

Name: Noor-ul-sehar

Slot: Friday 9 to 12

Roll #: 00349943

Day 5 - Testing, Error Handling, and Backend Integration Refinement

Introduction:

This document outlines the essential steps taken on Day 5 of my project, focusing on testing, error handling, and backend integration refinement. The objective is to prepare the marketplace for deployment by performing comprehensive testing, optimizing performance, and ensuring a seamless user experience across multiple platforms.

Steps for Implementation:

Step 1: Functional Testing

1. Test Core Features:

a. Product Listing:

Description:

Verify that products are displayed correctly on the marketplace, including product name, image, description, and price.

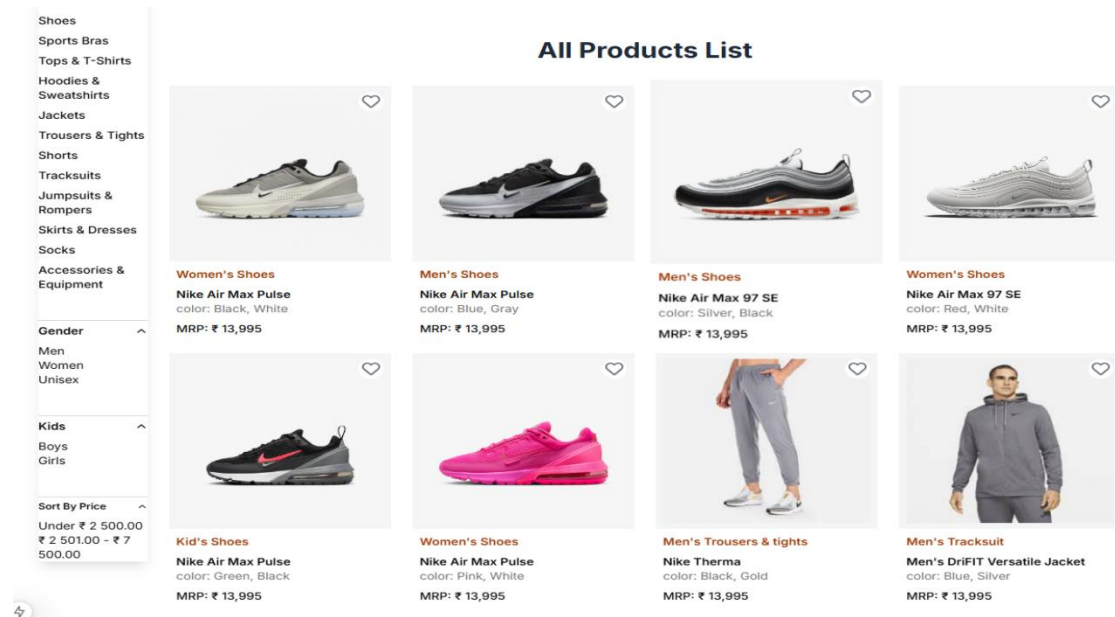
Expected Outcome:

The products should load dynamically and in the right categories.

Action Taken:

Checked various product listings to confirm all elements were displayed properly.

Screenshots:



b. Filters and Search:

Description:

Test filters (e.g., price range, categories, and ratings) to confirm that only relevant products appear when applied.

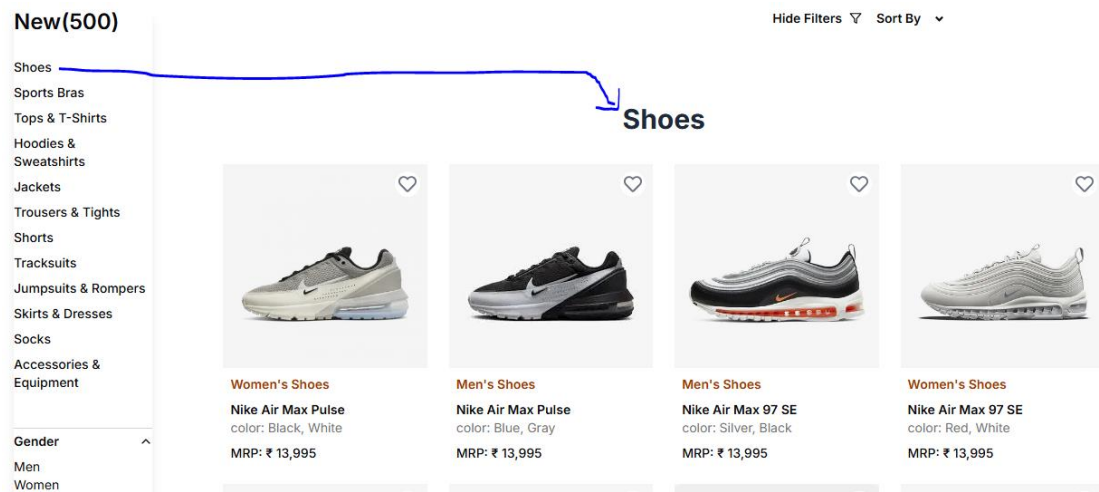
Expected Outcome:

The filters should work correctly and display only the relevant products.

Action Taken:

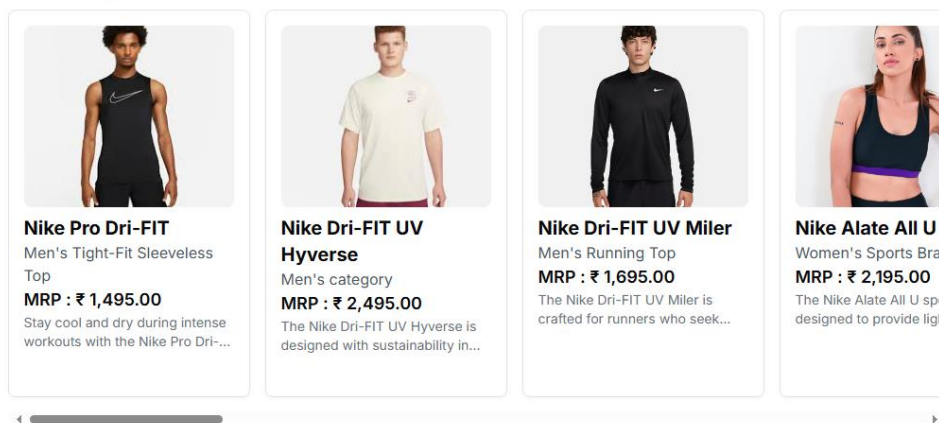
Applied various filters and searched for products to validate results.

Screenshots:

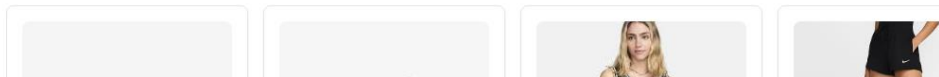


Filtered Products

Under ₹2,500.00



₹2,501.00 - ₹7,500.00



c. Cart Operations:

Description:

Ensure the cart operations (Add, Update, Remove) are functional and work as expected.


- i. **Add:** Test adding items to the cart.
- ii. **Update:** Test updating quantities in the cart.
- iii. **Remove:** Test removing items from the cart.

Expected Outcome:

Items should be added, updated, and removed correctly from the cart.

Action Taken:
Added items, updated quantities, and removed items from the cart.

Screenshots:



Nike Air Max Pulse

Men's Shoes

Color: Blue, Gray

Crafted for all-day wear, the Nike Air Max Pulse Men's Shoes offer superior comfort with a durable design. Featuring breathable mesh and a cushioned sole, these shoes are perfect for both casual outings and active lifestyles. Whether you're hitting the gym or running errands, the Nike Air Max Pulse ensures a supportive fit and a stylish look that stands out.

₹ 13,995

♥ Add to Wishlist

✓ Add to Cart

Shopping Cart



Nike Air Max Pulse
Add to Favorites

- 3 +

\$13995 (Original: \$13995)
Total: \$41985.00

🗑

Order Summary

Nike Air Max Pulse (x3)	₹ 41985.00
Subtotal	₹ 41985.00
Delivery Charges	₹ 200.00
Total to Pay	₹ 42185.00

Proceed to Checkout

Continue Shopping →

Shopping Cart



Nike Air Max Pulse
Add to Favorites

- 1 +

\$13995 (Original: \$13995)
Total: \$13995.00

🗑

Order Summary

Nike Air Max Pulse (x1)	₹ 13995.00
Subtotal	₹ 13995.00
Delivery Charges	₹ 200.00
Total to Pay	₹ 14195.00

Proceed to Checkout

Continue Shopping →

Shopping Cart



Nike Air Max Pulse
Add to Favorites

- 1 +

\$13995 (Original: \$13995)
Total: \$13995.00

Are you sure?
Yes No

Order Summary

Nike Air Max Pulse (x1) ₹ 13995.00

Subtotal ₹ 13995.00

Delivery Charges ₹ 200.00

Total to Pay ₹ 14195.00

[Proceed to Checkout](#)

[Continue Shopping →](#)

Shopping Cart

Your cart is empty!

Order Summary

Subtotal ₹ 0.00

Delivery Charges ₹ 200.00

Total to Pay ₹ 200.00

[Proceed to Checkout](#)

[Continue Shopping →](#)

d. Cart Summary:

Description:

Check that the cart reflects the correct number of items and the updated total price.

Expected Outcome:


The cart should display an accurate total and item count.

Action Taken:

Verified the summary after adding/removing items.

Screenshots:


Shopping Cart




Nike Air Max Pulse
Add to Favorites

- 1 +

\$13995 (Original: \$13995)
Total: \$13995.00






Nike Air Max Pulse
Add to Favorites

- 1 +

\$39200 (Original: \$39200)
Total: \$39200.00



Order Summary

Nike Air Max Pulse (x1) ₹ 13995.00

Nike Air Max Pulse (x1) ₹ 39200.00

Subtotal ₹ 53195.00

Delivery Charges ₹ 200.00

Total to Pay ₹ 53395.00

[Proceed to Checkout](#)

[Continue Shopping →](#)

e. Dynamic Routing:

Description:

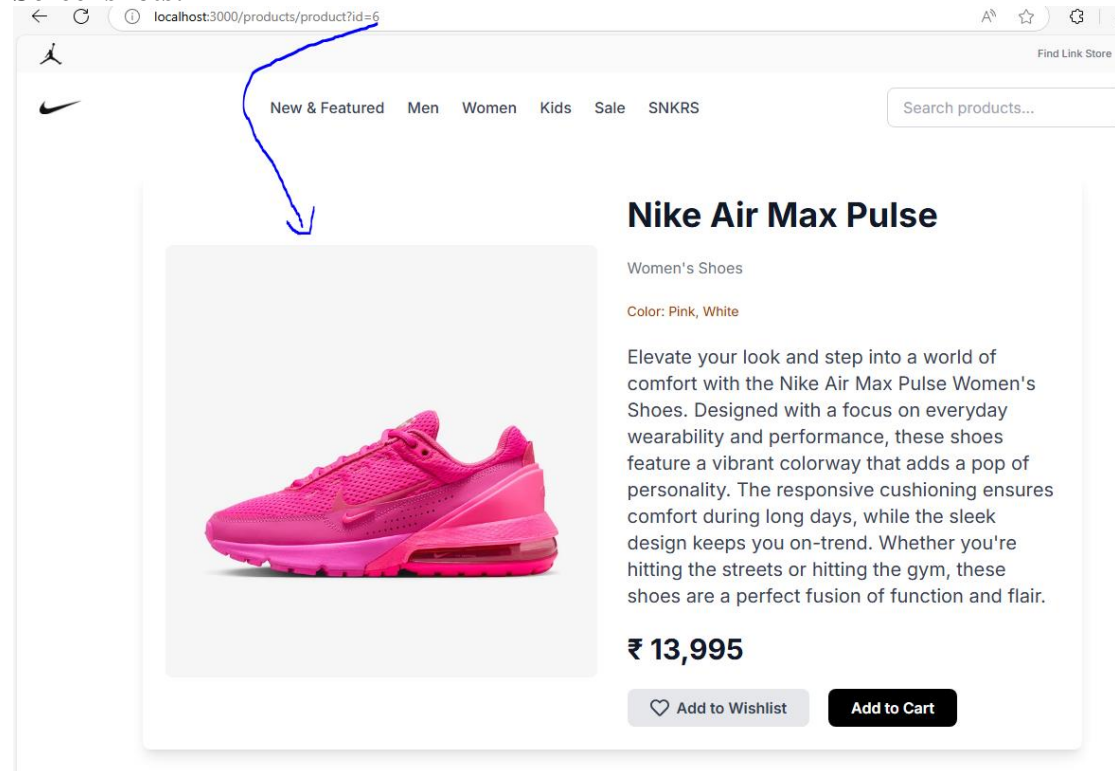
Ensure individual product detail pages load correctly when clicked.

Expected Outcome:

Clicking a product in the list should direct to the correct product page.

Action Taken:

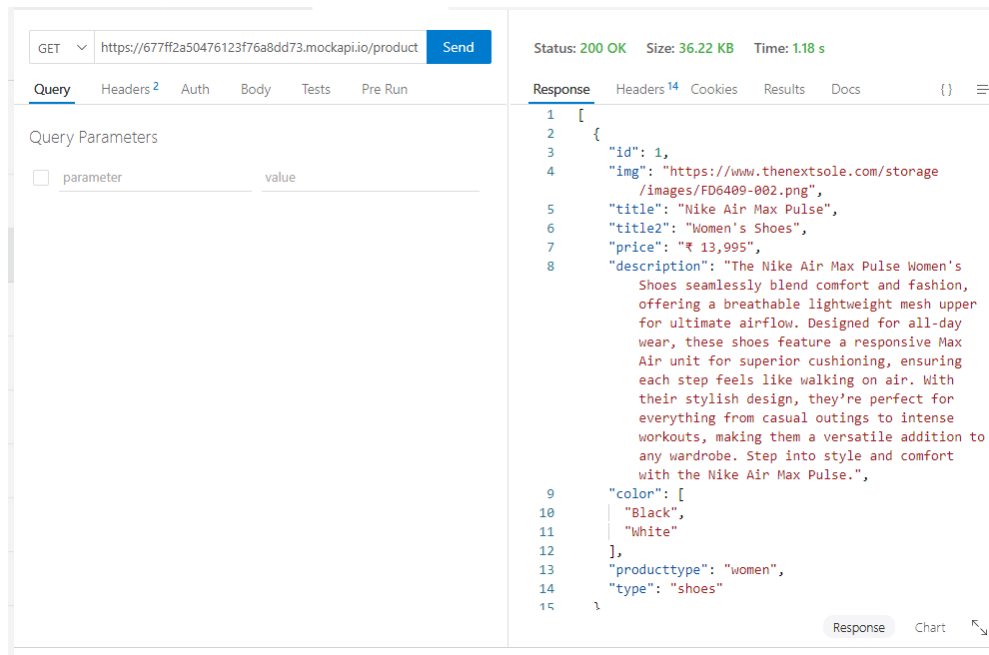
Tested dynamic routes and verified product details page functionality.

Screenshots:**2. Testing Tools:****a) Postman:****Purpose:**

Used for API response testing to ensure that the server responds with the expected results for various endpoints.

Use Case:

It allows sending requests to the API and inspecting the responses, such as checking for status codes, response times, headers, and body content.



Step 2: Error Handling

Error Handling:

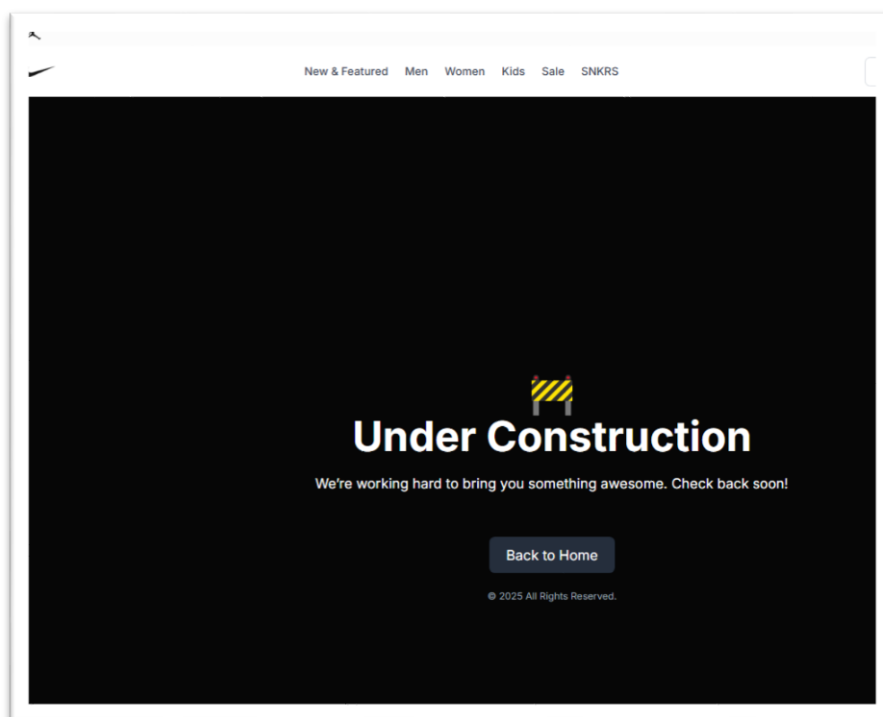
The try-catch block is used inside the `fetchProducts` function to handle any errors that might occur during the API call. If an error occurs, the `setError` function is triggered to display a friendly error message to the user.

```
// Fetch product details and related products
useEffect(() => {
  const fetchProduct = async () => {
    try {
      const response = await fetch("https://677ff2a50476123f76a8dd73.mockapi.io/product");
      if (!response.ok) {
        throw new Error("Failed to fetch product details");
      }
      const data: Product[] = await response.json();
      const selectedProduct = data.find((item: Product) => item.id === parseInt(id!, 10));
      if (selectedProduct) {
        setProduct(selectedProduct);
        const related = data.filter(
          (item: Product) =>
            item.producttype === selectedProduct.producttype && item.id !== selectedProduct.id
        );
        setRelatedProducts(related.slice(0, 4)); // Show up to 4 related products
      } else {
        setError("Product not found.");
      }
    } catch (error) {
      setError(error instanceof Error ? error.message : "Failed to fetch product details");
    } finally {
      setLoading(false);
    }
  };
});
```

Fallback UI:

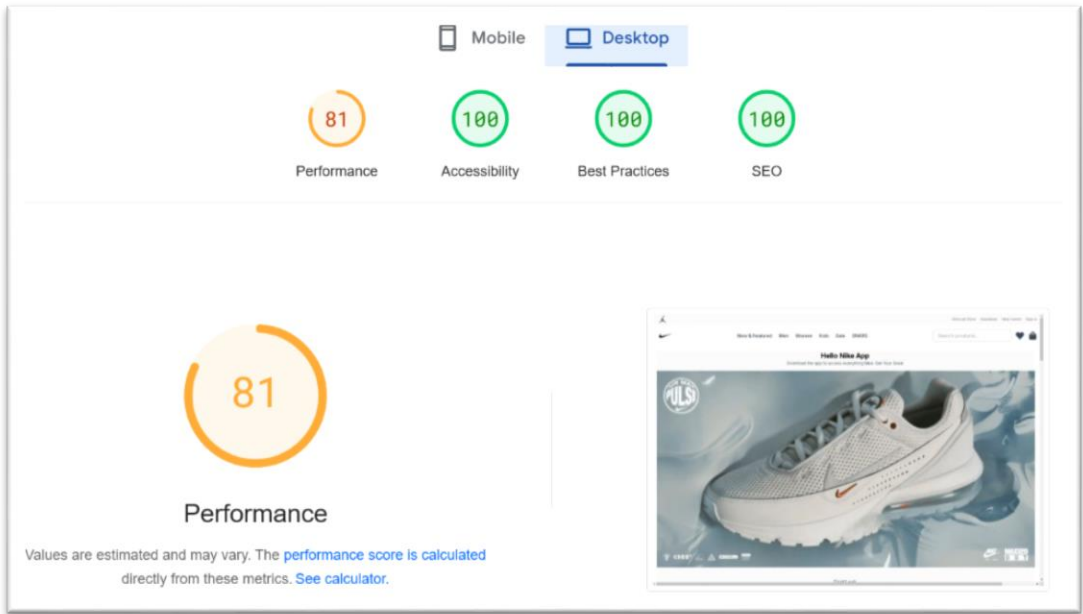
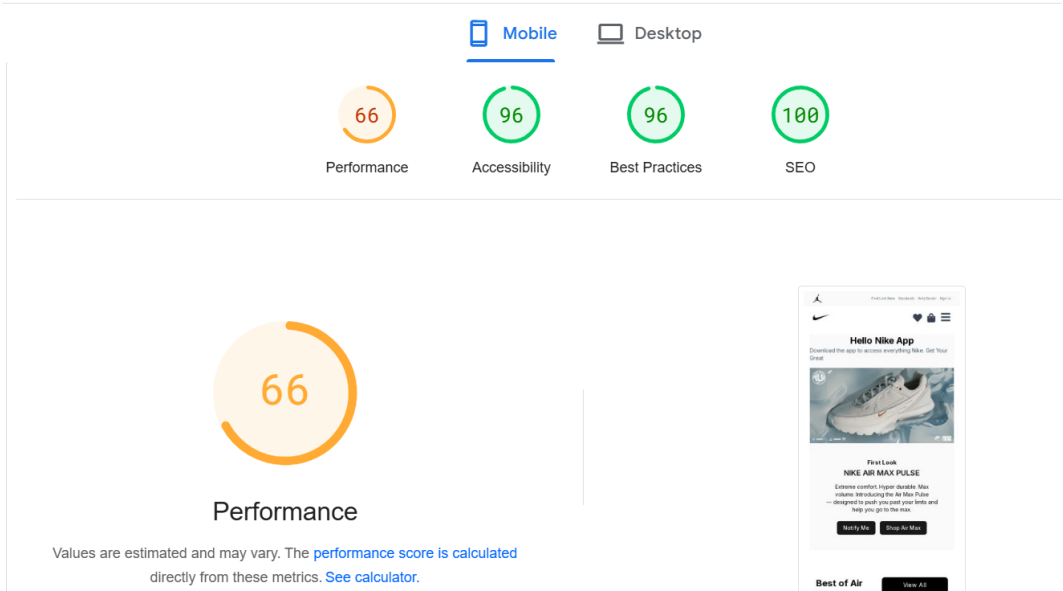
If the products array is empty, a message "No items found" is shown. If there's an error, an error message is displayed to inform the user that the products couldn't be load.

Not found page



Step 3: Performance Optimization

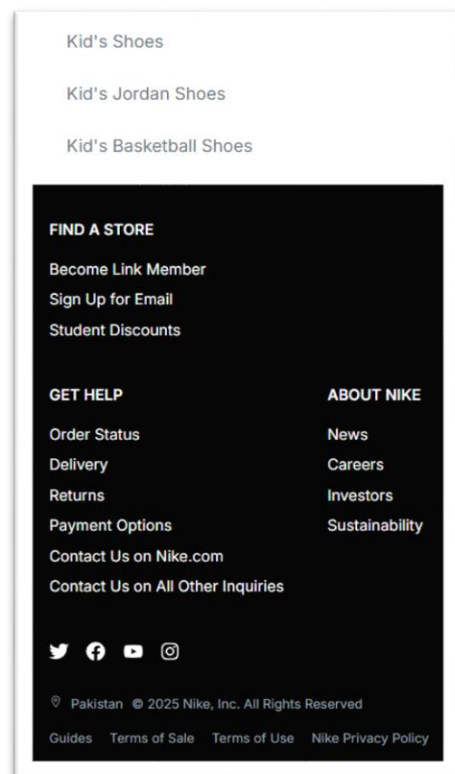
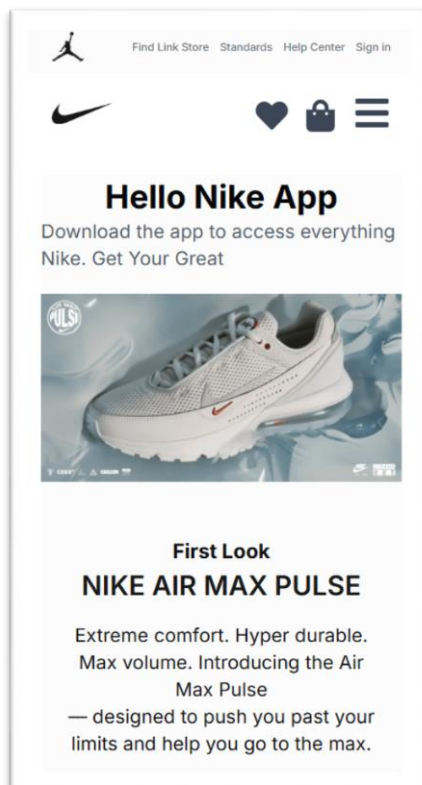
- Identify bottlenecks using tools like Lighthouse, GTmetrix, WebPageTest or google pagespeed.
- Optimize images, minimize JavaScript and CSS, and implement caching strategies.

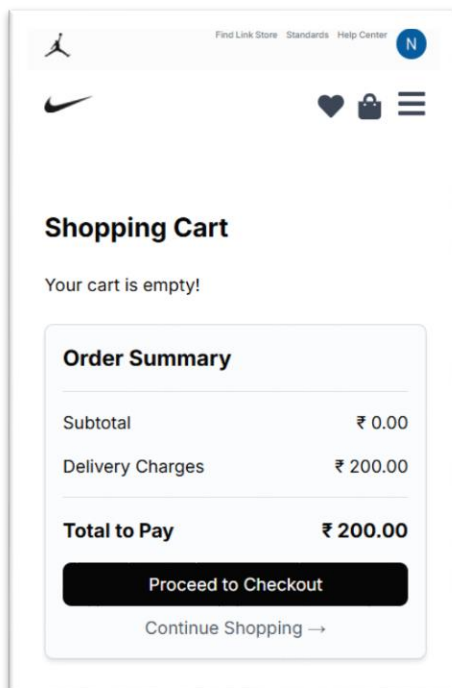
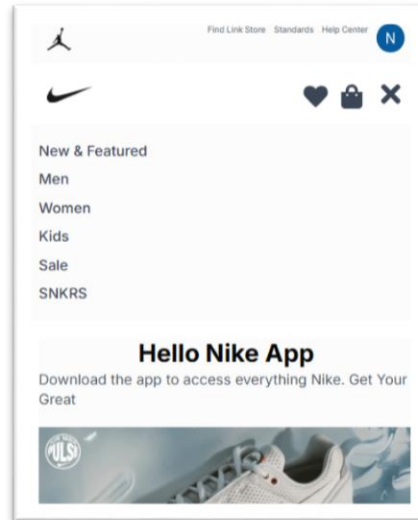
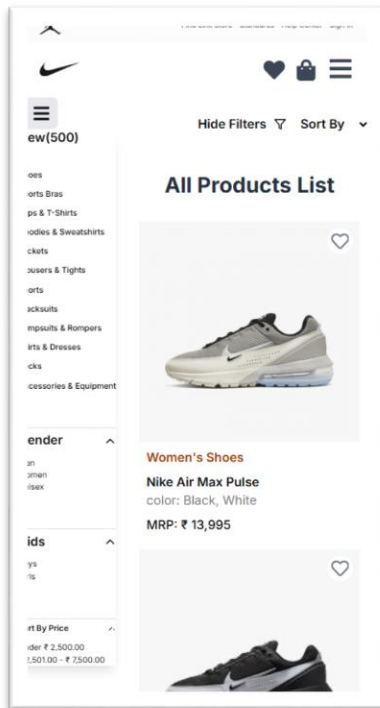


Cross-Browser and Device Testing

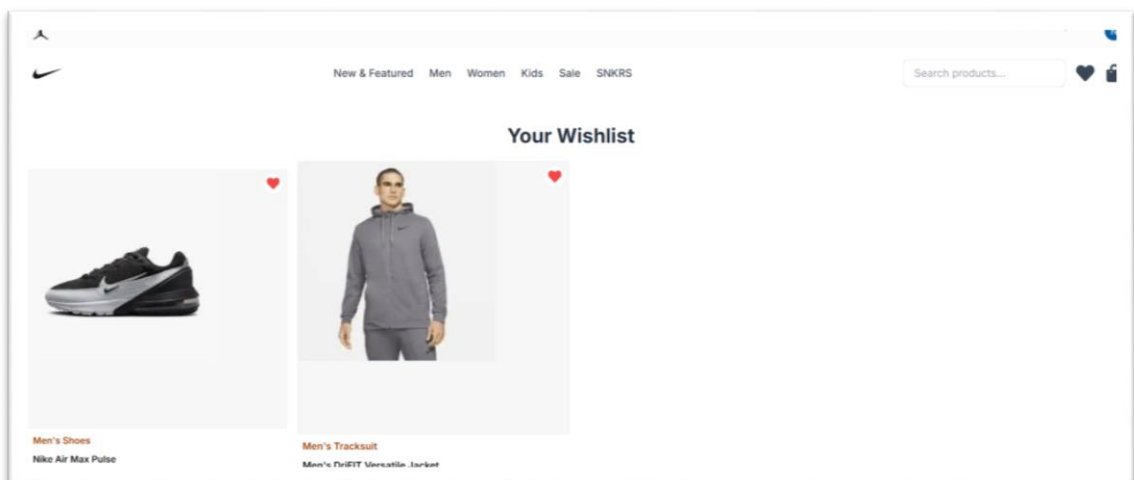
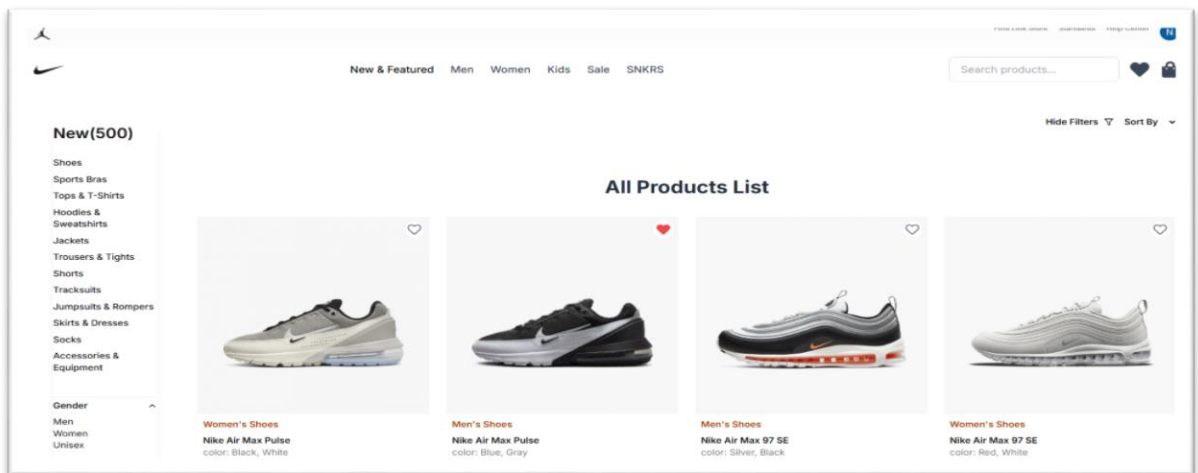
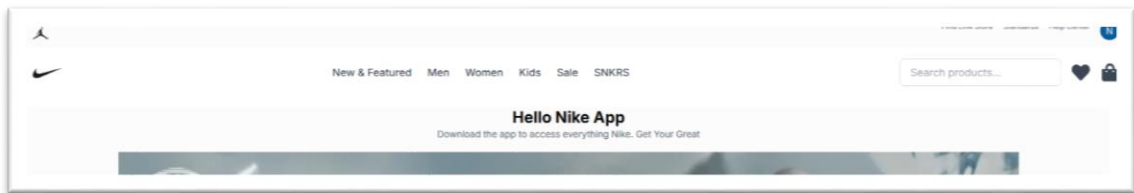
- Test your marketplace on popular browsers (Chrome, Firefox, Safari, Edge) and devices (desktop, tablet, mobile).
- Use responsive design testing tools like BrowserStack or LambdaTest.


Sm screen








lg screen








Find Link Store Standards Help Center


New & Featured Men Women Kids Sale SNKRS


Shopping Cart

	Nike Air Max Pulse <small>Add to Favorites</small>	<div>- 1 +</div>	\$13995 (Original: \$13995) <small>Total: \$13995.00</small>	
	Nike Air Max Pulse <small>Add to Favorites</small>	<div>- 1 +</div>	\$13995 (Original: \$13995) <small>Total: \$13995.00</small>	


Order Summary

Nike Air Max Pulse (x1)	₹ 13995.00
Nike Air Max Pulse (x1)	₹ 13995.00
Subtotal	₹ 27990.00
Delivery Charges	₹ 200.00
Total to Pay	₹ 28190.00


Proceed to Checkout
Continue Shopping →




Women's Sports
Nike Sportswear
color: Black, Red
MRP: ₹ 13,995



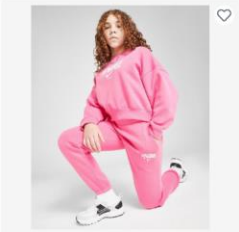
Men's Running
Nike Air Max Pulse
color: Black, Yellow
MRP: ₹ 3,9200



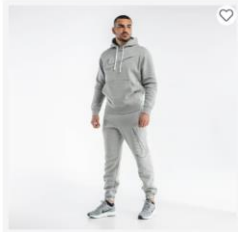
Men's Shoes
Nike Air Force 1 Mid '07
color: White, Black
MRP: ₹ 10,995




Men's Shoes
Nike Court Vision Low Next Nature
color: Green, White
MRP: ₹ 4,995



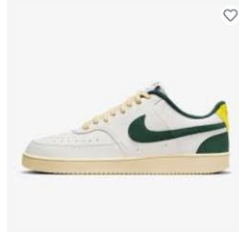
Women's Tracksuit



Men's Tracksuit



Kid's Shoes



Kid's Shoes

Step 5: Security Testing

1. Input Validation:

Enter your name and address:

First Name *

Last Name *

Address Line 1 *

We do not ship to P.O. boxes

Address Line 2

Address Line 3

Postal Code *

Locality *

State/Territory *

Country *

Account Holder Name

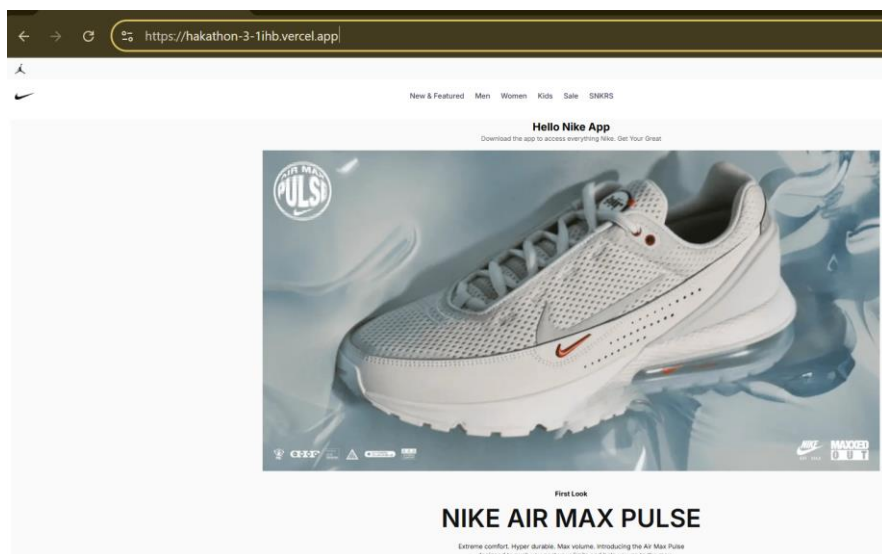
Enter account holder name

Email Address

! Please fill out this field.

2. Secure API Communication:

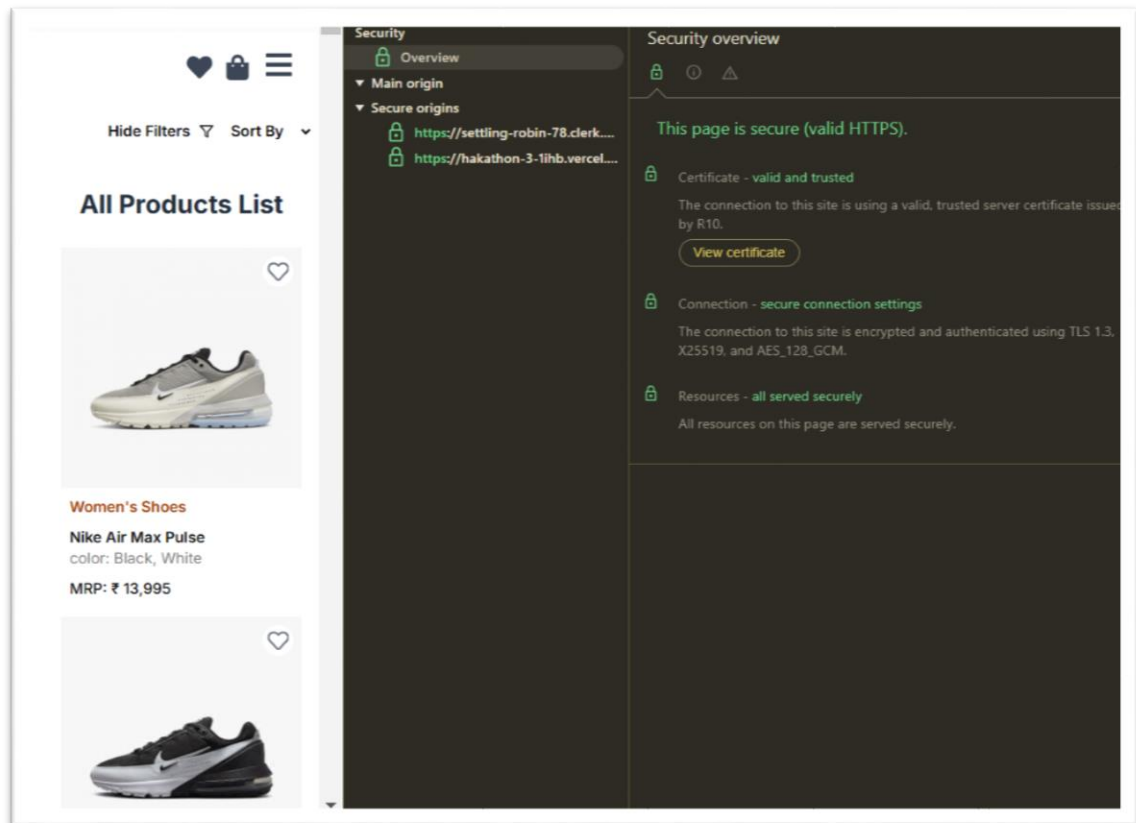
- Ensure API calls are made over HTTPS



3. Testing Tools:

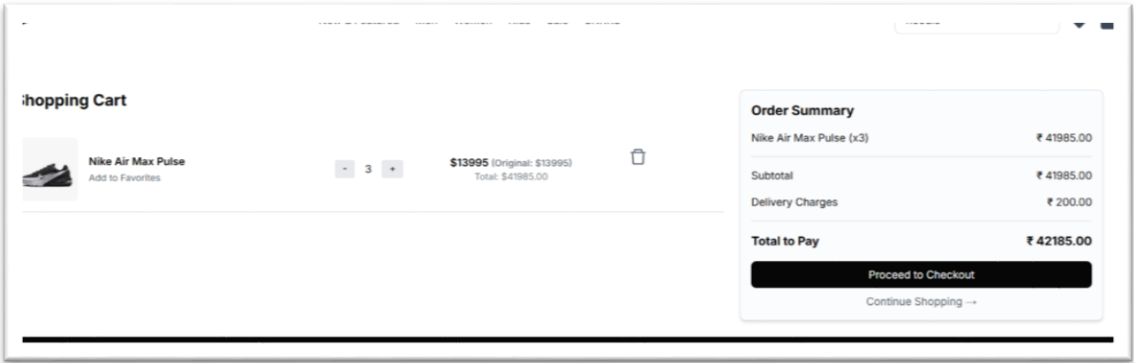
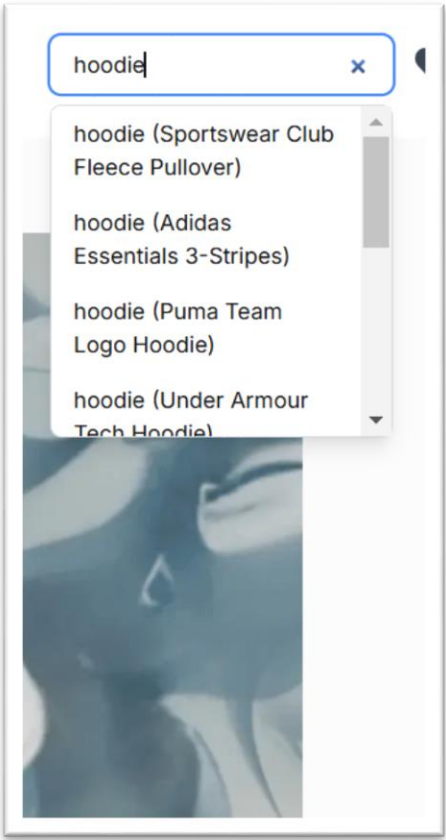
- OWASP ZAP: For automated vulnerability scanning.

By-Google



Step 6: User Acceptance Testing (UAT)

- Simulate real-world scenarios by interacting with your marketplace as a user.
- Verify that workflows like browsing, searching, and checkout are intuitive and error-free.



Shipping Rates Calculator

Ship To Address

Get Shipping Rates

Available Shipping Rates

Carrier: UPS

☐ Service: UPS Express

Cost: 379.39 usd

Carrier: UPS

☐ Service: UPS Express Early

Cost: 469.39 usd

Carrier: UPS

☐ Service: UPS Express Saver

Cost: 342.61 usd

Carrier: UPS

☐ Service: UPS 2nd Day Air®

Cost: 165.3 usd

Carrier: UPS

☐ Service: UPS 2nd Day Air AM®

Cost: 190.09 usd

Carrier: UPS

☐ Service: UPS 3 Day Select®

Cost: 141.79 usd

Carrier: UPS

☐ Service: UPS® Ground

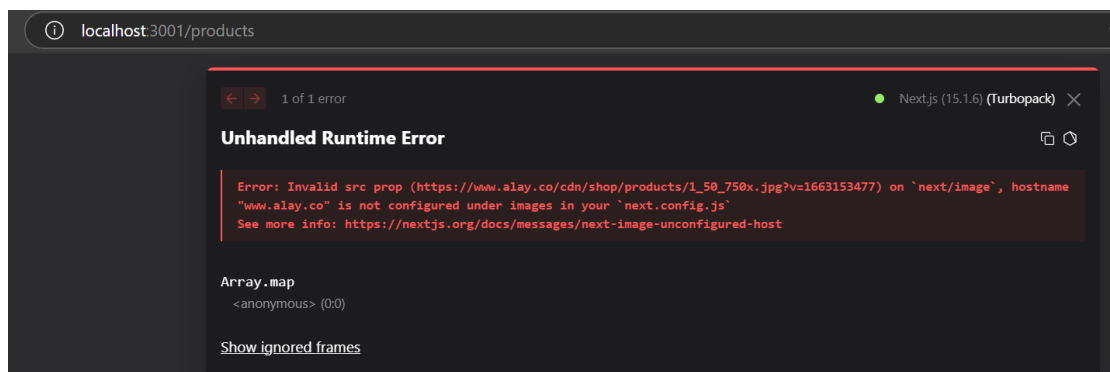
Cost: 49.92 usd

Step 7: Documentation Updates

1. Include Testing Results:
 - o Summarize key issues found and how they were resolved.
 - o Provide before-and-after screenshots for fixes.

Problem:

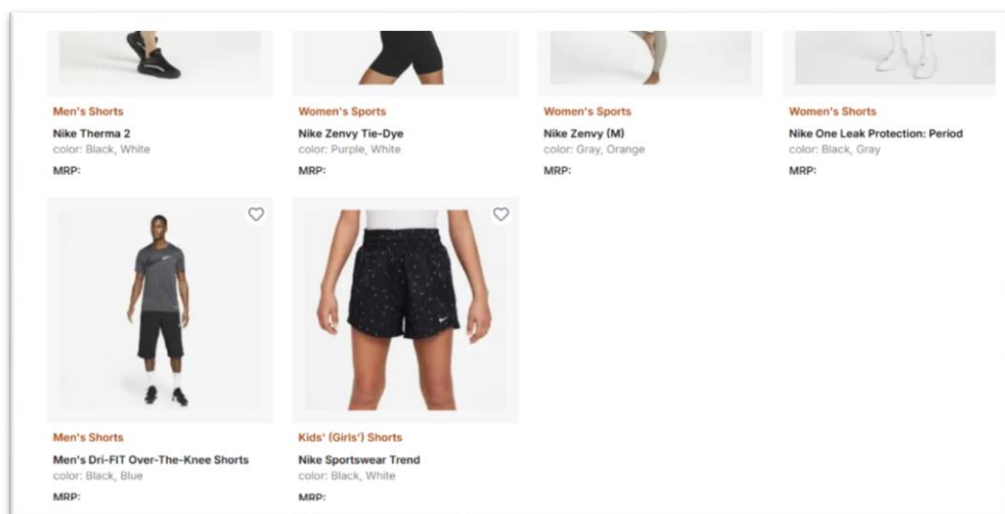
Face this type of error multiple times in multiple file due to images fetching API & API have different links of images.



Solution:

```
const nextConfig = {
  images: {
    domains: [
      "www.nike.com.kw",
      "www.nike.sa",
      "www.thenextsole.com",
      "cdn.shopify.com",
      "atmos-kl.com",
      "media.jdsports.com",
      "images.clerk.dev",
      "images.puma.com",
      "reebok.bynder.com",
      "i.ebayimg.com",
      "www.nike.ae",
      "speedsports.pk",
      "www.jdsports.cy",
      "www.buymo.co.za",
      "stndrdnola.com",
      "konfor.pk",
      "dks.scene7.com",
      "images.bike24.com",
      "www.alay.co",
    ],
  },
}
```

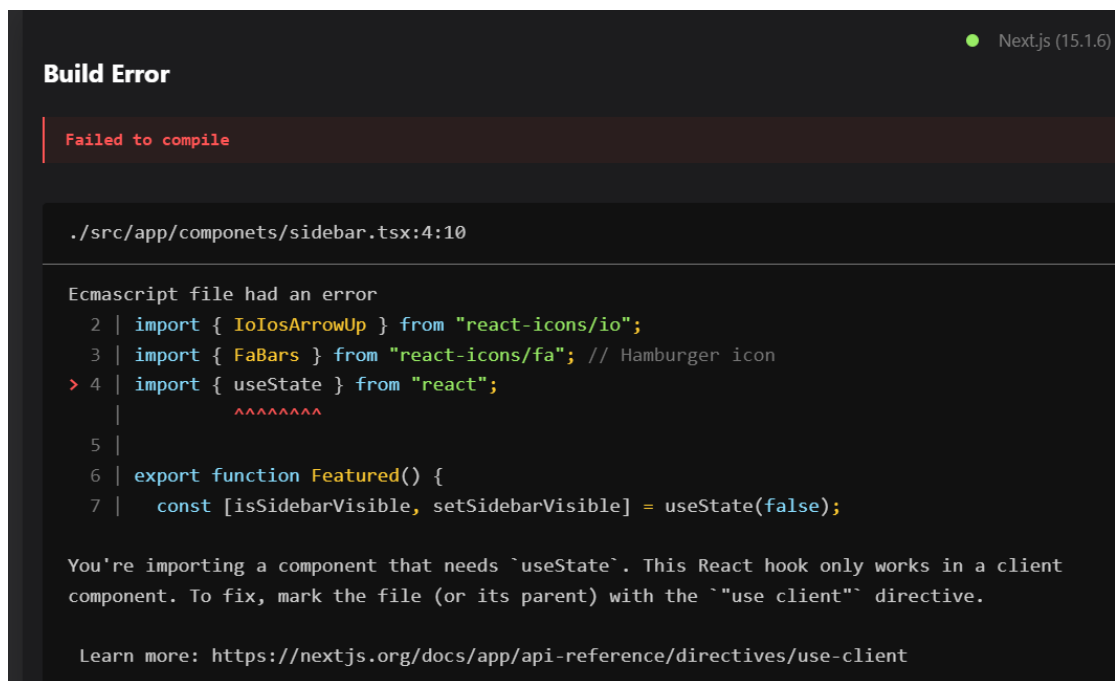
Problem: price don't show due to link error



Solution:

```
<h1 className="font-semibold text-sm mt-2">{item.title}</h1>
<h4 className="text-sm text-[#757575] justify-between">
  color: {item.color.join(", ")}
</h4>
<h2 className="font-medium text-sm mt-2">MRP: {item.price}</h2>
</div>
```

This type of 100 of errors I am faced during coding.

A screenshot of a Next.js build error. The title is 'Build Error' with a green dot and 'Next.js (15.1.6)' in the top right. Below the title is a red bar with the text 'Failed to compile'. The error message is 'EcmaScript file had an error' followed by a code snippet from './src/app/componets/sidebar.tsx:4:10'. The code shows imports for 'IoIosArrowUp', 'FaBars', and 'useState' from 'react'. A red arrow points to the 'useState' import on line 4. Below the code, a message explains that 'useState' is a client-only hook and the file needs the 'use client' directive. A link to the Next.js documentation is provided at the bottom.

```
Build Error Next.js (15.1.6)
```

Failed to compile

./src/app/componets/sidebar.tsx:4:10

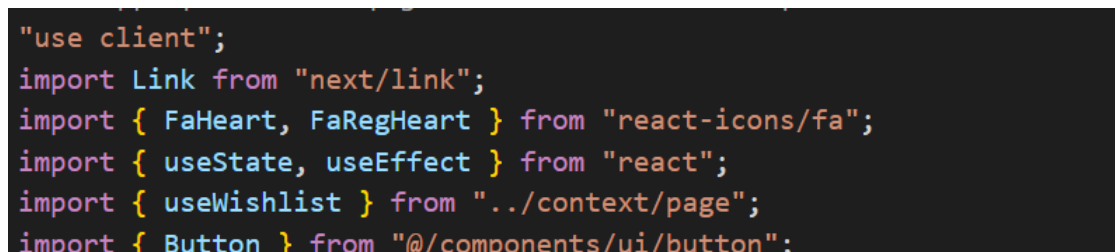
EcmaScript file had an error

```
2 | import { IoIosArrowUp } from "react-icons/io";
3 | import { FaBars } from "react-icons/fa"; // Hamburger icon
> 4 | import { useState } from "react";
    |         ^^^^^^^^^
5 |
6 | export function Featured() {
7 |   const [isSidebarVisible, setSidebarVisible] = useState(false);
```

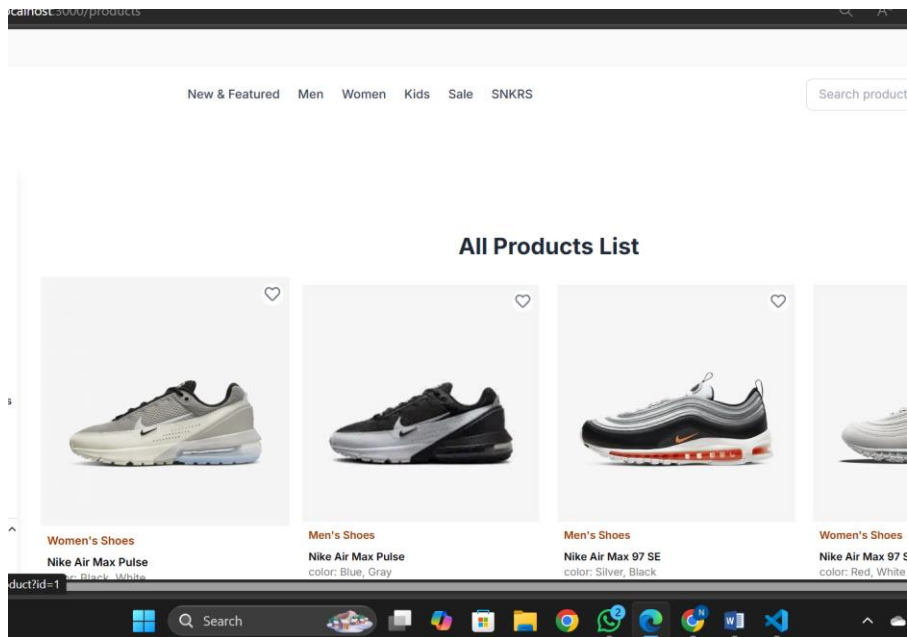
You're importing a component that needs `useState`. This React hook only works in a client component. To fix, mark the file (or its parent) with the `"use client"` directive.

Learn more: <https://nextjs.org/docs/app/api-reference/directives/use-client>

Solve by using 'use client'

A screenshot of a code editor showing the corrected code. The first line is '"use client";'. This is followed by several import statements: 'Link' from 'next/link', 'FaHeart' and 'FaRegHeart' from 'react-icons/fa', 'useState' and 'useEffect' from 'react', 'useWishlist' from '../context/page', and 'Button' from '@components/ui/button'.

```
"use client";
import Link from "next/link";
import { FaHeart, FaRegHeart } from "react-icons/fa";
import { useState, useEffect } from "react";
import { useWishlist } from "../context/page";
import { Button } from "@components/ui/button";
```



Found this error due to wrong tag

```
const response = await fetch("https://677ff2a50476123f76a8dd73.mockapi.io/product");
if (!response.ok) {
  throw new Error("Failed to fetch product details");
}
const data: Product[] = await response.json();
const selectedProduct = data.find((items: Product) => items.id === parseInt(id!, 10));
if (selectedProduct) {
  setProduct(selectedProduct);
  const related = data.filter(
    (item: Product) =>
      item.producttype === selectedProduct.producttype && item.id !== selectedProduct.id
  );
}
```

that's why show this

New & Featured Men Women Kids Sale SNKRS

Product not found.

Solution:

```
const fetchProduct = async () => {
  const response = await fetch("https://677ff2a50476123f76a8dd73.mockapi.io/product");
  if (!response.ok) {
    throw new Error("Failed to fetch product details");
  }
  const data: Product[] = await response.json();
  const selectedProduct = data.find((item: Product) => item.id !== parseInt(id!, 10));
  if (selectedProduct) {
    setProduct(selectedProduct);
  }
}
```

Day 5 Summary: Testing, Error Handling & Backend Refinement

On Day 5, I focused on testing, error handling, and optimizing the backend to prepare the marketplace for deployment.

Key Steps Taken:

- ✓ **Functional Testing** – Checked product listings, filters, cart operations, and dynamic routing to ensure smooth functionality.
- ✓ **Error Handling** – Used try-catch to handle API errors and displayed user-friendly messages.
- ✓ **Performance Optimization** – Used tools like Lighthouse and GTmetrix to improve speed and efficiency.
- ✓ **Cross-Browser & Device Testing** – Ensured the marketplace works on different browsers and screen sizes.
- ✓ **Security Testing** – Verified input validation and secure API communication using HTTPS and OWASP ZAP.
- ✓ **User Testing (UAT)** – Simulated real-world user interactions for a seamless experience.
- ✓ **Documentation Updates** – Logged errors and fixes with screenshots for future reference.

Final Status: ✓ All tests completed, and the marketplace is ready for deployment.

Checklist for Day 5:

Functional Testing:

- ✓

Error Handling:

- ✓

Performance Optimization:

- ✓

Cross-Browser and Device Testing:

- ✓

Security Testing:

- ✓

Documentation:

- ✓

Final Review:

- ✓