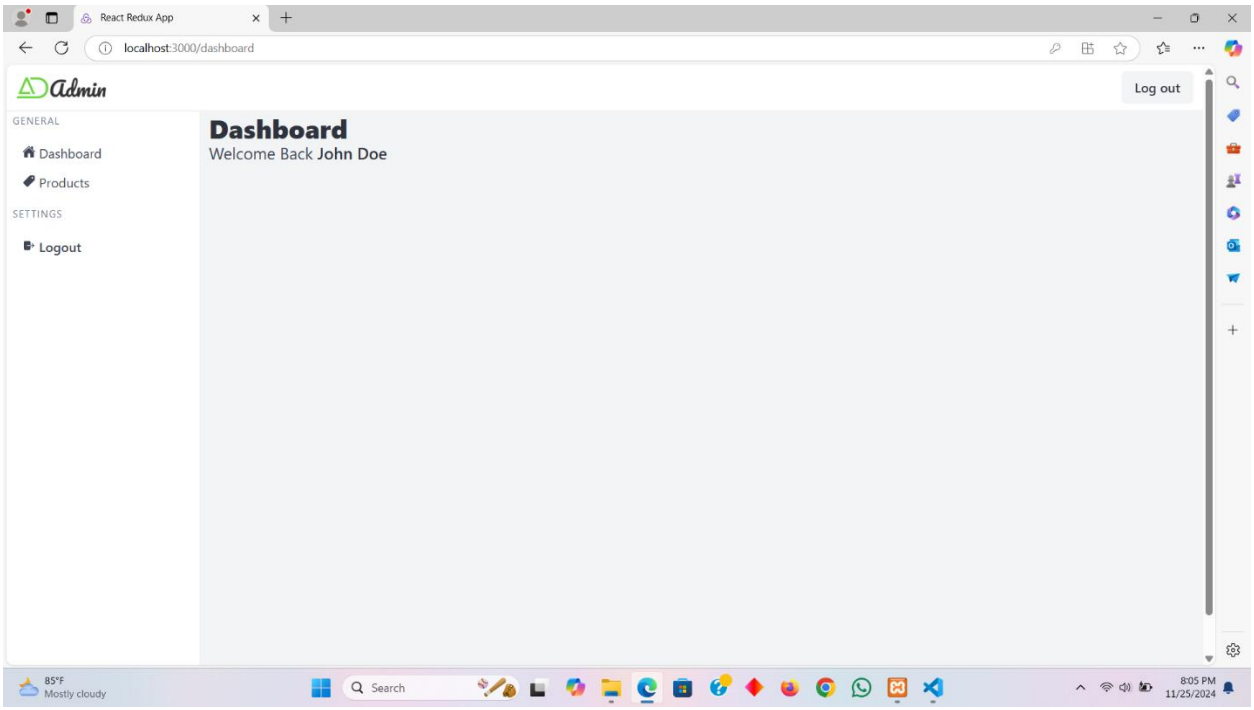


-Tampilan Dashboard Sebagai Admin

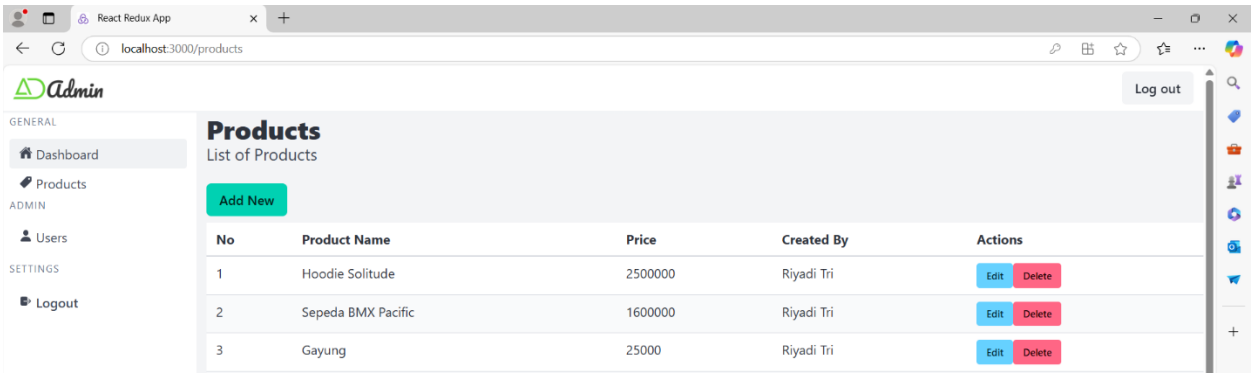


Bedanya kalau Admin kita bisa mengedit siapa saja yang mau menajdi admin ataupun users sedangkan kalau login sebagai users tidak bisa

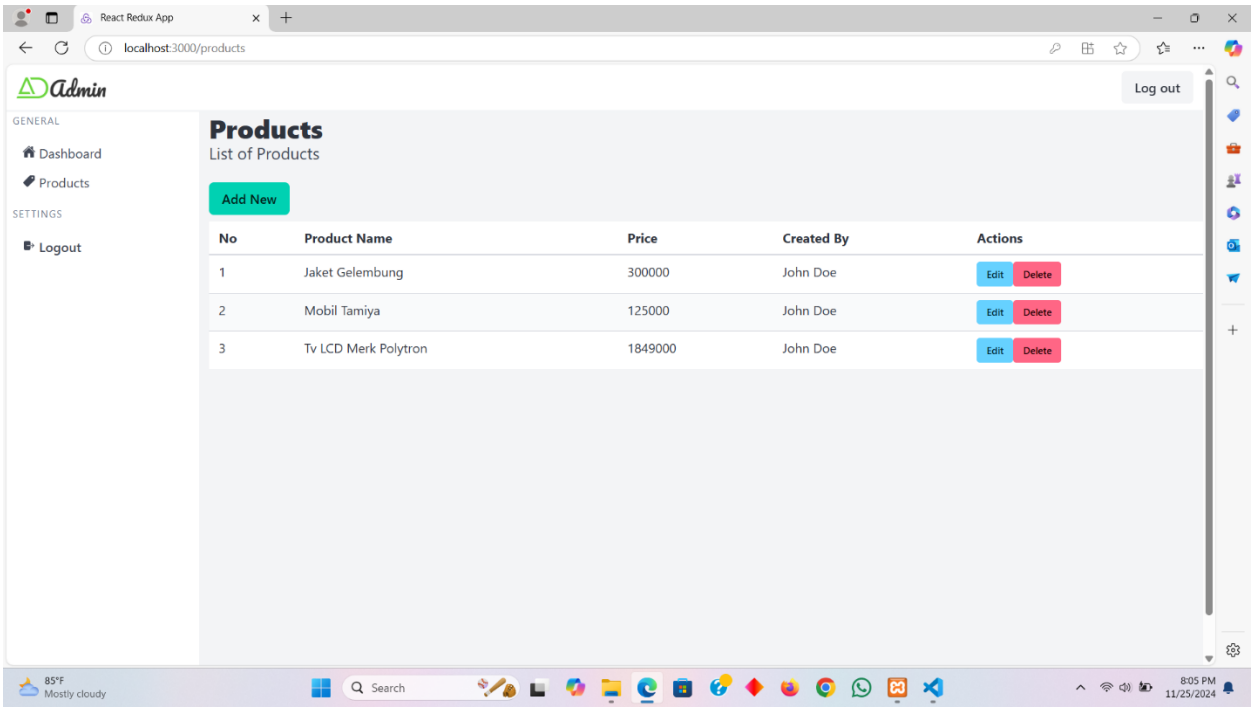
-Tampilan Dashboard Sebagai Users



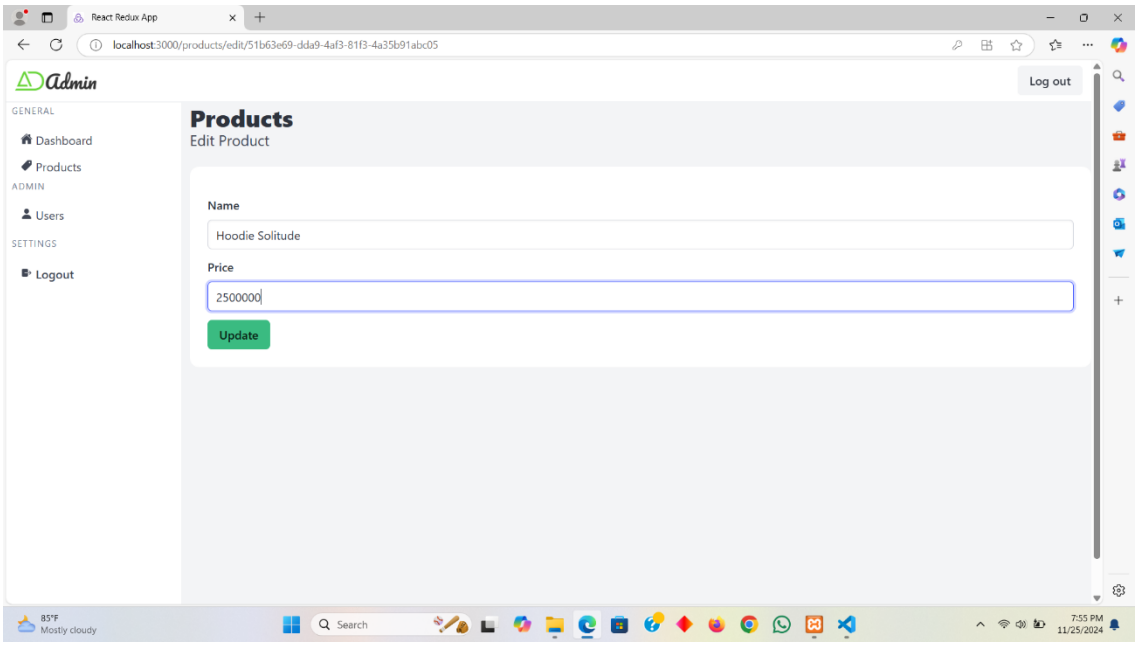
-Tampilan List Products Admin



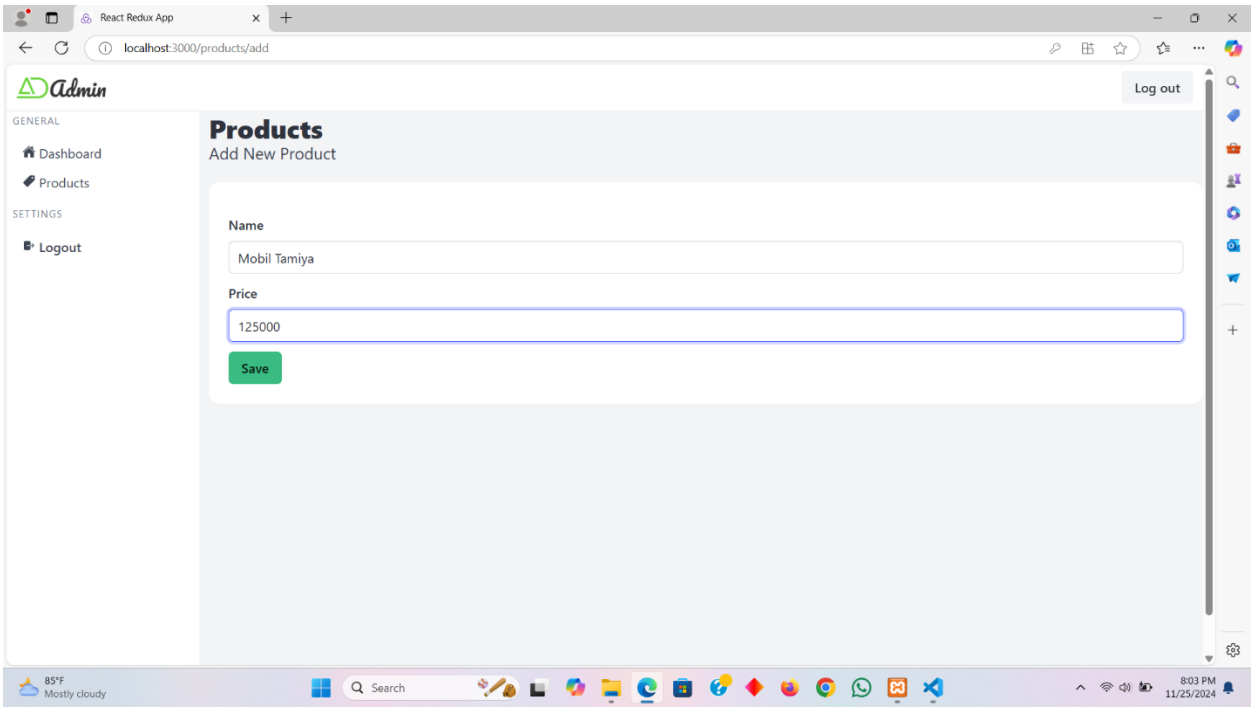
-Tampilan List Products Users



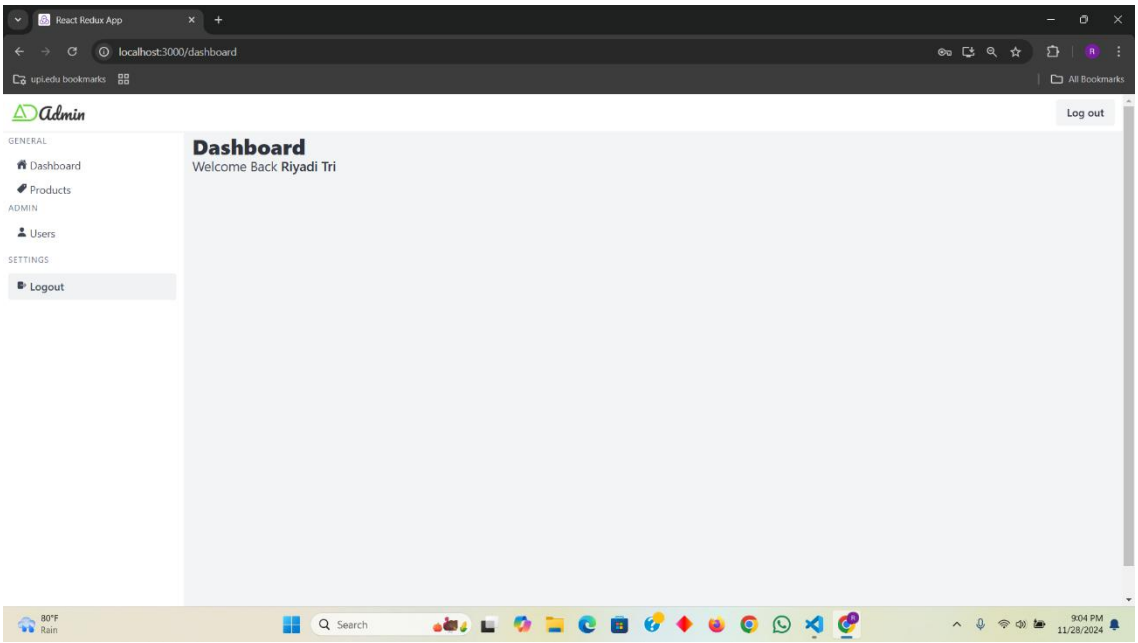
-Menambahkan Barang Sebagai Admin



-Menambahkan Barang Sebagai Users



-Logout



*Kesimpulan

Autentikasi website adalah langkah krusial untuk melindungi data pengguna dan mencegah akses tidak sah. Praktikum ini menekankan pentingnya penggunaan hashing password, validasi input, token autentikasi (seperti JWT), dan penerapan protokol keamanan seperti HTTPS. Pengujian keamanan dan penerapan fitur seperti logout aman dan multi-factor authentication (MFA) sangat diperlukan untuk menjaga integritas sistem