Hackathon day 5

Testing, Error Handling, and Backend Integration Refinement

Furniture Brand

Objective

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

CVS Content

Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Assigned To	Remarks
TC001	Validate furniture listing page	"Open product page > Verify furniture items displayed"	"Furniture items displayed correctly"	"Furniture items 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	"Layout adjusts properly to	"Responsive layout	Passed	Medium	-	Test successful

Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Assigned To	Remarks
		window > Check layout"	screen size"	working as intended"				

Key Learning Outcomes

Comprehensive Testing

- **Functional Testing**: Ensured all marketplace features (e.g., product listing, cart functionality, and checkout) worked as intended.
- **Non-Functional Testing**: Verified performance, responsiveness, and error handling for a seamless shopping experience.
- **Security Testing**: Protected the platform against vulnerabilities such as injection attacks and unsecured data.

Error Handling Mechanisms

Clear, user-friendly fallback messages were implemented for network errors and data unavailability.

Example Code:

```
javascript
CopyEdit
try {
  const data = await fetchFurniture();
  setFurniture(data);
} catch (error) {
  console.error("Error fetching furniture items:", error);
  setError("Unable to load furniture. Please try again later.");
}
```

Performance Optimization

- **Tools Used**: Leveraged tools like Lighthouse to identify bottlenecks.
- **Image Optimization**: Compressed high-resolution furniture images and implemented lazy loading for faster page loads.
- **Code Optimization**: Minimized unused CSS and JavaScript to reduce load times and improve speed.

Cross-Browser and Device Compatibility

- **Browsers Tested**: Chrome, Firefox, Safari, and Edge.
- **Device Responsiveness**: Verified using tools like BrowserStack to ensure smooth performance across devices and screen sizes.

Conclusion

Day 5 focused on testing and refining the e-commerce furniture marketplace. All major features, including furniture listing, cart functionality, and API error handling, were successfully tested with no significant issues. Error handling was enhanced with user-friendly fallback messages to improve the customer experience. Performance optimizations, such as image compression and lazy loading, ensured faster page load speeds. Cross-browser and device compatibility were verified to deliver a consistent experience across platforms.

Comprehensive documentation, including a detailed testing report and updated GitHub repository, has been prepared. The furniture marketplace is now ready for deployment, with improved functionality and performance.