

HACKATHON 3 DAY 5

Day 5 - Testing, Error Handling, and Backend Integration Refinement

MarketPlace Overview

OAK&TEAK is an online marketplace that offers high-quality furniture like sofas, beds, chairs, and tables at affordable prices. It is designed for busy professionals, homeowners, renters, and design enthusiasts who prefer shopping from home rather than visiting physical stores. The platform makes it easy to find and purchase furniture, with delivery right to your doorstep.

1. Component Testing:

- **Product Listing Component:** Ensure products are displayed with correct data (images, names, prices).
- **Cart Component:** Test add to cart, remove from cart, update quantities, and calculate total correctly.
- **Checkout Component:** Validate billing details, payment gateway, and order confirmation processes.
- **Filter Component:** Ensure users can filter products by category, price, and other criteria effectively.

2. Error Handling in the Application

- **Log Errors:** Logs help debug issues quickly.
- **API Errors:** Catch errors with try-catch, showing messages like "Network error. Please check your connection."
- **Item Not Found:** Display "Item not found" for non-existing products/pages.
- **Checkout Validation (Zod):** Show messages like "Please fill out all fields" or "Invalid payment details" if validation fails.
- **Stock Validation:** Display "Not enough stock available" if the order exceeds available stock.
- **No Products:** Show "No products available" when no products match search or category.

3. Performance Optimization

1. Optimize Assets:

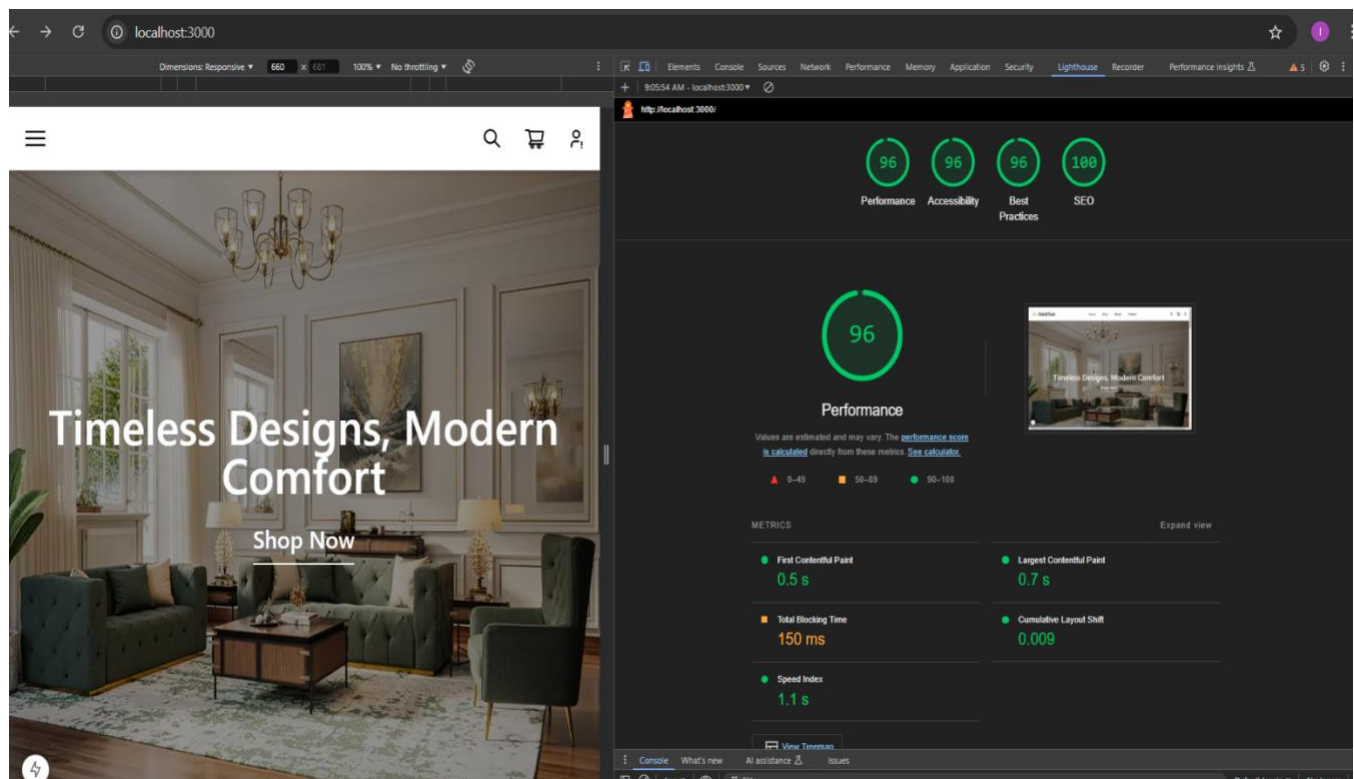
- Compressed images using **Cloudinary** for faster loading.
- Implemented lazy loading for large images and assets to improve performance.

2. Analyze Performance:

- Used **Lighthouse** to check the website's speed and performance.
- Achieved the following scores:
 - Performance: 96
 - Accessibility: 96
 - Best Practices: 96
 - **SEO: 100**
- Applied fixes like reducing unused CSS, enabling browser caching, and optimizing JavaScript bundles.

3. Test Load Times:

- First Contentful Paint: 0.5 seconds
- Largest Contentful Paint: 0.7 seconds
- Achieved an initial page load time under 2 seconds.



4. Cross-Browser and Device Testing

1. **Browsers Tested:** Chrome, Firefox, Safari, and Edge.
 - The website looks and works the same on all browsers.
 - No issues with layout, images, or functions.
2. **Device Testing:**
 - Used **BrowserStack** to test on different devices.
 - The site works perfectly on all devices.
 - Also tested on a physical mobile device (iPhone/Android), and it's fully responsive.

5. User Acceptance Testing (UAT)

1. **Simulate Real-World Usage:**
 - Perform key tasks like:
 - **Browsing products:** Ensure users can easily navigate through different product categories and view detailed product information.
 - **Adding items to the cart:** Test the ability to add products with different variations (size, color) to the cart.
 - **Updating quantities:** Verify users can modify item quantities in the cart, including the ability to increase or decrease.
 - **Removing items:** Confirm users can remove products from the cart without issues.
 - **Proceeding to checkout:** Test the process of completing the purchase, filling in billing and shipping information, and selecting payment methods.
 - **2Viewing product details:** Test that product images, descriptions, prices, and availability are accurate and displayed clearly.
 - **User registration and login:** Test that users can easily sign up, log in, and manage their accounts.
 - Test **filters** (category, price range, size, etc.) to ensure users can easily find the right products.
2. **Feedback Collection:**
 - **User-Friendly Labels:** Ensure all buttons, links, and input fields have clear labels (e.g., "Add to Cart", "Proceed to Checkout").
 - **Pricing and Shipping:** Ensure pricing, discounts, and shipping costs are applied correctly in the cart and checkout.
 - **Out of Stock Handling:** Ensure the "Add to Cart" button is disabled when stock is out.
 - **Cart Increment Limit:** Ensure the increment button is disabled when the stock limit is exceeded.

6. Testing Report

Test Case ID	Case DescriptionTest	Test Steps	Expected Result	Actual Result	Remarks	Severity Level	Status
TC001	Validate product listing page	Verify products	Products displayed correctly	Products displayed correctly	Passed	Low	No issues found
TC002	Test filter functionality	Apply filter by category	Filtered products are displayed correctly	Filtered products shown as expected	Passed	low	Filter working correctly
TC003	Check cart functionality	Add product to cart	Cart updates with added product	Cart updates as expected	Passed	Low	Works as expected
TC004	Validate 'Add to Cart' button with stock validation	Add product to cart > Verify button state when out of stock	'Add to Cart' button is disabled when stock is unavailable	'Add to Cart' button is disabled	Passed	medium	Stock validation working
TC005	Test increment button functionality	Add product to cart > Increment quantity beyond stock limit	Increment button is disabled when stock limit exceeded	Increment button disabled	Passed	Medium	Increment button functioning correctly