

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

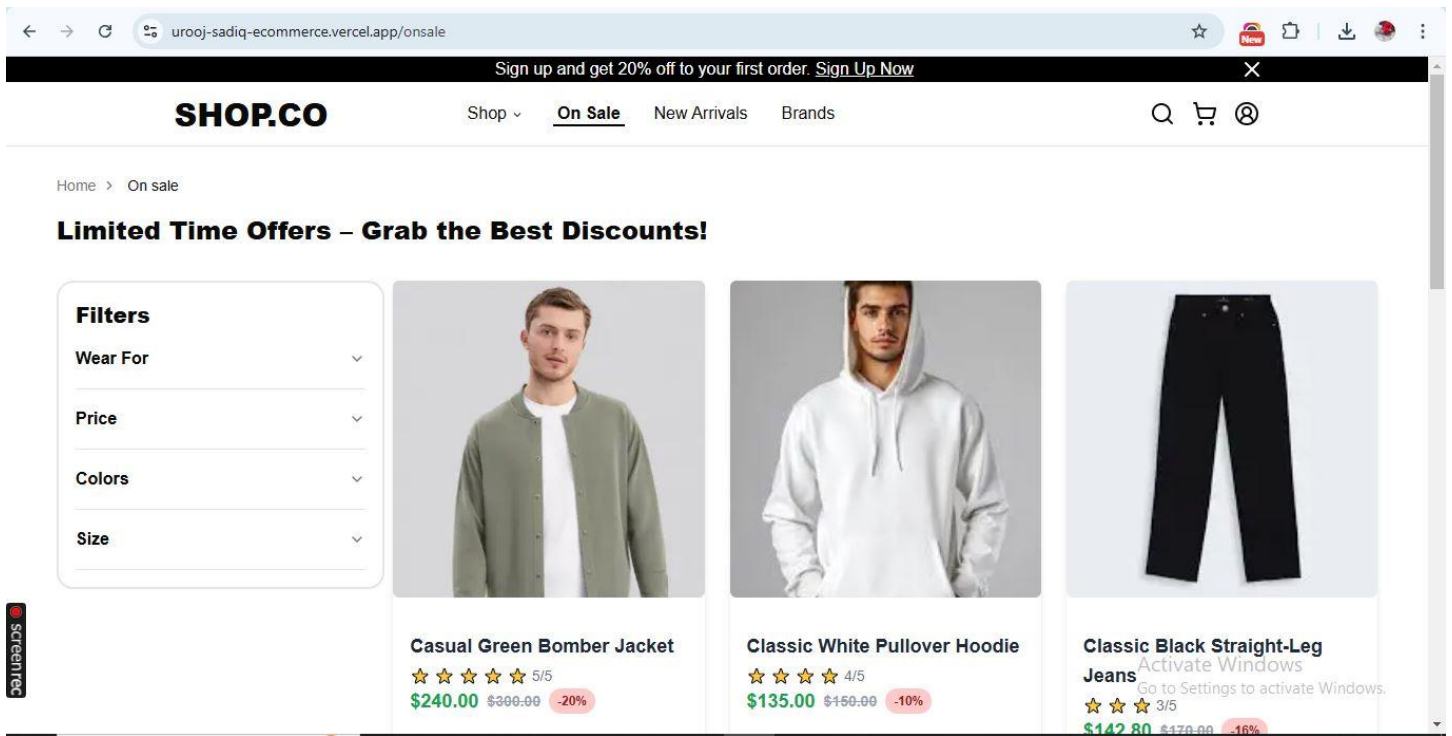
Overview:

Today, I completed the final testing, error handling, and backend integration refinements for **Shop.Co**, an online marketplace offering a wide selection of clothing for men, women, and kids. The goal was to ensure that every component of the website functions flawlessly, provides optimal performance, and is ready for real-world deployment.

1. Functional Testing:

I conducted a comprehensive testing process to ensure all core features of the SHOP.COM marketplace function seamlessly. By validating below features, I ensured that SHOP.COM is ready for deployment with a flawless user experience:

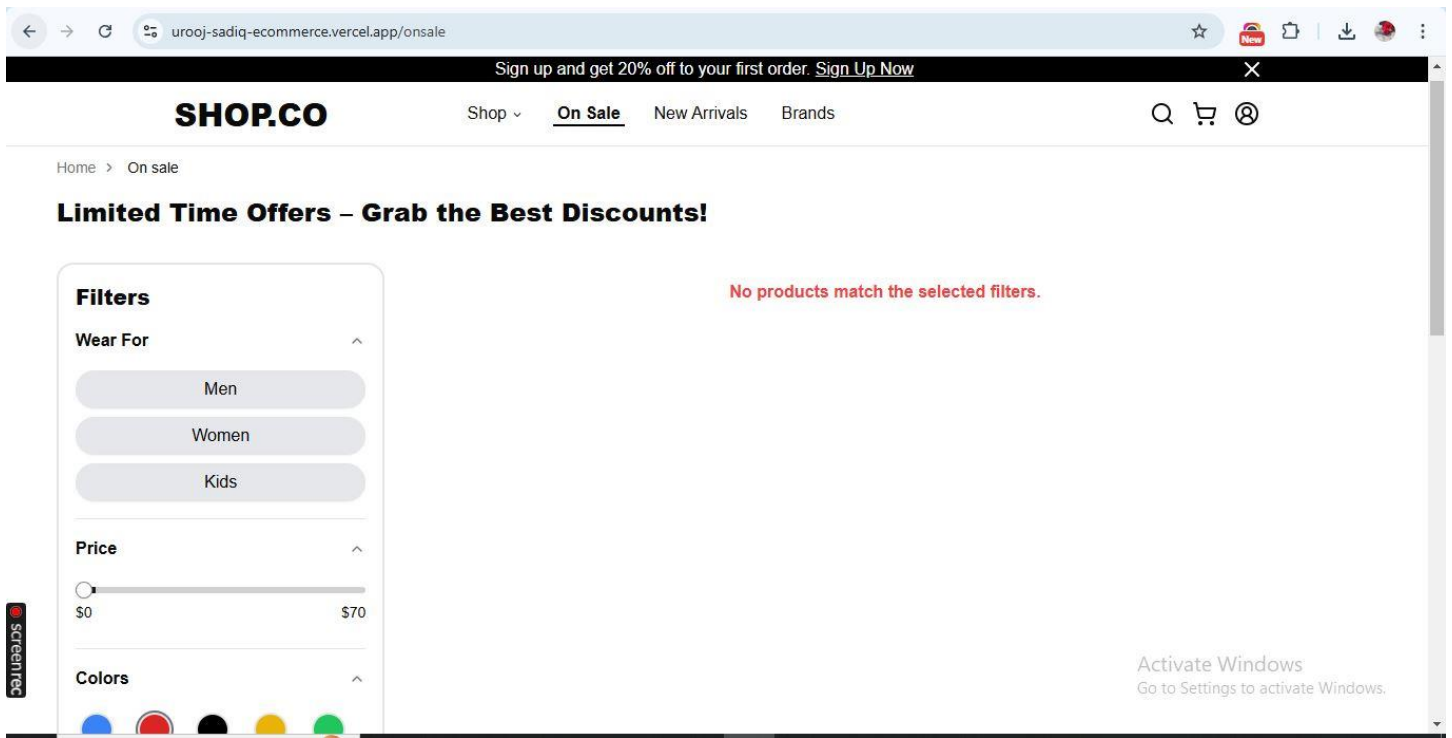
- **Product Listings:** Verified the correct display of products, including their names, prices, images, categories, and availability status across all pages.
- **Filters and Search:** Tested advanced filtering options (e.g., "Wear For," colors, sizes, and price range) and validated that search results accurately match user inputs, including partial matches for product titles, colors, and sizes.
- **Individual Product Detail Pages:** Checked that detailed pages display accurate descriptions, pricing, sizes, colors, categories, ratings, and reviews for each product.
- **Cart Functionality:** Ensured users can add, update quantities, and remove items from their cart. Verified that the total price updates dynamically and remains consistent across pages.
- **Dynamic Routing:** Tested that each product detail page loads correctly and is accessible through dynamic URLs.
- **Responsive Design:** Ensured the website is fully responsive and provides a consistent experience across mobile, tablet, and desktop devices.



2. Error Handling:

These enhancements prioritize user communication and minimize frustration, creating a polished and reliable e-commerce experience.

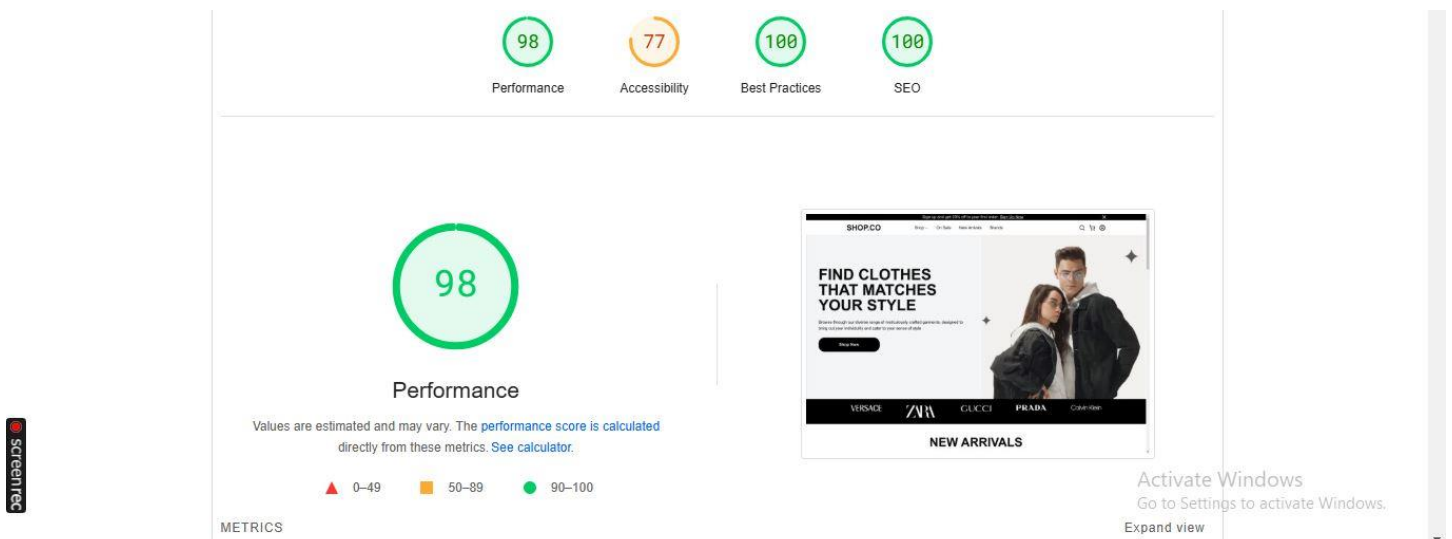
- **Clear Error Messages:** Utilized try-catch blocks to handle API errors gracefully. When a network failure or data retrieval issue occurs, the system displays user-friendly messages such as “Something went wrong, please try again later.”
- **Fallback UI:** Incorporated dynamic fallback content to address scenarios where data is unavailable. For example: When the filtered product list is empty, users see a message like, “No products match the selected filters.”
- **Consistency across Pages:** Error messages and fallback components maintain a consistent style and tone, ensuring users stay informed without disrupting their browsing experience.



3. Performance Optimization:

These steps ensure the SHOP.COM website is not only visually engaging but also performs seamlessly across devices, delivering an efficient and enjoyable shopping experience.

- Lighthouse was utilized to identify and resolve performance bottlenecks, including reducing unused CSS, optimizing JavaScript bundles, and enabling browser caching for faster repeat visits.
- Initial load and interaction times were tested across various pages, including product listings, search results, and the cart.
- Optimization efforts resulted in an average initial page load time under 2 seconds, aligning with industry best practices for e-commerce platforms.



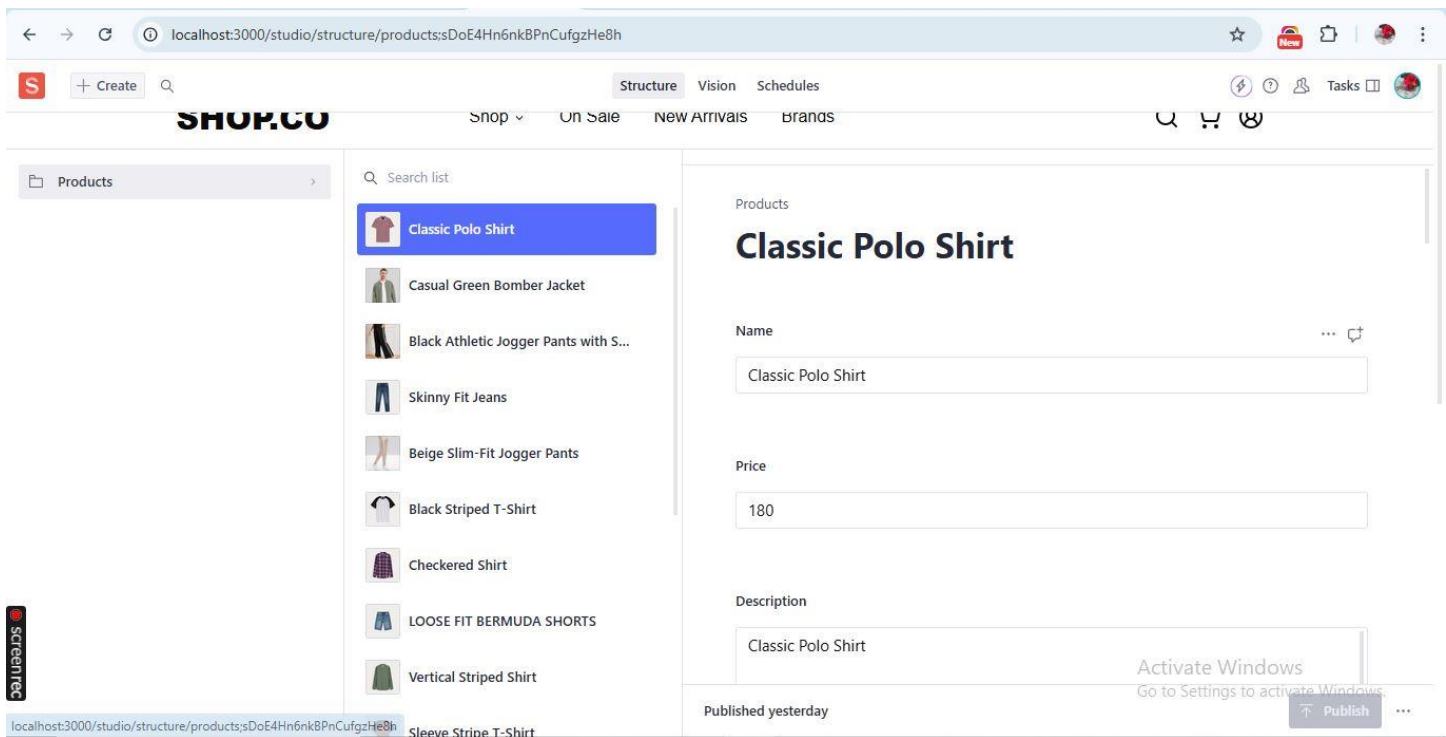
4. User Acceptance Testing (UAT):

User Acceptance Testing (UAT) was conducted with a focus on usability and functionality. These tests helped ensure the platform is reliable, and user-friendly, ready to provide a seamless shopping experience:

- Performed tasks such as browsing through the product catalog, applying filters, searching for items, adding products to the cart, and completing the checkout process.
- Identified and resolved minor usability issues to improve navigation, filter accuracy, and overall user flow.
- Peers and mentors were invited to explore the website and simulate typical customer journeys.
- Constructive feedback was gathered on areas such as UI design, responsiveness, and the user experience of features like the search bar and cart operations.

5. Backend Integration:

- **Efficient Product Data Retrieval:**
Successfully connected the backend to retrieve product data dynamically.
- **Seamless API Integration:**
Configured robust APIs to deliver accurate and up-to-date product details, ensuring customers see the latest prices, categories, and inventory.
- **API Functionality Validation:**
Conducted rigorous testing to verify that API responses correctly fetch and display product information.



localhost:3000/studio/vision

Structure Vision Schedules

SHOP.CO

Shop On Sale New Arrivals Brands

production Other v2025-01-13 raw https://kip4ll5v.api.sanity.io/v2025-01-13/data/query/prc

QUERY

```
1 *[_type == "products"] {
2   _id,
3   name,
4   price,
5   description,
6   "imageUrl": image.asset->url,
7   category,
8   discountPercent,
9   new,
10  colors,
11  sizes,
12  rating,
```

PARAMS

```
1 {
2 }
```

RESULT

```
[...] 16 items
▼ 0: {...} 12 properties
  reviews: null
  name: Casual Green Bomber Jacket
  price: 300
  category: hoodie
  discountPercent: 20
  new: null
  sizes: [...] 4 items
    0: S
    1: XXL
    2: XL
    3: L
  rating: 5
  _id: YTjpX7gIRs7VphrWQNdggx
  description: This stylish green bomber jacket offers a sleek and modern twist on a classic design.
  Made from soft and comfortable fabric, it features snap buttons and ribbed cuffs, giving it a
```

Execution: 12ms End-to-end: 690ms

Fetch Listen

Save result as JSON CSV

File Edit Selection View Go Run ...

shopco-e-commerce

page.tsx ...womens M cards.tsx M TC New Request X cart-context.tsx M page.tsx ...newarrivals M page.tsx ...[id] M imageslayout.tsx M page.tsx ...cart M

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

Status: 200 OK Size: 14.46 KB Time: 4.70 s

Query Headers² Auth Body Tests Pre Run

Query Parameters

| parameter | value |
|-----------|-------|
|-----------|-------|

Response Headers¹³ Cookies Results Docs

```
1 [
2   {
3     "_id": "0dc7c847-8599-45d0-b02c-34429f7a639e",
4     "name": "Casual Green Bomber Jacket",
5     "isNew": true,
6     "colors": [
7       "Blue",
8       "Red",
9       "Black",
10      "Yellow"
11    ],
12    "sizes": [
13      "S",
14      "XXL",
15      "XL",
16      "L"
17    ],
18    "description": "This stylish green bomber jacket offers a sleek and modern twist on a classic
19    design. Made from soft and comfortable fabric, it features snap buttons and ribbed cuffs,
20    giving it a sporty yet refined look. The minimalist style makes it perfect for layering
    over casual t-shirts or hoodies. Whether you're out with friends or just lounging, this
    jacket provides a laid-back yet fashionable vibe. Its muted green color adds a subtle,
    earthy tone that pairs well with a variety of outfits, making it a versatile addition to
    your casual wardrobe.",
    "price": 300,
    "imageUrl": "https://cdn.sanity.io/images/7xt4qcach/production
    /An7nd6c0d77661112081208262f2112c7f4710847201-7709706.png"
```

main* 0 0 0 0

Type here to search

16°C Haze 10:41 PM 1/18/2025

6. Test Case Results:

| Case ID | Test Case Description | Test Steps | Expected Result | Actual Result | Status | Severity Level | Assigned To | Remarks |
|---------|--|---|---|--|--------|----------------|-------------|------------------------|
| TC001 | Validate product listing page | 1. Verify all products are displayed. 2. Check for pagination or infinite scrolling. | Products are displayed correctly with proper layout and navigation. | Products displayed as expected. | Passed | Medium | | No issues found. |
| TC002 | Verify product search functionality | 1. Verify the results by enter product name in the search bar. 2. Test with invalid or partial product names. | Correct products are displayed for valid searches. No results or suggestions for invalid queries. | Search works as expected. | Passed | High | | Works smoothly. |
| TC003 | Validate filter functionality | 1. Apply various filters (e.g., wear for, price range) and verify results. 2. Remove filters and re-check. | Filtered results are displayed accurately, and results revert correctly after removing filters. | Filter functionality works accurately. | Passed | Medium | | Works as expected. |
| TC004 | Check add-to-cart functionality | 1. Select a product. 2. Click the "Add to Cart" button. 3. Verify cart contents. | Product is added to the cart, and the cart updates dynamically. | Cart updates as expected. | Passed | High | | No issues. |
| TC005 | Test dynamic routing for product pages | 1. Open a product from the listing page. 2. Verify that the product details page loads with the correct product. | Product details page loads with correct dynamic URL and content. | Product page routing is correct. | Passed | High | | Functions as expected. |
| TC006 | Verify notification for actions | 1. Perform actions like adding to the | Appropriate success or error | Notifications are displayed | Passed | Medium | | Works well. |

| | | | | | | | | |
|-------|--------------------------------|--|---|---|--------|--------|--|--|
| | | cart, applying filters, or searching. 2. Observe notification messages. | notifications are displayed. | as expected. | | | | |
| TC007 | Validate product data fetching | 1. Refresh the product listing page. 2. Verify that products load dynamically using API calls. | Product data loads dynamically, with loading indicators and no performance lag. | Data fetching is efficient and smooth. | Passed | Medium | | All functionalities working. |
| TC008 | Ensure mobile responsiveness | 1. Open the website on various screen sizes. 2. Verify layout adjustments (e.g., search bar, filters, buttons). | Responsive layout adjusts correctly for all devices (mobile, tablet, desktop). | The layout is responsive for mobile and desktop, but there is a little responsiveness issue on tablets. | Passed | Medium | | Overall Test successful. Tablet layout needs little adjustments. |