

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

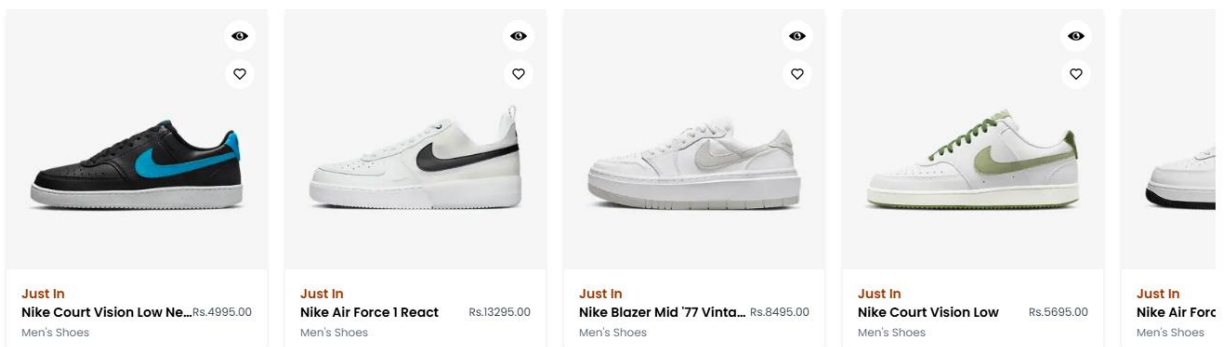
## STEP 1: FUNCTIONAL TESTING

### 1. Test Core Features:

#### • Product listing

Test code -ID	Test Case	Steps	Expected Outcome	Status
Tc00-1	Display product list	Open the product listing page	All products appear with correct details	PASS
Tc00-3	Image rendering	Verify product images load correctly	All images are displayed without distortion	PASS

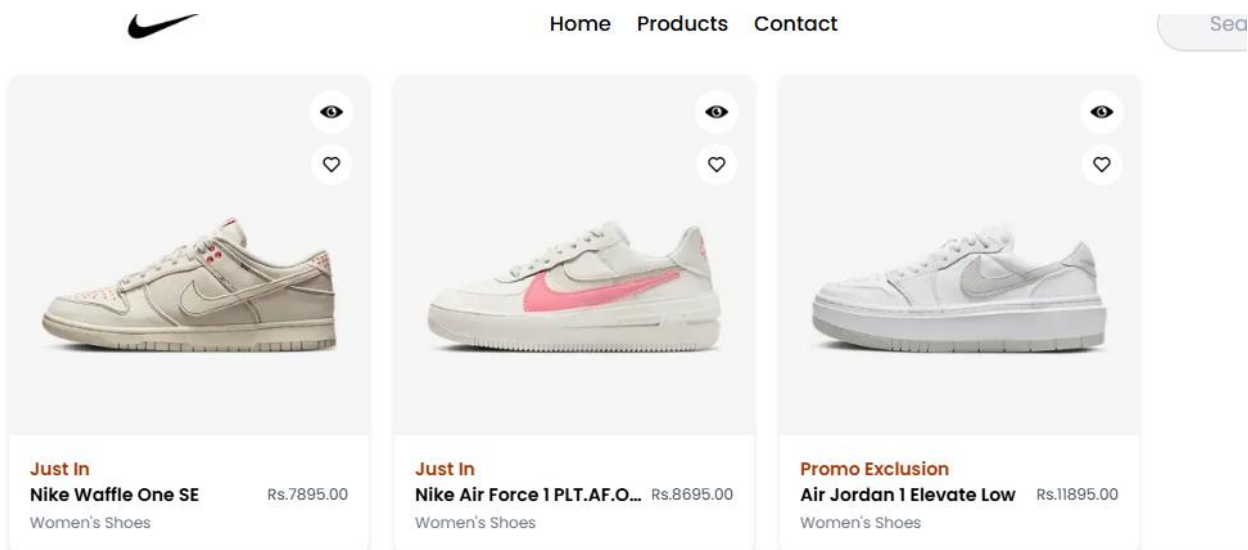
Mens Shoes



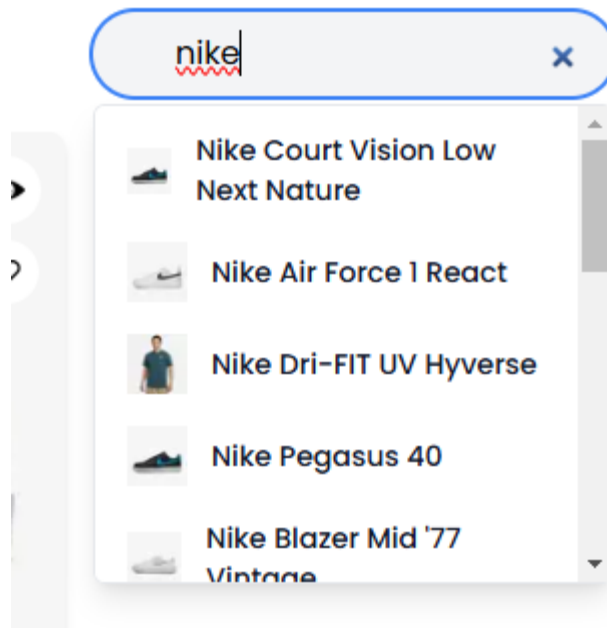
#### • FILTERS

Test code -ID	Test Case	Steps	Expected Outcome	Status
Tc00-4	Category filter	Select a specific category filter	only products within the selected category appear	PASS

Tc00-5	Keyword Search			PASS
--------	----------------	--	--	------



Test code -ID	Test Case	steps	Expected Outcome	st a t u s
	Combined filter and Seach	Enter a keyword [ skinny fit jeans]	only product matching the keyword appear	P A S S
Tc00-6		Apply multiple filters and a keyword search	Results meet all applied conditions	



## cart Operations\*

Test code - ID	Test Case	steps	Expected Outcome	s t a t u s
Tc00-7	product detail page load	Click on a shop from the listing page	products detail page opens with correct details	P A S S
Tc00-8	URL validation	Verify URL format (e.g., /shop/product-name)	URL matches the product ID	P A S S
Tc00-9	Navigation back to list	Use browser back button	Returns to the product listing page	P A S S
Tc00-10	Add/ remove items cart	Click on the add or remove button to perform function	It will add or remove items from user's cart	P A S S

## Order Summary



Nike Air Force 1 React

1 x Rs.13295.00

**Rs.13295.00**



Subtotal

Rs.13295.00

Shipping

Rs.5.99

**Total**

**Rs.13300.99**

## STEP 2: ERROR HANDLING

Add Error Messages:

Use try-catch blocks to handle API errors.

```
useEffect(() => {
  const fetchProducts = async () => {
    try {
      const query = `*[_type == "product" && category == "Men's
Shoes"]{
        description,
        inventory,
        colors[0],
        price,
        _id,
        "imageUrl":image.asset->url,
        status,
        productName,
        category,
        "slug":slug.current
      }`;
      const data = await client.fetch(query);
      setData(data);
      setLoading(false);
    } catch (error) {
      console.error(`Error fetching products for Men's Shoes:`, error);
    }
  }
}
```

```
        setLoading(false);
    }
};
fetchProducts();
}, []);
```

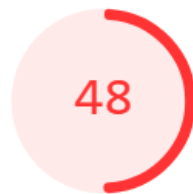
FALLBACK UI:

```
console.error(`Error fetching products for Men's Shoes:`, error);
```

## STEP 3: PERFORMANCE OPTIMIZATION

- Optimize Assets
- Analyze Performance
- Test Load Times

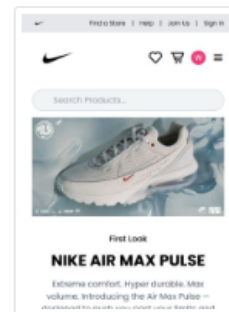
PERFORMANCE



### Performance

Values are estimated and may vary. The [performance score is calculated](#) directly from these metrics. [See calculator.](#)

▲ 0–49    ■ 50–89    ● 90–100

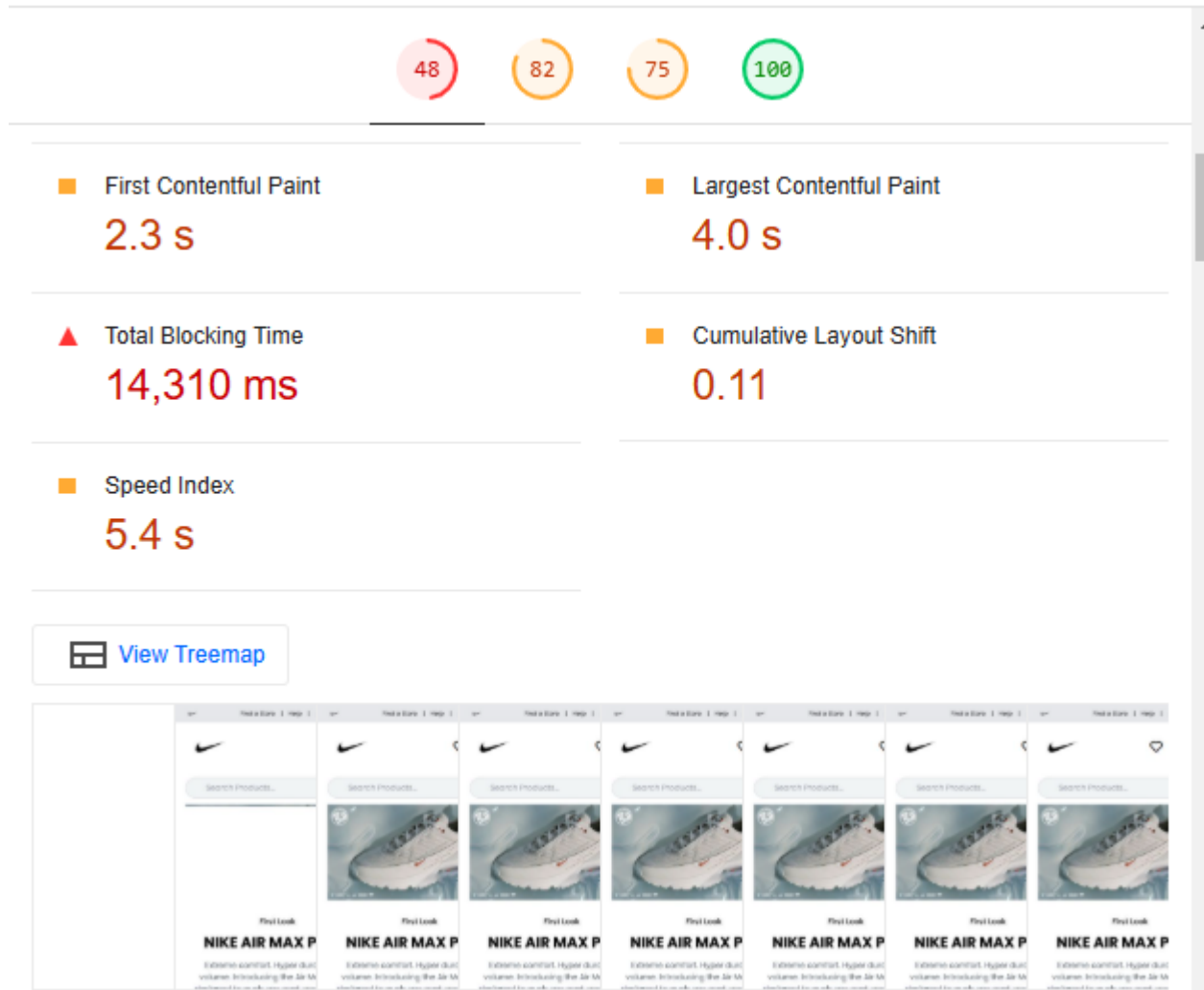


### METRICS

Expand view

■ First Contentful Paint  
**2.3 s**

■ Largest Contentful Paint  
**4.0 s**

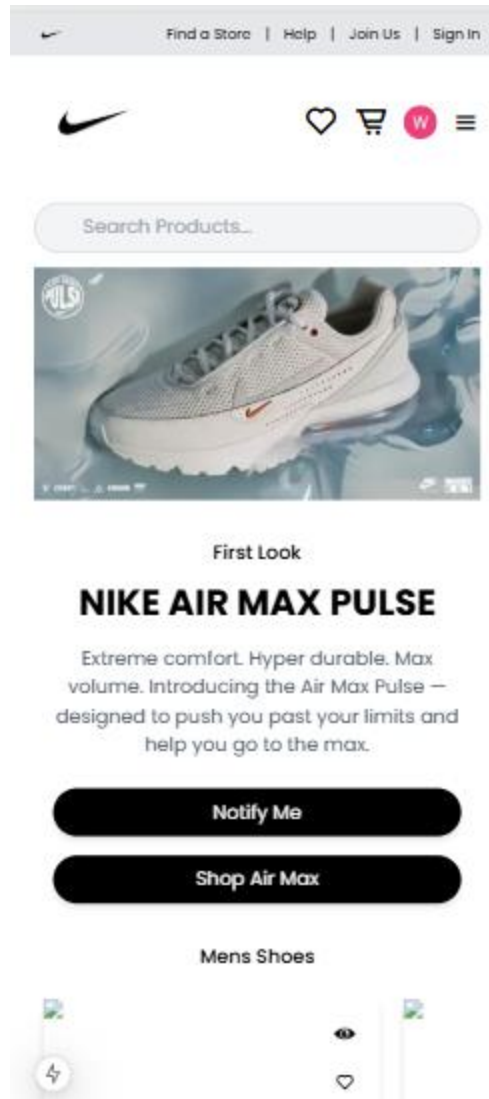


## STEP 4: CROSS-BROWSER AND DEVICE TESTING

1. Browser Testing
2. Device Testing

Test code - ID	Test Case	steps	Expected Outcome	status
Tc00-11	View on Google Chrome	Open the chrome and see website	It shows responsive	PASS
Tc00-12	View on Microsoft Edge	Open the Microsoft Edge and see website	It shows responsive	PASS
Tc00-13	View on Mobile	Open the Mobile and see website	It shows responsive	PASS

My website is responsive For All devices



## STEP 5: SECURITY TESTING

**DESCRIPTION: SECURE THE WEBSITE AGAINST VULNERABILITIES**

**KEY ACTIONS:**

- *Sanitize user inputs to prevent SQL injection and XSS attacks.*
- *Ensure API calls are made over HTTPS.*
- *Store sensitive information in environment variables.*
- *Conduct penetration testing to identify hidden vulnerabilities.*

## STEP 6: USER ACCEPTANCE TESTING (UAT)

**DESCRIPTION:** SIMULATE REAL-WORLD USER INTERACTIONS TO IDENTIFY USABILITY ISSUES.

**SCENARIOS TESTED:**

- *Browsing products.*
- *Adding and removing items from the cart.*
- *Completing the checkout process.*
- *Testing multi-step workflows to ensure an intuitive user experience.*

**FEEDBACK COLLECTED:**

- *Minor UI inconsistencies identified and resolved.*
- *Improved workflows for better user experience.*
- *Adjusted visual hierarchy to emphasize key actions like "Add to Cart."*

## STEP 7: DOCUMENTATION UPDATES

**DESCRIPTION:** COMPILE FINDINGS AND RESOLUTIONS INTO A PROFESSIONAL REPORT. INCLUDES:

- **Test case descriptions and results.**
- **Performance optimization steps.**
- **Security measures implemented.**
- **Screenshots of issues and fixes.**
- **Additional insights into areas for future improvement.**

CSV CONTENT

**Test Case ID,Description,Expected Result,Actual**

*Result,Status,Severity,Remarks TC001,Test navigation links,All links navigate correctly,All links function as intended,Pass,Low,None TC002,Verify product listing display,Products display correctly,Products display correctly,Pass,Medium,None TC003,Test shopping cart functionality,Items add/remove/update correctly,Cart functions as expected,Pass,High,None TC004,Check accessible,Pass,Low,None TC005,score ≥ 88,Score: 83,Fail,High,Optimization needed TC007,Check accessibility features,Accessibility score ≥ 94,Score: 100,Pass,Medium,Ensure ongoing compliance TC008,Evaluate SEO metrics,SEO score ≥ 100,Score: 79,Fail,Medium,Implement recommended SEO practices*



