

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

Submitted By: Nihal Naveed

Instructor: Sir Ameen Alam

1. Introduction

This document presents the **testing, error handling, and backend integration refinement** process for the marketplace project. The focus was on ensuring:

- ✓ **Functional testing** of core features.
 - ✓ **Error handling mechanisms** for API failures.
 - ✓ **Performance optimization** to improve speed and responsiveness.
 - ✓ **Cross-browser and device compatibility testing**.
 - ✓ **Security testing** to prevent vulnerabilities.
 - ✓ **User Acceptance Testing (UAT)** for real-world usability validation.
 - ✓ **Comprehensive documentation and test report (CSV format)**.
-

2. Completed Testing & Refinements

1. Functional Testing

- ✓ Verified **product listings, filters, cart, and checkout** functionality.
- ✓ Ensured smooth **API data fetching** and rendering.
- ✓ Tested dynamic routing for product pages.

2. Error Handling

- ✓ Implemented **try-catch blocks** for API calls.
- ✓ Displayed **user-friendly error messages** for network failures.
- ✓ Handled **empty states** (e.g., "No products found" when API fails).

3. Performance Optimization

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

4. Cross-Browser & Device Testing

- ✓ Tested on **Chrome, Firefox, Safari, and Edge**.
- ✓ Ensured **responsive UI** on mobile, tablet, and desktop screens.

5. Security Testing

- ✓ Stored **API keys** securely in **environment variables**.

6. User Acceptance Testing (UAT)

- ✓ Simulated **real-world shopping experience** (browsing, adding to cart, checkout).
 - ✓ Collected **peer feedback** for usability improvements.
-

3. Test Report (CSV Format)

 **CSV File:**

4. GitHub Repository

 **All files, including documentation and test reports, have been uploaded to the repository.**

 **GitHub Repository:** [Your Repo Link]

5. Challenges & Solutions

Challenge 1: API Delays & Failures

- ✓ **Solution:** Implemented **fallback UI** and **retry mechanisms** for failed requests.

Challenge 2: Mobile Responsiveness Issues

- ✓ **Solution:** Used **Tailwind CSS** for adaptive layouts.

Challenge 3: Security Vulnerabilities

✓ **Solution:** Enforced **input sanitization** and **secured API requests**.

Conclusion

This submission ensures the **marketplace is well-tested, optimized, and ready for real-world deployment**. The combination of **functional testing, security measures, and performance optimizations** makes it a scalable solution.

Nihal Naveed

nihalnaveed044@gmail.com

[Portfolio Link](#)