# File Paths and Contents

File Path: .\.gitignore

Content:

# Logs  
logs  
\*.log  
npm-debug.log\*  
yarn-debug.log\*  
yarn-error.log\*  
pnpm-debug.log\*  
lerna-debug.log\*  
  
node\_modules  
dist  
dist-ssr  
\*.local  
  
# Editor directories and files  
.vscode/\*  
!.vscode/extensions.json  
.idea  
.DS\_Store  
\*.suo  
\*.ntvs\*  
\*.njsproj  
\*.sln  
\*.sw?

File Path: .\index.html

Content:

<!doctype html>  
<html lang="en">  
 <head>  
 <meta charset="UTF-8" />  
 <link rel="icon" type="image/svg+xml" href="/vite.svg" />  
 <meta name="viewport" content="width=device-width, initial-scale=1.0" />  
 <link href="https://fonts.googleapis.com/css2?family=Jost:wght@300;400;500;600;700&display=swap" rel="stylesheet">  
 <title>Vite + React</title>  
 </head>  
 <body>  
 <div id="root"></div>  
 <script type="module" src="/src/main.jsx"></script>  
 </body>  
</html>

File Path: .\package.json

Content:

{  
 "name": "perfume-store",  
 "private": true,  
 "version": "0.0.0",  
 "type": "module",  
 "scripts": {  
 "dev": "vite",  
 "build": "vite build",  
 "lint": "eslint .",  
 "preview": "vite preview"  
 },  
 "dependencies": {  
 "axios": "^1.9.0",  
 "framer-motion": "^12.10.4",  
 "i18next": "^25.1.2",  
 "react": "^19.1.0",  
 "react-dom": "^19.1.0",  
 "react-i18next": "^15.5.1",  
 "react-icons": "^5.5.0",  
 "react-router-dom": "^7.6.0",  
 "styled-components": "^6.1.18"  
 },  
 "devDependencies": {  
 "@eslint/js": "^9.25.0",  
 "@types/react": "^19.1.2",  
 "@types/react-dom": "^19.1.2",  
 "@vitejs/plugin-react": "^4.4.1",  
 "eslint": "^9.25.0",  
 "eslint-plugin-react-hooks": "^5.2.0",  
 "eslint-plugin-react-refresh": "^0.4.19",  
 "globals": "^16.0.0",  
 "vite": "^6.3.5"  
 }  
}

File Path: .\README.md

Content:

# React + Vite  
  
This template provides a minimal setup to get React working in Vite with HMR and some ESLint rules.  
  
Currently, two official plugins are available:  
  
- [@vitejs/plugin-react](https://github.com/vitejs/vite-plugin-react/blob/main/packages/plugin-react) uses [Babel](https://babeljs.io/) for Fast Refresh  
- [@vitejs/plugin-react-swc](https://github.com/vitejs/vite-plugin-react/blob/main/packages/plugin-react-swc) uses [SWC](https://swc.rs/) for Fast Refresh  
  
## Expanding the ESLint configuration  
  
If you are developing a production application, we recommend using TypeScript with type-aware lint rules enabled. Check out the [TS template](https://github.com/vitejs/vite/tree/main/packages/create-vite/template-react-ts) for information on how to integrate TypeScript and [`typescript-eslint`](https://typescript-eslint.io) in your project.

File Path: .\vite.config.js

Content:

import { defineConfig } from 'vite'  
import react from '@vitejs/plugin-react'  
  
// https://vitejs.dev/config/  
export default defineConfig({  
 plugins: [react()],  
 server: {  
  
 }  
})

File Path: .\public\vite.svg

Content:

<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" aria-hidden="true" role="img" class="iconify iconify--logos" width="31.88" height="32" preserveAspectRatio="xMidYMid meet" viewBox="0 0 256 257"><defs><linearGradient id="IconifyId1813088fe1fbc01fb466" x1="-.828%" x2="57.636%" y1="7.652%" y2="78.411%"><stop offset="0%" stop-color="#41D1FF"></stop><stop offset="100%" stop-color="#BD34FE"></stop></linearGradient><linearGradient id="IconifyId1813088fe1fbc01fb467" x1="43.376%" x2="50.316%" y1="2.242%" y2="89.03%"><stop offset="0%" stop-color="#FFEA83"></stop><stop offset="8.333%" stop-color="#FFDD35"></stop><stop offset="100%" stop-color="#FFA800"></stop></linearGradient></defs><path fill="url(#IconifyId1813088fe1fbc01fb466)" d="M255.153 37.938L134.897 252.976c-2.483 4.44-8.862 4.466-11.382.048L.875 37.958c-2.746-4.814 1.371-10.646 6.827-9.67l120.385 21.517a6.537 6.537 0 0 0 2.322-.004l117.867-21.483c5.438-.991 9.574 4.796 6.877 9.62Z"></path><path fill="url(#IconifyId1813088fe1fbc01fb467)" d="M185.432.063L96.44 17.501a3.268 3.268 0 0 0-2.634 3.014l-5.474 92.456a3.268 3.268 0 0 0 3.997 3.378l24.777-5.718c2.318-.535 4.413 1.507 3.936 3.838l-7.361 36.047c-.495 2.426 1.782 4.5 4.151 3.78l15.304-4.649c2.372-.72 4.652 1.36 4.15 3.788l-11.698 56.621c-.732 3.542 3.979 5.473 5.943 2.437l1.313-2.028l72.516-144.72c1.215-2.423-.88-5.186-3.54-4.672l-25.505 4.922c-2.396.462-4.435-1.77-3.759-4.114l16.646-57.705c.677-2.35-1.37-4.583-3.769-4.113Z"></path></svg>

File Path: .\public\images\hero-background.jpg

Could not read file .\public\images\hero-background.jpg: 'utf-8' codec can't decode byte 0xff in position 0: invalid start byte

File Path: .\public\images\login-image.jpg

Could not read file .\public\images\login-image.jpg: 'utf-8' codec can't decode byte 0xff in position 0: invalid start byte

File Path: .\public\images\register-image.jpg

Could not read file .\public\images\register-image.jpg: 'utf-8' codec can't decode byte 0xff in position 0: invalid start byte

File Path: .\public\images\sirm-logo.png

Could not read file .\public\images\sirm-logo.png: 'utf-8' codec can't decode byte 0x89 in position 0: invalid start byte

File Path: .\public\images\story-image.jpg

Could not read file .\public\images\story-image.jpg: 'utf-8' codec can't decode byte 0xff in position 0: invalid start byte

File Path: .\src\App.css

Content:

#root {  
 max-width: 1280px;  
 margin: 0 auto;  
 padding: 2rem;  
 text-align: center;  
}  
  
.logo {  
 height: 6em;  
 padding: 1.5em;  
 will-change: filter;  
 transition: filter 300ms;  
}  
.logo:hover {  
 filter: drop-shadow(0 0 2em #646cffaa);  
}  
.logo.react:hover {  
 filter: drop-shadow(0 0 2em #61dafbaa);  
}  
  
@keyframes logo-spin {  
 from {  
 transform: rotate(0deg);  
 }  
 to {  
 transform: rotate(360deg);  
 }  
}  
  
@media (prefers-reduced-motion: no-preference) {  
 a:nth-of-type(2) .logo {  
 animation: logo-spin infinite 20s linear;  
 }  
}  
  
.card {  
 padding: 2em;  
}  
  
.read-the-docs {  
 color: #888;  
}

File Path: .\src\App.jsx

Content:

import { useState } from 'react'  
import { Routes, Route } from 'react-router-dom'  
import { useTranslation } from 'react-i18next'  
import Header from './components/Header'  
import Footer from './components/Footer'  
import Home from './pages/Home'  
import ProductsPage from './pages/ProductsPage'  
import ProductDetail from './pages/ProductDetail'  
import Cart from './pages/Cart'  
import Login from './pages/Login'  
import Register from './pages/Register'  
import AdminDashboard from './pages/admin/Dashboard'  
import AdminProducts from './pages/admin/Products'  
import AdminOrders from './pages/admin/Orders'  
import Notifications from './components/Notifications'  
import Loading from './components/Loading'  
import PrivateRoute from './components/PrivateRoute'  
import GuestRoute from './components/GuestRoute'  
import AdminRoute from './components/AdminRoute'  
import './App.css'  
  
function App() {  
 const { i18n } = useTranslation();  
 const [isRTL, setIsRTL] = useState(i18n.language === 'ar');  
  
 const changeLanguage = (lng) => {  
 i18n.changeLanguage(lng);  
 setIsRTL(lng === 'ar');  
 document.documentElement.dir = lng === 'ar' ? 'rtl' : 'ltr';  
 document.documentElement.lang = lng;  
 };  
  
 return (  
 <div className={`app ${isRTL ? 'rtl' : 'ltr'}`}>  
 <Notifications />  
 <Loading />  
 <Header changeLanguage={changeLanguage} currentLang={i18n.language} />  
 <main className="main-content">  
 <Routes>  
 <Route path="/" element={<Home />} />  
 <Route path="/products" element={<ProductsPage />} />  
 <Route path="/product/:id" element={<ProductDetail />} />  
 <Route path="/cart" element={<Cart />} />  
  
 {/\* Public Routes which is not for authenticated users \*/}  
 <Route path="/login" element={  
 <GuestRoute>  
 <Login />  
 </GuestRoute>  
 } />  
 <Route path="/register" element={  
 <GuestRoute>  
 <Register />  
 </GuestRoute>  
 } />  
   
 {/\* Private Routes \*/}  
 <Route path="/checkout" element={  
 <PrivateRoute>  
 <div>Checkout Page</div> {/\* You can replace with actual Checkout component \*/}  
 </PrivateRoute>  
 } />  
 <Route path="/profile" element={  
 <PrivateRoute>  
 <div>User Profile</div> {/\* You can replace with actual Profile component \*/}  
 </PrivateRoute>  
 } />  
   
 {/\* Admin Routes \*/}  
 <Route path="/admin" element={  
 <AdminRoute>  
 <AdminDashboard />  
 </AdminRoute>  
 } />  
 <Route path="/admin/products" element={  
 <AdminRoute>  
 <AdminProducts />  
 </AdminRoute>  
 } />  
 <Route path="/admin/orders" element={  
 <AdminRoute>  
 <AdminOrders />  
 </AdminRoute>  
 } />  
 </Routes>  
 </main>  
 <Footer />  
 </div>  
 )  
}  
  
export default App

File Path: .\src\i18n.js

Content:

import i18n from 'i18next';  
import { initReactI18next } from 'react-i18next';  
  
const resources = {  
 en: {  
 translation: {  
 'add\_to\_cart': 'Add to cart',  
 'out\_of\_stock': 'Out of stock',  
 'all\_over\_spray': 'All Over Spray',  
 'home': 'Home',  
 'products': 'Products',  
 'about': 'About',  
 'contact': 'Contact',  
 'login': 'Login',  
 'search': 'Search',  
 'cart': 'Cart',  
 // Add these new translations  
 'profile': 'My Profile',  
 'my\_orders': 'My Orders',  
 'admin\_dashboard': 'Admin Dashboard',  
 'logout': 'Logout',  
 'welcome\_back': 'Welcome back',  
 'account\_settings': 'Account Settings',  
 'no\_account': 'Don\'t have an account?',  
 'create\_account': 'Create Account',  
 'already\_account': 'Already have an account?',  
 'order\_summary': 'Order Summary',  
 'continue\_shopping': 'Continue Shopping',  
 'proceed\_to\_checkout': 'Proceed to Checkout',  
 'register\_subtitle': 'Create your account to start shopping',  
 'full\_name': 'Full Name',  
 'name\_placeholder': 'Enter your full name',  
 'email\_address': 'Email Address',  
 'email\_placeholder': 'Enter your email address',  
 'password': 'Password',  
 'password\_placeholder': 'Enter your password',  
 'confirm\_password': 'Confirm Password',  
 'confirm\_password\_placeholder': 'Confirm your password',  
 'phone\_number': 'Phone Number',  
 'phone\_placeholder': 'Enter your phone number',  
 'agree\_terms': 'I agree to the',  
 'terms\_conditions': 'Terms and Conditions',  
 'create\_account': 'Create Account'  
   
 }  
 },  
 ar: {  
 translation: {  
 'add\_to\_cart': 'أضف إلى السلة',  
 'out\_of\_stock': 'نفذت الكمية',  
 'all\_over\_spray': 'عطر الجسم',  
 'home': 'الرئيسية',  
 'products': 'المنتجات',  
 'about': 'من نحن',  
 'contact': 'اتصل بنا',  
 'login': 'تسجيل الدخول',  
 'search': 'بحث',  
 'cart': 'سلة التسوق',  
 // Add these new translations  
 'profile': 'الملف الشخصي',  
 'my\_orders': 'طلباتي',  
 'admin\_dashboard': 'لوحة التحكم',  
 'logout': 'تسجيل الخروج',  
 'welcome\_back': 'مرحبا بعودتك',  
 'account\_settings': 'إعدادات الحساب',  
 'no\_account': 'ليس لديك حساب؟',  
 'create\_account': 'إنشاء حساب',  
 'already\_account': 'لديك حساب بالفعل؟',  
 'order\_summary': 'ملخص الطلب',  
 'continue\_shopping': 'مواصلة التسوق',  
 'proceed\_to\_checkout': 'إتمام عملية الشراء',  
 'register\_subtitle': 'أنشئ حسابك لبدء التسوق',  
 'full\_name': 'الاسم الكامل',  
 'name\_placeholder': 'أدخل اسمك الكامل',  
 'email\_address': 'البريد الإلكتروني',  
 'email\_placeholder': 'أدخل بريدك الإلكتروني',  
 'password': 'كلمة المرور',  
 'password\_placeholder': 'أدخل كلمة المرور',  
 'confirm\_password': 'تأكيد كلمة المرور',  
 'confirm\_password\_placeholder': 'تأكيد كلمة المرور',  
 'phone\_number': 'رقم الهاتف',  
 'phone\_placeholder': 'أدخل رقم هاتفك',  
 'agree\_terms': 'أوافق على',  
 'terms\_conditions': 'الشروط والأحكام',  
 'create\_account': 'إنشاء حساب'  
 }  
 }  
};  
  
i18n  
 .use(initReactI18next)  
 .init({  
 resources,  
 lng: 'ar', // Default language  
 fallbackLng: 'en',  
 interpolation: {  
 escapeValue: false,  
 },  
 react: {  
 useSuspense: false,  
 }  
 });  
  
export default i18n;

File Path: .\src\index.css

Content:

:root {  
 --primary-color: #111111;  
 --secondary-color: #D4AF37;  
 --accent-color: #E6BE8A;  
 --text-color: #F5F5F5;  
 --highlight-color: #FFF8E1;  
 --danger-color: #e74c3c;  
 --success-color: #27ae60;  
 --background-color: #ffffff;  
  
 /\* New AlBidaa-inspired colors \*/  
 --primary-color: #38281D; /\* Rich brown \*/  
 --secondary-color: #C7A976; /\* Gold \*/  
 --accent-color: #DBC8A0; /\* Light beige \*/  
 --text-color: #F5F5F5; /\* Keep light text \*/  
 --highlight-color: #F4EEE1; /\* Pale cream \*/  
 --background-color: #ffffff;  
 --light-beige: #F9F5EB; /\* Additional beige for backgrounds \*/  
 --dark-accent: #6B4B35; /\* Darker brown for some elements \*/  
  
  
 --card-shadow: 0 4px 12px rgba(0, 0, 0, 0.1);  
 --transition-speed: 0.3s;  
 --border-radius: 4px;  
}  
  
\* {  
 box-sizing: border-box;  
 margin: 0;  
 padding: 0;  
}  
  
/\* body {  
 font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
 background-color: var(--background-color);  
 color: #333;  
 line-height: 1.6;  
} \*/  
  
body {  
 font-family: 'Jost', sans-serif;  
 /\* background-color: var(--background-color); \*/  
 background-color: var(--light-beige);  
 color: #333;  
 line-height: 1.6;  
}  
  
/\* RTL Support \*/  
.rtl {  
 direction: rtl;  
 text-align: right;  
}  
  
.ltr {  
 direction: ltr;  
 text-align: left;  
}  
  
/\* Container \*/  
.container {  
 max-width: 1200px;  
 margin: 0 auto;  
 padding: 0 15px;  
}  
  
/\* Typography \*/  
h1, h2, h3, h4, h5, h6 {  
 font-family: 'Playfair Display', serif;  
 margin-bottom: 1rem;  
 font-weight: 600;  
}  
  
button {  
 cursor: pointer;  
 transition: all var(--transition-speed) ease;  
}  
  
/\* Arabic Font Support \*/  
.rtl {  
 font-family: 'Dubai', 'Amiri', sans-serif;  
}  
  
.rtl h1, .rtl h2, .rtl h3, .rtl h4, .rtl h5, .rtl h6 {  
 font-family: 'Dubai', 'Amiri', serif;  
}  
  
.main-content {  
 min-height: calc(100vh - 160px);  
 padding: 20px 0;  
}  
  
/\* @import url('https://fonts.googleapis.com/css2?family=Playfair+Display:wght@400;500;600;700&family=Montserrat:wght@300;400;500;600&display=swap'); \*/

File Path: .\src\main.jsx

Content:

import React from 'react'  
import ReactDOM from 'react-dom/client'  
import App from './App.jsx'  
import './index.css'  
import { BrowserRouter } from 'react-router-dom'  
import { I18nextProvider } from 'react-i18next'  
import i18n from './i18n'  
import { AppProviders } from './context'  
  
ReactDOM.createRoot(document.getElementById('root')).render(  
 <React.StrictMode>  
 <BrowserRouter>  
 <I18nextProvider i18n={i18n}>  
 <AppProviders>  
 <App />  
 </AppProviders>  
 </I18nextProvider>  
 </BrowserRouter>  
 </React.StrictMode>,  
)

File Path: .\src\api\config.js

Content:

import axios from 'axios';  
  
// Create custom axios instance with the correct base URL  
const api = axios.create({  
 baseURL: 'http://localhost:5000', // Your backend server URL  
 headers: {  
 'Content-Type': 'application/json',  
 }  
});  
  
// Add request interceptor to handle authentication  
api.interceptors.request.use(  
 (config) => {  
 const token = localStorage.getItem('token');  
 if (token) {  
 config.headers['Authorization'] = `Bearer ${token}`;  
 }  
 return config;  
 },  
 (error) => Promise.reject(error)  
);  
  
export default api;

File Path: .\src\assets\react.svg

Content:

<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" aria-hidden="true" role="img" class="iconify iconify--logos" width="35.93" height="32" preserveAspectRatio="xMidYMid meet" viewBox="0 0 256 228"><path fill="#00D8FF" d="M210.483 73.824a171.49 171.49 0 0 0-8.24-2.597c.465-1.9.893-3.777 1.273-5.621c6.238-30.281 2.16-54.676-11.769-62.708c-13.355-7.7-35.196.329-57.254 19.526a171.23 171.23 0 0 0-6.375 5.848a155.866 155.866 0 0 0-4.241-3.917C100.759 3.829 77.587-4.822 63.673 3.233C50.33 10.957 46.379 33.89 51.995 62.588a170.974 170.974 0 0 0 1.892 8.48c-3.28.932-6.445 1.924-9.474 2.98C17.309 83.498 0 98.307 0 113.668c0 15.865 18.582 31.778 46.812 41.427a145.52 145.52 0 0 0 6.921 2.165a167.467 167.467 0 0 0-2.01 9.138c-5.354 28.2-1.173 50.591 12.134 58.266c13.744 7.926 36.812-.22 59.273-19.855a145.567 145.567 0 0 0 5.342-4.923a168.064 168.064 0 0 0 6.92 6.314c21.758 18.722 43.246 26.282 56.54 18.586c13.731-7.949 18.194-32.003 12.4-61.268a145.016 145.016 0 0 0-1.535-6.842c1.62-.48 3.21-.974 4.76-1.488c29.348-9.723 48.443-25.443 48.443-41.52c0-15.417-17.868-30.326-45.517-39.844Zm-6.365 70.984c-1.4.463-2.836.91-4.3 1.345c-3.24-10.257-7.612-21.163-12.963-32.432c5.106-11 9.31-21.767 12.459-31.957c2.619.758 5.16 1.557 7.61 2.4c23.69 8.156 38.14 20.213 38.14 29.504c0 9.896-15.606 22.743-40.946 31.14Zm-10.514 20.834c2.562 12.94 2.927 24.64 1.23 33.787c-1.524 8.219-4.59 13.698-8.382 15.893c-8.067 4.67-25.32-1.4-43.927-17.412a156.726 156.726 0 0 1-6.437-5.87c7.214-7.889 14.423-17.06 21.459-27.246c12.376-1.098 24.068-2.894 34.671-5.345a134.17 134.17 0 0 1 1.386 6.193ZM87.276 214.515c-7.882 2.783-14.16 2.863-17.955.675c-8.075-4.657-11.432-22.636-6.853-46.752a156.923 156.923 0 0 1 1.869-8.499c10.486 2.32 22.093 3.988 34.498 4.994c7.084 9.967 14.501 19.128 21.976 27.15a134.668 134.668 0 0 1-4.877 4.492c-9.933 8.682-19.886 14.842-28.658 17.94ZM50.35 144.747c-12.483-4.267-22.792-9.812-29.858-15.863c-6.35-5.437-9.555-10.836-9.555-15.216c0-9.322 13.897-21.212 37.076-29.293c2.813-.98 5.757-1.905 8.812-2.773c3.204 10.42 7.406 21.315 12.477 32.332c-5.137 11.18-9.399 22.249-12.634 32.792a134.718 134.718 0 0 1-6.318-1.979Zm12.378-84.26c-4.811-24.587-1.616-43.134 6.425-47.789c8.564-4.958 27.502 2.111 47.463 19.835a144.318 144.318 0 0 1 3.841 3.545c-7.438 7.987-14.787 17.08-21.808 26.988c-12.04 1.116-23.565 2.908-34.161 5.309a160.342 160.342 0 0 1-1.76-7.887Zm110.427 27.268a347.8 347.8 0 0 0-7.785-12.803c8.168 1.033 15.994 2.404 23.343 4.08c-2.206 7.072-4.956 14.465-8.193 22.045a381.151 381.151 0 0 0-7.365-13.322Zm-45.032-43.861c5.044 5.465 10.096 11.566 15.065 18.186a322.04 322.04 0 0 0-30.257-.006c4.974-6.559 10.069-12.652 15.192-18.18ZM82.802 87.83a323.167 323.167 0 0 0-7.227 13.238c-3.184-7.553-5.909-14.98-8.134-22.152c7.304-1.634 15.093-2.97 23.209-3.984a321.524 321.524 0 0 0-7.848 12.897Zm8.081 65.352c-8.385-.936-16.291-2.203-23.593-3.793c2.26-7.3 5.045-14.885 8.298-22.6a321.187 321.187 0 0 0 7.257 13.246c2.594 4.48 5.28 8.868 8.038 13.147Zm37.542 31.03c-5.184-5.592-10.354-11.779-15.403-18.433c4.902.192 9.899.29 14.978.29c5.218 0 10.376-.117 15.453-.343c-4.985 6.774-10.018 12.97-15.028 18.486Zm52.198-57.817c3.422 7.8 6.306 15.345 8.596 22.52c-7.422 1.694-15.436 3.058-23.88 4.071a382.417 382.417 0 0 0 7.859-13.026a347.403 347.403 0 0 0 7.425-13.565Zm-16.898 8.101a358.557 358.557 0 0 1-12.281 19.815a329.4 329.4 0 0 1-23.444.823c-7.967 0-15.716-.248-23.178-.732a310.202 310.202 0 0 1-12.513-19.846h.001a307.41 307.41 0 0 1-10.923-20.627a310.278 310.278 0 0 1 10.89-20.637l-.001.001a307.318 307.318 0 0 1 12.413-19.761c7.613-.576 15.42-.876 23.31-.876H128c7.926 0 15.743.303 23.354.883a329.357 329.357 0 0 1 12.335 19.695a358.489 358.489 0 0 1 11.036 20.54a329.472 329.472 0 0 1-11 20.722Zm22.56-122.124c8.572 4.944 11.906 24.881 6.52 51.026c-.344 1.668-.73 3.367-1.15 5.09c-10.622-2.452-22.155-4.275-34.23-5.408c-7.034-10.017-14.323-19.124-21.64-27.008a160.789 160.789 0 0 1 5.888-5.4c18.9-16.447 36.564-22.941 44.612-18.3ZM128 90.808c12.625 0 22.86 10.235 22.86 22.86s-10.235 22.86-22.86 22.86s-22.86-10.235-22.86-22.86s10.235-22.86 22.86-22.86Z"></path></svg>

File Path: .\src\components\AdminRoute.jsx

Content:

import { Navigate } from 'react-router-dom';  
import { useAuth } from '../context/AuthContext';  
  
const AdminRoute = ({ children }) => {  
 const { isAuthenticated, isAdmin, loading } = useAuth();  
  
 if (loading) {  
 return <div>Loading...</div>;  
 }  
  
 if (!isAuthenticated) {  
 return <Navigate to="/login?redirect=/admin" replace />;  
 }  
  
 if (!isAdmin) {  
 return <Navigate to="/" replace />;  
 }  
  
 return children;  
};  
  
export default AdminRoute;

File Path: .\src\components\Footer.jsx

Content:

import { Link } from 'react-router-dom';  
import { useTranslation } from 'react-i18next';  
import styled from 'styled-components';  
import { FaFacebookF, FaTwitter, FaInstagram, FaPinterestP } from 'react-icons/fa';  
  
const Footer = () => {  
 const { t } = useTranslation();  
   
 return (  
 <FooterContainer>  
 <div className="container">  
 <FooterContent>  
 <FooterColumn>  
 {/\* <FooterLogo>sirm</FooterLogo> \*/}  
 <FooterLogo>  
 <img src="/images/sirm-logo.png" alt="sirm" height="32" />  
 </FooterLogo>   
 <FooterDescription>  
 A passionate story sirm started in 2018, looking for excellence and exclusivity,   
 taking everything in the field and taking it far to attract the largest category   
 that loves perfumes in the Middle East.  
 </FooterDescription>  
 <SocialLinks>  
 <SocialLink href="#" target="\_blank" rel="noopener noreferrer">  
 <FaFacebookF />  
 </SocialLink>  
 <SocialLink href="#" target="\_blank" rel="noopener noreferrer">  
 <FaTwitter />  
 </SocialLink>  
 <SocialLink href="#" target="\_blank" rel="noopener noreferrer">  
 <FaInstagram />  
 </SocialLink>  
 <SocialLink href="#" target="\_blank" rel="noopener noreferrer">  
 <FaPinterestP />  
 </SocialLink>  
 </SocialLinks>  
 </FooterColumn>  
   
 <FooterColumn>  
 <FooterTitle>Quick Links</FooterTitle>  
 <FooterNav>  
 <FooterLink to="/">{t('home')}</FooterLink>  
 <FooterLink to="/products">{t('products')}</FooterLink>  
 <FooterLink to="/about">{t('about')}</FooterLink>  
 <FooterLink to="/contact">{t('contact')}</FooterLink>  
 </FooterNav>  
 </FooterColumn>  
   
 <FooterColumn>  
 <FooterTitle>Categories</FooterTitle>  
 <FooterNav>  
 <FooterLink to="/products?category=spray">All Over Spray</FooterLink>  
 <FooterLink to="/products?category=perfume">Perfumes</FooterLink>  
 <FooterLink to="/products?category=candle">Candles</FooterLink>  
 <FooterLink to="/products?category=gift">Gift Sets</FooterLink>  
 </FooterNav>  
 </FooterColumn>  
   
 <FooterColumn>  
 <FooterTitle>Contact</FooterTitle>  
 <ContactInfo>  
 <ContactItem>Email: info@sirmperfumes.com</ContactItem>  
 <ContactItem>Phone: +966 123 456 789</ContactItem>  
 <ContactItem>Address: Riyadh, Saudi Arabia</ContactItem>  
 </ContactInfo>  
 </FooterColumn>  
 </FooterContent>  
   
 <FooterBottom>  
 <Copyright>&copy; {new Date().getFullYear()} sirm Perfumes. All Rights Reserved.</Copyright>  
 <PaymentMethods>  
 <img src="/images/payment-methods.png" alt="Payment Methods" />  
 </PaymentMethods>  
 </FooterBottom>  
 </div>  
 </FooterContainer>  
 );  
};  
  
// Styled Components  
const FooterContainer = styled.footer`  
 background-color: var(--primary-color);  
 color: var(--text-color);  
 padding: 50px 0 20px;  
`;  
  
const FooterContent = styled.div`  
 display: grid;  
 grid-template-columns: 2fr 1fr 1fr 1fr;  
 gap: 30px;  
 margin-bottom: 30px;  
   
 @media (max-width: 768px) {  
 grid-template-columns: 1fr 1fr;  
 }  
   
 @media (max-width: 480px) {  
 grid-template-columns: 1fr;  
 }  
`;  
  
const FooterColumn = styled.div``;  
  
const FooterLogo = styled.div`  
 font-size: 26px;  
 font-weight: 700;  
 margin-bottom: 15px;  
 color: var(--secondary-color);  
`;  
  
const FooterDescription = styled.p`  
 font-size: 14px;  
 margin-bottom: 20px;  
 line-height: 1.6;  
 opacity: 0.8;  
`;  
  
const SocialLinks = styled.div`  
 display: flex;  
`;  
  
const SocialLink = styled.a`  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 width: 36px;  
 height: 36px;  
 background-color: rgba(255, 255, 255, 0.1);  
 color: var(--text-color);  
 border-radius: 50%;  
 margin-right: 10px;  
 transition: all 0.2s ease;  
   
 &:hover {  
 background-color: var(--secondary-color);  
 color: var(--primary-color);  
 }  
`;  
  
const FooterTitle = styled.h3`  
 font-size: 18px;  
 margin-bottom: 20px;  
 position: relative;  
   
 &:after {  
 content: '';  
 display: block;  
 width: 30px;  
 height: 2px;  
 background-color: var(--secondary-color);  
 margin-top: 8px;  
 }  
`;  
  
const FooterNav = styled.div`  
 display: flex;  
 flex-direction: column;  
`;  
  
const FooterLink = styled(Link)`  
 color: var(--text-color);  
 text-decoration: none;  
 margin-bottom: 10px;  
 opacity: 0.8;  
 transition: all 0.2s ease;  
 font-size: 14px;  
   
 &:hover {  
 opacity: 1;  
 color: var(--secondary-color);  
 transform: translateX(5px);  
 }  
   
 .rtl & {  
 &:hover {  
 transform: translateX(-5px);  
 }  
 }  
`;  
  
const ContactInfo = styled.div``;  
  
const ContactItem = styled.div`  
 margin-bottom: 10px;  
 font-size: 14px;  
 opacity: 0.8;  
`;  
  
const FooterBottom = styled.div`  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 padding-top: 20px;  
 border-top: 1px solid rgba(255, 255, 255, 0.1);  
   
 @media (max-width: 768px) {  
 flex-direction: column;  
 gap: 15px;  
 }  
`;  
  
const Copyright = styled.div`  
 font-size: 14px;  
 opacity: 0.7;  
`;  
  
const PaymentMethods = styled.div`  
 img {  
 height: 24px;  
 }  
`;  
  
export default Footer;

File Path: .\src\components\GuestRoute.jsx

Content:

// src/components/GuestRoute.jsx  
import { Navigate } from 'react-router-dom';  
import { useAuth } from '../context/AuthContext';  
  
const GuestRoute = ({ children }) => {  
 const { isAuthenticated, loading } = useAuth();  
   
 if (loading) {  
 return <div>Loading...</div>;  
 }  
  
 if (isAuthenticated) {  
 return <Navigate to="/" replace />;  
 }  
  
 return children;  
};  
  
export default GuestRoute;

File Path: .\src\components\Header.jsx

Content:

import { useState } from 'react';  
import { Link, useNavigate } from 'react-router-dom';  
import { useTranslation } from 'react-i18next';  
import {   
 FaShoppingCart,   
 FaUser,   
 FaSearch,   
 FaBars,   
 FaTimes,   
 FaSignOutAlt,   
 FaUserCog,   
 FaListAlt,   
 FaHeart,   
 FaCog,   
 FaTachometerAlt   
} from 'react-icons/fa';  
import styled from 'styled-components';  
import { useAuth } from '../context/AuthContext';  
import { useCart } from '../context/CartContext';  
  
const Header = ({ changeLanguage, currentLang }) => {  
 const { t } = useTranslation();  
 const navigate = useNavigate();  
 const [searchQuery, setSearchQuery] = useState('');  
 const [isMenuOpen, setIsMenuOpen] = useState(false);  
 const { isAuthenticated, isAdmin, currentUser, logout } = useAuth();  
 const { cartCount } = useCart();  
   
 const [userMenuOpen, setUserMenuOpen] = useState(false);  
  
 const handleSearch = (e) => {  
 e.preventDefault();  
 if (searchQuery.trim()) {  
 navigate(`/products?search=${searchQuery}`);  
 }  
 };  
  
 const handleLogout = () => {  
 logout();  
 navigate('/');  
 setUserMenuOpen(false);  
 };  
  
 return (  
 <HeaderContainer>  
 <div className="container">  
 <HeaderInner>  
 {/\* Logo Section \*/}  
 {/\* <Logo>  
 <Link to="/">sirm</Link>  
 </Logo> \*/}  
 <Logo>  
 <Link to="/">  
 <LogoImage src="/images/sirm-logo.png" alt="sirm" />  
 </Link>  
 </Logo>  
  
 {/\* Mobile Menu Button \*/}  
 <MobileMenuButton onClick={() => setIsMenuOpen(!isMenuOpen)}>  
 {isMenuOpen ? <FaTimes /> : <FaBars />}  
 </MobileMenuButton>  
  
 {/\* Navigation Links \*/}  
 <Navigation className={isMenuOpen ? 'active' : ''}>  
 <NavLinks>  
 <NavItem>  
 <Link to="/">{t('home')}</Link>  
 </NavItem>  
 <NavItem>  
 <Link to="/products">{t('products')}</Link>  
 </NavItem>  
 <NavItem>  
 <Link to="/about">{t('about')}</Link>  
 </NavItem>  
 <NavItem>  
 <Link to="/contact">{t('contact')}</Link>  
 </NavItem>  
   
 {/\* Admin specific navigation item \*/}  
 {isAdmin && (  
 <NavItem isAdmin={true}>  
 <Link to="/admin">  
 <FaTachometerAlt style={{ marginRight: '5px' }} />  
 {t('admin\_panel') || 'Admin Panel'}  
 </Link>  
 </NavItem>  
 )}  
 </NavLinks>  
 </Navigation>  
  
 {/\* Right Side Actions \*/}  
 <HeaderActions>  
 {/\* Language Toggle \*/}  
 <LangToggle onClick={() => changeLanguage(currentLang === 'en' ? 'ar' : 'en')}>  
 {currentLang === 'en' ? 'عربي' : 'EN'}  
 </LangToggle>  
  
 {/\* Search \*/}  
 <SearchForm onSubmit={handleSearch}>  
 <SearchInput  
 type="text"  
 placeholder={t('search')}  
 value={searchQuery}  
 onChange={(e) => setSearchQuery(e.target.value)}  
 />  
 <SearchButton type="submit">  
 <FaSearch />  
 </SearchButton>  
 </SearchForm>  
  
 {/\* Cart \*/}  
 <IconLink to="/cart">  
 <CartIcon>  
 <FaShoppingCart />  
 {cartCount > 0 && <CartBadge>{cartCount}</CartBadge>}  
 </CartIcon>  
 </IconLink>  
  
 {/\* User Menu - Different based on auth state \*/}  
 {isAuthenticated ? (  
 <UserMenuContainer>  
 <UserMenuTrigger   
 onClick={() => setUserMenuOpen(!userMenuOpen)}  
 isAdmin={isAdmin}  
 >  
 {isAdmin ? <FaUserCog /> : <FaUser />}  
 <UserName>{currentUser?.name?.split(' ')[0]}</UserName>  
 </UserMenuTrigger>  
   
 {userMenuOpen && (  
 <UserMenuDropdown onClick={(e) => e.stopPropagation()}>  
 {/\* User details section \*/}  
 <UserInfoSection>  
 <UserAvatar>{currentUser?.name?.charAt(0).toUpperCase()}</UserAvatar>  
 <UserDetails>  
 <UserFullName>{currentUser?.name}</UserFullName>  
 <UserEmail>{currentUser?.email}</UserEmail>  
 </UserDetails>  
 </UserInfoSection>  
   
 <UserMenuDivider />  
   
 {/\* Common user options \*/}  
 <UserMenuItem onClick={() => {  
 navigate('/profile');  
 setUserMenuOpen(false);  
 }}>  
 <FaUser style={{marginRight: '10px'}} />  
 {t('profile') || 'Profile'}  
 </UserMenuItem>  
   
 <UserMenuItem onClick={() => {  
 navigate('/orders');  
 setUserMenuOpen(false);  
 }}>  
 <FaListAlt style={{marginRight: '10px'}} />  
 {t('my\_orders') || 'My Orders'}  
 </UserMenuItem>  
   
 <UserMenuItem onClick={() => {  
 navigate('/wishlist');  
 setUserMenuOpen(false);  
 }}>  
 <FaHeart style={{marginRight: '10px'}} />  
 {t('wishlist') || 'Wishlist'}  
 </UserMenuItem>  
   
 <UserMenuItem onClick={() => {  
 navigate('/settings');  
 setUserMenuOpen(false);  
 }}>  
 <FaCog style={{marginRight: '10px'}} />  
 {t('settings') || 'Settings'}  
 </UserMenuItem>  
   
 {/\* Admin-specific options \*/}  
 {isAdmin && (  
 <>  
 <UserMenuDivider />  
 <UserMenuSectionTitle>{t('admin\_options') || 'Admin Options'}</UserMenuSectionTitle>  
   
 <UserMenuItem onClick={() => {  
 navigate('/admin');  
 setUserMenuOpen(false);  
 }} isAdmin={true}>  
 <FaTachometerAlt style={{marginRight: '10px'}} />  
 {t('admin\_dashboard') || 'Admin Dashboard'}  
 </UserMenuItem>  
   
 <UserMenuItem onClick={() => {  
 navigate('/admin/products');  
 setUserMenuOpen(false);  
 }} isAdmin={true}>  
 <FaListAlt style={{marginRight: '10px'}} />  
 {t('manage\_products') || 'Manage Products'}  
 </UserMenuItem>  
   
 <UserMenuItem onClick={() => {  
 navigate('/admin/orders');  
 setUserMenuOpen(false);  
 }} isAdmin={true}>  
 <FaShoppingCart style={{marginRight: '10px'}} />  
 {t('manage\_orders') || 'Manage Orders'}  
 </UserMenuItem>  
 </>  
 )}  
   
 <UserMenuDivider />  
   
 {/\* Logout option \*/}  
 <UserMenuItem onClick={handleLogout}>  
 <FaSignOutAlt style={{marginRight: '10px'}} />  
 {t('logout') || 'Logout'}  
 </UserMenuItem>  
 </UserMenuDropdown>  
 )}  
 </UserMenuContainer>  
 ) : (  
 /\* Login/Register buttons for non-authenticated users \*/  
 <AuthButtons>  
 <LoginButton to="/login">  
 {t('login')}  
 </LoginButton>  
 <RegisterButton to="/register">  
 {t('register') || 'Register'}  
 </RegisterButton>  
 </AuthButtons>  
 )}  
 </HeaderActions>  
 </HeaderInner>  
 </div>  
 </HeaderContainer>  
 );  
};  
  
// Styled Components  
const HeaderContainer = styled.header`  
 // background-color: var(--primary-color);  
 background-color: #d3b58d; /\* Changed from var(--primary-color) to beige \*/  
 padding: 15px 0;  
 box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);  
 position: sticky;  
 top: 0;  
 z-index: 1000;  
`;  
  
const HeaderInner = styled.div`  
 display: flex;  
 align-items: center;  
 justify-content: space-between;  
   
 @media (max-width: 768px) {  
 flex-wrap: wrap;  
 }  
`;  
  
// const Logo = styled.div`  
// font-size: 28px;  
// font-weight: 700;  
// letter-spacing: 1px;  
   
// a {  
// color: var(--secondary-color);  
// text-decoration: none;  
// transition: color var(--transition-speed) ease;  
   
// &:hover {  
// color: var(--accent-color);  
// }  
// }  
// `;  
  
const Logo = styled.div`  
 display: flex;  
 align-items: center;  
 margin-right: 15px; // Add space between logo and navigation  
   
 a {  
 text-decoration: none;  
 display: flex;  
 align-items: center;  
 }  
`;  
  
const LogoImage = styled.img`  
 height: 38px; // Adjust based on your navbar height  
 max-width: 100%;  
 display: block;  
`;  
  
const Navigation = styled.nav`  
 @media (max-width: 768px) {  
 width: 100%;  
 max-height: 0;  
 overflow: hidden;  
 transition: max-height var(--transition-speed) ease;  
 order: 3;  
   
 &.active {  
 max-height: 300px;  
 }  
 }  
`;  
  
const NavLinks = styled.ul`  
 display: flex;  
 list-style: none;  
   
 @media (max-width: 768px) {  
 flex-direction: column;  
 padding: 15px 0;  
 }  
`;  
  
const NavItem = styled.li`  
 margin: 0 15px;  
   
 a {  
 // color: ${props => props.isAdmin ? 'var(--secondary-color)' : 'var(--text-color)'};  
 color: ${props => props.isAdmin ? 'var(--primary-color)' : 'var(--text-color)'};  
 text-decoration: none;  
 font-weight: ${props => props.isAdmin ? '600' : '500'};  
 transition: color var(--transition-speed) ease;  
 display: flex;  
 align-items: center;  
   
 &:hover {  
 // color: var(--secondary-color);  
 color: var(--dark-accent);  
 }  
 }  
   
 @media (max-width: 768px) {  
 margin: 10px 0;  
 }  
`;  
  
const HeaderActions = styled.div`  
 display: flex;  
 align-items: center;  
`;  
  
const LangToggle = styled.button`  
 background: none;  
 // border: 1px solid var(--secondary-color);  
 // color: var(--secondary-color);  
 border: 1px solid var(--primary-color);  
 color: var(--primary-color);  
 padding: 5px 10px;  
 border-radius: var(--border-radius);  
 margin-right: 15px;  
 font-size: 14px;  
   
 &:hover {  
 // background-color: var(--secondary-color);  
 // color: var(--primary-color);  
 background-color: var(--primary-color);  
 color: #d3b58d;  
 }  
`;  
  
const SearchForm = styled.form`  
 display: flex;  
 position: relative;  
 margin-right: 15px;  
`;  
  
const SearchInput = styled.input`  
 padding: 8px 35px 8px 15px;  
 border: none;  
 border-radius: 20px;  
 // background-color: rgba(255, 255, 255, 0.1);  
 background-color: rgba(56, 40, 29, 0.1); /\* Changed color \*/  
 color: var(--text-color);  
 font-size: 14px;  
 width: 150px;  
 transition: all var(--transition-speed) ease;  
   
 &:focus {  
 outline: none;  
 background-color: rgba(255, 255, 255, 0.2);  
 width: 180px;  
 }  
   
 &::placeholder {  
 color: rgba(255, 255, 255, 0.6);  
 }  
`;  
  
const SearchButton = styled.button`  
 position: absolute;  
 right: 10px;  
 top: 50%;  
 transform: translateY(-50%);  
 background: none;  
 border: none;  
 color: var(--text-color);  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 font-size: 14px;  
`;  
  
const IconLink = styled(Link)`  
 color: var(--text-color);  
 margin-left: 15px;  
 font-size: 18px;  
 display: flex;  
 transition: color var(--transition-speed) ease;  
   
 &:hover {  
 color: var(--secondary-color);  
 }  
`;  
  
const MobileMenuButton = styled.button`  
 display: none;  
 background: none;  
 border: none;  
 color: var(--text-color);  
 font-size: 24px;  
   
 @media (max-width: 768px) {  
 display: block;  
 order: 2;  
 }  
`;  
  
const CartIcon = styled.div`  
 position: relative;  
 display: flex;  
`;  
  
const CartBadge = styled.span`  
 position: absolute;  
 top: -8px;  
 right: -8px;  
 background-color: var(--secondary-color);  
 color: var(--primary-color);  
 font-size: 10px;  
 font-weight: bold;  
 height: 16px;  
 width: 16px;  
 border-radius: 50%;  
 display: flex;  
 align-items: center;  
 justify-content: center;  
`;  
  
const UserMenuContainer = styled.div`  
 position: relative;  
 margin-left: 15px;  
`;  
  
const UserMenuTrigger = styled.div`  
 display: flex;  
 align-items: center;  
 // color: ${props => props.isAdmin ? 'var(--secondary-color)' : 'var(--text-color)'};  
 // color: ${props => props.isAdmin ? 'var(--primary-color)' : 'var(--text-color)'};  
 font-weight: ${props => props.isAdmin ? '600' : 'normal'};  
 cursor: pointer;  
 transition: color var(--transition-speed) ease;  
 padding: 5px 10px;  
 border-radius: 20px;  
 background-color: ${props => props.isAdmin ? 'rgba(212, 175, 55, 0.1)' : 'transparent'};  
   
 &:hover {  
 color: var(--dark-accent);  
 background-color: rgba(212, 175, 55, 0.1);  
 }  
`;  
  
const UserName = styled.span`  
 margin-left: 5px;  
 display: none;  
   
 @media (min-width: 992px) {  
 display: inline;  
 }  
`;  
  
const UserMenuDropdown = styled.div`  
 position: absolute;  
 top: calc(100% + 10px);  
 right: 0;  
 width: 250px;  
 background-color: white;  
 border-radius: 8px;  
 box-shadow: 0 5px 25px rgba(0, 0, 0, 0.15);  
 padding: 10px 0;  
 z-index: 100;  
   
 .rtl & {  
 right: auto;  
 left: 0;  
 }  
`;  
  
const UserInfoSection = styled.div`  
 display: flex;  
 align-items: center;  
 padding: 15px;  
`;  
  
const UserAvatar = styled.div`  
 width: 40px;  
 height: 40px;  
 border-radius: 50%;  
 background-color: var(--primary-color);  
 color: var(--text-color);  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 font-weight: 600;  
 font-size: 18px;  
 margin-right: 10px;  
`;  
  
const UserDetails = styled.div`  
 flex: 1;  
`;  
  
const UserFullName = styled.div`  
 font-weight: 600;  
 color: var(--primary-color);  
`;  
  
const UserEmail = styled.div`  
 font-size: 12px;  
 color: #777;  
`;  
  
const UserMenuItem = styled.div`  
 padding: 10px 15px;  
 // color: ${props => props.isAdmin ? 'var(--secondary-color)' : 'var(--primary-color)'};  
 color: ${props => props.isAdmin ? 'var(--secondary-color)' : 'var(--primary-color)'};  
 cursor: pointer;  
 display: flex;  
 align-items: center;  
 font-weight: ${props => props.isAdmin ? '600' : 'normal'};  
   
 &:hover {  
 background-color: ${props => props.isAdmin ? 'rgba(212, 175, 55, 0.1)' : '#f5f5f5'};  
 color: var(--secondary-color);  
 }  
`;  
  
const UserMenuDivider = styled.div`  
 height: 1px;  
 background-color: #eee;  
 margin: 5px 0;  
`;  
  
const UserMenuSectionTitle = styled.div`  
 padding: 5px 15px;  
 font-size: 12px;  
 color: #777;  
 text-transform: uppercase;  
 letter-spacing: 1px;  
`;  
  
const AuthButtons = styled.div`  
 display: flex;  
 align-items: center;  
`;  
  
const LoginButton = styled(Link)`  
 color: var(--text-color);  
 text-decoration: none;  
 margin-right: 10px;  
 transition: color var(--transition-speed) ease;  
   
 &:hover {  
 color: var(--secondary-color);  
 }  
`;  
  
const RegisterButton = styled(Link)`  
 background-color: var(--secondary-color);  
 color: var(--primary-color);  
 text-decoration: none;  
 padding: 5px 12px;  
 border-radius: 4px;  
 font-weight: 500;  
 transition: background-color var(--transition-speed) ease;  
   
 &:hover {  
 background-color: var(--accent-color);  
 }  
`;  
  
export default Header;

File Path: .\src\components\Loading.jsx

Content:

import styled, { keyframes } from 'styled-components';  
import { useApp } from '../context/AppContext';  
  
const Loading = () => {  
 const { isLoading } = useApp();  
  
 if (!isLoading) {  
 return null;  
 }  
  
 return (  
 <LoadingOverlay>  
 <LoadingSpinner />  
 </LoadingOverlay>  
 );  
};  
  
// Styled Components  
const spin = keyframes`  
 0% {  
 transform: rotate(0deg);  
 }  
 100% {  
 transform: rotate(360deg);  
 }  
`;  
  
const LoadingOverlay = styled.div`  
 position: fixed;  
 top: 0;  
 left: 0;  
 right: 0;  
 bottom: 0;  
 background-color: rgba(0, 0, 0, 0.5);  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 z-index: 9999;  
`;  
  
const LoadingSpinner = styled.div`  
 width: 50px;  
 height: 50px;  
 border: 5px solid rgba(255, 255, 255, 0.3);  
 border-radius: 50%;  
 border-top-color: var(--secondary-color);  
 animation: ${spin} 1s ease-in-out infinite;  
`;  
  
export default Loading;

File Path: .\src\components\Notifications.jsx

Content:

import { useEffect } from 'react';  
import styled, { keyframes } from 'styled-components';  
import { FaCheckCircle, FaInfoCircle, FaExclamationTriangle, FaTimesCircle } from 'react-icons/fa';  
import { useApp } from '../context/AppContext';  
  
const Notifications = () => {  
 const { notifications, removeNotification } = useApp();  
  
 useEffect(() => {  
 // Clean up all notifications on component unmount  
 return () => {  
 notifications.forEach(notification => {  
 removeNotification(notification.id);  
 });  
 };  
 }, []);  
  
 if (notifications.length === 0) {  
 return null;  
 }  
  
 return (  
 <NotificationsContainer>  
 {notifications.map(notification => (  
 <Notification   
 key={notification.id}  
 type={notification.type}  
 >  
 <NotificationIcon type={notification.type}>  
 {notification.type === 'success' && <FaCheckCircle />}  
 {notification.type === 'info' && <FaInfoCircle />}  
 {notification.type === 'warning' && <FaExclamationTriangle />}  
 {notification.type === 'error' && <FaTimesCircle />}  
 </NotificationIcon>  
 <NotificationMessage>{notification.message}</NotificationMessage>  
 <CloseButton onClick={() => removeNotification(notification.id)}>  
 &times;  
 </CloseButton>  
 </Notification>  
 ))}  
 </NotificationsContainer>  
 );  
};  
  
// Styled Components  
const slideIn = keyframes`  
 from {  
 transform: translateX(100%);  
 opacity: 0;  
 }  
 to {  
 transform: translateX(0);  
 opacity: 1;  
 }  
`;  
  
const NotificationsContainer = styled.div`  
 position: fixed;  
 top: 20px;  
 right: 20px;  
 z-index: 9999;  
 display: flex;  
 flex-direction: column;  
 gap: 10px;  
 max-width: 350px;  
   
 .rtl & {  
 right: auto;  
 left: 20px;  
 }  
`;  
  
const Notification = styled.div`  
 display: flex;  
 align-items: center;  
 padding: 15px;  
 border-radius: 4px;  
 background-color: white;  
 box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);  
 animation: ${slideIn} 0.3s ease;  
   
 ${props => {  
 switch (props.type) {  
 case 'success':  
 return `border-left: 4px solid #2ecc71;`;  
 case 'info':  
 return `border-left: 4px solid #3498db;`;  
 case 'warning':  
 return `border-left: 4px solid #f39c12;`;  
 case 'error':  
 return `border-left: 4px solid #e74c3c;`;  
 default:  
 return `border-left: 4px solid #3498db;`;  
 }  
 }}  
   
 .rtl & {  
 border-left: none;  
   
 ${props => {  
 switch (props.type) {  
 case 'success':  
 return `border-right: 4px solid #2ecc71;`;  
 case 'info':  
 return `border-right: 4px solid #3498db;`;  
 case 'warning':  
 return `border-right: 4px solid #f39c12;`;  
 case 'error':  
 return `border-right: 4px solid #e74c3c;`;  
 default:  
 return `border-right: 4px solid #3498db;`;  
 }  
 }}  
 }  
`;  
  
const NotificationIcon = styled.div`  
 margin-right: 10px;  
 font-size: 20px;  
   
 ${props => {  
 switch (props.type) {  
 case 'success':  
 return `color: #2ecc71;`;  
 case 'info':  
 return `color: #3498db;`;  
 case 'warning':  
 return `color: #f39c12;`;  
 case 'error':  
 return `color: #e74c3c;`;  
 default:  
 return `color: #3498db;`;  
 }  
 }}  
   
 .rtl & {  
 margin-right: 0;  
 margin-left: 10px;  
 }  
`;  
  
const NotificationMessage = styled.div`  
 flex: 1;  
 font-size: 14px;  
`;  
  
const CloseButton = styled.button`  
 background: none;  
 border: none;  
 font-size: 18px;  
 color: #999;  
 cursor: pointer;  
 padding: 0 5px;  
 margin-left: 10px;  
   
 &:hover {  
 color: #555;  
 }  
   
 .rtl & {  
 margin-left: 0;  
 margin-right: 10px;  
 }  
`;  
  
export default Notifications;

File Path: .\src\components\PrivateRoute.jsx

Content:

import { Navigate, useLocation } from 'react-router-dom';  
import { useAuth } from '../context/AuthContext';  
  
const PrivateRoute = ({ children }) => {  
 const { isAuthenticated, loading } = useAuth();  
 const location = useLocation();  
  
 if (loading) {  
 return <div>Loading...</div>;  
 }  
  
 if (!isAuthenticated) {  
 // Redirect to login and save the intended destination  
 return <Navigate to={`/login?redirect=${location.pathname}`} replace />;  
 }  
  
 return children;  
};  
  
export default PrivateRoute;

File Path: .\src\components\ProductCard.jsx

Content:

import { useNavigate } from 'react-router-dom';  
import { useState } from 'react';  
import styled from 'styled-components';  
import { motion } from 'framer-motion';  
import { FaHeart, FaRegHeart, FaShoppingCart } from 'react-icons/fa';  
import { useTranslation } from 'react-i18next';  
  
const ProductCard = ({ product, isRTL }) => {  
 const { t } = useTranslation();  
 const navigate = useNavigate();  
 const [isFavorite, setIsFavorite] = useState(false);  
 const [isHovered, setIsHovered] = useState(false);  
  
 const toggleFavorite = (e) => {  
 e.stopPropagation();  
 setIsFavorite(!isFavorite);  
 };  
  
 const addToCart = (e) => {  
 e.stopPropagation();  
 // Add cart functionality here  
 console.log(`Added ${product.name} to cart`);  
 };  
  
 const goToProductDetail = () => {  
 navigate(`/product/${product.\_id}`);  
 };  
  
 return (  
 <CardContainer  
 onClick={goToProductDetail}  
 onMouseEnter={() => setIsHovered(true)}  
 onMouseLeave={() => setIsHovered(false)}  
 whileHover={{   
 y: -5,  
 boxShadow: '0 10px 20px rgba(0, 0, 0, 0.15)'  
 }}  
 initial={{ opacity: 0, y: 20 }}  
 animate={{ opacity: 1, y: 0 }}  
 transition={{ duration: 0.3 }}  
 >  
 <FavoriteButton onClick={toggleFavorite}>  
 {isFavorite ? <FaHeart color="#e74c3c" /> : <FaRegHeart />}  
 </FavoriteButton>  
   
 <ImageContainer>  
 <ProductImage src={product.image} alt={product.name} />  
 <ProductBox src={product.boxImage} alt={`${product.name} Box`} />  
 </ImageContainer>  
   
 {isHovered && (  
 <CardOverlay  
 initial={{ opacity: 0 }}  
 animate={{ opacity: 1 }}  
 exit={{ opacity: 0 }}  
 >  
 <AddToCartButton onClick={addToCart} isRTL={isRTL}>  
 <CartIcon />  
 <span>{t('add\_to\_cart')}</span>  
 </AddToCartButton>  
 </CardOverlay>  
 )}  
   
 <ProductInfo>  
 <ProductName>{product.name}</ProductName>  
 <ProductPrice>  
 {product.currency} {product.price.toFixed(2)}  
 </ProductPrice>  
 </ProductInfo>  
 </CardContainer>  
 );  
};  
  
// Styled Components  
const CardContainer = styled(motion.div)`  
 background-color: white;  
 border-radius: 8px;  
 overflow: hidden;  
 box-shadow: var(--card-shadow);  
 position: relative;  
 cursor: pointer;  
 height: 100%;  
 display: flex;  
 flex-direction: column;  
`;  
  
const FavoriteButton = styled.button`  
 position: absolute;  
 top: 10px;  
 left: 10px;  
 z-index: 10;  
 background: none;  
 border: none;  
 font-size: 20px;  
 color: #888;  
 cursor: pointer;  
   
 .rtl & {  
 left: auto;  
 right: 10px;  
 }  
`;  
  
const ImageContainer = styled.div`  
 position: relative;  
 padding-top: 100%;  
 background-color: #f9f9f9;  
 display: flex;  
 justify-content: center;  
 align-items: center;  
`;  
  
const ProductImage = styled.img`  
 position: absolute;  
 top: 50%;  
 left: 30%;  
 transform: translate(-50%, -50%);  
 height: 80%;  
 object-fit: contain;  
 transition: transform 0.3s ease;  
   
 ${CardContainer}:hover & {  
 transform: translate(-60%, -50%) scale(1.05);  
 }  
`;  
  
const ProductBox = styled.img`  
 position: absolute;  
 top: 50%;  
 right: 30%;  
 transform: translate(50%, -50%);  
 height: 80%;  
 object-fit: contain;  
 transition: transform 0.3s ease;  
   
 ${CardContainer}:hover & {  
 transform: translate(60%, -50%) scale(1.05);  
 }  
`;  
  
const CardOverlay = styled(motion.div)`  
 position: absolute;  
 bottom: 0;  
 left: 0;  
 right: 0;  
 padding: 15px;  
 background: linear-gradient(to top, rgba(0, 0, 0, 0.7), transparent);  
 display: flex;  
 justify-content: center;  
`;  
  
const AddToCartButton = styled.button`  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 padding: 8px 15px;  
 background-color: var(--secondary-color);  
 color: #111;  
 border: none;  
 border-radius: 4px;  
 font-weight: 600;  
 transition: all 0.2s ease;  
   
 &:hover {  
 background-color: var(--accent-color);  
 }  
   
 span {  
 margin: ${props => props.isRTL ? '0 0 0 8px' : '0 8px 0 0'};  
 }  
`;  
  
const CartIcon = styled(FaShoppingCart)`  
 font-size: 16px;  
`;  
  
const ProductInfo = styled.div`  
 padding: 15px;  
 text-align: center;  
 flex-grow: 1;  
 display: flex;  
 flex-direction: column;  
 justify-content: space-between;  
`;  
  
const ProductName = styled.h3`  
 font-size: 16px;  
 font-weight: 500;  
 margin-bottom: 8px;  
 color: #333;  
`;  
  
const ProductPrice = styled.div`  
 font-size: 18px;  
 font-weight: 600;  
 color: var(--secondary-color);  
`;  
  
export default ProductCard;

File Path: .\src\components\admin\Sidebar.jsx

Content:

import { Link, useLocation } from 'react-router-dom';  
import styled from 'styled-components';  
import { FaTachometerAlt, FaBox, FaShoppingCart, FaUsers, FaCog, FaSignOutAlt } from 'react-icons/fa';  
  
const AdminSidebar = () => {  
 const location = useLocation();  
   
 const menuItems = [  
 { path: '/admin', label: 'Dashboard', icon: <FaTachometerAlt /> },  
 { path: '/admin/products', label: 'Products', icon: <FaBox /> },  
 { path: '/admin/orders', label: 'Orders', icon: <FaShoppingCart /> },  
 { path: '/admin/customers', label: 'Customers', icon: <FaUsers /> },  
 { path: '/admin/settings', label: 'Settings', icon: <FaCog /> },  
 ];  
   
 return (  
 <SidebarContainer>  
 <SidebarHeader>  
 <Logo>sirm</Logo>  
 <AdminTag>Admin Panel</AdminTag>  
 </SidebarHeader>  
   
 <SidebarMenu>  
 {menuItems.map((item) => (  
 <MenuItem key={item.path} active={location.pathname === item.path}>  
 <MenuLink to={item.path}>  
 <MenuIcon>{item.icon}</MenuIcon>  
 <MenuLabel>{item.label}</MenuLabel>  
 </MenuLink>  
 </MenuItem>  
 ))}  
 </SidebarMenu>  
   
 <SidebarFooter>  
 <LogoutButton>  
 <FaSignOutAlt />  
 <span>Logout</span>  
 </LogoutButton>  
 <AdminInfo>  
 <AdminName>Admin User</AdminName>  
 <AdminRole>Super Admin</AdminRole>  
 </AdminInfo>  
 </SidebarFooter>  
 </SidebarContainer>  
 );  
};  
  
// Styled Components  
const SidebarContainer = styled.aside`  
 width: 250px;  
 background-color: var(--primary-color);  
 color: var(--text-color);  
 display: flex;  
 flex-direction: column;  
 height: 100vh;  
 position: sticky;  
 top: 0;  
   
 @media (max-width: 768px) {  
 width: 70px;  
 }  
`;  
  
const SidebarHeader = styled.div`  
 padding: 20px;  
 border-bottom: 1px solid rgba(255, 255, 255, 0.1);  
   
 @media (max-width: 768px) {  
 padding: 15px 10px;  
 text-align: center;  
 }  
`;  
  
const Logo = styled.div`  
 font-size: 24px;  
 font-weight: 700;  
 color: var(--secondary-color);  
 margin-bottom: 5px;  
`;  
  
const AdminTag = styled.div`  
 font-size: 12px;  
 opacity: 0.7;  
   
 @media (max-width: 768px) {  
 display: none;  
 }  
`;  
  
const SidebarMenu = styled.ul`  
 list-style: none;  
 padding: 0;  
 margin: 0;  
 flex: 1;  
`;  
  
const MenuItem = styled.li`  
 margin: 5px 0;  
   
 a {  
 display: flex;  
 align-items: center;  
 padding: 12px 20px;  
 color: ${props => props.active ? 'var(--secondary-color)' : 'var(--text-color)'};  
 background-color: ${props => props.active ? 'rgba(255, 255, 255, 0.1)' : 'transparent'};  
 border-left: ${props => props.active ? '4px solid var(--secondary-color)' : '4px solid transparent'};  
 transition: all 0.2s ease;  
   
 &:hover {  
 background-color: rgba(255, 255, 255, 0.1);  
 color: var(--secondary-color);  
 }  
 }  
   
 @media (max-width: 768px) {  
 a {  
 padding: 15px 0;  
 justify-content: center;  
 }  
 }  
`;  
  
const MenuLink = styled(Link)`  
 text-decoration: none;  
 width: 100%;  
`;  
  
const MenuIcon = styled.span`  
 display: flex;  
 align-items: center;  
 font-size: 18px;  
 margin-right: 10px;  
   
 @media (max-width: 768px) {  
 margin-right: 0;  
 }  
`;  
  
const MenuLabel = styled.span`  
 @media (max-width: 768px) {  
 display: none;  
 }  
`;  
  
const SidebarFooter = styled.div`  
 padding: 15px 20px;  
 border-top: 1px solid rgba(255, 255, 255, 0.1);  
 display: flex;  
 flex-direction: column;  
   
 @media (max-width: 768px) {  
 padding: 15px 10px;  
 }  
`;  
  
const LogoutButton = styled.button`  
 display: flex;  
 align-items: center;  
 background: none;  
 border: none;  
 color: var(--text-color);  
 opacity: 0.8;  
 padding: 8px 0;  
 cursor: pointer;  
   
 span {  
 margin-left: 10px;  
 }  
   
 &:hover {  
 opacity: 1;  
 color: #e74c3c;  
 }  
   
 @media (max-width: 768px) {  
 justify-content: center;  
   
 span {  
 display: none;  
 }  
 }  
`;  
  
const AdminInfo = styled.div`  
 margin-top: 15px;  
   
 @media (max-width: 768px) {  
 display: none;  
 }  
`;  
  
const AdminName = styled.div`  
 font-weight: 600;  
 font-size: 14px;  
`;  
  
const AdminRole = styled.div`  
 font-size: 12px;  
 opacity: 0.7;  
`;  
  
export default AdminSidebar;

File Path: .\src\context\AppContext.jsx

Content:

// In src/context/AppContext.jsx  
import { createContext, useState, useContext, useCallback } from 'react';  
  
export const AppContext = createContext();  
  
export const useApp = () => useContext(AppContext);  
  
export const AppProvider = ({ children }) => {  
 const [notifications, setNotifications] = useState([]);  
 const [isLoading, setIsLoading] = useState(false);  
  
 const addNotification = useCallback((message, type = 'info', timeout = 5000) => {  
 const id = Date.now();  
 const notification = { id, message, type };  
   
 setNotifications(prev => [...prev, notification]);  
   
 if (timeout > 0) {  
 setTimeout(() => {  
 removeNotification(id);  
 }, timeout);  
 }  
   
 return id;  
 }, []);  
  
 const removeNotification = useCallback((id) => {  
 setNotifications(prev => prev.filter(notification => notification.id !== id));  
 }, []);  
  
 const showLoading = useCallback(() => setIsLoading(true), []);  
 const hideLoading = useCallback(() => setIsLoading(false), []);  
  
 const value = {  
 notifications,  
 addNotification,  
 removeNotification,  
 isLoading,  
 showLoading,  
 hideLoading  
 };  
  
 return (  
 <AppContext.Provider value={value}>  
 {children}  
 </AppContext.Provider>  
 );  
};

File Path: .\src\context\AuthContext.jsx

Content:

import { createContext, useState, useEffect, useContext } from 'react';  
import axios from 'axios';  
import api from '../api/config';  
import { authService } from '../services';  
  
  
export const AuthContext = createContext();  
  
export const useAuth = () => useContext(AuthContext);  
  
export const AuthProvider = ({ children }) => {  
 const [currentUser, setCurrentUser] = useState(null);  
 const [loading, setLoading] = useState(true);  
 const [error, setError] = useState(null);  
  
 useEffect(() => {  
 const fetchUser = async () => {  
 try {  
 const token = localStorage.getItem('token');  
 if (!token) {  
 setLoading(false);  
 return;  
 }  
   
 const response = await api.get('/api/auth/me', {  
 headers: {  
 Authorization: `Bearer ${token}`  
 }  
 });  
   
 setCurrentUser(response.data);  
 setLoading(false);  
 } catch (err) {  
 console.error('Error fetching user:', err);  
 localStorage.removeItem('token');  
 setLoading(false);  
 }  
 };  
  
 fetchUser();  
 }, []);  
  
const login = async (email, password) => {  
 try {  
 setError(null);  
 console.log('Attempting login for:', email);  
   
 const response = await api.post('/api/auth/login', { email, password });  
 // const response = await authService.login(email, password);  
 console.log('Login response:', response.data);  
   
 if (response.data && response.data.token) {  
 localStorage.setItem('token', response.data.token);  
 setCurrentUser(response.data.user);  
 return response.data.user; // Return the user data including role  
 } else {  
 throw new Error('Invalid response format');  
 }  
 } catch (err) {  
 console.error('Login error:', err);  
 const errorMessage = err.response?.data?.message || 'Login failed';  
 setError(errorMessage);  
 throw err;  
 }  
};  
  
 const register = async (userData) => {  
 try {  
 setError(null);  
 const response = await api.post('/api/auth/register', userData);  
 localStorage.setItem('token', response.data.token);  
 setCurrentUser(response.data.user);  
 return response.data.user;  
 } catch (err) {  
 setError(err.response?.data?.message || 'Registration failed');  
 throw err;  
 }  
 };  
  
const logout = () => {  
 localStorage.removeItem('token');  
 setCurrentUser(null);  
   
 // Optionally make a server request to invalidate the token  
 api.post('/api/auth/logout')  
 .catch(err => console.error('Error during logout:', err));  
};  
  
 const updateProfile = async (userData) => {  
 try {  
 setError(null);  
 const token = localStorage.getItem('token');  
 const response = await api.put('/api/auth/profile', userData, {  
 headers: {  
 Authorization: `Bearer ${token}`  
 }  
 });  
   
 setCurrentUser(response.data);  
 return response.data;  
 } catch (err) {  
 setError(err.response?.data?.message || 'Profile update failed');  
 throw err;  
 }  
 };  
  
 const value = {  
 currentUser,  
 loading,  
 error,  
 login,  
 register,  
 logout,  
 updateProfile,  
 isAuthenticated: !!currentUser,  
 isAdmin: currentUser?.role === 'admin'  
 };  
  
 return (  
 <AuthContext.Provider value={value}>  
 {children}  
 </AuthContext.Provider>  
 );  
};

File Path: .\src\context\CartContext.jsx

Content:

import { createContext, useState, useEffect, useContext } from 'react';  
  
export const CartContext = createContext();  
  
export const useCart = () => useContext(CartContext);  
  
export const CartProvider = ({ children }) => {  
 const [cartItems, setCartItems] = useState([]);  
 const [cartTotal, setCartTotal] = useState(0);  
 const [cartCount, setCartCount] = useState(0);  
  
 // Load cart from localStorage on initial render  
 useEffect(() => {  
 const storedCart = localStorage.getItem('cart');  
 if (storedCart) {  
 try {  
 const parsedCart = JSON.parse(storedCart);  
 setCartItems(parsedCart);  
 } catch (error) {  
 console.error('Error parsing cart from localStorage:', error);  
 localStorage.removeItem('cart');  
 }  
 }  
 }, []);  
  
 // Update localStorage whenever cart changes  
 useEffect(() => {  
 localStorage.setItem('cart', JSON.stringify(cartItems));  
   
 // Calculate totals  
 const total = cartItems.reduce((sum, item) => sum + (item.price \* item.quantity), 0);  
 const count = cartItems.reduce((sum, item) => sum + item.quantity, 0);  
   
 setCartTotal(total);  
 setCartCount(count);  
 }, [cartItems]);  
  
 const addToCart = (product, quantity = 1) => {  
 setCartItems(prevItems => {  
 // Check if product already exists in cart  
 const existingItem = prevItems.find(item => item.\_id === product.\_id);  
   
 if (existingItem) {  
 // Update quantity if product exists  
 return prevItems.map(item =>   
 item.\_id === product.\_id   
 ? { ...item, quantity: item.quantity + quantity }   
 : item  
 );  
 } else {  
 // Add new item if product doesn't exist  
 return [...prevItems, { ...product, quantity }];  
 }  
 });  
 };  
  
 const removeFromCart = (productId) => {  
 setCartItems(prevItems => prevItems.filter(item => item.\_id !== productId));  
 };  
  
 const updateQuantity = (productId, quantity) => {  
 if (quantity <= 0) {  
 removeFromCart(productId);  
 return;  
 }  
   
 setCartItems(prevItems =>   
 prevItems.map(item =>   
 item.\_id === productId   
 ? { ...item, quantity }   
 : item  
 )  
 );  
 };  
  
 const clearCart = () => {  
 setCartItems([]);  
 };  
  
 const value = {  
 cartItems,  
 cartTotal,  
 cartCount,  
 addToCart,  
 removeFromCart,  
 updateQuantity,  
 clearCart  
 };  
  
 return (  
 <CartContext.Provider value={value}>  
 {children}  
 </CartContext.Provider>  
 );  
};

File Path: .\src\context\index.jsx

Content:

import { AuthProvider } from './AuthContext';  
import { CartProvider } from './CartContext';  
import { AppProvider } from './AppContext';  
  
export const AppProviders = ({ children }) => {  
 return (  
 <AppProvider>  
 <AuthProvider>  
 <CartProvider>  
 {children}  
 </CartProvider>  
 </AuthProvider>  
 </AppProvider>  
 );  
};

File Path: .\src\pages\Cart.jsx

Content:

import { useState, useEffect } from 'react';  
import { Link, useNavigate } from 'react-router-dom';  
import { useTranslation } from 'react-i18next';  
import styled from 'styled-components';  
import { FaTrash, FaArrowRight, FaArrowLeft } from 'react-icons/fa';  
import { useCart } from '../context/CartContext';  
import { useAuth } from '../context/AuthContext';  
  
const Cart = () => {  
 const { t, i18n } = useTranslation();  
 const navigate = useNavigate();  
 const { cartItems, cartTotal, updateQuantity, removeFromCart, clearCart } = useCart();  
 const { isAuthenticated } = useAuth();  
 const [shippingCost, setShippingCost] = useState(30);  
 const isRTL = i18n.language === 'ar';  
  
 const handleCheckout = () => {  
 if (isAuthenticated) {  
 navigate('/checkout');  
 } else {  
 navigate('/login?redirect=checkout');  
 }  
 };  
  
 if (cartItems.length === 0) {  
 return (  
 <div className="container">  
 <EmptyCartContainer>  
 <EmptyCartIcon>🛒</EmptyCartIcon>  
 <EmptyCartTitle>{t('empty\_cart')}</EmptyCartTitle>  
 <EmptyCartText>{t('empty\_cart\_message')}</EmptyCartText>  
 <ContinueShoppingButton to="/products">  
 {t('continue\_shopping')}  
 {isRTL ? <FaArrowLeft /> : <FaArrowRight />}  
 </ContinueShoppingButton>  
 </EmptyCartContainer>  
 </div>  
 );  
 }  
  
 return (  
 <div className="container">  
 <PageHeader>  
 <h1>{t('shopping\_cart')}</h1>  
 </PageHeader>  
   
 <CartContainer>  
 <CartItemsSection>  
 <CartItemsTable>  
 <thead>  
 <tr>  
 <th>{t('product')}</th>  
 <th>{t('price')}</th>  
 <th>{t('quantity')}</th>  
 <th>{t('total')}</th>  
 <th></th>  
 </tr>  
 </thead>  
 <tbody>  
 {cartItems.map(item => (  
 <tr key={item.\_id}>  
 <td>  
 <ProductInfo>  
 <ProductImage src={item.image} alt={item.name} />  
 <ProductName>{isRTL ? item.nameAr : item.name}</ProductName>  
 </ProductInfo>  
 </td>  
 <td>  
 <ProductPrice>{item.currency} {item.price.toFixed(2)}</ProductPrice>  
 </td>  
 <td>  
 <QuantityControls>  
 <QuantityButton   
 onClick={() => updateQuantity(item.\_id, item.quantity - 1)}  
 disabled={item.quantity <= 1}  
 >  
 -  
 </QuantityButton>  
 <QuantityInput   
 type="number"   
 value={item.quantity}   
 onChange={(e) => updateQuantity(item.\_id, parseInt(e.target.value))}   
 min="1"   
 />  
 <QuantityButton   
 onClick={() => updateQuantity(item.\_id, item.quantity + 1)}  
 >  
 +  
 </QuantityButton>  
 </QuantityControls>  
 </td>  
 <td>  
 <ProductTotal>{item.currency} {(item.price \* item.quantity).toFixed(2)}</ProductTotal>  
 </td>  
 <td>  
 <RemoveButton onClick={() => removeFromCart(item.\_id)}>  
 <FaTrash />  
 </RemoveButton>  
 </td>  
 </tr>  
 ))}  
 </tbody>  
 </CartItemsTable>  
   
 <CartActions>  
 <ContinueShopping to="/products">  
 {isRTL ? <FaArrowLeft /> : <FaArrowRight />}  
 {t('continue\_shopping')}  
 </ContinueShopping>  
 <ClearCartButton onClick={clearCart}>  
 {t('clear\_cart')}  
 </ClearCartButton>  
 </CartActions>  
 </CartItemsSection>  
   
 <CartSummary>  
 <SummaryTitle>{t('order\_summary')}</SummaryTitle>  
   
 <SummaryRow>  
 <SummaryLabel>{t('subtotal')}</SummaryLabel>  
 <SummaryValue>﷼ {cartTotal.toFixed(2)}</SummaryValue>  
 </SummaryRow>  
   
 <SummaryRow>  
 <SummaryLabel>{t('shipping')}</SummaryLabel>  
 <SummaryValue>﷼ {shippingCost.toFixed(2)}</SummaryValue>  
 </SummaryRow>  
   
 <SummaryTotal>  
 <SummaryLabel>{t('total')}</SummaryLabel>  
 <TotalValue>﷼ {(cartTotal + shippingCost).toFixed(2)}</TotalValue>  
 </SummaryTotal>  
   
 <CheckoutButton onClick={handleCheckout}>  
 {t('proceed\_to\_checkout')}  
 {isRTL ? <FaArrowLeft /> : <FaArrowRight />}  
 </CheckoutButton>  
   
 <AcceptedPayments>  
 <PaymentTitle>{t('we\_accept')}</PaymentTitle>  
 <PaymentMethods>  
 <img src="/images/payment-methods.png" alt="Accepted Payment Methods" />  
 </PaymentMethods>  
 </AcceptedPayments>  
 </CartSummary>  
 </CartContainer>  
 </div>  
 );  
};  
  
// Styled Components  
const PageHeader = styled.div`  
 margin: 30px 0;  
   
 h1 {  
 font-size: 28px;  
 color: var(--primary-color);  
 position: relative;  
 display: inline-block;  
   
 &:after {  
 content: '';  
 display: block;  
 width: 40px;  
 height: 2px;  
 background-color: var(--secondary-color);  
 margin-top: 10px;  
   
 .rtl & {  
 margin-left: auto;  
 }  
 }  
 }  
`;  
  
const CartContainer = styled.div`  
 display: grid;  
 grid-template-columns: 2fr 1fr;  
 gap: 30px;  
 margin-bottom: 50px;  
   
 @media (max-width: 768px) {  
 grid-template-columns: 1fr;  
 }  
`;  
  
const CartItemsSection = styled.div``;  
  
const CartItemsTable = styled.table`  
 width: 100%;  
 border-collapse: collapse;  
   
 th {  
 text-align: left;  
 padding: 15px 10px;  
 border-bottom: 1px solid #eee;  
 color: var(--primary-color);  
 font-weight: 600;  
   
 .rtl & {  
 text-align: right;  
 }  
 }  
   
 td {  
 padding: 15px 10px;  
 border-bottom: 1px solid #eee;  
   
 .rtl & {  
 text-align: right;  
 }  
 }  
   
 @media (max-width: 768px) {  
 display: block;  
   
 thead, tbody, tr, th, td {  
 display: block;  
 }  
   
 thead tr {  
 position: absolute;  
 top: -9999px;  
 left: -9999px;  
 }  
   
 tr {  
 border: 1px solid #eee;  
 margin-bottom: 20px;  
 position: relative;  
 padding: 10px;  
 }  
   
 td {  
 border: none;  
 position: relative;  
 padding-left: 40%;  
   
 &:before {  
 position: absolute;  
 top: 15px;  
 left: 10px;  
 width: 35%;  
 white-space: nowrap;  
 font-weight: 600;  
 }  
   
 &:nth-of-type(1):before { content: "${props => props.productLabel || 'Product'}"; }  
 &:nth-of-type(2):before { content: "${props => props.priceLabel || 'Price'}"; }  
 &:nth-of-type(3):before { content: "${props => props.quantityLabel || 'Quantity'}"; }  
 &:nth-of-type(4):before { content: "${props => props.totalLabel || 'Total'}"; }  
   
 .rtl & {  
 padding-left: 10px;  
 padding-right: 40%;  
   
 &:before {  
 left: auto;  
 right: 10px;  
 }  
 }  
 }  
 }  
`;  
  
const ProductInfo = styled.div`  
 display: flex;  
 align-items: center;  
   
 .rtl & {  
 flex-direction: row-reverse;  
 }  
`;  
  
const ProductImage = styled.img`  
 width: 60px;  
 height: 60px;  
 object-fit: contain;  
 margin-right: 15px;  
   
 .rtl & {  
 margin-right: 0;  
 margin-left: 15px;  
 }  
`;  
  
const ProductName = styled.div`  
 font-weight: 500;  
 color: var(--primary-color);  
`;  
  
const ProductPrice = styled.div`  
 font-weight: 500;  
`;  
  
const QuantityControls = styled.div`  
 display: flex;  
 align-items: center;  
   
 @media (max-width: 768px) {  
 justify-content: flex-end;  
   
 .rtl & {  
 justify-content: flex-start;  
 }  
 }  
`;  
  
const QuantityButton = styled.button`  
 width: 30px;  
 height: 30px;  
 background-color: #f1f1f1;  
 border: none;  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 cursor: pointer;  
   
 &:first-child {  
 border-radius: 4px 0 0 4px;  
   
 .rtl & {  
 border-radius: 0 4px 4px 0;  
 }  
 }  
   
 &:last-child {  
 border-radius: 0 4px 4px 0;  
   
 .rtl & {  
 border-radius: 4px 0 0 4px;  
 }  
 }  
   
 &:hover {  
 background-color: #e1e1e1;  
 }  
   
 &:disabled {  
 opacity: 0.5;  
 cursor: not-allowed;  
 }  
`;  
  
const QuantityInput = styled.input`  
 width: 40px;  
 height: 30px;  
 border: 1px solid #f1f1f1;  
 text-align: center;  
 -moz-appearance: textfield;  
   
 &::-webkit-outer-spin-button,  
 &::-webkit-inner-spin-button {  
 -webkit-appearance: none;  
 margin: 0;  
 }  
   
 &:focus {  
 outline: none;  
 border-color: var(--secondary-color);  
 }  
`;  
  
const ProductTotal = styled.div`  
 font-weight: 600;  
 color: var(--secondary-color);  
`;  
  
const RemoveButton = styled.button`  
 background: none;  
 border: none;  
 color: #999;  
 cursor: pointer;  
 transition: color 0.2s ease;  
   
 &:hover {  
 color: var(--danger-color);  
 }  
`;  
  
const CartActions = styled.div`  
 display: flex;  
 justify-content: space-between;  
 margin-top: 30px;  
   
 @media (max-width: 580px) {  
 flex-direction: column;  
 gap: 15px;  
 }  
`;  
  
const ContinueShopping = styled(Link)`  
 display: flex;  
 align-items: center;  
 color: var(--primary-color);  
 text-decoration: none;  
 transition: color 0.2s ease;  
   
 svg {  
 margin-right: 8px;  
   
 .rtl & {  
 margin-right: 0;  
 margin-left: 8px;  
 }  
 }  
   
 &:hover {  
 color: var(--secondary-color);  
 }  
`;  
  
const ClearCartButton = styled.button`  
 background: none;  
 border: 1px solid #ddd;  
 padding: 8px 15px;  
 color: #666;  
 cursor: pointer;  
 transition: all 0.2s ease;  
 border-radius: 4px;  
   
 &:hover {  
 border-color: var(--danger-color);  
 color: var(--danger-color);  
 }  
`;  
  
const CartSummary = styled.div`  
 background-color: #f9f9f9;  
 padding: 25px;  
 border-radius: 8px;  
 position: sticky;  
 top: 100px;  
`;  
  
const SummaryTitle = styled.h2`  
 font-size: 20px;  
 color: var(--primary-color);  
 margin-bottom: 20px;  
 padding-bottom: 15px;  
 border-bottom: 1px solid #eee;  
`;  
  
const SummaryRow = styled.div`  
 display: flex;  
 justify-content: space-between;  
 margin-bottom: 15px;  
   
 .rtl & {  
 flex-direction: row-reverse;  
 }  
`;  
  
const SummaryLabel = styled.div`  
 color: #666;  
`;  
  
const SummaryValue = styled.div`  
 font-weight: 500;  
`;  
  
const SummaryTotal = styled.div`  
 display: flex;  
 justify-content: space-between;  
 margin-top: 20px;  
 padding-top: 15px;  
 border-top: 1px solid #eee;  
 margin-bottom: 25px;  
   
 .rtl & {  
 flex-direction: row-reverse;  
 }  
`;  
  
const TotalValue = styled.div`  
 font-size: 20px;  
 font-weight: 600;  
 color: var(--secondary-color);  
`;  
  
const CheckoutButton = styled.button`  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 width: 100%;  
 padding: 12px;  
 background-color: var(--secondary-color);  
 color: var(--primary-color);  
 border: none;  
 border-radius: 4px;  
 font-size: 16px;  
 font-weight: 600;  
 cursor: pointer;  
 transition: all 0.3s ease;  
   
 svg {  
 margin-left: 10px;  
   
 .rtl & {  
 margin-left: 0;  
 margin-right: 10px;  
 }  
 }  
   
 &:hover {  
 background-color: var(--accent-color);  
 }  
`;  
  
const AcceptedPayments = styled.div`  
 margin-top: 25px;  
`;  
  
const PaymentTitle = styled.div`  
 font-size: 14px;  
 color: #666;  
 margin-bottom: 10px;  
`;  
  
const PaymentMethods = styled.div`  
 img {  
 max-width: 100%;  
 height: 24px;  
 }  
`;  
  
const EmptyCartContainer = styled.div`  
 text-align: center;  
 padding: 60px 0;  
`;  
  
const EmptyCartIcon = styled.div`  
 font-size: 60px;  
 margin-bottom: 20px;  
`;  
  
const EmptyCartTitle = styled.h2`  
 font-size: 24px;  
 color: var(--primary-color);  
 margin-bottom: 15px;  
`;  
  
const EmptyCartText = styled.p`  
 color: #666;  
 margin-bottom: 30px;  
`;  
  
const ContinueShoppingButton = styled(Link)`  
 display: inline-flex;  
 align-items: center;  
 padding: 12px 25px;  
 background-color: var(--secondary-color);  
 color: var(--primary-color);  
 text-decoration: none;  
 font-weight: 600;  
 border-radius: 4px;  
 transition: all 0.3s ease;  
   
 svg {  
 margin-left: 10px;  
   
 .rtl & {  
 margin-left: 0;  
 margin-right: 10px;  
 }  
 }  
   
 &:hover {  
 background-color: var(--accent-color);  
 }  
`;  
  
export default Cart;

File Path: .\src\pages\Home.jsx

Content:

import { useState, useEffect } from 'react';  
import { Link } from 'react-router-dom';  
import { useTranslation } from 'react-i18next';  
import styled from 'styled-components';  
import { motion } from 'framer-motion';  
import axios from 'axios';  
import ProductCard from '../components/ProductCard';  
import { FaArrowRight, FaArrowLeft } from 'react-icons/fa';  
import { productService } from '../services';  
  
  
const Home = () => {  
 const { t, i18n } = useTranslation();  
 const [featuredProducts, setFeaturedProducts] = useState([]);  
 const [newArrivals, setNewArrivals] = useState([]);  
 const [loading, setLoading] = useState(true);  
 const isRTL = i18n.language === 'ar';  
  
 useEffect(() => {  
 const fetchProducts = async () => {  
 try {  
 setLoading(true);  
   
 // Use the service instead of direct axios calls  
 const featuredData = await productService.getFeaturedProducts();  
 const newArrivalsData = await productService.getAllProducts({ limit: 4 });  
   
 console.log('Featured data:', featuredData);  
 console.log('New arrivals data:', newArrivalsData);  
   
 // Ensure we're setting arrays  
 setFeaturedProducts(Array.isArray(featuredData) ? featuredData : []);  
 setNewArrivals(Array.isArray(newArrivalsData) ? newArrivalsData : []);  
 } catch (err) {  
 console.error('Error fetching products:', err);  
 // Set empty arrays on error  
 setFeaturedProducts([]);  
 setNewArrivals([]);  
 } finally {  
 setLoading(false);  
 }  
 };  
  
 // // Mock data for development  
 // setTimeout(() => {  
 // const mockProducts = [  
 // {  
 // \_id: '1',  
 // name: 'Musk 100ml',  
 // nameAr: 'مسك 100مل',  
 // price: 230.00,  
 // currency: '﷼',  
 // image: '/images/musk\_bottle.png',  
 // boxImage: '/images/musk\_box.png',  
 // category: 'spray'  
 // },  
 // {  
 // \_id: '2',  
 // name: 'The fifth 100ml',  
 // nameAr: 'ذا فيفث 100مل',  
 // price: 230.00,  
 // currency: '﷼',  
 // image: '/images/fifth\_bottle.png',  
 // boxImage: '/images/fifth\_box.png',  
 // category: 'spray'  
 // },  
 // {  
 // \_id: '3',  
 // name: 'Flow of three 100ml',  
 // nameAr: 'فلو أوف ثري 100مل',  
 // price: 230.00,  
 // currency: '﷼',  
 // image: '/images/flow\_bottle.png',  
 // boxImage: '/images/flow\_box.png',  
 // category: 'spray'  
 // },  
 // {  
 // \_id: '4',  
 // name: 'Bliss 100ml',  
 // nameAr: 'بليس 100مل',  
 // price: 230.00,  
 // currency: '﷼',  
 // image: '/images/bliss\_bottle.png',  
 // boxImage: '/images/bliss\_box.png',  
 // category: 'spray'  
 // }  
 // ];  
   
 // setFeaturedProducts(mockProducts);  
 // setNewArrivals(mockProducts);  
 // setLoading(false);  
 // }, 1000);  
  
 fetchProducts(); // Uncomment when backend is connected  
 }, []);  
  
 if (loading) {  
 return <LoadingContainer>Loading...</LoadingContainer>;  
 }  
  
 return (  
 <>  
 <HeroSection>  
 <div className="container">  
 <HeroContent>  
 <motion.div  
 initial={{ opacity: 0, y: 50 }}  
 animate={{ opacity: 1, y: 0 }}  
 transition={{ duration: 0.8 }}  
 >  
 <HeroTitle>Discover The Perfect Fragrance</HeroTitle>  
 <HeroSubtitle>Luxury scents that captivate your senses</HeroSubtitle>  
 <HeroButton to="/products">  
 Shop Now  
 {isRTL ? <FaArrowLeft /> : <FaArrowRight />}  
 </HeroButton>  
 </motion.div>  
 </HeroContent>  
 </div>  
 </HeroSection>  
  
 <Section>  
 <div className="container">  
 <SectionHeader>  
 <SectionTitle>Featured Products</SectionTitle>  
 <SectionLink to="/products">  
 View All  
 {isRTL ? <FaArrowLeft /> : <FaArrowRight />}  
 </SectionLink>  
 </SectionHeader>  
   
 <ProductGrid>  
 {Array.isArray(featuredProducts) && featuredProducts.length > 0 ? (  
 featuredProducts.map(product => (  
 <ProductItem key={product.\_id}>  
 <ProductCard product={product} isRTL={isRTL} />  
 </ProductItem>  
 ))  
 ) : (  
 <NoProductsMessage>No featured products available</NoProductsMessage>  
 )}  
 </ProductGrid>  
 </div>  
 </Section>  
  
 <StorySection>  
 <div className="container">  
 <StoryContainer>  
 <StoryImage src="/images/story-image.jpg" alt="Our Story" />  
 <StoryContent>  
 <motion.div  
 initial={{ opacity: 0, x: 50 }}  
 animate={{ opacity: 1, x: 0 }}  
 transition={{ duration: 0.8 }}  
 >  
 <SectionTitle>Our Passion</SectionTitle>  
 <StoryText>  
 A passionate story sirm started in 2018, looking for excellence and exclusivity,   
 taking everything in the field and taking it far to attract the largest category   
 that loves perfumes in the Middle East.  
 </StoryText>  
 <StoryButton to="/about">Learn More</StoryButton>  
 </motion.div>  
 </StoryContent>  
 </StoryContainer>  
 </div>  
 </StorySection>  
  
 <Section>  
 <div className="container">  
 <SectionHeader>  
 <SectionTitle>New Arrivals</SectionTitle>  
 <SectionLink to="/products">  
 View All  
 {isRTL ? <FaArrowLeft /> : <FaArrowRight />}  
 </SectionLink>  
 </SectionHeader>  
   
 <ProductGrid>  
 {Array.isArray(newArrivals) && newArrivals.length > 0 ? (  
 newArrivals.map(product => (  
 <ProductItem key={product.\_id}>  
 <ProductCard product={product} isRTL={isRTL} />  
 </ProductItem>  
 ))  
 ) : (  
 <NoProductsMessage>No new arrivals available</NoProductsMessage>  
 )}  
 </ProductGrid>  
 </div>  
 </Section>  
  
 <SubscribeSection>  
 <div className="container">  
 <SubscribeContainer>  
 <SubscribeTitle>Subscribe to Our Newsletter</SubscribeTitle>  
 <SubscribeText>Stay updated with our latest products and exclusive offers</SubscribeText>  
 <SubscribeForm>  
 <SubscribeInput type="email" placeholder="Your email address" />  
 <SubscribeButton type="submit">Subscribe</SubscribeButton>  
 </SubscribeForm>  
 </SubscribeContainer>  
 </div>  
 </SubscribeSection>  
 </>  
 );  
};  
  
// Styled Components  
const LoadingContainer = styled.div`  
 display: flex;  
 justify-content: center;  
 align-items: center;  
 height: 400px;  
 font-size: 18px;  
 color: var(--primary-color);  
`;  
  
const NoProductsMessage = styled.div`  
 grid-column: 1 / -1;  
 text-align: center;  
 padding: 40px;  
 font-size: 18px;  
 color: var(--primary-color);  
`;  
  
const HeroSection = styled.section`  
 background-image: url('/images/hero-background.jpg');  
 background-size: cover;  
 background-position: center;  
 height: 80vh;  
 min-height: 500px;  
 display: flex;  
 align-items: center;  
 position: relative;  
 color: var(--text-color);  
   
 &:before {  
 content: '';  
 position: absolute;  
 top: 0;  
 left: 0;  
 right: 0;  
 bottom: 0;  
 background: rgba(0, 0, 0, 0.5);  
 }  
`;  
  
const HeroContent = styled.div`  
 position: relative;  
 z-index: 1;  
 max-width: 600px;  
   
 .rtl & {  
 margin-left: auto;  
 text-align: right;  
 }  
`;  
  
const HeroTitle = styled.h1`  
 font-size: 48px;  
 margin-bottom: 20px;  
   
 @media (max-width: 768px) {  
 font-size: 36px;  
 }  
`;  
  
const HeroSubtitle = styled.p`  
 font-size: 20px;  
 margin-bottom: 30px;  
 opacity: 0.9;  
   
 @media (max-width: 768px) {  
 font-size: 18px;  
 }  
`;  
  
const HeroButton = styled(Link)`  
 display: inline-flex;  
 align-items: center;  
 padding: 12px 25px;  
 background-color: var(--secondary-color);  
 color: var(--primary-color);  
 text-decoration: none;  
 font-weight: 600;  
 font-size: 16px;  
 border-radius: 4px;  
 transition: all 0.3s ease;  
   
 svg {  
 margin-left: 10px;  
   
 .rtl & {  
 margin-left: 0;  
 margin-right: 10px;  
 }  
 }  
   
 &:hover {  
 background-color: var(--accent-color);  
 }  
`;  
  
const Section = styled.section`  
 padding: 80px 0;  
   
 @media (max-width: 768px) {  
 padding: 50px 0;  
 }  
`;  
  
const SectionHeader = styled.div`  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 margin-bottom: 40px;  
   
 @media (max-width: 768px) {  
 flex-direction: column;  
 align-items: flex-start;  
 gap: 15px;  
 }  
`;  
  
const SectionTitle = styled.h2`  
 font-size: 32px;  
 color: var(--primary-color);  
 position: relative;  
   
 &:after {  
 content: '';  
 display: block;  
 width: 60px;  
 height: 2px;  
 background-color: var(--secondary-color);  
 margin-top: 15px;  
   
 .rtl & {  
 margin-left: auto;  
 }  
 }  
   
 @media (max-width: 768px) {  
 font-size: 24px;  
 }  
`;  
  
const SectionLink = styled(Link)`  
 display: flex;  
 align-items: center;  
 color: var(--secondary-color);  
 text-decoration: none;  
 font-weight: 600;  
 transition: all 0.3s ease;  
   
 svg {  
 margin-left: 8px;  
   
 .rtl & {  
 margin-left: 0;  
 margin-right: 8px;  
 }  
 }  
   
 &:hover {  
 color: var(--accent-color);  
 }  
`;  
  
const ProductGrid = styled.div`  
 display: grid;  
 grid-template-columns: repeat(4, 1fr);  
 gap: 30px;  
   
 @media (max-width: 1200px) {  
 grid-template-columns: repeat(3, 1fr);  
 }  
   
 @media (max-width: 768px) {  
 grid-template-columns: repeat(2, 1fr);  
 gap: 20px;  
 }  
   
 @media (max-width: 480px) {  
 grid-template-columns: 1fr;  
 }  
`;  
  
const ProductItem = styled.div``;  
  
const StorySection = styled.section`  
 padding: 80px 0;  
 background-color: #f9f9f9;  
   
 @media (max-width: 768px) {  
 padding: 50px 0;  
 }  
`;  
  
const StoryContainer = styled.div`  
 display: flex;  
 align-items: center;  
   
 @media (max-width: 768px) {  
 flex-direction: column;  
 gap: 30px;  
 }  
`;  
  
const StoryImage = styled.img`  
 width: 50%;  
 height: 400px;  
 object-fit: cover;  
 border-radius: 8px;  
   
 @media (max-width: 768px) {  
 width: 100%;  
 height: 300px;  
 }  
`;  
  
const StoryContent = styled.div`  
 width: 50%;  
 padding-left: 50px;  
   
 .rtl & {  
 padding-left: 0;  
 padding-right: 50px;  
 }  
   
 @media (max-width: 768px) {  
 width: 100%;  
 padding-left: 0;  
 padding-right: 0;  
   
 .rtl & {  
 padding-right: 0;  
 }  
 }  
`;  
  
const StoryText = styled.p`  
 font-size: 16px;  
 line-height: 1.6;  
 margin-bottom: 30px;  
 color: #555;  
`;  
  
const StoryButton = styled(Link)`  
 display: inline-block;  
 padding: 10px 20px;  
 background: none;  
 border: 2px solid var(--secondary-color);  
 color: var(--secondary-color);  
 text-decoration: none;  
 font-weight: 600;  
 border-radius: 4px;  
 transition: all 0.3s ease;  
   
 &:hover {  
 background-color: var(--secondary-color);  
 color: white;  
 }  
`;  
  
const SubscribeSection = styled.section`  
 padding: 80px 0;  
 background-color: var(--primary-color);  
 color: var(--text-color);  
   
 @media (max-width: 768px) {  
 padding: 50px 0;  
 }  
`;  
  
const SubscribeContainer = styled.div`  
 max-width: 600px;  
 margin: 0 auto;  
 text-align: center;  
`;  
  
const SubscribeTitle = styled.h2`  
 font-size: 32px;  
 margin-bottom: 20px;  
   
 @media (max-width: 768px) {  
 font-size: 24px;  
 }  
`;  
  
const SubscribeText = styled.p`  
 font-size: 16px;  
 margin-bottom: 30px;  
 opacity: 0.8;  
`;  
  
const SubscribeForm = styled.form`  
 display: flex;  
   
 @media (max-width: 480px) {  
 flex-direction: column;  
 gap: 15px;  
 }  
`;  
  
const SubscribeInput = styled.input`  
 flex: 1;  
 padding: 12px 15px;  
 border: none;  
 border-radius: 4px 0 0 4px;  
 font-size: 16px;  
   
 &:focus {  
 outline: none;  
 }  
   
 @media (max-width: 480px) {  
 border-radius: 4px;  
 }  
`;  
  
const SubscribeButton = styled.button`  
 padding: 12px 25px;  
 background-color: var(--secondary-color);  
 color: var(--primary-color);  
 border: none;  
 border-radius: 0 4px 4px 0;  
 font-weight: 600;  
 cursor: pointer;  
 transition: all 0.3s ease;  
   
 &:hover {  
 background-color: var(--accent-color);  
 }  
   
 @media (max-width: 480px) {  
 border-radius: 4px;  
 }  
`;  
  
export default Home;

File Path: .\src\pages\Login.jsx

Content:

import { useState } from 'react';  
import { Link, useNavigate, useLocation } from 'react-router-dom';  
import { useTranslation } from 'react-i18next';  
import styled from 'styled-components';  
import { FaEnvelope, FaLock, FaSignInAlt } from 'react-icons/fa';  
import { useAuth } from '../context/AuthContext';  
import { useApp } from '../context/AppContext';  
  
const Login = () => {  
 const { t } = useTranslation();  
 const navigate = useNavigate();  
 const location = useLocation();  
 const { login } = useAuth();  
 const { addNotification } = useApp();  
   
 const [formData, setFormData] = useState({  
 email: '',  
 password: ''  
 });  
   
 const [errors, setErrors] = useState({});  
 const [isSubmitting, setIsSubmitting] = useState(false);  
   
 const redirectTo = new URLSearchParams(location.search).get('redirect') || '/';  
  
 const validateForm = () => {  
 const newErrors = {};  
   
 if (!formData.email.trim()) {  
 newErrors.email = t('email\_required');  
 } else if (!/\S+@\S+\.\S+/.test(formData.email)) {  
 newErrors.email = t('valid\_email\_required');  
 }  
   
 if (!formData.password) {  
 newErrors.password = t('password\_required');  
 }  
   
 setErrors(newErrors);  
 return Object.keys(newErrors).length === 0;  
 };  
  
 const handleChange = (e) => {  
 const { name, value } = e.target;  
 setFormData(prev => ({  
 ...prev,  
 [name]: value  
 }));  
   
 // Clear error when user types  
 if (errors[name]) {  
 setErrors(prev => ({  
 ...prev,  
 [name]: ''  
 }));  
 }  
 };  
  
 const handleSubmit = async (e) => {  
 e.preventDefault();  
   
 if (!validateForm()) {  
 return;  
 }  
   
 setIsSubmitting(true);  
   
 // try {  
 // // await login(formData.email, formData.password);  
 // const userData = await login(formData.email, formData.password);  
 // console.log("Login successful, user data:", userData);  
 // console.log("Is admin?", userData.role === 'admin');  
 // addNotification(t('login\_success'), 'success');  
 // navigate(redirectTo);  
 // } catch (err) {  
 // const errorMessage = err.response?.data?.message || t('login\_failed');  
 // setErrors(prev => ({  
 // ...prev,  
 // general: errorMessage  
 // }));  
  
 try {  
 const userData = await login(formData.email, formData.password);  
 addNotification(t('login\_success'), 'success');  
 console.log("Login successful, user data:", userData);  
 console.log("Is admin?", userData.role === 'admin');  
 // Redirect based on role  
 if (userData.role === 'admin') {  
 navigate('/admin'); // Redirect to admin dashboard  
 } else {  
 navigate(redirectTo); // Regular user redirect  
 }  
 } catch (err) {  
 const errorMessage = err.response?.data?.message || t('login\_failed');  
 setErrors(prev => ({  
 ...prev,  
 general: errorMessage  
 }));  
 } finally {  
 setIsSubmitting(false);  
 }  
 };  
  
 return (  
 <div className="container">  
 <LoginContainer>  
 <LoginFormContainer>  
 <LoginHeader>  
 <h1>{t('login')}</h1>  
 <p>{t('login\_subtitle')}</p>  
 </LoginHeader>  
   
 {errors.general && (  
 <ErrorMessage>{errors.general}</ErrorMessage>  
 )}  
   
 <LoginForm onSubmit={handleSubmit}>  
 <FormGroup>  
 <FormLabel>  
 <FaEnvelope />  
 <span>{t('email\_address')}</span>  
 </FormLabel>  
 <FormInput  
 type="email"  
 name="email"  
 placeholder={t('email\_placeholder')}  
 value={formData.email}  
 onChange={handleChange}  
 hasError={!!errors.email}  
 />  
 {errors.email && (  
 <ErrorMessage>{errors.email}</ErrorMessage>  
 )}  
 </FormGroup>  
   
 <FormGroup>  
 <FormLabel>  
 <FaLock />  
 <span>{t('password')}</span>  
 </FormLabel>  
 <FormInput  
 type="password"  
 name="password"  
 placeholder={t('password\_placeholder')}  
 value={formData.password}  
 onChange={handleChange}  
 hasError={!!errors.password}  
 />  
 {errors.password && (  
 <ErrorMessage>{errors.password}</ErrorMessage>  
 )}  
 </FormGroup>  
   
 <RememberForgot>  
 <RememberMe>  
 <input type="checkbox" id="remember" />  
 <label htmlFor="remember">{t('remember\_me')}</label>  
 </RememberMe>  
   
 <ForgotPassword to="/forgot-password">  
 {t('forgot\_password')}  
 </ForgotPassword>  
 </RememberForgot>  
   
 <SubmitButton type="submit" disabled={isSubmitting}>  
 {isSubmitting ? t('logging\_in') : t('login')}  
 {!isSubmitting && <FaSignInAlt />}  
 </SubmitButton>  
 </LoginForm>  
   
 <SignupPrompt>  
 {t('no\_account')}{' '}  
 <SignupLink to={`/register${location.search}`}>  
 {t('create\_account')}  
 </SignupLink>  
 </SignupPrompt>  
 </LoginFormContainer>  
   
 <LoginImageContainer>  
 <img src="/images/login-image.jpg" alt="Login" />  
 </LoginImageContainer>  
 </LoginContainer>  
 </div>  
 );  
};  
  
// Styled Components  
const LoginContainer = styled.div`  
 display: grid;  
 grid-template-columns: 1fr 1fr;  
 min-height: 70vh;  
 margin: 50px 0;  
 box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);  
 border-radius: 8px;  
 overflow: hidden;  
   
 @media (max-width: 768px) {  
 grid-template-columns: 1fr;  
 }  
`;  
  
const LoginFormContainer = styled.div`  
 padding: 40px;  
 display: flex;  
 flex-direction: column;  
 justify-content: center;  
   
 @media (max-width: 480px) {  
 padding: 30px 20px;  
 }  
`;  
  
const LoginHeader = styled.div`  
 margin-bottom: 30px;  
   
 h1 {  
 font-size: 28px;  
 color: var(--primary-color);  
 margin-bottom: 10px;  
 }  
   
 p {  
 color: #666;  
 }  
`;  
  
const LoginForm = styled.form`  
 display: flex;  
 flex-direction: column;  
 gap: 20px;  
`;  
  
const FormGroup = styled.div`  
 display: flex;  
 flex-direction: column;  
`;  
  
const FormLabel = styled.label`  
 display: flex;  
 align-items: center;  
 gap: 8px;  
 margin-bottom: 8px;  
 color: var(--primary-color);  
 font-weight: 500;  
   
 svg {  
 color: var(--secondary-color);  
 }  
`;  
  
const FormInput = styled.input`  
 padding: 12px 15px;  
 border: 1px solid ${props => props.hasError ? 'var(--danger-color)' : '#ddd'};  
 border-radius: 4px;  
 font-size: 16px;  
   
 &:focus {  
 outline: none;  
 border-color: ${props => props.hasError ? 'var(--danger-color)' : 'var(--secondary-color)'};  
 }  
`;  
  
const ErrorMessage = styled.div`  
 color: var(--danger-color);  
 font-size: 14px;  
 margin-top: 5px;  
`;  
  
const RememberForgot = styled.div`  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
`;  
  
const RememberMe = styled.div`  
 display: flex;  
 align-items: center;  
 gap: 8px;  
 color: #666;  
   
 input {  
 accent-color: var(--secondary-color);  
 }  
`;  
  
const ForgotPassword = styled(Link)`  
 color: var(--secondary-color);  
 text-decoration: none;  
 font-size: 14px;  
   
 &:hover {  
 text-decoration: underline;  
 }  
`;  
  
const SubmitButton = styled.button`  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 gap: 10px;  
 padding: 12px 15px;  
 background-color: var(--secondary-color);  
 color: var(--primary-color);  
 border: none;  
 border-radius: 4px;  
 font-size: 16px;  
 font-weight: 600;  
 cursor: pointer;  
 transition: all 0.3s ease;  
 margin-top: 10px;  
   
 &:hover {  
 background-color: var(--accent-color);  
 }  
   
 &:disabled {  
 opacity: 0.7;  
 cursor: not-allowed;  
 }  
`;  
  
const SignupPrompt = styled.div`  
 margin-top: 30px;  
 text-align: center;  
 color: #666;  
`;  
  
const SignupLink = styled(Link)`  
 color: var(--secondary-color);  
 text-decoration: none;  
 font-weight: 600;  
   
 &:hover {  
 text-decoration: underline;  
 }  
`;  
  
const LoginImageContainer = styled.div`  
 overflow: hidden;  
   
 img {  
 width: 100%;  
 height: 100%;  
 object-fit: cover;  
 }  
   
 @media (max-width: 768px) {  
 display: none;  
 }  
`;  
  
export default Login;

File Path: .\src\pages\ProductDetail.jsx

Content:

import { useState, useEffect } from 'react';  
import { useParams, useNavigate } from 'react-router-dom';  
import { useTranslation } from 'react-i18next';  
import styled from 'styled-components';  
import { FaHeart, FaRegHeart, FaShoppingCart, FaArrowLeft } from 'react-icons/fa';  
import axios from 'axios';  
import { useCart } from '../context/CartContext';  
import { useApp } from '../context/AppContext';  
  
const ProductDetail = () => {  
 const { id } = useParams();  
 const navigate = useNavigate();  
 const { t, i18n } = useTranslation();  
 const { addToCart } = useCart();  
 const { addNotification } = useApp();  
 const [product, setProduct] = useState(null);  
 const [loading, setLoading] = useState(true);  
 const [quantity, setQuantity] = useState(1);  
 const [isFavorite, setIsFavorite] = useState(false);  
 const [relatedProducts, setRelatedProducts] = useState([]);  
 const isRTL = i18n.language === 'ar';  
  
 useEffect(() => {  
 const fetchProductData = async () => {  
 try {  
 // Fetch product details  
 const productResponse = await axios.get(`/api/products/${id}`);  
 setProduct(productResponse.data);  
   
 // Fetch related products  
 const relatedResponse = await axios.get(`/api/products?category=${productResponse.data.category}&limit=4`);  
 setRelatedProducts(relatedResponse.data.filter(item => item.\_id !== id));  
   
 setLoading(false);  
 } catch (err) {  
 console.error('Error fetching product:', err);  
 setLoading(false);  
 }  
 };  
  
 // Mock data for development  
 // setTimeout(() => {  
 // const mockProduct = {  
 // \_id: id,  
 // name: 'Musk 100ml',  
 // nameAr: 'مسك 100مل',  
 // price: 230.00,  
 // currency: '﷼',  
 // image: '/images/musk\_bottle.png',  
 // boxImage: '/images/musk\_box.png',  
 // category: 'spray',  
 // description: 'A refreshing fragrance that opens with citrus grapefruit, ginger and Italian bergamot. The heart of the perfume consists of the scent of cold water, rosemary and the luxurious geranium flower. The base of the perfume is characterized by the strength of ambroxan, a touch of amber and a little labdanum.',  
 // descriptionAr: 'عطر يتميز بأنه منعش ابتداءاً بالجريب فروت الحمضي و الزنجبيل و البرغموت الإيطالي و يتكون قلب العطر من رائحة الماء البارده و إكليل الجبل و زهرة إبره الراعي الفاخرة قاعدة العطر تتميز بقوة الأمبروكسان ولمسة العنبر وقليل من اللابدانوم',  
 // stock: 45,  
 // features: [  
 // { name: 'Volume', value: '100ml' },  
 // { name: 'Top Notes', value: 'Grapefruit, Ginger, Bergamot' },  
 // { name: 'Heart Notes', value: 'Cold Water, Rosemary, Geranium' },  
 // { name: 'Base Notes', value: 'Ambroxan, Amber, Labdanum' }  
 // ]  
 // };  
   
 // const mockRelated = [  
 // {  
 // \_id: '2',  
 // name: 'The fifth 100ml',  
 // nameAr: 'ذا فيفث 100مل',  
 // price: 230.00,  
 // currency: '﷼',  
 // image: '/images/fifth\_bottle.png',  
 // boxImage: '/images/fifth\_box.png',  
 // category: 'spray'  
 // },  
 // {  
 // \_id: '3',  
 // name: 'Flow of three 100ml',  
 // nameAr: 'فلو أوف ثري 100مل',  
 // price: 230.00,  
 // currency: '﷼',  
 // image: '/images/flow\_bottle.png',  
 // boxImage: '/images/flow\_box.png',  
 // category: 'spray'  
 // },  
 // {  
 // \_id: '4',  
 // name: 'Bliss 100ml',  
 // nameAr: 'بليس 100مل',  
 // price: 230.00,  
 // currency: '﷼',  
 // image: '/images/bliss\_bottle.png',  
 // boxImage: '/images/bliss\_box.png',  
 // category: 'spray'  
 // }  
 // ];  
   
 // setProduct(mockProduct);  
 // setRelatedProducts(mockRelated);  
 // setLoading(false);  
 // }, 1000);  
  
 fetchProductData(); // Uncomment when backend is connected  
 }, [id]);  
  
 const handleQuantityChange = (e) => {  
 const value = parseInt(e.target.value);  
 if (value > 0 && value <= product.stock) {  
 setQuantity(value);  
 }  
 };  
  
 const decrementQuantity = () => {  
 if (quantity > 1) {  
 setQuantity(quantity - 1);  
 }  
 };  
  
 const incrementQuantity = () => {  
 if (quantity < product.stock) {  
 setQuantity(quantity + 1);  
 }  
 };  
  
 const handleAddToCart = () => {  
 addToCart(product, quantity);  
 addNotification(`${product.name} added to cart`, 'success');  
 };  
  
 const toggleFavorite = () => {  
 setIsFavorite(!isFavorite);  
 // Add to favorites API call would go here  
 };  
  
 if (loading) {  
 return <LoadingContainer>Loading...</LoadingContainer>;  
 }  
  
 if (!product) {  
 return <ErrorContainer>Product not found</ErrorContainer>;  
 }  
  
 return (  
 <div className="container">  
 <NavigationBack onClick={() => navigate(-1)}>  
 <FaArrowLeft />  
 <span>{t('back')}</span>  
 </NavigationBack>  
   
 <ProductContainer>  
 <ProductGallery>  
 <MainImage>  
 <img src={product.image} alt={product.name} />  
 <BoxImage src={product.boxImage} alt={`${product.name} box`} />  
 </MainImage>  
 </ProductGallery>  
   
 <ProductInfo>  
 <ProductTitle>  
 {isRTL ? product.nameAr : product.name}  
 </ProductTitle>  
   
 <ProductPrice>  
 {product.currency} {product.price.toFixed(2)}  
 </ProductPrice>  
   
 <ProductDescription>  
 {isRTL ? product.descriptionAr : product.description}  
 </ProductDescription>  
   
 {product.features && (  
 <ProductFeatures>  
 {product.features.map((feature, index) => (  
 <FeatureItem key={index}>  
 <FeatureLabel>{feature.name}:</FeatureLabel>  
 <FeatureValue>{feature.value}</FeatureValue>  
 </FeatureItem>  
 ))}  
 </ProductFeatures>  
 )}  
   
 <StockInfo>  
 {product.stock > 0 ? (  
 <InStock>{t('in\_stock')}</InStock>  
 ) : (  
 <OutOfStock>{t('out\_of\_stock')}</OutOfStock>  
 )}  
 </StockInfo>  
   
 {product.stock > 0 && (  
 <>  
 <QuantityContainer>  
 <QuantityLabel>{t('quantity')}:</QuantityLabel>  
 <QuantityControls>  
 <QuantityButton onClick={decrementQuantity}>-</QuantityButton>  
 <QuantityInput   
 type="number"   
 value={quantity}   
 onChange={handleQuantityChange}   
 min="1"   
 max={product.stock}  
 />  
 <QuantityButton onClick={incrementQuantity}>+</QuantityButton>  
 </QuantityControls>  
 </QuantityContainer>  
   
 <ActionButtons>  
 <AddToCartButton onClick={handleAddToCart}>  
 <FaShoppingCart />  
 <span>{t('add\_to\_cart')}</span>  
 </AddToCartButton>  
   
 <FavoriteButton onClick={toggleFavorite}>  
 {isFavorite ? <FaHeart /> : <FaRegHeart />}  
 </FavoriteButton>  
 </ActionButtons>  
 </>  
 )}  
 </ProductInfo>  
 </ProductContainer>  
   
 {relatedProducts.length > 0 && (  
 <RelatedSection>  
 <SectionTitle>{t('related\_products')}</SectionTitle>  
 <RelatedGrid>  
 {relatedProducts.map(related => (  
 <RelatedItem key={related.\_id} onClick={() => navigate(`/product/${related.\_id}`)}>  
 <RelatedImage>  
 <img src={related.image} alt={related.name} />  
 </RelatedImage>  
 <RelatedName>  
 {isRTL ? related.nameAr : related.name}  
 </RelatedName>  
 <RelatedPrice>  
 {related.currency} {related.price.toFixed(2)}  
 </RelatedPrice>  
 </RelatedItem>  
 ))}  
 </RelatedGrid>  
 </RelatedSection>  
 )}  
 </div>  
 );  
 };  
   
 // Styled Components  
 const LoadingContainer = styled.div`  
 display: flex;  
 justify-content: center;  
 align-items: center;  
 height: 400px;  
 font-size: 18px;  
 color: var(--primary-color);  
 `;  
   
 const ErrorContainer = styled.div`  
 display: flex;  
 justify-content: center;  
 align-items: center;  
 height: 400px;  
 font-size: 18px;  
 color: var(--danger-color);  
 `;  
   
 const NavigationBack = styled.button`  
 display: flex;  
 align-items: center;  
 background: none;  
 border: none;  
 color: var(--primary-color);  
 font-size: 16px;  
 cursor: pointer;  
 margin: 20px 0;  
   
 span {  
 margin-left: 10px;  
   
 .rtl & {  
 margin-left: 0;  
 margin-right: 10px;  
 }  
 }  
   
 &:hover {  
 color: var(--secondary-color);  
 }  
 `;  
   
 const ProductContainer = styled.div`  
 display: grid;  
 grid-template-columns: 1fr 1fr;  
 gap: 50px;  
 margin-bottom: 80px;  
   
 @media (max-width: 768px) {  
 grid-template-columns: 1fr;  
 gap: 30px;  
 }  
 `;  
   
 const ProductGallery = styled.div``;  
   
 const MainImage = styled.div`  
 position: relative;  
 background-color: #f9f9f9;  
 border-radius: 8px;  
 height: 500px;  
 display: flex;  
 justify-content: center;  
 align-items: center;  
   
 img {  
 max-height: 80%;  
 max-width: 50%;  
 object-fit: contain;  
 }  
   
 @media (max-width: 768px) {  
 height: 400px;  
 }  
 `;  
   
 const BoxImage = styled.img`  
 position: absolute;  
 top: 50%;  
 right: 25%;  
 transform: translateY(-50%);  
 max-height: 80%;  
 max-width: 40%;  
 object-fit: contain;  
 `;  
   
 const ProductInfo = styled.div``;  
   
 const ProductTitle = styled.h1`  
 font-size: 32px;  
 color: var(--primary-color);  
 margin-bottom: 15px;  
   
 @media (max-width: 768px) {  
 font-size: 24px;  
 }  
 `;  
   
 const ProductPrice = styled.div`  
 font-size: 24px;  
 font-weight: 600;  
 color: var(--secondary-color);  
 margin-bottom: 20px;  
 `;  
   
 const ProductDescription = styled.div`  
 margin-bottom: 30px;  
 line-height: 1.6;  
 color: #555;  
 `;  
   
 const ProductFeatures = styled.div`  
 margin-bottom: 30px;  
 `;  
   
 const FeatureItem = styled.div`  
 display: flex;  
 margin-bottom: 10px;  
   
 .rtl & {  
 flex-direction: row-reverse;  
 }  
 `;  
   
 const FeatureLabel = styled.span`  
 font-weight: 600;  
 width: 120px;  
 color: var(--primary-color);  
 `;  
   
 const FeatureValue = styled.span`  
 color: #555;  
 `;  
   
 const StockInfo = styled.div`  
 margin-bottom: 20px;  
 `;  
   
 const InStock = styled.span`  
 color: var(--success-color);  
 font-weight: 600;  
 `;  
   
 const OutOfStock = styled.span`  
 color: var(--danger-color);  
 font-weight: 600;  
 `;  
   
 const QuantityContainer = styled.div`  
 display: flex;  
 align-items: center;  
 margin-bottom: 30px;  
   
 .rtl & {  
 flex-direction: row-reverse;  
 }  
 `;  
   
 const QuantityLabel = styled.span`  
 margin-right: 15px;  
   
 .rtl & {  
 margin-right: 0;  
 margin-left: 15px;  
 }  
 `;  
   
 const QuantityControls = styled.div`  
 display: flex;  
 align-items: center;  
 `;  
   
 const QuantityButton = styled.button`  
 width: 36px;  
 height: 36px;  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 background-color: #f1f1f1;  
 border: none;  
 font-size: 18px;  
 cursor: pointer;  
   
 &:first-child {  
 border-radius: 4px 0 0 4px;  
   
 .rtl & {  
 border-radius: 0 4px 4px 0;  
 }  
 }  
   
 &:last-child {  
 border-radius: 0 4px 4px 0;  
   
 .rtl & {  
 border-radius: 4px 0 0 4px;  
 }  
 }  
   
 &:hover {  
 background-color: #e1e1e1;  
 }  
 `;  
   
 const QuantityInput = styled.input`  
 width: 50px;  
 height: 36px;  
 border: 1px solid #f1f1f1;  
 text-align: center;  
 font-size: 16px;  
 -moz-appearance: textfield;  
   
 &::-webkit-outer-spin-button,  
 &::-webkit-inner-spin-button {  
 -webkit-appearance: none;  
 margin: 0;  
 }  
   
 &:focus {  
 outline: none;  
 border-color: var(--secondary-color);  
 }  
 `;  
   
 const ActionButtons = styled.div`  
 display: flex;  
 `;  
   
 const AddToCartButton = styled.button`  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 background-color: var(--secondary-color);  
 color: var(--primary-color);  
 border: none;  
 padding: 12px 25px;  
 font-size: 16px;  
 font-weight: 600;  
 border-radius: 4px;  
 cursor: pointer;  
 transition: all 0.3s ease;  
 flex: 1;  
   
 svg {  
 margin-right: 10px;  
   
 .rtl & {  
 margin-right: 0;  
 margin-left: 10px;  
 }  
 }  
   
 &:hover {  
 background-color: var(--accent-color);  
 }  
 `;  
   
 const FavoriteButton = styled.button`  
 width: 50px;  
 height: 50px;  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 background: none;  
 border: 1px solid #ddd;  
 color: ${props => props.isFavorite ? 'var(--danger-color)' : 'var(--primary-color)'};  
 border-radius: 4px;  
 margin-left: 15px;  
 cursor: pointer;  
 transition: all 0.3s ease;  
   
 .rtl & {  
 margin-left: 0;  
 margin-right: 15px;  
 }  
   
 &:hover {  
 border-color: var(--secondary-color);  
 color: var(--secondary-color);  
 }  
 `;  
   
 const RelatedSection = styled.div`  
 margin-bottom: 80px;  
 `;  
   
 const SectionTitle = styled.h2`  
 font-size: 24px;  
 color: var(--primary-color);  
 margin-bottom: 30px;  
 position: relative;  
   
 &:after {  
 content: '';  
 display: block;  
 width: 50px;  
 height: 2px;  
 background-color: var(--secondary-color);  
 margin-top: 10px;  
   
 .rtl & {  
 margin-left: auto;  
 }  
 }  
 `;  
   
 const RelatedGrid = styled.div`  
 display: grid;  
 grid-template-columns: repeat(4, 1fr);  
 gap: 20px;  
   
 @media (max-width: 1024px) {  
 grid-template-columns: repeat(3, 1fr);  
 }  
   
 @media (max-width: 768px) {  
 grid-template-columns: repeat(2, 1fr);  
 }  
   
 @media (max-width: 480px) {  
 grid-template-columns: 1fr;  
 }  
 `;  
   
 const RelatedItem = styled.div`  
 background-color: white;  
 border-radius: 8px;  
 overflow: hidden;  
 box-shadow: var(--card-shadow);  
 cursor: pointer;  
 transition: all 0.3s ease;  
   
 &:hover {  
 transform: translateY(-5px);  
 box-shadow: 0 10px 20px rgba(0, 0, 0, 0.1);  
 }  
 `;  
   
 const RelatedImage = styled.div`  
 height: 200px;  
 background-color: #f9f9f9;  
 display: flex;  
 align-items: center;  
 justify-content: center;  
   
 img {  
 max-height: 80%;  
 max-width: 80%;  
 object-fit: contain;  
 }  
 `;  
   
 const RelatedName = styled.div`  
 padding: 15px 15px 5px;  
 font-size: 16px;  
 font-weight: 500;  
 color: var(--primary-color);  
 `;  
   
 const RelatedPrice = styled.div`  
 padding: 0 15px 15px;  
 font-size: 18px;  
 font-weight: 600;  
 color: var(--secondary-color);  
 `;  
   
 export default ProductDetail;

File Path: .\src\pages\ProductsPage.jsx

Content:

import { useState, useEffect } from 'react';  
import { useTranslation } from 'react-i18next';  
import { useSearchParams } from 'react-router-dom';  
import styled from 'styled-components';  
import axios from 'axios';  
import ProductCard from '../components/ProductCard';  
import { productService } from '../services';  
import { useApp } from '../context/AppContext';  
  
const ProductsPage = () => {  
 const { t, i18n } = useTranslation();  
 const [searchParams] = useSearchParams();  
 const [products, setProducts] = useState([]);  
 const [loading, setLoading] = useState(true);  
 const [error, setError] = useState(null);  
 const [currentCategory, setCurrentCategory] = useState('all');  
 const { showLoading, hideLoading, addNotification } = useApp();  
   
 const searchQuery = searchParams.get('search') || '';  
 const isRTL = i18n.language === 'ar';  
  
 useEffect(() => {  
 const fetchProducts = async () => {  
 try {  
 showLoading();  
 setLoading(true);  
   
 let params = {};  
 if (currentCategory !== 'all') {  
 params.category = currentCategory;  
 }  
 if (searchQuery) {  
 params.search = searchQuery;  
 }  
   
 const data = await productService.getAllProducts(params);  
 console.log('Fetched products:', data);  
 setProducts(Array.isArray(data) ? data : []);  
 setLoading(false);  
 hideLoading();  
 } catch (err) {  
 console.error('Error fetching products:', err);  
 setError(err.message);  
 setLoading(false);  
 hideLoading();  
 addNotification(t('error\_fetching\_products') || 'Error fetching products', 'error');  
 }  
 };  
  
 fetchProducts();  
 }, [currentCategory, searchQuery, showLoading, hideLoading, addNotification, t]);  
  
  
 // // Mock data until backend is connected  
 // useEffect(() => {  
 // const mockProducts = [  
 // {  
 // \_id: '1',  
 // name: 'Musk 100ml',  
 // price: 230.00,  
 // currency: '﷼',  
 // image: '/images/musk\_bottle.png',  
 // boxImage: '/images/musk\_box.png',  
 // category: 'spray'  
 // },  
 // {  
 // \_id: '2',  
 // name: 'The fifth 100ml',  
 // price: 230.00,  
 // currency: '﷼',  
 // image: '/images/fifth\_bottle.png',  
 // boxImage: '/images/fifth\_box.png',  
 // category: 'spray'  
 // },  
 // {  
 // \_id: '3',  
 // name: 'Flow of three 100ml',  
 // price: 230.00,  
 // currency: '﷼',  
 // image: '/images/flow\_bottle.png',  
 // boxImage: '/images/flow\_box.png',  
 // category: 'spray'  
 // },  
 // {  
 // \_id: '4',  
 // name: 'Bliss 100ml',  
 // price: 230.00,  
 // currency: '﷼',  
 // image: '/images/bliss\_bottle.png',  
 // boxImage: '/images/bliss\_box.png',  
 // category: 'spray'  
 // }  
 // ];  
   
 // setProducts(mockProducts);  
 // setLoading(false);  
 // }, []);  
  
 const categories = [  
 { id: 'all', name: { en: 'All Products', ar: 'جميع المنتجات' } },  
 { id: 'spray', name: { en: 'All Over Spray', ar: 'عطر الجسم' } },  
 { id: 'perfume', name: { en: 'Perfumes', ar: 'العطور' } },  
 { id: 'candle', name: { en: 'Candles', ar: 'الشموع' } }  
 ];  
  
   
 if (loading) return <LoadingMessage>{t('loading') || 'Loading...'}</LoadingMessage>;  
 if (error) return <ErrorMessage>{t('error')}: {error}</ErrorMessage>;  
  
 return (  
 <div className="container">  
 <PageHeader>{t('all\_over\_spray')}</PageHeader>  
   
 <CategoryNav>  
 {categories.map(category => (  
 <CategoryButton   
 key={category.id}  
 active={currentCategory === category.id}  
 onClick={() => setCurrentCategory(category.id)}  
 >  
 {category.name[i18n.language]}  
 </CategoryButton>  
 ))}  
 </CategoryNav>  
   
 <ProductsGrid>  
 {products.length > 0 ? (  
 products.map(product => (  
 <ProductItem key={product.\_id}>  
 <ProductCard product={product} isRTL={isRTL} />  
 </ProductItem>  
 ))  
 ) : (  
 <NoProductsMessage>{t('no\_products\_found') || 'No products found'}</NoProductsMessage>  
 )}  
 </ProductsGrid>  
 </div>  
 );  
};  
  
  
  
// Styled Components  
  
const NoProductsMessage = styled.div`  
 grid-column: 1 / -1;  
 text-align: center;  
 padding: 40px;  
 font-size: 18px;  
 color: var(--primary-color);  
`;  
const PageHeader = styled.h1`  
 text-align: center;  
 margin: 30px 0;  
 color: var(--primary-color);  
 font-size: 28px;  
 position: relative;  
   
 &:after {  
 content: '';  
 display: block;  
 width: 60px;  
 height: 2px;  
 background-color: var(--secondary-color);  
 margin: 15px auto 0;  
 }  
`;  
  
const CategoryNav = styled.div`  
 display: flex;  
 justify-content: center;  
 flex-wrap: wrap;  
 margin-bottom: 30px;  
`;  
  
const CategoryButton = styled.button`  
 background: none;  
 border: none;  
 padding: 8px 16px;  
 margin: 0 10px 10px;  
 color: ${props => props.active ? 'var(--secondary-color)' : 'var(--primary-color)'};  
 font-weight: ${props => props.active ? '600' : '400'};  
 font-size: 16px;  
 position: relative;  
 transition: all 0.2s ease;  
   
 &:after {  
 content: '';  
 position: absolute;  
 bottom: 0;  
 left: 50%;  
 transform: translateX(-50%);  
 width: ${props => props.active ? '30px' : '0'};  
 height: 2px;  
 background-color: var(--secondary-color);  
 transition: all 0.2s ease;  
 }  
   
 &:hover {  
 color: var(--secondary-color);  
   
 &:after {  
 width: 30px;  
 }  
 }  
`;  
  
const ProductsGrid = styled.div`  
 display: grid;  
 grid-template-columns: repeat(4, 1fr);  
 gap: 30px;  
   
 @media (max-width: 1200px) {  
 grid-template-columns: repeat(3, 1fr);  
 }  
   
 @media (max-width: 768px) {  
 grid-template-columns: repeat(2, 1fr);  
 gap: 20px;  
 }  
   
 @media (max-width: 480px) {  
 grid-template-columns: 1fr;  
 }  
`;  
  
const ProductItem = styled.div`  
 height: 100%;  
`;  
  
const LoadingMessage = styled.div`  
 text-align: center;  
 padding: 40px;  
 font-size: 18px;  
 color: var(--primary-color);  
`;  
  
const ErrorMessage = styled.div`  
 text-align: center;  
 padding: 40px;  
 font-size: 18px;  
 color: var(--danger-color);  
`;  
  
export default ProductsPage;

File Path: .\src\pages\Register.jsx

Content:

import { useState } from 'react';  
import { Link, useNavigate, useLocation } from 'react-router-dom';  
import { useTranslation } from 'react-i18next';  
import styled from 'styled-components';  
import { FaUser, FaEnvelope, FaLock, FaPhone, FaUserPlus } from 'react-icons/fa';  
import { useAuth } from '../context/AuthContext';  
import { useApp } from '../context/AppContext';  
  
const Register = () => {  
 const { t } = useTranslation();  
 const navigate = useNavigate();  
 const location = useLocation();  
 const { register } = useAuth();  
 const { addNotification } = useApp();  
   
 const [formData, setFormData] = useState({  
 name: '',  
 email: '',  
 password: '',  
 confirmPassword: '',  
 phone: ''  
 });  
   
 const [errors, setErrors] = useState({});  
 const [isSubmitting, setIsSubmitting] = useState(false);  
   
 const redirectTo = new URLSearchParams(location.search).get('redirect') || '/';  
  
 const validateForm = () => {  
 const newErrors = {};  
   
 if (!formData.name.trim()) {  
 newErrors.name = t('name\_required');  
 }  
   
 if (!formData.email.trim()) {  
 newErrors.email = t('email\_required');  
 } else if (!/\S+@\S+\.\S+/.test(formData.email)) {  
 newErrors.email = t('valid\_email\_required');  
 }  
   
 if (!formData.password) {  
 newErrors.password = t('password\_required');  
 } else if (formData.password.length < 6) {  
 newErrors.password = t('password\_length');  
 }  
   
 if (!formData.confirmPassword) {  
 newErrors.confirmPassword = t('confirm\_password\_required');  
 } else if (formData.confirmPassword !== formData.password) {  
 newErrors.confirmPassword = t('passwords\_not\_match');  
 }  
   
 setErrors(newErrors);  
 return Object.keys(newErrors).length === 0;  
 };  
  
 const handleChange = (e) => {  
 const { name, value } = e.target;  
 setFormData(prev => ({  
 ...prev,  
 [name]: value  
 }));  
   
 // Clear error when user types  
 if (errors[name]) {  
 setErrors(prev => ({  
 ...prev,  
 [name]: ''  
 }));  
 }  
 };  
  
 const handleSubmit = async (e) => {  
 e.preventDefault();  
   
 if (!validateForm()) {  
 return;  
 }  
   
 setIsSubmitting(true);  
   
 try {  
 // Omit confirmPassword from the data sent to the server  
 const { confirmPassword, ...registrationData } = formData;  
   
 await register(registrationData);  
 addNotification(t('registration\_success'), 'success');  
 navigate(redirectTo);  
 } catch (err) {  
 const errorMessage = err.response?.data?.message || t('registration\_failed');  
 setErrors(prev => ({  
 ...prev,  
 general: errorMessage  
 }));  
 } finally {  
 setIsSubmitting(false);  
 }  
 };  
  
 return (  
 <div className="container">  
 <RegisterContainer>  
 <RegisterImageContainer>  
 <img src="/images/register-image.jpg" alt="Register" />  
 </RegisterImageContainer>  
   
 <RegisterFormContainer>  
 <RegisterHeader>  
 <h1>{t('create\_account')}</h1>  
 <p>{t('register\_subtitle')}</p>  
 </RegisterHeader>  
   
 {errors.general && (  
 <ErrorMessage>{errors.general}</ErrorMessage>  
 )}  
   
 <RegisterForm onSubmit={handleSubmit}>  
 <FormGroup>  
 <FormLabel>  
 <FaUser />  
 <span>{t('full\_name')}</span>  
 </FormLabel>  
 <FormInput  
 type="text"  
 name="name"  
 placeholder={t('name\_placeholder')}  
 value={formData.name}  
 onChange={handleChange}  
 hasError={!!errors.name}  
 />  
 {errors.name && (  
 <ErrorMessage>{errors.name}</ErrorMessage>  
 )}  
 </FormGroup>  
   
 <FormGroup>  
 <FormLabel>  
 <FaEnvelope />  
 <span>{t('email\_address')}</span>  
 </FormLabel>  
 <FormInput  
 type="email"  
 name="email"  
 placeholder={t('email\_placeholder')}  
 value={formData.email}  
 onChange={handleChange}  
 hasError={!!errors.email}  
 />  
 {errors.email && (  
 <ErrorMessage>{errors.email}</ErrorMessage>  
 )}  
 </FormGroup>  
   
 <FormRow>  
 <FormGroup>  
 <FormLabel>  
 <FaLock />  
 <span>{t('password')}</span>  
 </FormLabel>  
 <FormInput  
 type="password"  
 name="password"  
 placeholder={t('password\_placeholder')}  
 value={formData.password}  
 onChange={handleChange}  
 hasError={!!errors.password}  
 />  
 {errors.password && (  
 <ErrorMessage>{errors.password}</ErrorMessage>  
 )}  
 </FormGroup>  
   
 <FormGroup>  
 <FormLabel>  
 <FaLock />  
 <span>{t('confirm\_password')}</span>  
 </FormLabel>  
 <FormInput  
 type="password"  
 name="confirmPassword"  
 placeholder={t('confirm\_password\_placeholder')}  
 value={formData.confirmPassword}  
 onChange={handleChange}  
 hasError={!!errors.confirmPassword}  
 />  
 {errors.confirmPassword && (  
 <ErrorMessage>{errors.confirmPassword}</ErrorMessage>  
 )}  
 </FormGroup>  
 </FormRow>  
   
 <FormGroup>  
 <FormLabel>  
 <FaPhone />  
 <span>{t('phone\_number')}</span>  
 </FormLabel>  
 <FormInput  
 type="tel"  
 name="phone"  
 placeholder={t('phone\_placeholder')}  
 value={formData.phone}  
 onChange={handleChange}  
 />  
 </FormGroup>  
   
 <TermsAgreement>  
 <input type="checkbox" id="terms" required />  
 <label htmlFor="terms">  
 {t('agree\_terms')}{' '}  
 <Link to="/terms">{t('terms\_conditions')}</Link>  
 </label>  
 </TermsAgreement>  
   
 <SubmitButton type="submit" disabled={isSubmitting}>  
 {isSubmitting ? t('creating\_account') : t('create\_account')}  
 {!isSubmitting && <FaUserPlus />}  
 </SubmitButton>  
 </RegisterForm>  
   
 <LoginPrompt>  
 {t('already\_account')}{' '}  
 <LoginLink to={`/login${location.search}`}>  
 {t('login')}  
 </LoginLink>  
 </LoginPrompt>  
 </RegisterFormContainer>  
 </RegisterContainer>  
 </div>  
 );  
};  
  
// Styled Components  
const RegisterContainer = styled.div`  
 display: grid;  
 grid-template-columns: 1fr 1fr;  
 min-height: 80vh;  
 margin: 50px 0;  
 box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);  
 border-radius: 8px;  
 overflow: hidden;  
   
 @media (max-width: 768px) {  
 grid-template-columns: 1fr;  
 }  
`;  
  
const RegisterImageContainer = styled.div`  
 overflow: hidden;  
   
 img {  
 width: 100%;  
 height: 100%;  
 object-fit: cover;  
 }  
   
 @media (max-width: 768px) {  
 display: none;  
 }  
`;  
  
const RegisterFormContainer = styled.div`  
 padding: 40px;  
 display: flex;  
 flex-direction: column;  
 justify-content: center;  
   
 @media (max-width: 480px) {  
 padding: 30px 20px;  
 }  
`;  
  
const RegisterHeader = styled.div`  
 margin-bottom: 30px;  
   
 h1 {  
 font-size: 28px;  
 color: var(--primary-color);  
 margin-bottom: 10px;  
 }  
   
 p {  
 color: #666;  
 }  
`;  
  
const RegisterForm = styled.form`  
 display: flex;  
 flex-direction: column;  
 gap: 20px;  
`;  
  
const FormGroup = styled.div`  
 display: flex;  
 flex-direction: column;  
 flex: 1;  
`;  
  
const FormRow = styled.div`  
 display: flex;  
 gap: 20px;  
   
 @media (max-width: 580px) {  
 flex-direction: column;  
 }  
`;  
  
const FormLabel = styled.label`  
 display: flex;  
 align-items: center;  
 gap: 8px;  
 margin-bottom: 8px;  
 color: var(--primary-color);  
 font-weight: 500;  
   
 svg {  
 color: var(--secondary-color);  
 }  
`;  
  
const FormInput = styled.input`  
 padding: 12px 15px;  
 border: 1px solid ${props => props.hasError ? 'var(--danger-color)' : '#ddd'};  
 border-radius: 4px;  
 font-size: 16px;  
   
 &:focus {  
 outline: none;  
 border-color: ${props => props.hasError ? 'var(--danger-color)' : 'var(--secondary-color)'};  
 }  
`;  
  
const ErrorMessage = styled.div`  
 color: var(--danger-color);  
 font-size: 14px;  
 margin-top: 5px;  
`;  
  
const TermsAgreement = styled.div`  
 display: flex;  
 align-items: flex-start;  
 gap: 10px;  
 color: #666;  
   
 input {  
 margin-top: 5px;  
 accent-color: var(--secondary-color);  
 }  
   
 a {  
 color: var(--secondary-color);  
 text-decoration: none;  
   
 &:hover {  
 text-decoration: underline;  
 }  
 }  
`;  
  
const SubmitButton = styled.button`  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 gap: 10px;  
 padding: 12px 15px;  
 background-color: var(--secondary-color);  
 color: var(--primary-color);  
 border: none;  
 border-radius: 4px;  
 font-size: 16px;  
 font-weight: 600;  
 cursor: pointer;  
 transition: all 0.3s ease;  
 margin-top: 10px;  
   
 &:hover {  
 background-color: var(--accent-color);  
 }  
   
 &:disabled {  
 opacity: 0.7;  
 cursor: not-allowed;  
 }  
`;  
  
const LoginPrompt = styled.div`  
 margin-top: 30px;  
 text-align: center;  
 color: #666;  
`;  
  
const LoginLink = styled(Link)`  
 color: var(--secondary-color);  
 text-decoration: none;  
 font-weight: 600;  
   
 &:hover {  
 text-decoration: underline;  
 }  
`;  
  
export default Register;

File Path: .\src\pages\admin\Dashboard.jsx

Content:

import { useState, useEffect } from 'react';  
import { Link } from 'react-router-dom';  
import styled from 'styled-components';  
import { FaBox, FaShoppingCart, FaUsers, FaDollarSign, FaChartLine } from 'react-icons/fa';  
import axios from 'axios';  
import AdminSidebar from '../../components/admin/Sidebar';  
  
const Dashboard = () => {  
 const [stats, setStats] = useState({  
 totalProducts: 0,  
 totalOrders: 0,  
 totalCustomers: 0,  
 totalRevenue: 0,  
 recentOrders: []  
 });  
   
 const [loading, setLoading] = useState(true);  
  
 useEffect(() => {  
 const fetchDashboardData = async () => {  
 try {  
 const response = await axios.get('/api/admin/dashboard');  
 setStats(response.data);  
 setLoading(false);  
 } catch (err) {  
 console.error('Error fetching dashboard data:', err);  
 setLoading(false);  
 }  
 };  
  
 // // Mock data for development  
 // setTimeout(() => {  
 // setStats({  
 // totalProducts: 24,  
 // totalOrders: 156,  
 // totalCustomers: 89,  
 // totalRevenue: 36540,  
 // recentOrders: [  
 // { id: '1001', customer: 'Ahmed Ali', date: '2023-05-07', total: 460, status: 'completed' },  
 // { id: '1002', customer: 'Sara Mohammed', date: '2023-05-06', total: 230, status: 'processing' },  
 // { id: '1003', customer: 'Khalid Ibrahim', date: '2023-05-05', total: 690, status: 'completed' },  
 // { id: '1004', customer: 'Fatima Ahmed', date: '2023-05-04', total: 230, status: 'shipped' },  
 // { id: '1005', customer: 'Omar Saeed', date: '2023-05-03', total: 460, status: 'completed' }  
 // ]  
 // });  
 // setLoading(false);  
 // }, 1000);  
  
 fetchDashboardData(); // Uncomment when backend is connected  
 }, []);  
  
 if (loading) return <div>Loading dashboard...</div>;  
  
 return (  
 <AdminLayout>  
 <AdminSidebar />  
 <MainContent>  
 <PageHeader>  
 <h1>Dashboard</h1>  
 <AdminBadge>Adming Access</AdminBadge>  
 <p>Welcome to your admin dashboard</p>  
 </PageHeader>  
   
 <StatsGrid>  
 <StatCard>  
 <StatIcon style={{ backgroundColor: '#e74c3c33' }}>  
 <FaBox style={{ color: '#e74c3c' }} />  
 </StatIcon>  
 <StatInfo>  
 <StatValue>{stats.totalProducts}</StatValue>  
 <StatLabel>Products</StatLabel>  
 </StatInfo>  
 </StatCard>  
   
 <StatCard>  
 <StatIcon style={{ backgroundColor: '#3498db33' }}>  
 <FaShoppingCart style={{ color: '#3498db' }} />  
 </StatIcon>  
 <StatInfo>  
 <StatValue>{stats.totalOrders}</StatValue>  
 <StatLabel>Orders</StatLabel>  
 </StatInfo>  
 </StatCard>  
   
 <StatCard>  
 <StatIcon style={{ backgroundColor: '#2ecc7133' }}>  
 <FaUsers style={{ color: '#2ecc71' }} />  
 </StatIcon>  
 <StatInfo>  
 <StatValue>{stats.totalCustomers}</StatValue>  
 <StatLabel>Customers</StatLabel>  
 </StatInfo>  
 </StatCard>  
   
 <StatCard>  
 <StatIcon style={{ backgroundColor: '#f39c1233' }}>  
 <FaDollarSign style={{ color: '#f39c12' }} />  
 </StatIcon>  
 <StatInfo>  
 <StatValue>﷼ {stats.totalRevenue.toLocaleString()}</StatValue>  
 <StatLabel>Revenue</StatLabel>  
 </StatInfo>  
 </StatCard>  
 </StatsGrid>  
   
 <SectionTitle>  
 <FaChartLine style={{ marginRight: '10px' }} />  
 Recent Orders  
 </SectionTitle>  
   
 <OrdersTable>  
 <thead>  
 <tr>  
 <th>Order ID</th>  
 <th>Customer</th>  
 <th>Date</th>  
 <th>Amount</th>  
 <th>Status</th>  
 <th>Action</th>  
 </tr>  
 </thead>  
 <tbody>  
 {stats.recentOrders.map((order) => (  
 <tr key={order.id}>  
 <td>#{order.id}</td>  
 <td>{order.customer}</td>  
 <td>{new Date(order.date).toLocaleDateString()}</td>  
 <td>﷼ {order.total.toLocaleString()}</td>  
 <td>  
 <OrderStatus status={order.status}>  
 {order.status.charAt(0).toUpperCase() + order.status.slice(1)}  
 </OrderStatus>  
 </td>  
 <td>  
 <Link to={`/admin/orders/${order.id}`}>View</Link>  
 </td>  
 </tr>  
 ))}  
 </tbody>  
 </OrdersTable>  
 </MainContent>  
 </AdminLayout>  
 );  
};  
  
// Styled Components  
const AdminLayout = styled.div`  
 display: flex;  
 min-height: 100vh;  
`;  
  
const MainContent = styled.main`  
 flex: 1;  
 padding: 20px;  
 background-color: #f5f7fa;  
`;  
  
const PageHeader = styled.div`  
 margin-bottom: 30px;  
   
 h1 {  
 font-size: 24px;  
 color: var(--primary-color);  
 margin-bottom: 5px;  
 }  
   
 p {  
 color: #777;  
 font-size: 14px;  
 }  
`;  
  
const StatsGrid = styled.div`  
 display: grid;  
 grid-template-columns: repeat(4, 1fr);  
 gap: 20px;  
 margin-bottom: 30px;  
   
 @media (max-width: 1024px) {  
 grid-template-columns: repeat(2, 1fr);  
 }  
   
 @media (max-width: 580px) {  
 grid-template-columns: 1fr;  
 }  
`;  
  
const StatCard = styled.div`  
 background-color: white;  
 border-radius: 8px;  
 padding: 20px;  
 box-shadow: 0 2px 5px rgba(0, 0, 0, 0.05);  
 display: flex;  
 align-items: center;  
`;  
  
const StatIcon = styled.div`  
 width: 50px;  
 height: 50px;  
 border-radius: 8px;  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 font-size: 20px;  
 margin-right: 15px;  
`;  
  
const StatInfo = styled.div``;  
  
const StatValue = styled.div`  
 font-size: 20px;  
 font-weight: 600;  
 color: var(--primary-color);  
 margin-bottom: 5px;  
`;  
  
const StatLabel = styled.div`  
 font-size: 14px;  
 color: #777;  
`;  
  
const SectionTitle = styled.h2`  
 font-size: 18px;  
 color: var(--primary-color);  
 margin-bottom: 20px;  
 display: flex;  
 align-items: center;  
`;  
  
const OrdersTable = styled.table`  
 width: 100%;  
 border-collapse: collapse;  
 background-color: white;  
 border-radius: 8px;  
 overflow: hidden;  
 box-shadow: 0 2px 5px rgba(0, 0, 0, 0.05);  
   
 th, td {  
 padding: 15px;  
 text-align: left;  
 border-bottom: 1px solid #eee;  
 }  
   
 th {  
 background-color: #f9f9f9;  
 font-weight: 600;  
 color: var(--primary-color);  
 }  
   
 tbody tr:hover {  
 background-color: #f9f9f9;  
 }  
   
 a {  
 color: var(--secondary-color);  
 text-decoration: none;  
   
 &:hover {  
 text-decoration: underline;  
 }  
 }  
`;  
  
const OrderStatus = styled.span`  
 display: inline-block;  
 padding: 4px 10px;  
 border-radius: 20px;  
 font-size: 12px;  
 font-weight: 500;  
   
 ${props => {  
 switch (props.status) {  
 case 'completed':  
 return `  
 background-color: #e6f7e9;  
 color: #2ecc71;  
 `;  
 case 'processing':  
 return `  
 background-color: #e6f0f9;  
 color: #3498db;  
 `;  
 case 'shipped':  
 return `  
 background-color: #fef4e6;  
 color: #f39c12;  
 `;  
 case 'cancelled':  
 return `  
 background-color: #fde9e9;  
 color: #e74c3c;  
 `;  
 default:  
 return `  
 background-color: #f5f5f5;  
 color: #777;  
 `;  
 }  
 }}  
`;  
const AdminBadge = styled.span`  
 background-color: var(--secondary-color);  
 color: var(--primary-color);  
 padding: 4px 10px;  
 border-radius: 20px;  
 font-size: 12px;  
 font-weight: 500;  
 margin-left: 10px;  
`;  
  
export default Dashboard;

File Path: .\src\pages\admin\Orders.jsx

Content:

import { useState, useEffect } from 'react';  
import { Link } from 'react-router-dom';  
import styled from 'styled-components';  
import { FaSearch, FaEye, FaFileDownload } from 'react-icons/fa';  
import axios from 'axios';  
import AdminSidebar from '../../components/admin/Sidebar';  
  
const AdminOrders = () => {  
 const [orders, setOrders] = useState([]);  
 const [loading, setLoading] = useState(true);  
 const [searchQuery, setSearchQuery] = useState('');  
 const [filterStatus, setFilterStatus] = useState('all');  
 const [currentPage, setCurrentPage] = useState(1);  
 const [totalPages, setTotalPages] = useState(1);  
 const [selectedOrder, setSelectedOrder] = useState(null);  
 const [showOrderDetails, setShowOrderDetails] = useState(false);  
  
 useEffect(() => {  
 const fetchOrders = async () => {  
 try {  
 let url = `/api/admin/orders?page=${currentPage}`;  
   
 if (filterStatus !== 'all') {  
 url += `&status=${filterStatus}`;  
 }  
   
 if (searchQuery) {  
 url += `&search=${searchQuery}`;  
 }  
   
 const response = await axios.get(url);  
 setOrders(response.data.orders);  
 setTotalPages(response.data.totalPages);  
 setLoading(false);  
 } catch (err) {  
 console.error('Error fetching orders:', err);  
 setLoading(false);  
 }  
 };  
  
 // Mock data for development  
 setTimeout(() => {  
 const mockOrders = [  
 {  
 \_id: '1001',  
 orderNumber: 'ORD-1001',  
 customer: {  
 name: 'Ahmed Ali',  
 email: 'ahmed@example.com',  
 phone: '+966 123 456 789'  
 },  
 items: [  
 {  
 product: {  
 \_id: '1',  
 name: 'Musk 100ml',  
 price: 230.00,  
 image: '/images/musk\_bottle.png'  
 },  
 quantity: 2,  
 price: 230.00  
 },  
 {  
 product: {  
 \_id: '2',  
 name: 'The fifth 100ml',  
 price: 230.00,  
 image: '/images/fifth\_bottle.png'  
 },  
 quantity: 1,  
 price: 230.00  
 }  
 ],  
 totalAmount: 690.00,  
 status: 'completed',  
 shippingAddress: {  
 address: '123 Main St, Riyadh',  
 city: 'Riyadh',  
 postalCode: '12345',  
 country: 'Saudi Arabia'  
 },  
 paymentMethod: 'credit\_card',  
 createdAt: '2023-05-10T10:30:00.000Z',  
 updatedAt: '2023-05-12T14:20:00.000Z'  
 },  
 {  
 \_id: '1002',  
 orderNumber: 'ORD-1002',  
 customer: {  
 name: 'Sara Mohammed',  
 email: 'sara@example.com',  
 phone: '+966 987 654 321'  
 },  
 items: [  
 {  
 product: {  
 \_id: '4',  
 name: 'Bliss 100ml',  
 price: 230.00,  
 image: '/images/bliss\_bottle.png'  
 },  
 quantity: 1,  
 price: 230.00  
 }  
 ],  
 totalAmount: 230.00,  
 status: 'processing',  
 shippingAddress: {  
 address: '456 Oak St, Jeddah',  
 city: 'Jeddah',  
 postalCode: '54321',  
 country: 'Saudi Arabia'  
 },  
 paymentMethod: 'cash\_on\_delivery',  
 createdAt: '2023-05-15T09:15:00.000Z',  
 updatedAt: '2023-05-15T09:15:00.000Z'  
 },  
 {  
 \_id: '1003',  
 orderNumber: 'ORD-1003',  
 customer: {  
 name: 'Khalid Ibrahim',  
 email: 'khalid@example.com',  
 phone: '+966 456 789 123'  
 },  
 items: [  
 {  
 product: {  
 \_id: '3',  
 name: 'Flow of three 100ml',  
 price: 230.00,  
 image: '/images/flow\_bottle.png'  
 },  
 quantity: 1,  
 price: 230.00  
 },  
 {  
 product: {  
 \_id: '1',  
 name: 'Musk 100ml',  
 price: 230.00,  
 image: '/images/musk\_bottle.png'  
 },  
 quantity: 2,  
 price: 230.00  
 }  
 ],  
 totalAmount: 690.00,  
 status: 'shipped',  
 shippingAddress: {  
 address: '789 Pine St, Dammam',  
 city: 'Dammam',  
 postalCode: '67890',  
 country: 'Saudi Arabia'  
 },  
 paymentMethod: 'credit\_card',  
 createdAt: '2023-05-18T15:45:00.000Z',  
 updatedAt: '2023-05-19T11:30:00.000Z'  
 },  
 {  
 \_id: '1004',  
 orderNumber: 'ORD-1004',  
 customer: {  
 name: 'Fatima Ahmed',  
 email: 'fatima@example.com',  
 phone: '+966 789 123 456'  
 },  
 items: [  
 {  
 product: {  
 \_id: '2',  
 name: 'The fifth 100ml',  
 price: 230.00,  
 image: '/images/fifth\_bottle.png'  
 },  
 quantity: 1,  
 price: 230.00  
 }  
 ],  
 totalAmount: 230.00,  
 status: 'cancelled',  
 shippingAddress: {  
 address: '321 Elm St, Mecca',  
 city: 'Mecca',  
 postalCode: '13579',  
 country: 'Saudi Arabia'  
 },  
 paymentMethod: 'cash\_on\_delivery',  
 createdAt: '2023-05-20T08:00:00.000Z',  
 updatedAt: '2023-05-21T10:15:00.000Z'  
 }  
 ];  
   
 // Filter orders based on search and status filter  
 let filteredOrders = mockOrders;  
   
 if (filterStatus !== 'all') {  
 filteredOrders = filteredOrders.filter(order => order.status === filterStatus);  
 }  
   
 if (searchQuery) {  
 const query = searchQuery.toLowerCase();  
 filteredOrders = filteredOrders.filter(order =>   
 order.orderNumber.toLowerCase().includes(query) ||  
 order.customer.name.toLowerCase().includes(query) ||  
 order.customer.email.toLowerCase().includes(query)  
 );  
 }  
   
 setOrders(filteredOrders);  
 setTotalPages(Math.ceil(filteredOrders.length / 10));  
 setLoading(false);  
 }, 1000);  
  
 // fetchOrders(); // Uncomment when backend is connected  
 }, [currentPage, filterStatus, searchQuery]);  
  
 const handleSearch = (e) => {  
 e.preventDefault();  
 setCurrentPage(1); // Reset to first page on new search  
 };  
  
 const handleViewOrder = (order) => {  
 setSelectedOrder(order);  
 setShowOrderDetails(true);  
 };  
  
 const handleStatusChange = async (orderId, newStatus) => {  
 try {  
 // For mock data, just update locally  
 setOrders(orders.map(order =>   
 order.\_id === orderId ? { ...order, status: newStatus } : order  
 ));  
   
 if (selectedOrder && selectedOrder.\_id === orderId) {  
 setSelectedOrder({ ...selectedOrder, status: newStatus });  
 }  
   
 // With backend:  
 // await axios.put(`/api/admin/orders/${orderId}/status`, { status: newStatus });  
 // Refresh orders list  
 } catch (err) {  
 console.error('Error updating order status:', err);  
 }  
 };  
  
 const formatDate = (dateString) => {  
 const options = { year: 'numeric', month: 'long', day: 'numeric', hour: '2-digit', minute: '2-digit' };  
 return new Date(dateString).toLocaleDateString(undefined, options);  
 };  
  
 if (loading) return <div>Loading orders...</div>;  
  
 return (  
 <AdminLayout>  
 <AdminSidebar />  
 <MainContent>  
 <PageHeader>  
 <h1>Orders Management</h1>  
 <p>Manage customer orders</p>  
 </PageHeader>  
   
 <FilterBar>  
 <SearchForm onSubmit={handleSearch}>  
 <SearchInput  
 type="text"  
 placeholder="Search orders..."  
 value={searchQuery}  
 onChange={(e) => setSearchQuery(e.target.value)}  
 />  
 <SearchButton type="submit">  
 <FaSearch />  
 </SearchButton>  
 </SearchForm>  
   
 <FilterDropdown  
 value={filterStatus}  
 onChange={(e) => {  
 setFilterStatus(e.target.value);  
 setCurrentPage(1); // Reset to first page on filter change  
 }}  
 >  
 <option value="all">All Orders</option>  
 <option value="pending">Pending</option>  
 <option value="processing">Processing</option>  
 <option value="shipped">Shipped</option>  
 <option value="completed">Completed</option>  
 <option value="cancelled">Cancelled</option>  
 </FilterDropdown>  
 </FilterBar>  
   
 <OrdersTable>  
 <thead>  
 <tr>  
 <th>Order #</th>  
 <th>Customer</th>  
 <th>Date</th>  
 <th>Total</th>  
 <th>Status</th>  
 <th>Actions</th>  
 </tr>  
 </thead>  
 <tbody>  
 {orders.map((order) => (  
 <tr key={order.\_id}>  
 <td>{order.orderNumber}</td>  
 <td>{order.customer.name}</td>  
 <td>{formatDate(order.createdAt)}</td>  
 <td>﷼ {order.totalAmount.toFixed(2)}</td>  
 <td>  
 <OrderStatus status={order.status}>  
 {order.status.charAt(0).toUpperCase() + order.status.slice(1)}  
 </OrderStatus>  
 </td>  
 <td>  
 <ActionButtons>  
 <ViewButton onClick={() => handleViewOrder(order)}>  
 <FaEye />  
 </ViewButton>  
 <InvoiceButton>  
 <FaFileDownload />  
 </InvoiceButton>  
 </ActionButtons>  
 </td>  
 </tr>  
 ))}  
 </tbody>  
 </OrdersTable>  
   
 {orders.length === 0 && (  
 <NoOrdersMessage>No orders found matching your criteria.</NoOrdersMessage>  
 )}  
   
 {totalPages > 1 && (  
 <Pagination>  
 <PaginationButton   
 onClick={() => setCurrentPage(prev => Math.max(prev - 1, 1))}  
 disabled={currentPage === 1}  
 >  
 Previous  
 </PaginationButton>  
   
 <PageInfo>  
 Page {currentPage} of {totalPages}  
 </PageInfo>  
   
 <PaginationButton   
 onClick={() => setCurrentPage(prev => Math.min(prev + 1, totalPages))}  
 disabled={currentPage === totalPages}  
 >  
 Next  
 </PaginationButton>  
 </Pagination>  
 )}  
   
 {showOrderDetails && selectedOrder && (  
 <OrderDetailsOverlay>  
 <OrderDetailsModal>  
 <ModalHeader>  
 <h2>Order Details - {selectedOrder.orderNumber}</h2>  
 <CloseButton onClick={() => setShowOrderDetails(false)}>×</CloseButton>  
 </ModalHeader>  
   
 <ModalContent>  
 <OrderDetailsSection>  
 <h3>Order Information</h3>  
 <OrderDetail>  
 <span>Status:</span>  
 <OrderStatus status={selectedOrder.status}>  
 {selectedOrder.status.charAt(0).toUpperCase() + selectedOrder.status.slice(1)}  
 </OrderStatus>  
 </OrderDetail>  
 <OrderDetail>  
 <span>Date:</span>  
 <div>{formatDate(selectedOrder.createdAt)}</div>  
 </OrderDetail>  
 <OrderDetail>  
 <span>Payment Method:</span>  
 <div>  
 {selectedOrder.paymentMethod === 'credit\_card' ? 'Credit Card' : 'Cash on Delivery'}  
 </div>  
 </OrderDetail>  
 </OrderDetailsSection>  
   
 <OrderDetailsGrid>  
 <OrderDetailsSection>  
 <h3>Customer</h3>  
 <div>  
 <p><strong>{selectedOrder.customer.name}</strong></p>  
 <p>{selectedOrder.customer.email}</p>  
 <p>{selectedOrder.customer.phone}</p>  
 </div>  
 </OrderDetailsSection>  
   
 <OrderDetailsSection>  
 <h3>Shipping Address</h3>  
 <div>  
 <p>{selectedOrder.shippingAddress.address}</p>  
 <p>{selectedOrder.shippingAddress.city}, {selectedOrder.shippingAddress.postalCode}</p>  
 <p>{selectedOrder.shippingAddress.country}</p>  
 </div>  
 </OrderDetailsSection>  
 </OrderDetailsGrid>  
   
 <OrderDetailsSection>  
 <h3>Order Items</h3>  
 <OrderItemsTable>  
 <thead>  
 <tr>  
 <th>Product</th>  
 <th>Price</th>  
 <th>Quantity</th>  
 <th>Total</th>  
 </tr>  
 </thead>  
 <tbody>  
 {selectedOrder.items.map((item, index) => (  
 <tr key={index}>  
 <td>  
 <OrderItemInfo>  
 <OrderItemImage src={item.product.image} alt={item.product.name} />  
 <div>{item.product.name}</div>  
 </OrderItemInfo>  
 </td>  
 <td>﷼ {item.price.toFixed(2)}</td>  
 <td>{item.quantity}</td>  
 <td>﷼ {(item.price \* item.quantity).toFixed(2)}</td>  
 </tr>  
 ))}  
 </tbody>  
 </OrderItemsTable>  
 </OrderDetailsSection>  
   
 <OrderSummary>  
 <OrderSummaryRow>  
 <span>Subtotal:</span>  
 <span>﷼ {selectedOrder.totalAmount.toFixed(2)}</span>  
 </OrderSummaryRow>  
 <OrderSummaryRow>  
 <span>Shipping:</span>  
 <span>﷼ 30.00</span>  
 </OrderSummaryRow>  
 <OrderSummaryRow total>  
 <span>Total:</span>  
 <span>﷼ {(selectedOrder.totalAmount + 30).toFixed(2)}</span>  
 </OrderSummaryRow>  
 </OrderSummary>  
 </ModalContent>  
   
 <ModalFooter>  
 <UpdateStatus>  
 <span>Update Status:</span>  
 <StatusSelect  
 value={selectedOrder.status}  
 onChange={(e) => handleStatusChange(selectedOrder.\_id, e.target.value)}  
 >  
 <option value="pending">Pending</option>  
 <option value="processing">Processing</option>  
 <option value="shipped">Shipped</option>  
 <option value="completed">Completed</option>  
 <option value="cancelled">Cancelled</option>  
 </StatusSelect>  
 </UpdateStatus>  
   
 <div>  
 <CloseModalButton onClick={() => setShowOrderDetails(false)}>  
 Close  
 </CloseModalButton>  
 <PrintInvoiceButton>  
 Print Invoice  
 </PrintInvoiceButton>  
 </div>  
 </ModalFooter>  
 </OrderDetailsModal>  
 </OrderDetailsOverlay>  
 )}  
 </MainContent>  
 </AdminLayout>  
 );  
};  
  
// Styled Components  
const AdminLayout = styled.div`  
 display: flex;  
 min-height: 100vh;  
`;  
  
const MainContent = styled.main`  
 flex: 1;  
 padding: 20px;  
 background-color: #f5f7fa;  
`;  
  
const PageHeader = styled.div`  
 margin-bottom: 30px;  
   
 h1 {  
 font-size: 24px;  
 color: var(--primary-color);  
 margin-bottom: 5px;  
 }  
   
 p {  
 color: #777;  
 font-size: 14px;  
 }  
`;  
  
const FilterBar = styled.div`  
 display: flex;  
 justify-content: space-between;  
 margin-bottom: 20px;  
   
 @media (max-width: 580px) {  
 flex-direction: column;  
 gap: 15px;  
 }  
`;  
  
const SearchForm = styled.form`  
 display: flex;  
 max-width: 400px;  
 width: 100%;  
`;  
  
const SearchInput = styled.input`  
 flex: 1;  
 padding: 10px 15px;  
 border: 1px solid #ddd;  
 border-radius: 4px 0 0 4px;  
 font-size: 14px;  
   
 &:focus {  
 outline: none;  
 border-color: var(--secondary-color);  
 }  
`;  
  
const SearchButton = styled.button`  
 padding: 0 15px;  
 background-color: var(--secondary-color);  
 color: var(--primary-color);  
 border: none;  
 border-radius: 0 4px 4px 0;  
 cursor: pointer;  
 transition: all 0.2s ease;  
   
 &:hover {  
 background-color: var(--accent-color);  
 }  
`;  
  
const FilterDropdown = styled.select`  
 padding: 10px 15px;  
 border: 1px solid #ddd;  
 border-radius: 4px;  
 font-size: 14px;  
   
 &:focus {  
 outline: none;  
 border-color: var(--secondary-color);  
 }  
`;  
  
const OrdersTable = styled.table`  
 width: 100%;  
 border-collapse: collapse;  
 background-color: white;  
 border-radius: 8px;  
 overflow: hidden;  
 box-shadow: 0 2px 5px rgba(0, 0, 0, 0.05);  
   
 th, td {  
 padding: 15px;  
 text-align: left;  
 border-bottom: 1px solid #eee;  
 }  
   
 th {  
 background-color: #f9f9f9;  
 font-weight: 600;  
 color: var(--primary-color);  
 }  
   
 tbody tr:hover {  
 background-color: #f9f9f9;  
 }  
   
 @media (max-width: 768px) {  
 display: block;  
   
 thead, tbody, tr, th, td {  
 display: block;  
 }  
   
 thead tr {  
 position: absolute;  
 top: -9999px;  
 left: -9999px;  
 }  
   
 tr {  
 border: 1px solid #eee;  
 margin-bottom: 20px;  
 position: relative;  
 padding: 10px;  
 }  
   
 td {  
 border: none;  
 position: relative;  
 padding-left: 50%;  
   
 &:before {  
 position: absolute;  
 top: 15px;  
 left: 10px;  
 width: 45%;  
 white-space: nowrap;  
 font-weight: 600;  
 }  
   
 &:nth-of-type(1):before { content: "Order #"; }  
 &:nth-of-type(2):before { content: "Customer"; }  
 &:nth-of-type(3):before { content: "Date"; }  
 &:nth-of-type(4):before { content: "Total"; }  
 &:nth-of-type(5):before { content: "Status"; }  
 &:nth-of-type(6):before { content: "Actions"; }  
 }  
 }  
`;  
  
const OrderStatus = styled.span`  
 display: inline-block;  
 padding: 4px 10px;  
 border-radius: 20px;  
 font-size: 12px;  
 font-weight: 500;  
   
 ${props => {  
 switch (props.status) {  
 case 'completed':  
 return `  
 background-color: #e6f7e9;  
 color: #2ecc71;  
 `;  
 case 'processing':  
 return `  
 background-color: #e6f0f9;  
 color: #3498db;  
 `;  
 case 'shipped':  
 return `  
 background-color: #fef4e6;  
 color: #f39c12;  
 `;  
 case 'pending':  
 return `  
 background-color: #f5f5f5;  
 color: #777;  
 `;  
 case 'cancelled':  
 return `  
 background-color: #fde9e9;  
 color: #e74c3c;  
 `;  
 default:  
 return `  
 background-color: #f5f5f5;  
 color: #777;  
 `;  
 }  
 }}  
`;  
  
const ActionButtons = styled.div`  
 display: flex;  
 gap: 10px;  
`;  
  
const ViewButton = styled.button`  
 background-color: #e6f0f9;  
 color: #3498db;  
 border: none;  
 width: 32px;  
 height: 32px;  
 border-radius: 4px;  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 cursor: pointer;  
   
 &:hover {  
 background-color: #3498db;  
 color: white;  
 }  
`;  
  
const InvoiceButton = styled.button`  
 background-color: #f5f5f5;  
 color: #777;  
 border: none;  
 width: 32px;  
 height: 32px;  
 border-radius: 4px;  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 cursor: pointer;  
   
 &:hover {  
 background-color: #777;  
 color: white;  
 }  
`;  
  
const NoOrdersMessage = styled.div`  
 text-align: center;  
 padding: 50px 0;  
 color: #777;  
 font-size: 16px;  
`;  
  
const Pagination = styled.div`  
 display: flex;  
 justify-content: center;  
 align-items: center;  
 margin-top: 30px;  
`;  
  
const PaginationButton = styled.button`  
 padding: 8px 16px;  
 background-color: white;  
 border: 1px solid #ddd;  
 color: var(--primary-color);  
 cursor: pointer;  
 transition: all 0.2s ease;  
   
 &:first-child {  
 border-radius: 4px 0 0 4px;  
 }  
   
 &:last-child {  
 border-radius: 0 4px 4px 0;  
 }  
   
 &:hover {  
 background-color: #f5f5f5;  
 }  
   
 &:disabled {  
 opacity: 0.5;  
 cursor: not-allowed;  
 }  
`;  
  
const PageInfo = styled.div`  
 padding: 8px 16px;  
 background-color: #f9f9f9;  
 border-top: 1px solid #ddd;  
 border-bottom: 1px solid #ddd;  
 color: #777;  
 font-size: 14px;  
`;  
  
const OrderDetailsOverlay = styled.div`  
 position: fixed;  
 top: 0;  
 left: 0;  
 right: 0;  
 bottom: 0;  
 background-color: rgba(0, 0, 0, 0.5);  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 z-index: 1000;  
`;  
  
const OrderDetailsModal = styled.div`  
 background-color: white;  
 border-radius: 8px;  
 width: 90%;  
 max-width: 900px;  
 max-height: 90vh;  
 overflow-y: auto;  
 box-shadow: 0 4px 15px rgba(0, 0, 0, 0.15);  
`;  
  
const ModalHeader = styled.div`  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 padding: 20px;  
 border-bottom: 1px solid #eee;  
 position: sticky;  
 top: 0;  
 background-color: white;  
 z-index: 1;  
   
 h2 {  
 font-size: 20px;  
 color: var(--primary-color);  
 margin: 0;  
 }  
`;  
  
const CloseButton = styled.button`  
 background: none;  
 border: none;  
 font-size: 24px;  
 cursor: pointer;  
 color: #777;  
   
 &:hover {  
 color: #e74c3c;  
 }  
`;  
  
const ModalContent = styled.div`  
 padding: 20px;  
`;  
  
const OrderDetailsSection = styled.div`  
 margin-bottom: 30px;  
   
 h3 {  
 font-size: 16px;  
 color: var(--primary-color);  
 margin-bottom: 15px;  
 position: relative;  
   
 &:after {  
 content: '';  
 display: block;  
 width: 30px;  
 height: 2px;  
 background-color: var(--secondary-color);  
 margin-top: 5px;  
 }  
 }  
`;  
  
const OrderDetail = styled.div`  
 display: flex;  
 margin-bottom: 10px;  
   
 span {  
 width: 150px;  
 font-weight: 600;  
 color: #555;  
 }  
`;  
  
const OrderDetailsGrid = styled.div`  
 display: grid;  
 grid-template-columns: 1fr 1fr;  
 gap: 30px;  
   
 @media (max-width: 768px) {  
 grid-template-columns: 1fr;  
 gap: 20px;  
 }  
`;  
  
const OrderItemsTable = styled.table`  
 width: 100%;  
 border-collapse: collapse;  
   
 th, td {  
 padding: 10px;  
 text-align: left;  
 border-bottom: 1px solid #eee;  
 }  
   
 th {  
 font-weight: 600;  
 color: var(--primary-color);  
 }  
`;  
  
const OrderItemInfo = styled.div`  
 display: flex;  
 align-items: center;  
`;  
  
const OrderItemImage = styled.img`  
 width: 40px;  
 height: 40px;  
 object-fit: contain;  
 margin-right: 10px;  
`;  
  
const OrderSummary = styled.div`  
 margin-top: 30px;  
 border-top: 1px solid #eee;  
 padding-top: 20px;  
`;  
  
const OrderSummaryRow = styled.div`  
 display: flex;  
 justify-content: space-between;  
 padding: 10px 0;  
   
 ${props => props.total && `  
 font-weight: 600;  
 font-size: 18px;  
 color: var(--secondary-color);  
 border-top: 1px solid #eee;  
 margin-top: 10px;  
 padding-top: 15px;  
 `}  
`;  
  
const ModalFooter = styled.div`  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 padding: 20px;  
 border-top: 1px solid #eee;  
 position: sticky;  
 bottom: 0;  
 background-color: white;  
 z-index: 1;  
   
 @media (max-width: 768px) {  
 flex-direction: column;  
 gap: 15px;  
 }  
`;  
  
const UpdateStatus = styled.div`  
 display: flex;  
 align-items: center;  
   
 span {  
 margin-right: 10px;  
 font-weight: 500;  
 }  
`;  
  
const StatusSelect = styled.select`  
 padding: 8px 15px;  
 border: 1px solid #ddd;  
 border-radius: 4px;  
   
 &:focus {  
 outline: none;  
 border-color: var(--secondary-color);  
 }  
`;  
  
const CloseModalButton = styled.button`  
 padding: 8px 15px;  
 background-color: #f5f5f5;  
 color: #333;  
 border: none;  
 border-radius: 4px;  
 cursor: pointer;  
 margin-right: 10px;  
   
 &:hover {  
 background-color: #e0e0e0;  
 }  
`;  
  
const PrintInvoiceButton = styled.button`  
 padding: 8px 15px;  
 background-color: var(--secondary-color);  
 color: var(--primary-color);  
 border: none;  
 border-radius: 4px;  
 cursor: pointer;  
   
 &:hover {  
 background-color: var(--accent-color);  
 }  
`;  
  
export default AdminOrders;

File Path: .\src\pages\admin\Products.jsx

Content:

import { useState, useEffect } from 'react';  
import styled from 'styled-components';  
import { FaEdit, FaTrash, FaPlus, FaSearch } from 'react-icons/fa';  
import axios from 'axios';  
import AdminSidebar from '../../components/admin/Sidebar';  
import { productService } from '../../services'; // Use service instead of direct axios  
import { adminService } from '../../services'; // Use service instead of direct axios  
  
  
const AdminProducts = () => {  
 const [products, setProducts] = useState([]);  
 const [loading, setLoading] = useState(true);  
 const [searchQuery, setSearchQuery] = useState('');  
 const [showAddForm, setShowAddForm] = useState(false);  
 const [newProduct, setNewProduct] = useState({  
 name: '',  
 nameAr: '',  
 price: '',  
 category: '',  
 description: '',  
 descriptionAr: '',  
 image: null,  
 boxImage: null  
 });  
 const [editingProduct, setEditingProduct] = useState(null);  
 const [error, setError] = useState(null); // Add error state  
  
  
 useEffect(() => {  
 const fetchProducts = async () => {  
 try {  
 const data = await productService.getAllProducts();  
 // Ensure products is always an array  
 setProducts(Array.isArray(data) ? data : []);  
 setError(null);  
 } catch (err) {  
 console.error('Error fetching products:', err);  
 setProducts([]); // Set to empty array on error  
 setError('Failed to load products. Please try again.');  
 } finally {  
 setLoading(false);  
 }  
 };  
  
 // Mock data for development  
 // setTimeout(() => {  
 // const mockProducts = [  
 // {  
 // \_id: '1',  
 // name: 'Musk 100ml',  
 // nameAr: 'مسك 100مل',  
 // price: 230.00,  
 // currency: '﷼',  
 // image: '/images/musk\_bottle.png',  
 // boxImage: '/images/musk\_box.png',  
 // category: 'spray',  
 // stock: 45  
 // },  
 // {  
 // \_id: '2',  
 // name: 'The fifth 100ml',  
 // nameAr: 'ذا فيفث 100مل',  
 // price: 230.00,  
 // currency: '﷼',  
 // image: '/images/fifth\_bottle.png',  
 // boxImage: '/images/fifth\_box.png',  
 // category: 'spray',  
 // stock: 32  
 // },  
 // {  
 // \_id: '3',  
 // name: 'Flow of three 100ml',  
 // nameAr: 'فلو أوف ثري 100مل',  
 // price: 230.00,  
 // currency: '﷼',  
 // image: '/images/flow\_bottle.png',  
 // boxImage: '/images/flow\_box.png',  
 // category: 'spray',  
 // stock: 18  
 // },  
 // {  
 // \_id: '4',  
 // name: 'Bliss 100ml',  
 // nameAr: 'بليس 100مل',  
 // price: 230.00,  
 // currency: '﷼',  
 // image: '/images/bliss\_bottle.png',  
 // boxImage: '/images/bliss\_box.png',  
 // category: 'spray',  
 // stock: 27  
 // }  
 // ];  
   
 // setProducts(mockProducts);  
 // setLoading(false);  
 // }, 1000);  
  
 fetchProducts(); // Uncomment when backend is connected  
 }, []);  
  
 const filteredProducts = Array.isArray(products) ?   
 products.filter(product =>   
 product && (  
 (product.name && product.name.toLowerCase().includes(searchQuery.toLowerCase())) ||   
 (product.nameAr && product.nameAr.includes(searchQuery))  
 )  
 ) : [];  
  
 const handleInputChange = (e) => {  
 const { name, value } = e.target;  
 setNewProduct({  
 ...newProduct,  
 [name]: value  
 });  
 };  
  
 const handleFileChange = (e) => {  
 const { name, files } = e.target;  
 setNewProduct({  
 ...newProduct,  
 [name]: files[0]  
 });  
 };  
  
 const handleAddProduct = async (e) => {  
 e.preventDefault();  
 try {  
 const formData = new FormData();  
   
 Object.keys(newProduct).forEach(key => {  
 if (key === 'image' || key === 'boxImage') {  
 if (newProduct[key]) {  
 formData.append(key, newProduct[key]);  
 }  
 } else {  
 formData.append(key, newProduct[key]);  
 }  
 });  
   
 // Mock response for development  
 // const mockResponse = {  
 // \_id: Date.now().toString(),  
 // ...newProduct,  
 // image: '/images/new\_bottle.png',  
 // boxImage: '/images/new\_box.png',  
 // currency: '﷼',  
 // stock: 20  
 // };  
   
 // setProducts([...products, mockResponse]);  
   
  
   
 // With backend:  
 const newProductData = await adminService.createProduct(formData);  
  
  
 setProducts(prevProducts =>   
 Array.isArray(prevProducts) ? [...prevProducts, newProductData] : [newProductData]  
 );  
 setNewProduct({  
 name: '',  
 nameAr: '',  
 price: '',  
 category: '',  
 description: '',  
 descriptionAr: '',  
 image: null,  
 boxImage: null  
 });  
 setShowAddForm(false);  
 setError(null);  
 } catch (err) {  
 console.error('Error adding product:', err);  
 setError('Failed to add product. Please try again.');  
 }  
 };  
  
 const handleEditProduct = (product) => {  
 setEditingProduct({  
 ...product,  
 image: null,  
 boxImage: null  
 });  
 setShowAddForm(true);  
 };  
  
 const handleUpdateProduct = async (e) => {  
 e.preventDefault();  
 try {  
 const formData = new FormData();  
   
 Object.keys(editingProduct).forEach(key => {  
 if (key === 'image' || key === 'boxImage') {  
 if (editingProduct[key]) {  
 formData.append(key, editingProduct[key]);  
 }  
 } else if (key !== '\_id') {  
 formData.append(key, editingProduct[key]);  
 }  
 });  
   
 // Mock update for development  
 // const updatedProducts = products.map(product =>   
 // product.\_id === editingProduct.\_id ?   
 // {...product, ...editingProduct, image: product.image, boxImage: product.boxImage} :   
 // product  
 // );  
 await productService.updateProduct(editingProduct.\_id, formData);  
  
   
 // Get fresh data after update  
 const updatedProducts = await productService.getAllProducts();  
 setProducts(Array.isArray(updatedProducts) ? updatedProducts : []);  
   
 setEditingProduct(null);  
 setShowAddForm(false);  
 setError(null);  
 } catch (err) {  
 console.error('Error updating product:', err);  
 setError('Failed to update product. Please try again.');  
 }  
 };  
   
  
  
 const handleDeleteProduct = async (id) => {  
 if (window.confirm('Are you sure you want to delete this product?')) {  
 try {  
  
 // // Mock delete for development  
 // setProducts(products.filter(product => product.\_id !== id));  
 // Use service instead of direct axios  
 await productService.deleteProduct(id);  
   
 // Update state safely  
 setProducts(prevProducts =>   
 Array.isArray(prevProducts) ? prevProducts.filter(product => product.\_id !== id) : []  
 );  
 setError(null);  
 } catch (err) {  
 console.error('Error deleting product:', err);  
 setError('Failed to delete product. Please try again.');  
 }  
 }  
 };  
  
  
 if (loading) return <div>Loading products...</div>;  
  
 return (  
 <AdminLayout>  
 <AdminSidebar />  
 <MainContent>  
 <PageHeader>  
 <div>  
 <h1>Products Management</h1>  
 <p>Manage your store products</p>  
 </div>  
 <AddButton onClick={() => {  
 setEditingProduct(null);  
 setShowAddForm(true);  
 }}>  
 <FaPlus />  
 <span>Add Product</span>  
 </AddButton>  
 </PageHeader>  
   
 <SearchBar>  
 <SearchIcon>  
 <FaSearch />  
 </SearchIcon>  
 <SearchInput   
 type="text"   
 placeholder="Search products..."  
 value={searchQuery}  
 onChange={(e) => setSearchQuery(e.target.value)}  
 />  
 </SearchBar>  
   
 {showAddForm && (  
 <FormOverlay>  
 <ProductForm onSubmit={editingProduct ? handleUpdateProduct : handleAddProduct}>  
 <FormHeader>  
 <h2>{editingProduct ? 'Edit Product' : 'Add New Product'}</h2>  
 <CloseButton onClick={() => {  
 setShowAddForm(false);  
 setEditingProduct(null);  
 }}>×</CloseButton>  
 </FormHeader>  
   
 <FormGrid>  
 <FormGroup>  
 <label>Product Name (English)</label>  
 <input   
 type="text"   
 name="name"   
 value={editingProduct ? editingProduct.name : newProduct.name}   
 onChange={editingProduct ?   
 (e) => setEditingProduct({...editingProduct, name: e.target.value}) :   
 handleInputChange  
 }  
 required   
 />  
 </FormGroup>  
   
 <FormGroup>  
 <label>Product Name (Arabic)</label>  
 <input   
 type="text"   
 name="nameAr"   
 value={editingProduct ? editingProduct.nameAr : newProduct.nameAr}   
 onChange={editingProduct ?   
 (e) => setEditingProduct({...editingProduct, nameAr: e.target.value}) :   
 handleInputChange  
 }  
 required   
 />  
 </FormGroup>  
   
 <FormGroup>  
 <label>Price (SAR)</label>  
 <input   
 type="number"   
 name="price"   
 value={editingProduct ? editingProduct.price : newProduct.price}   
 onChange={editingProduct ?   
 (e) => setEditingProduct({...editingProduct, price: e.target.value}) :   
 handleInputChange  
 }  
 required   
 />  
 </FormGroup>  
   
 <FormGroup>  
 <label>Category</label>  
 <select   
 name="category"   
 value={editingProduct ? editingProduct.category : newProduct.category}   
 onChange={editingProduct ?   
 (e) => setEditingProduct({...editingProduct, category: e.target.value}) :   
 handleInputChange  
 }  
 required  
 >  
 <option value="">Select Category</option>  
 <option value="spray">All Over Spray</option>  
 <option value="perfume">Perfume</option>  
 <option value="candle">Candle</option>  
 <option value="gift">Gift Set</option>  
 </select>  
 </FormGroup>  
   
 <FormGroup fullWidth>  
 <label>Description (English)</label>  
 <textarea   
 name="description"   
 value={editingProduct ? editingProduct.description : newProduct.description}   
 onChange={editingProduct ?   
 (e) => setEditingProduct({...editingProduct, description: e.target.value}) :   
 handleInputChange  
 }  
 rows="3"  
 />  
 </FormGroup>  
   
 <FormGroup fullWidth>  
 <label>Description (Arabic)</label>  
 <textarea   
 name="descriptionAr"   
 value={editingProduct ? editingProduct.descriptionAr : newProduct.descriptionAr}   
 onChange={editingProduct ?   
 (e) => setEditingProduct({...editingProduct, descriptionAr: e.target.value}) :   
 handleInputChange  
 }  
 rows="3"  
 />  
 </FormGroup>  
   
 <FormGroup>  
 <label>Product Image</label>  
 <input   
 type="file"   
 name="image"   
 onChange={editingProduct ?   
 (e) => setEditingProduct({...editingProduct, image: e.target.files[0]}) :   
 handleFileChange  
 }   
 accept="image/\*"  
 required={!editingProduct}  
 />  
 {editingProduct && editingProduct.image && (  
 <ImagePreview>Current: {editingProduct.image}</ImagePreview>  
 )}  
 </FormGroup>  
   
 <FormGroup>  
 <label>Box Image</label>  
 <input   
 type="file"   
 name="boxImage"   
 onChange={editingProduct ?   
 (e) => setEditingProduct({...editingProduct, boxImage: e.target.files[0]}) :   
 handleFileChange  
 }   
 accept="image/\*"  
 required={!editingProduct}  
 />  
 {editingProduct && editingProduct.boxImage && (  
 <ImagePreview>Current: {editingProduct.boxImage}</ImagePreview>  
 )}  
 </FormGroup>  
 </FormGrid>  
   
 <FormActions>  
 <CancelButton type="button" onClick={() => {  
 setShowAddForm(false);  
 setEditingProduct(null);  
 }}>  
 Cancel  
 </CancelButton>  
 <SubmitButton type="submit">  
 {editingProduct ? 'Update Product' : 'Add Product'}  
 </SubmitButton>  
 </FormActions>  
 </ProductForm>  
 </FormOverlay>  
 )}  
   
 <ProductsTable>  
 <thead>  
 <tr>  
 <th>Image</th>  
 <th>Name</th>  
 <th>Arabic Name</th>  
 <th>Price</th>  
 <th>Category</th>  
 <th>Stock</th>  
 <th>Actions</th>  
 </tr>  
 </thead>  
 <tbody>  
 {filteredProducts.map((product) => (  
 <tr key={product.\_id}>  
 <td>  
 <ProductImage src={product.image} alt={product.name} />  
 </td>  
 <td>{product.name}</td>  
 <td>{product.nameAr}</td>  
 <td>{product.currency} {product.price.toFixed(2)}</td>  
 <td>  
 <CategoryBadge category={product.category}>  
 {product.category.charAt(0).toUpperCase() + product.category.slice(1)}  
 </CategoryBadge>  
 </td>  
 <td>{product.stock}</td>  
 <td>  
 <ActionButtons>  
 <EditButton onClick={() => handleEditProduct(product)}>  
 <FaEdit />  
 </EditButton>  
 <DeleteButton onClick={() => handleDeleteProduct(product.\_id)}>  
 <FaTrash />  
 </DeleteButton>  
 </ActionButtons>  
 </td>  
 </tr>  
 ))}  
 </tbody>  
 </ProductsTable>  
 </MainContent>  
 </AdminLayout>  
 );  
};  
  
// Styled Components  
const AdminLayout = styled.div`  
 display: flex;  
 min-height: 100vh;  
`;  
  
const MainContent = styled.main`  
 flex: 1;  
 padding: 20px;  
 background-color: #f5f7fa;  
`;  
  
const PageHeader = styled.div`  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 margin-bottom: 30px;  
   
 h1 {  
 font-size: 24px;  
 color: var(--primary-color);  
 margin-bottom: 5px;  
 }  
   
 p {  
 color: #777;  
 font-size: 14px;  
 }  
`;  
  
const AddButton = styled.button`  
 display: flex;  
 align-items: center;  
 background-color: var(--secondary-color);  
 color: var(--primary-color);  
 border: none;  
 padding: 10px 16px;  
 border-radius: 4px;  
 font-weight: 600;  
 cursor: pointer;  
   
 span {  
 margin-left: 8px;  
 }  
   
 &:hover {  
 background-color: var(--accent-color);  
 }  
`;  
  
const SearchBar = styled.div`  
 position: relative;  
 margin-bottom: 20px;  
`;  
  
const SearchIcon = styled.span`  
 position: absolute;  
 left: 15px;  
 top: 50%;  
 transform: translateY(-50%);  
 color: #777;  
`;  
  
const SearchInput = styled.input`  
 width: 100%;  
 padding: 12px 12px 12px 40px;  
 border: 1px solid #ddd;  
 border-radius: 4px;  
 font-size: 14px;  
   
 &:focus {  
 outline: none;  
 border-color: var(--secondary-color);  
 }  
`;  
  
const ProductsTable = styled.table`  
 width: 100%;  
 border-collapse: collapse;  
 background-color: white;  
 border-radius: 8px;  
 overflow: hidden;  
 box-shadow: 0 2px 5px rgba(0, 0, 0, 0.05);  
   
 th, td {  
 padding: 15px;  
 text-align: left;  
 border-bottom: 1px solid #eee;  
 }  
   
 th {  
 background-color: #f9f9f9;  
 font-weight: 600;  
 color: var(--primary-color);  
 }  
   
 tbody tr:hover {  
 background-color: #f9f9f9;  
 }  
`;  
  
const ProductImage = styled.img`  
 width: 50px;  
 height: 50px;  
 object-fit: contain;  
 border-radius: 4px;  
`;  
  
const CategoryBadge = styled.span`  
 display: inline-block;  
 padding: 4px 10px;  
 border-radius: 20px;  
 font-size: 12px;  
 font-weight: 500;  
   
 ${props => {  
 switch (props.category) {  
 case 'spray':  
 return `  
 background-color: #e6f7e9;  
 color: #2ecc71;  
 `;  
 case 'perfume':  
 return `  
 background-color: #e6f0f9;  
 color: #3498db;  
 `;  
 case 'candle':  
 return `  
 background-color: #fef4e6;  
 color: #f39c12;  
 `;  
 case 'gift':  
 return `  
 background-color: #f0e6f9;  
 color: #9b59b6;  
 `;  
 default:  
 return `  
 background-color: #f5f5f5;  
 color: #777;  
 `;  
 }  
 }}  
`;  
  
const ActionButtons = styled.div`  
 display: flex;  
 gap: 10px;  
`;  
  
const EditButton = styled.button`  
 background-color: #e6f0f9;  
 color: #3498db;  
 border: none;  
 width: 32px;  
 height: 32px;  
 border-radius: 4px;  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 cursor: pointer;  
   
 &:hover {  
 background-color: #3498db;  
 color: white;  
 }  
`;  
  
const DeleteButton = styled.button`  
 background-color: #fde9e9;  
 color: #e74c3c;  
 border: none;  
 width: 32px;  
 height: 32px;  
 border-radius: 4px;  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 cursor: pointer;  
   
 &:hover {  
 background-color: #e74c3c;  
 color: white;  
 }  
`;  
  
const FormOverlay = styled.div`  
 position: fixed;  
 top: 0;  
 left: 0;  
 right: 0;  
 bottom: 0;  
 background-color: rgba(0, 0, 0, 0.5);  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 z-index: 1000;  
`;  
  
const ProductForm = styled.form`  
 background-color: white;  
 border-radius: 8px;  
 width: 90%;  
 max-width: 800px;  
 max-height: 90vh;  
 overflow-y: auto;  
 box-shadow: 0 4px 15px rgba(0, 0, 0, 0.15);  
`;  
  
const FormHeader = styled.div`  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 padding: 20px;  
 border-bottom: 1px solid #eee;  
   
 h2 {  
 font-size: 20px;  
 color: var(--primary-color);  
 margin: 0;  
 }  
`;  
  
const CloseButton = styled.button`  
 background: none;  
 border: none;  
 font-size: 24px;  
 cursor: pointer;  
 color: #777;  
   
 &:hover {  
 color: #e74c3c;  
 }  
`;  
  
const FormGrid = styled.div`  
 display: grid;  
 grid-template-columns: repeat(2, 1fr);  
 gap: 20px;  
 padding: 20px;  
   
 @media (max-width: 768px) {  
 grid-template-columns: 1fr;  
 }  
`;  
  
const FormGroup = styled.div`  
 grid-column: ${props => props.fullWidth ? 'span 2' : 'span 1'};  
   
 label {  
 display: block;  
 font-size: 14px;  
 font-weight: 600;  
 margin-bottom: 8px;  
 color: var(--primary-color);  
 }  
   
 input, textarea, select {  
 width: 100%;  
 padding: 10px;  
 border: 1px solid #ddd;  
 border-radius: 4px;  
 font-size: 14px;  
   
 &:focus {  
 outline: none;  
 border-color: var(--secondary-color);  
 }  
 }  
   
 @media (max-width: 768px) {  
 grid-column: span 1;  
 }  
`;  
  
const ImagePreview = styled.div`  
 font-size: 12px;  
 color: #777;  
 margin-top: 5px;  
`;  
  
const FormActions = styled.div`  
 display: flex;  
 justify-content: flex-end;  
 padding: 20px;  
 border-top: 1px solid #eee;  
 gap: 10px;  
`;  
  
const CancelButton = styled.button`  
 background-color: #f5f5f5;  
 color: #333;  
 border: none;  
 padding: 10px 16px;  
 border-radius: 4px;  
 font-weight: 600;  
 cursor: pointer;  
   
 &:hover {  
 background-color: #e0e0e0;  
 }  
`;  
  
const SubmitButton = styled.button`  
 background-color: var(--secondary-color);  
 color: var(--primary-color);  
 border: none;  
 padding: 10px 16px;  
 border-radius: 4px;  
 font-weight: 600;  
 cursor: pointer;  
   
 &:hover {  
 background-color: var(--accent-color);  
 }  
`;  
  
export default AdminProducts;

File Path: .\src\services\adminService.js

Content:

// src/services/adminService.js  
import api from '../api/config';  
  
export const adminService = {  
 // Dashboard stats  
 getDashboardStats: async () => {  
 try {  
 const response = await api.get('/api/admin/dashboard');  
 return response.data;  
 } catch (error) {  
 console.error('Error getting dashboard stats:', error);  
 throw error;  
 }  
 },  
   
 // Get all orders with pagination and filtering  
 getAllOrders: async (params = {}) => {  
 try {  
 const response = await api.get('/api/admin/orders', { params });  
 return response.data;  
 } catch (error) {  
 console.error('Error getting all orders:', error);  
 throw error;  
 }  
 },  
   
 // Update order status  
 updateOrderStatus: async (orderId, status) => {  
 try {  
 const response = await api.put(`/api/admin/orders/${orderId}/status`, { status });  
 return response.data;  
 } catch (error) {  
 console.error(`Error updating order ${orderId} status:`, error);  
 throw error;  
 }  
 },  
   
 // Create product  
 createProduct: async (formData) => {  
 try {  
 const response = await api.post('/api/products', formData, {  
 headers: {  
 'Content-Type': 'multipart/form-data'  
 }  
 });  
 return response.data;  
 } catch (error) {  
 console.error('Error creating product:', error);  
 throw error;  
 }  
 },  
   
 // Update product  
 updateProduct: async (productId, formData) => {  
 try {  
 const response = await api.put(`/api/products/${productId}`, formData, {  
 headers: {  
 'Content-Type': 'multipart/form-data'  
 }  
 });  
 return response.data;  
 } catch (error) {  
 console.error(`Error updating product ${productId}:`, error);  
 throw error;  
 }  
 },  
   
 // Delete product  
 deleteProduct: async (productId) => {  
 try {  
 const response = await api.delete(`/api/products/${productId}`);  
 return response.data;  
 } catch (error) {  
 console.error(`Error deleting product ${productId}:`, error);  
 throw error;  
 }  
 },  
   
 // Get all customers  
 getAllCustomers: async (params = {}) => {  
 try {  
 const response = await api.get('/api/admin/customers', { params });  
 return response.data;  
 } catch (error) {  
 console.error('Error getting customers:', error);  
 throw error;  
 }  
 }  
};

File Path: .\src\services\authService.js

Content:

// src/services/authService.js  
import api from '../api/config';  
  
export const authService = {  
 // Login user  
 login: async (email, password) => {  
 try {  
 const response = await api.post('/api/auth/login', { email, password });  
 return response.data;  
 } catch (error) {  
 console.error('Login error:', error);  
 throw error;  
 }  
 },  
   
 // Register user  
 register: async (userData) => {  
 try {  
 const response = await api.post('/api/auth/register', userData);  
 return response.data;  
 } catch (error) {  
 console.error('Registration error:', error);  
 throw error;  
 }  
 },  
   
 // Get current user  
 getCurrentUser: async () => {  
 try {  
 const response = await api.get('/api/auth/me');  
 return response.data;  
 } catch (error) {  
 console.error('Error getting current user:', error);  
 throw error;  
 }  
 },  
   
 // Update user profile  
 updateProfile: async (userData) => {  
 try {  
 const response = await api.put('/api/auth/profile', userData);  
 return response.data;  
 } catch (error) {  
 console.error('Error updating profile:', error);  
 throw error;  
 }  
 },  
   
 // Logout  
 logout: async () => {  
 try {  
 const response = await api.post('/api/auth/logout');  
 return response.data;  
 } catch (error) {  
 console.error('Logout error:', error);  
 // Still clear local storage even if server logout fails  
 localStorage.removeItem('token');  
 }  
 }  
};

File Path: .\src\services\index.js

Content:

// src/services/index.js  
export { productService } from './productService';  
export { authService } from './authService';  
export { orderService } from './orderService';  
export { adminService } from './adminService';

File Path: .\src\services\orderService.js

Content:

// src/services/orderService.js  
import api from '../api/config';  
  
export const orderService = {  
 // Create new order  
 createOrder: async (orderData) => {  
 try {  
 const response = await api.post('/api/orders', orderData);  
 return response.data;  
 } catch (error) {  
 console.error('Error creating order:', error);  
 throw error;  
 }  
 },  
   
 // Get user's orders  
 getUserOrders: async () => {  
 try {  
 const response = await api.get('/api/orders');  
 return response.data;  
 } catch (error) {  
 console.error('Error getting user orders:', error);  
 throw error;  
 }  
 },  
   
 // Get single order details  
 getOrderById: async (orderId) => {  
 try {  
 const response = await api.get(`/api/orders/${orderId}`);  
 return response.data;  
 } catch (error) {  
 console.error(`Error getting order ${orderId}:`, error);  
 throw error;  
 }  
 },  
   
 // Cancel order  
 cancelOrder: async (orderId) => {  
 try {  
 const response = await api.put(`/api/orders/${orderId}/cancel`);  
 return response.data;  
 } catch (error) {  
 console.error(`Error cancelling order ${orderId}:`, error);  
 throw error;  
 }  
 }  
};

File Path: .\src\services\productService.js

Content:

// src/services/productService.js  
import api from '../api/config';  
  
export const productService = {  
 // Get all products with optional filtering  
 getAllProducts: async (params = {}) => {  
 try {  
 const response = await api.get('/api/products', { params });  
 return response.data;  
 } catch (error) {  
 console.error('Error getting products:', error);  
 throw error;  
 }  
 },  
   
 // Get a single product by ID  
 getProductById: async (id) => {  
 try {  
 const response = await api.get(`/api/products/${id}`);  
 return response.data;  
 } catch (error) {  
 console.error(`Error getting product ${id}:`, error);  
 throw error;  
 }  
 },  
   
 // Get featured products  
 getFeaturedProducts: async () => {  
 try {  
 const response = await api.get('/api/products', { params: { featured: true } });  
 return response.data;  
 } catch (error) {  
 console.error('Error getting featured products:', error);  
 throw error;  
 }  
 },  
   
 // Search products  
 searchProducts: async (query) => {  
 try {  
 const response = await api.get('/api/products', { params: { search: query } });  
 return response.data;  
 } catch (error) {  
 console.error(`Error searching products with query "${query}":`, error);  
 throw error;  
 }  
 },  
   
 // Filter products by category  
 getProductsByCategory: async (category) => {  
 try {  
 const response = await api.get('/api/products', { params: { category } });  
 return response.data;  
 } catch (error) {  
 console.error(`Error getting products for category "${category}":`, error);  
 throw error;  
 }  
 }  
};