**Day 5: Testing, Error Handling, and Backend Integration Refinement.**

# Objective

Goals:

 Test all important features thoroughly.

 Handle errors gracefully.

 Improve performance, accessibility, and SEO.

 Make sure the app works on different browsers and devices.

 Document testing results in a clear format.

### Key Outcomes

1. Ensure all features work properly through testing.
2. Use tools like Lighthouse to boost performance.
3. Make the app accessible to everyone, including users with disabilities.
4. Optimize SEO for better visibility.
5. Document findings and create a detailed testing report.

### Steps to Achieve These Goals

#### **1. Functional Testing**

Check if key features are working as they should:

* **Navigation links:** Ensure links take users to the right pages.
* **Product listings:** Verify products are displayed correctly.
* **Cart functionality:** Test adding, updating, and removing items.
* **Contact form:** Make sure forms are submitted successfully.
* **Tools:** Postman (API), React Testing Library, Cypress.

### **2. Error Handling**

Make the app user-friendly during errors:

* Use **try-catch** for handling API issues.
* Show fallback messages like **"No products available"** when data is missing.
* Use **fallback images** when API images fail to load or are broken.
* Log errors for easier debugging and future fixes.

#### **3. Performance Optimization**

Use Lighthouse to improve speed and usability.

* Fix slow server response times.
* Minimize unused JavaScript.
* Add lazy loading for large images.

#### **4. Cross-Browser & Device Testing**

Test the app on Chrome, Firefox, Safari, and Edge using desktop, tablets, and mobiles. Ensure:

* Responsive design.
* Smooth navigation and interaction.
* Accessibility (screen readers, keyboard navigation).

#### **User Testing**

Simulate real user interactions to ensure ease of use:

* Browse products.
* Add and remove items from the cart.
* Test the checkout process.