

Day 4 - Dynamic Frontend Component

Introduction:

This document provides a detailed overview of the development and implementation of the **Dynamic Frontend Component** for an online marketplace. The project focuses on building essential features such as a dynamic product listing page, an advanced cart with persistent state, pagination, and an interactive checkout flow. All components were designed to deliver an efficient, responsive, and user-friendly eCommerce experience, with seamless API integration to ensure real-time data rendering and functionality.

Functional Deliverables

1. Product Listing Component:

The product listing page is designed to dynamically render product data in a grid layout. Key features include:

- **Fields Displayed:**
 - Product Name
 - Price
 - Image
 - Stock Status (In Stock/Out of Stock)
- **Layout:**
 - Card-based design to display product details for enhanced user engagement.
 - Fully responsive grid layout for consistent rendering across devices.



Vase Set

Elevate your home decor with the timeless beauty of the Vase Set. Designed to complement any interior...

Rs 150 In Stock



Bed

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

Rs 250 In Stock



Zen Table

Introduce tranquility and balance into your home with ZenTable—a beautifully crafted piece designed ...

Rs 250 Low Stock



Tropical Vibe

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

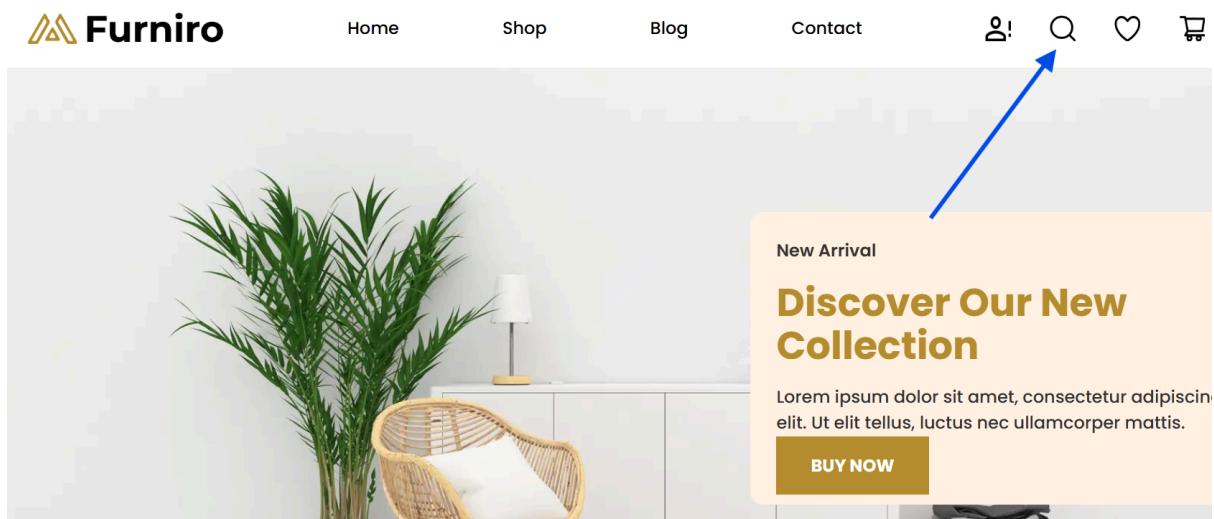
Rs 550 In Stock

2. Search Bar and Pagination

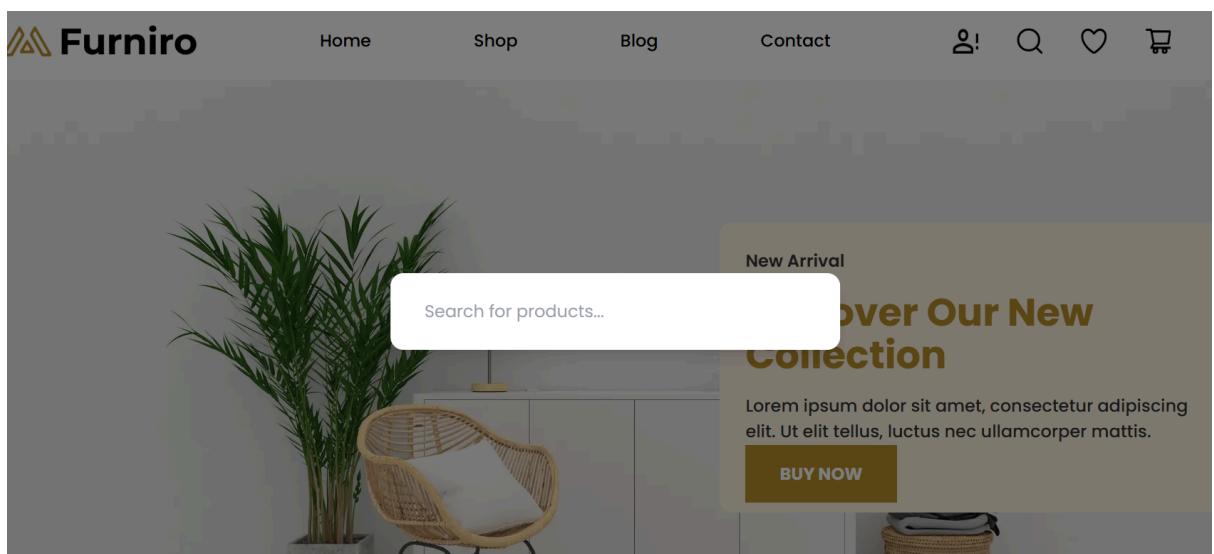
Features implemented:

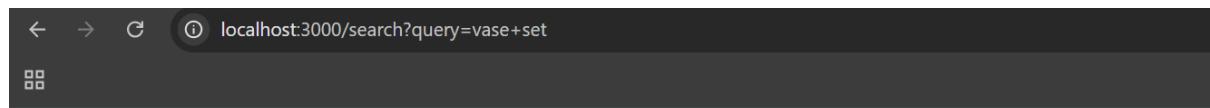
- **Search Bar:**

- Filters products dynamically by name to provide instant results.



- Search Bar with Pop-Up UI Design



[Home](#)[Shop](#)[Blog](#)[Contact](#)

Search Results for "vase set"



Vase Set

Elevate your home decor with the timeless beauty of the Vase Set. Designed to complement any interior...

Rs 150 [In Stock](#)



Rustic Vase Set

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

Rs 210 [In Stock](#)

● Pagination Component:

- Divides large product lists into manageable pages.
- Includes "Next" button functionality for effortless navigation through pages.



Modern Serenity

Introducing ModernSerenity—a collection that redefines contemporary living by combining tranquility ...

Rs 480 [In Stock](#)



Timber Craft

Introducing TimberCraft—a collection that celebrates the timeless beauty of wood craftsmanship and t...

Rs 320 [In Stock](#)



Amber Haven

Step into a world of warmth and tranquility with Amber Haven—a collection inspired by the golden glo...

Rs 150 [In Stock](#)



Bold Nest

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

Rs 260 [In Stock](#)

1 2 [Next](#)

3. Cart Component:

- Displays added items with key details:
 - Product Name
 - Quantity
 - Price per item and total price
- **State Management:**
 - Uses **local storage** to track cart items and preserve data across page refreshes for a seamless user experience.

The screenshot displays a 'Cart' component with the following details:

Product	Price	Quantity	Subtotal
Bed	250	5	Rp 1,250
Sunny Chic	400	2	Rp 800

Cart Totals

Subtotal	Rs. 2,050
Total	Rs. 2,050

Check Out

4. Cart Sidebar:

- **Quick Access:** Provides a collapsible sidebar for easy access to cart items.
- **Real-Time Updates:** Displays added items with key details:
 - Product Name
 - Quantity
 - Price per item and total price
- **User-Friendly Design:**
 - Editable quantities and removal of items directly from the sidebar.
 - Automatically updates the total price dynamically.
- **Persistent State Management:** Utilizes local storage to track cart items, ensuring data is preserved across page refreshes for a seamless user experience.

This screenshot shows a product listing interface with a sidebar for the shopping cart.

Product Listings:

- Tropical Vibe**: Rs 550, In Stock. Description: Escape to paradise with TropicalVibe—a collection that brings the vibrant energy of the tropics right... (Image: Living room with green leaf wall)
- Modern Serenity**: Rs 480, In Stock. Description: Introducing ModernSerenity—a collection that redefines contemporary living by combining tranquility ... (Image: Bright living room with large windows)
- Marble Ease**: Rs 419, In Stock. Description: Introducing MarbleEase—a luxurious collection that brings the timeless elegance of marble into your ... (Image: Room with marble cabinet)

Shopping Cart Sidebar:

- Tropical Vibe: 15 x Rs. 550
- Marble Ease: 3 x Rs. 419
- Retro Vibe: 1 x Rs. 340

Cart Summary:

- Subtotal**: Rs. 9,847
- Buttons:** Cart, Checkout, Comparison

5. Filter Panel Component

Provides intuitive filtering options to improve product discovery:

- **Sorting Options:**
 - **Price range:** Low-to-High, High-to-Low, Default, or Newest arrivals.

This screenshot shows a product listing interface with a sorting dropdown open.

Product Listings:

- Wood Chair
- Vase Set
- Amber Haven
- The Dandy chair

Sort by: Price: Lc ▾ (Default selected)

- Default
- Price: Low to High**
- Price: High to Low
- Newest

- **Products Per Page:**
 - Adjustable settings to control the number of products displayed.
- **Availability Toggle:**
 - Option to filter products by stock status, showing only "In Stock" items for more efficient browsing.

Filter Showing 1–6 of 24 results

In Stock Show 6 Sort by Default ▾

			
Vase Set Elevate your home decor with the timeless beauty of the Vase Set. Designed to complement any interior...	Bed Introducing the Bed—your sanctuary for rest and relaxation, designed with both comfort and style in ...	Tropical Vibe Escape to paradise with TropicalVibe—a collection that brings the vibrant energy of the tropics right...	Sleek Living Welcome to SleekLiving, where modern sophistication meets functional design for the ultimate living ...
Rs 150 	Rs 250 	Rs 550 	Rs 300 

6. Footer and Header Components

- **Header Features:**

- Includes navigation links to essential sections like Home, Shop, Blog and Contact.
- Ensures responsiveness across devices for optimal usability.

Furniro

Home Shop Blog Contact

- **Footer Features:**

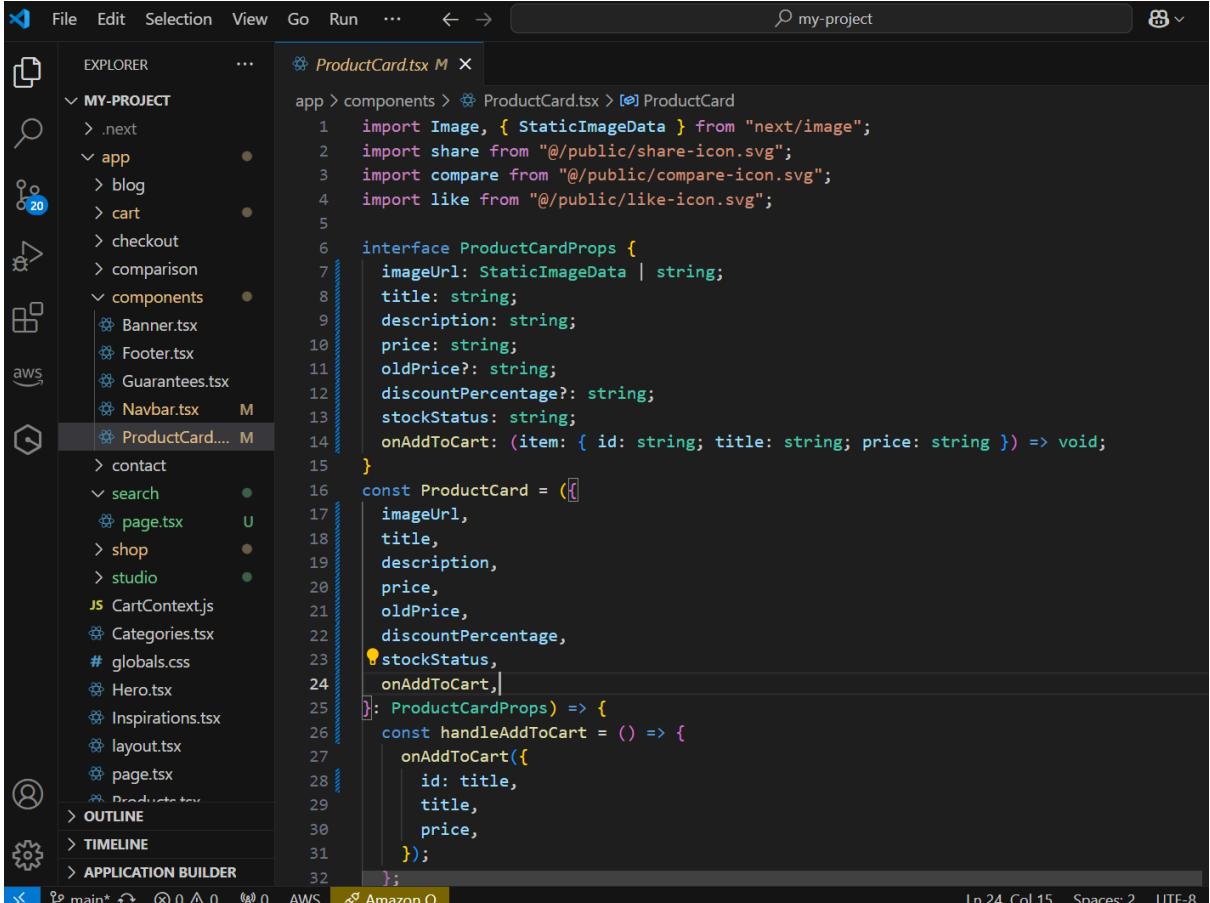
- Includes links to Return, Privacy Policy, and additional resources.
- Designed with accessibility and responsive behavior in mind.

Funiro.	Links	Help	Newsletter
400 University Drive Suite 200 Coral Gables, FL 33134 USA	Home	Payment Options	<input type="text"/> Enter Your Email Address SUBSCRIBE
	Shop	Return	
	About	Privacy Policies	
	Contact		
<hr/> 2023 furino. All rights reserved			

Code Deliverables

Key Components and Features:

1. **ProductCard Component:** Displays product details dynamically within the grid layout.



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

- File Path:** ProductCard.tsx
- Project Structure (Explorer):** MY-PROJECT > app > components > ProductCard.tsx
- Code Content (ProductCard.tsx):**

```
app > components > ProductCard.tsx > ProductCard
1 import Image, { StaticImageData } from "next/image";
2 import share from "@/public/share-icon.svg";
3 import compare from "@/public/compare-icon.svg";
4 import like from "@/public/like-icon.svg";
5
6 interface ProductCardProps {
7   imageUrl: StaticImageData | string;
8   title: string;
9   description: string;
10  price: string;
11  oldPrice?: string;
12  discountPercentage?: string;
13  stockStatus: string;
14  onAddToCart: (item: { id: string; title: string; price: string }) => void;
15 }
16 const ProductCard = ([
17   imageUrl,
18   title,
19   description,
20   price,
21   oldPrice,
22   discountPercentage,
23   stockStatus,
24   onAddToCart,
25 ]: ProductCardProps) => {
26   const handleAddToCart = () => {
27     onAddToCart({
28       id: title,
29       title,
30       price,
31     });
32   };
33 }
```

- Status Bar:** Ln 24, Col 15 Spaces: 2 UTF-8

2. **SearchBar Component:** Enables instant filtering by product name.

```

    app > search > page.tsx > SearchPage > useEffect() callback > then() callback
    36   }
    37   export default function SearchPage() {
    38     const searchParams = useSearchParams();
    39     const query = searchParams.get("query");
    40     const [products, setProducts] = useState<Product[]>([]);
    41     const [loading, setLoading] = useState(true);
    42
    43     useEffect(() => {
    44       if (query) {
    45         fetchProducts(query)
    46           .then((data) => {
    47             setProducts(data);
    48             setLoading(false);
    49           })
    50           .catch(() => setLoading(false));
    51     }, [query]);
    52     const handleAddToCart = (item:{id:string; title:string; price:string})=>{
    53       console.log("Added to cart:", item);
    54     }
    55
    56     return (
    57       <div>
    58         <h1 className="font-bold text-2xl text-[#333333] ml-4 mb-4 mt-2">Search Results for "{query}"</h1>
    59         {loading ? (
    60           <p>Loading...</p>
    61         ) : (
    62           <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-2 justify-items-center">
    63             {products.length ? (
    64               products.map((product) => (
    65                 <ProductCard
    66                   key={product._id}
    67                   imageUrl={product.imageUrl}
    68               </ProductCard>
    69             ))
    70           ) : (
    71             <p>No results found for "{query}"</p>
    72           )}
    73         </div>
    74       </div>
    75     );
    76   }
  
```

Ln 47, Col 29 Spaces: 2 UTF-8 CRLF {} TypeScript JSX

3. Cart Sidebar: Displays cart items and supports real-time updates using local storage for state persistence.

```

    app > sidebar.tsx > CartSidebar
    7   import { useCart } from "./CartContext";
    8   import { useRouter } from "next/navigation";
    9   interface CartItem {
    10     title:string;
    11     price:string;
    12     quantity:number;
    13     imageUrl:string;
    14   }
    15   interface CartSidebarProps {
    16     isVisible:boolean;
    17     onClose: ()=>void;
    18   }
    19   export default function CartSidebar({isVisible, onClose}:CartSidebarProps) {
    20     const {cartItems, removeFromCart} = useCart();
    21     const router = useRouter();
    22
    23     if(!isVisible) return null;
    24
    25     const getNumericPrice = (price:string):number =>{
    26       const priceString = price.toString();
    27       return parseInt(priceString.replace(/[^0-9]/g, ""),10) || 0;
    28     }
    29     const subtotal = cartItems.reduce((total:number, item:CartItem)=> total + getNumericPrice(item.price))
    30
    31     const navigateTo = (path:string)=>{
    32       router.push(path);
    33       onClose();
    34     }
    35     return (
    36       <>
    37         <div className="fixed inset-0 bg-black bg-opacity-20 z-40" onClick={onClose}></div>
    38         <div className="cart-sidebar h-full w-96 flex-col p-4 bg-white flex justify-between z-50">
  
```

Ln 30, Col 2 Spaces: 2 UTF-8 CRLF {} TypeScript JS

Technical Documentation

Steps Taken to Build and Integrate Components:

1. Developed dynamic UI components for a seamless shopping experience.
2. Utilized **local storage** to maintain cart state and ensure data persistence across sessions.
3. Integrated APIs for fetching product details and enhancing component interactivity.

Challenges Faced and Solutions Implemented:

- **Challenge:** Implementing persistent cart state using local storage.
 - **Solution:** Designed logic to save, retrieve, and update cart data in local storage effectively, ensuring real-time updates without performance loss.

Best Practices Followed:

- Adopted a mobile-first approach for responsive design.
- Ensured accessibility compliance to create an inclusive user experience.
- Modularized code for maintainability and scalability.

Conclusion:

This technical implementation demonstrates the successful development of a dynamic frontend for an eCommerce platform. From product listings to persistent cart functionality, the project embodies a user-focused approach with seamless API integration and robust state management. This foundation provides a scalable base for future enhancements and further elevates the shopping experience.