import React, { useState, useEffect } from 'react';

import styled, { createGlobalStyle } from 'styled-components';

import { BrowserRouter as Router, Route, Routes, Navigate, useLocation, useNavigate } from 'react-router-dom';

import { AuthProvider, useAuth } from './components/AuthContext'; // Controleer de import van useAuth

import Header from './components/Header';

import MainSection from './components/MainSection';

import SideSection from './components/SideSection';

import FAQ from './components/FAQ';

import Footer from './components/Footer';

import Sidebar from './components/Sidebar';

import WalletOverview from './components/WalletOverview';

import Login from './components/Login';

import Register from './components/Register';

import PrivateRoute from './components/PrivateRoute';

import Modal from './components/Modal';

import 'slick-carousel/slick/slick.css';

import 'slick-carousel/slick/slick-theme.css';

const GlobalStyle = createGlobalStyle`

body {

margin: 0;

font-family: 'Roboto', sans-serif;

background: linear-gradient(135deg, #1e3a8a, #2563eb);

color: white;

}

`;

const AppContainer = styled.div`

text-align: center;

`;

const App: React.FC = () => {

return (

<AuthProvider>

<Router>

<AppContainer>

<GlobalStyle />

<Content />

</AppContainer>

</Router>

</AuthProvider>

);

};

const Content: React.FC = () => {

const location = useLocation();

const navigate = useNavigate();

const { isAuthenticated } = useAuth();

const [isModalOpen, setIsModalOpen] = useState(false);

useEffect(() => {

// Toon de modal als de gebruiker niet geauthenticeerd is en naar de about- of servicespagina probeert te navigeren

if (!isAuthenticated && (location.pathname === '/about' || location.pathname === '/services')) {

setIsModalOpen(true);

} else {

setIsModalOpen(false);

}

}, [isAuthenticated, location.pathname]);

const closeModal = () => {

setIsModalOpen(false);

};

const handleLoginRedirect = () => {

setIsModalOpen(false);

navigate('/login');

};

const handleAboutClick = (event: React.MouseEvent<HTMLAnchorElement>) => {

event.preventDefault();

setIsModalOpen(true); // Open de modal wanneer op about-link wordt geklikt

};

const handleServicesClick = (event: React.MouseEvent<HTMLAnchorElement>) => {

event.preventDefault();

setIsModalOpen(true); // Open de modal wanneer op services-link wordt geklikt

};

const isAuthPage = location.pathname === '/login' || location.pathname === '/register';

const isHiddenPage = location.pathname === '/about' || location.pathname === '/services';

const isDashboardPage = location.pathname.startsWith('/dashboard');

const isOverviewBalancePage = location.pathname === '/dashboard/OverviewBalance';

return (

<>

<a href="/about" onClick={handleAboutClick}>About</a>

<a href="/services" onClick={handleServicesClick}>Services</a>

{!isHiddenPage && !isAuthPage && <Header />}

<Routes>

<Route path="/" element={<MainSection />} />

<Route path="/login" element={<Login />} />

<Route path="/register" element={<Register />} />

<Route path="/dashboard/\*" element={<DashboardRoutes />} />

<Route path="\*" element={<Navigate to="/" replace />} />

</Routes>

{!isAuthPage && !isHiddenPage && !isDashboardPage && !isOverviewBalancePage && <SideSection />}

{!isAuthPage && !isHiddenPage && !isDashboardPage && !isOverviewBalancePage && <FAQ />}

{!isAuthPage && !isHiddenPage && !isDashboardPage && !isOverviewBalancePage && <Footer />}

<Modal isOpen={isModalOpen} onClose={closeModal}>

<h2>Go to Login</h2>

<p>You must be logged in to view this page.</p>

<StyledButton onClick={handleLoginRedirect}>Go to Login</StyledButton>

</Modal>

</>

);

};

const DashboardRoutes: React.FC = () => {

return (

<div style={{ display: 'flex' }}>

<Sidebar />

<div style={{ flex: 1 }}>

<Routes>

<Route path="/" element={<Dashboard />} />

<Route path="OverviewBalance" element={<PrivateRoute element={<WalletOverview />} path="/dashboard/OverviewBalance" />} />

</Routes>

</div>

</div>

);

};

const Dashboard: React.FC = () => {

return (

<div>

<h1>Welcome to the Dashboard</h1>

{/\* Your dashboard content \*/}

</div>

);

};

const StyledButton = styled.button`

background-color: #1e3a8a;

color: white;

border: none;

padding: 10px 20px;

border-radius: 5px;

font-size: 16px;

cursor: pointer;

margin-top: 20px;

&:hover {

background-color: #2563eb;

}

`;

export default App;