

Market Place Builder Hackathon: Day 4

Building Dynamic Frontend Components

Overview

On Day 4 of the Market Place Builder Hackathon, we focused on enhancing the app by building dynamic frontend components. The main goal was to implement a seamless shopping experience by integrating the following features:

1. **Cart Page**
 2. **Functionality for the "Add to Cart" Button**
 3. **Cart Summary**
 4. **Checkout Page**
-

Features Implemented

1. **Add Cart Page**



Tropical Vibe

Escape to paradise with TropicalVibe—a collection that brings the vibrant energy...

tropical

vibe

furniture

exotic

decor

Price: \$550.00

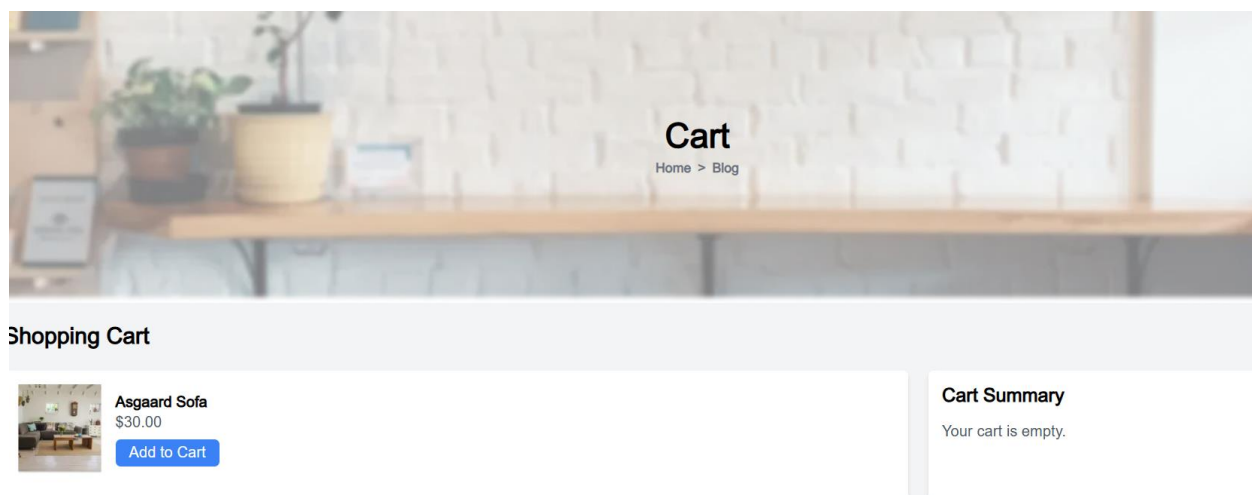
Add to Cart

2.

- **Description:** A dedicated page to display the items added to the cart.
- **Features:**
 - List of products added to the cart.
 - Quantity selection for each product.
 - Option to remove items from the cart.
- **Technologies:** React, Material-UI, or Tailwind CSS for styling.

```
HACKATON-MAIN - COPY  hackaton-main > src > app > cart > page.tsx > [x] Cart
144
145 "use client"
146 import React, { useState } from 'react';
147 import Link from 'next/link';
148 import Image from 'next/image';
149 import { Header } from '@components/Header';
150 import Footer from '@components/Footer';
151
152 // Define product type for better type safety
153 interface Product {
154   id: number;
155   name: string;
156   price: number;
157   image: string;
158   quantity?: number;
159 }
160
161 const Cart = () => {
162   const [cartItems, setCartItems] = useState<Product[]>([]); // State to track cart items
163
164   const handleAddToCart = (product: Product) => {
165     setCartItems((prevItems) => {
166       // Check if the product is already in the cart
167       const existingItem = prevItems.find((item) => item.id === product.id);
168       if (existingItem) {
169         // Update quantity if product exists
170         return prevItems.map((item) =>
171           item.id === product.id
172             ? { ...item, quantity: (item.quantity || 1) + 1 }
173             : item
174         );
175       }
176       // Add new product to the cart
177       return [...prevItems, { ...product, quantity: 1 }];
178     });
179   };
180 }
```

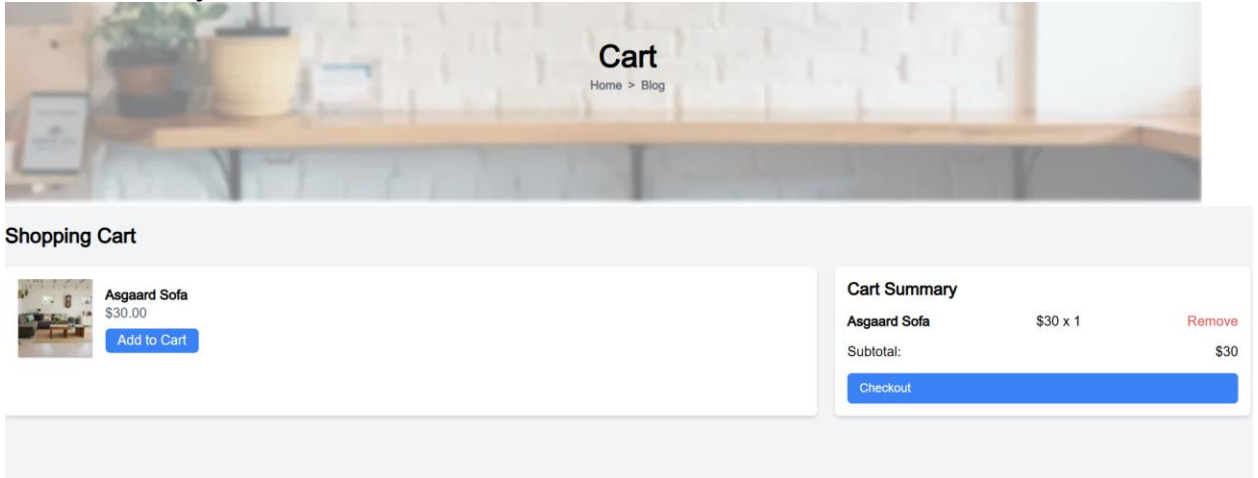
2. Functionality for the "Add to Cart" Button



- **Description:** The "Add to Cart" button allows users to add products to their cart dynamically.
- **Steps:**
 - Capture the selected product details when the button is clicked.
 - Update the cart state with the new item.
 - Provide visual feedback (e.g., toast notification) to confirm the action.
- **Implementation Details:**

- Use React hooks (`useState` and `useContext`) to manage cart state globally.
- Ensure that duplicate items are handled by incrementing the quantity instead of adding a new entry.

3. Cart Summary



4.

- **Description:** Display a summary of the cart's contents, including the total cost.
- **Features:**
 - List of products with prices and quantities.
 - Subtotal calculation.
 - Tax estimation (if applicable).
 - Grand total.
- **UI Enhancements:**
 - Highlight discounted items (if applicable).
 - Responsive design to adapt to different screen sizes.

5. Checkout Page:

hackaton-main	17	const Billing = () => {
.next	242	</button>
node_modules	243	</form>
public	244	</div>
src	245	
app	246	{/* Order Summary Section */}
> about	247	<div style={{ flex: 1, background: "white", padding: "60px" }}>
> blog	248	{/* Order Summary */}
> cart	249	<div
page.tsx	250	style={{
checkout	251	display: "flex",
page.tsx	252	justifyContent: "space-between",
contact	253	marginBottom: "20px",
product	254	}}>
page.tsx	255	>
shop	256	<h2>Product</h2>
studio	257	<h2>Subtotal</h2>
favicon.ico	258	</div>
globals.css	259	<div style={{ display: "flex", justifyContent: "space-between" }}>
layout.tsx	260	<p>Asgaard Sofa * 1</p>
page.tsx	261	<p>\$250</p>
components	262	</div>
pages	263	<div style={{ display: "flex", justifyContent: "space-between" }}>
> blog	264	<p>Another Product * 2</p>
> cart	265	<p>\$150</p>
page.tsx	266	</div>
contact	267	<div
shop	268	style={{
cartContext.ts	269	display: "flex",
	270	justifyContent: "space-between",
	271	marginTop: "20px",
	272	fontWeight: "bold",
	273	}}>

Challenges Faced

1. Managing global state for the cart across multiple components.
 2. Ensuring accurate price calculations in the cart summary.
 3. Handling edge cases such as adding the same item multiple times.
-

Conclusion

Day 4 of the hackathon was productive as we successfully implemented core e-commerce functionalities, improving the user experience and making significant progress toward completing the project.