E-Commerce App Test Plan

# Test Objectives

Describe the main objectives of testing this application.

# Scope

Define what is in and out of scope for this test plan.

# Test Methodology

Outline the testing methods to be used (manual, automated, etc.).

# Approach

Describe the overall approach to testing, including levels and types of testing.

# Assumptions

List any assumptions made during the planning of tests.

# Risks

Identify potential risks that could impact testing.

# Role and Responsibilities

Define the roles and responsibilities of the testing team.

# Schedule

Provide a high-level schedule or timeline for testing activities.

# Defect Logging

Describe the process and tools for defect logging and tracking.

# Test Environment

Specify the environment(s) in which testing will be conducted.

# Entry and Exit Condition

Define the criteria for starting and ending testing phases.

# Automation

Describe the automation strategy and tools to be used.

# Effort Estimation

Estimate the effort required for testing activities.

# Deliverables

List the expected deliverables from the testing process.

# Template

The following table provides the detailed test cases for the application.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Test Cases | Test Scripts | Test Data |
| App.js | Renders Navbar | Render <App />, check for Navbar component | N/A |
| App.js | Renders Home at '/' | Render <App /> at '/', expect 'Welcome to ShopZen' | N/A |
| App.js | Renders Register at '/register' | Render <App /> at '/register', expect 'Register' heading | N/A |
| App.js | Renders Login at '/login' | Render <App /> at '/login', expect 'Login' heading | N/A |
| App.js | Redirects unauthenticated from '/dashboard' and '/cart' | Render <App /> at '/dashboard' or '/cart' without login, expect redirect to '/login' | N/A |
| App.js | Renders Dashboard/Cart for authenticated users | Set localStorage, render <App /> at '/dashboard' or '/cart', expect correct page | { loggedInUser: "testuser" } |
| Navbar.js | Shows Register/Login links when logged out | Render <Navbar /> without login, expect 'Register' and 'Login' links | N/A |
| Navbar.js | Shows Dashboard/Cart/Logout when logged in | Render <Navbar /> with login, expect 'Dashboard', 'Cart', 'Logout' | { loggedInUser: "testuser" } |
| Navbar.js | Logout clears localStorage and redirects | Click 'Logout', expect localStorage cleared and redirect to '/' | { loggedInUser: "testuser" } |
| Home.js | Renders welcome message | Render <Home />, expect 'Welcome to ShopZen' | N/A |
| Home.js | Register/Login buttons navigate correctly | Click 'Register'/'Login' button, expect navigation to '/register'/'/login' | N/A |
| Register.js | Renders username/password fields and button | Render <Register />, expect input fields and 'Register' button | N/A |
| Register.js | Calls API on register, shows success message | Mock axios, fill form, click 'Register', expect success message | { username: "alice", password: "pass" } |
| Register.js | Handles API error and shows error message | Mock axios error, fill form, click 'Register', expect error message | { username: "", password: "" } |
| Login.js | Renders username/password fields and button | Render <Login />, expect input fields and 'Login' button | N/A |
| Login.js | Calls API on login, sets localStorage, redirects | Mock axios, fill form, click 'Login', expect localStorage set and redirect | { username: "bob", password: "pass" } |
| Login.js | Handles API error and shows error message | Mock axios error, fill form, click 'Login', expect error message | { username: "bob", password: "wrong" } |
| Dashboard.js | Greets user by username | Set localStorage, render <Dashboard />, expect greeting with username | { loggedInUser: "bob" } |
| Dashboard.js | Renders all products from data | Render <Dashboard />, expect product cards for each product in products.json | products.json |
| Dashboard.js | Add to cart updates localStorage | Click 'Add to Cart', expect product added to localStorage | products.json, { loggedInUser: "bob" } |
| Cart.js | Renders items from localStorage | Set localStorage cart, render <Cart />, expect items displayed | { cart: [product1, product2] } |
| Cart.js | Remove item updates cart | Click 'Remove' on item, expect item removed from localStorage and UI | { cart: [product1, product2] } |
| Cart.js | Clear all empties cart and localStorage | Click 'Clear All', expect cart empty and localStorage cleared | { cart: [product1, product2] } |
| Cart.js | Shows total cost | Render <Cart />, expect total cost equals sum of product prices | { cart: [product1, product2] } |
| Cart.js | Generate Invoice displays invoice section | Click 'Generate Invoice', expect invoice section with itemized list and total | { cart: [product1, product2] } |