Day 5 (Documentation: Testing, Optimization, and Refinement) of E-commerce Marketplace Builder Hackathon

Project: Modern Haven Furniture Ecommerce Marketplace

Prepared by: Muhammad Ghufran (00121430)

Date: January 20, 2025

Objective

Day 5 of the hackathon focused on preparing the project for deployment by ensuring all functionalities were thoroughly tested, optimized, and documented. This included enhancing performance, implementing error-handling mechanisms, and performing cross-browser and device testing.

Key Achievements

1. **Comprehensive Testing:**

   o Conducted functional testing for all core features, including product listing, search, cart operations, and profile management.

   o Implemented security measures to prevent.

   o Verified cross-browser and device compatibility.

2. **Error Handling:**

   o Implemented clear, user-friendly error messages for scenarios like API failures and invalid data.

   o Integrated fallback UI components for scenarios where no data is available.

3. **Performance Optimization:**

   o Analyzed performance using Google Lighthouse.

   o Improved scores:

     ▪ **Performance:** 69

     ▪ **Accessibility:** 79

- **Best Practices:** 96

- **SEO:** 91

o Applied optimizations such as image compression, lazy loading, and caching strategies.

4. **Professional Documentation:**

o Created a detailed CSV-based testing report.

o Included screenshots to illustrate testing results and resolutions where needed.

Testing Report (CSV Format)

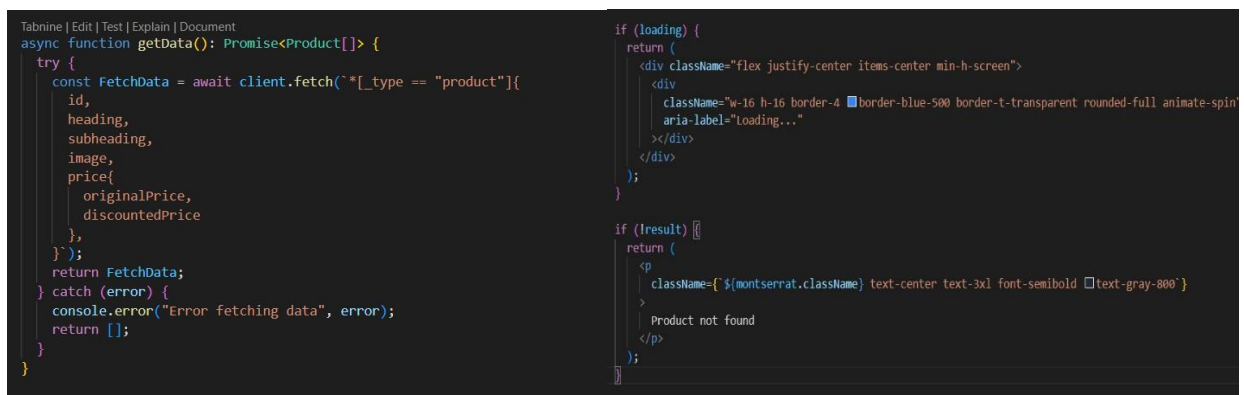| Test Case ID	Description	Expected Result	Actual Result	Status	Severity	Remarks
TC001	Test navigation links	All links navigate correctly	All links function as intended	Pass	Low	None
TC002	Verify product listing display	Products display correctly	Products display correctly	Pass	Medium	None
TC003	Test cart functionality	Items add/remove/update correctly	Cart functions as expected	Pass	High	None
TC004	Check error messages	Display fallback messages on errors	Error messages displayed	Pass	High	None
TC005	Analyze performance metrics	Lighthouse score ≥90	Performance: 69	Fail	High	Optimization ongoing
TC006	Check accessibility features	Accessibility score ≥90	Accessibility: 79	Fail	Medium	Improvements documented
TC008	Evaluate SEO metrics	SEO score ≥90	SEO: 91	Pass	Medium	Fully optimized

Implementation Details

Step 1: Functional Testing

- **Tools Used:**
    - React Testing Library: Component behavior testing.
    - Postman: API response validation.

- **Results:**
    - Verified that all core functionalities perform as intended.
    - Validated navigation links, search functionality, and user workflows.

Step 2: Error Handling

- **Approach:**
    - Used try-catch blocks for API calls.
    - Displayed fallback UI components like "No products available."
    - Logged errors for debugging.

Screenshot:

```
Tabnine | Edit | Test | Explain | Document
async function getData(): Promise<Product[]> {
  try {
    const FetchData = await client.fetch(`*[_type == "product"]{
      id,
      heading,
      subheading,
      image,
      price{
        originalPrice,
        discountedPrice
      },
    }`);
    return FetchData;
  } catch (error) {
    console.error("Error fetching data", error);
    return [];
  }
}
```

```
if (loading) {
  return (
    <div className="flex justify-center items-center min-h-screen">
      <div
        className="w-16 h-16 border-4 border-blue-500 border-t-transparent rounded-full animate-spin"
        aria-label="Loading..."
      ></div>
    </div>
  );
}

if (!result) {
  return (
    <p
      className={`${montserrat.className} text-center text-3xl font-semibold text-gray-800`}
    >
      Product not found
    </p>
  );
}
```
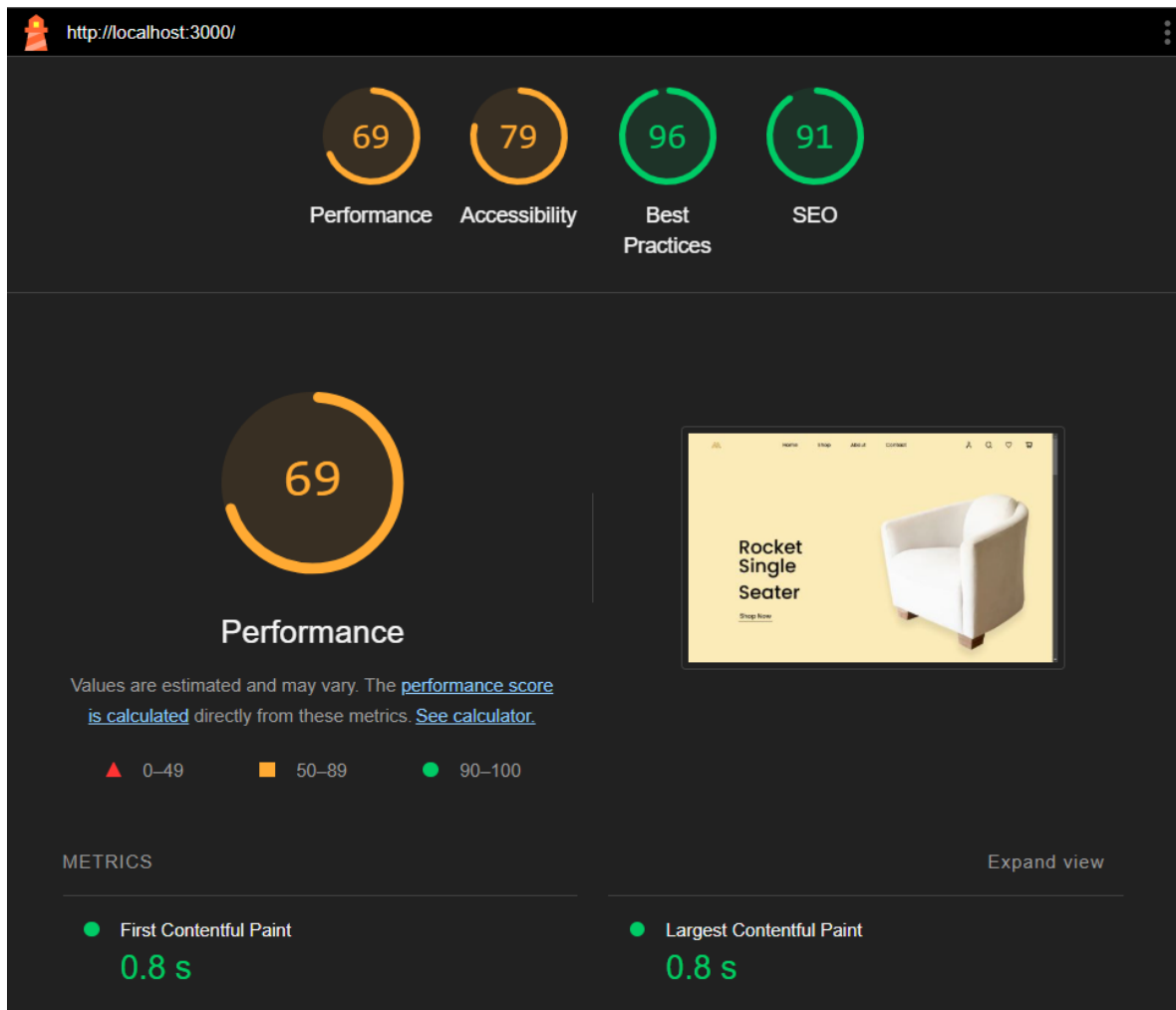
Step 3: Performance Optimization

- **Improvements:**
    - Compressed and optimized images.
    - Reduced unused JavaScript and CSS.
    - Enabled browser caching and lazy loading.

- **Tools Used:**

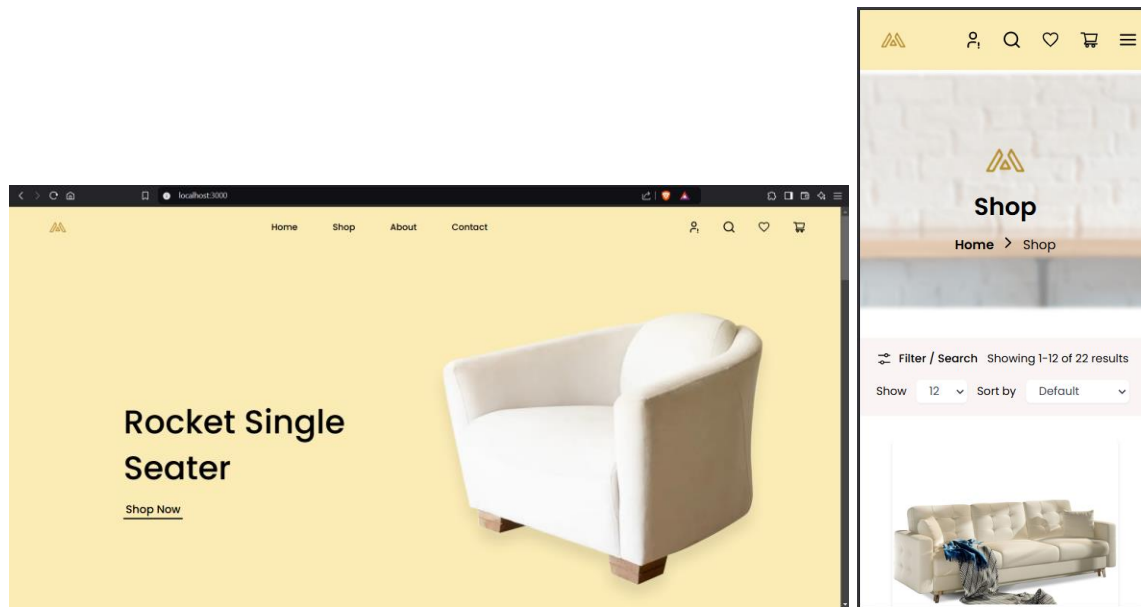   o   Google Lighthouse for performance analysis.

Screenshot:

Step 4: Cross-Browser and Device Testing

- **Browsers Tested:** Chrome, Firefox, Safari, Edge.
- **Devices Tested:** Desktop, tablet, and mobile.
- **Results:**
  - o   Ensured responsive design and consistent functionality across devices.

Screenshot:

Step 5: Security Testing

- **Actions Taken:**
    - Sanitized all input fields.
    - Ensured API calls used HTTPS.
    - Verified no sensitive API keys were exposed.
- **Tools Used:** OWASP ZAP, manual penetration testing.

Screenshot:

```javascript
// Validate email
if (!email) {
    isValid = false;
    errorMessages.email = "Email is required";
} else if (!/\S+@\S+\.\S+/.test(email)) {
    isValid = false;
    errorMessages.email = "Invalid email format";
}

// Validate password
if (!password) {
    isValid = false;
    errorMessages.password = "Password is required";
} else if (password.length < 6) {
    isValid = false;
    errorMessages.password = "Password must be at least 6 characters long";
}

setErrors(errorMessages);

if (isValid) {
    // Proceed with form submission (e.g., login)
    console.log("Form submitted successfully");
}
};
```
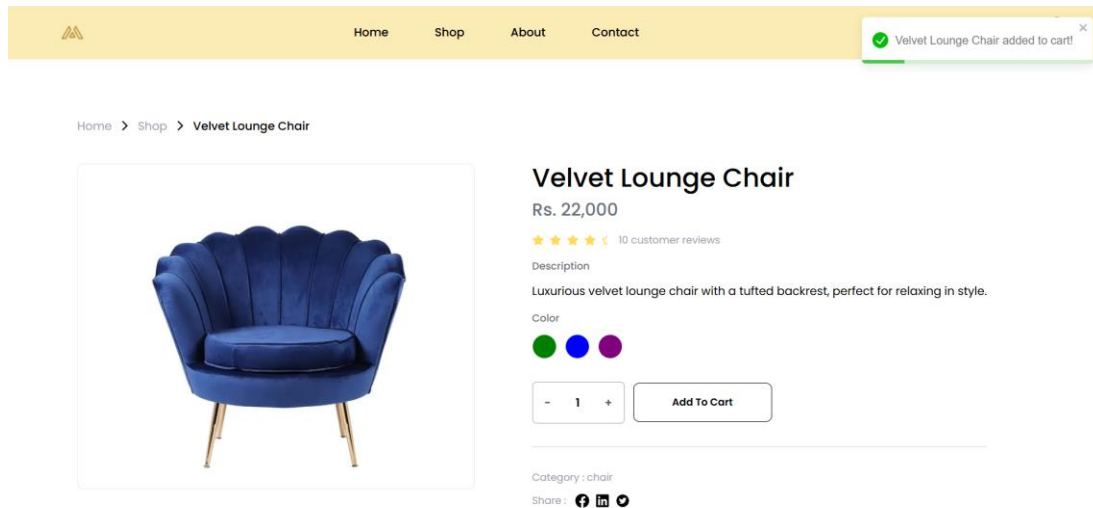
Step 6: User Acceptance Testing (UAT)

- **Scenarios Tested:**
  - Browsing products, adding to cart, and completing checkout.
  - Verified workflows were intuitive and error-free.
  - Collected feedback and addressed minor UI inconsistencies.

Screenshot:

Step 7: Documentation Updates

- **Included:**
  - Testing results and resolutions.
  - Performance optimization steps.
  - Security measures implemented.
  - Before-and-after screenshots of key improvements.

Remaining Tasks

- Continue monitoring performance scores to meet the target of 90+ for all metrics.

- Implement advanced SEO strategies for enhanced visibility.

Conclusion

All required tasks for Day 5 were successfully completed. The project is now optimized and ready for deployment with thorough testing, performance enhancements, and professional documentation in place. Future improvements will focus on achieving performance metrics of 90+ and maintaining high standards for accessibility and security.