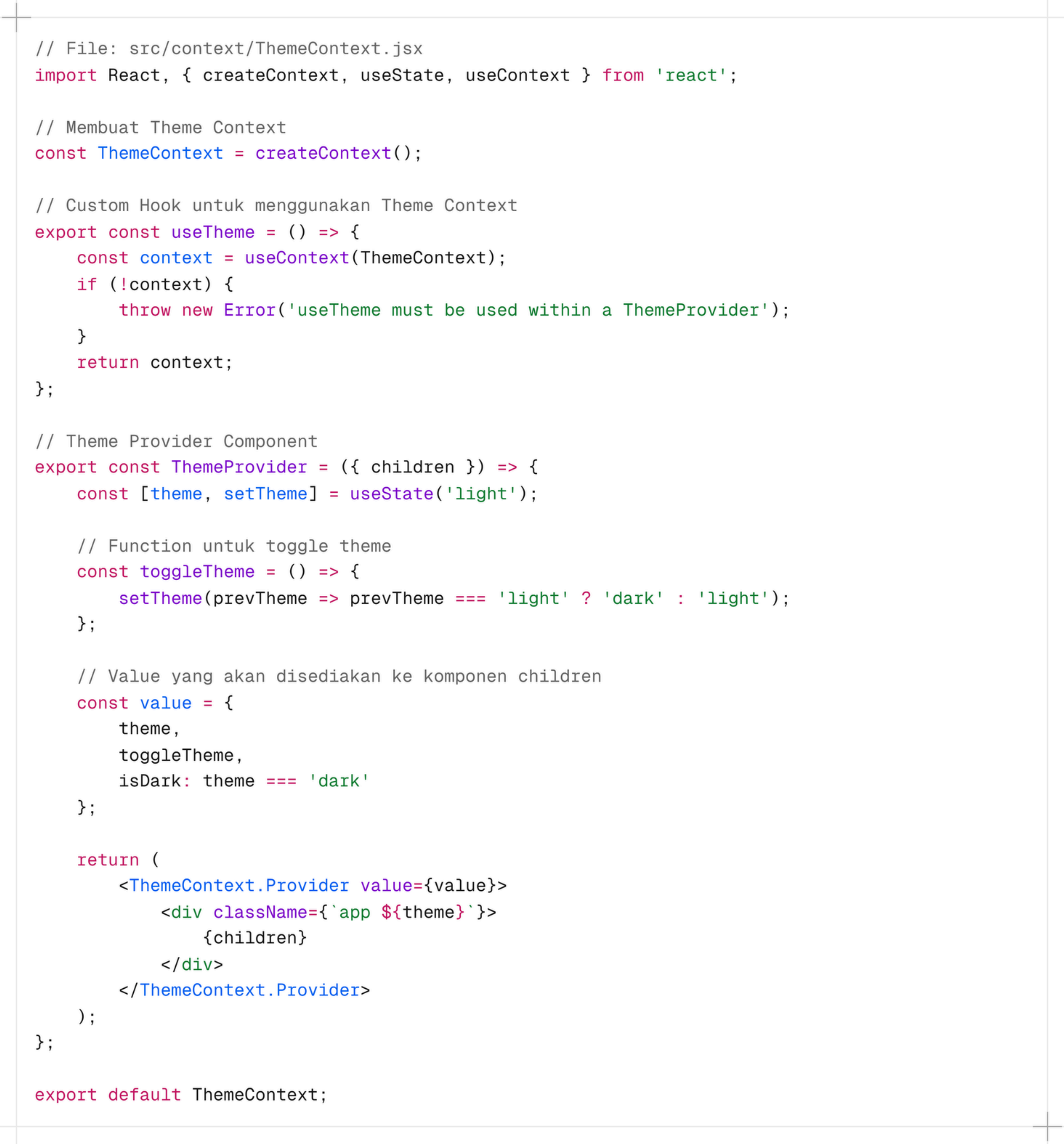
Praktik Pertemuan 9

Langkah 1 : Buat Project

```
npm create vite@latest praktik-9 -- --template react

cd praktik-9

npm install

npm run dev
```

Langkah 2 : ThemeContext.jsx

File : src/context/ThemeContext.jsx



```jsx
// File: src/context/ThemeContext.jsx
import React, { createContext, useState, useContext } from 'react';

// Membuat Theme Context
const ThemeContext = createContext();

// Custom Hook untuk menggunakan Theme Context
export const useTheme = () => {
    const context = useContext(ThemeContext);
    if (!context) {
        throw new Error('useTheme must be used within a ThemeProvider');
    }
    return context;
};

// Theme Provider Component
export const ThemeProvider = ({ children }) => {
    const [theme, setTheme] = useState('light');

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

    // Value yang akan disediakan ke komponen children
    const value = {
        theme,
        toggleTheme,
        isDark: theme === 'dark'
    };

    return (
        <ThemeContext.Provider value={value}>
            <div className={`app ${theme}`}>
                {children}
            </div>
        </ThemeContext.Provider>
    );
};

export default ThemeContext;
```

Langkah 3 : src/context/UserContext.jsx

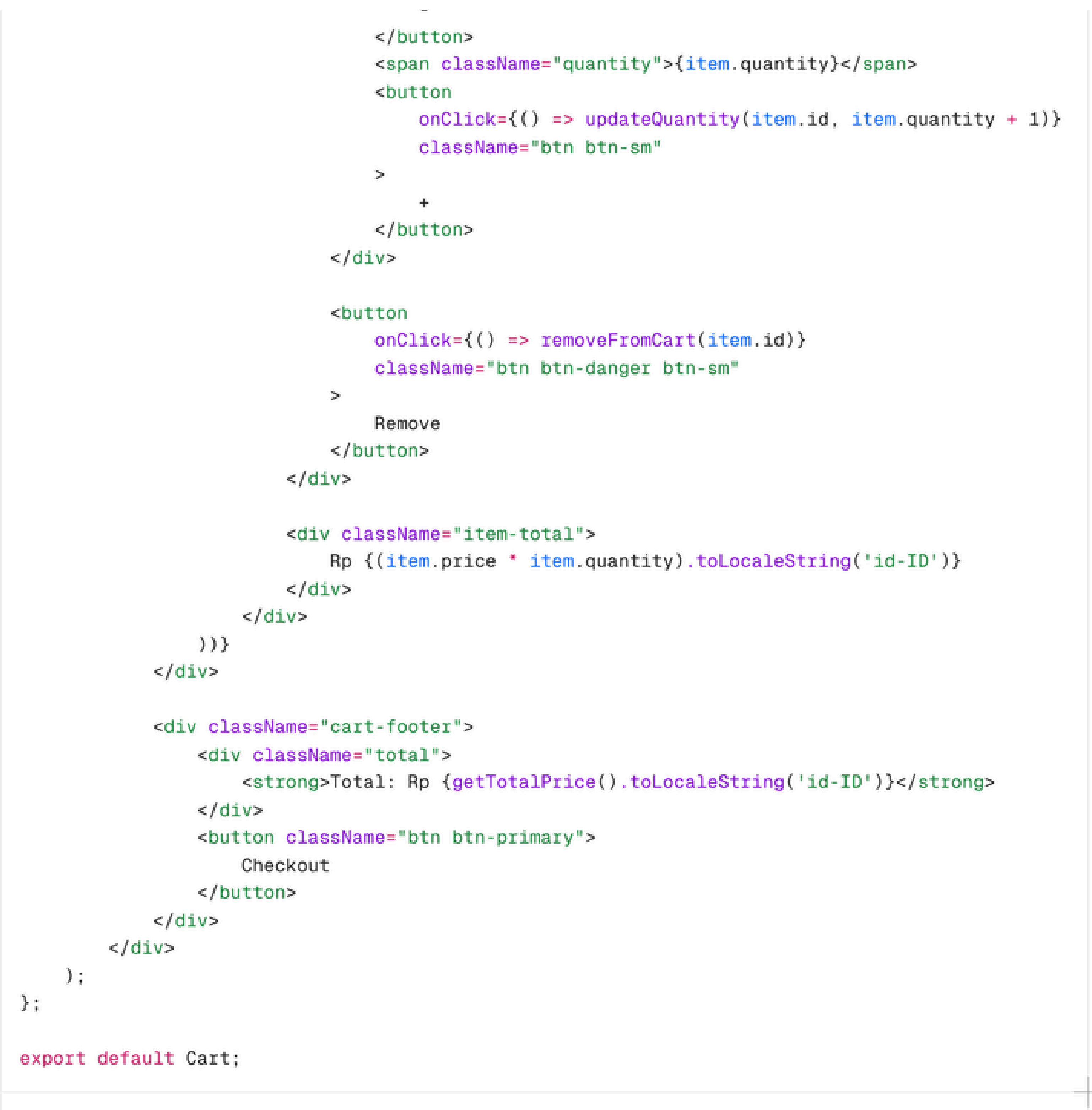
```
// File: src/context/UserContext.jsx
import React, { createContext, useState, useContext } from 'react';

// Membuat User Context
const UserContext = createContext();

// Custom Hook untuk menggunakan User Context
export const useUser = () => {
    const context = useContext(UserContext);
    if (!context) {
        throw new Error('useUser must be used within a UserProvider');
    }
    return context;
};

// User Provider Component
export const UserProvider = ({ children }) => {
    const [user, setUser] = useState(null);
    const [isLoading, setIsLoading] = useState(false);

    // Login function
    const login = async (email, password) => {
        setIsLoading(true);
        // Simulasi API call
        setTimeout(() => {
            setUser({
                id: 1,
                name: 'John Doe',
                email: email,
                avatar: '\u{1F464}'
            });
            setIsLoading(false);
        }, 1000);
    };

    // Logout function
    const logout = () => {
        setUser(null);
    };

    // Update user profile
    const updateProfile = (updatedData) => {
        setUser(prev => ({ ...prev, ...updatedData }));
    };

    const value = {
        user,
        isLoading,
        login,
        logout,
        updateProfile,
        isAuthenticated: !!user
    };
    };

    return (
        <UserContext.Provider value={value}>
            {children}
        </UserContext.Provider>
    );
};

export default UserContext;
```

Langkah 4 : src/context/CartContext.jsx

```
// File: src/context/CartContext.jsx
import React, { createContext, useState, useContext } from 'react';

// Membuat Cart Context
const CartContext = createContext();

// Custom Hook untuk menggunakan Cart Context
export const useCart = () => {
    const context = useContext(CartContext);
    if (!context) {
        throw new Error('useCart must be used within a CartProvider');
    }
    return context;
};

// Cart Provider Component
export const CartProvider = ({ children }) => {
    const [cartItems, setCartItems] = useState([]);

    // Add item 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
                );
            }

            return [...prevItems, { ...product, quantity: 1 }];
        });
    };

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

    // Update item quantity
    const updateQuantity = (productId, newQuantity) => {
        if (newQuantity <= 0) {
            removeFromCart(productId);
            return;
        }

        setCartItems(prevItems =>
            prevItems.map(item =>
                item.id === productId
                    ? { ...item, quantity: newQuantity }
                    : item
                    : item
            )
        );
    };

    // Clear cart
    const clearCart = () => {
        setCartItems([]);
    };

    // Calculate total items
    const getTotalItems = () => {
        return cartItems.reduce((total, item) => total + item.quantity, 0);
    };

    // Calculate total price
    const getTotalPrice = () => {
        return cartItems.reduce((total, item) => total + (item.price * item.quantity), 0);
    };

    const value = {
        cartItems,
        addToCart,
        removeFromCart,
        updateQuantity,
        clearCart,
        getTotalItems,
        getTotalPrice
    };

    return (
        <CartContext.Provider value={value}>
            {children}
        </CartContext.Provider>
    );
};

export default CartContext;
```

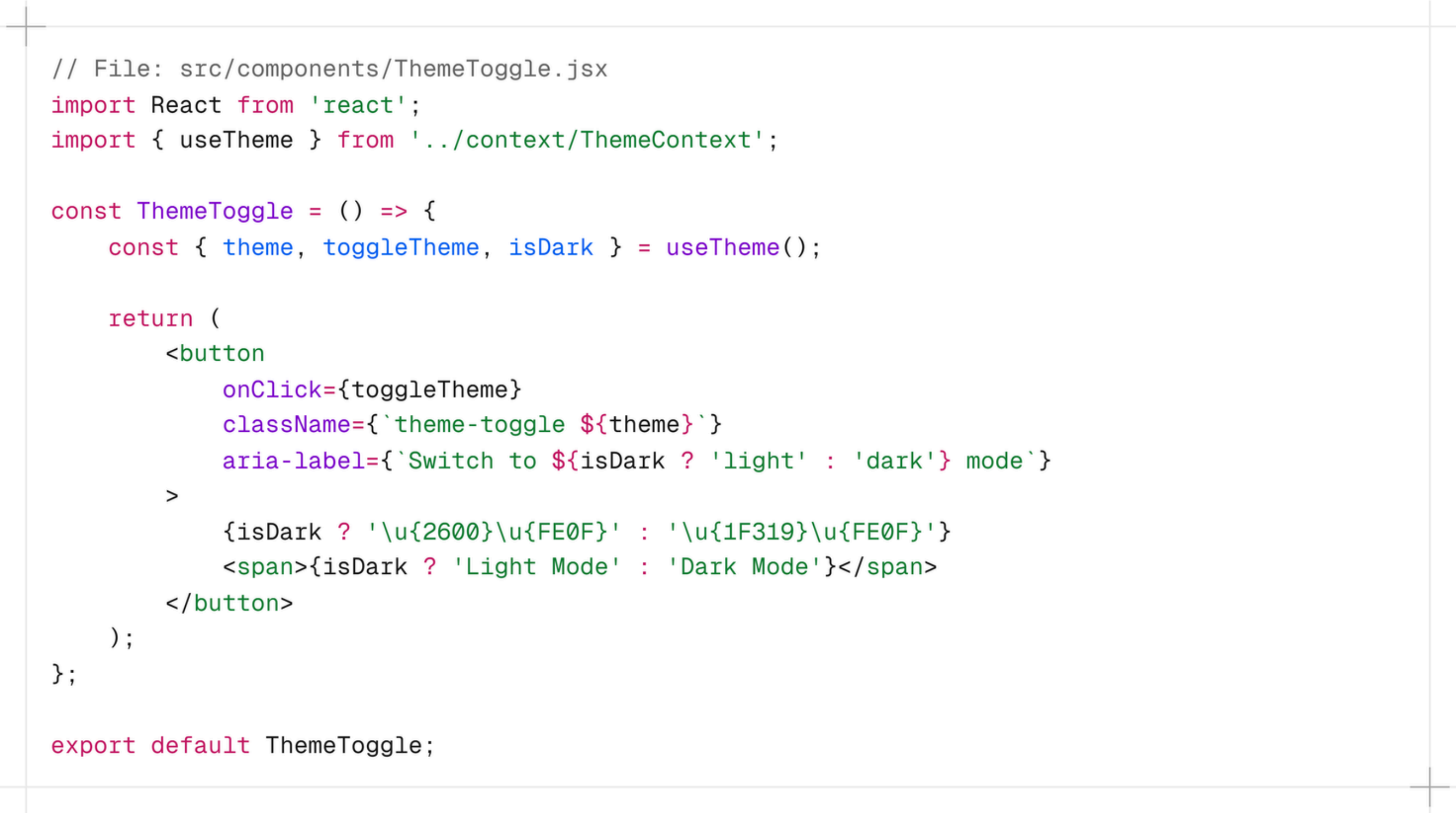
Langkah 5 : src/components/Cart.jsx

```
// File: src/components/Cart.jsx
import React from 'react';
import { useTheme } from '../context/ThemeContext';
import { useCart } from '../context/CartContext';
import { useUser } from '../context/UserContext';

const Cart = () => {
    const { theme } = useTheme();
    const { cartItems, removeFromCart, updateQuantity, clearCart, getTotalPrice } = useCart();
    const { isAuthenticated } = useUser();

    if (!isAuthenticated) {
        return (
            <div className={`cart ${theme}`}>
                <h2>Shopping Cart</h2>
                <p>Please login to view your cart</p>
            </div>
        );
    }

    if (cartItems.length === 0) {
        return (
            <div className={`cart ${theme}`}>
                <h2>Shopping Cart</h2>
                <p>Your cart is empty</p>
            </div>
        );
    }

    return (
        <div className={`cart ${theme}`}>
            <div className="cart-header">
                <h2>Shopping Cart ({cartItems.length} items)</h2>
                <button onClick={clearCart} className="btn btn-outline">
                    Clear Cart
                </button>
            </div>

            <div className="cart-items">
                {cartItems.map(item => (
                    <div key={item.id} className="cart-item">
                        <div className="item-info">
                            <h4>{item.name}</h4>
                            <p>Rp {item.price.toLocaleString('id-ID')}</p>
                        </div>

                        <div className="item-controls">
                            <div className="quantity-controls">
                                <button
                                    onClick={() => updateQuantity(item.id, item.quantity - 1)}
                                    className="btn btn-sm"
                                >
                                    -
```



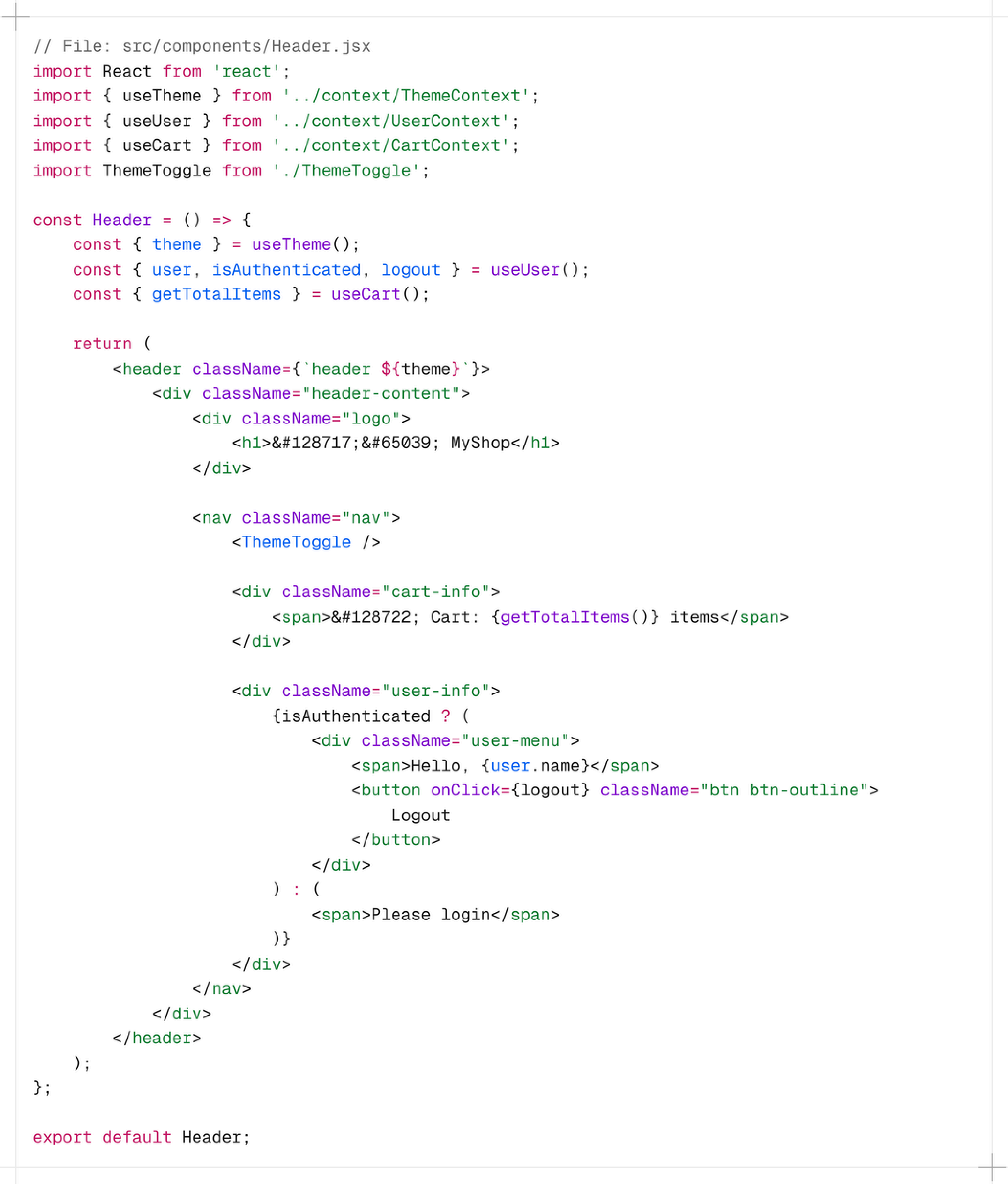
```
                                    -
                                </button>
                                <span className="quantity">{item.quantity}</span>
                                <button
                                    onClick={() => updateQuantity(item.id, item.quantity + 1)}
                                    className="btn btn-sm"
                                >
                                    +
                                </button>
                            </div>

                            <button
                                onClick={() => removeFromCart(item.id)}
                                className="btn btn-danger btn-sm"
                            >
                                Remove
                            </button>
                        </div>

                        <div className="item-total">
                            Rp {(item.price * item.quantity).toLocaleString('id-ID')}
                        </div>
                    </div>
                ))}
            </div>

            <div className="cart-footer">
                <div className="total">
                    <strong>Total: Rp {getTotalPrice().toLocaleString('id-ID')}</strong>
                </div>
                <button className="btn btn-primary">
                    Checkout
                </button>
            </div>
        </div>
    );
};

export default Cart;
```

Langkah 6 : src/components/ThemeToggle.jsx



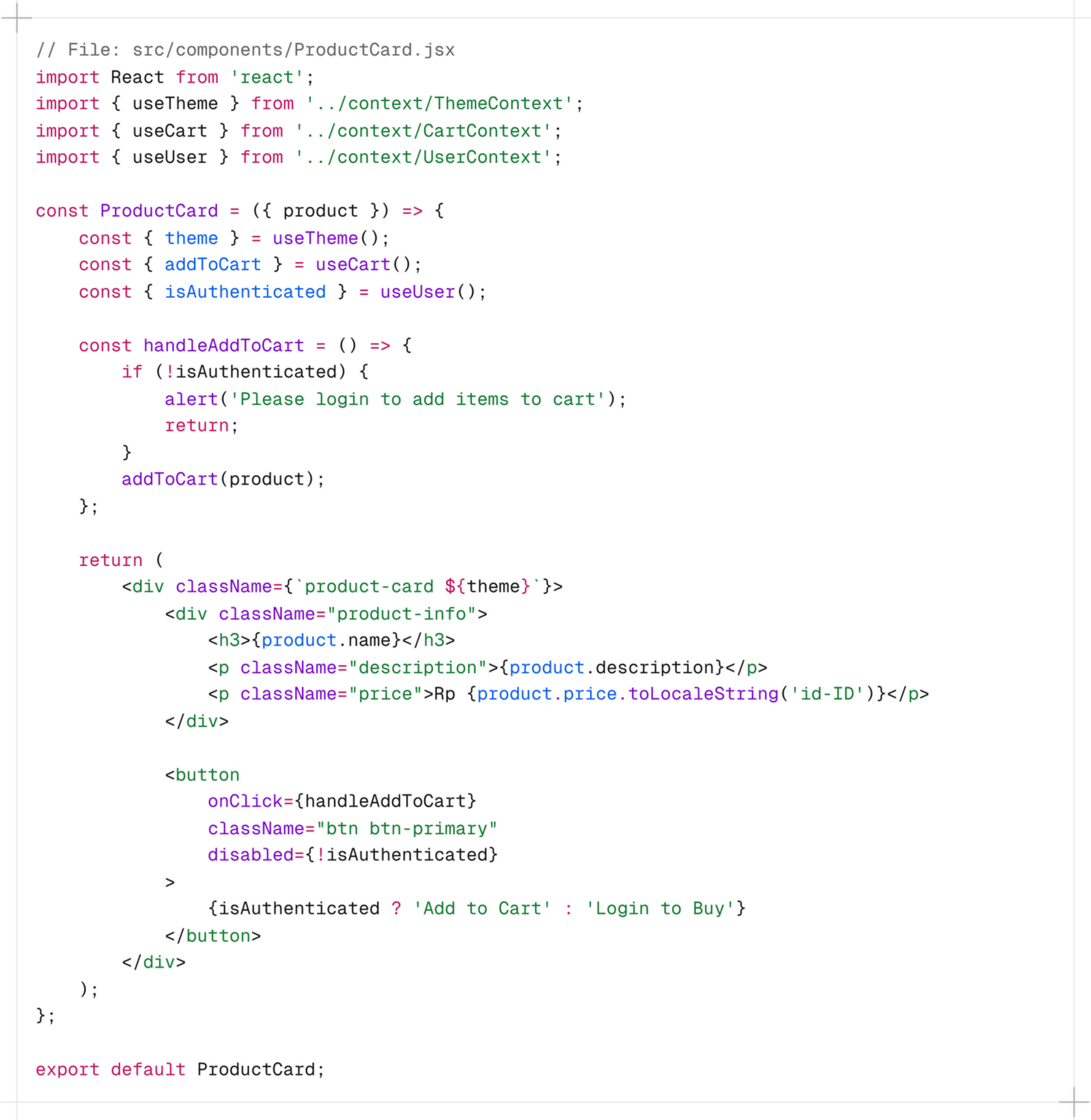
```jsx
// File: src/components/ThemeToggle.jsx
import React from 'react';
import { useTheme } from '../context/ThemeContext';

const ThemeToggle = () => {
    const { theme, toggleTheme, isDark } = useTheme();

    return (
        <button
            onClick={toggleTheme}
            className={`theme-toggle ${theme}`}
            aria-label={`Switch to ${isDark ? 'light' : 'dark'} mode`}
        >
            {isDark ? '\u{2600}\u{FE0F}' : '\u{1F319}\u{FE0F}'}
            <span>{isDark ? 'Light Mode' : 'Dark Mode'}</span>
        </button>
    );
};

export default ThemeToggle;
```

Langkah 7 : src/components/Header.jsx



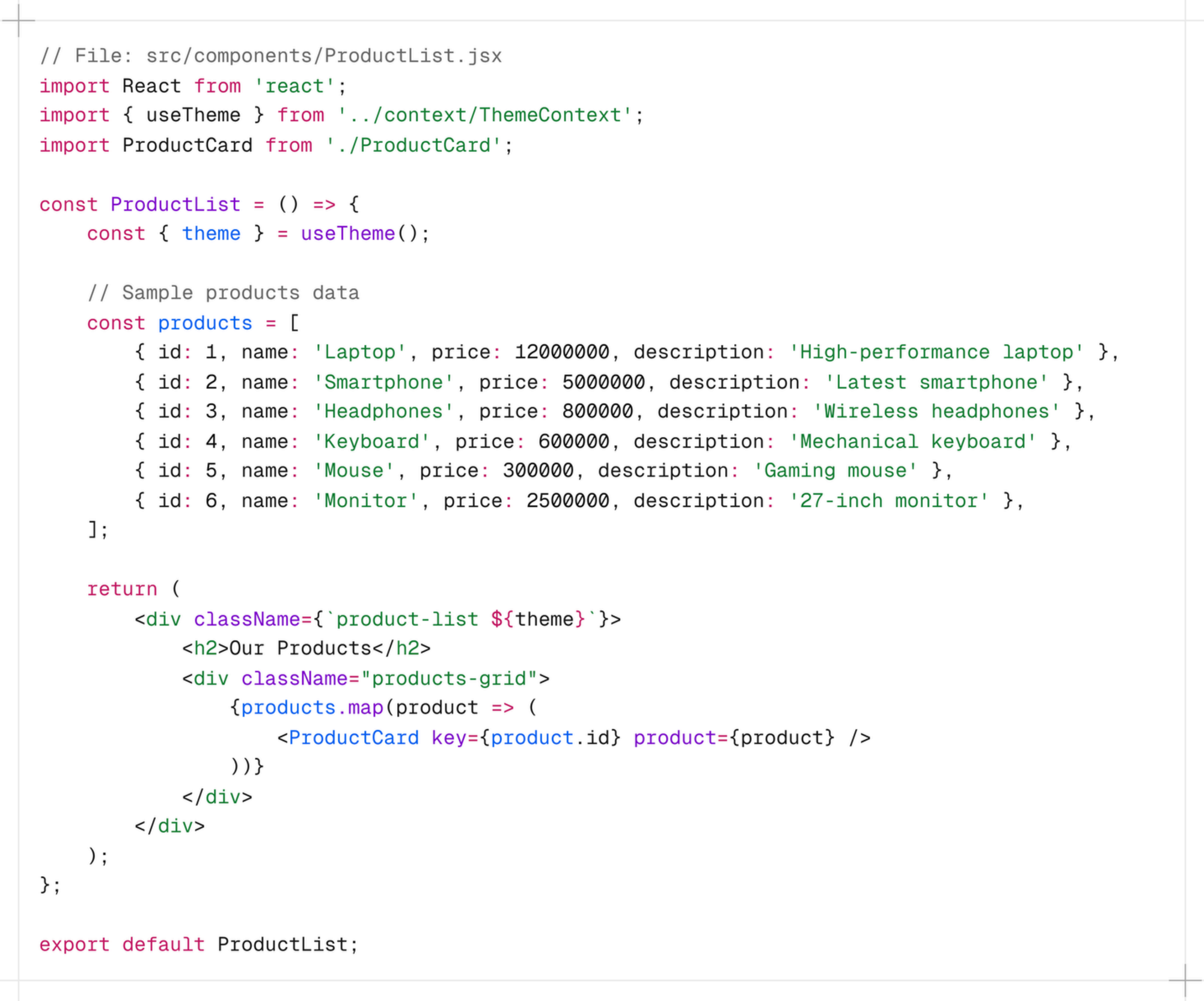
```jsx
// File: src/components/Header.jsx
import React from 'react';
import { useTheme } from '../context/ThemeContext';
import { useUser } from '../context/UserContext';
import { useCart } from '../context/CartContext';
import ThemeToggle from './ThemeToggle';

const Header = () => {
    const { theme } = useTheme();
    const { user, isAuthenticated, logout } = useUser();
    const { getTotalItems } = useCart();

    return (
        <header className={`header ${theme}`}>
            <div className="header-content">
                <div className="logo">
                    <h1>&#128717;&#65039; MyShop</h1>
                </div>

                <nav className="nav">
                    <ThemeToggle />

                    <div className="cart-info">
                        <span>&#128722; Cart: {getTotalItems()} items</span>
                    </div>

                    <div className="user-info">
                        {isAuthenticated ? (
                            <div className="user-menu">
                                <span>Hello, {user.name}</span>
                                <button onClick={logout} className="btn btn-outline">
                                    Logout
                                </button>
                            </div>
                        ) : (
                            <span>Please login</span>
                        )}
                    </div>
                </nav>
            </div>
        </header>
    );
};

export default Header;
```

Langkah 8 : src/components/ProductCard.jsx



```jsx
// File: src/components/ProductCard.jsx
import React from 'react';
import { useTheme } from '../context/ThemeContext';
import { useCart } from '../context/CartContext';
import { useUser } from '../context/UserContext';

const ProductCard = ({ product }) => {
    const { theme } = useTheme();
    const { addToCart } = useCart();
    const { isAuthenticated } = useUser();

    const handleAddToCart = () => {
        if (!isAuthenticated) {
            alert('Please login to add items to cart');
            return;
        }
        addToCart(product);
    };

    return (
        <div className={`product-card ${theme}`}>
            <div className="product-info">
                <h3>{product.name}</h3>
                <p className="description">{product.description}</p>
                <p className="price">Rp {product.price.toLocaleString('id-ID')}</p>
            </div>

            <button
                onClick={handleAddToCart}
                className="btn btn-primary"
                disabled={!isAuthenticated}
            >
                {isAuthenticated ? 'Add to Cart' : 'Login to Buy'}
            </button>
        </div>
    );
};

export default ProductCard;
```

Langkah 9 : src/components/ProductList.jsx

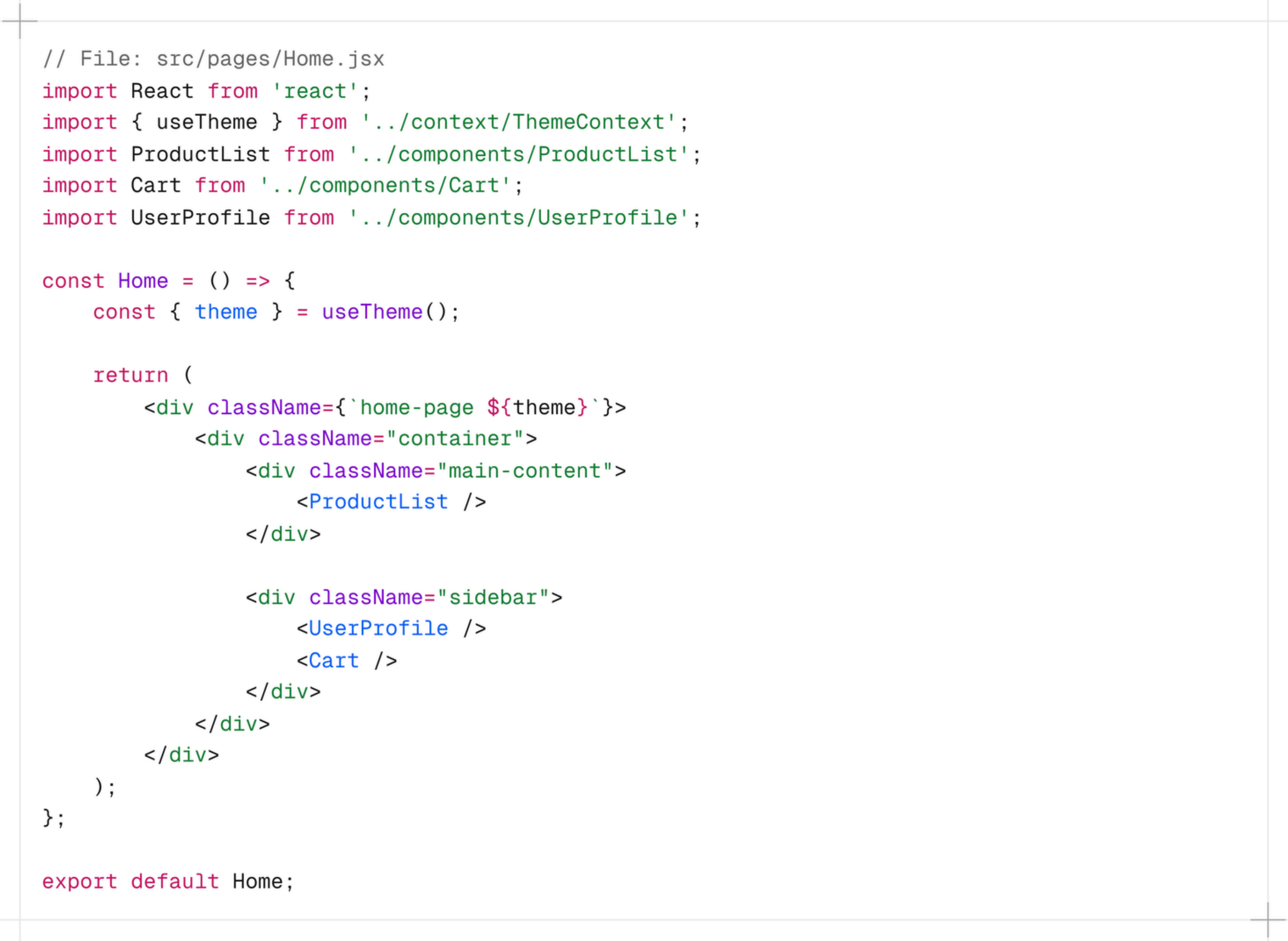


```jsx
// File: src/components/ProductList.jsx
import React from 'react';
import { useTheme } from '../context/ThemeContext';
import ProductCard from './ProductCard';

const ProductList = () => {
    const { theme } = useTheme();

    // Sample products data
    const products = [
        { id: 1, name: 'Laptop', price: 12000000, description: 'High-performance laptop' },
        { id: 2, name: 'Smartphone', price: 5000000, description: 'Latest smartphone' },
        { id: 3, name: 'Headphones', price: 800000, description: 'Wireless headphones' },
        { id: 4, name: 'Keyboard', price: 600000, description: 'Mechanical keyboard' },
        { id: 5, name: 'Mouse', price: 300000, description: 'Gaming mouse' },
        { id: 6, name: 'Monitor', price: 2500000, description: '27-inch monitor' },
    ];

    return (
        <div className={`product-list ${theme}`}>
            <h2>Our Products</h2>
            <div className="products-grid">
                {products.map(product => (
                    <ProductCard key={product.id} product={product} />
                ))}
            </div>
        </div>
    );
};

export default ProductList;
```

Langkah 10 : src/components/UserProfile.jsx

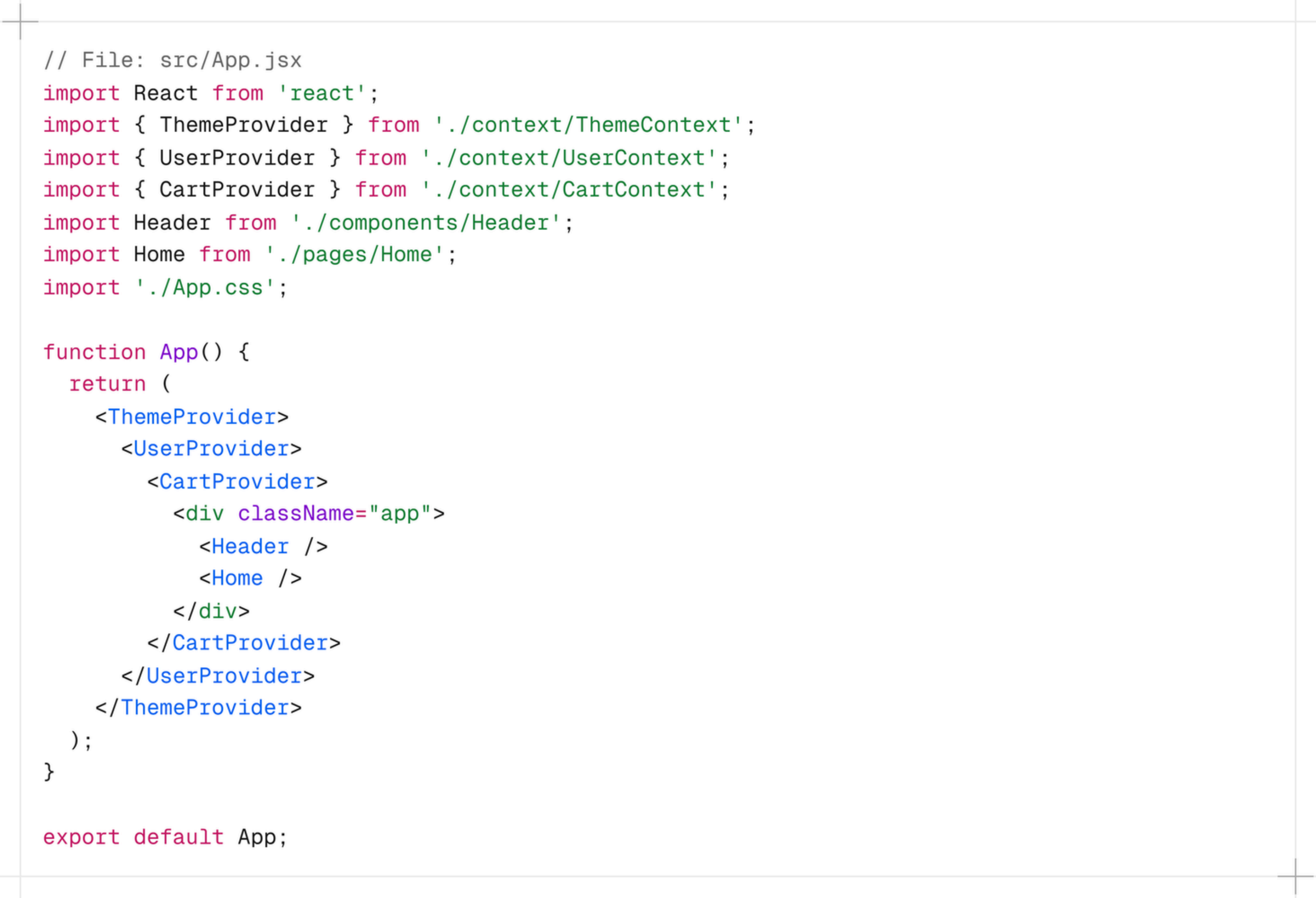
```
// File: src/components/UserProfile.jsx
import React, { useState } from 'react';
import { useTheme } from '../context/ThemeContext';
import { useUser } from '../context/UserContext';

const UserProfile = () => {
    const { theme } = useTheme();
    const { user, isAuthenticated, login, logout, updateProfile, isLoading } = useUser();
    const [email, setEmail] = useState('');
    const [password, setPassword] = useState('');
    const [editMode, setEditMode] = useState(false);
    const [name, setName] = useState(user?.name || '');

    const handleLogin = (e) => {
        e.preventDefault();
        login(email, password);
        setEmail('');
        setPassword('');
    };

    const handleUpdateProfile = (e) => {
        e.preventDefault();
        updateProfile({ name });
        setEditMode(false);
    };

    if (!isAuthenticated) {
        return (
            <div className={`user-profile ${theme}`}>
                <h2>Login</h2>
                <form onSubmit={handleLogin} className="login-form">
                    <div className="form-group">
                        <label>Email:</label>
                        <input type="email" value={email} onChange={(e) =>
setEmail(e.target.value)} required/>
                    </div>

                    <div className="form-group">
                        <label>Password:</label>
                        <input type="password" value={password} onChange={(e) =>
setPassword(e.target.value)} required/>
                    </div>

                    <button type="submit" disabled={isLoading} className="btn btn-primary">
                        {isLoading ? 'Logging in...' : 'Login'}
                    </button>
                </form>

                <div className="demo-credentials">
                    <p><strong>Demo:</strong> Use any email and password</p>
                </div>
            </div>
        );
    }

    return (
        <div className={`user-profile ${theme}`}>
            <h2>User Profile</h2>

            {!editMode ? (
                <div className="profile-info">
                    <div className="avatar">{user.avatar}</div>
                    <div className="details">
                        <p><strong>Name:</strong> {user.name}</p>
                        <p><strong>Email:</strong> {user.email}</p>
                    </div>
                    <button onClick={() => setEditMode(true)} className="btn btn-outline">
                        Edit Profile
                    </button>
                </div>
            ) : (
                <form onSubmit={handleUpdateProfile} className="edit-form">
                    <div className="form-group">
                        <label>Name:</label>
                        <input type="text" value={name} onChange={(e) =>
setName(e.target.value)} required />
                    </div>

                    <div className="form-actions">
                        <button type="submit" className="btn btn-primary">
                            Save
                        </button>
                        <button type="button" onClick={() => setEditMode(false)}
className="btn btn-secondary">
                            Cancel
                        </button>
                    </div>
                </form>
            )}
        </div>
    );
};

export default UserProfile;
```

Langkah 11 : src/pages/Home.jsx



```
// File: src/pages/Home.jsx
import React from 'react';
import { useTheme } from '../context/ThemeContext';
import ProductList from '../components/ProductList';
import Cart from '../components/Cart';
import UserProfile from '../components/UserProfile';

const Home = () => {
    const { theme } = useTheme();

    return (
        <div className={`home-page ${theme}`}>
            <div className="container">
                <div className="main-content">
                    <ProductList />
                </div>

                <div className="sidebar">
                    <UserProfile />
                    <Cart />
                </div>
            </div>
        </div>
    );
};

export default Home;
```

Langkah 12 : src/App.jsx



```
// File: src/App.jsx
import React from 'react';
import { ThemeProvider } from './context/ThemeContext';
import { UserProvider } from './context/UserContext';
import { CartProvider } from './context/CartContext';
import Header from './components/Header';
import Home from './pages/Home';
import './App.css';

function App() {
  return (
    <ThemeProvider>
      <UserProvider>
        <CartProvider>
          <div className="app">
            <Header />
            <Home />
          </div>
        </CartProvider>
      </UserProvider>
    </ThemeProvider>
  );
}

export default App;
```

Langkah 13 : App.css

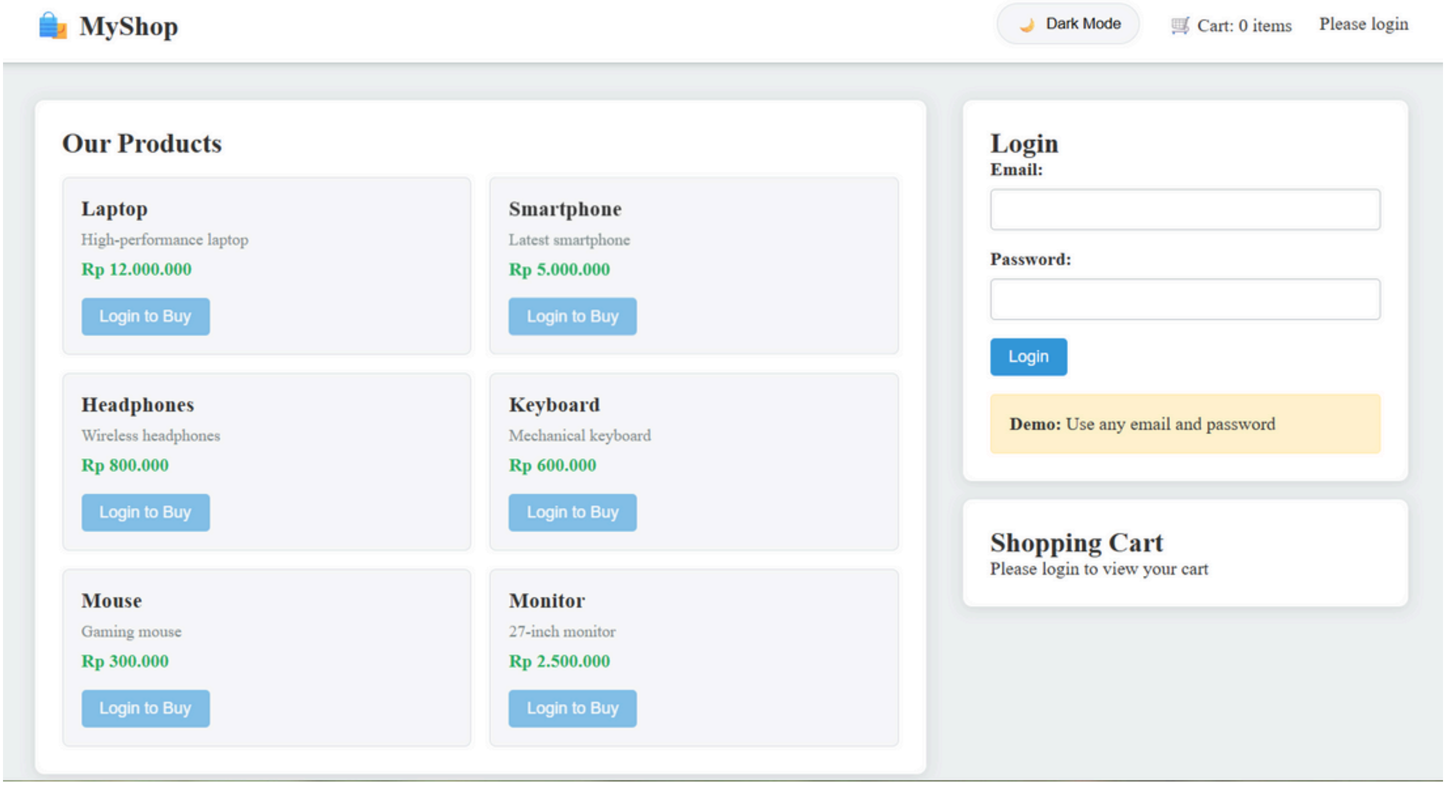
```css
/* File: src/App.css */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

:root {
  --primary-color: #3498db;
  --secondary-color: #2c3e50;
  --success-color: #27ae60;
  --danger-color: #e74c3c;
  --warning-color: #f39c12;
  --light-bg: #ecf0f1;
  --dark-bg: #2c3e50;
  --light-text: #333;
  --dark-text: #ecf0f1;
}

/* Base Theme Styles */
.app.light {
  background-color: var(--light-bg);
  color: var(--light-text);
  min-height: 100vh;
}

.app.dark {
  background-color: var(--dark-bg);
  color: var(--dark-text);
  min-height: 100vh;
}

/* Header Styles */
.header {
  padding: 1rem 2rem;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
}

.header.light {
  background: white;
  border-bottom: 1px solid #bdc3c7;
}

.header.dark {
  background: #34495e;
  border-bottom: 1px solid #46637f;
```

```css
}
.header-content {
  display: flex;
  justify-content: space-between;
  align-items: center;
  max-width: 1200px;
  margin: 0 auto;
}
.logo h1 {
  font-size: 1.5rem;
}
.nav {
  display: flex;
  align-items: center;
  gap: 1.5rem;
}
/* Theme Toggle */
.theme-toggle {
  display: flex;
  align-items: center;
  gap: 0.5rem;
  padding: 0.5rem 1rem;
  border: none;
  border-radius: 20px;
  cursor: pointer;
  transition: all 0.3s ease;
}
.theme-toggle.light {
  background: #f8f9fa;
  border: 1px solid #dee2e6;
}
.theme-toggle.dark {
  background: #46637f;
  border: 1px solid #5d7a95;
  color: white;
}
.theme-toggle:hover {
  transform: translateY(-2px);
}
/* Button Styles */
.btn {
  padding: 0.5rem 1
  padding: 0.5rem 1rem;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  font-size: 0.9rem;
  transition: all 0.3s ease;
  text-decoration: none;
  display: inline-block;
}

.btn-primary {
  background: var(--primary-color);
  color: white;
}

.btn-primary:hover:not(:disabled) {
  background: #2980b9;
}

.btn-secondary {
  background: #95a5a6;
  color: white;
}

.btn-outline {
  background: transparent;
  border: 1px solid currentColor;
}

.btn-danger {
  background: var(--danger-color);
  color: white;
}

.btn-sm {
  padding: 0.25rem 0.5rem;
  font-size: 0.8rem;
}

.btn:disabled {
  opacity: 0.6;
  cursor: not-allowed;
}

/* Home Page Layout */
.home-page {
  padding: 2rem;
}

.container {
```

```css
  display: grid;
  grid-template-columns: 2fr 1fr;
  gap: 2rem;
  max-width: 1200px;
  margin: 0 auto;
}

/* Product Styles */
.product-list.light {
  background: white;
}

.product-list.dark {
  background: #34495e;
}

.product-list {
  padding: 1.5rem;
  border-radius: 8px;
  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
}

.products-grid {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));
  gap: 1rem;
  margin-top: 1rem;
}

.product-card {
  padding: 1rem;
  border-radius: 6px;
  transition: transform 0.3s ease;
}

.product-card.light {
  background: #f8f9fa;
  border: 1px solid #dee2e6;
}

.product-card.dark {
  background: #46637f;
  border: 1px solid #5d7a95;
}

.product-card:hover {
  transform: translateY(-2px);
}

.product-info h3 {
  margin-bottom: 0.5rem;
}

.description {
  color: #7f8c8d;
  font-size: 0.9rem;
  margin-bottom: 0.5rem;
}

.price {
  font-weight: bold;
  color: var(--success-color);
  margin-bottom: 1rem;
}

/* Cart Styles */
.cart {
  padding: 1.5rem;
  border-radius: 8px;
  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
  margin-bottom: 1rem;
}

.cart.light {
  background: white;
}

.cart.dark {
  background: #34495e;
}

.cart-header {
  display: flex;
  justify-content: space-between;
  align-items: center;
  margin-bottom: 1rem;
}

.cart-item {
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 1rem 0;
  border-bottom: 1px solid #bdc3c7;
}

.item-controls {
  display: flex;
  align-items: center;
  gap: 1rem;
}

.quantity-controls {
  display: flex;
  align-items: center;
  gap: 0.5rem;
}

.quantity {
  padding: 0 0.5rem;
}

.cart-footer {
  margin-top: 1rem;
  padding-top: 1rem;
  border-top: 2px solid #bdc3c7;
  display: flex;
  justify-content: space-between;
  align-items: center;
}

/* User Profile Styles */
.user-profile {
  padding: 1.5rem;
  border-radius: 8px;
  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
  margin-bottom: 1rem;
}

.user-profile.light {
  background: white;
}

.user-profile.dark {
  background: #34495e;
}

.profile-info {
  display: flex;
  flex-direction: column;
  gap: 1rem;
}

.avatar {
  font-size: 3rem;
  text-align: center;
}

.details p {
  margin-bottom: 0.5
```

```css
  margin-bottom: 0.5rem;
}

/* Form Styles */
.form-group {
  margin-bottom: 1rem;
}

.form-group label {
  display: block;
  margin-bottom: 0.5rem;
  font-weight: bold;
}

.form-group input {
  width: 100%;
  padding: 0.5rem;
  border: 1px solid #bdc3c7;
  border-radius: 4px;
  font-size: 1rem;
}

.form-actions {
  display: flex;
  gap: 0.5rem;
}

.demo-credentials {
  margin-top: 1rem;
  padding: 1rem;
  background: #fff3cd;
  border-radius: 4px;
  border: 1px solid #ffeaa7;
}

/* Responsive Design */
@media (max-width: 768px) {
  .container {
    grid-template-columns: 1fr;
  }

  .header-content {
    flex-direction: column;
    gap: 1rem;
  }

  .nav {
    flex-wrap: wrap;
    justify-content: center;
  }
  }

  .products-grid {
    grid-template-columns: 1fr;
  }

  .cart-item {
    flex-direction: column;
    align-items: flex-start;
    gap: 1rem;
  }

  .item-controls {
    width: 100%;
    justify-content: space-between;
  }
}
```

Hasil



🛍️ MyShop

🌙 Dark Mode

🛒 Cart: 0 items

Please login

Our Products

Laptop

High-performance laptop

Rp 12.000.000

Login to Buy

Smartphone

Latest smartphone

Rp 5.000.000

Login to Buy

Headphones

Wireless headphones

Rp 800.000

Login to Buy

Keyboard

Mechanical keyboard

Rp 600.000

Login to Buy

Mouse

Gaming mouse

Rp 300.000

Login to Buy

Monitor

27-inch monitor

Rp 2.500.000

Login to Buy

Login

Email:

Password:

Login

Demo: Use any email and password

Shopping Cart

Please login to view your cart