

```

import React, { useState, createContext, useContext, useEffect } from
'react';
import { Home, Phone, Settings, ShoppingCart, Sun, Moon, X, Minus,
Plus } from 'lucide-react';

// 1. Theme Context
const ThemeContext = createContext();

// 2. Cart Context
const CartContext = createContext();

// Mock Product Data - شما باید اطلاعات محصولات خود را ویرایش
کنید
const products = [
  {
    id: 1, // شناسه منحصر به فرد محصول
    name: 'پیراهن مجلسی زنانه', // نام محصول
    price: 350000, // قیمت محصول به تومان (فقط عدد را وارد کنید)
    images: [ // آرایه‌ای از آدرس‌های عکس‌های محصول
      'https://placeholder.co/600x400/9333ea/ffffff?text=, پیراهن+مجلسی+جلو',
      'https://placeholder.co/600x400/9333ea/ffffff?text=, پیراهن+مجلسی+پشت',
      'https://placeholder.co/600x400/9333ea/ffffff?text=, عکس پشت لباس',
      'https://placeholder.co/600x400/9333ea/ffffff?text=, عکس جزئیات لباس',
    ],
    description: 'یک پیراهن مجلسی زیبا و شیک برای مهمانی‌ها و مراسم خاص. طراحی
    منحصر به فرد با پارچه‌ای لطیف و با کیفیت. // توضیحات محصول',
  },
  {
    id: 2,
    name: 'تشرت مردانه اسپرت',
    price: 120000,
    images: [
      'https://placeholder.co/600x400/8b5cf6/ffffff?text=, تشرت+اسپرت+جلو',
      'https://placeholder.co/600x400/8b5cf6/ffffff?text=, تشرت+اسپرت+پشت',
    ],
    description: 'تشرت اسپرت و راحت برای استفاده روزمره. ساخته شده از نخ پنبه با
    دوام و تنفس‌پذیر.',
  },
  {
    id: 3,
    name: 'شلوار جین زنانه',
    price: 280000,
    images: [
      'https://placeholder.co/600x400/a78bfa/ffffff?text=, شلوار+جین+جلو',

```

```

        'https://placeholder.co/600x400/a78bfa/ffffff?text=شلوار+چین+پشت',
    ],
    description: 'شلوار چین زنانه با برش راسته و فیت عالی. مناسب برای استایل کژوال'
},
{
    id: 4,
    name: 'کاپشن چرم مردانه',
    price: 950000,
    images: [
        'https://placeholder.co/600x400/6d28d9/ffffff?text=کاپشن+چرم+جلو',
        'https://placeholder.co/600x400/6d28d9/ffffff?text=کاپشن+چرم+پشت'
    ],
    description: 'کاپشن چرم با کیفیت بالا و طراحی کلاسیک. ایده آل برای فصول سرد'
},
{
    id: 5,
    name: 'ماتو تابستانه',
    price: 420000,
    images: [
        'https://placeholder.co/600x400/c084fc/ffffff?text=ماتو+تابستانه+جلو',
        'https://placeholder.co/600x400/c084fc/ffffff?text=ماتو+تابستانه+پشت'
    ],
    description: 'ماتو خنک و سبک مناسب فصل تابستان. طراحی شیک و راحت'
},
{
    id: 6,
    name: 'کت و شلوار رسمی',
    price: 1500000,
    images: [
        'https://placeholder.co/600x400/7c3aed/ffffff?text=کت+و+شلوار+جلو',
        'https://placeholder.co/600x400/7c3aed/ffffff?text=کت+و+شلوار+پشت'
    ],
    description: 'کت و شلوار رسمی مردانه با برش اسلیم فیت. مناسب برای مراسم و جلسات'
},
{
    // شما می توانید محصولات جدیدی را در اینجا اضافه کنید
    /*
    {
        id: 7, // شناسه منحصر به فرد جدید
        name: 'نام محصول جدید',
        price: 100000,
        images: [
            'https://example.com/new_product_front.jpg', // آدرس عکس واقعی
            'https://example.com/new_product_back.jpg'
        ],
        description: 'توضیحات محصول جدید شما'
    }
    */
}

```

```

    },
    */
];

// Header Component
const Header = ({ navigate }) => {
  const { theme, toggleTheme } = useContext(ThemeContext);
  const { cartItems } = useContext(CartContext);
  const totalCartItems = cartItems.reduce((acc, item) => acc +
item.quantity, 0);

  return (
    <header className={`bg-purple-700 text-white p-4 shadow-lg sticky
top-0 z-50`} >
      <div className="container mx-auto flex justify-between
items-center flex-wrap">
        <h1 className="text-3xl font-bold font-inter cursor-pointer"
onClick={() => navigate('home')} >
          AYLAR_SHOP
        </h1>
        <nav className="flex items-center space-x-4 mt-2 sm:mt-0">
          <button
            onClick={() => navigate('home')}
            className="flex items-center px-3 py-2 rounded-lg
hover:bg-purple-600 transition-colors duration-300"
          >
            <Home className="w-5 h-5 mr-1" />
            <span>صفحه اصلی</span>
          </button>
          <button
            onClick={() => navigate('contact')}
            className="flex items-center px-3 py-2 rounded-lg
hover:bg-purple-600 transition-colors duration-300"
          >
            <Phone className="w-5 h-5 mr-1" />
            <span>تماس با ما</span>
          </button>
          <button
            onClick={() => navigate('settings')}
            className="flex items-center px-3 py-2 rounded-lg
hover:bg-purple-600 transition-colors duration-300"
          >
            <Settings className="w-5 h-5 mr-1" />
            <span>تنظیمات</span>
          </button>
          <button
            onClick={() => navigate('cart')}
            className="relative flex items-center px-3 py-2 rounded-lg

```

```

hover:bg-purple-600 transition-colors duration-300"
    >
      <ShoppingCart className="w-5 h-5 mr-1" />
      <span>سبد خرید</span>
      {totalCartItem > 0 && (
        <span className="absolute -top-1 -right-1 bg-red-500
text-white text-xs font-bold rounded-full h-5 w-5 flex items-center
justify-center">
          {totalCartItem}
        </span>
      )}
    </button>
    <button
      onClick={toggleTheme}
      className="p-2 rounded-full bg-purple-600
hover:bg-purple-500 transition-colors duration-300"
      aria-label="Toggle theme"
    >
      {theme === 'light' ? (
        <Moon className="w-5 h-5 text-white" />
      ) : (
        <Sun className="w-5 h-5 text-yellow-300" />
      )}
    </button>
  </nav>
</div>
</header>
);
};

// Footer Component
const Footer = () => {
  const { theme } = useContext(ThemeContext);
  return (
    <footer className={`p-4 mt-8 shadow-inner ${theme === 'dark' ?
'bg-gray-800 text-gray-300' : 'bg-gray-200 text-gray-700'}`}>
      <div className="container mx-auto text-center">
        <p>© 2024 AYLAR_SHOP. تمامی حقوق محفوظ است.</p>
      </div>
    </footer>
  );
};

// Product Card Component with Image Slider
const ProductCard = ({ product }) => {
  const { addToCart } = useContext(CartContext);
  const [currentImageIndex, setCurrentImageIndex] = useState(0);

```

```

const nextImage = () => {
  setCurrentImageIndex((prevIndex) => (prevIndex + 1) %
product.images.length);
};

const prevImage = () => {
  setCurrentImageIndex((prevIndex) => (prevIndex - 1 +
product.images.length) % product.images.length);
};

return (
  <div className={`bg-white dark:bg-gray-800 rounded-xl shadow-lg
overflow-hidden transform transition-transform duration-300
hover:scale-105`} >
    <div className="relative h-64 overflow-hidden">
      <img
        src={product.images[currentImageIndex]}
        alt={product.name}
        className="w-full h-full object-cover transition-opacity
duration-300"
        onError={(e) => { e.target.onerror = null; e.target.src =
`https://placeholder.co/600x400/cccccc/333333?text=تصویر ناموجود`; }}
      />
      {product.images.length > 1 && (
        <>
          <button
            onClick={prevImage}
            className="absolute left-2 top-1/2 -translate-y-1/2
bg-black bg-opacity-50 text-white p-2 rounded-full hover:bg-opacity-75
transition-opacity duration-300"
            aria-label="Previous image"
          >
            <
          </button>
          <button
            onClick={nextImage}
            className="absolute right-2 top-1/2 -translate-y-1/2
bg-black bg-opacity-50 text-white p-2 rounded-full hover:bg-opacity-75
transition-opacity duration-300"
            aria-label="Next image"
          >
            >
          </button>
        </>
      )}
    </div>
    <div className="p-4 flex flex-col items-center text-center">
      <h3 className="text-xl font-semibold text-gray-800

```

```

dark:text-gray-100 mb-2">{product.name}</h3>
    <p className="text-purple-600 dark:text-purple-400 text-2xl
font-bold mb-4">
        {product.price.toLocaleString('fa-IR')} تومان
    </p>
    <button
        onClick={() => addToCart(product)}
        className="bg-purple-600 text-white px-6 py-3 rounded-lg
font-semibold hover:bg-purple-700 transition-colors duration-300
shadow-md transform hover:scale-105"
    >
        افزودن به سبد خرید
    </button>
</div>
</div>
);
};

// Home Page Component
const HomePage = () => {
    return (
        <div className="py-8">
            <h2 className="text-4xl font-extrabold text-center
text-purple-800 dark:text-purple-300 mb-10 drop-shadow-md">
                جدیدترین محصولات
            </h2>
            <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3
xl:grid-cols-4 gap-8">
                {products.map((product) => (
                    <ProductCard key={product.id} product={product} />
                ))}
            </div>
        </div>
    );
};

// Contact Us Page Component
const ContactUsPage = () => {
    const { theme } = useContext(ThemeContext);
    return (
        <div className={`p-8 rounded-xl shadow-lg max-w-2xl mx-auto
${theme === 'dark' ? 'bg-gray-800 text-gray-100' : 'bg-white
text-gray-900'}`>
            <h2 className="text-3xl font-bold text-center text-purple-700
dark:text-purple-300 mb-6">ما تماس با ما</h2>
            <p className="text-center mb-8">
                برای هرگونه سوال یا پشتیبانی، لطفاً از طریق فرم زیر با ما در تماس باشید.
            </p>

```

```

<form className="space-y-6">
  <div>
    <label htmlFor="name" className="block text-sm font-medium
mb-2">نام و نام خانوادگی:</label>
    <input
      type="text"
      id="name"
      className={`w-full p-3 rounded-lg border focus:ring-2
focus:ring-purple-500 outline-none ${theme === 'dark' ? 'bg-gray-700
border-gray-600 text-gray-100' : 'bg-gray-50 border-gray-300
text-gray-900'}`}
      placeholder="نام کامل شما"
    />
  </div>
  <div>
    <label htmlFor="email" className="block text-sm font-medium
mb-2">ایمیل:</label>
    <input
      type="email"
      id="email"
      className={`w-full p-3 rounded-lg border focus:ring-2
focus:ring-purple-500 outline-none ${theme === 'dark' ? 'bg-gray-700
border-gray-600 text-gray-100' : 'bg-gray-50 border-gray-300
text-gray-900'}`}
      placeholder="ایمیل شما"
    />
  </div>
  <div>
    <label htmlFor="message" className="block text-sm
font-medium mb-2">پیام شما:</label>
    <textarea
      id="message"
      rows="5"
      className={`w-full p-3 rounded-lg border focus:ring-2
focus:ring-purple-500 outline-none resize-y ${theme === 'dark' ?
'bg-gray-700 border-gray-600 text-gray-100' : 'bg-gray-50
border-gray-300 text-gray-900'}`}
      placeholder="پیام خود را اینجا بنویسید..."
    ></textarea>
  </div>
  <button
    type="submit"
    className="w-full bg-purple-600 text-white px-6 py-3
rounded-lg font-semibold hover:bg-purple-700 transition-colors
duration-300 shadow-md transform hover:scale-105"
  >
    ارسال پیام
  </button>

```

```

        </form>
      </div>
    );
  };

// Settings Page Component
const SettingsPage = () => {
  const { theme, toggleTheme } = useContext(ThemeContext);
  return (
    <div className={`p-8 rounded-xl shadow-lg max-w-xl mx-auto ${theme
    === 'dark' ? 'bg-gray-800 text-gray-100' : 'bg-white
    text-gray-900'}`}>
      <h2 className="text-3xl font-bold text-center text-purple-700
      dark:text-purple-300 mb-6">تنظیمات</h2>
      <div className="flex items-center justify-between p-4 border-b
      border-gray-200 dark:border-gray-700 last:border-b-0">
        <span className="text-lg font-medium">تم سایت:</span>
        <button
          onClick={toggleTheme}
          className="flex items-center px-4 py-2 rounded-full
          bg-purple-600 text-white hover:bg-purple-700 transition-colors
          duration-300 shadow-md"
        >
          {theme === 'light' ? (
            <>
              <Moon className="w-5 h-5 mr-2" />
              <span>حالت شب</span>
            </>
          ) : (
            <>
              <Sun className="w-5 h-5 mr-2 text-yellow-300" />
              <span>حالت روز</span>
            </>
          )}
        </button>
      </div>
    </div>
  );
};

```

```

// Cart Page Component
const CartPage = () => {
  const { cartItems, removeFromCart, updateQuantity } =
    useContext(CartContext);
  const { theme } = useContext(ThemeContext);

  const totalPrice = cartItems.reduce((acc, item) => acc + item.price
  * item.quantity, 0);

```



```

return (
  <div className={`p-8 rounded-xl shadow-lg max-w-4xl mx-auto
    ${theme === 'dark' ? 'bg-gray-800 text-gray-100' : 'bg-white
    text-gray-900'}`}>
    <h2 className="text-3xl font-bold text-center text-purple-700
    dark:text-purple-300 mb-8">سبد خرید شما</h2>

    {cartItems.length === 0 ? (
      <p className="text-center text-lg text-gray-600
    dark:text-gray-400">سبد خرید شما خالی است.</p>
    ) : (
      <>
        <div className="space-y-6">
          {cartItems.map((item) => (
            <div
              key={item.id}
              className={`flex flex-col sm:flex-row items-center
              justify-between p-4 rounded-lg shadow-sm ${theme === 'dark' ?
              'bg-gray-700' : 'bg-gray-100'}`}
            >
              <div className="flex items-center w-full sm:w-auto
              mb-4 sm:mb-0">
                <img
                  src={item.images[0]}
                  alt={item.name}
                  className="w-20 h-20 object-cover rounded-lg mr-4"
                  onError={(e) => { e.target.onerror = null;
e.target.src = `https://placeholder.co/80x80/cccccc/333333?text=تصویر`; }}
                />
                <div>
                  <h3 className="text-lg
font-semibold">{item.name}</h3>
                  <p className="text-purple-600 dark:text-purple-400
font-bold">
                    {(item.price).toLocaleString('fa-IR')} تومان
                  </p>
                </div>
              </div>
              <div className="flex items-center space-x-3 w-full
              sm:w-auto justify-end">
                <button
                  onClick={() => updateQuantity(item.id,
item.quantity - 1)}
                  className="p-2 rounded-full bg-purple-500
text-white hover:bg-purple-600 transition-colors duration-300"
                  disabled={item.quantity <= 1}
                  aria-label="Decrease quantity"

```

```

        >
        <Minus className="w-4 h-4" />
      </button>
      <span className="font-bold
text-xl">{item.quantity}</span>
      <button
        onClick={() => updateQuantity(item.id,
item.quantity + 1)}
        className="p-2 rounded-full bg-purple-500
text-white hover:bg-purple-600 transition-colors duration-300"
        aria-label="Increase quantity"
      >
        <Plus className="w-4 h-4" />
      </button>
      <button
        onClick={() => removeFromCart(item.id)}
        className="p-2 rounded-full bg-red-500 text-white
hover:bg-red-600 transition-colors duration-300 ml-4"
        aria-label="Remove item"
      >
        <X className="w-4 h-4" />
      </button>
    </div>
  </div>
  )))}
</div>
  <div className={`mt-8 p-6 rounded-lg shadow-md flex
justify-between items-center ${theme === 'dark' ? 'bg-gray-700' :
'bg-gray-100'}`}>
    <span className="text-xl font-bold">مجموع کل</span>
    <span className="text-3xl font-extrabold text-purple-700
dark:text-purple-300">
      {totalPrice.toLocaleString('fa-IR')} تومان
    </span>
  </div>
  <div className="mt-8 text-center">
    <button className="bg-green-600 text-white px-8 py-4
rounded-lg font-semibold text-xl hover:bg-green-700 transition-colors
duration-300 shadow-lg transform hover:scale-105">
      ادامه و پرداخت
    </button>
  </div>
</>
  )}
</div>
);
};

```

```

// Main App Component
const App = () => {
  const [currentPage, setCurrentPage] = useState('home');
  const [theme, setTheme] = useState('light'); // 'light' or 'dark'
  const [cartItems, setCartItems] = useState([]);

  // Function to toggle theme
  const toggleTheme = () => {
    setTheme((prevTheme) => (prevTheme === 'light' ? 'dark' :
'light'));
  };

  // Function to navigate between pages
  const navigate = (page) => {
    setCurrentPage(page);
  };

  // Function to add product to cart
  const addToCart = (product) => {
    setCartItems((prevItems) => {
      const existingItem = prevItems.find((item) => item.id ===
product.id);
      if (existingItem) {
        return prevItems.map((item) =>
          item.id === product.id ? { ...item, quantity: item.quantity
+ 1 } : item
        );
      } else {
        return [...prevItems, { ...product, quantity: 1 }];
      }
    });
  };

  // Function to remove product from cart
  const removeFromCart = (productId) => {
    setCartItems((prevItems) => prevItems.filter((item) => item.id !==
productId));
  };

  // Function to update quantity of product in cart
  const updateQuantity = (productId, newQuantity) => {
    if (newQuantity <= 0) {
      removeFromCart(productId);
      return;
    }
    setCartItems((prevItems) =>
      prevItems.map((item) =>

```

```

        item.id === productId ? { ...item, quantity: newQuantity } :
item
    )
    );
};

// Effect to apply theme classes to the document's root element
(html tag)
useEffect(() => {
    document.documentElement.classList.remove('light', 'dark');
    document.documentElement.classList.add(theme);
    // Set background color for the body based on theme
    document.body.style.backgroundColor = theme === 'dark' ? '#111827'
: '#f3f4f6';
    document.body.style.color = theme === 'dark' ? '#e5e7eb' :
'#1f2937';
    document.body.style.fontFamily = 'Inter, sans-serif'; // Apply
Inter font
}, [theme]);

// Render the current page based on currentPage state
const renderPage = () => {
    switch (currentPage) {
        case 'home':
            return <HomePage />;
        case 'contact':
            return <ContactUsPage />;
        case 'settings':
            return <SettingsPage />;
        case 'cart':
            return <CartPage />;
        default:
            return <HomePage />;
    }
};

return (
    // Provide theme and cart contexts to all children components
    <ThemeContext.Provider value={{ theme, toggleTheme }}>
    <CartContext.Provider value={{ cartItems, addToCart,
removeFromCart, updateQuantity }}>
        <div className={`min-h-screen flex flex-col transition-colors
duration-500`}>
            <Header navigate={navigate} />
            <main className="flex-grow container mx-auto p-4">
                {renderPage()}
            </main>
            <Footer />

```

```
        </div>
      </CartContext.Provider>
    </ThemeContext.Provider>
  );
};

export default App;
```