Day 5: Testing Error Handling and Backend Integration Refinement

Objective:

Day 5 focuses on ensuring the **e-commerce marketplace** is deployment-ready by thoroughly testing its functionalities, optimizing performance, and documenting results professionally.

Key Areas Covered:

1. Conducting comprehensive testing of core functionalities.
2. Implementing robust error-handling mechanisms.
3. Optimizing for performance, accessibility, and SEO.
4. Ensuring cross-browser and cross-device compatibility.
5. Documenting findings and fixes in a structured format.

Key Learning Outcomes:

- Validate all functionalities through functional and user acceptance testing.
- Improve website performance metrics using tools like Lighthouse.
- Ensure high accessibility standards for users with disabilities.
- Enhance SEO for better search engine visibility.
- Prepare detailed documentation and a CSV-based testing report.

Step 1: Functional Testing

Description:

Validating the functionality of key components to ensure they work as expected.

Features Tested:

- **Navigation Links:** Verified that all links navigate correctly.
- **Product Listing and Details:** Ensured accurate rendering of product data.
- **Shopping Cart Operations:** Validated add, update, and remove functionalities.
- **Blog Accessibility:** Confirmed that blog content is accessible.
- **Contact Form:** Ensured successful form submissions.

Tools Used:

- **Postman:** API response testing.
- **React Testing Library:** Component behavior validation.
- **Cypress:** End-to-end UI testing.

Step 2: Error Handling

Description:

Implemented mechanisms to gracefully handle errors and provide user-friendly feedback.

Approach:

- **Utilized try-catch blocks to handle API errors.**
- **Displayed fallback UI elements, such as "No products available" when data is unavailable.**
- **Logged errors for debugging and future analysis.**
- **Ensured graceful handling of false API responses to maintain user trust and interface consistency.**

Step 3: Performance Optimization

Steps Taken:

Image Optimization:

- Implemented **lazy loading** for large assets.

Code Optimization:

- Minimized unused **CSS and JavaScript**.
- Enabled **code-splitting** for faster page load times.

Testing Tools:

- **Lighthouse:** Achieved a performance score of **92**.

Results:

- Faster initial page load.
- Improved user interaction time.

Step 4: Cross-Browser and Device Testing

Browsers Tested:

- **Google Chrome**
- **Mozilla Firefox**
- **Microsoft Edge**

Devices Tested:

- **Desktop**
- **Tablet**
- **Mobile devices**

Tools Used:

- **Manual testing on a physical mobile device.**

Results:

- **Verified consistent rendering and functionality across all platforms.**

Step 5: Security Testing

Measures Taken:

Input Validation:

- Used regular expressions for form validation (e.g., email and phone fields).

API Security:

- Ensured all API calls were made over HTTPS.
- Stored sensitive data (e.g., API keys) in environment variables.

Step 6: User Acceptance Testing (UAT)

Description:

Simulated real-world user interactions to identify usability issues.

Scenarios Tested:

- Browsing products.
- Adding and removing items from the cart.
- Completing the checkout process.
- Testing multi-step workflows to ensure an intuitive user experience.
- Adjusted visual hierarchy to emphasize key actions like "Add to Cart."

Testing report (csv format):

Test case id	description	Case result	Case status	Case remarks
Case 001	Add to cart functionality	Added in cart well	Pass completed	Functionality works as expected
Case 002	Category	Sort by category	Pass completed	Categories are

	functionality			display correctly
Case 003	Dynamic routing	Routing done	Pass completed	Dynamically routed
Case 004	Error handling		Fail completed	Errors are not handled gracefully
Case 005	Responsiveness	Responsive	Pass completed	Responsive across all devices