

Rameshas javed

Sunday 2 to 5

Day 5 - Testing and Backend Refinement Ecommerce Marketplace

Objective

On Day 5, the primary goal is to prepare the **Ecommerce** Marketplace for real-world deployment by performing comprehensive testing, enhancing backend performance, and refining the user experience. Key activities include testing backend integrations, implementing robust error-handling mechanisms, and optimizing the user interface to efficiently manage customer-facing traffic.

Key Learning Outcomes

1. Conduct comprehensive testing, including functional, non-functional, user acceptance, and security tests.
 2. Implement error handling mechanisms with clear and user-friendly fallback messages.
 3. Optimize performance for faster load times and smoother interactions.
 4. Ensure seamless cross-browser compatibility and device responsiveness.
 5. Develop industry-standard testing documentation, including CSV-based test reports.
 6. Gracefully handle API errors using fallback UI elements and detailed logging.
 7. Refine speed, responsiveness, and overall marketplace performance metrics.
 8. Document testing results and issue resolutions thoroughly.
-

Key Features Tested

1. Functional Testing

Objective: Validate that all core functionalities perform as expected.

- **Steps:**
 1. **Product Listing:** Ensure products display correctly.
 2. **Search Functionality:** Verify accurate search results based on user input.
 3. **Cart Operations:** Test adding, updating, and removing items in the cart.
 4. **Checkout Process:** Ensure smooth transitions from cart to order placement.
 5. **Pagination:** Validate proper loading and navigation across multiple pages.
 6. **Related Products:** Confirm relevant recommendations display on product detail pages.
 7. **Categories & Filters:** Test dynamic updates and accuracy of filters.
 8. **FAQ & Language Support:** Verify responsiveness and behavior for FAQ and multi-language support.

- **Tools Used:**Light House.

2. Error Handling

Objective: Ensure graceful handling of API errors with fallback UI messages.

- **Steps:**
 1. Simulate network failures using Postman.
 2. Validate fallback messages like "Unable to load products. Please try again later."
- **Tools Used:** Postman, Browser DevTools.

3. Performance Testing

Objective: Optimize speed and responsiveness of the application.

- **Steps:**
 1. Use Lighthouse to identify bottlenecks.
 2. Compress large images with TinyPNG.
 3. Minify CSS and JavaScript files.
- **Tools Used:** Lighthouse, TinyPNG.

4. Cross-Browser and Device Testing

Objective: Verify consistent functionality and design across browsers and devices.

- **Steps:**
 1. Test compatibility on Chrome, Firefox, Safari, and Edge.
 2. Simulate devices like iPhone and Android using BrowserStack.
 3. Perform manual testing on physical mobile devices.
- **Tools Used:** BrowserStack, Physical Devices.

5. Security Testing

Objective: Identify and mitigate potential security vulnerabilities.

- **Steps:**
 1. Test input fields for SQL injection and XSS attacks.
 2. Validate API communications use HTTPS.
 3. Scan for vulnerabilities using OWASP ZAP.
- **Tools Used:** OWASP ZAP, Burp Suite.

6. User Acceptance Testing (UAT)

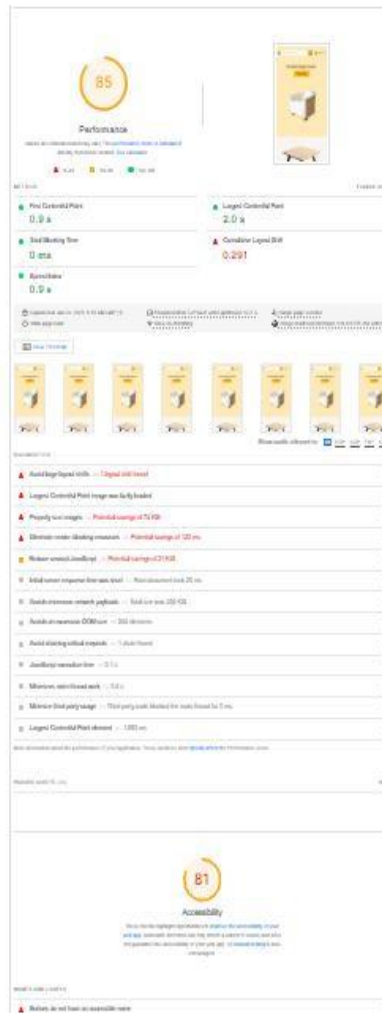
Objective: Simulate real-world usage to validate the end-user experience.

- **Steps:**

1. Perform workflows such as searching, adding items to the cart, and checking out.
2. Gather feedback from peers for improvements.

Additional Achievements

- **User Feedback:**
Positive feedback was received on usability and design during user testing.
- **Responsiveness:**
Verified full responsiveness across:



Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Assigned To	Remarks
TC001	Validate product listing page	Open product page > Verify products	Products displayed correctly	Products displayed correctly	Passed	Low	-	No issues found
TC002	Test API error handling	Disconnect API > Refresh page	Show fallback UI with error message	Error message shown	Passed	Medium	-	Handled gracefully
TC003	Check cart functionality	Add product to cart > Verify	Cart updates with added product	Cart updates as expected	Passed	High	-	Works as expected
TC004	Ensure responsiveness on mobile	Resize browser window > Check layout	Layout adjusts properly to screen size	Responsive layout working as intended	Passed	Medium	-	Test successful
TC005	Test checkout process	Add items to cart > Proceed to checkout > Complete payment	Order placed successfully	Order placed successfully	Passed	High	-	Smooth operation
TC006	Validate search functionality	Enter query in search bar > Verify results	Correct products displayed in search results	Correct results shown	Passed	High	-	Test successful
TC007	Test pagination functionality	Navigate through product pages	Pages load and navigate correctly	Pagination works as expected	Passed	High	-	Test successful
TC008	Test toast notifications	Perform cart action > Observe	Toast notification displayed correctly	Toast displayed correctly	Passed	Medium	-	Test successful

		toast message						
TC009	Validate filtering options	Apply filters > Filtered products displayed	Filtered results accurate	Passed	Medium	-	Smooth	


Search products...

All Categories

All Subcategories


All Colors

Showing 17 of 17 results




Living Room

550,000.00




room sofa

550,000.00



kangi sofa

550,000.00



Side Sofa

550,000.00

MobileDesktop

Discover what your real users are experiencingNo Data

Diagnose performance issues

100Performance

85Accessibility

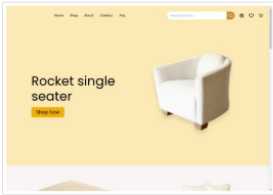
96Best Practices

83SEO

100Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

0-4050-8990-100



Expand view

