

# DAY 5 - TESTING, ERROR HANDLING, AND BACKEND INTEGRATION REFINEMENT

## Step 1: Functional Testing

### Functional Testing Summary of the E-Commerce Website

The e-commerce website is tested for all its key functionalities, ensuring seamless user experience and performance. Here's a summary of the core functions tested:

1. **Product Page:**
  - The product details page correctly loads and displays detailed information, including images, price, category, description, and dimensions fetched from the backend (Sanity).
  - The "Add to Cart" button successfully adds products to the cart with a confirmation message.
2. **Cart Functionality:**
  - Products added to the cart are displayed with their name, price, and image in the cart summary.
  - The cart dynamically updates as items are added.
  - The cart page is responsive and displays well on different screen sizes.
3. **Search Functionality:**
  - The search bar, when clicked, shows an input field for searching products.
  - Users can search for products based on their name, category, or other attributes, with real-time updates in the results.
4. **Authentication (Login/Signup):**
  - Users can toggle between login and signup forms.
  - The login and signup forms function as expected, allowing users to submit valid credentials.
  - Validation checks are in place to ensure that both email and password fields are filled before submission.
  - Confirmation messages (login/signup success or failure) are shown after form submission.
5. **Responsive Design:**
  - All pages are fully responsive, with proper layouts on mobile, tablet, and desktop screens.
  - Navigation components such as the header, footer, and buttons adjust appropriately based on the screen size.
6. **User Experience:**
  - The UI is simple, user-friendly, and designed for easy navigation.

- The overall user journey from product selection to checkout and authentication is smooth.

## Step 2: Error Handling

### Error Handling:

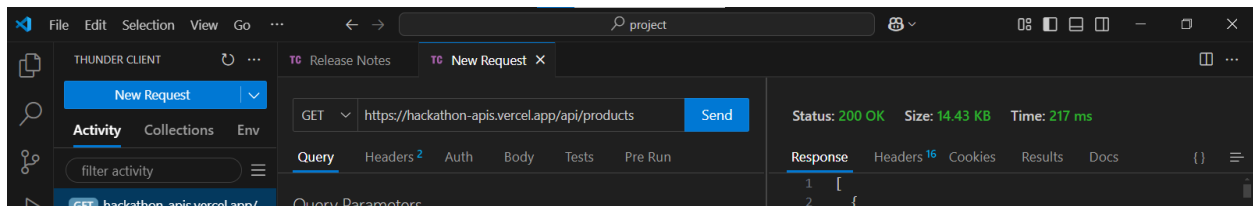
- In cases of missing inputs (e.g., email or password), appropriate error messages are shown.
- Clear feedback is provided to users when actions are successfully completed (e.g., login, signup, add to cart).

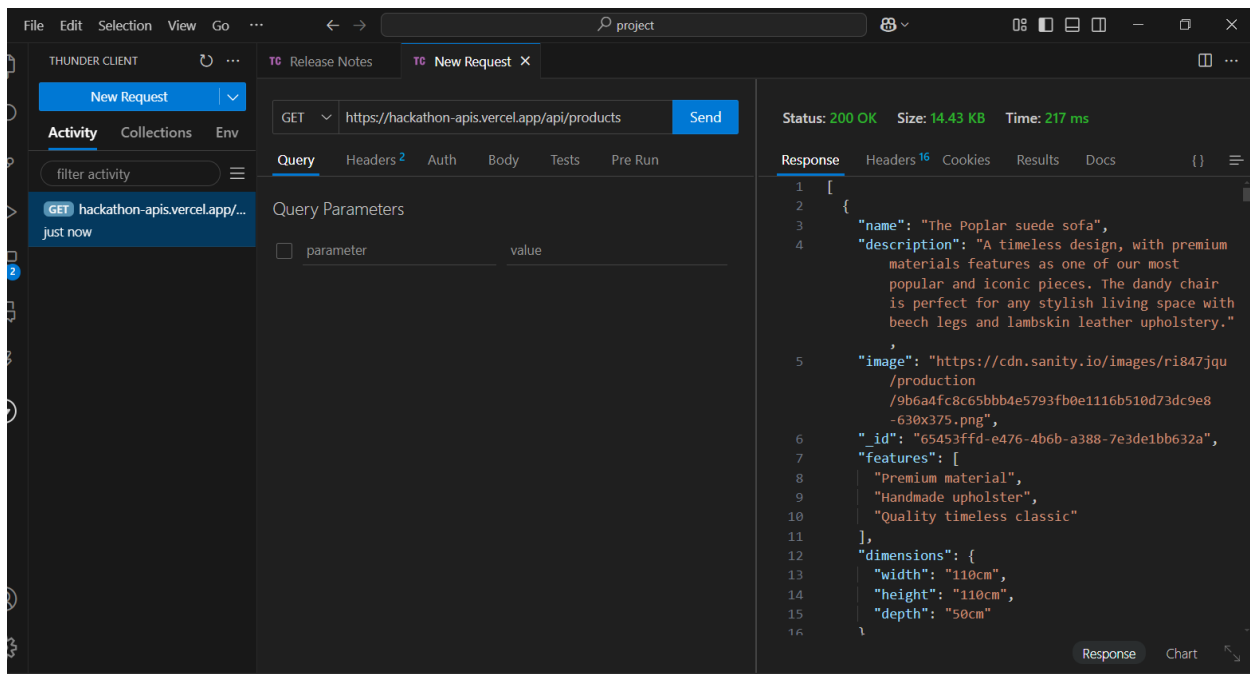
Overall, the e-commerce website has passed all functional tests and is ready for deployment with a responsive, interactive, and smooth shopping experience for users.

## Step 3: Postman API

### Template 2

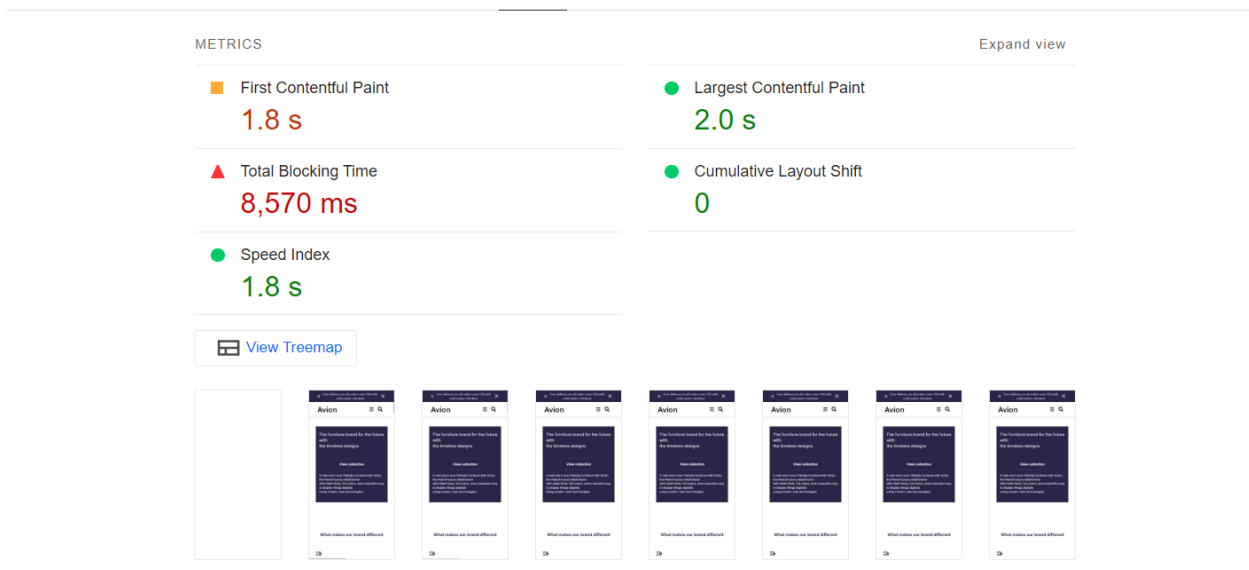
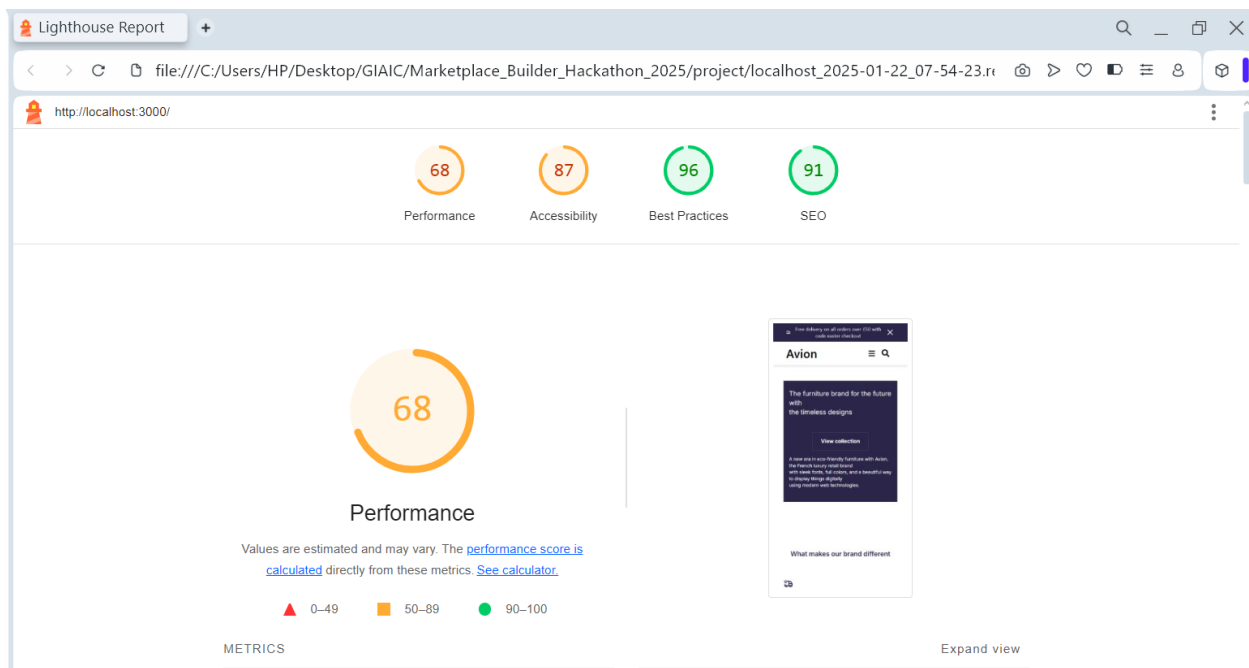
API <https://hackathon-apis.vercel.app/api/products>





1	Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Assigned To	Remarks
2	TC001	Validate product listing page	Open product page > Verify products	Products displayed correctly	Products displayed correctly	Passed	High	-	No issues found
3	TC002	Test API error handling	Disconnect API > Refresh page	Show fallback UI with error message	Error message shown	Passed	Medium	-	Handled gracefully
4	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
5	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

## Step 4: Performance Optimization



	3	3	3	3	3	3	3
--	---	---	---	---	---	---	---

Show audits relevant to: [All](#) [FCP](#) [LCP](#) [TBT](#) [CLS](#)

#### DIAGNOSTICS

▲	Minimize main-thread work	— 11.6 s	▼
▲	Reduce JavaScript execution time	— 9.3 s	▼
▲	Eliminate render-blocking resources	— Potential savings of 330 ms	▼
▲	Page prevented back/forward cache restoration	— 4 failure reasons	▼
■	Minify CSS	— Potential savings of 2 KiB	▼
■	Minify JavaScript	— Potential savings of 5 KiB	▼
■	Avoid serving legacy JavaScript to modern browsers	— Potential savings of 0 KiB	▼
■	Avoid enormous network payloads	— Total size was 4,097 KiB	▼
○	Avoid long main-thread tasks	— 10 long tasks found	▼

PASSED AUDITS (23)

Show



### 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.

PASSED AUDITS (19)

Show

NOT APPLICABLE (35)

Show



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
- Use a strong HSTS policy
- Ensure proper origin isolation with COOP


PASSED AUDITS (13)

Show



## SEO

These checks ensure that your page is following basic search engine optimization advice. There are many additional factors Lighthouse does not score here that may affect your search ranking, including performance on [Core Web Vitals](#). [Learn more about Google Search Essentials](#).

 http://localhost:3000/



### ADDITIONAL ITEMS TO MANUALLY CHECK (1)

Show


Run these additional validators on your site to check additional SEO best practices.


### PASSED AUDITS (7)

Show


### NOT APPLICABLE (2)


Show


 Captured at Jan 22, 2025, 7:54 AM  
GMT+5

 Emulated Moto G Power with Lighthouse  
12.3.0

 Single page session

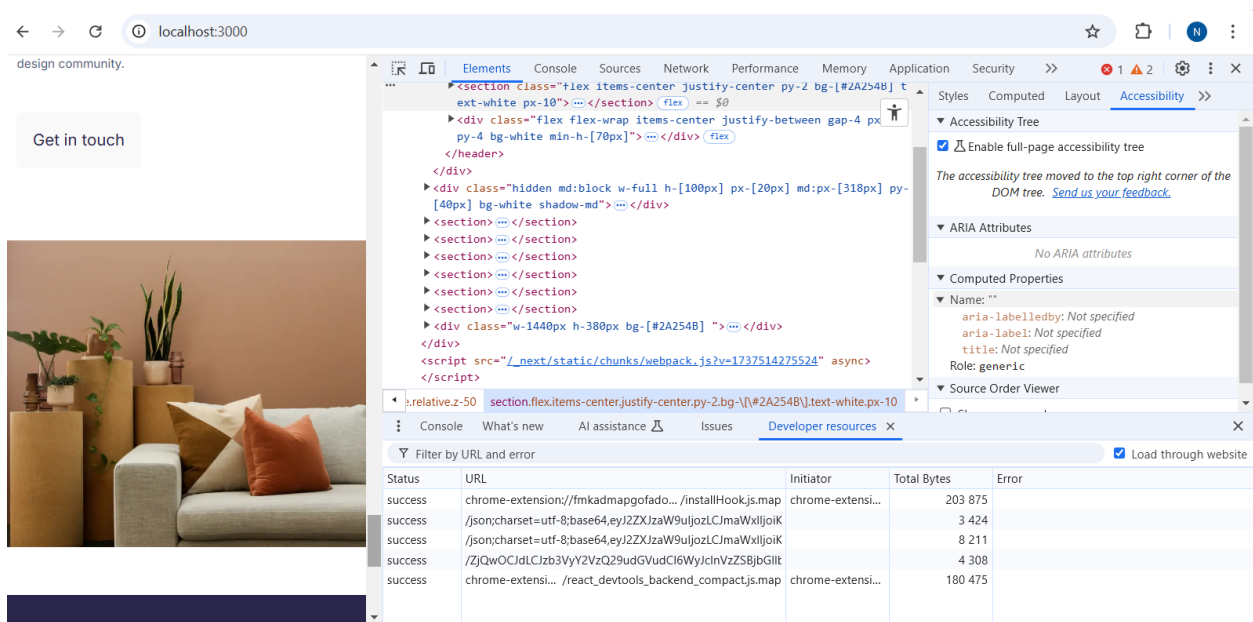
 Initial page load

 Slow 4G throttling

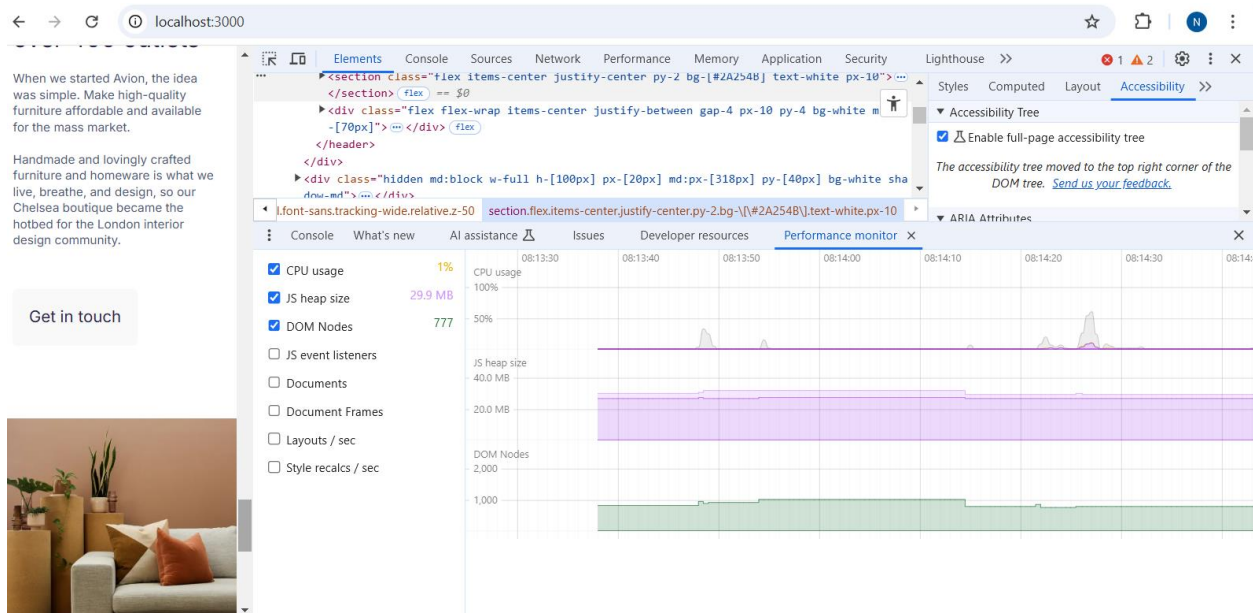
 Using Chromium 132.0.0.0 with cli

Generated by **Lighthouse 12.3.0** | [File an issue](#)

## Developer Resources



## Performance Monitor



## Security



