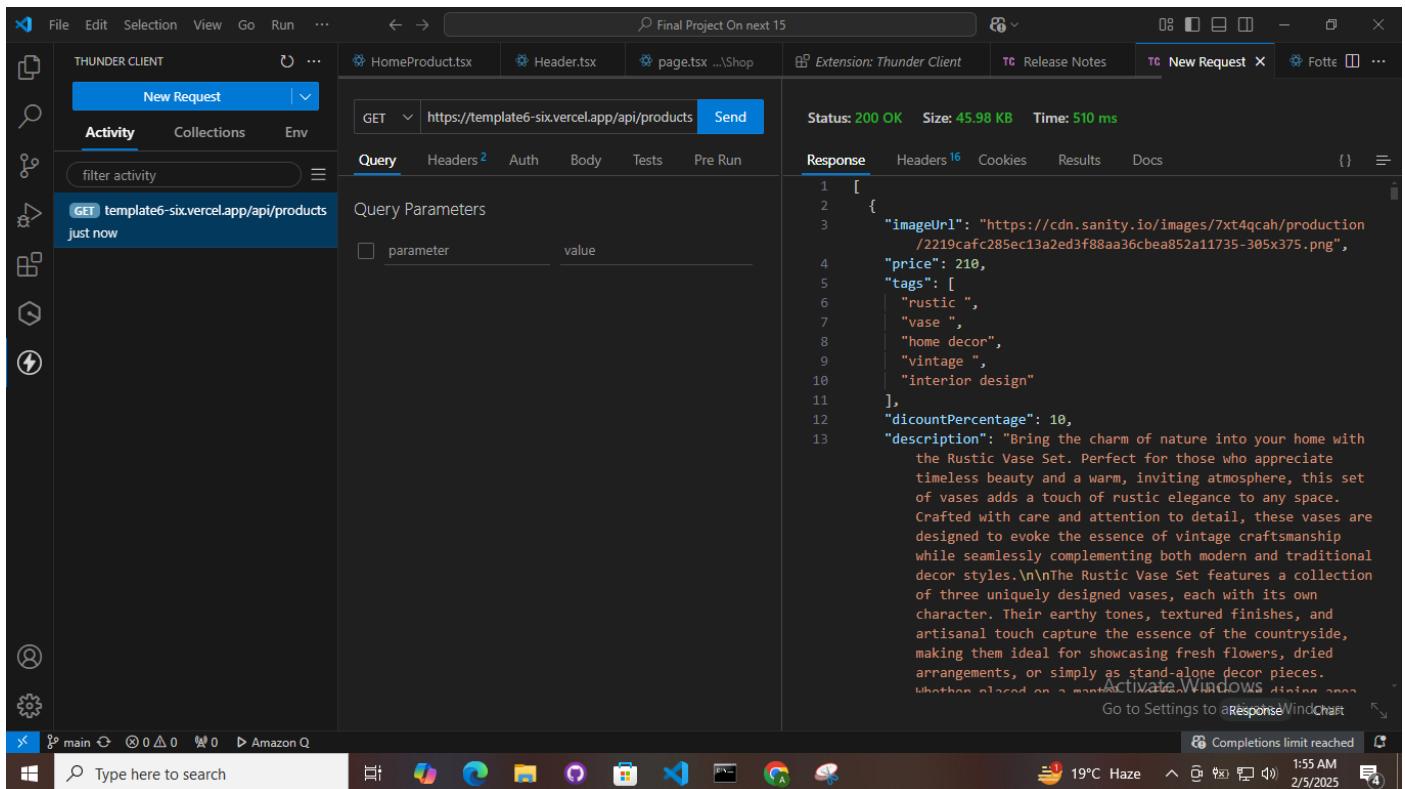


“Hackathon 3 (Day 5)”

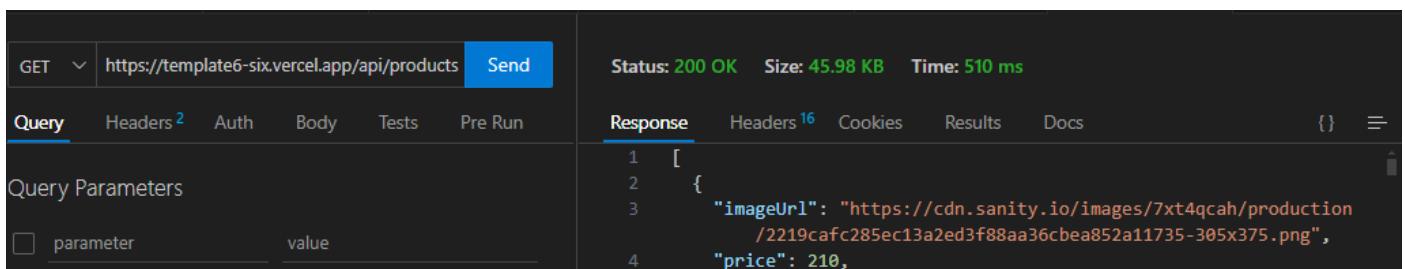
TESTING, ERROR HANDLING, AND BACKEND INTEGRATION REFINEMENT

Product API Testing By Thunder Client



The screenshot shows the Thunder Client interface. On the left, there's a sidebar with icons for file operations, search, collections, environment variables, and a timer. The main area has tabs for HomeProduct.tsx, Header.tsx, page.tsx, Extension: Thunder Client, Release Notes, New Request, and Footer. A search bar at the top says "Final Project On next 15". In the center, a "New Request" button is highlighted. Below it, a "Activity" tab is selected, showing a list of recent requests: "GET template6-six.vercel.app/api/products just now". To the right, a "Query" tab is selected for the current request. The URL "https://template6-six.vercel.app/api/products" is entered. A "Send" button is to the right of the URL. Below the URL, there are tabs for Headers (16), Auth, Body, Tests, and Pre Run. Under Headers, there's a section for "Query Parameters" with a table for "parameter" and "value". The "Response" tab is selected, showing the API response details: Status: 200 OK, Size: 45.98 KB, Time: 510 ms. The response body is a JSON object:

```
1 [  
2 {  
3   "imageUrl": "https://cdn.sanity.io/images/7xt4qcah/production  
/2219caf285ec13a2ed3f88aa36cbea852a11735-305x375.png",  
4   "price": 210,  
5   "tags": [  
6     "rustic ",  
7     "vase ",  
8     "home decor",  
9     "vintage ",  
10    "interior design"  
11  ],  
12  "discountPercentage": 10,  
13  "description": "Bring the charm of nature into your home with  
the Rustic Vase Set. Perfect for those who appreciate  
timeless beauty and a warm, inviting atmosphere, this set  
of vases adds a touch of rustic elegance to any space.  
Crafted with care and attention to detail, these vases are  
designed to evoke the essence of vintage craftsmanship  
while seamlessly complementing both modern and traditional  
decor styles.\n\nThe Rustic Vase Set features a collection  
of three uniquely designed vases, each with its own  
character. Their earthy tones, textured finishes, and  
artisanal touch capture the essence of the countryside,  
making them ideal for showcasing fresh flowers, dried  
arrangements, or simply as stand-alone decor pieces.  
Whether placed on a mantel, side table, or dining room  
table, this set is sure to add a touch of rustic charm to  
any room."}
```



This screenshot shows the same Thunder Client interface as the previous one, but the response body is truncated. It still displays the status, size, and time details.

Test Case ID	Test Case Description	Test Steps	Expected Result
TC001	Validate product listing page	Open product page > Verify products	Products displayed correctly
TC002	Test API error handling	Disconnect API > Refresh page	Show fallback UI with error
TC003	Check cart functionality	Add product to cart > Verify cart contents	Cart updates with added products
TC004	Ensure responsiveness on mobile	Resize browser window > Check layout	Layout adjusts properly to screen size

1. Test Core Features:

Product Listing

Product List



Amber Haven

Step into a world of warmth and tranquility with Amber Haven—a...

\$150 \$180.00

Sleek Living

Welcome to SleekLiving, where modern sophistication meets functio...

\$300 \$360.00

Serene Seat

Step into comfort and tranquility with SereneSeat—a chair designed to...

\$350 \$420.00

Nordic Elegance

Elevate your space with Nordic Elegance—a collection that brings th...

\$280 \$336.00



Rustic Vase Set

Bring the charm of nature into your home with the Rustic Vase Set. Perfect for those who appreciate...

\$210 \$252.00

Bold Nest

Welcome to BoldNest—where fearless design meets comfort and creativity. Crafted for those who embrac...

\$260 \$312.00

Cloud Haven Chair

Sink into comfort with the Cloud Haven Chair—where softness meets support in a beautifully designed piec...

\$230 \$276.00

Bed

Introducing the Bed—your sanctuary for rest and relaxation, designed with both comfort and style in...

\$250 \$300.00

rustic vase home decor vintage interior design

bold nest furniture modern contemporary

cloud chair comfy home decor modern furniture

bed furniture sleep cozy modern

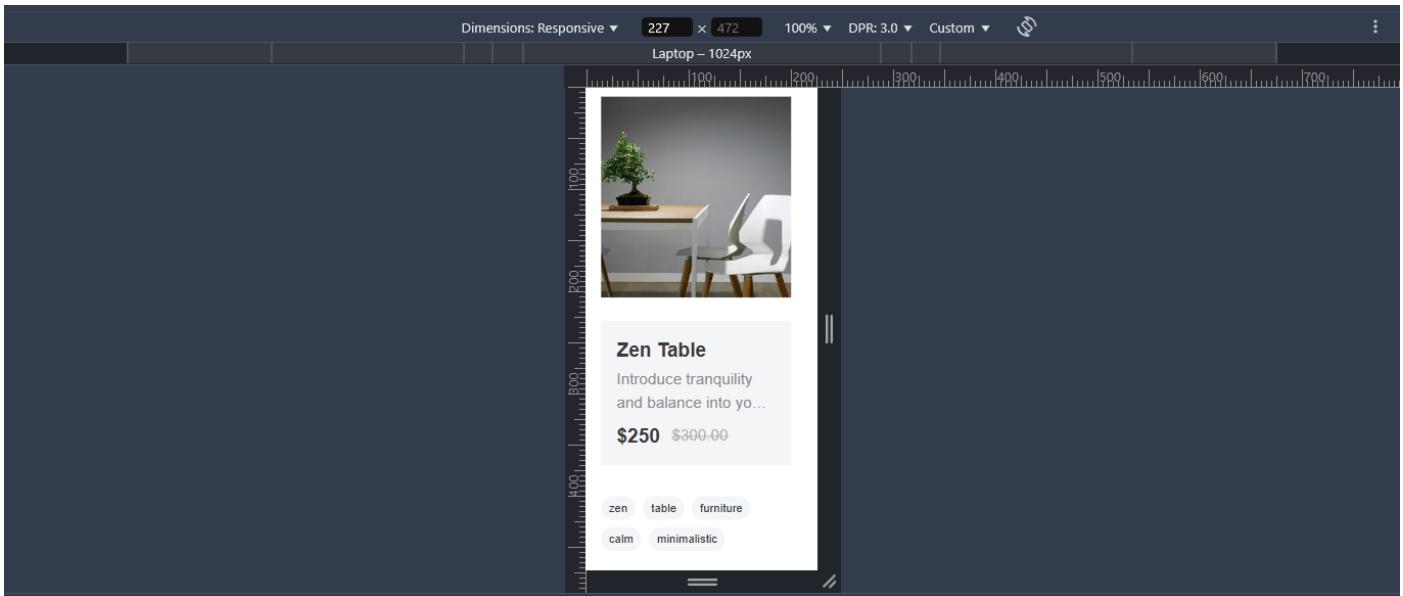
Add To Cart

Add To Cart

Add To Cart

Add To Cart

With Responsiveness



Timber Craft

Introducing TimberCraft—a collection that celebrates the timeless beauty o...

\$320 \$384.00

wooden craftsmanship furniture modern nature inspired

Add To Cart



Bright Space

Welcome to BrightSpace—a collection designed to infuse your home with...

\$180 \$216.00

bright space minimalistic modern decor

Add To Cart



The Dandy chair

Meet The Dandy Chair—the epitome of comfort, style, and sophistication....

\$150 \$180.00

chair elegant vintage classic furniture

living fancy elegance design

Add To Cart



Syltherine

Introducing Syltherine – the ultimate fusion of elegance and power. Crafted...

\$200 \$240.00

living fancy elegance design

Add To Cart



The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The left sidebar contains various icons for file operations like Open, Save, Find, and Undo. The top navigation bar includes File, Edit, Selection, View, Go, and a search bar with the placeholder "Final Project On next 15". The title bar shows the current file path: "HomeProduct.tsx" and "Header.tsx" in the tabs, with "page.tsx" as the active tab. Below the tabs, the status bar displays "Ln 20 Col 1 Spaces: 2 UTF-8 CRLF TypeScript JSX Completions limit reached". The main code editor area contains the following TypeScript code:

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
```

```
use client";
import React, { useEffect, useState } from "react";
import Image from "next/image";
import createClient from "@sanity/client";
import Link from "next/link";
import Header from "@/components/Header";
import Swal from "sweetalert2";
import Feature from "@/components/Feature";
import Footer from "@/components/Footer";
import ProductTypeGlobale from "../ProductType/producttypes";
const sanity = createClient({
  projectId: "g0g0Praq",
  dataset: "production",
  useCdn: true,
  apiVersion: "2021-03-25",
});
const Shop = () => {
  const [products, setProducts] = useState<ProductTypeGlobale[]>([]);
  const [cart, setCart] = useState<ProductTypeGlobale[]>([]);
  const [filteredProducts, setFilteredProducts] = useState<ProductTypeGlobale[]>([]);
  const [searchQuery, setSearchQuery] = useState("");
  const fetchProducts = async () => {
    try {
      const query = `*[_type == "product"]{
        _id,
        title,
        price,
        description,
        "imageurl": productImage.asset->url,
        productImage,
        tags
      }`;
      const data = await sanity.fetch<ProductTypeGlobale[]>(query);
      setProducts(data);
      setFilteredProducts(data); // Set the filtered list initially to all products
    } catch (error) {
      console.error(error);
    }
  };
  useEffect(() => {
    fetchProducts();
  }, []);
  return (
    <div>
      <Header />
      <Feature />
      <div>
        <h2>Shop</h2>
        <div>
          <h3>All Products</h3>
          <ul>
            {products.map((product) => (
              <li>
                <Link href={`/shop/${product._id}`}>{product.title}</Link>
                <span>${product.price}</span>
              </li>
            ))}
          </ul>
        </div>
        <div>
          <h3>Search Results</h3>
          <input type="text" value={searchQuery} onChange={(e) => setSearchQuery(e.target.value)} />
          <ul>
            {filteredProducts.map((product) => (
              <li>
                <Link href={`/shop/${product._id}`}>{product.title}</Link>
                <span>${product.price}</span>
              </li>
            ))}
          </ul>
        </div>
      </div>
      <Footer />
    </div>
  );
}
export default Shop;
```

The screenshot shows a Microsoft Edge browser window with a dark theme. The address bar says "Final Project On next 15". The left sidebar shows a file tree for a "Final-Project-E-commerce-Website" project. The "src" folder contains "app", "public", and "script" subfolders. "app" contains "Blog", "Cart", "CheckOut", "Contact", "ProductType", "Shop", and "page.tsx". "Shop" contains "index.tsx", "product.tsx", "envs.ts", "structure.ts", ".env.local", ".gitignore", and ".eslintrc.mjs". "page.tsx" is currently selected. The main content area shows a React component for a shop page. The component uses hooks like useState and useEffect to fetch products from an API. It includes a header with a logo and navigation links for Home and Shop. Below the header is a search bar with a placeholder "Search products...". The code is written in TypeScript.

```
const Shop = () => {
  const [cart, setCart] = useState(0);
  const [products, setProducts] = useState([]);
  const [searchQuery, setSearchQuery] = useState("");
  const [handleSearch, setHandleSearch] = useState(() => {});

  useEffect(() => {
    fetchProducts();
  }, [cart]);

  const fetchProducts = async () => {
    const response = await fetch("http://localhost:3001/api/products");
    const data = await response.json();
    setProducts(data);
  };

  return (
    <>
      <Header />
      <div className="w-full max-w-[1440px] mx-auto bg-[url('/BackgroundImage.jpg')] py-12 md:py-16 bg-cover bg-center bg-no-repeat opacity-90">
        <div className="text-center space-y-5">
          <h1 className="text-5xl font-bold leading-8 text-black">Shop</h1>
          <div className="text-lg">
            <a href="/">
              <span>{">"}</span>
              <span className="text-black">Shop</span>
            </a>
          </div>
        </div>
        <div style={{ position: "relative" }}>
          <input
            type="text"
            placeholder="Search products..."
            value={searchQuery}
            onChange={handleSearch}
            className="border border-gray-300 rounded-lg px-4 py-2 w-full max-w-[300px] focus:outline-none" />
        </div>
      </div>
    </>
  );
}

export default Shop;
```

The screenshot shows a Microsoft Edge browser window with the following details:

- Title Bar:** Final Project On next 15
- Address Bar:** http://localhost:3001
- Page Content:** A product list card component. Each card contains:
 - A placeholder image for the product.
 - The product's title.
 - A short description.
 - The original price.
 - The discounted price (original * 1.2).
 - A line-through effect for the original price.
 - A list of tags associated with the product.
- Side Panel (Explorer):** Shows the project structure, including `node_modules`, `public`, `script`, `src` (with `app`, `Blog`, `Cart`, `CheckOut`, `Contact`, `ProductType`, `Shop`, `[id]`, and `page.tsx`), and configuration files like `favicon.ico`, `globals.css`, `layout.tsx`, `page.tsx`, `components`, `sanity`, `lib`, `schematypes`, `ts_index.ts`, `ts_product.ts`, `ts_env.ts`, `.env.local`, `.gitignore`, and `eslint.config.mjs`.
- Code Editor:** An open file is `Shop/page.tsx`, containing the following code:

```
const Shop = () => {
  return (
    <div>
      <h1>Product List</h1>
      <div>
        {filteredProducts.map((product) => (
          <Link key={product._id} href={`/shop/${product._id}`}>
            <div>
              <Image src={product.imageurl || "/placeholder.jpg"} alt={product.title} width={285} height={301} />
              <div>
                <h3>{product.title}</h3>
                <p>{product.description}</p>
                <div>
                  <span>${product.price}</span>
                  <span>${(product.price * 1.2).toFixed(2)}</span>
                </div>
                <div>
                  {product.tags.map((tag, index) => (
                    <span key={`${product._id}-${tag}-${index}`}
                      <span>${tag}</span>
                    </span>
                  ))}
                </div>
              </div>
            </div>
          </Link>
        ))
      </div>
    </div>
  )
}
```

File Edit Selection View Go ... ← → ⌘ Final Project On next 15

EXPLORER ...

FINAL P... ↗ ↘ ⌂ Final-Project-Ecomm... ↗ ↘

.next
node_modules
public
script
src
app
addtocart
Blog
Cart
CheckOut
Contact
ProductType
Shop
[id]
page.tsx

HomeProduct.tsx Header.tsx page.tsx ...Shop TS product.ts Footer.tsx page.tsx ...app

```

21 const Shop = () => {
115   </div>
142   </div>
143   <div className="mt-8 flex flex-wrap">
144     {filteredProducts.map((product) => (
145       <span
146         key={ `${product._id}-${tag}-${index}` }
147         className="text-xs bg-[#F4F5F7] text-[#3A3A3A] px-2 py-1 rounded-full mr-2 mb-2"
148       >
149         {tag}
150       </span>
151     )));
152   </div>
153   /* Add To Cart */
154   <div>
155     <button
156       onClick={(e) => handleAddToCart(e, product)}
157         className="bg-[#B88E2F] text-white px-4 py-2 rounded mt-4"
158       >
159         Add To Cart
160       </button>
161     </div>
162   </div>
163   <Link>
164     <div>
165       <div>
166         <Feature />
167         <Footer />
168       </div>
169     </div>
170   </div>
171   );
172 >;
173 >
174 export default Shop;
175

```

Ln 20, Col 1 Spaces: 2 UTF-8 CRLF () TypeScript JSX Completions limit reached ✓ Prettier

Render Product Dynamically

 **Furniro**

Home Shop Blog Contact



Product Details

Home > Shop >

Home > Shop > | Timber Craft



Timber Craft

★★★★★ (5 Customer Review)

Rs. 320



Timber Craft

★★★★★ (5 Customer Review)

Rs. 320

Introducing TimberCraft—a collection that celebrates the timeless beauty of wood craftsmanship and the art of nature's finest materials. Inspired by the rustic charm and durability of natural wood, TimberCraft brings warmth, character, and a touch of handcrafted elegance to any space. Perfect for those who value authenticity and sustainability, this collection combines the strength of timber with refined design, making it the ideal choice for modern and traditional homes alike. Each piece in the TimberCraft collection is meticulously crafted to highlight the unique grains and textures of the wood, bringing an organic, earthy feel to your space. Whether you're looking for furniture, decor, or accents, TimberCraft offers a variety of

products that bring the beauty of nature into your home.
TimberCraft—where the beauty of wood meets masterful craftsmanship, creating lasting pieces that stand out in both style and substance.

Size



Color



- 1 +

Add To Cart

+ Compare

Description

Additional Information

Reviews [5]



Timber Craft

★★★★★ (5 Customer Review)

Rs. 320

Introducing TimberCraft—a collection that celebrates the timeless beauty of wood craftsmanship and the art of nature's finest materials. Inspired by the rustic charm and durability of natural wood, TimberCraft brings warmth, character, and a touch of handcrafted elegance to any space. Perfect for those who value authenticity and sustainability, this collection combines the strength of timber with refined design, making it the ideal choice for modern and traditional homes alike. Each piece in the TimberCraft collection is meticulously crafted to highlight the unique grains and textures of the wood, bringing an organic, earthy feel to your space. Whether you're looking for furniture, decor, or accents, TimberCraft offers a variety of beautifully designed pieces that blend rustic appeal with contemporary sophistication. The collection is designed to stand the test of time, with materials chosen for their durability and lasting beauty. TimberCraft is perfect for those who appreciate quality, craftsmanship, and the enduring beauty of wood. From enlivening tables and chairs to decorative accessories, this collection adds a natural elegance to any room, transforming your living space into a warm and welcoming retreat. Key Features: Made from high-quality, sustainable timber for durability and lasting appeal. Rustic yet refined design that complements both modern and traditional interiors. Unique wood grains and texture that bring natural beauty to your space. Expert craftsmanship that highlights the authenticity of each piece. Perfect for creating a cozy, earthy atmosphere in any room. Bring the timeless charm of nature into your home with TimberCraft—where the beauty of wood meets masterful craftsmanship, creating lasting pieces that stand out in both style and substance.

Size:

L XL XS

Color:

● ● ●

+

Add To Cart + Compare

The screenshot shows a code editor interface with the title "Final Project On next 15". The left sidebar displays a project structure under "FINAL P...". The "src" folder contains "Shop" and "app". Inside "Shop" is a file named "page.tsx". The code editor shows the content of "page.tsx".

```
1 import Image from "next/image";
2 import Link from "next/link";
3 import Header from "@components/Header";
4 import createClient from "@sanity/client"; // Import the sanity client
5 import ProductInfo from "@components/ProductInfo";
6 import MidHeader from "@components/midheader";
7 import Footer from "@components/Footer";
8 import Feature from "@components/Feature";
9 import ProductTypeGlobl from "@app/ProductType/producttypes";
10
11 const client = createClient({
12   projectId: "e66g9rag",
13   dataset: "production",
14   useCdn: true,
15   apiVersion: "2021-03-25", // use a UTC date string
16   token: process.env.SANITY_API_TOKEN,
17 });
18
19 interface PageProps {
20   params: Promise<{ id: string }>; // Ensure params is async
21 }
22
23 export default async function ProductDetail({ params }: PageProps) {
24   const { id } = await params; // Ensure params is awaited properly
25
26   const product = await client.fetch<ProductTypeGlobl>(
27     `[_type == "product" && _id == ${id}]`[0]
28     _id,
29     title,
30     price,
31     description,
32     "imageurl": productImage.asset->url,
33     productImage,
34     tags
35   );
36
37   if (!product) {
38     return (
39       <main className="min-h-screen">
40         <div>Product Not Found</div>
41       </main>
42     );
43   }
44
45   return (
46     <><main>
47       <Header />
48       <MidHeader />
49       <div className="max-w-[1440px] mx-auto px-4 md:px-12 bg-[#F9F1E7] py-8">
50         <div className="space-x-6">
51           <Link href="/" className="text-[#9F9F9F] hover:text-black transition-colors font-semibold">
52             Home
53           </Link>
54           <span> {">"} </span>
55           <Link href="/Shop" className="text-[#9F9F9F] hover:text-black transition-colors font-semibold">
56             Shop
57           </Link>
58         </div>
59       </div>
60     </main></>
61   );
62 }
```

The screenshot shows the same code editor interface with the title "Final Project On next 15". The project structure is identical to the first screenshot. The "src" folder contains "Shop" and "app". Inside "Shop" is a file named "page.tsx". The code editor shows the content of "page.tsx". The code has been modified to include a main component structure and specific styling for the header and midheader.

```
1 export default async function ProductDetail({ params }: PageProps) {
2   const product = await client.fetch<ProductTypeGlobl>(
3     `[_type == "product" && _id == ${id}]`[0]
4     _id,
5     title,
6     price,
7     description,
8     "imageurl": productImage.asset->url,
9     productImage,
10    tags
11  );
12
13  if (!product) {
14    return (
15      <main className="min-h-screen">
16        <div>Product Not Found</div>
17      </main>
18    );
19  }
20
21  return (
22    <><main>
23      <Header />
24      <MidHeader />
25      <div className="max-w-[1440px] mx-auto px-4 md:px-12 bg-[#F9F1E7] py-8">
26        <div className="space-x-6">
27          <Link href="/" className="text-[#9F9F9F] hover:text-black transition-colors font-semibold">
28            Home
29          </Link>
30          <span> {">"} </span>
31          <Link href="/Shop" className="text-[#9F9F9F] hover:text-black transition-colors font-semibold">
32            Shop
33          </Link>
34        </div>
35      </div>
36    </main></>
37  );
38 }
```

The screenshot shows a code editor interface with the following details:

- File Explorer:** On the left, it lists the project structure. Key files include `HomeProduct.tsx`, `Header.tsx`, `page.tsx`, `Shop`, `page.tsx` (selected), `Footer.tsx`, and `page.tsx` (app).
- Code Editor:** The main area displays the `page.tsx` file content. The code is a functional component named `ProductDetail` that takes `params` and `PageProps`. It contains two `Image` components with URLs `product.imageUrl1` and `product.imageUrl2`. The component uses CSS-in-JS classes like `grid grid-cols-1 sm:grid-cols-2 gap-4 sm:gap-6 mt-6 sm:mt-8`.
- Search Bar:** At the top center, it says "Final Project On next 15".
- System Tray:** At the bottom right, it shows the date and time as "1:09 AM 2/5/2025".

Search Bar



Search products...

Product List

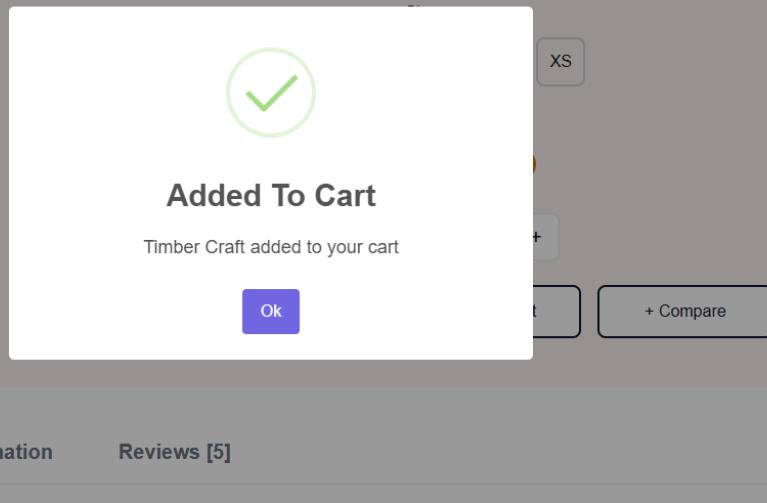


Search products...



Product AddtoCart

timeless charm of nature into your home with TimberCraft—where the beauty of wood meets masterful craftsmanship, creating lasting pieces that stand out in both style and substance.



Home Shop Blog Contact



Your Shopping Cart

Bold Nest

Welcome to BoldNest—where fearless design meets comfort and creativity. Crafted for those who embrace individuality and bold expressions, BoldNest is more than just a piece of furniture; it's a statement. With its striking design, exceptional comfort, and modern aesthetics, BoldNest is perfect for anyone looking to add a unique touch to their home or office. The BoldNest collection combines daring colors, dynamic shapes, and high-quality materials to create an environment where style and comfort coexist in perfect harmony. Whether it's a chair, sofa, or accent piece, each item is designed to stand out while offering a welcoming space for relaxation and enjoyment. Its sturdy construction and thoughtful design ensure that BoldNest not only makes a bold visual impact but also provides lasting durability. Perfect for those who are looking to break away from the ordinary and make their home a reflection of their bold personality, BoldNest is an ideal choice for creating an energetic and unique atmosphere. Key Features: Bold and unique design that adds personality to any space. High-quality materials for comfort, durability, and style. Available in a variety of striking colors and patterns. Perfect for modern homes and offices that embrace creativity. Versatile design that complements contemporary and eclectic decor. Transform your home into a space that reflects your bold, unique style with BoldNest—where standout design and comfort meet.



\$260.00

- 2 +

Timeless Elegance

Introducing TimelessElegance—a collection that embodies the perfect fusion of classic beauty and modern sophistication. Designed for those who appreciate the enduring appeal of refined style, TimelessElegance brings grace, charm, and unparalleled quality to any space. Each piece in this collection is crafted to stand the test of time, offering not only lasting durability but also a sense of sophistication that never goes out of style. With its clean lines, luxurious materials, and subtle detailing, TimelessElegance seamlessly complements any décor, from contemporary urban apartments to traditional homes. Whether you're adding a statement piece to your living room, bedroom, or office, this collection brings an air of refinement that enhances every environment. The timeless design ensures that your space remains stylish for years to come, while its elegance offers a sense of warmth and comfort. Crafted with exceptional attention to detail, TimelessElegance pieces are built to withstand the rigors of daily life while retaining their beauty. Perfect for those who seek a blend of classic style and modern functionality, this collection is ideal for creating spaces that exude sophistication, grace, and



- 1 +

File Edit Selection View Go Run ... ← → 🔍 Final Project On next 15

EXPLORER FINAL P... HomeProduct.tsx Header.tsx page.tsx ...\\Shop page.tsx ...\\Cart Footer.tsx page.tsx ...\\app

```
1 "use client";
2 import React, { useEffect, useState } from "react";
3 import Image from "next/image";
4 import { Trash2, Plus, Minus } from "lucide-react";
5 import Swal from "sweetalert2";
6 import {
7   getCartItems,
8   removeFromCart,
9   updateCartQuantity,
10 } from "../addtocart/action";
11 import Link from "next/link";
12 import { useRouter } from "next/navigation";
13 import Features from "@components/feature";
14 import Footer from "@components/Footer";
15 import Header from "@components/Header";
16 import ProductTypeGloble from "../ProductType/producttypes";
17
18 const Cart:React.FC = () => {
19   const [cartItems, setCartItems] = useState<ProductTypeGloble[]>([]);
20   const [isLoading, setIsLoading] = useState(true);
21   const router = useRouter();
22
23   useEffect(() => {
24     const cart = getCartItems();
25     setCartItems(cart);
26     setIsLoading(false); // Mark loading as complete
27   }, []);
28
29   const handleRemove = (productId: string) => {
30     Swal.fire({
31       title: "Are you sure?",
32       text: "You won't be able to revert this!",
33       icon: "warning",
34       showCancelButton: true,
35       confirmButtonColor: "#3085d6",
36       cancelButtonColor: "#d33",
37       confirmButtonText: "Yes, delete it!",
38     }).then(result) => {
39       if (result.isConfirmed) {
40         removeFromCart(productId);
41         setCartItems(getCartItems());
42       }
43     });
44
45   const handleRemove = (productId: string) => {
46     Swal.fire({
47       icon: "warning",
48       showCancelButton: true,
49       confirmButtonColor: "#3085d6",
50       cancelButtonColor: "#d33",
51       confirmButtonText: "Yes, delete it!",
52     }).then(result) => {
53       if (result.isConfirmed) {
54         removeFromCart(productId);
55         setCartItems(getCartItems());
56       }
57     });
58
59   const handleQuantityChange = (id: string, quantity: number) => {
60     updateCartQuantity(id, quantity);
61     setCartItems(getCartItems());
62   };
63
64   const handleIncrement = (id: string, quantity: number) => {
65     handleQuantityChange(id, quantity + 1);
66   };
67
68   const handleDecrement = (id: string, quantity: number) => {
69     if (quantity > 1) {
70       handleQuantityChange(id, quantity - 1);
71     }
72   };
73
74   const calculateTotalPrice = () => {
75     return cartItems.reduce(
76       (total, item) => total + item.price * item.quantity,
77       0
78     );
79   };
80
81   return (
82     <div>
83       <h1>Cart</h1>
84       <table>
85         <thead>
86           <tr>
87             <th>Product Name</th>
88             <th>Image</th>
89             <th>Quantity</th>
90             <th>Price</th>
91             <th>Action</th>
92           </tr>
93         </thead>
94         <tbody>
95           {cartItems.map(item => (
96             <tr>
97               <td>{item.name}</td>
98               <td><Image alt={item.name} width={50} height={50}/></td>
99               <td>{item.quantity}</td>
100              <td>${item.price}</td>
101              <td>
102                <button onClick={()=>handleDecrement(item.id, item.quantity)}>-</button>
103                <button onClick={()=>handleIncrement(item.id, item.quantity)}>+</button>
104              </td>
105            </tr>
106          ))}
107        </tbody>
108      </table>
109      <div>
110        <strong>Total Price:</strong> ${calculateTotalPrice()}
111        <br/>
112        <button onClick={router.back}>Go Back</button>
113      </div>
114    </div>
115  );
116}
```

19°C Haze 1:10 AM 2/5/2025

File Edit Selection View Go ... ← → 🔍 Final Project On next 15

EXPLORER FINAL P... HomeProduct.tsx Header.tsx page.tsx ...\\Shop page.tsx ...\\Cart Footer.tsx page.tsx ...\\app

```
19 const Cart:React.FC = () => {
20   const handleRemove = (productId: string) => {
21     Swal.fire({
22       icon: "warning",
23       showCancelButton: true,
24       confirmButtonColor: "#3085d6",
25       cancelButtonColor: "#d33",
26       confirmButtonText: "Yes, delete it!",
27     }).then(result) => {
28       if (result.isConfirmed) {
29         removeFromCart(productId);
30         setCartItems(getCartItems());
31       }
32     });
33
34   const handleRemove = (productId: string) => {
35     Swal.fire({
36       icon: "warning",
37       showCancelButton: true,
38       confirmButtonColor: "#3085d6",
39       cancelButtonColor: "#d33",
40       confirmButtonText: "Yes, delete it!",
41     }).then(result) => {
42       if (result.isConfirmed) {
43         removeFromCart(productId);
44         setCartItems(getCartItems());
45       }
46     });
47
48   const handleQuantityChange = (id: string, quantity: number) => {
49     updateCartQuantity(id, quantity);
50     setCartItems(getCartItems());
51   };
52
53   const handleIncrement = (id: string, quantity: number) => {
54     handleQuantityChange(id, quantity + 1);
55   };
56
57   const handleDecrement = (id: string, quantity: number) => {
58     if (quantity > 1) {
59       handleQuantityChange(id, quantity - 1);
60     }
61   };
62
63   const calculateTotalPrice = () => {
64     return cartItems.reduce(
65       (total, item) => total + item.price * item.quantity,
66       0
67     );
68
69   return (
70     <div>
71       <h1>Cart</h1>
72       <table>
73         <thead>
74           <tr>
75             <th>Product Name</th>
76             <th>Image</th>
77             <th>Quantity</th>
78             <th>Price</th>
79             <th>Action</th>
80           </tr>
81         </thead>
82         <tbody>
83           {cartItems.map(item => (
84             <tr>
85               <td>{item.name}</td>
86               <td><Image alt={item.name} width={50} height={50}/></td>
87               <td>{item.quantity}</td>
88               <td>${item.price}</td>
89               <td>
90                 <button onClick={()=>handleDecrement(item.id, item.quantity)}>-</button>
91                 <button onClick={()=>handleIncrement(item.id, item.quantity)}>+</button>
92               </td>
93             </tr>
94           ))}
95         </tbody>
96       </table>
97       <div>
98         <strong>Total Price:</strong> ${calculateTotalPrice()}
99         <br/>
100        <button onClick={router.back}>Go Back</button>
101      </div>
102    </div>
103  );
104}
```

Ln 125, Col 12 Spaces: 2 UTF-8 CRLF TypeScript JSX Completions limit reached Prettier

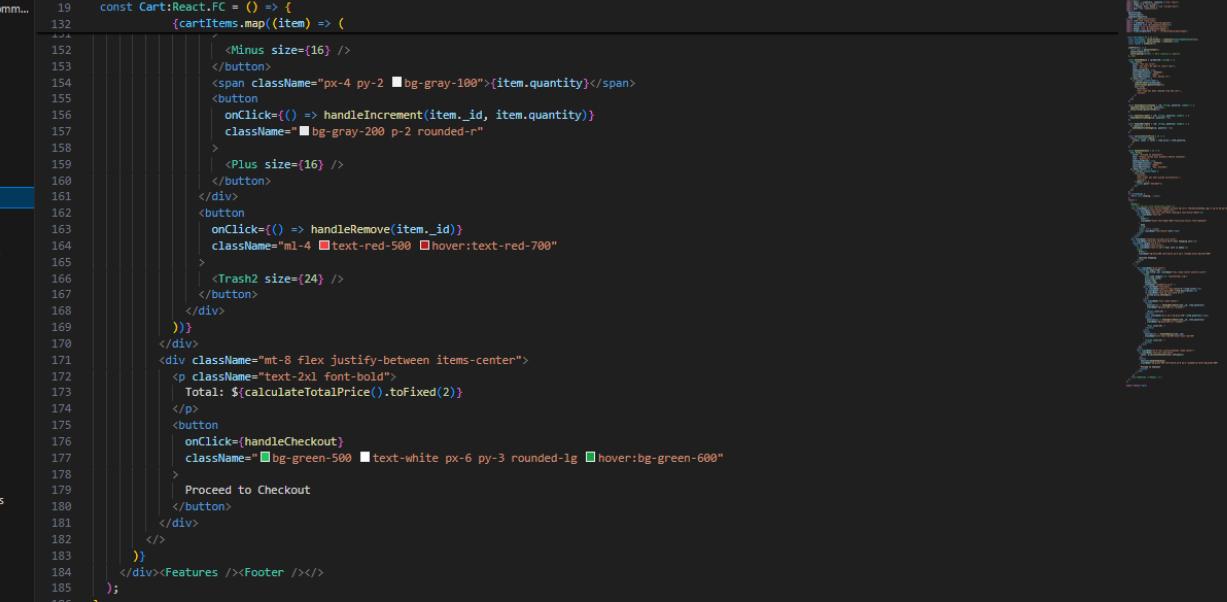
```
const Cart:React.FC = () => {
  const handleCheckout = () => {
    Swal.fire({
      title: "Proceed to Checkout?",
      text: "Please review your products before checkout",
      icon: "question",
      showCancelButton: true,
      confirmButtonColor: "#3088d6",
      cancelButtonColor: "#d33",
      confirmButtonText: "Yes, proceed!"
    }).then(result) => {
      if (result.isConfirmed) {
        Swal.fire(
          "Success",
          "Your order has been placed successfully.",
          "success"
        ).then(() => {
          router.push("/CheckOut");
        });
      }
    };
  };
  if (isLoading) {
    return <div>Loading...</div>;
  }
  return (
    <>
      <Header/>
      {/* Hero Section with background image */}
      <div className="w-full max-w-[1440px] mx-auto bg-[url('/BackgroundImage.jpg')] py-12 md:py-16 bg-cover bg-center bg-no-repeat opacity-90">
        <div className="text-center space-y-5">
          <h1 className="text-5xl font-bold leading-8 text-black">Cart</h1>
          <div className="text-lg">
            <Link href="/" className="hover:text-amber-800 transition-colors font-semibold">
              Home
            </Link>
            <span> {>} </span>
          </div>
        </div>
      </div>
    </>
  );
}
```

The screenshot shows a Microsoft Edge browser window with the title "Final Project On next 15". The address bar shows the URL "Final-Project-Ecommerce-Website / final-project > src > app > Cart > page.tsx [Cart]". The browser's developer tools are open, displaying the DOM structure and the Network tab.

The DOM structure shows a shopping cart page with a header "Your Shopping Cart", a message "Your cart is empty", a "Continue Shopping" link, and a grid of items. One item is visible: "Laptop" by "Dell" for \$1200. The quantity is currently set to 1, indicated by the text "1" and the minus sign icon.

The Network tab shows several requests to "api/cart", including "GET /api/cart" and "PUT /api/cart".

The code editor on the left shows the "Cart" component's logic. It defines a state "cart" and handles actions like "handleDecrement" and "handleIncrement". The component renders a container with a heading, a message if the cart is empty, and a grid of items. Each item has a title, description, price, and a quantity input field with a minus sign icon. The component also includes a "Continue Shopping" link.



The screenshot shows a Microsoft Edge browser window with a tab bar at the top containing 'Final Project On next 15'. The main content area displays a React application for an E-commerce website. The application shows a shopping cart with several items, a total price of \$120.00, and a 'Proceed to Checkout' button. The browser's developer tools are open, showing the DOM structure and network requests.

File Edit Selection View Go ... ↶ ↷ 🔍 Final Project On next 15

EXPLORER ... HomeProduct.tsx Header.tsx page.tsx ...Shop page.tsx ...Cart Footer.tsx page.tsx ...app

FINAL_P... Final-Project-Ecomm... .next node_modules public script src app addtocart Blog Cart page.tsx CheckOut Contact ProductType Shop [id] page.tsx studio favicon.ico # globals.css layout.tsx page.tsx components sanity lib schemaTypes index.ts product.ts env.ts structure.ts .env.local

19 const Cart:React.FC = () => {
122 [cartItems.map(item) => (
152 <Minus size={16} />
153 <button>
154 {item.quantity}
155 <button onClick={() => handleIncrement(item.id, item.quantity)}
156 className="bg-gray-200 p-2 rounded-r"
157 > +<Plus size={16} />
158 </button>
159 </div>
160 <button onClick={() => handleRemove(item.id)}
161 className="ml-4 text-red-500 hover:text-red-700"
162 > <Trash2 size={24}></button>
163 </div>
164))
165 </div>
166 <div className="mt-8 flex justify-between items-center">
167 <p className="text-2xl font-bold">
168 Total: \${calculateTotalPrice().toFixed(2)}
169 </p>
170 <button onClick={handleCheckout}
171 className="bg-green-500 text-white px-6 py-3 rounded-lg hover:bg-green-600"
172 > Proceed to Checkout
173 </button>
174 </div><Features /><Footer /></div>
175)
176);
177 };

178 export default Cart;

main 0 0 0 Amazon Q

Ln 125, Col 12 Spaces: 2 UTF-8 CRLF Completions limit reached Prettier

19°C Haze 11:1 AM 2/5/2025

```
1 import ProductTypeGlobale from "../ProductType/producttypes"
2
3 export const addToCart = (product: ProductTypeGlobale) => {
4   const cart: ProductTypeGlobale[] = JSON.parse(localStorage.getItem("cart") || "[]")
5   const existingProductIndex = cart.findIndex((item) => item._id === product._id)
6
7   if (existingProductIndex > -1) {
8     cart[existingProductIndex].quantity += 1
9   } else {
10    cart.push({ ...product, quantity: 1 })
11  }
12
13  localStorage.setItem("cart", JSON.stringify(cart))
14}
15
16 export const removeFromCart = (productId: string) => {
17   let cart: ProductTypeGlobale[] = JSON.parse(localStorage.getItem("cart") || "[]")
18   cart = cart.filter((item) => item._id !== productId)
19   localStorage.setItem("cart", JSON.stringify(cart))
20 }
21
22 export const updateCartQuantity = (productId: string, quantity: number) => {
23   const cart: ProductTypeGlobale[] = JSON.parse(localStorage.getItem("cart") || "[]")
24   const productIndex = cart.findIndex((item) => item._id === productId)
25   if (productIndex > -1) {
26     cart[productIndex].quantity = quantity
27     localStorage.setItem("cart", JSON.stringify(cart))
28   }
29 }
30
31 export const getCartItems = (): ProductTypeGlobale[] => {
32   if (typeof window !== "undefined") {
33     return JSON.parse(localStorage.getItem("cart") || "[]")
34   }
35   return []
36 }
```

Checkout Flow Components



room, bedroom, or home office, the Nordic Elegance collection creates a sophisticated yet relaxed vibe that suits any modern interior. With natural wood finishes, neutral tones and soft textures, Nordic Elegance fosters a sense of calm and tranquility, making it the perfect choice for those who value simplicity and quality. Ideal for creating a serene, clutter-free space, this collection is a true reflection of Scandinavian design principles: clean, functional, and beautifully crafted. Key Features: Minimalist, Scandinavian-inspired design that emphasizes simplicity and functionality. Crafted from natural materials like wood, stone, and soft textiles for a warm, inviting feel. Light, neutral color palette that complements various interior styles. High-quality craftsmanship for lasting beauty and durability. Idea for creating a peaceful, elegant environment in any room. Bring the peaceful elegance of Nordic design into your home with Nordic Elegance—where simplicity, quality, and style unite to create the ultimate atmosphere of calm and sophistication.

\$280.00

Timber Craft

Introducing TimberCraft—a collection that celebrates the timeless beauty of wood craftsmanship and the art of nature's finest materials. Inspired by the rustic charm and durability of natural wood, TimberCraft brings warmth, character, and a touch of handcrafted elegance to any space. Perfect for those who value authenticity and sustainability, this collection combines the strength of timber with refined design. Each piece is meticulously crafted to highlight the unique grains and textures of the wood, making it a focal point in any room. Whether you're looking for furniture, decor, or accessories, the TimberCraft collection is designed to stand the test of time, with materials chosen for their durability and lasting beauty. From striking tables and chairs to decorative accessories, this collection offers a variety of beautifully designed pieces that will enhance any interior. Features: Made from high-quality, sustainable timber for durability and lasting beauty. Unique wood grains and textures that bring natural beauty to your interior. Perfect for creating a cozy, earthy atmosphere in any room. Bring the timeless charm of natural wood into your home with TimberCraft—where style and substance stand out in both form and function.

\$320.00

Total: \$2080.00



Proceed to Checkout?

Please review your products before checkout

Yes, proceed! Cancel

Proceed to Checkout



High Quality
Crafted from top materials



Warranty Protection
Over 2 years



Free Shipping
Order over 150 \$



24 / 7 Support
Dedicated support



and soft textures, Nordic Elegance fosters a sense of calm and tranquility, making it the perfect choice for those who value simplicity and quality. Ideal for creating a serene, clutter-free space, this collection is a true reflection of Scandinavian design principles: clean, functional, and beautifully crafted. Key Features: Minimalist, Scandinavian-inspired design that emphasizes simplicity and functionality. Crafted from natural materials like wood, stone, and soft textiles for a warm, inviting feel. Light, neutral color palette that complements various interior styles. High-quality craftsmanship for lasting beauty and durability. Ideal for creating a peaceful, elegant environment in any room. Bring the peaceful elegance of Nordic design into your home with Nordic Elegance—where simplicity, quality, and style unite to create the ultimate atmosphere of calm and sophistication.

\$280.00

Timber Craft

Introducing TimberCraft—a collection that celebrates the timeless beauty of wood craftsmanship and the art of nature's finest materials. Inspired by the rustic charm and durability of natural wood, TimberCraft brings warmth, character, and a touch of handcrafted elegance to any space. Perfect for those who value authenticity and sustainability, this collection combines the strength of timber with refined design. Each piece is meticulously crafted to highlight the unique grains and subtle imperfections of the wood. As accents, TimberCraft offers a variety of beautifully designed pieces, from rustic-style dining tables and chairs to more modern, minimalist accessories. Striking tables and chairs to decorative accessories, this collection is a testament to the enduring beauty of wood.

Features: Made from high-quality, sustainable timber for a natural finish. The collection features wood grains and textures that bring natural beauty to your home. Perfect for creating a cozy, earthy atmosphere in any room. Bring the timeless charm of natural wood into your home with TimberCraft, pieces that stand out in both style and substance.

\$320.00

Total: \$2080.00



SUCCESS

Your order has been placed successfully.

OK

Proceed to Checkout



High Quality
crafted from top materials



Warranty Protection
Over 2 years



Free Shipping
Order over 150 \$



24 / 7 Support
Dedicated support



Billing details

Product	Subtotal
 Bold Nest Quantity: 2	Rs. 520.00
 Timeless Elegance Quantity: 1	Rs. 320.00
 Nordic Elegance Quantity: 1	Rs. 280.00

Product	Subtotal
 Bold Nest Quantity: 2	Rs. 520.00
 Timeless Elegance Quantity: 1	Rs. 320.00
 Nordic Elegance Quantity: 1	Rs. 280.00
 Timber Craft Quantity: 3	Rs. 960.00

Subtotal: Rs. 2080.00

Discount: Rs. 0.00

Total: **Rs. 2080.00**

Direct Bank Transfer

Make your payment directly into our bank account. Your order will not be shipped until the funds have cleared in our account.

Cash on Delivery

Place Order

The screenshot shows a Microsoft Visual Studio Code interface with the following details:

- File Explorer:** On the left, it shows the project structure under "FINAL P...". The "src" folder contains "app", "Checkout", and "CheckOut". Inside "Checkout", there is a "page.tsx" file which is currently selected.
- Code Editor:** The main area displays the code for "page.tsx". The code handles a "CheckOut" function that takes "cartItems" and "discount" as parameters. It uses useState to manage state variables like "firstName", "lastName", "address", etc., and "formError".

```
1  "use client";
2
3  import { useState, useEffect } from "react";
4  import { getCartItems } from "../adtocart/action";
5  import Header from "@components/Header";
6  import Footer from "@components/Footer";
7  import Features from "@components/feature";
8  import Image from "next/image";
9  import Link from "next/link";
10 import ProductTypeGloble from "../ProductType/producttypes";
11 function CheckOut() {
12  const [cartItems, setCartItems] = useState([]);
13  const [discount, setDiscount] = useState<number>(0);
14  const [isLoading, setIsLoading] = useState(true);
15  const [formData, setFormData] = useState<{
16    firstName: "",
17    lastName: "",
18    address: "",
19    phone: "",
20    email: "",
21    city: "",
22    state: "",
23    zip: "",
24    country: "",
25    paymentMethod: "",
26    cartNumber: ""
27  }>;
18  const [formError, setFormError] = useState<{
19    firstName: false,
20    lastName: false,
21    address: false,
22    phone: false,
23    email: false,
24    city: false,
25    state: false,
26    zip: false,
27    country: false,
28    paymentMethod: false,
29    cartNumber: false
30  }>;
31}
```
- Search Bar:** At the top center, it says "Final Project On next 15".
- Status Bar:** At the bottom, it shows "Ln 7, Col 45" and "TypeScript JSX".

```
function CheckOut() {
  zip: false,
  country: false,
  paymentMethod: false,
  cartNumber: false,
};

useEffect(() => {
  const fetchCart = getCartItems();
  setCartItems(fetchCart);
  // fetchCart.then((data) => {
  //   setCartItems(data);
  //   setIsloading(false);
  // });

  const appliedDiscount = localStorage.getItem("applyDiscount");
  if (appliedDiscount) {
    setDiscount(Number(appliedDiscount));
  }
  setIsloading(false);
}, []);

const SubTotal = cartItems.reduce(
  (total, item) => total + item.price * item.quantity,
  0
);

const handleInputChange = (e: React.ChangeEvent<HTMLInputElement>) => {
  setFormValue({
    ...formValue,
    [e.target.id]: e.target.value,
  });
};

if (isloading) {
  return (
    <div className="flex justify-center items-center h-screen">
      <div className="animate-spin rounded-full h-32 w-32 border-t-2 border-b-2 border-gray-900">
        Loading...
      </div>
    </div>
  );
}

const Footer = () => {
  return (
    <div>
      <div>
        <div>
          <div>
            <div>
              <div>
                <div>
                  <div>
                    <div>
                      <div>
                        <div>
                          <div>
                            <div>
                              <div>
                                <div>
                                  <div>
                                    <div>
                                      <div>
                                        <div>
                                          <div>
                                            <div>
                                              <div>
                                                <div>
                                                  <div>
                                                    <div>
                                                      <div>
                                                        <div>
                                                          <div>
                                                            <div>
                                                              <div>
                                                                <div>
                                                                  <div>
                                                                    <div>
                                                                      <div>
                                                                        <div>
                                                                          <div>
                                                                            <div>
                                                                              <div>
                                                                                <div>
                                                                                  <div>
                                                                                    <div>
                                                                                      <div>
                                                                                        <div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  );
};

const Header = () => {
  return (
    <div>
      <div>
        <div>
          <div>
            <div>
              <div>
                <div>
                  <div>
                    <div>
                      <div>
                        <div>
                          <div>
                            <div>
                              <div>
                                <div>
                                  <div>
                                    <div>
                                      <div>
                                        <div>
                                          <div>
                                            <div>
                                              <div>
                                                <div>
                                                  <div>
                                                    <div>
                                                      <div>
                                                        <div>
                                                          <div>
                                                            <div>
                                                              <div>
                                                                <div>
                                                                  <div>
                                                                    <div>
                                                                      <div>
                                                                        <div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  );
};

const Home = () => {
  return (
    <div>
      <div>
        <div>
          <div>
            <div>
              <div>
                <div>
                  <div>
                    <div>
                      <div>
                        <div>
                          <div>
                            <div>
                              <div>
                                <div>
                                  <div>
                                    <div>
                                      <div>
                                        <div>
                                          <div>
                                            <div>
                                              <div>
                                                <div>
                                                  <div>
                                                    <div>
                                                      <div>
                                                        <div>
                                                          <div>
                                                            <div>
                                                              <div>
                                                                <div>
                                                                  <div>
                                                                    <div>
                                                                      <div>
                                                                        <div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  );
};

const ProductType = () => {
  return (
    <div>
      <div>
        <div>
          <div>
            <div>
              <div>
                <div>
                  <div>
                    <div>
                      <div>
                        <div>
                          <div>
                            <div>
                              <div>
                                <div>
                                  <div>
                                    <div>
                                      <div>
                                        <div>
                                          <div>
                                            <div>
                                              <div>
                                                <div>
                                                  <div>
                                                    <div>
                                                      <div>
                                                        <div>
                                                          <div>
                                                            <div>
                                                              <div>
                                                                <div>
                                                                  <div>
                                                                    <div>
                                                                      <div>
                                                                        <div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  );
};

const Contact = () => {
  return (
    <div>
      <div>
        <div>
          <div>
            <div>
              <div>
                <div>
                  <div>
                    <div>
                      <div>
                        <div>
                          <div>
                            <div>
                              <div>
                                <div>
                                  <div>
                                    <div>
                                      <div>
                                        <div>
                                          <div>
                                            <div>
                                              <div>
                                                <div>
                                                  <div>
                                                    <div>
                                                      <div>
                                                        <div>
                                                          <div>
                                                            <div>
                                                              <div>
                                                                <div>
                                                                  <div>
                                                                    <div>
                                                                      <div>
                                                                        <div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  );
};

const Blog = () => {
  return (
    <div>
      <div>
        <div>
          <div>
            <div>
              <div>
                <div>
                  <div>
                    <div>
                      <div>
                        <div>
                          <div>
                            <div>
                              <div>
                                <div>
                                  <div>
                                    <div>
                                      <div>
                                        <div>
                                          <div>
                                            <div>
                                              <div>
                                                <div>
                                                  <div>
                                                    <div>
                                                      <div>
                                                        <div>
                                                          <div>
                                                            <div>
                                                              <div>
                                                                <div>
                                                                  <div>
                                                                    <div>
                                                                      <div>
                                                                        <div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  );
};

const Cart = () => {
  return (
    <div>
      <div>
        <div>
          <div>
            <div>
              <div>
                <div>
                  <div>
                    <div>
                      <div>
                        <div>
                          <div>
                            <div>
                              <div>
                                <div>
                                  <div>
                                    <div>
                                      <div>
                                        <div>
                                          <div>
                                            <div>
                                              <div>
                                                <div>
                                                  <div>
                                                    <div>
                                                      <div>
                                                        <div>
                                                          <div>
                                                            <div>
                                                              <div>
                                                                <div>
                                                                  <div>
                                                                    <div>
                                                                      <div>
                                                                        <div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  );
};

const CheckOut = () => {
  return (
    <div>
      <div>
        <div>
          <div>
            <div>
              <div>
                <div>
                  <div>
                    <div>
                      <div>
                        <div>
                          <div>
                            <div>
                              <div>
                                <div>
                                  <div>
                                    <div>
                                      <div>
                                        <div>
                                          <div>
                                            <div>
                                              <div>
                                                <div>
                                                  <div>
                                                    <div>
                                                      <div>
                                                        <div>
                                                          <div>
                                                            <div>
                                                              <div>
                                                                <div>
                                                                  <div>
                                                                    <div>
                                                                      <div>
                                                                        <div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  );
};

const Footer = () => {
  return (
    <div>
      <div>
        <div>
          <div>
            <div>
              <div>
                <div>
                  <div>
                    <div>
                      <div>
                        <div>
                          <div>
                            <div>
                              <div>
                                <div>
                                  <div>
                                    <div>
                                      <div>
                                        <div>
                                          <div>
                                            <div>
                                              <div>
                                                <div>
                                                  <div>
                                                    <div>
                                                      <div>
                                                        <div>
                                                          <div>
                                                            <div>
                                                              <div>
                                                                <div>
                                                                  <div>
                                                                    <div>
                                                                      <div>
                                                                        <div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  );
};

const page = () => {
  return (
    <div>
      <Header />
      <CheckOut />
      <Cart />
      <Footer />
    </div>
  );
};

export default page;
```

File Edit Selection View Go ... ⏪ Final Project On next 15 ⏩

EXPLORER FINAL P... Header.tsx page.tsx ...Shop page.tsx ...CheckOut Footer.tsx page.tsx ...app

```
11 function Checkout() {
  ...
  const ValidateForm = () => {
    const error = {
      firstName: !formValue.firstName,
      lastName: !formValue.lastName,
      address: !formValue.address,
      phone: !formValue.phone,
      email: !formValue.email,
      city: !formValue.city,
      state: !formValue.state,
      zip: !formValue.zip,
      country: !formValue.country,
      paymentMethod: !formValue.paymentMethod,
      cardNumber: !formValue.cardNumber,
    };
    setFormError(error);
    return Object.values(error).every((error) => !error);
  };
  const handleSubmit = (e: React.FormEvent<HTMLFormElement>) => {
    e.preventDefault();
    if (ValidateForm()) {
      localStorage.removeItem("applyDiscount");
      console.log("Form Submitted", formValue);
    }
  };
  ...
  return (
    ...
    <Header />
    {/* Hero Section with background image */}
    <div className="w-full max-w-[1440px] mx-auto bg-[url('/BackgroundImage.jpg')] py-12 md:py-16 bg-cover bg-center bg-no-repeat opacity-90">
      <div className="text-center space-y-5">
        <h1 className="text-5xl font-bold leading-8 text-black">Checkout</h1>
        <div className="text-lg">
          <Link href="#">
            ...
            ...
          </Link>
        </div>
      </div>
    </div>
  );
}
```

main ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ Amazon Q

Ln 7, Col 45 Spaces: 2 UTF-8 CRLF () TypeScript JSX Completions limit reached ✓ Prettier

File Edit Selection View Go ... ⏪ Final Project On next 15 ⏩

EXPLORER FINAL P... Header.tsx page.tsx ...Shop page.tsx ...CheckOut Footer.tsx page.tsx ...app

```
11 function Checkout() {
  ...
  </div>
  <div className="min-h-screen bg-gray-50 py-12 px-6 sm:px-8 lg:px-10">
    <div className="max-w-7xl mx-auto">
      <h1 className="text-3xl font-semibold mb-8">Billing details</h1>

      <div className="grid grid-cols-1 lg:grid-cols-2 gap-12">
        {/* Billing Form */}
        <div>
          <form onSubmit={handleSubmit} className="space-y-6">
            <div className="grid grid-cols-1 md:grid-cols-2 gap-6">
              <div>
                <input type="text" placeholder="First Name" id="firstName" value={formValue.firstName} onChange={handleInputChange} className="w-full px-4 py-3 rounded-lg border ${formError.firstName ? "border-red-500" : "border-gray-300"} focus:outline-none focus:border-gray-500" />
              </div>
              <div>
                <input type="text" placeholder="Last Name" id="lastName" value={formValue.lastName} onChange={handleInputChange} className="w-full px-4 py-3 rounded-lg border ${formError.lastName ? "border-red-500" : "border-gray-300"} focus:outline-none focus:border-gray-500" />
              </div>
              <div>
                <select id="country" value={formValue.country} onChange={(e) => setFormValue({ ...formValue, country: e.target.value })}>
                  ...
                </select>
              </div>
            </div>
          </form>
        </div>
      </div>
    </div>
  </div>
}
```

main ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ Amazon Q

Ln 7, Col 45 Spaces: 2 UTF-8 CRLF () TypeScript JSX Completions limit reached ✓ Prettier

```
File Edit Selection View Go ... ← → ⚙ Final Project On next 15
EXPLORER ... FINAL P... 🏷️ 🎯 🔍
FINAL Project-E-commerce-Website > final-project > src > app > CheckOut > page.tsx ...app
11   function Checkout() {
12
13     const [formValue, setFormValue] = useState({
14       country: "Select Country/Region",
15       address: "Street address",
16       city: "Town / City",
17       zip: ""
18     });
19
20     const [formError, setFormError] = useState({
21       country: false,
22       address: false,
23       city: false,
24       zip: false
25     });
26
27     const handleInputChange = (e) => {
28       const { name, value } = e.target;
29       setFormValue({ ...formValue, [name]: value });
30
31       if (name === "country" && value === "Select Country/Region") {
32         setFormError({ ...formError, country: true });
33       } else {
34         setFormError({ ...formError, country: false });
35       }
36
37       if (name === "address" && value === "") {
38         setFormError({ ...formError, address: true });
39       } else {
40         setFormError({ ...formError, address: false });
41       }
42
43       if (name === "city" && value === "") {
44         setFormError({ ...formError, city: true });
45       } else {
46         setFormError({ ...formError, city: false });
47       }
48
49       if (name === "zip" && value === "") {
50         setFormError({ ...formError, zip: true });
51       } else {
52         setFormError({ ...formError, zip: false });
53       }
54     };
55
56
57     const handleSubmit = (e) => {
58       e.preventDefault();
59
60       if (Object.values(formError).filter(Boolean).length === 0) {
61         alert("Form submitted successfully!");
62       } else {
63         alert("Please fill in all required fields.");
64       }
65     };
66
67
68   return (
69     <div>
70       <form onSubmit={handleSubmit}>
71         <div>
72           <select
73             id="country"
74             value={formValue.country}
75             onChange={(e) => handleInputChange(e)}
76           >
77             <option value="">Select Country/Region</option>
78             <option value="LK">Sri Lanka</option>
79           </select>
80         </div>
81
82         <div>
83           <input
84             type="text"
85             placeholder="Street address"
86             id="address"
87             value={formValue.address}
88             onChange={handleInputChange}
89           >
90         </div>
91
92         <div>
93           <input
94             type="text"
95             placeholder="Town / City"
96             id="city"
97             value={formValue.city}
98             onChange={handleInputChange}
99           >
100        </div>
101
102        <div>
103          <input
104            type="text"
105            placeholder="Zip Code"
106            id="zip"
107            value={formValue.zip}
108            onChange={handleInputChange}
109          >
110        </div>
111
112      </form>
113    </div>
114  );
115
116  <script>
117    window.addEventListener("load", () => {
118      document.getElementById("country").focus();
119    });
120  </script>
121
122  <style>
123    <!-- CSS styles for form elements -->
124  </style>
125
126
```

```
function CheckOut() {
  return (
    <div>
      <input
        type="text"
        placeholder="State / Province"
        id="state"
        value={formValue.state}
        onChange={handleInputChange}
        className="w-full px-4 py-3 rounded-lg border ${formError.state ? "border-red-500" : "border-gray-300"} focus:outline-none focus:ring border-2 border-gray-300">
      </div>

      <div>
        <input
          type="text"
          placeholder="Phone"
          id="phone"
          value={formValue.phone}
          onChange={handleInputChange}
          className="w-full px-4 py-3 rounded-lg border ${formError.phone ? "border-red-500" : "border-gray-300"} focus:outline-none focus:ring border-2 border-gray-300">
      </div>

      <div>
        <input
          type="email"
          placeholder="Email"
          id="email"
          value={formValue.email}
          onChange={handleInputChange}
          className="w-full px-4 py-3 rounded-lg border ${formError.email ? "border-red-500" : "border-gray-300"} focus:outline-none focus:ring border-2 border-gray-300">
      </div>
    </div>
  )
}

export default page
```

File Edit Selection View Go ... ← → ⚡ Final Project On next 15

EXPLORER FINAL PROJECT ON NEXT 15

- Final-Project-Ecomm...
- .next
- node_modules
- public
- script
- src
 - app
 - addtocart
 - Blog
 - Cart
 - CheckOut
 - page.tsx
 - Contact
 - ProductType
 - Shop
 - [id]
 - page.tsx
 - page.tsx
 - studio
- favicon.ico
- # globals.css
- layout.tsx
- page.tsx
- components
- sanity
- lib
- schemaTypes
 - TS index.ts
 - TS products
 - TS env.ts
 - TS structure.ts
 - \$.env.local
- OUTLINE
- TIMELINE

HomeProduct.tsx Header.tsx page.tsx ...Shop page.tsx ...CheckOut Footer.tsx page.tsx ...app

```
11 function CheckOut() {  
  220      </div>  
  221      <div className="flex justify-center items-center min-h-screen bg-gray-100">  
  222        <div className="bg-white p-6 rounded-lg shadow-md max-w-md w-full">  
  223          <div className="mb-6">  
  224            <div className="flex justify-between border-b pb-4 mb-4">  
  225              <h2 className="text-lg font-bold">Product</h2>  
  226              <h2 className="text-lg font-bold">Subtotal</h2>  
  227            </div>  
  228            <div items={cartItems.length > 0 ? cartItems.map(item) : <div>No items in cart</div>}>  
  229              <div key={item.id} className="flex justify-between items-center border-b py-4">  
  230                <Image alt={item.title} src={item.imageUrl} className="w-16 h-16 object-cover rounded" width={64} height={64}/>  
  231                <div>  
  232                  <h3 className="text-sm font-medium">{item.title}</h3>  
  233                  <p className="text-gray-500 text-sm">Quantity: {item.quantity}</p>  
  234                </div>  
  235                <div>  
  236                  <p className="text-gray-700 font-medium">Rs. {(item.price * item.quantity).toFixed(2)}</p>  
  237                </div>  
  238            </div>  
  239          </div>  
  240        </div>  
  241      </div>  
  242    </div>  
  243  </div>  
  244  </div>  
  245  </div>  
  246  </div>  
  247  </div>  
  248  </div>  
  249  </div>  
  250  </div>  
  251  </div>  
  252  </div>  
  253  </div>  
  254  </div>  
  255  </div>  
  256  </div>  
  257  </div>
```

Ln 7, Col 45 Spaces: 2 UTF-8 CRLF () TypeScript JSX Completions limit reached Prettier

File Edit Selection View Go ... ← → ⚡ Final Project On next 15

EXPLORER FINAL P... 🔍 ⌂ ⌂ ⌂

- Final-Project-Ecomm...
- .next
- node_modules
- public
- script
- src
 - app
 - addtocart
 - Blog
 - Cart
 - CheckOut
 - page.tsx
 - Contact
 - ProductType
 - Shop
 - [id]
 - page.tsx
 - page.tsx
 - studio
- favicon.ico
- # globals.css
- layout.tsx
- page.tsx
- components
- sanity
- lib
- schemaTypes
 - TS index.ts
 - TS products
 - TS env.ts
 - TS structure.ts
 - \$.env.local
- OUTLINE
- TIMELINE

HomeProduct.tsx Header.tsx page.tsx ...Shop page.tsx ...CheckOut Footer.tsx page.tsx ...app

```
11 function CheckOut() {  
  256      </div>  
  257      <div className="text-sm space-y-2 border-t pt-4">  
  258        <p className="flex justify-between">  
  259          <span className="font-medium">Subtotal:</span>  
  260          <span>Rs. {SubTotal.toFixed(2)}</span>  
  261        </p>  
  262        <p className="flex justify-between">  
  263          <span className="font-medium">Discount:</span>  
  264          <span>Rs. {discount.toFixed(2)}</span>  
  265        </p>  
  266        <p className="flex justify-between text-lg font-bold">  
  267          <span>Total:</span>  
  268          <span>Rs. {((SubTotal - discount).toFixed(2))}</span>  
  269        </p>  
  270      </div>  
  271      <div className="mt-6">  
  272        <div className="space-y-2">  
  273          <div>  
  274            <input type="radio" id="bank-transfer" name="payment" className="mr-2" defaultChecked/>  
  275            <label htmlFor="bank-transfer" className="text-sm font-medium">Direct Bank Transfer</label>  
  276            <p className="text-xs text-gray-500 ml-6">Make your payment directly into our bank account. Your order will not be shipped until the funds have cleared in our account.</p>  
  277          </div>  
  278        </div>  
  279      </div>  
  280      <div>  
  281        <input type="radio" id="cash-on-delivery"/>
```

Ln 7, Col 45 Spaces: 2 UTF-8 CRLF () TypeScript JSX Completions limit reached Prettier

```

    function CheckOut() {
      ...
    }
  
```

Analyze Website Performance with Lighthouse

http://localhost:3000/

Performance	Accessibility	Best Practices	SEO
34	88	100	92

There were issues affecting this run of Lighthouse:

- The tested device appears to have a slower CPU than Lighthouse expects. This can negatively affect your performance score. Learn more about [calibrating an appropriate CPU slowdown multiplier](#).

34

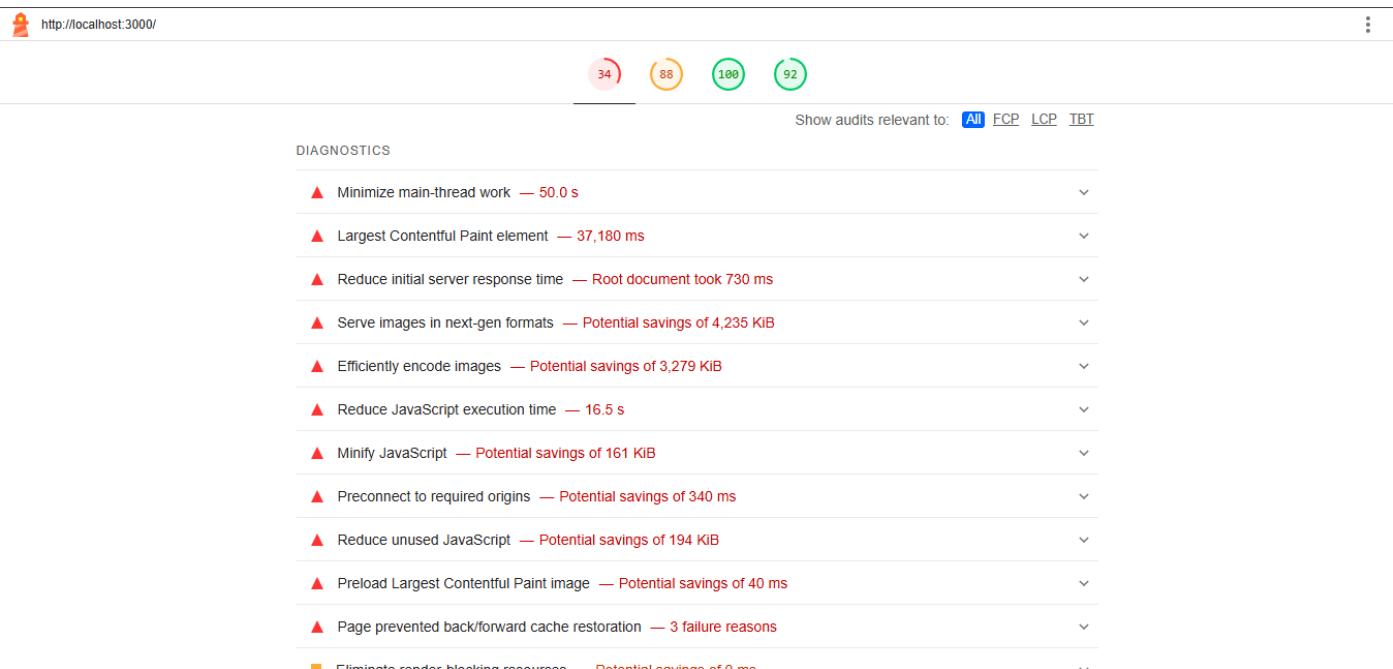
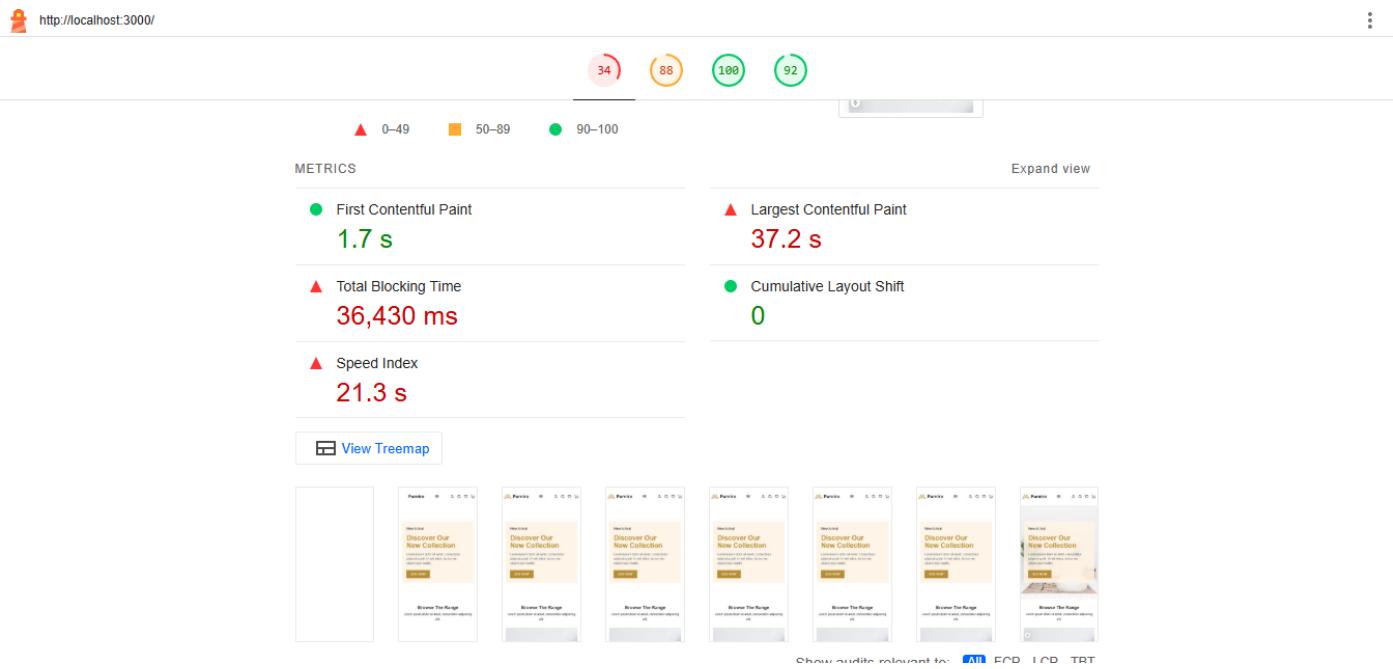
Performance

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

▲ 0–49 ■ 50–89 ● 90–100

Discover Our New Collection

Browse The Range



 http://localhost:3000/ 

34 88 100 92

- Avoid serving legacy JavaScript to modern browsers — Potential savings of 0 KiB
- Avoid enormous network payloads — Total size was 6,839 KiB
- Avoid long main-thread tasks — 20 long tasks found
- Avoids an excessive DOM size — 576 elements
- Avoid chaining critical requests — 1 chain found
- Minimize third-party usage — Third-party code blocked the main thread for 0 ms

More information about the performance of your application. These numbers don't [directly affect](#) the Performance score.

PASSED AUDITS (20)

Show

 http://localhost:3000/ 

34 88 100 92

88

Accessibility

These checks highlight opportunities to [improve the accessibility of your web app](#). Automatic detection can only detect a subset of issues and does not guarantee the accessibility of your web app, so [manual testing](#) is also encouraged.

NAMES AND LABELS

▲ Buttons do not have an accessible name

These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

CONTRAST

 http://localhost:3000/ 

▲ Buttons do not have an accessible name

These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

CONTRAST

▲ Background and foreground colors do not have a sufficient contrast ratio.

These are opportunities to improve the legibility of your content.

NAVIGATION

▲ Heading elements are not in a sequentially-descending order

These are opportunities to improve keyboard navigation in your application.

ADDITIONAL ITEMS TO MANUALLY CHECK (10) Show

These items address areas which an automated testing tool cannot cover. Learn more in our guide on [conducting an accessibility review](#).

 http://localhost:3000/ 

ADDITIONAL ITEMS TO MANUALLY CHECK (10) Show

These items address areas which an automated testing tool cannot cover. Learn more in our guide on [conducting an accessibility review](#).

PASSED AUDITS (20) Show

NOT APPLICABLE (34) Show



100

Best Practices

 http://localhost:3000/ 

TRUST AND SAFETY

- Ensure CSP is effective against XSS attacks
- Use a strong HSTS policy
- Ensure proper origin isolation with COOP

GENERAL

▲ Missing source maps for large first-party JavaScript

PASSED AUDITS (14) Show

NOT APPLICABLE (2) Show

 http://localhost:3000/ ⋮

34 88 100 92

92
SEO

These checks ensure that your page is following basic search engine optimization advice. There are many additional factors Lighthouse does not score here that may affect your search ranking, including performance on [Core Web Vitals](#). [Learn more about Google Search Essentials](#).

CRAWLING AND INDEXING

⚠️ robots.txt is not valid [Lighthouse was unable to download a robots.txt file](#) ▼

To appear in search results, crawlers need access to your app.

ADDITIONAL ITEMS TO MANUALLY CHECK (1) Show

Run these additional validators on your site to check additional SEO best practices.

 http://localhost:3000/ ⋮

34 88 100 92

To appear in search results, crawlers need access to your app.

ADDITIONAL ITEMS TO MANUALLY CHECK (1) Show

Run these additional validators on your site to check additional SEO best practices.

PASSED AUDITS (8) Show

NOT APPLICABLE (1) Show

🕒 Captured at Feb 5, 2025, 4:02 PM GMT+5 💻 Emulated Moto G Power with Lighthouse 12.3.0 ✖ Single page session

⌚ Initial page load ✖ Slow 4G throttling ✖ Using Chromium 132.0.0.0 with cli

Generated by **Lighthouse** 12.3.0 | [File an issue](#)

Developer Resource

Dimensions: Responsive ▾ 768 × 737 74% ▾ DPR: 3.0 ▾

New Arrival

Discover Our New Collection

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut elit tellus, luctus nec ullamcorper mattis.

[BUY NOW](#)

11 resources

Network tab showing resource list:

Status	URL	Initiator	Total Bytes	Error
success	/node_modules/_next_dist_compiled_react-dom_1f56dc...js	http://local...	1 375 517	
success	/node_modules/_next_dist_compiled_107ce8...js	http://local...	391 108	
success	/node_modules/_%40swc_helpers_cjs_00636a...js	http://local...	5 662	
success	http://localhost:3000/_next.../_d95469...js.map	http://local...	105 793	
success	http://localhost:3000/_next.../_d95469...js.map	http://local...	131 566	
success	http://localhost:3000/_next.../_a91c21...js.map	http://local...	150 542	
success	http://localhost:3000/_.../_src_dfc5212...js.map	http://local...	12 366	
success	/node_modules/_next_dist_client_523921...js.m	http://local...	1 059 476	
success	http://localhost:3000/_node_modules/_fd23bb...js.map	http://local...	1 018 452	
success	%5Bturbopack%5D_browser_dev_hmr-client.js	http://local...	31 112	
success	%5Broot%20of%20the%20server%5D_8bbb6	http://local...	46 767	

Elements tab showing DOM structure:

```
<div class="relative h-full flex items-center px-4"> ... </div>
```

Performance tab showing CPU usage over time:

CPU usage: 66.1% (Peak: 100%)

JS heap size: 10.2 MB (Peak: 20.0 MB)

DOM Nodes: 1,470 (Peak: 2,000)

JS event listeners: 806 (Peak: 1,000)