**Functional Testing**

Product Listing:

All products are successfully listed on the marketplace homepage, with correct details such

title, image, price, and availability.

* Filters (category, price range, etc.) are functioning as expected and display appropriate products.
* Pagination works seamlessly, and infinite scrolling operates as intended on pages with many product

Product Detail Pages:

Product detail pages are displayed correctly with accurate information.

* Product images, descriptions, specifications, and pricing details are all correct and properly formatted.
* Variations (e.g., size, color) can be selected, and the "Add to Cart" button functions without issues.
* The “Add to Cart” button works as expected, and users are able to proceed smoothly with adding items to their cart.

Cart Operations:

All cart operations are fully functional, providing a smooth experience for users.

* Items are successfully added to the cart, and the cart updates automatically when items are added or removed.
* The system calculates the total price correctly based on selected products, including quantity adjustments.
* The user can update quantities and remove products from the cart without any errors.

Responsive Design:

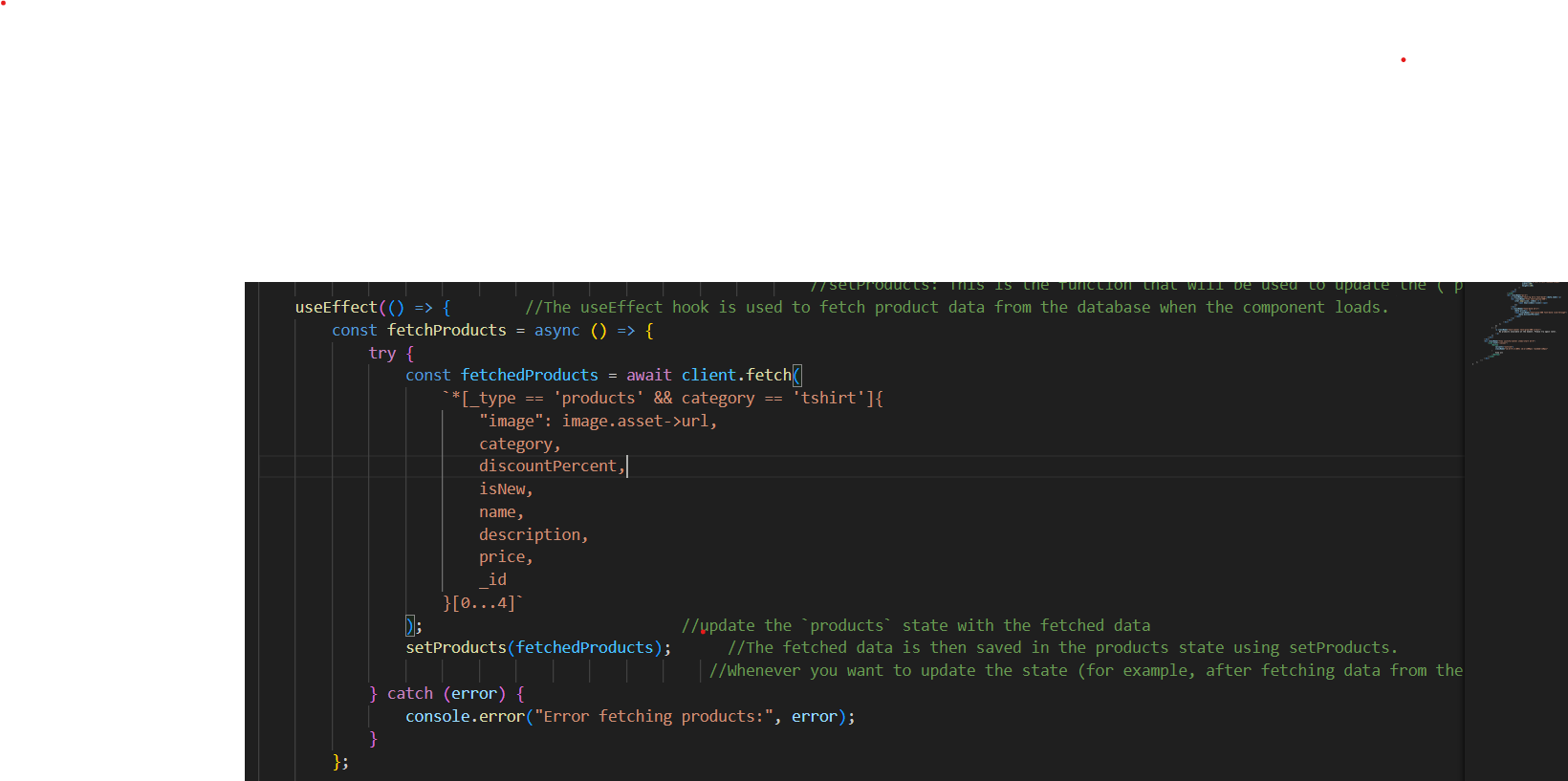
The website is fully responsive and provides an optimal viewing experience across a variety

devices, from desktop computers to tablets and smartphones.

* The layout adjusts dynamically based on screen size, ensuring that content is easily readable and accessible on all devices.
* On desktop screens, the navigation bar displays as a horizontal menu, while on mobile and tablet screens, it collapses into a hamburger menu for easy accesable.

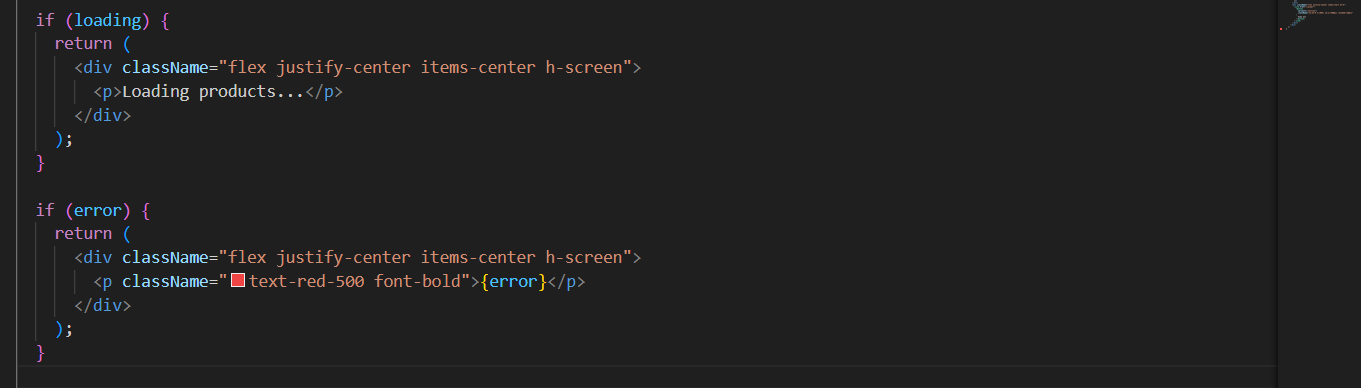
#### **Error Handling:**

"Implemented error handling with a try-catch block to manage any potential errors that may occur during the data retrieval process from the API. This approach ensures that failures are caught and appropriate error messages are logged for troubleshooting."



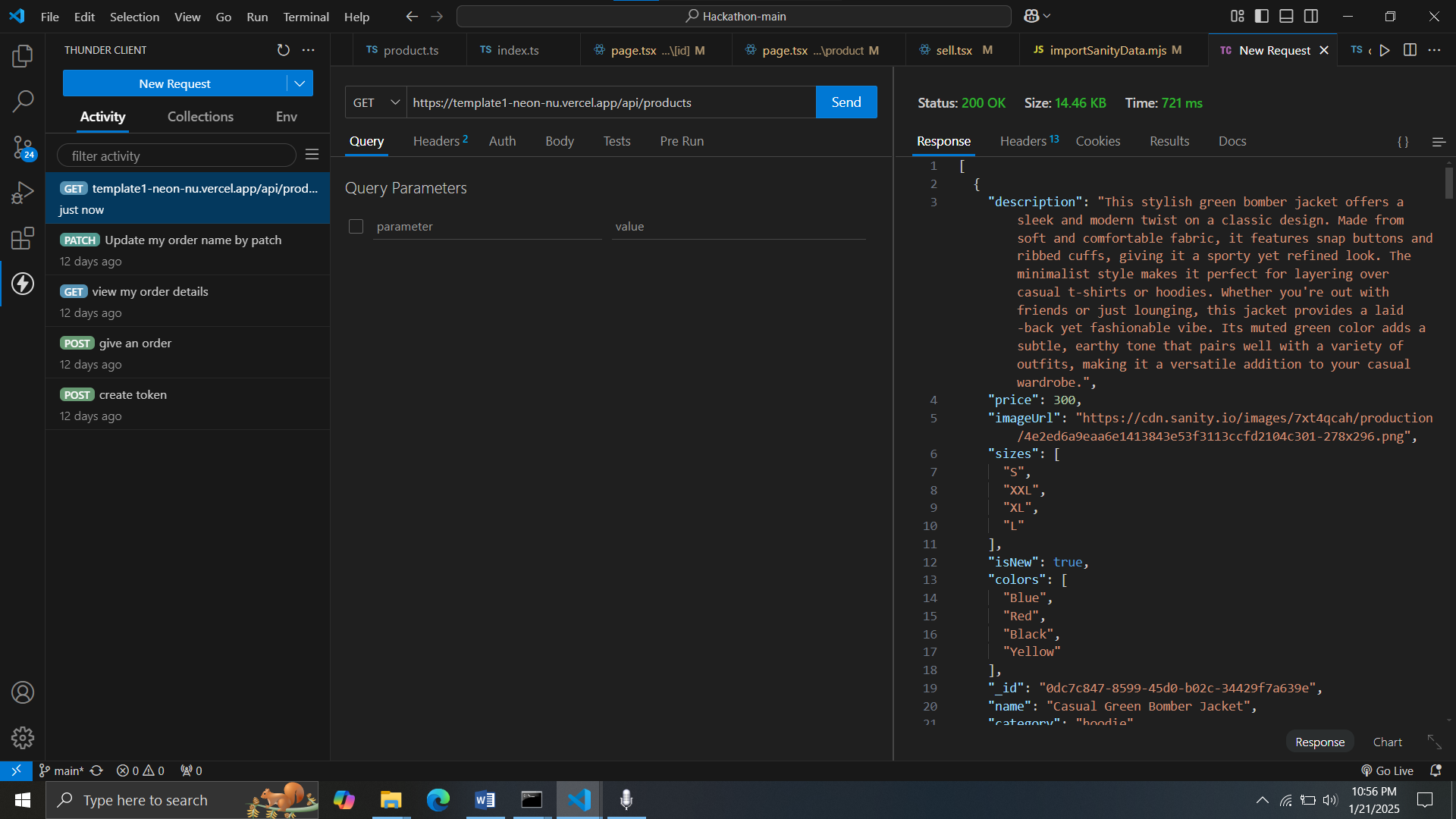
Fallback UI Elements:

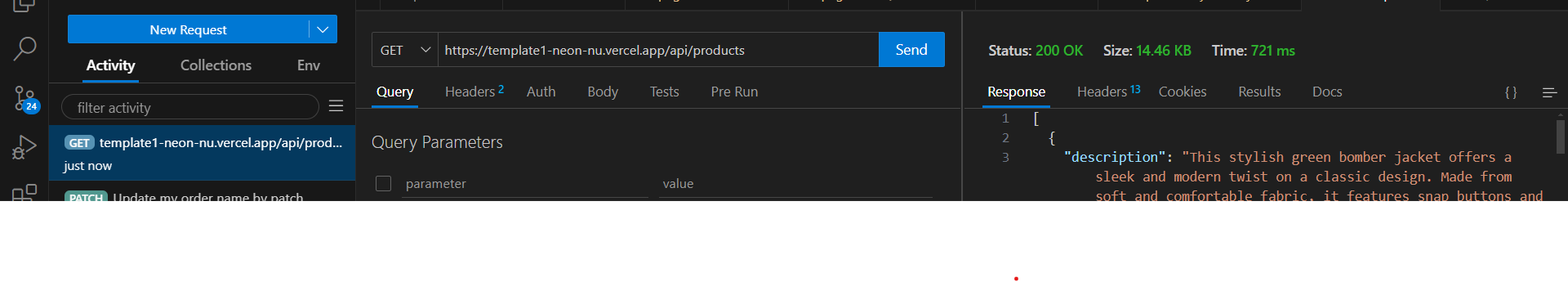
Introduced fallback UI elements to manage situations where API data retrieval fails. By displaying loading spinners, error messages, or placeholder content, we keep users informed and engaged while data is being fetched or when errors occur."



**API Response Validation Using Thunder Client**  
"Utilized the Thunder Client extension to test and validate API responses. The results confirmed that the API endpoints are returning the expected status codes, data structure, and response times, ensuring the reliability and correctness of the API."

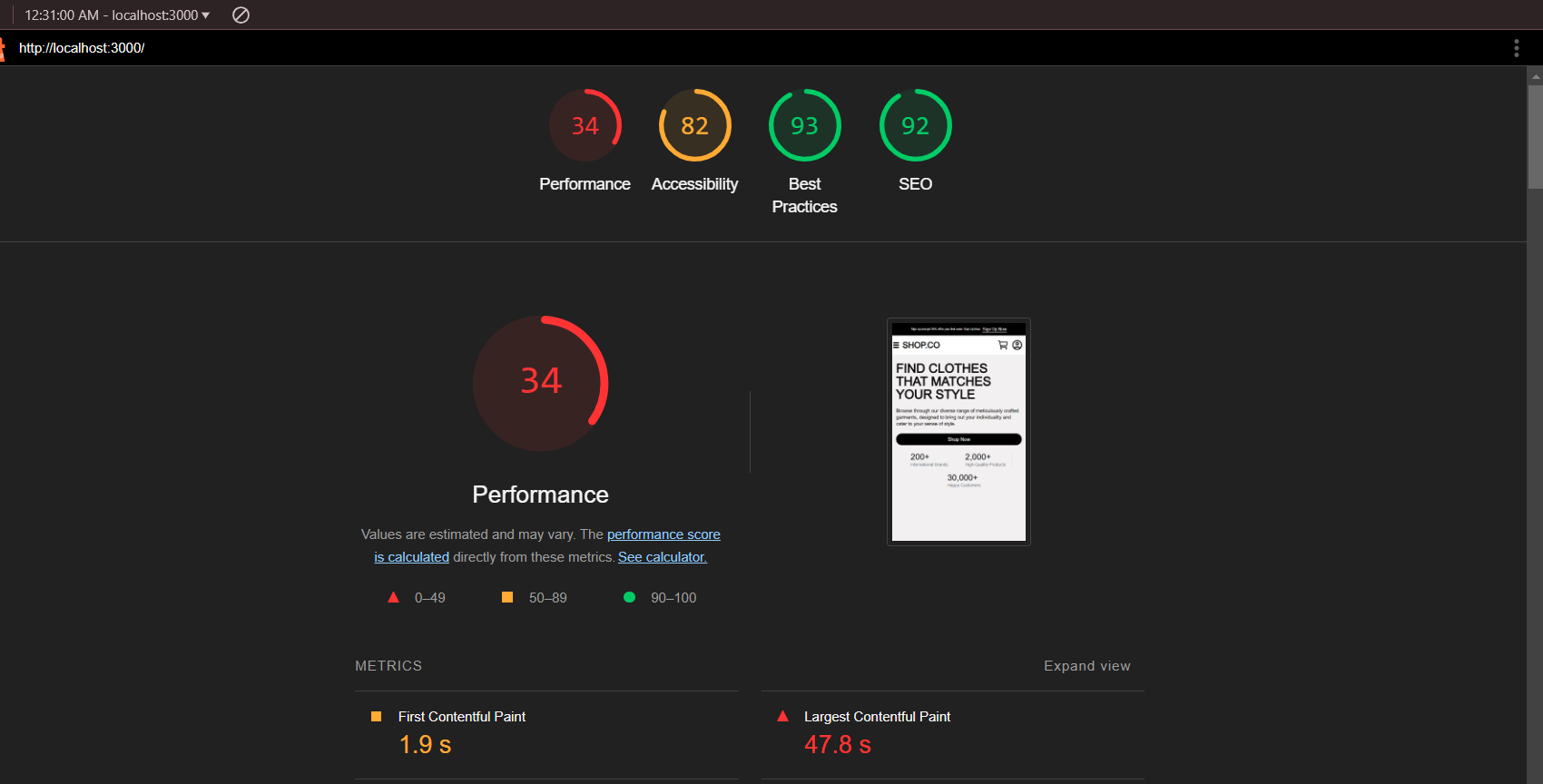
"Thunder Client was used to perform direct API tests, verifying the correctness and performance of each endpoint. The results demonstrated that all endpoints returned the correct data and met expected performance thresholds, with minor optimizations needed for a few endpoints."

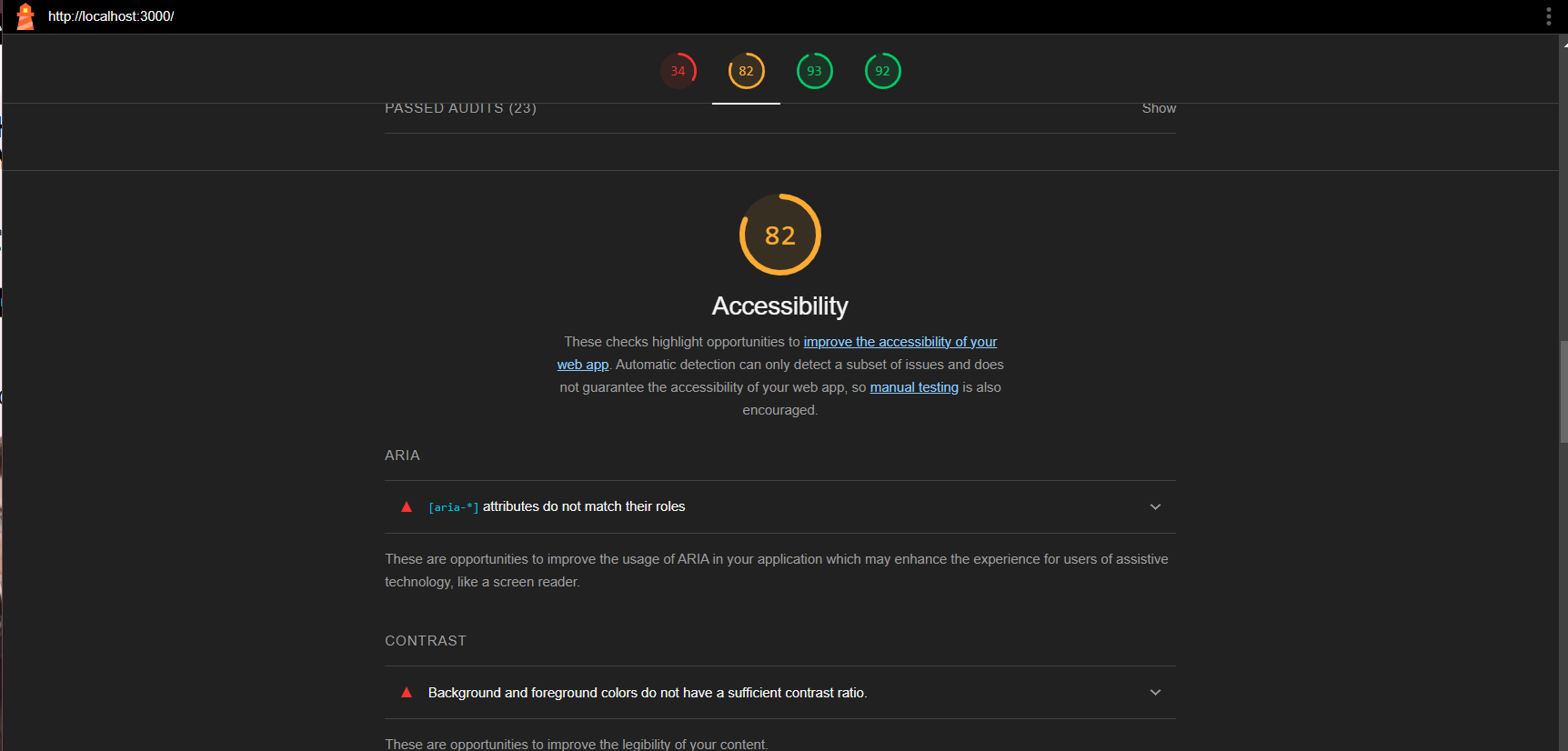




Analyze Performance Testing:

Conducted thorough performance testing to analyze the application's behavior under various load conditions. The results helped identify bottlenecks, memory usage patterns, and response times, leading to optimize code and improved overall system performance."

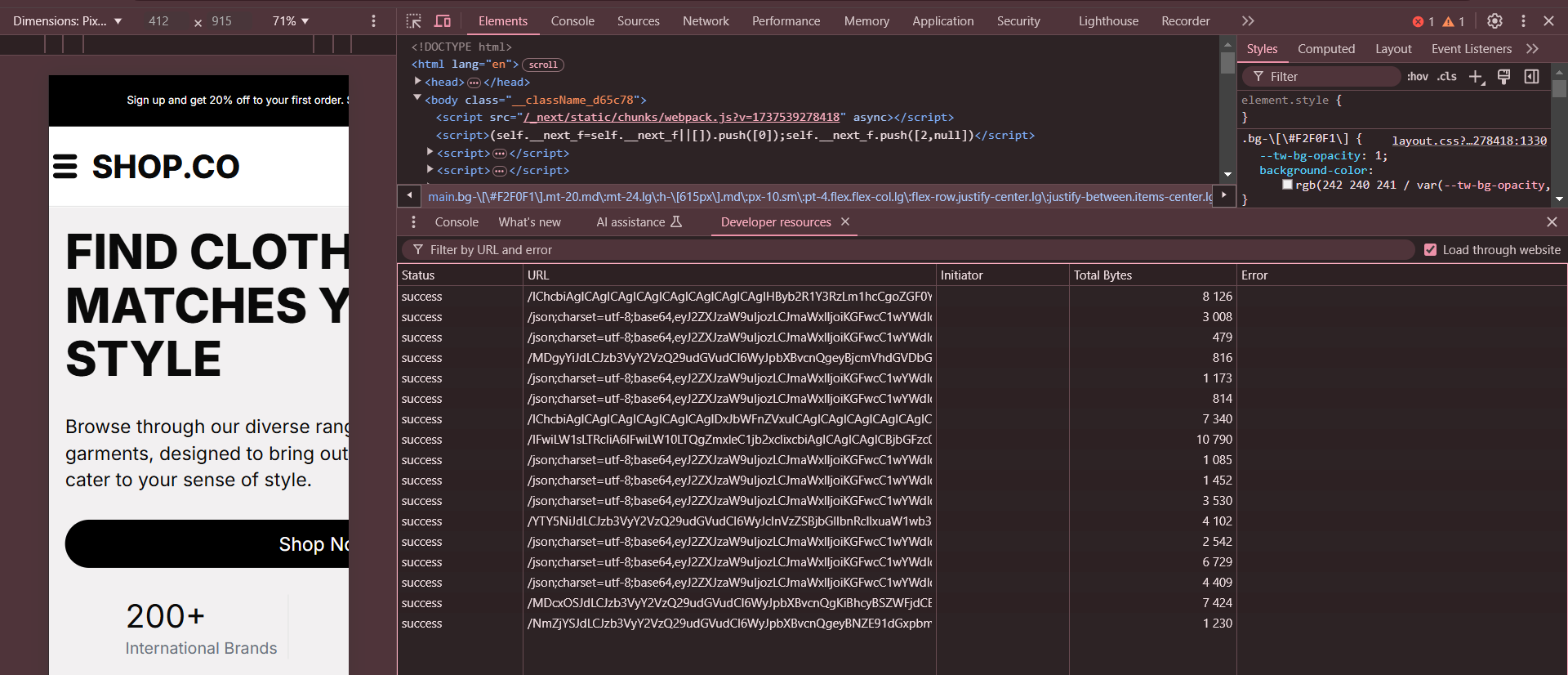




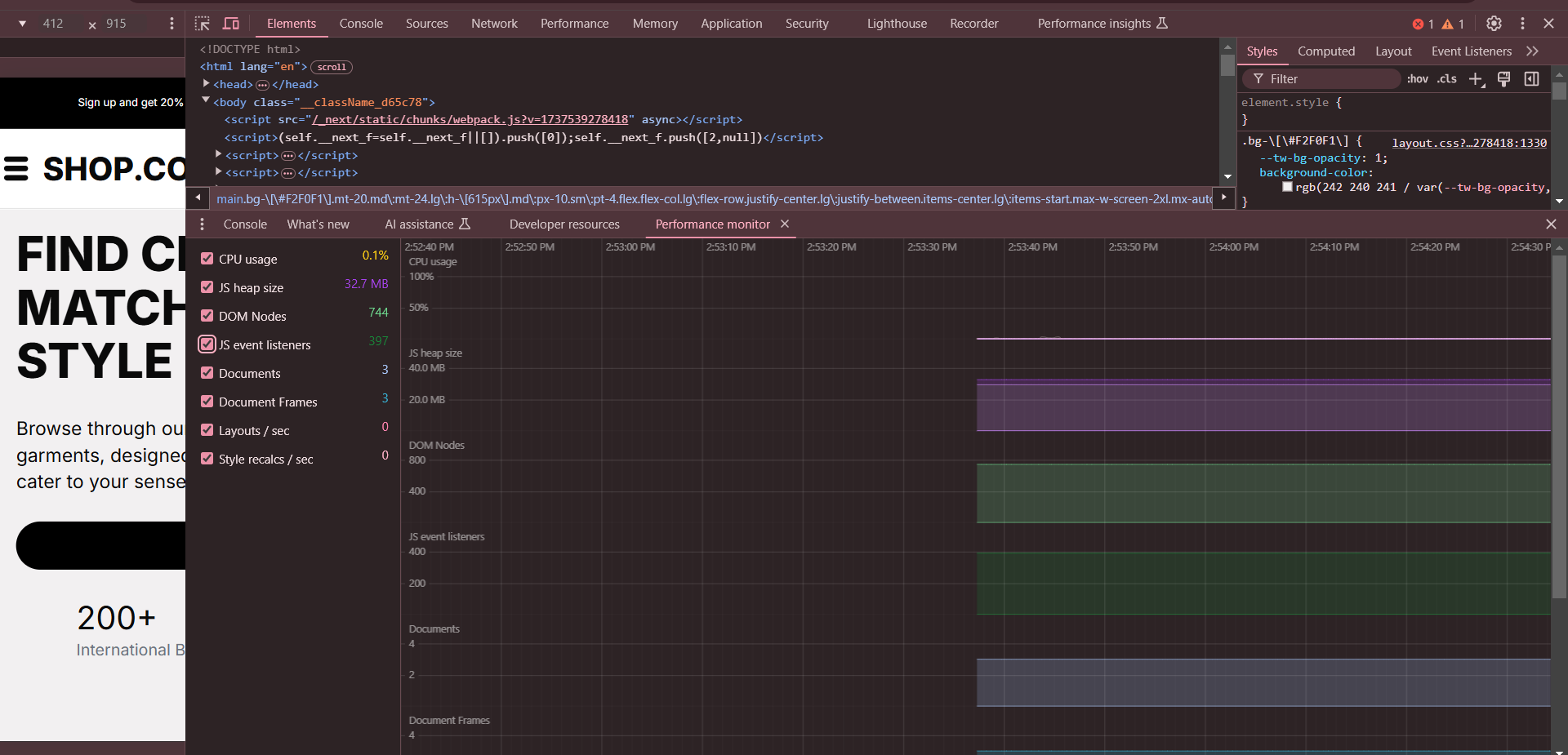
CSV Table:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  |  |  |  |  |   **Test Case ID** | **Test Case Description** | | | **Test Steps** | | **Expected Result** |
| **TC001** | **Validate product listing page** | | | **Open product page > Verify products** | | **Products displayed correctly** |
| **TC002** | **Test API error handling** | | | **Disconnect API > Refresh page** | | **Show fallback UI with error message** |
| **TC003** | **Check cart functionality** | | | **Add product to cart > Verify cart contents** | | **Cart updates with added product** |
| **TC004** | **Ensure responsiveness on mobile** | | | **Resize browser window > Check layout** | | **Layout adjusts properly to screen size** |
|  |  | | |  | |  |
|  |  | | |  | |  |
| **Actual Result** | **Status** | **Severity Level** | **Assigned To** | | **Remarks** |
| **Products displayed correctly** | **Passed** | **High** | **-** | | **No issues found** |
| **Error message shown** | **Passed** | **High** | **-** | | **Handled gracefully** |
| **Cart updates as expected** | **Passed** | **Medium** | **-** | | **Little bit good** |
| **Responsive layout working as intended** | **Passed** | **High** | **-** | | **Test successful** |

Developer Resources:



Performance Monitor:



Step 6: User Acceptance Testing (UAT):

1. Simulate Real-World Usage: Test tasks like browsing chairs, adding chairs to the cart, and completing the checkout and complete the checkout process.

2. Identify usability issues such as slow loading confusing navigation or broken forms.

3. Solution: If an issues arises, fix UI/UX problems (e.g., improve navigation flow, fix form validation), enhance speed by optimizing code or improve the design for better user interaction.

Step 07: Documentation Updates:

• Issues and fixes: key issues have been founded and resolved (e.g. performance optimization error handling improvements.

• Screenshots: Screenshots have been included to show the changes.

• PDF/Markdown: Documentation has been updated and formatted as per requirements.

• Test Cases and Tools: Test cases, tools like Postman, or thunder client, Lighthouse, and optimization strategies have been clearly documented.