

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

Testing Report

	A	B	C	D	E	F	G	H	I
1	Test Case ID	Test Case Descr	Test Steps	Expected Result	Actual Result	Status	Severity Level	Assigned To	Remarks
2	TC001	Validate product	Open product pa	Products display	Products display	Passed	Low	Not Assigned	No issues found
3	TC002	Test API error ha	Disconnect API >	Show fallback UI	Error message s	Passed	Medium	Not Assigned	Handled gracefu
4	TC003	Check cart functi	Add product to c	Cart updates with	Cart updates as	Passed	High	Not Assigned	Works as expect
5	TC004	Ensure responsib	Resize browser	Layout adjusts pr	Responsive layo	Passed	Medium	Not Assigned	Test successful
6	TC005	Test dynamic rou	Open Product Pe	Open Dynamic f	Product details d	passed	Low	Not Assigned	Successfully

1. Test Case ID: TC001

Test Case Description: Validate Product Listing Page

Test Steps: Open Products Page > Validate Products

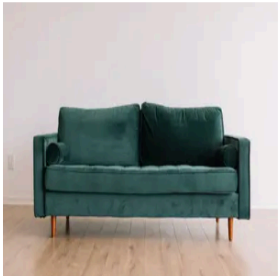
Expected Result: Products Display Correctly

Actual Result: Products Display Correctly

Status:passed

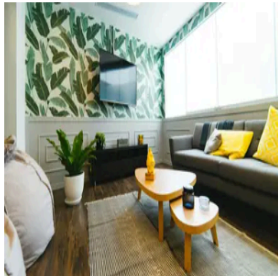
by Areesha khan

Severity Level: High
Assigned To: Not Assigned
Remarks: No Issue Found

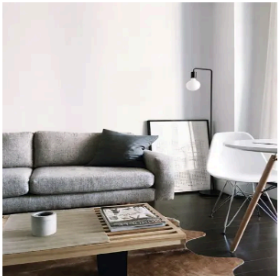


The Poplar suede sofa
980\$

popular products



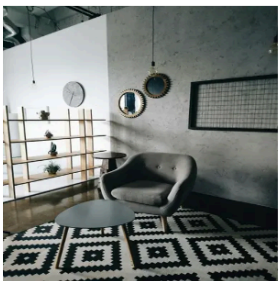
Tropical Vibe
550\$



Sleek Living
300\$



Serene Seat
350\$



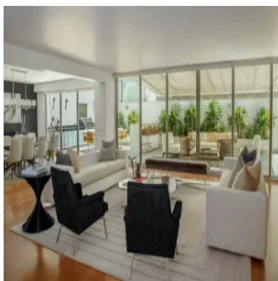
Nordic Elegance
280\$



TimberCraft
320\$



Timeless Elegance
320\$



Modern Serenity
480\$

by Areesha khan

2. Test Case ID: TC002

Test Case Description: Test API Error Handling

Test Steps: Disconnect APIs> Refresh Page

Expected Result: Show fallback UI with error message

Actual Result: Show Error Message on Screen

status:passed

Severity Level: Medium

Assigned To: Not Assigned

Remarks: Handle Gracefully

```
81 import { urlFor } from "@sanity/lib/image";
82
83 const Page = () => {
84   const [products, setProducts] = useState<ProductTypes[]>([]);
85   const [error, setError] = useState<string | null>(null);
86
87   useEffect(() => {
88     async function fetchData() {
89       try {
90         const fetchedData: ProductTypes[] = await client.fetch(allproducts);
91         setProducts(fetchedData);
92       } catch (err) {
93         console.error("Error fetching products:", err);
94         setError("Your Network issue. Please try refreshing the page.");
95       }
96     }
97     fetchData();
98   }, []);
99
100   if (error) {
101     // Fallback UI when an error occurs
102     return [
103       <div className="padding max-container mt-3 w-full min-h-screen flex items-center justify-center">
104         <div className="flex flex-col items-center justify-center text-center">
105           <h1 className="text-2xl font-bold text-red-500">Oops!</h1>
106           <p className="mt-2 text-gray-700">{error}</p>
107           <button
108             onClick={() => window.location.reload()}
109             className="mt-4 px-4 py-2 bg-blue-500 text-white rounded-lg hover:bg-blue-600 transition">
110             Refresh Page
111           </button>
112         </div>
113       </div>
114     ];
115   }
116 }
117
118 return (
119   <>
120     <div className="padding-x max-container">
121       <div className="max-lg:justify-items-center grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 xl:grid-cols-4 mt-6 gap-4 py-6">
122         {products.map((product) => (
123           <div key={product._id} className="product-card">
124             <Link
125               href={`/${sections}/${product.name
126                 .toLowerCase()
127                 .replace(/ /g, "-")`}
128             >
129             <div>
130               <div>
131                 <div>
132                   <div>
133                     <div>
134                       <div>
135                         <div>
136                           <div>
137                             <div>
138                               <div>
139                                 <div>
140                                   <div>
141                                     <div>
```

Oops!

Your Network issue. Please try refreshing the page.

Refresh Page

3. Test Case ID: TC003

Test Case Description: Check cart functionality

Test Steps: Add product to cart > Verify cart contents

Expected Result: Cart updates with added product

Actual Result: Cart updates as expected

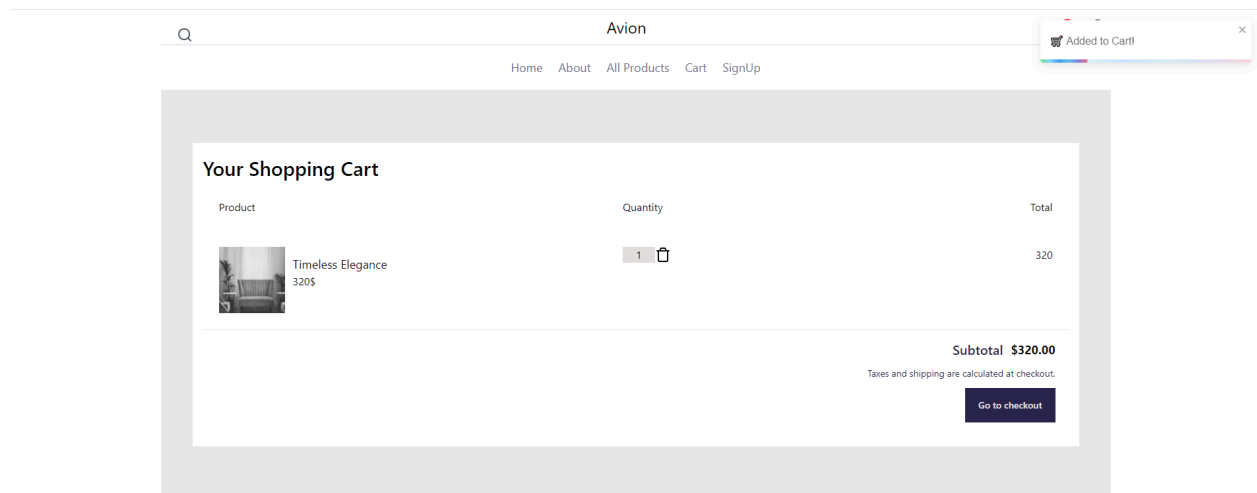
status: passed

Severity Level: Medium

Assigned To: Not Assigned

Remarks: Works as expected

by Areesha khan



4. Test Case ID: TC004

Test Case Description: Ensure responsiveness on mobile

Test Steps: Inspect > Device Toolbar > Select Mobile Device

Expected Result: Layout adjusts properly to screen size

Actual Result: Responsive layout working as intended

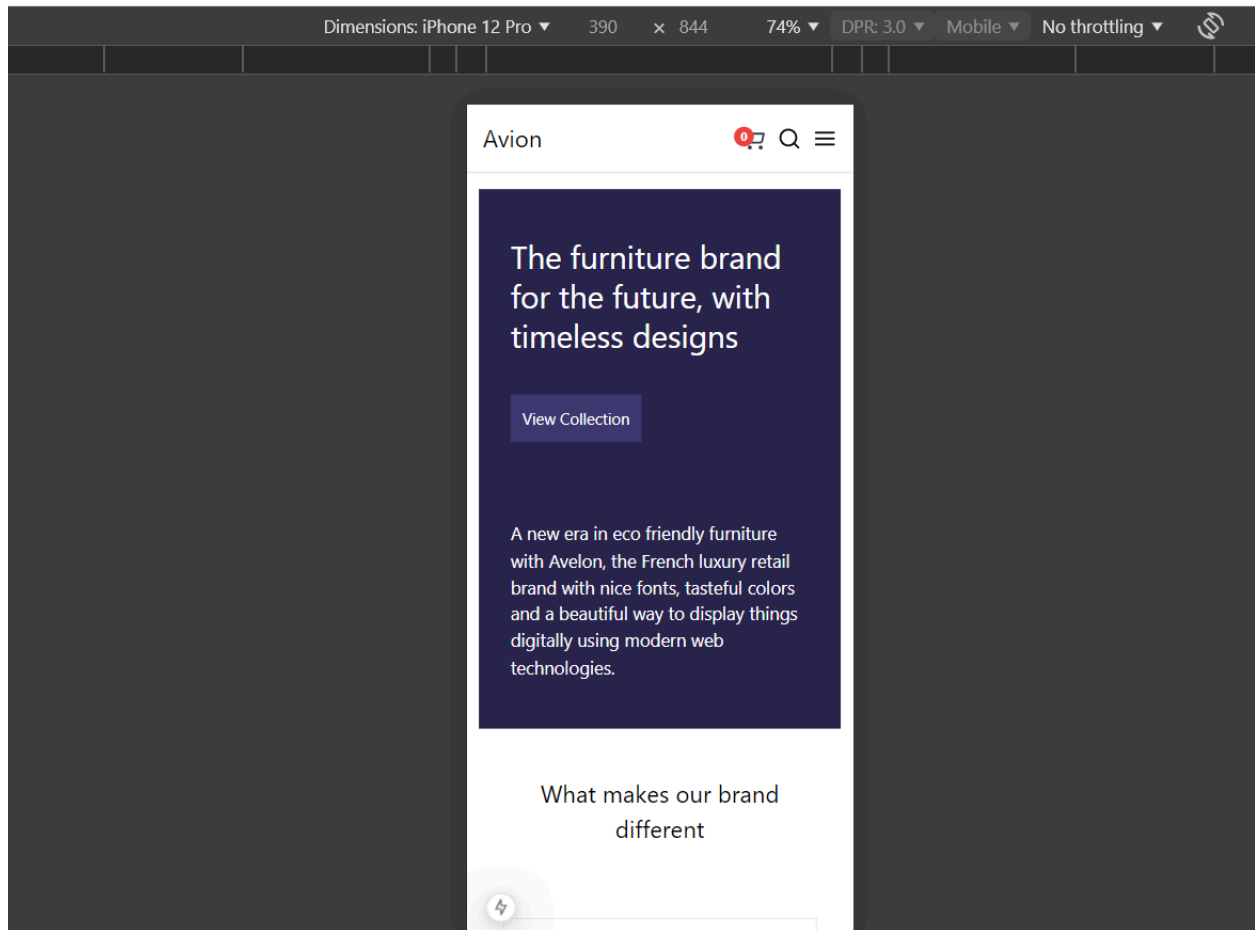
status: passed

Severity Level: Medium

Assigned To: Not Assigned

Remarks: Test successful

by Areesha khan



5. Test Case ID: TC005

Test Case Description: Test dynamic routing functionality

Test Steps: Open Product Page > Click to any product card

Expected Result: Open Dynamic Product Detail Pages

Actual Result: Product details displayed correctly based on URL structure

status: passed

Severity Level: Low

Assigned To: Not Assigned

Remarks: Successfully

by Areesha khan



TimberCraft

\$320

Description

A timeless design, with premium materials features as one of our most popular and iconic pieces. The dandy chair is perfect for any stylish living space with beech legs and lambskin leather upholstery.

Features:

- Premium material
- Handmade upholstery
- Quality timeless classic

popular products

Quantity : 50

Dimensions

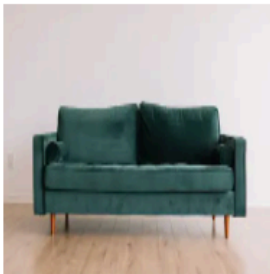
Height	Width	Depth
110cm	75cm	50cm

Quantity

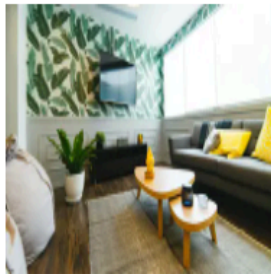
- 0 +

Add To Card

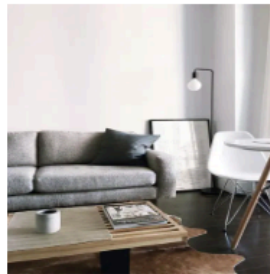
You might also like



The Poplar suede sofa
980\$



Tropical Vibe
550\$



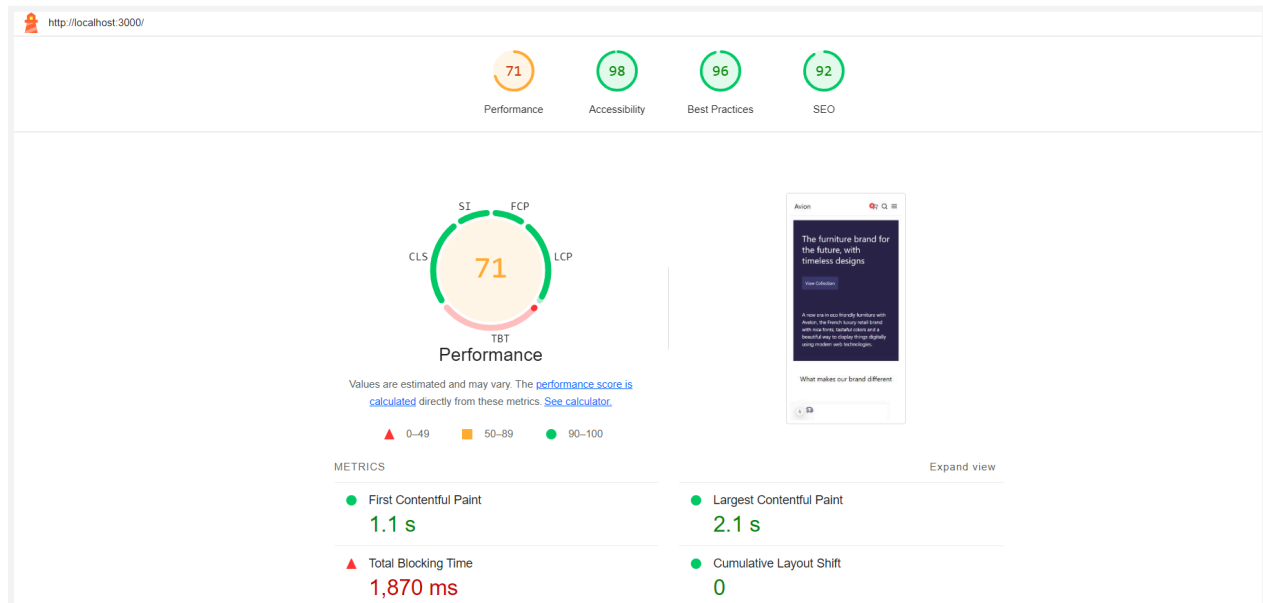
Sleek Living
300\$



Serene Seat
350\$

by Areesha khan

Performance Optimization steps taken:



by Areesha khan

Checklist for Day 5:

Functional Testing: ✓

Error Handling: ✓

Performance Optimization: ✓

Cross-Browser and Device Testing: ✓

Security Testing: ✓

Documentation: ✓

Final Review: ✓

by Areesha khan