

Day 5 - Testing and Backend

Refinement – Nike

Test cases executed and their results.

1. Test Case ID: TC001

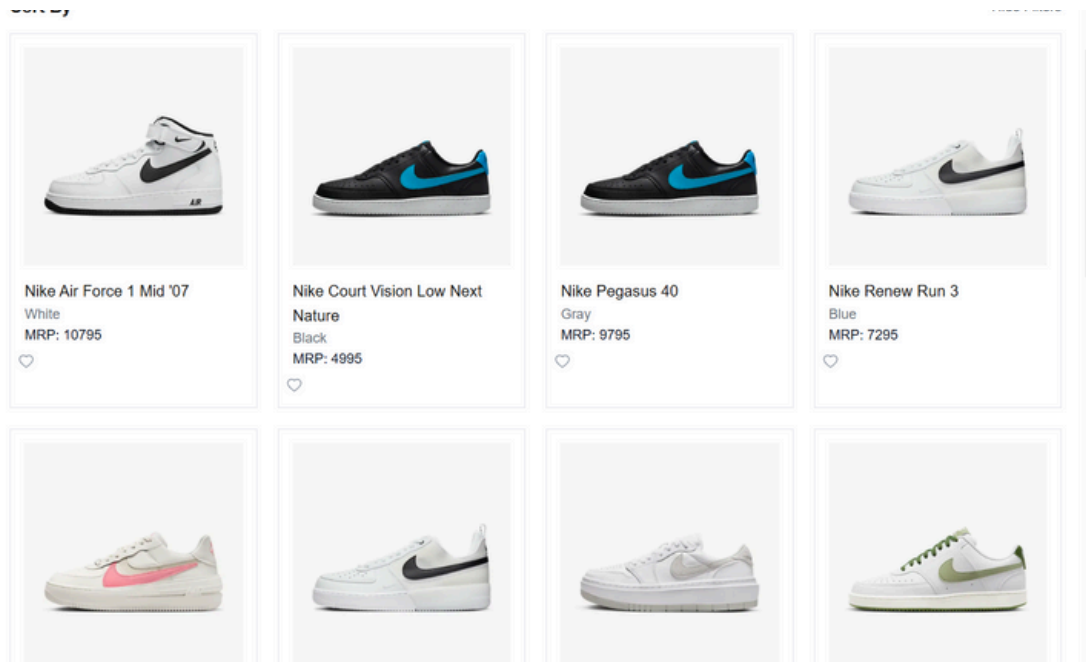
Test Case Description: Validate Product Listing Page Test

Steps: Open Products Page > Validate Products Expected

Result: Products Display Successfully Actual Result: Products

Display Successfully Severity Level: Low Assigned To: -

Remarks: No Issue Found



2. Test Case ID: TC002

Test Case Description: Test API Error Handling

Test Steps: Disconnect Gorq Query > Refresh Page

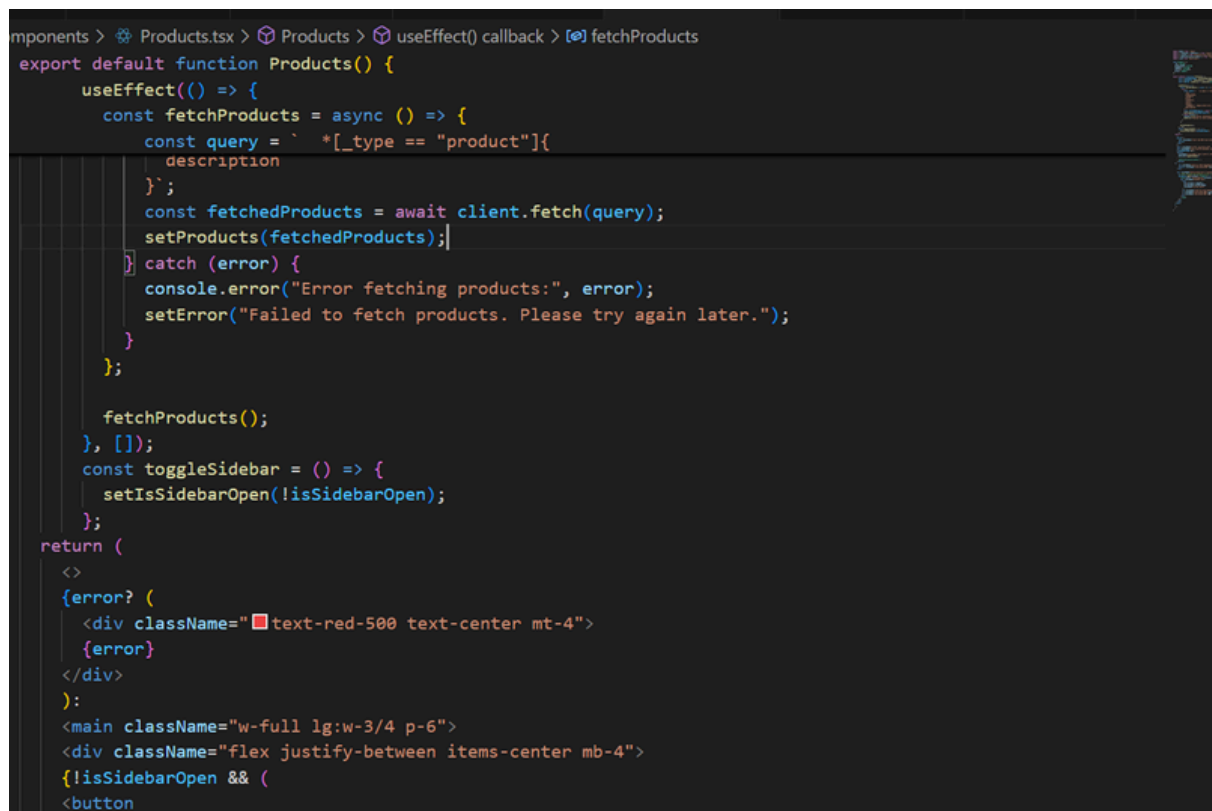
Expected Result: Show Error Message on Screen

Actual Result: Show Error Message on Screen

Severity Level: Low

Assigned To: -

Remarks: Handle Gracefully

A screenshot of a code editor (VS Code) showing a React component file named Products.tsx. The code uses useEffect to fetch products from an API. It includes a try-catch block to handle errors. In the catch block, it logs an error to the console and sets an error state with the message "Failed to fetch products. Please try again later.". The component also renders a UI element that displays the error message when an error occurs. The breadcrumb at the top indicates the file path: components > Products.tsx > Products > useEffect() callback > fetchProducts.

```
components > Products.tsx > Products > useEffect() callback > fetchProducts
export default function Products() {
  useEffect(() => {
    const fetchProducts = async () => {
      const query = `*[_type == "product"]{
        description
      }`;
      const fetchedProducts = await client.fetch(query);
      setProducts(fetchedProducts);
    }
  }) catch (error) {
    console.error("Error fetching products:", error);
    setError("Failed to fetch products. Please try again later.");
  }
};

  fetchProducts();
}, []);
const toggleSidebar = () => {
  setIsSidebarOpen(!isSidebarOpen);
};
return (
  <>
    {error? (
      <div className="text-red-500 text-center mt-4">
        {error}
      </div>
    ) :
    <main className="w-full lg:w-3/4 p-6">
      <div className="flex justify-between items-center mb-4">
        {!isSidebarOpen && (
          <button
```

Failed to fetch products. Please try again later.

3. Test Case ID: TC003

Test Case Description: Check Cart Functionality

Test Steps: Add To Cart > Navigate To Cart

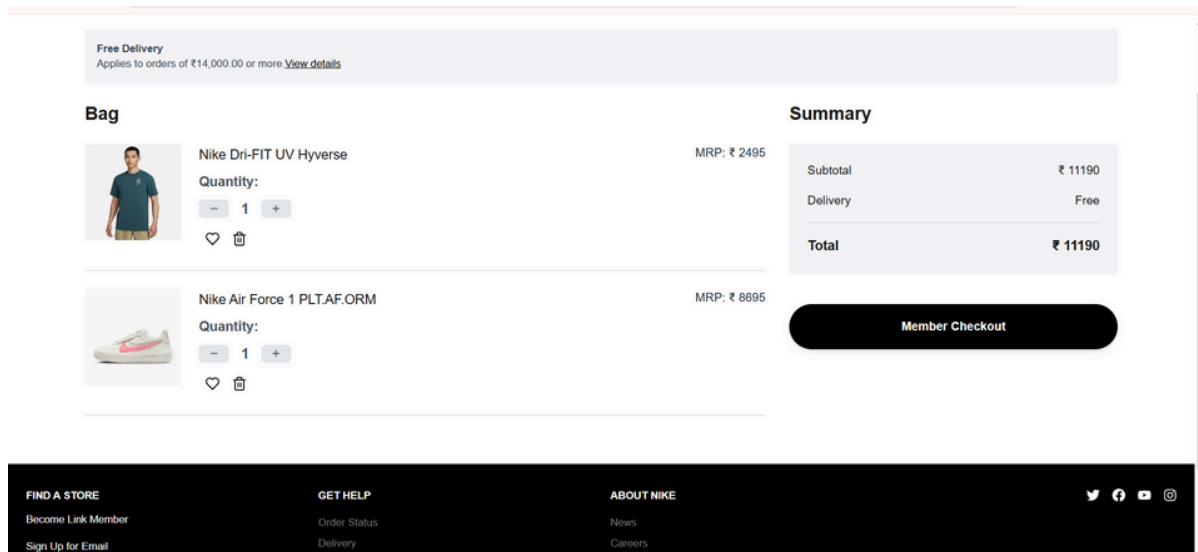
Expected Result: Displaying Products On Cart

Actual Result: Displaying Products On Cart

Severity Level: Medium

Assigned To: -

Remarks: Successfully



4. Test Case ID: TC004

Test Case Description: Ensure Responsiveness On Mobile

Test Steps: Inspect > Device Toolbar > Select Mobile Device

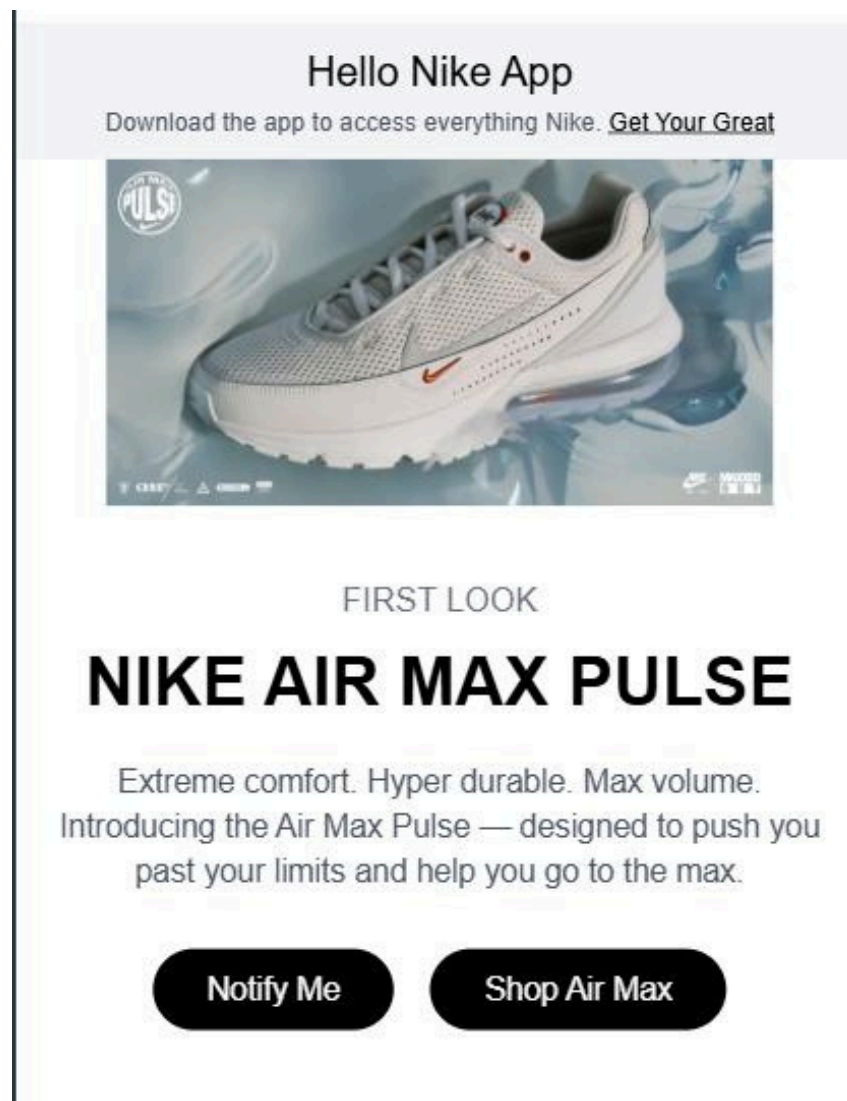
Expected Result: Showing Content according to the device

Actual Result: Showing Content according to the device

Severity Level: Medium

Assigned To: -

Remarks: Responsive



5. Test Case ID: TC005

Test Case Description: Validate Search Bar

Test Steps: Search Those Items which has not in sanity

Expected Result: Showing Dropdown > No Product Found

Actual Result: Dropdown not showing while searching irrelevant items

Severity Level: High

Assigned To: -

Remarks: Don't show error



6. Test Case ID: TC006

Test Case Description: Validate Dynamic Product Detail Pages

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

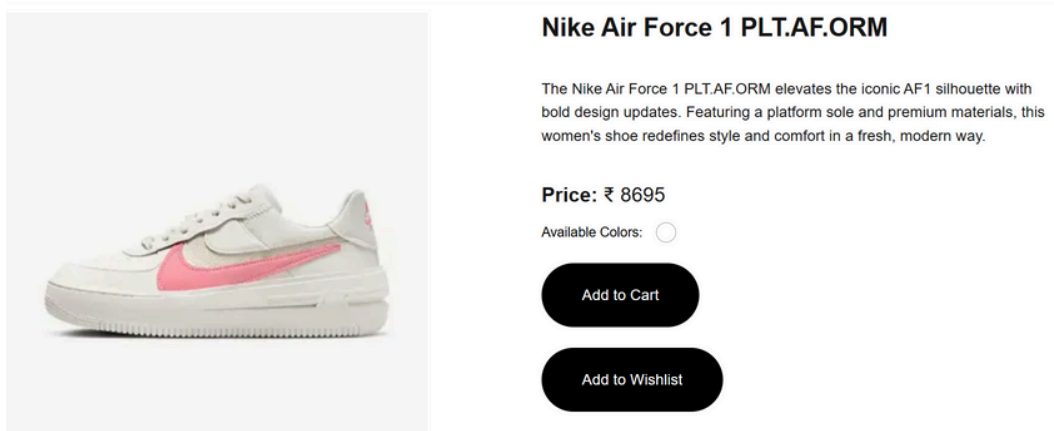
Expected Result: Open Dynamic Product Detail Pages

Actual Result: Open Dynamic Product Detail Pages

Severity Level: Low

Assigned To: -

Remarks: Successfully



7. Test Case ID: TC007

Test Case Description: Cross Browsing

Test Steps: Open our website on different Browsers such as Yahoo

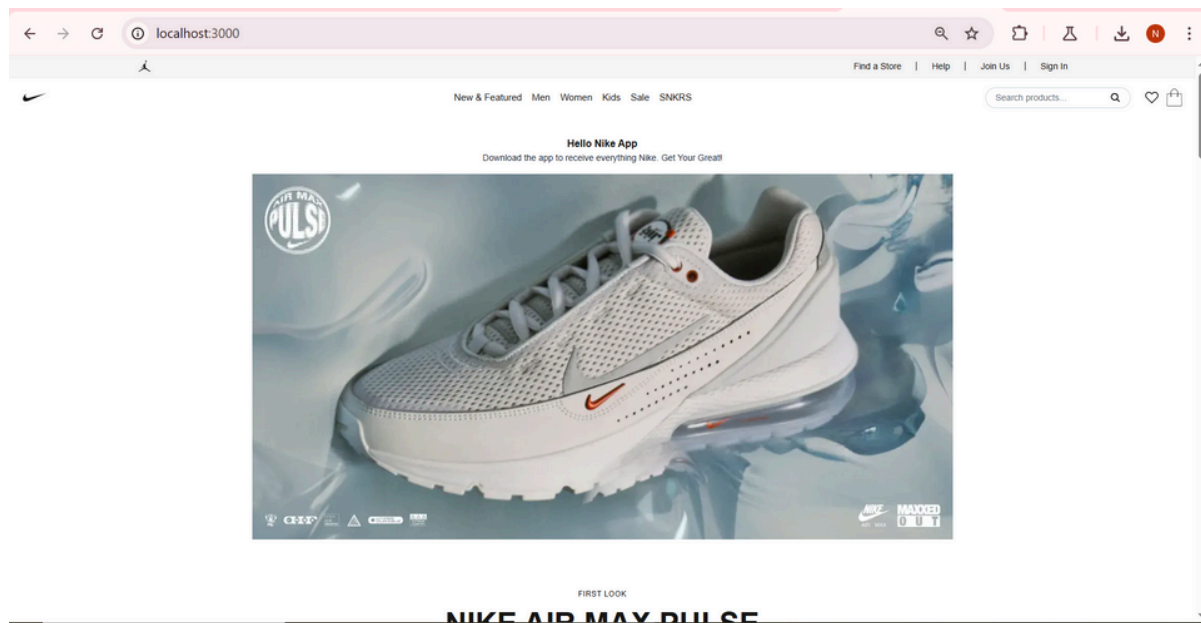
Expected Result: Working Correctly on every browser

Actual Result: Working Correctly on every browser

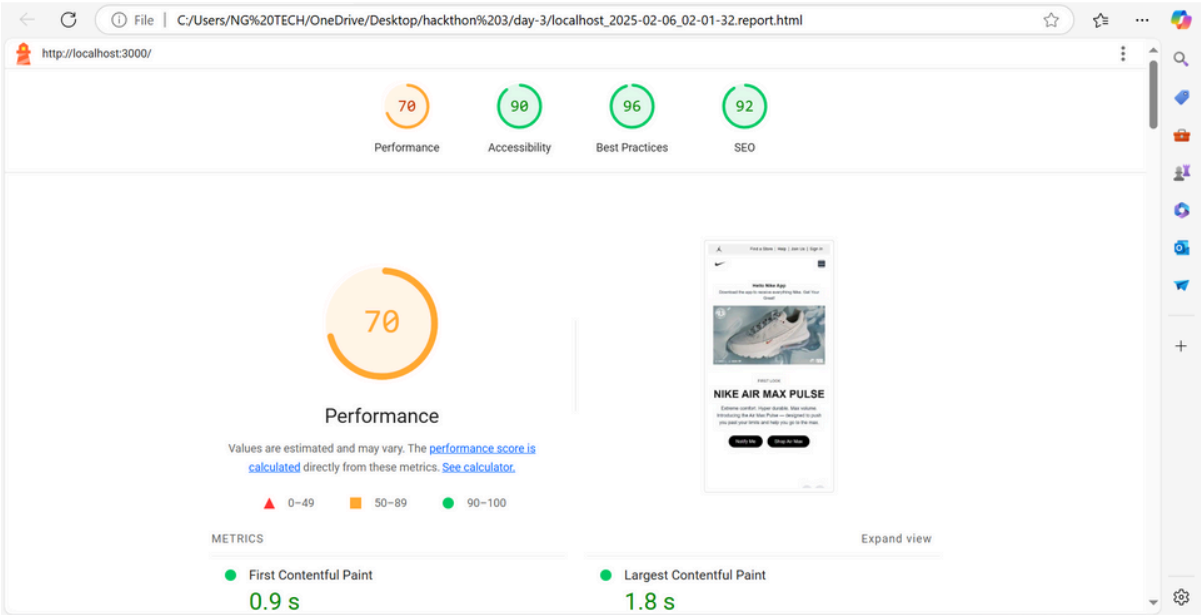
Severity Level: Low

Assigned To: -

Remarks: Successfully



Performance optimization steps taken



Checklist

Functional

	Error Handling	Performance Optimization	Cross-Browser	Security Testing	Documentation	Final Review
✓	✓	✓	✓	✓	✓	✓