

## **Marketplace Testing Report:**

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

#### **Functional testing:**

##### **1. Product Listing**

- Verify that all products are displayed with correct details (name, image, price, and description).
  - Ensure pagination or infinite scrolling works as expected.
  - Test sorting and filtering functionality (e.g., by price, category, or popularity).
- 

##### **2. Product Details Page**

- Check that clicking on a product opens its detailed page.
  - Validate that the product details (e.g., images, price, availability) are correct.
  - Test the "Add to Cart" button functionality.
- 

##### **3. Shopping Cart**

- Ensure products can be added, removed, and updated (e.g., quantity adjustment) in the cart.
  - Verify the total price calculation, including discounts, taxes, and shipping fees.
  - Check if the cart persists after page reload or user login.
- 

##### **4. Checkout Process**

- Test that users can complete a purchase successfully.
    - Validate payment gateway integration.
    - Test order confirmation email generation.
  - Verify address entry or selection (if applicable).
  - Test promo code or discount functionality.
- 

##### **5. User Authentication**

- Ensure users can register, log in, and log out successfully.
  - Verify password reset functionality via email.
  - Test login persistence across sessions and devices.
- 

## 7. Responsiveness

- Verify that the website adjusts seamlessly across different screen sizes (mobile, tablet, desktop).
  - Ensure that critical actions (like adding to the cart or checkout) are accessible and functional on all devices.
- 

## 8. Error Handling

- Test error messages for invalid or missing inputs in forms (e.g., invalid email, missing fields).
  - Simulate API failures and check that fallback UIs (e.g., "Products not available") are displayed correctly.
  - Validate that unauthorized actions (like accessing admin features) are blocked.
- 

### Error Handling:

#### 1. Frontend Errors

- **Input Validation:** Show user-friendly error messages for invalid inputs (e.g., email format or missing fields).
- **Search Failures:** Display *"No products found"* when no results match.
- **Cart Errors:** Handle failures like *"Unable to add item to cart. Try again."*
- **Network Issues:** Show fallback UI like *"Failed to load products. Check your internet."*

#### 2. Backend Errors

- **API Responses:** Use clear error codes (400, 401, 404, 500) and messages (e.g., *"Product not found."*).
- **Stock Handling:** Disable "Add to Cart" for out-of-stock items with a message.
- **Payment Failures:** Inform users of errors (e.g., *"Payment failed. Try again."*).

#### 3. Fallback UI

- Custom error pages (404, 500) with helpful navigation options.
- Placeholders like *"No reviews available"* for missing data.

#### 4. Monitoring

- Use tools like **Sentry** for frontend and **Winston** for backend error logging.

#### 5. Testing Scenarios

- Simulate API failures, invalid inputs, and payment errors during QA.

## **Cross-Browser**

1. **Browser Testing:**
  - Test on **Chrome** to ensure compatibility.
2. **Device Testing:**
  - Verify responsiveness on **desktop**, **tablet**, and **mobile** screens.
3. **Tools:**
  - Use **BrowserStack** to simulate different browsers and devices.
4. **Focus Areas:**
  - Check layouts, navigation, and interactivity on all screen sizes.
  - Ensure consistent fonts, images, and styles across devices.

## **Security Testing**

1. **Input Validation:**
  - Ensure all forms (e.g., login, signup, search) sanitize inputs to prevent
2. **Secure Communication:**
  - Use **HTTPS** to encrypt all data exchanges.
3. **API Key Safety:**
  - Store sensitive keys securely in environment variables, not in frontend code.

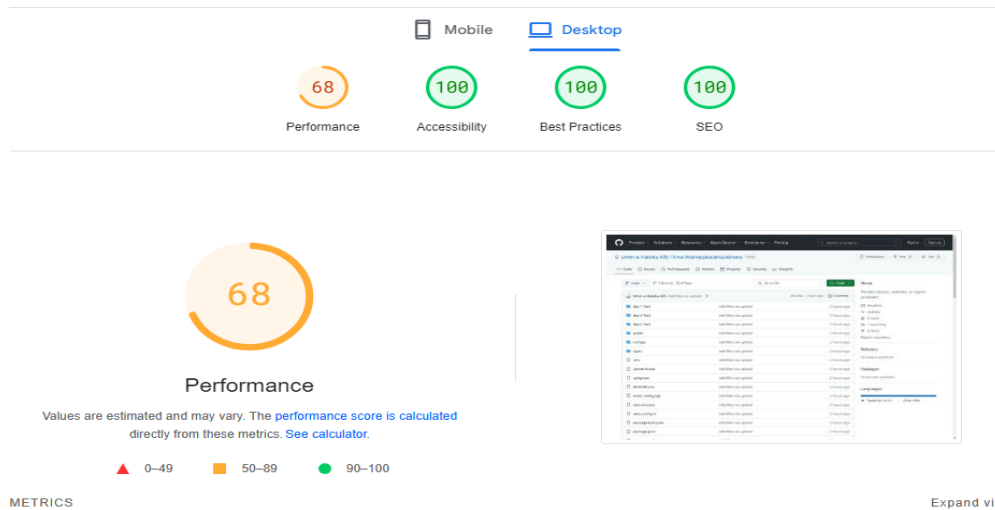
## **User Acceptance Testing (UAT)**

1. **Simulate User Scenarios:**
  - Test as a customer: browse products, search, add to cart, and checkout.
2. **Verify Usability:**
  - Ensure workflows are **intuitive** and navigation is **smooth**.
3. **Check for Errors:**
  - Confirm no issues like broken links, failed searches, or cart errors.

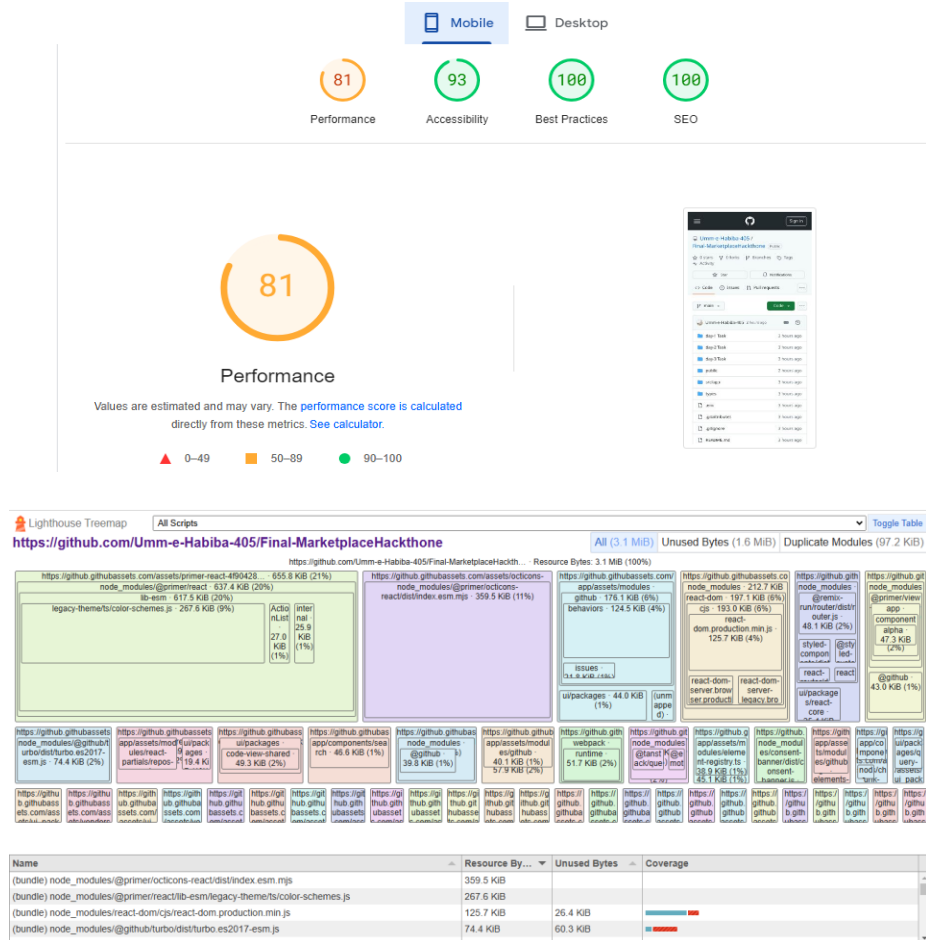
## **Performance Testing**

1. **Tool Used:**
  - Use **Google PageSpeed Insights** to analyze performance.
2. **Key Focus Areas:**
  - **Optimize Images:** Compress and use next-gen formats like WebP.
  - **Minimize Code:** Reduce JavaScript and CSS size.
  - **Caching:** Implement browser caching for faster load times.
3. **Actionable Insights:**
  - Address suggestions provided by PageSpeed, like reducing render-blocking resources and improving time-to-interactive.
4. **Goal:**
  - Achieve a **score of 90+** on desktop and mobile.

## **ON DESKTOP SCREEN**



**ON MOBILE SCREEN**



**Testing\_Report\_Sample.csv**

<b>TEST CASE ID</b>	<b>CASE DETAILS</b>	<b>CASE RESULT</b>	<b>CASE STATUS</b>	<b>CASE REMARKS</b>
<b>CASE 001</b>	<i>Product List</i>	<i>Sorted by product</i>	<i>Passed/Completed</i>	<i>Product listing works as expected.</i>
<b>CASE 002</b>	<i>Dynamic Routing</i>	<i>Routing done</i>	<i>Passed/Completed</i>	<i>Pages navigate correctly</i>
<b>CASE 003</b>	<i>Add to cart</i>	<i>Added in cart well</i>	<i>Passed/Completed</i>	<i>Items added to the cart without issues.</i>
<b>CASE 004</b>	<i>Shopping Cart</i>	<i>Product will be moved to the cart page</i>	<i>Passed/Completed</i>	<i>Cart updates displayed successfully.</i>
<b>CASE 005</b>	<i>Check Out</i>	<i>Smoothly works on the checkout page</i>	<i>Passed/Completed</i>	<i>Checkout process completed seamlessly.</i>
<b>CASE 006</b>	<i>Responsiveness</i>	<i>Responsive</i>	<i>Passed/Completed</i>	<i>Website adapts well to all screen sizes.</i>