

Day 5 - Testing and Backend Refinement of E - Commerce Website

As we progress toward launching our marketplace, Day 5 is dedicated to ensuring that all components are fully tested, optimized for performance, and ready to handle real-world customer interactions.

Key Areas of Focus :-

1. Functional Testing :

This document outlines the functional testing strategy for the Next.js e-commerce application. The testing covers core features such as product listing, filtering, cart operations, and dynamic routing using automated and manual testing approaches.

Tested Core Features :

1.1 Client-Side Navigation Between Components :

Tested Core Features :

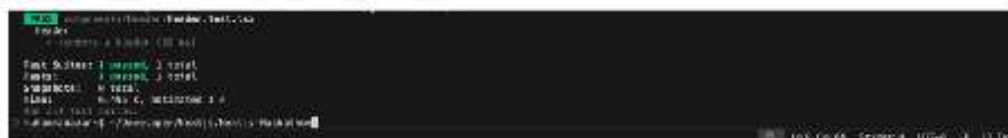
1.1 Client-Side Navigation Between Components :

- **Tested with React Testing Library:**
- Verified smooth navigation across different sections of the website (e.g., home page to product listing, product details, cart, and wishlist pages)
- Ensured navigation links and buttons function correctly without a full page reload

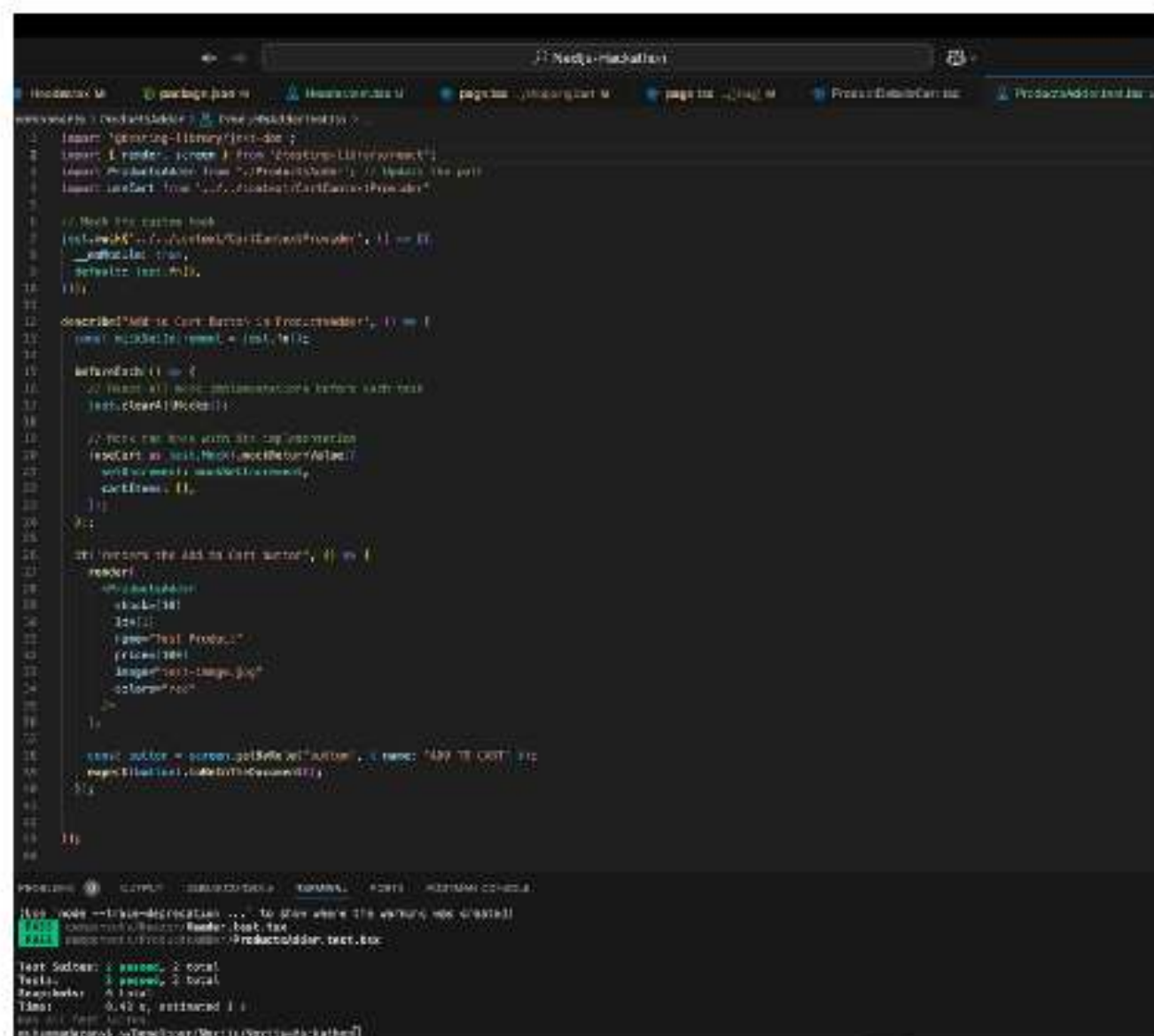


```
import { render, screen } from '@testing-library/react'
import { describe, it, expect } from '@jest/globals'
import { useRouter } from 'next/router'
import { useRouterMock } from '../utils/routerMock'

describe('useRouter', () => {
  it('returns a router object', () => {
    const router = useRouter()
    expect(router).toBeDefined()
    expect(router).toHaveProperty('push')
    expect(router).toHaveProperty('replace')
    expect(router).toHaveProperty('back')
    expect(router).toHaveProperty('forward')
    expect(router).toHaveProperty('reload')
    expect(router).toHaveProperty('pathname')
    expect(router).toHaveProperty('query')
    expect(router).toHaveProperty('as')
    expect(router).toHaveProperty('href')
    expect(router).toHaveProperty('isReady')
    expect(router).toHaveProperty('ready')
    expect(router).toHaveProperty('route')
    expect(router).toHaveProperty('url')
    expect(router).toHaveProperty('pathname')
    expect(router).toHaveProperty('query')
    expect(router).toHaveProperty('as')
    expect(router).toHaveProperty('href')
    expect(router).toHaveProperty('isReady')
    expect(router).toHaveProperty('ready')
    expect(router).toHaveProperty('route')
    expect(router).toHaveProperty('url')
  })
})
```

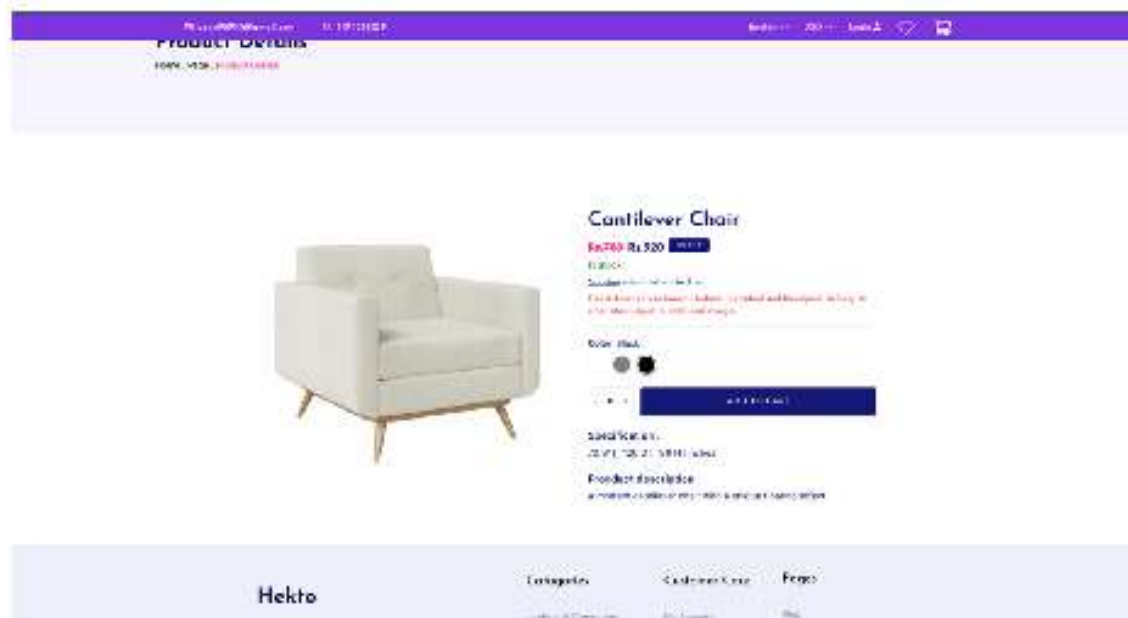


```
Test Suites: 1 passed, 1 total
Tests: 1 passed, 1 total
Snapshots: 0 total
Time: 0.101 s
Ran all test suites.
```



1.5 Dynamic Routing :

- Navigation to individual product detail pages
- Ensured product details (name, image, price, and description) are displayed correctly



1.6 Testing tools used :

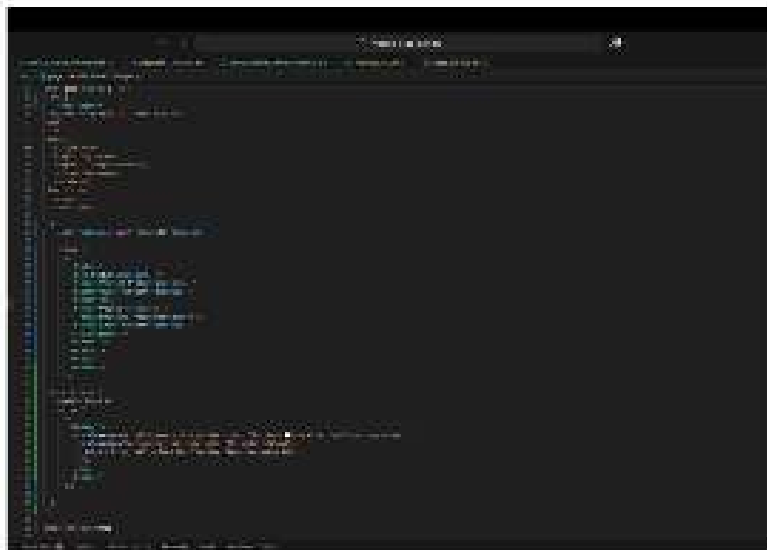
Feature	Tool Used
API Testing	Postman
Component Testing	React Testing Library
Manual Testing	Browser Navigation

Summary of Test Execution

Feature	Test Method
Product Listing Page	Manual Testing
Product Details Page	Manual Testing
Add to Cart Button	React Testing Library
Remove from Cart Button	React Testing Library
Add to Wishlist Button	React Testing Library
Remove from Wishlist Button	React Testing Library
Navigation (Dynamic Pages)	Manual Testing

2. Add Error Handling :

- implementing the error handling and fallback UI, the component will either render the fetched product data or show an error message if the fetch fails.
- The structure of the component includes:
- **Header**: Displays the header section.
- **Product Components**: Renders components such as Hero, FeaturedProduct, LatestProduct, etc., passing the fetched response data as props.
- **Footer**: Displays the footer of the page.

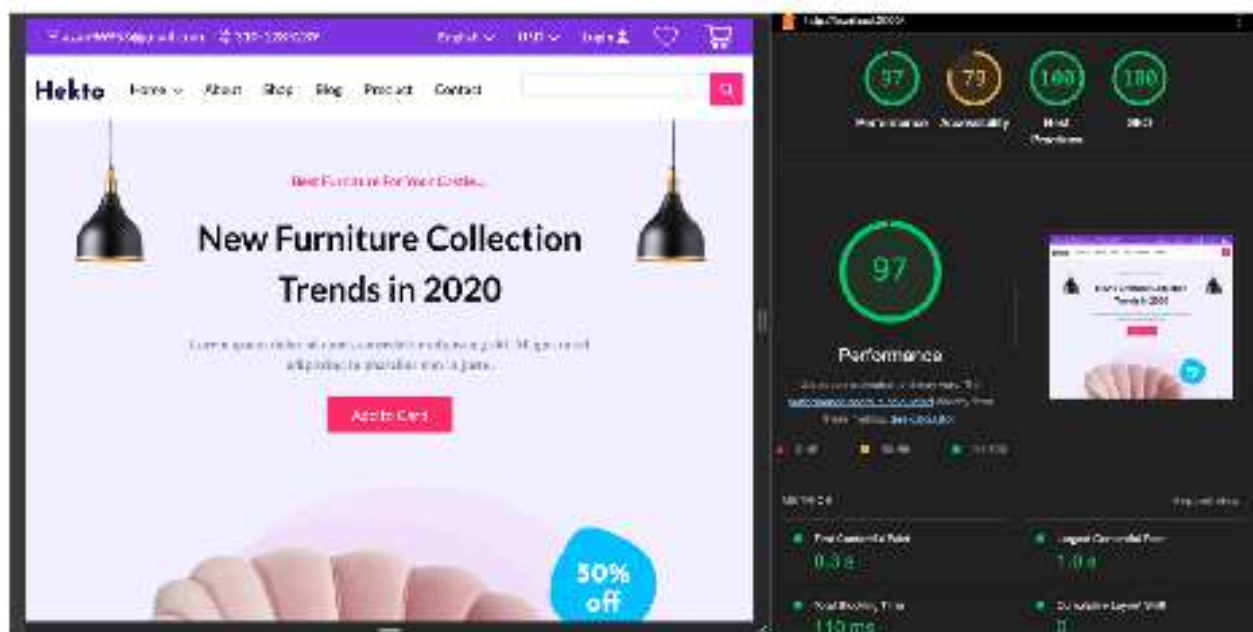


3.Performance Optimization :

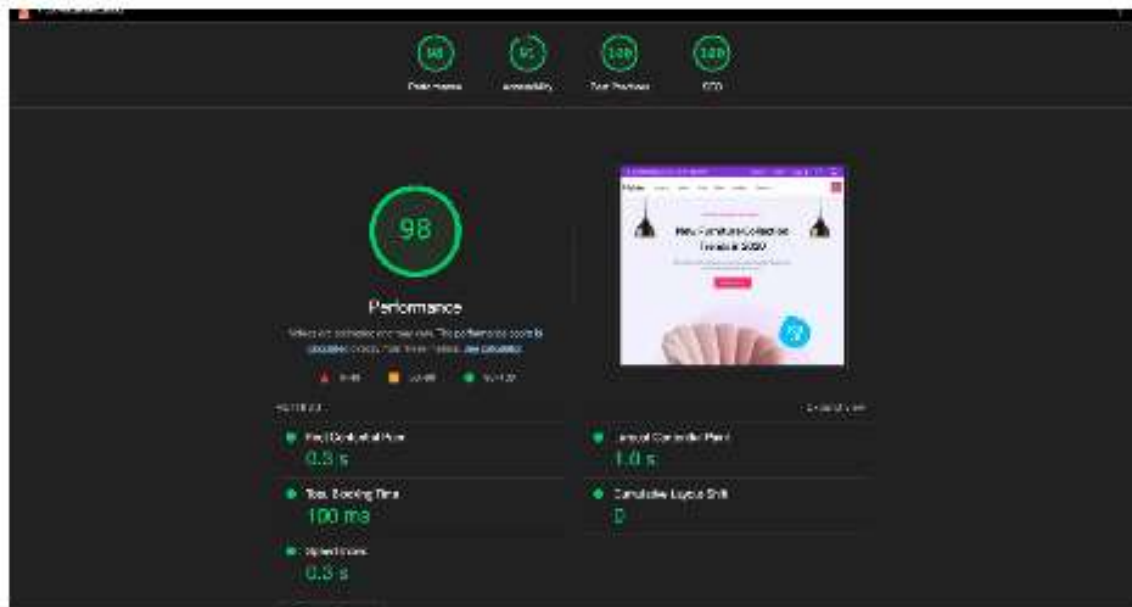
3.1. Optimize Assets

- **Image Compression**: Used tools like **TinyPNG** and **ImageOptim** to reduce image size without quality loss.
 - **Lazy Loading**: Implemented **lazy loading** for large images and assets to improve load times and reduce initial page weight.
-

3.3. Performance Scores (Before Optimization) :



3.4. Performance Scores (After Optimization) :



Metric	Score
Performance	98
Accessibility	91
Best Practices	100
SEO	100

4. Cross-Browser and Device Testing :

4.1. Browser Testing

- Tested on **Chrome, Firefox, Safari, and Edge** to ensure consistent rendering and

5 User Acceptance Testing (UAT) :

5.1 Document Version and Date:

- Version: 1.0
- [30 - jan - 2025]

5.2 Introduction :

- **Purpose:** To verify the functionality, usability, and overall user experience of the [E Commerce] marketplace before deployment.
- **Scope:** This document covers all major functionality including browsing products, adding/removing items to/from the cart, checking out, wishlist management, and responsive layout.

5.3 UAT Objectives :

Outline the key objectives for UAT:

- Ensure core features work as expected.
- Validate that APIs are functioning and fallback mechanisms are in place.
- Check for usability issues on different devices.
- Ensure that all user interactions (cart, wishlist, product detail pages) are seamless and user-friendly.

Based on (UAT) test I have created a CSV report :

Test Case ID	Test Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Remarks
T001	Shopping Product View Page	Click on item thumbnail > View Product	Product displayed correctly	Product displayed correctly	PASS	LOW	POSSIBLE PASS
T002	Shopping Cart Functionality	Click on item thumbnail > Add to Cart > View Cart > Remove item from the cart	Products added to cart successfully	Items as expected - null	PASS	LOW	NO ISSUE FOUND
T003	Shopping Wishlist Functionality	Click on item thumbnail > Add to Cart > Remove item from the cart > Add to Wishlist	Products added to wishlist successfully	Items as expected - null	PASS	LOW	NO ISSUE FOUND
T004	Adding items to different baskets	Click on item thumbnail > Add to Cart > Remove item from the cart > Add to Wishlist	Products added to cart successfully	Items as expected - null	PASS	LOW	NO ISSUE FOUND
T005	Shopping wishlist items	Click on item thumbnail > Add to Cart > Remove item from the cart > Add to Wishlist	Products added to cart successfully	Items as expected - null	PASS	LOW	NO ISSUE FOUND
T006	Shopping Product detail page	Click on item thumbnail > View Product > View Product Detail Page	Product displayed correctly	Product displayed correctly	PASS	LOW	NO ISSUE FOUND