Hackathon Day-05

Furniture E-Commerce Website

Functional Testing Document for Furniture E-Commerce Website

Introduction

This document outlines the functional testing plan for the Furniture E-Commerce Website. The website enables users to browse, view, and purchase furniture products seamlessly. It also incorporates features like user authentication, wishlist management, dynamic product pages, a cart slider, and order placement functionality. The backend for user authentication is implemented using Firebase, and the order management system is powered by Sanity CMS.

Components and Functional Testing

1. User Signup and Login

Heading: User Authentication
Functionality:

- Users can sign up using their email and password using Firebase Authentication.

- Existing users can log in to their accounts.

- Users can sign up directly using their Google or Facebook accounts.

- Validation messages appear for incorrect credentials or missing fields.

- After successful login, users are redirected to the home page.

Test Cases:

1. Verify that users can create accounts with valid email and password.

2. Ensure error messages appear for invalid email formats or weak passwords.

3. Confirm users can log in with valid credentials.

4. Verify that users can sign up and log in using Google and Facebook accounts.

5. Check that users are redirected to the home page after logging in.

6. Test the behavior when users provide incorrect credentials (e.g., wrong password).

Log In

Email Address

Email Address

Password

Enter Password

☐ Remember me

Log In   Lost Your Password?

Don't have an account? Register Now

Create an Account

First Name   Last Name

Enter Your First Name   Enter Your Last Name

Gender

Select Gender

Email Address

Email Address

Password   Confirm Password

Enter Password   Confirm Password

Sign Up

G   f

A link to set a new password will be sent to your email address.

---

2. Home Page

Heading: Home Page with Product Listings
Functionality:

- The home page displays a selection of products fetched from Sanity CMS.

- Users can view product names, images, and basic details.

- Clicking on a product redirects the user to a dynamic product detail page.

Test Cases:

1. Verify that products load correctly from the backend (Sanity CMS).

2. Confirm that clicking on a product redirect to its detail page.

3. Ensure that loading errors display appropriate fallback messages.

---

3. Shop Page

Heading: Product Listing on Shop Page
Functionality:

- Displays a list of products with details like name, price, and image.

- Users can view product prices and add products to their wishlist or cart.

- Users can remove products from the wishlist.

Test Cases:

1. Verify that all products are fetched and displayed correctly.

2. Confirm that users can add products to their wishlist.

3. Ensure users can remove items from the wishlist.

4. Test that users can add products to the cart.

---

4. Dynamic Product Page

Heading: Dynamic Product Detail Page
Functionality:

- Displays product details such as name, description, price, size, color, category, and quantity.

- Allows users to increase or decrease the quantity before adding the product to the cart.

- Related Products section shows additional items.

Test Cases:

1. Verify that the product details are displayed accurately.

2. Confirm users can adjust the quantity and add products to the cart.

3. Ensure that related products are fetched and displayed below the product details.

4. Test that clicking on related products redirects to their respective detail pages.

White Bed
Rs. 120

Blue Bed
Rs. 780

Replica Table
Rs. 750

Liberty Center
Rs. 1100
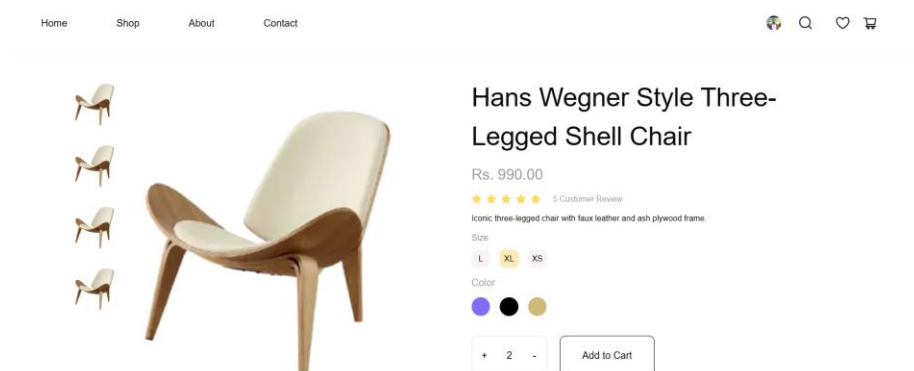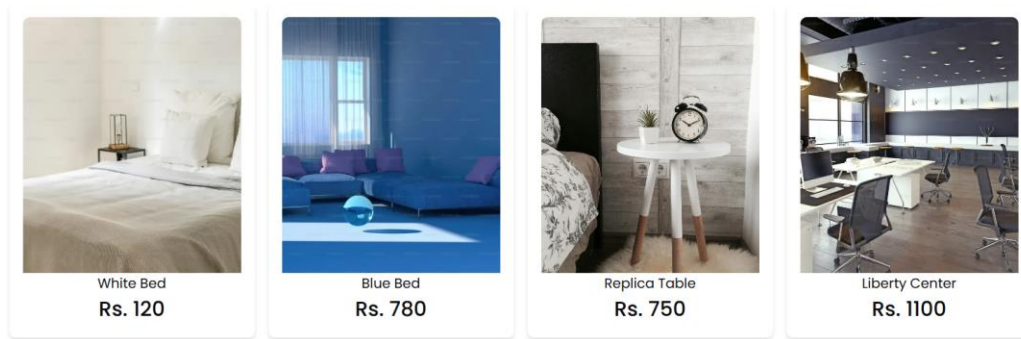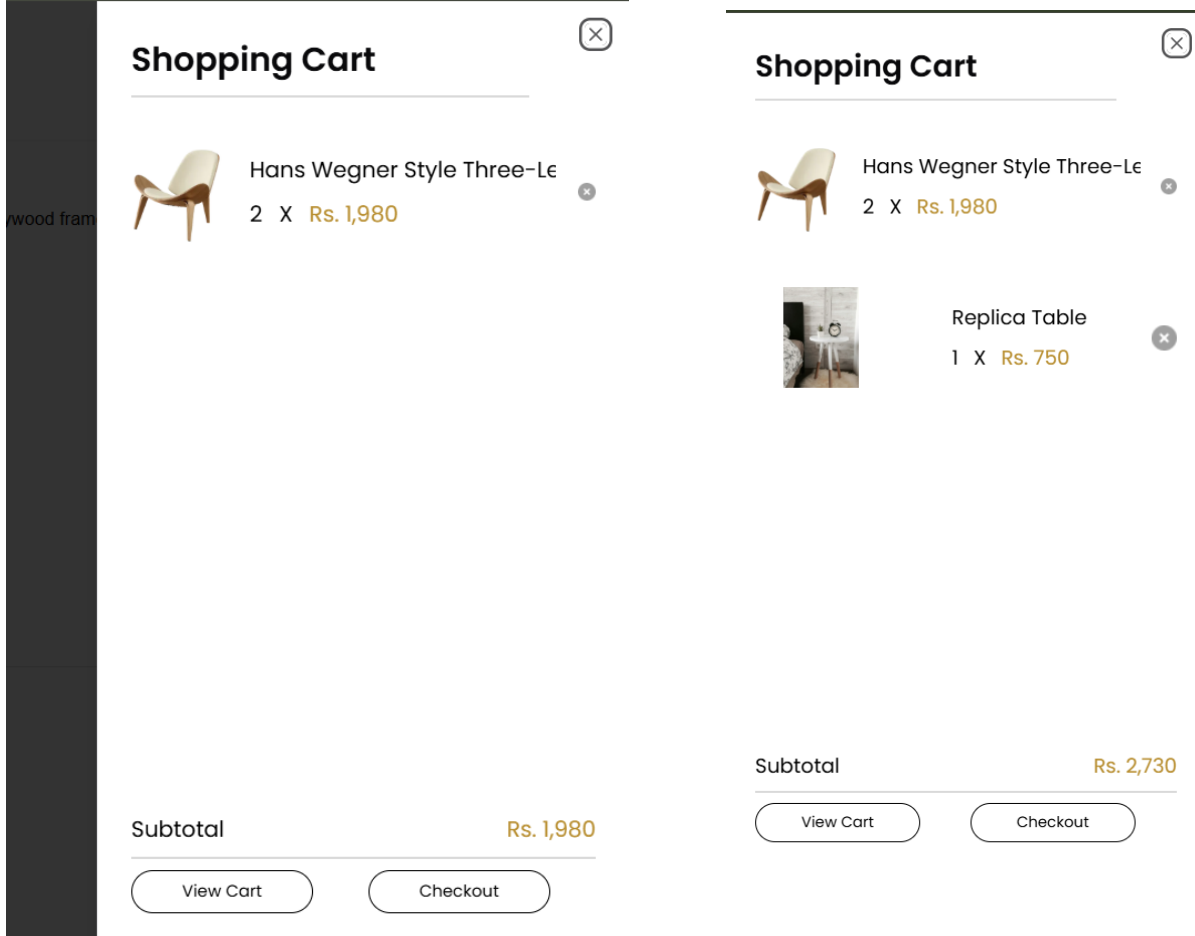
View More

---

5. Cart Slider

Heading: Add to Cart Functionality
Functionality:

- The cart slider shows selected products with their image, name, quantity, price, total price, and subtotal.

- Users can view two buttons: "View Cart" and "Checkout".

- Clicking "View Cart" takes users to the cart page.

- Clicking "Checkout" takes users to the checkout page.

Test Cases:

1. Verify that added products appear in the cart slider with accurate details.

2. Confirm that the subtotal is calculated correctly.

3. Ensure that the "View Cart" button redirects to the cart page.

4. Test that the "Checkout" button redirects to the checkout page.

Shopping Cart

Hans Wegner Style Three-Le
2 X Rs. 1,980

Subtotal   Rs. 1,980

View Cart   Checkout

Shopping Cart

Hans Wegner Style Three-Le
2 X Rs. 1,980

Replica Table
1 X Rs. 750

Subtotal   Rs. 2,730

View Cart   Checkout

6. Cart Page

Heading: Cart Management
Functionality:

- Displays all products added to the cart.

- Users can adjust the quantity or remove products from the cart.

- The cart updates the total price and subtotal dynamically.

Test Cases:

1. Verify that all selected products are displayed in the cart.

2. Confirm that adjusting the quantity updates the total price correctly.

3. Ensure that removing products updates the cart accurately.

4. Test for proper error handling when the cart is empty.

Product		Price	Quantity	Subtotal
	Hans Wegner Style Three-Legged Shell Chair	Rs. 990	− 4 +	Rs. 3,960
	Replica Table	Rs. 750	− 2 +	Rs. 1,500

Cart Totals

Subtotal	Rs. 5,460
Total	Rs. 5,460

Check Out

7. Checkout Page

Heading: Checkout and Order Placement

Functionality:

- Users can enter delivery details to place their order.

- After order placement, the data (user details and product information) is saved in Sanity CMS.

- Users are redirected to an order confirmation page.

Test Cases:

1. Verify that users can input and submit delivery details.

2. Confirm that the order is saved in Sanity CMS with the correct details.

3. Test that users are redirected to the confirmation page after successful order placement.

4. Ensure proper error messages appear for invalid or missing delivery details.

8. User Dashboard

Heading: User Dashboard for Order Management
Functionality:

- Users can view their profile and order history.

- Displays order details, including product names, quantities, and prices.

- Allows users to update their profile information.

Test Cases:

1. Verify that the user dashboard displays accurate order details.

2. Confirm that profile updates are saved successfully.

3. Ensure that order history is fetched correctly from Sanity CMS.

4. Test for proper error handling when no orders exist.

Dashboard

Welcome, **saud.saleem93@gmail.com**

Profile
Manage your profile details.

Orders
View your order history.

Settings
Adjust your account settings.

Support
Get help and support.

Logout

Profile

Name: Syed Saud Ali

Email: saud.saleem93@gmail.com

User ID:
cscmPLERSpUnauJYFN7r8aCk4se2

No customer data found.

Conclusion

The functional testing plan ensures that all components of the Furniture E-Commerce Website work as intended, providing a seamless user experience. Each feature has been tested for usability, accuracy, and responsiveness, ensuring that the website operates smoothly for end-users. Firebase Authentication and Sanity CMS integration have been validated for handling user authentication and order data management, respectively.

Responsiveness for Furniture E-Commerce Website

**Heading:** Website Responsiveness
Overview:

- The website is designed to be responsive, ensuring compatibility with various screen sizes, including desktops, tablets, and mobile devices.

- All components, including the home page, shop page, product detail page, and user dashboard, are fully responsive and adapt seamlessly to different screen resolutions.

- The cart component requires additional attention to address minor responsiveness issues.

Test Cases:

1. Verify that all components render properly on desktop, tablet, and mobile screens.

2. Confirm that text, images, and buttons adjust correctly for smaller screens.

3. Ensure the cart component's layout and functionality are tested on multiple screen sizes.

4. Test the website on various browsers and devices for consistent responsiveness.
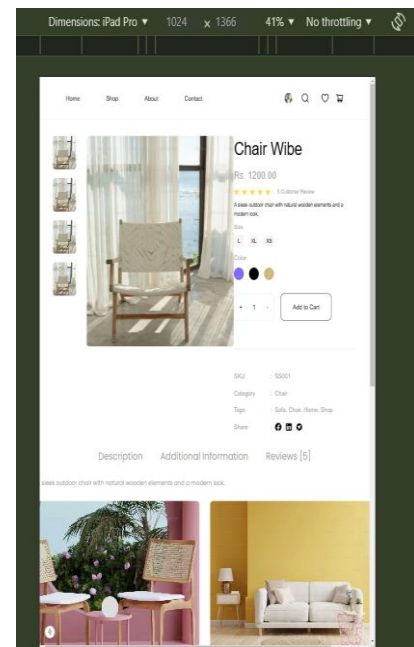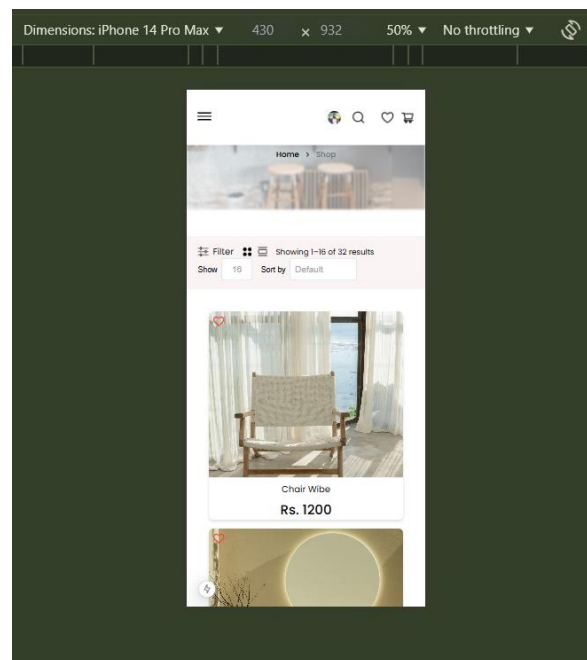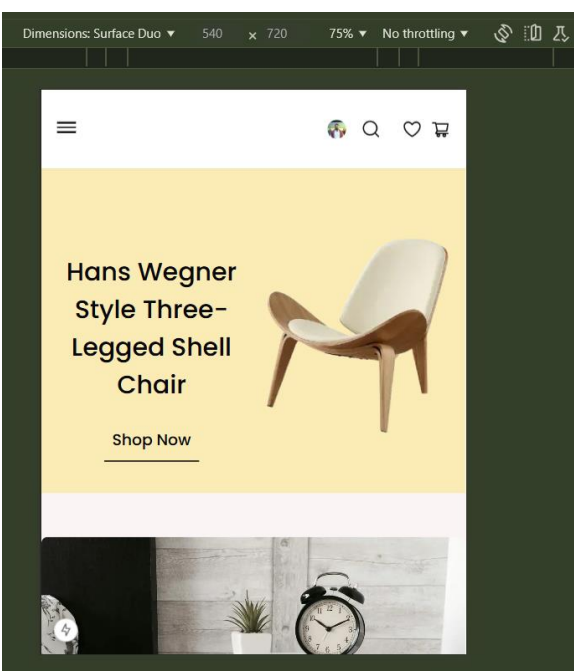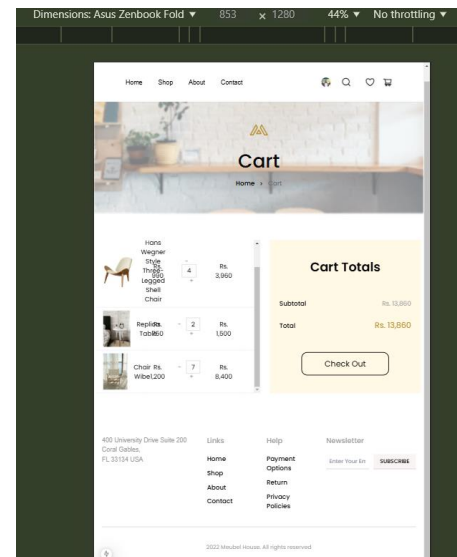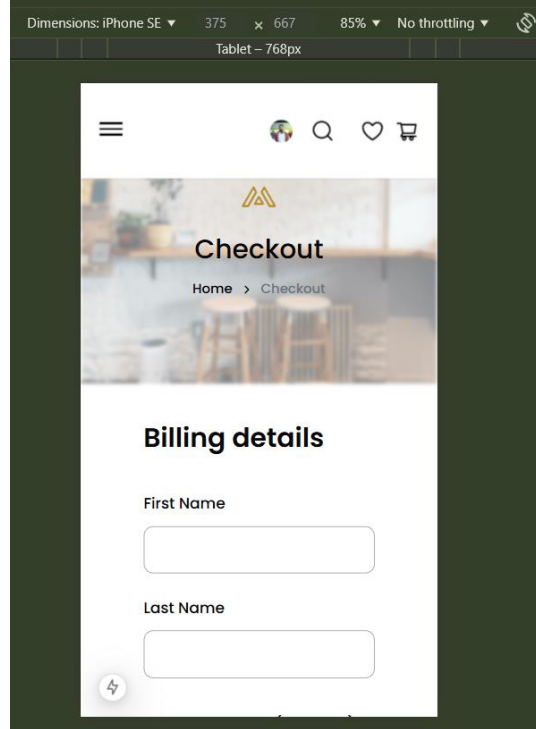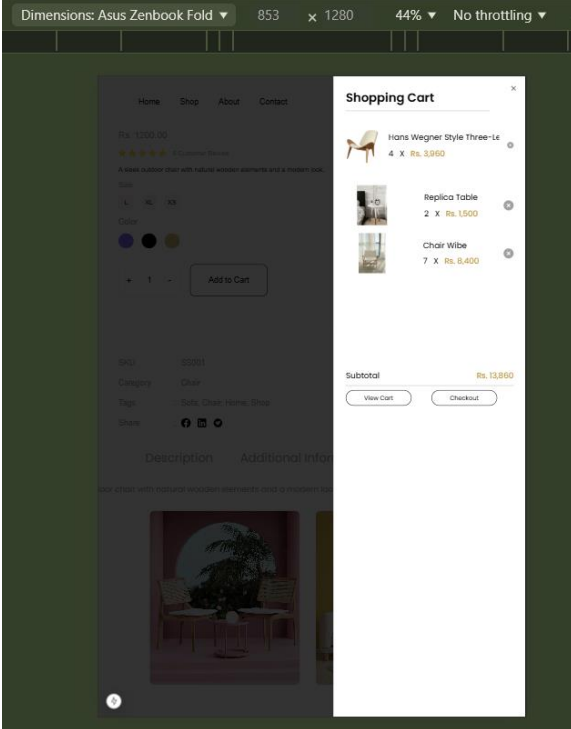
API for Furniture E-Commerce Website

1. **API Endpoint:**
   The API used for the Furniture E-Commerce Website is:
   https://template-0-beta.vercel.app/api/product

2. **Data Handling:**

   - The API provides product details such as name, price, description, image, size, color, and category.

   - The API returns JSON-formatted data for easier integration and manipulation in the frontend.

3. **API Testing Tool:**

   - **Postman** is used to test the API endpoints.

   - It verifies the accuracy and functionality of the API responses, ensuring that product data is fetched correctly.

4. **Common API Operations:**

- **GET**: Fetch product details for the website, ensuring dynamic rendering on the frontend.

- Additional operations (if applicable) can be implemented for CRUD functionality like creating, updating, or deleting products.
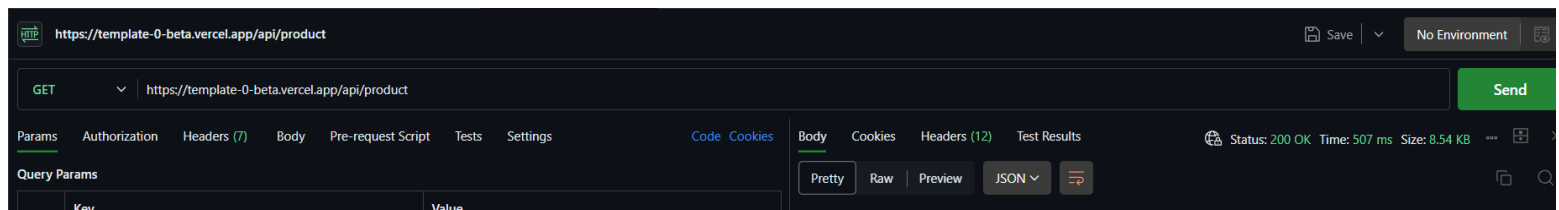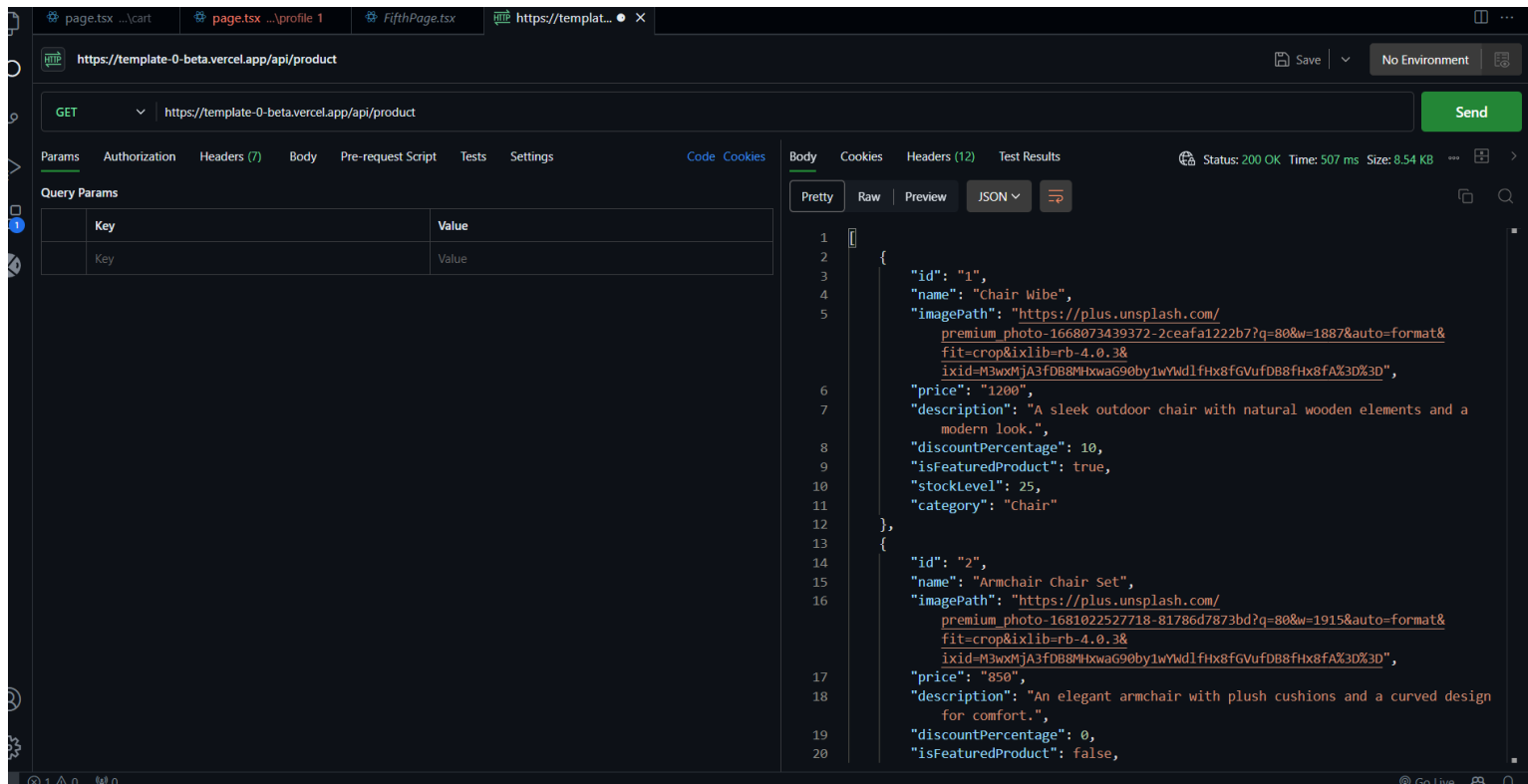
1	Test Case	Test Case De	Test Steps	Expected	Actual Result	Status	Severity	Assigne	Remarks
2	TC001	Validate Product listing page	Open product p > Verify p	Products displayed correctly	Products displayed	Passe	High	-	No issues
3	TC002	Test API Error handling	Disconnec > Refresh	Show fallback U with error mes	Error message sho	Passe	High	-	Handled g

| 4 | TC003 | Check cart functionality > Verify ca contents | Add prod to cart | Cart upda with adde product | Cart updates as ex | Passe | High | - | Works as e |
| 5 | TC004 | Ensure responsiven | Resize Browser window > Check la | Layout ad properly t screen siz | Responsive layout intend | Passe | Medium | - | Test succe |

Performance for Furniture E-Commerce Website

Tool Used: Lighthouse

Performance Metrics:

1. **Performance Score:** 29/100

2. **Accessibility Score:** 90/100

3. **Best Practices:** 100/100

4. **SEO:** 100/100

Detailed Metrics:

- **First Contentful Paint (FCP):** 1.1 seconds

- **Speed Index:** 2.7 seconds

- **Total Blocking Time (TBT):** 1,020 milliseconds

- **Largest Contentful Paint (LCP):** 12.9 seconds

- **Cumulative Layout Shift (CLS):** 0.654

29
Performance

90
Accessibility

100
Best Practices

100
SEO

29

Performance

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

▲ 0–49   ■ 50–89   ● 90–100
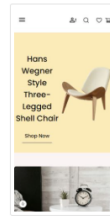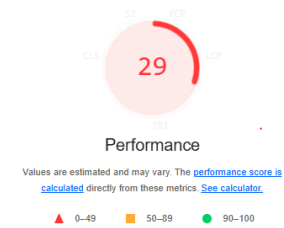
METRICS   Expand view

● First Contentful Paint
1.1 s

▲ Largest Contentful Paint
12.9 s

▲ Total Blocking Time
1,020 ms

▲ Cumulative Layout Shift
0.654

● Speed Index
2.7 s

▦ View Treemap

---

29   **90**   **100**   **100**

▦ View Treemap

Hans
Wegner
Style
Three-
Legged
Shell Chair
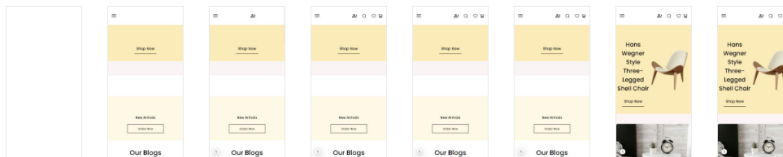
Show audits relevant to: **All**   FCP   LCP   TBT   CLS

DIAGNOSTICS

▲ Avoid large layout shifts — 3 layout shifts found   ⌄

▲ Largest Contentful Paint element — 12,940 ms   ⌄

▲ Reduce JavaScript execution time — 2.3 s   ⌄

▲ Minimize main-thread work — 4.2 s   ⌄

▲ Reduce unused JavaScript — Potential savings of 696 KiB   ⌄

▲ Minify JavaScript — Potential savings of 531 KiB   ⌄

▲ Preconnect to required origins — Potential savings of 370 ms   ⌄

! Page didn't prevent back/forward cache restoration — Error!   ⌄

■ Serve static assets with an efficient cache policy — 1 resource found   ⌄

29 90 100 100

	Serve static assets with an efficient cache policy — 1 resource found	⌄
	Eliminate render-blocking resources — Potential savings of 0 ms	⌄
	Avoid serving legacy JavaScript to modern browsers — Potential savings of 9 KiB	⌄
	Avoid long main-thread tasks — 14 long tasks found	⌄
	Minimize third-party usage — Third-party code blocked the main thread for 190 ms	⌄
	Initial server response time was short — Root document took 150 ms	⌄
	Avoids enormous network payloads — Total size was 2,172 KiB	⌄
	Avoids an excessive DOM size — 107 elements	⌄
	Avoid chaining critical requests — 1 chain found	⌄

More information about the performance of your application. These numbers don't directly affect the Performance score.

PASSED AUDITS (21)   Show

---

29 90 100 100

90

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.

BEST PRACTICES

| ⚠ Touch targets do not have sufficient size or spacing. | ⌄ |

These items highlight common accessibility best practices.

---

29 90 100 100

NAVIGATION

| ⚠ Heading elements are not in a sequentially-descending order | ⌄ |

These are opportunities to improve keyboard navigation in your application.

ADDITIONAL ITEMS TO MANUALLY CHECK (10)   Show

These items address areas which an automated testing tool cannot cover. Learn more in our guide on conducting an accessibility review.

PASSED AUDITS (22)   Show

NOT APPLICABLE (32)   Show

100

Best Practices

TRUST AND SAFETY

○ Ensure CSP is effective against XSS attacks

○ Use a strong HSTS policy

○ Ensure proper origin isolation with COOP

GENERAL

⚠ Missing source maps for large first-party JavaScript

PASSED AUDITS (14)   Show

Developer Resources for Furniture E-Commerce Website

Performance Monitor for Furniture E-Commerce Website