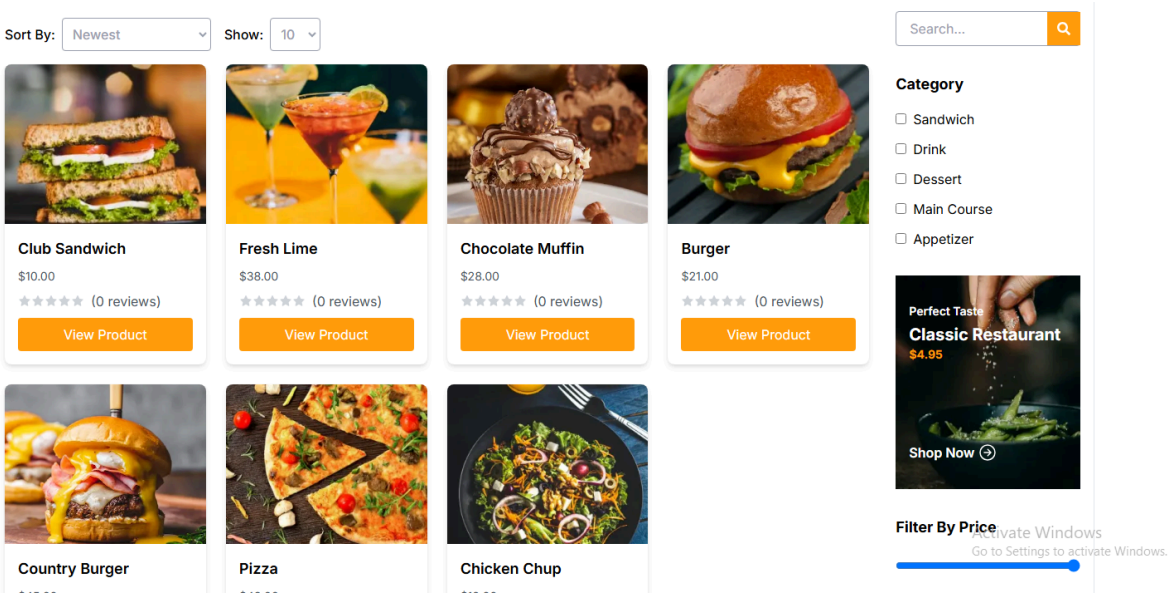
Day 5 - Testing and Backend Refinement - FoodTuck

Introduction

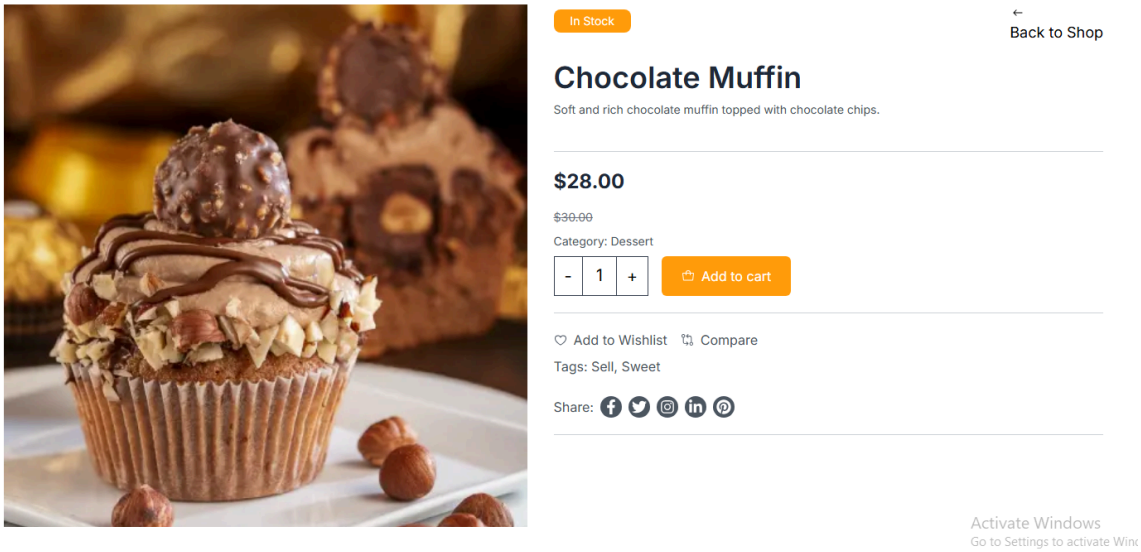
This submission document is prepared in accordance with the requirements outlined for Day 5: Testing, Error Handling, and Backend Integration Refinement. The focus is on comprehensive testing, performance optimization, error handling, and backend integration refinement.

Screenshots of Work

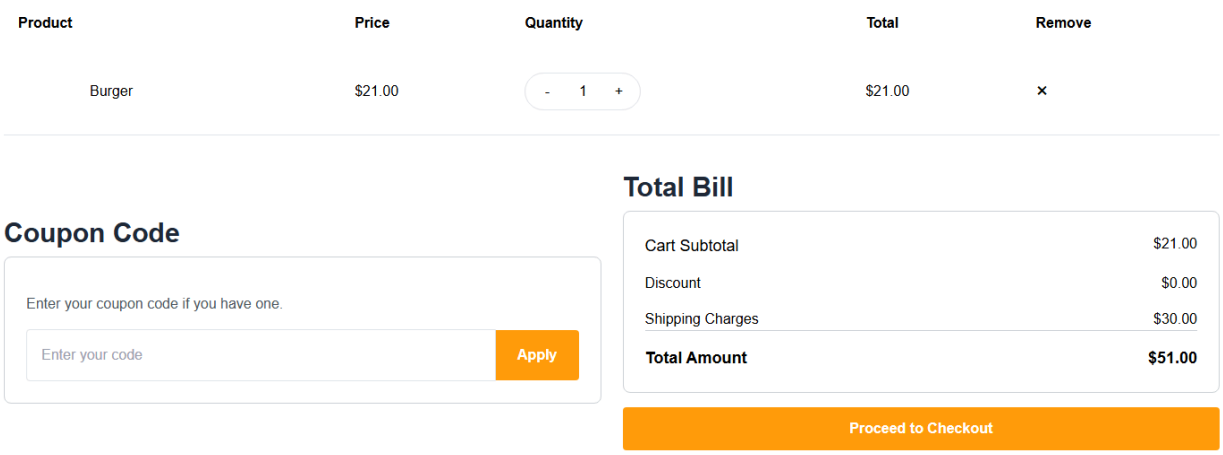
Include screenshots or images showcasing the following functionalities:



Product listing successfully



Product details page work perfectly.



Functional cart operations (add, update, and remove items).

Professional Report Detailing

Test Cases Executed and Results

Responsiveness Testing:

- Tested layout and components on various devices (mobile, tablet, desktop).
- Verified elements resize and reposition correctly across screen sizes.
- **Result:** All components passed responsiveness checks on multiple devices.

Cross-Browser Testing:

- Tested on Chrome, Firefox, Safari, and Edge.
- Validated consistent rendering and interactive functionality across browsers.
- **Result:** No major discrepancies observed.

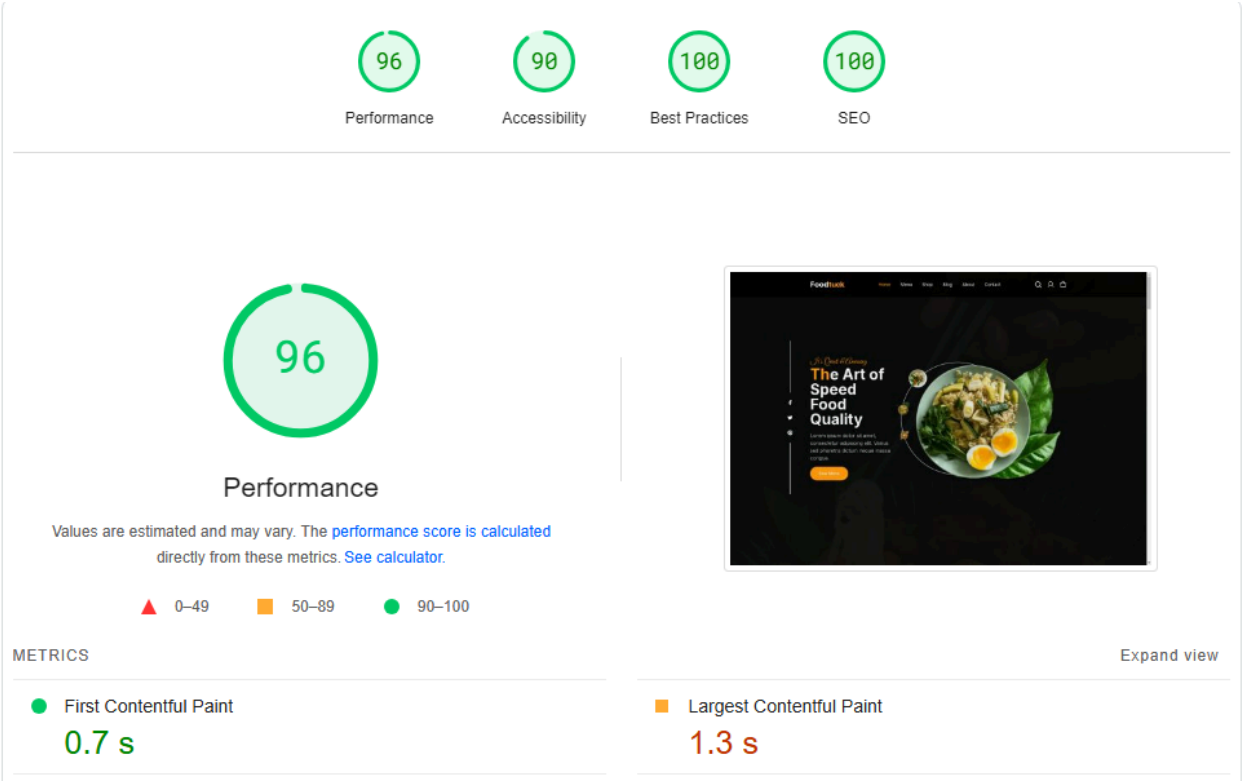
API Error Handling Testing:

- Simulated scenarios of network failure and incorrect data responses.
- Implemented graceful fallback UI with meaningful error messages.
- **Result:** All error states handled effectively.

Performance Optimization Steps

To enhance performance, the following optimization techniques were implemented:

- Compressed images using TinyPNG, reducing image sizes by up to 50%.
- Applied lazy loading for non-critical images to improve initial load time.
- Minimized unused CSS and JavaScript by tree-shaking during the build process.
- **Result:** Improved page load speed and get a score of 96 in performance.



Security Measures Implemented

Ensured secure operations by implementing the following measures:

- Validated input fields using regex to prevent injection attacks.
- Ensured all API communications were secured over HTTPS.
- Stored sensitive data like API keys in environment variables to avoid exposure.

Challenges Faced and Resolutions Applied

During development, the following challenges were encountered:

- **TypeScript Type Issues:** Faced issues with defining types for API responses and state management.
- **Resolution:** Resolved by carefully reviewing the TypeScript documentation and correctly implementing types for components and API responses.

Testing Report

Test C	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity	Remarks
TC001	Verify responsiveness on mobile devices	1. Open the application on a mobile device. 2. Check layout adjustments. 3. Interact with elements like buttons and menus.	Application layout adjusts correctly and elements are fully functional on mo	Passed: Layout and elements function as expected.	Passed	Medium	Tested on iPhone and Android devices.
TC002	Validate cross-browser compatibility	1. Open the application on Chrome, Firefox, Safari, and Edge.	All features work consistently across browsers.	Passed: No issues found on any browser.	Passed	High	Covered latest versions of each browser.
TC003	Simulate network failure for API error handlir	1. Disable network. 2. Perform operations like loading products or submitting	Fallback UI appears with user-friendly error messages.	Passed: Fallback UI and error messages displayed as expe	Passed	High	Handled with try-catch blocks.
TC004	Optimize performance using TinyPNG	1. Compress images with TinyPNG. 2. Measure page load times before and after optimization.	Page load time reduces significantly after image compression.	Passed: Performance score improved from 90 to 96.	Passed	Medium	Verified using Lighthouse.
TC005	Validate form input to prevent injection attac	1. Input invalid or malicious data in forms. 2. Submit the form and observe behavior.	Form validation prevents submission of invalid data.	Passed: Invalid inputs were successfully blocked.	Passed	High	Used regex for validation and sanitized inputs.
TC006	Test product detail page dynamic routing	1. Click on a product from the listing page. 2. Verify the correct product details load.	The correct product details page loads dynamically.	Passed: Verified with multiple product IDs.	Passed	Medium	Tested with valid and invalid product IDs.

Additional Notes

This document serves as the primary submission for Day 5 activities. Ensure all required files and reports are uploaded to the designated repository with clear folder structure and naming conventions.