Hackathon Day 5

Testing, Error Handling, and Backend Integration Refinement Clothing Brand

Objective:~

Prepare the e-commerce clothing marketplace for deployment by thoroughly testing backend integrations, implementing error handling, and refining performance to handle customer-facing traffic.

Testing Report:~

This report summarizes the testing efforts for the e-commerce clothing marketplace, detailing test cases, results, and identified improvements.

Testing Report:

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 response	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

CVS Content:

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 response, "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

Key Learning Outcomes

❖ Comprehensive Testing:

- ➢ Functional testing ensured all marketplace features worked as intended.
- ➢ Non-functional testing verified performance, responsiveness, and error handling.
- ➢ Security testing protected against vulnerabilities like injection attacks and unsecured data.

❖ Error Handling Mechanisms:

- ❖ Implemented clear, user-friendly fallback messages for network errors and data unavailability.

Example:

```
try {
  const data = await fetchProducts();
  setProducts(data);
} catch (error) {
  console.error("Error fetching products:", error);
  setError("Unable to load products. Please try again later.");
}
```

Performance Optimization:

- Used tools like Lighthouse to identify bottlenecks.
- Compressed images and implemented lazy loading for faster page loads.
- Minimized unused CSS and JavaScript to reduce load times.

Cross-Browser and Device Compatibility:

- Tested on Chrome, Firefox, Safari, and Edge.
- Verified responsiveness using tools like BrowserStack.

Conclusion:

Day 5 focused on testing and refining the e-commerce clothing marketplace. All major features, like product listing, cart functionality, and API error handling, were tested successfully with no major issues. Error handling was implemented with clear fallback messages for better user experience. Performance optimizations, such as image compression and lazy loading, improved page load speed. The platform was also tested for cross-browser and device compatibility. Finally, comprehensive documentation was prepared, including a detailed testing report and updated GitHub repository. The marketplace is now ready for deployment, with improved functionality and performance.