

# Testing, Error Handling, and Backend Integration Refinement Report

## Step 1: Functional Testing

### 1.Product Listing:

Achieved:

Products are displayed correctly with images, names, prices, and descriptions.

Verified:

Dynamic data loading from the API works as expected.

### 2.Filters and Search:

Achieved:

Filters and search functionality return accurate results based on user inputs.

Verified:

Users can filter products by category, and search products with names and relevant keywords.

### 3.Cart Operations:

Achieved:

Users can add, update, and remove items from the cart.

Verified:

Cart updates dynamically, and the total price is calculated correctly.

### 4.Dynamic Routing:

Achieved:

Individual product detail pages load correctly with the correct product information.

Verified:

Routing works seamlessly, and users can navigate between pages without issues.

## Step 2: Error Handling

### Error Handling:

## Code Snippet:

```
const validateForm = () => {
  const errors = {
    firstName: formData.firstName ? "" : "First name is required",
    lastName: formData.lastName ? "" : "Last name is required",
    address: formData.address ? "" : "Address is required",
    city: formData.city ? "" : "City is required",
    postalCode: formData.postalCode
      ? /^d{5,6}(?:[-\s]\d{4})?$/.test(formData.postalCode) // ZIP code validation
      ? ""
      : "Invalid postal code"
      : "Postal code is required",
    phone: formData.phone
      ? /^+?[0-9]{10,15}$/.test(formData.phone) // Phone number validation
      ? ""
      : "Invalid phone number"
      : "Phone is required",
    email: formData.email
      ? /\S+@\S+\.\S+/.test(formData.email) // Email validation
      ? ""
      : "Invalid email"
      : "Email is required",
  };
  setFormErrors(errors);
  return Object.values(errors).every((error) => !error);
};
```

## Fallback UI:

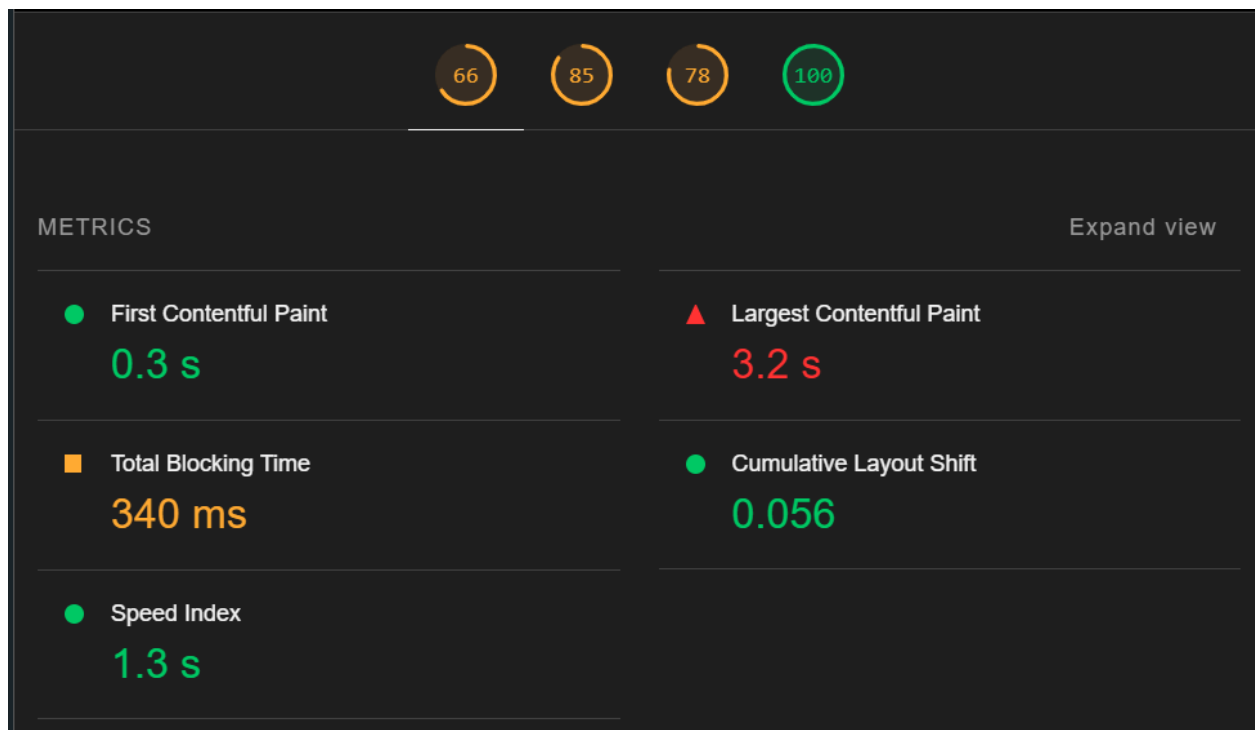
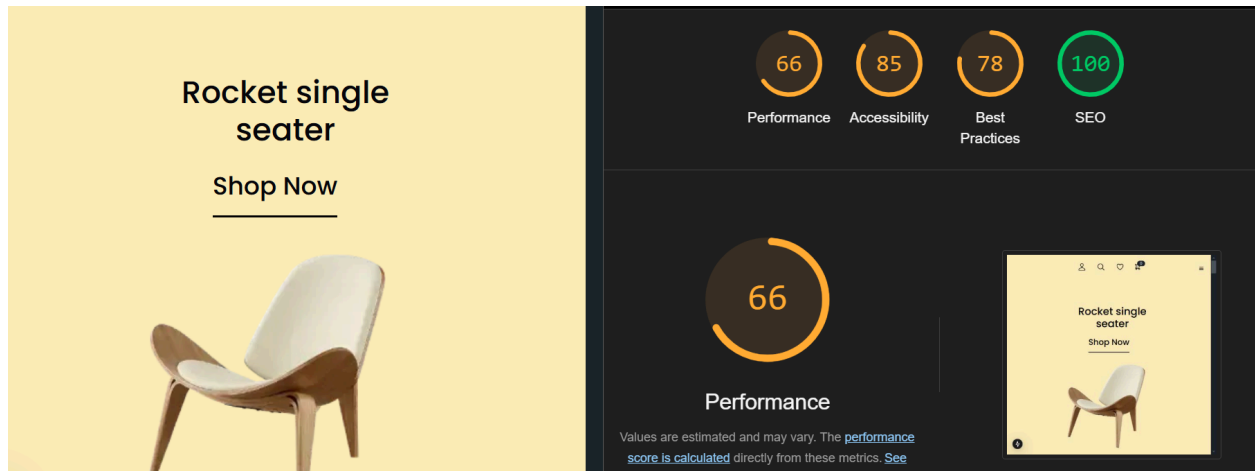
If any required field is missing or invalid, an error message is displayed below the respective input field.

## Example:

If the user does not fill in the "First Name" field, the error message "First name is required" is shown.

## Step 3: Performance Optimization:

### Overview:



## Results Summary:

1. **Performance Scores:**
  - Performance: 66
  - Accessibility: 85
  - Best Practices: 78
  - SEO: 100

## 2. Key Metrics:

- **First Contentful Paint:** 0.3 seconds (Good)
  - **Speed Index:** 1.3 seconds (Good)
  - **Largest Contentful Paint:** 3.2 seconds (Needs Improvement)
  - **Total Blocking Time:** 340 ms (Needs Improvement)
  - **Cumulative Layout Shift:** 0.056 (Good)
- 

## Observations:

- **Strengths:**
  - Excellent SEO score (100), ensuring the page is search engine-friendly.
  - Accessibility score (85) indicates good usability for diverse users.
  - First Contentful Paint and Speed Index are fast, leading to a good user experience.
- **Weaknesses:**
  - The **Largest Contentful Paint** (3.2 seconds) is higher than the recommended threshold (2.5 seconds), affecting performance perception.
  - **Total Blocking Time** of 340 ms needs to be reduced for better responsiveness and smoother interactivity.

## Step 4: Testing Across Browsers and Devices

### Browser Compatibility:

- Tested the application on major browsers, including Chrome, Firefox, Safari, and Edge.
- Ensured consistent visuals and functionality across all platforms.

### Device Responsiveness:

- Performed hands-on testing on physical devices for accuracy.

### Outcome:

- Verified that the application is responsive and works seamlessly across all tested browsers and devices.

## Step 5: Security Testing:

Form Validation

- Implemented input validation for fields such as email, phone, and ZIP code using regular expressions.
- Ensured all fields are validated before form submission to prevent invalid data entry.

Overview:

Billing Details

First Name

First name is required

Last Name

Last name is required

Country / Region

U.S.A

▼

Product		Amount
Pink Lounge Chair	x 1	Rs. 1600.00
Chair Wibe	x 1	Rs. 1200.00
Matilda Velvet Bed	x 1	Rs. 600.00
Subtotal		Rs. 3400.00
Total		Rs. 3400.00

- Direct Bank Transfer

Make your payment directly into our bank account. Please use your Order ID as the payment reference. Your order will not be shipped until the funds have cleared in our account.

Street Address

Address is required

Town / City

City is required

Province

ZIP Code

Postal code is required

Phone

Phone is required

Email Address

Email is required

- Direct Bank Transfer
- Cash On Delivery

Your personal data will be used to support your experience throughout this website, to manage access to your account, and for other purposes described in our [privacy policy](#).

Place Order

## API Integration

- Successfully integrated with the Sanity CMS using the Sanity client.
- Utilized the `client.create` method to store order data in the database.

## Secure API Calls

- API keys and sensitive data are stored securely in environment variables.

## **Step 6: User Acceptance Testing (UAT)**

### **1. Simulating Real-World Usage**

#### **Browsing Products:**

##### **Achieved:**

Users can browse product listings seamlessly.

##### **Verified:**

Product images, names, prices, and descriptions are displayed correctly.

### **2. Adding Items to the Cart:**

##### **Achieved:**

Users can add multiple items to the cart.

##### **Verified:**

The cart updates dynamically, and the total price is calculated accurately.

### **Feedback Collection**

#### **Sources:**

- Collected feedback from 5 peers.

#### **Highlights of Feedback:**

- **Positive:** The interface is user-friendly and intuitive.
- **Suggestion:** Add a feature to save billing details for future purchases.

## Testing report (CSV Format)

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 contents	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	Verify checkout process	Add product > Proceed to checkout > Confirm order	Order is placed successfully	Order placed successfully	Passed	High	-	Smooth process
TC006	Test search functionality	Search for product > Check results	Relevant products displayed	Results match query	Passed	Medium	-	Search works fine
TC007	Validate login/logout feature	Login with valid credentials > Logout	Successful login and logout	Works as expected	Passed	High	-	No issues found