# Document Title: Day 5 - Testing and Backend Refinement - StichHub

StichHub is an e-commerce platform designed to provide a seamless shopping experience with dynamic features and robust backend support. I have worked on building a responsive and fully functional website using modern web technologies like Next.js and Sanity CMS. The project focuses on delivering optimized performance, secure data handling, and a user-friendly interface.

Today, on Day 5 of the hackathon, the focus was on testing the platform's features, refining the backend integration, and ensuring all components work cohesively. Below is the detailed report of the tasks accomplished, challenges faced, and results achieved.
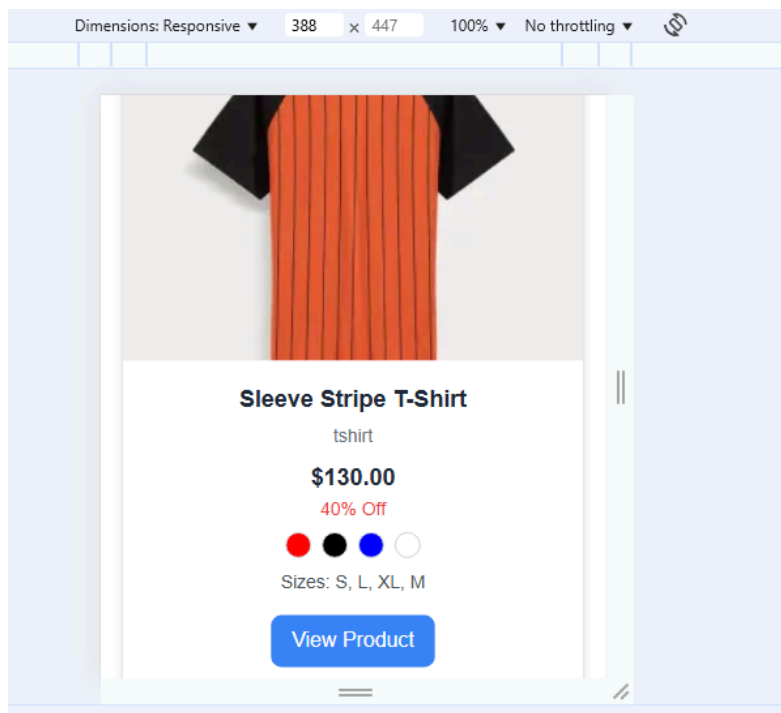
1. Test Cases Executed and Results

Website layout : proper styling of website using nextjs and tailwind optimizing images and showing data
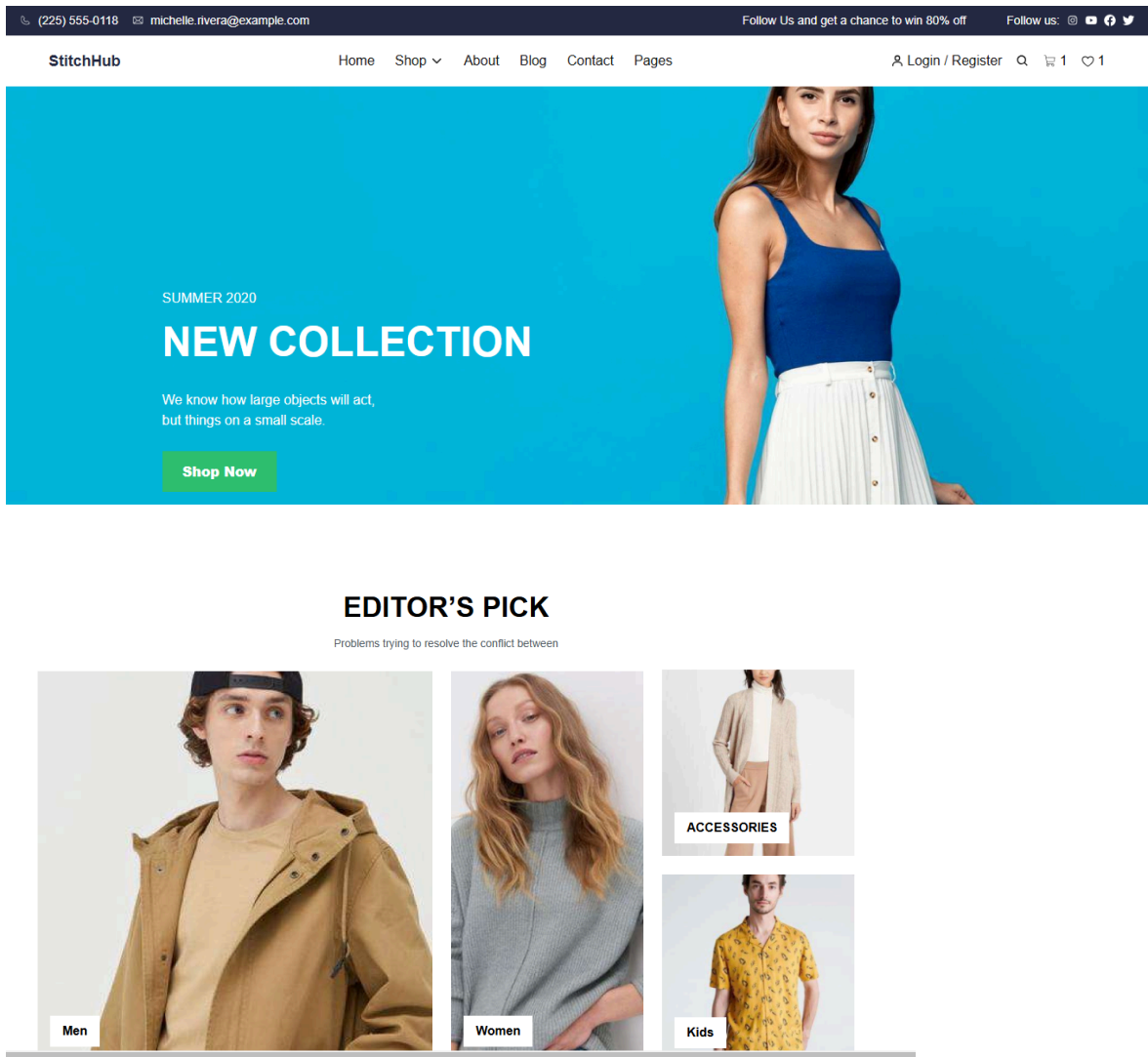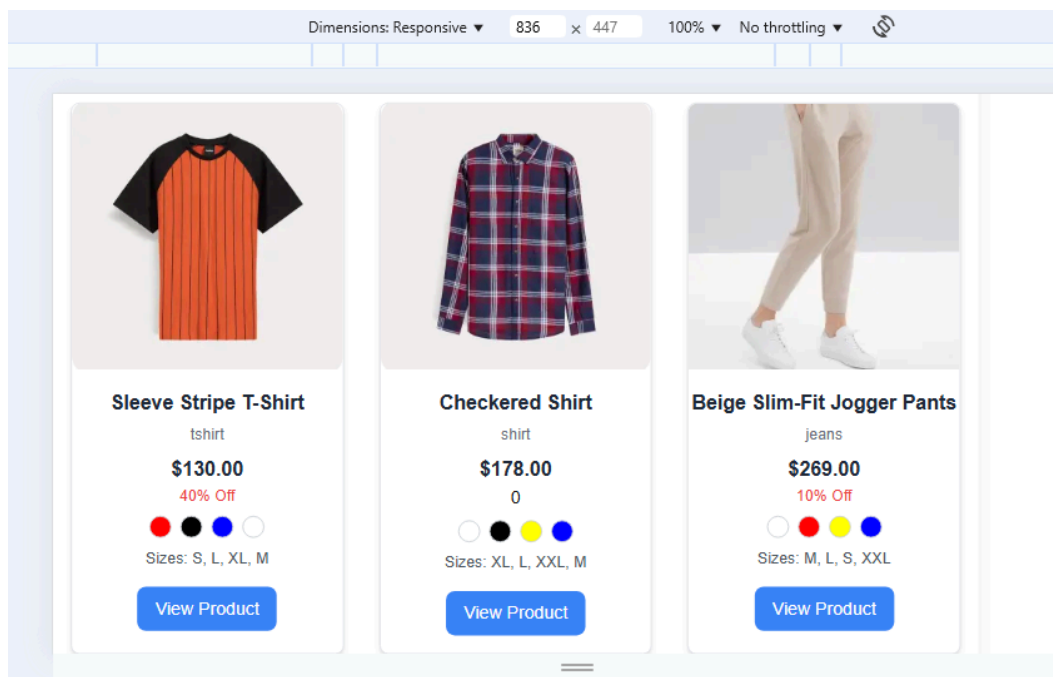
2.Responsive website for mobile and tablets

Implementing proper media queries to make the website responsive for mobile and tablet

For mobile

For tablet view

Product Pages

Products are fetch correctly from sanity CMS

- **Dynamic Product Pages:** Verified that all product pages load correctly with accurate details.

Category Filtering: Tested category filters to ensure products display under their respective categories.

Categories

Hoodie

Jeans

Shirt

Short
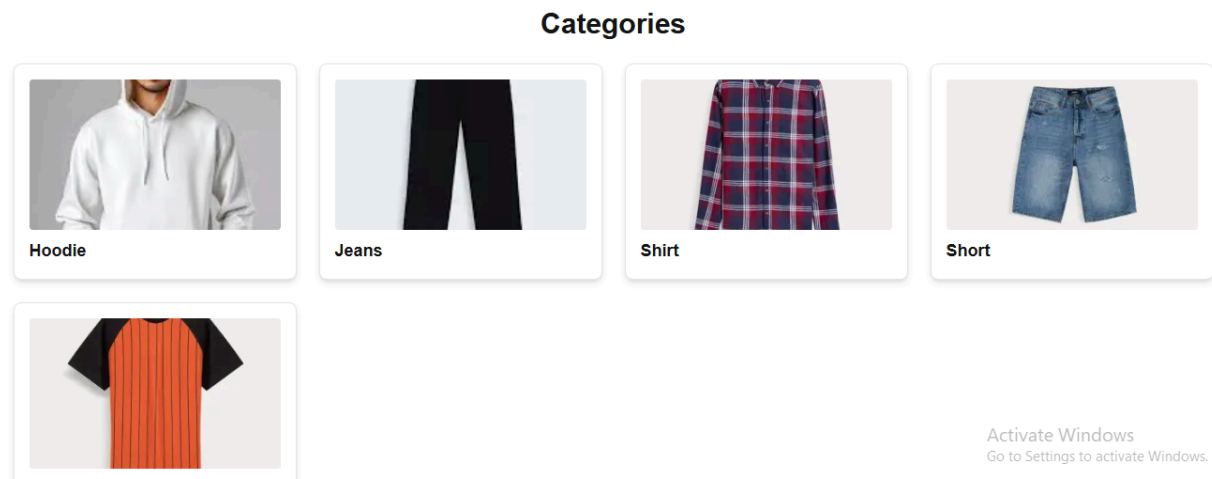
Dynamic display
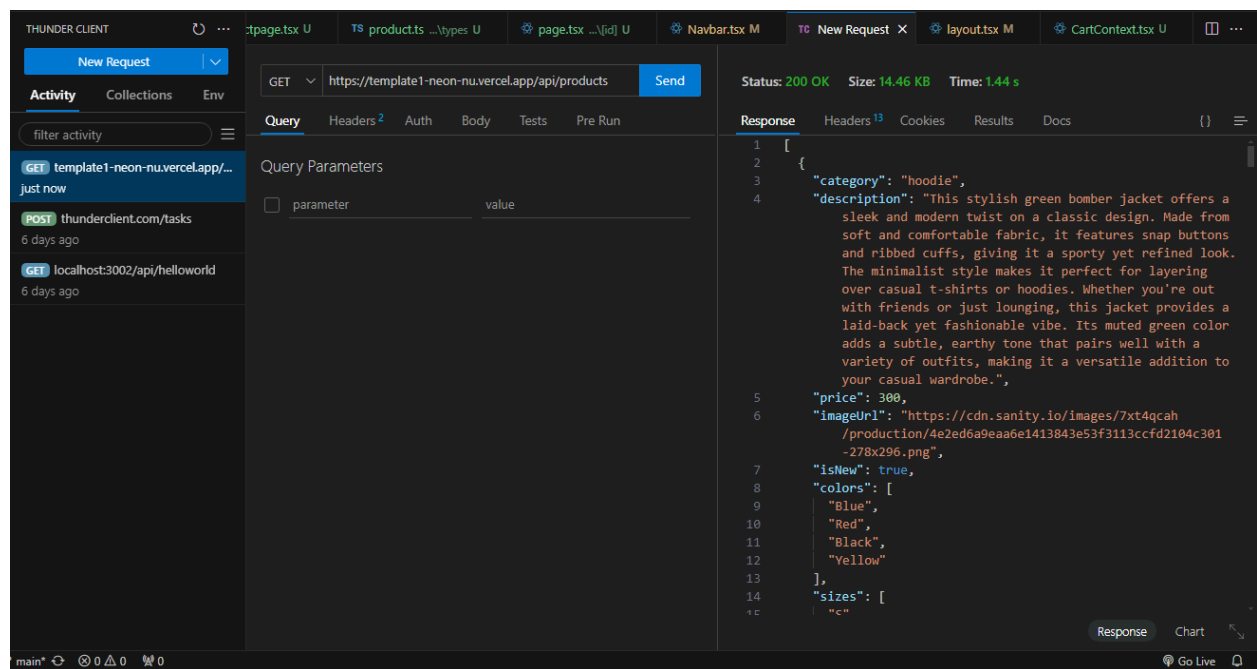
Products are linked to their respective category

Hoodie

Classic White Pullover Hoodie
$150

Casual Green Bomber Jacket
$300

Classic Black Pullover Hoodie
$128
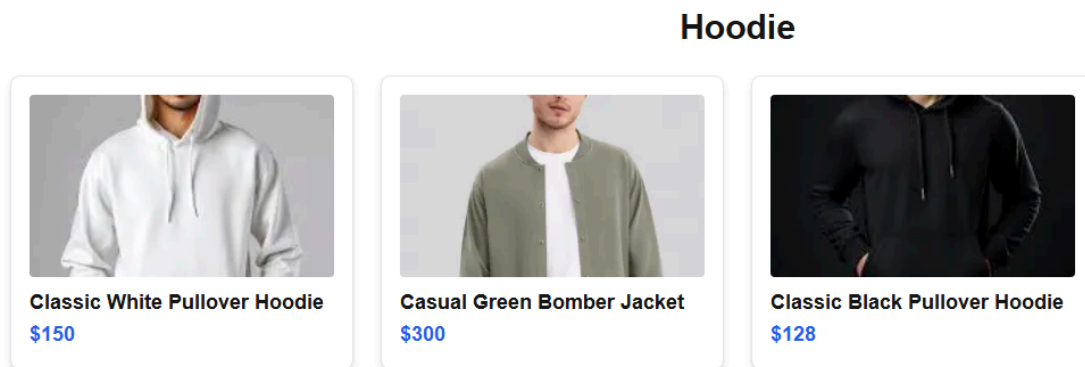
Add-to-Cart Functionality: Confirmed that items can be added, removed, and persisted in the cart.

API Testing: Validated API responses for all endpoints using thunderClient , ensuring correct data structure and handling of errors.using the GET method "https://template1-neon-nu.vercel.app/api/products"

| GET ∨ | https://template1-neon-nu.vercel.app/api/products | Send |

Status: 200 OK   Size: 14.46 KB   Time: 1.44 s

Query   Headers ²   Auth   Body   Tests   Pre Run

Response   Headers ¹³   Cookies   Results   Docs

Query Parameters

2. Performance Optimization

- **Lighthouse Analysis:** Improved performance scores by optimizing images, enabling caching, and reducing unused JavaScript.using command npm install -g lighthouse

http://localhost:3000/

53   86   96   92
Performance Accessibility Best Practices SEO

There were issues affecting this run of Lighthouse:
- Chrome extensions negatively affected this page's load performance. Try auditing the page in incognito mode or from a Chrome profile without extensions.

53

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

StitchHub

53   86   96   92

METRICS   Expand view

- First Contentful Paint   ▲ Largest Contentful Paint
  1.0 s   **4.2 s**

▲ Total Blocking Time   ● Cumulative Layout Shift
  10,890 ms   **0**

■ Speed Index
  4.9 s

⊞ View Treemap

Show audits relevant to: All FCP LCP TBT

DIAGNOSTICS

▲ Reduce JavaScript execution time — 12.8 s   ∨

▲ Minimize main-thread work — 16.0 s   ∨

A detail diagnostic report

53 86 96 92

▲ Reduce JavaScript execution time — 12.8 s ⌄

▲ Minimize main-thread work — 16.0 s ⌄

▲ Serve images in next-gen formats — Potential savings of 711 KiB ⌄

▲ Reduce initial server response time — Root document took 2,730 ms ⌄

▲ Efficiently encode images — Potential savings of 186 KiB ⌄

▲ Largest Contentful Paint element — 4,160 ms ⌄

▲ Page prevented back/forward cache restoration — 3 failure reasons ⌄

◼ Preload Largest Contentful Paint image — Potential savings of -10 ms ⌄

◼ Minify JavaScript — Potential savings of 5 KiB ⌄

◼ Eliminate render-blocking resources — Potential savings of 0 ms ⌄

◼ Avoid serving legacy JavaScript to modern browsers — Potential savings of 0 KiB ⌄

◼ Reduce unused JavaScript — Potential savings of 40 KiB ⌄

53 86 96 92

86

Accessibility

These checks highlight opportunities to improve the accessibility of your web app. Automatic detection can only detect a subset of issues and does not guarantee the accessibility of your web app, so manual testing is also encouraged.

CONTRAST

▲ Background and foreground colors do not have a sufficient contrast ratio. ⌄

These are opportunities to improve the legibility of your content.

NAMES AND LABELS

(53) (86) (96) (92)

96

Best Practices

GENERAL

▲ Browser errors were logged to the console ⌄

▲ Missing source maps for large first-party JavaScript ⌄

TRUST AND SAFETY

○ Ensure CSP is effective against XSS attacks ⌄

3. Security Measures

- **Input Validation:** Implemented strict validation to prevent injection attacks.
- **API Security:** Secured API endpoints with appropriate authentication and error handling.
- **User Data Protection:** Ensured compliance with data privacy standards by encrypting sensitive data.

Developer Tools: Monitored console for errors, network for failed requests, and application storage for caching and data handling.

Status	URL	Initiator	Total Bytes	Error
success	/json;charset=utf-8;base64,eyJ2ZXJzaW9uIjozLCJmaWxlIjoiKGFwcC1wYWdlcy1icm93c2VyKS8uL		11 042	
success	/json;charset=utf-8;base64,eyJ2ZXJzaW9uIjozLCJmaWxlIjoiKGFwcC1wYWdlcy1icm93c2VyKS8uL		7 180	
success	/Lm1hcCgoY29sb3IslGluZGV4KSA9PiAoXHJcbiAgICAgICAgICAgICAgICAgPHNwYW5 5cclxul(		6 514	
success	/json;charset=utf-8;base64,eyJ2ZXJzaW9uIjozLCJmaWxlIjoiKGFwcC1wYWdlcy1icm93c2VyKS8uL		674	
success	/json;charset=utf-8;base64,eyJ2ZXJzaW9uIjozLCJmaWxlIjoiKGFwcC1wYWdlcy1icm93c2VyKS8uL		1 173	
success	/MWMwYiJdLCJzb3VyY2VzQ29udGVudCI6WyJleHBvcnQgY29uc3QgdXBkYXRlZFZByb2R1Y3RRdV		482	

Performance Tab: Analyzed page load time, network performance, and rendering efficiency, optimizing slow-loading assets and ensuring smooth interactions.
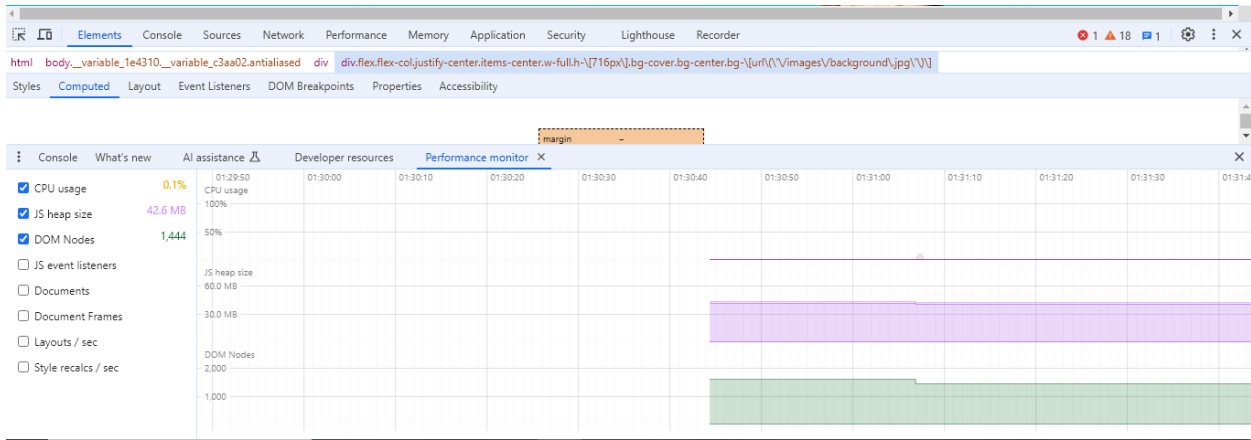
Security Tab: Checked for HTTPS, SSL certificate validity, mixed content, and security headers like CSP and X-Content-Type-Options.
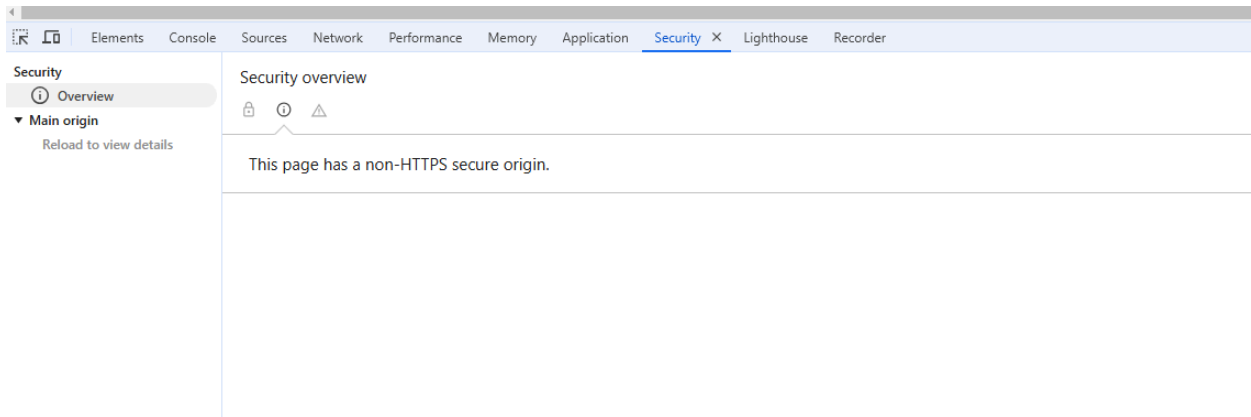
Functionality check

Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Remarks
TC001	Validate product listing page	Open product page > Verify products	Products displayed correctly	Products displayed correctly	passed	high	Test successful
TC002	Test API error handling	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	Hardly worked
TC004	Ensure responsiveness on mobile	Resize browser window > Check layout	Layout adjusts properly to screen size	Responsive layout working as intended	passed	low	Works as expected
TC005	Categories page				passed	low	sucessful

Conclution

StichHub is a fully functional e-commerce platform with dynamic product pages, real-time data fetching from Sanity, and an intuitive shopping experience. The integration of categories ensures that products are shown in their respective groups, improving product discoverability. The add-to-cart functionality further enhances the user experience, making it easy for customers to navigate the site and make purchases.