

Hackathon MarketPlace Day 4: API Integration Report - [Furniro Furniture website]

By



Muhammad fuzail

Overview

Today, I worked on creating dynamic components for the **Furniro** website, focusing on product listings, including **chairs, stools, and sofas**, **single product view, add to cart, checkout and Related products**. I integrated real-time data fetching and added **pagination** for smooth navigation. I also built search bar for easy product search and sorting. The design is responsive, ensuring a great shopping experience on both desktop and mobile.

Objective:

- Build a dynamic frontend for an e-commerce marketplace.
- Implement pagination, search filters, and responsive design.

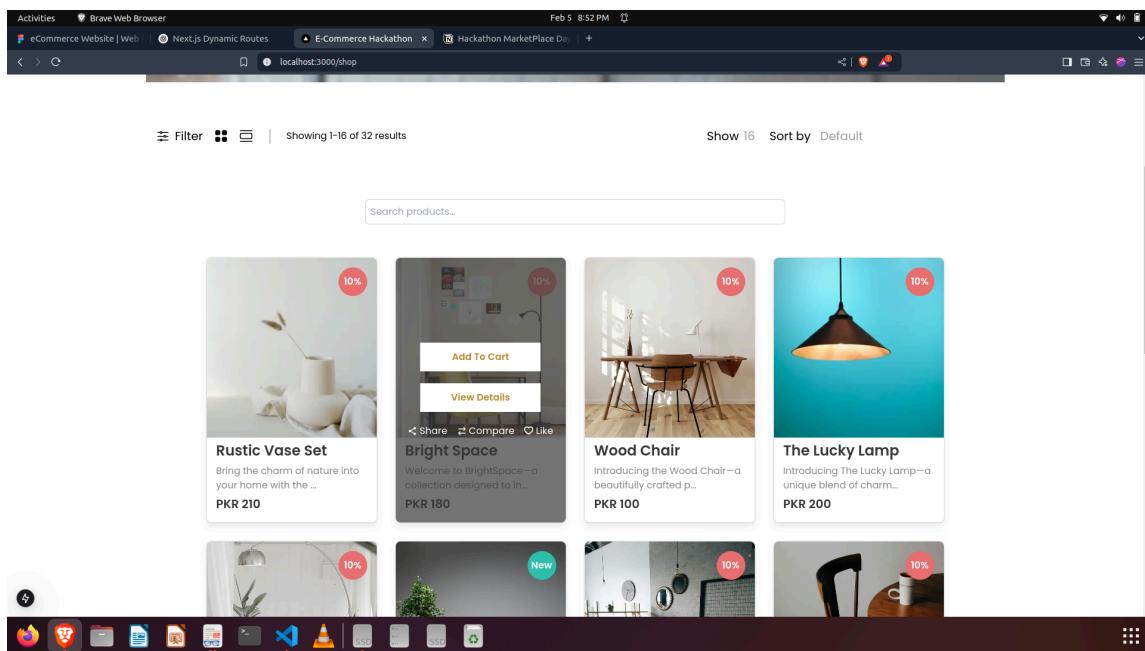
Features Delivered:

- Product listing with pagination.
- Product details page with all the features, description, **related products**.
- Related products on the basis of tags provided in each product.
- Cart functionality using local storage and global context API.
- Search filter to filter products based on query.
- Checkout page with the form built with validation through react-hook-form and functionality to show notification of successful order placement.
- Responsive and reusable components.
- Breadcrumbs based on dynamic routes.

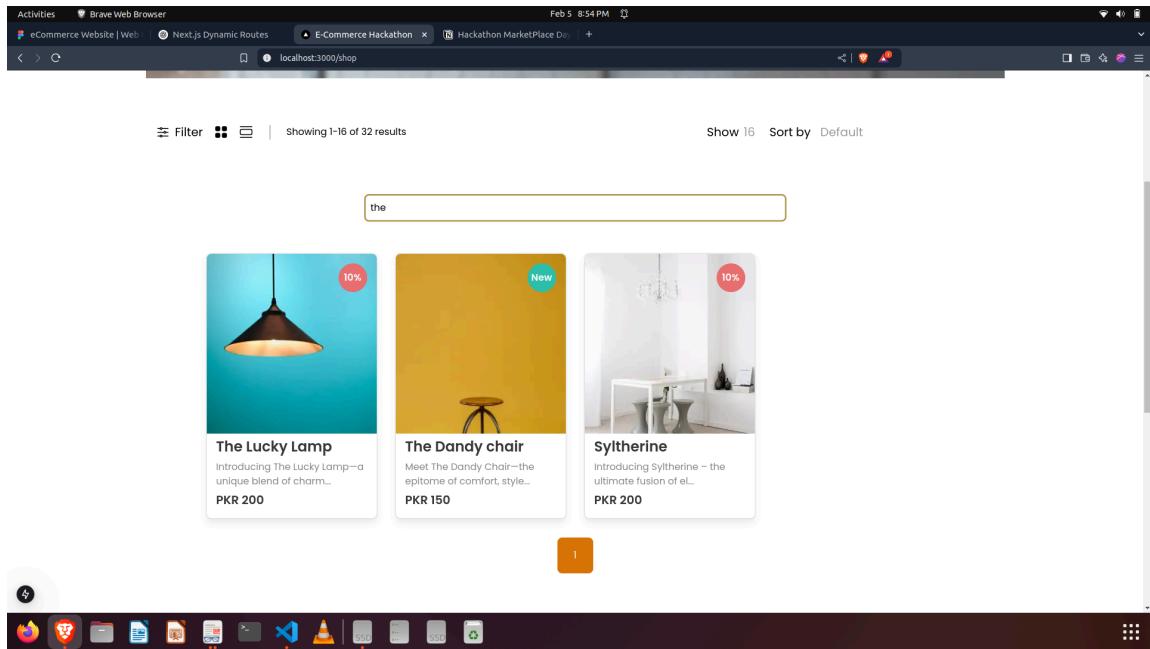
Features Explained and Showcased

Product Listing Page (Dynamic Data):

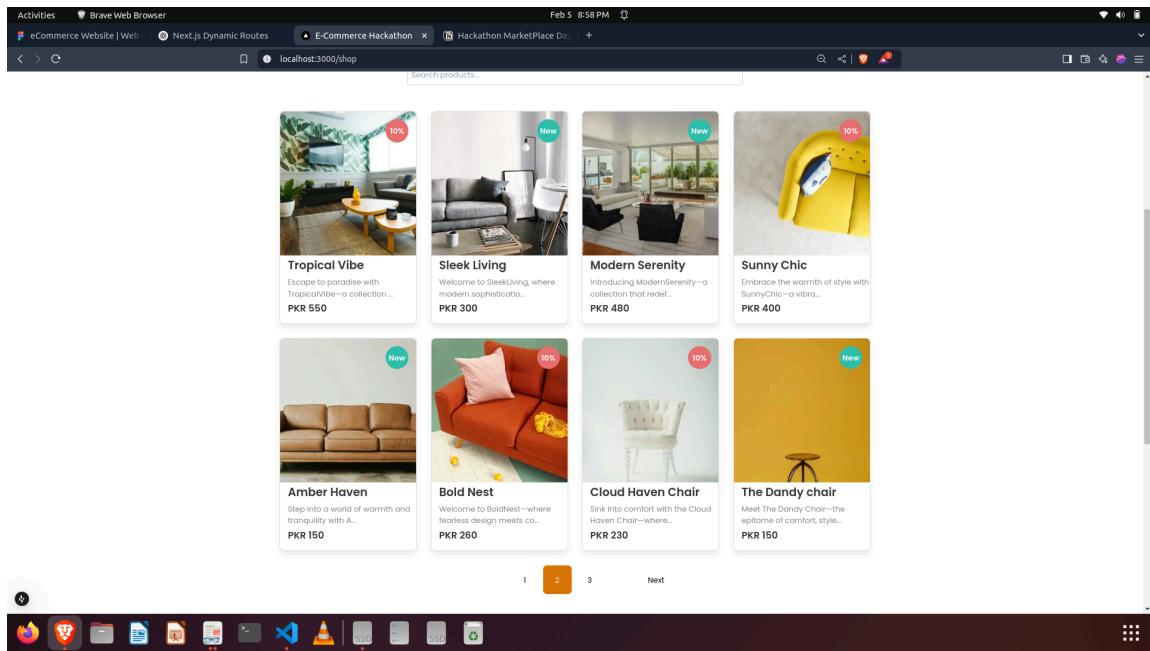
- The **product listing page** is designed to dynamically fetch and display products. The data is fetched in real-time, which ensures the content stays updated without the need for manual changes.
- **Product Details:** The page displays important information such as product titles, prices, and images fetched dynamically.
- **Add To Cart Option:** The user can click on Add To cart button to add product to cart.
- **View Details:** The user can click on View details button to see the product details page.
- **Search Functionality:** The user can search and filter the products being render by providing search query.
- **Responsive Design:** The page layout is fully responsive, ensuring optimal viewing across devices such as mobile phones, tablets, and desktops. Tailwind CSS utilities ensure that the content is rearranged according to the screen size, enhancing usability.



- **Search Functionality**



- **Pagination**



- **Product Details page based on title provided by dynamic route**

E-Commerce Hackathon

localhost:3000/shop/Sleek%20Living

Furniro

Home Shop Blog Contact

Home / Shop / Sleek Living

The screenshot shows a product page for 'Sleek Living' furniture. At the top, there's a navigation bar with links for Home, Shop, Blog, and Contact. Below the navigation is a user icon, a search bar, a heart icon, and a shopping cart icon with a notification badge. The main content area features a large image of a modern living room setup with a sofa, a coffee table, and a floor lamp. To the right of the image, the product title 'Sleek Living' is displayed in bold, followed by the price 'Rs. 300'. Below the price is a rating of 5 stars with 5 customer reviews. A brief description follows: 'Welcome to SleekLiving, where modern sophistication meets functional design for the ultimate living experience. Crafted for those who value style, convenience, and innovation, Sleekliving is a collection of premium home essentials that redefine the way you live. With an emphasis on minimalism and elegance, every product in the SleekLiving line is engineered to bring a refined touch to your home without compromising on performance or comfort.' Below the description are three small thumbnail images of the furniture pieces. At the bottom of the page, there are tabs for Description, Additional Information, and Reviews [5]. The 'Description' tab is currently active.

Activities Brave Web Browser

Feb 5 9:00 PM

eCommerce Website | Web

Next.js Dynamic Routes

Hackathon MarketPlace Day 4

localhost:3000/shop/Sleek%20Living

Description Additional Information Reviews [5]

This screenshot shows the same product page for 'Sleek Living' as the first one, but it appears to be a different version or a later update. The description text has been expanded and updated:

Welcome to SleekLiving, where modern sophistication meets functional design for the ultimate living experience. Crafted for those who value style, convenience, and innovation, Sleekliving is a collection of premium home essentials that redefine the way you live. With an emphasis on minimalism and elegance, every product in the SleekLiving line is engineered to bring a refined touch to your home without compromising on performance or comfort.

SleekLiving is all about transforming your space into a sanctuary of calm and sophistication. Featuring clean lines, contemporary aesthetics, and cutting-edge functionality, this collection offers versatile solutions for every room in your home. Whether you're upgrading your living room, kitchen, bedroom, or office, SleekLiving effortlessly blends into any decor, offering you the freedom to create a space that reflects your unique style.

Designed with the modern homeowner in mind, Sleekliving products are not just about looks—they are built for practicality and ease of use. Each item is thoughtfully crafted using high-quality materials that ensure durability and long-lasting performance. From smart furniture solutions to stylish accessories, SleekLiving brings together the best of contemporary design with everyday functionality.

Key Features:

- Modern, minimalist design that enhances any living space
- Versatile range of products for every room in your home
- High-quality materials for durability and long-lasting performance
- Innovative features that combine form and function
- Perfect for those who appreciate style and practicality
- Transform your home into a sleek, stylish haven with SleekLiving—where contemporary design meets everyday convenience.

At the bottom of the page, there are two horizontal sliders showing additional product images. The browser's activity bar at the top shows various open tabs and windows, including 'eCommerce Website | Web', 'Next.js Dynamic Routes', and 'Hackathon MarketPlace Day 4'.

The screenshot shows a product page for a grey sofa. At the top, there are three small thumbnail images of the sofa. Below them is a navigation bar with tabs: 'Description' (selected), 'Additional Information', and 'Reviews [5]'. Underneath the tabs, product details are listed: Price: Rs. 300, Discount: 0%, and Tags: sleek, modern, elegant, furniture, living. Two large, identical images of the sofa in a living room are displayed side-by-side. Below these images, a section titled 'Related Products' is visible, showing three items: 'Timber Craft' (PKR 320), 'Bold Nest' (PKR 260), and 'Marble Ease' (PKR 419). Each related product item has a red circular badge with '0%'.

- **Related Products**

The screenshot shows a 'Related Products' section with four items displayed in a grid:

- Timber Craft**: Introducing TimberCraft—a collection that celebrates...
PKR 320
- Bold Nest**: Welcome to BoldNest—where fearless design meets co...
PKR 260
- Marble Ease**: Introducing MarbleEase—a luxurious collection that...
PKR 419
- Retro Vibe**: Introducing RetroVibe—a perfect blend of vintage ...
PKR 340

Cart Functionality and Page:

Used Context API to make its context and defined functions to post data to local storage and to get it.

```

"use client";
import React, { createContext, useContext, useState, useEffect } from "react";

interface CartItem {
  id: string;
  name: string;
  price: number;
  quantity: number;
  image: string;
}

interface CartContextType {
  cart: CartItem[];
  addToCart: (item: CartItem) => void;
  removeFromCart: (id: string) => void;
  updateQuantity: (id: string, quantity: number) => void;
  clearCart: () => void;
}

const CartContext = createContext(undefined)

export const CartProvider = ({ children }: { children: React.ReactNode }) => {
  const [cart, setCart] = useState<CartItem[]>([]);

  // Load cart from local storage on initial render
  useEffect(() => {
    const savedCart = localStorage.getItem("cart");
    if (savedCart) {
      setCart(JSON.parse(savedCart));
    }
  }, []);

  // Save cart to local storage whenever it changes
  useEffect(() => {
    localStorage.setItem("cart", JSON.stringify(cart));
  }, [cart]);

  const addToCart = (item: CartItem) => {

```

```

const existingItem = cart.find((cartItem) => cartItem.id === item.id);
if (existingItem) {
  // If item already exists, update its quantity
  setCart(
    cart.map((cartItem) =>
      cartItem.id === item.id
        ? { ...cartItem, quantity: cartItem.quantity + 1 }
        : cartItem
    )
  );
} else {
  // If item doesn't exist, add it to the cart
  setCart([...cart, { ...item, quantity: 1 }]);
}
};

const removeFromCart = (id: string) => {
  setCart(cart.filter((item) => item.id !== id));
};

const updateQuantity = (id: string, quantity: number) => {
  setCart(
    cart.map((item) =>
      item.id === id ? { ...item, quantity: Math.max(1, quantity) } : item
    )
  );
};

const clearCart = () => {
  setCart([]);
};

return (
  <CartContext.Provider
    value={{ cart, addToCart, removeFromCart, updateQuantity, clearCart }}
  >
    {children}
  </CartContext.Provider>
)

```

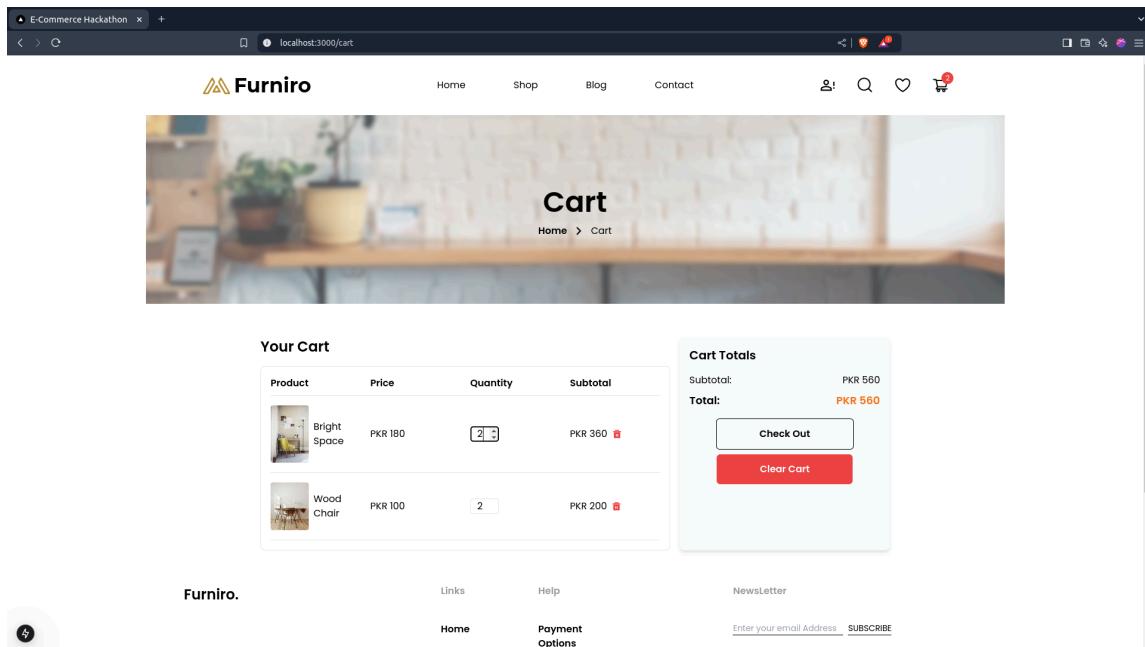
```

);
};

export const useCart = () => {
  const context = useContext(CartContext);
  if (!context) {
    throw new Error("useCart must be used within a CartProvider");
  }
  return context;
};

```

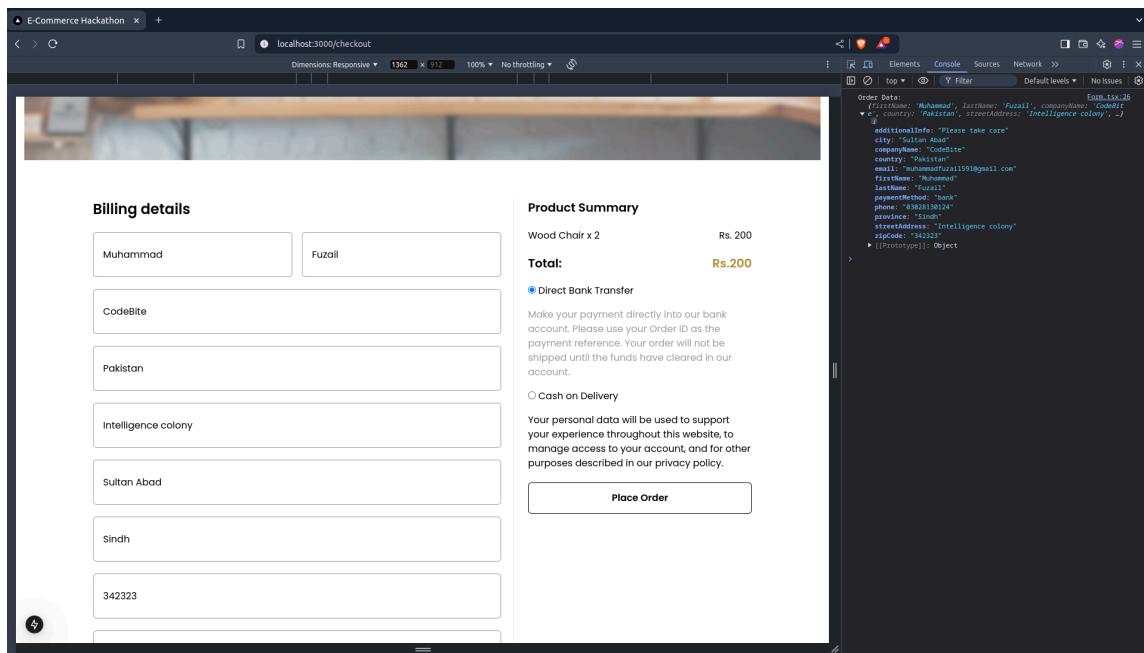
Fully Functional Cart component. Quantity, Deletion, Go to checkout all options are available.



- Demo video

[attachment:f0342607-f1be-4b77-8a3e-dd74e5e5bf5d:Screencast from 02-05-2025 091414 PM.webm](#)

Checkout Page and Order Placement Option:



What I Have Learned from this project:

- **Dynamic Data Handling:** We gained valuable skills in working with dynamic data fetching and routing, making our application flexible and responsive to data changes.
- **User Experience Enhancements:** By adding features like search, filters, pagination, and related products, we learned how to improve the usability and overall experience for customers.
- **Frontend Optimization:** We learned how to implement pagination, Image optimization and loading to optimize the performance of the frontend, making sure pages load faster and handle large datasets efficiently.
- **Responsive Web Design:** I have explored how to build responsive web pages using Tailwind CSS, ensuring that the website looks great on any device, which is crucial for modern web development.

Conclusion:

The development of the **Furniro** e-commerce website has been a comprehensive and rewarding project, focusing on creating a dynamic, user-friendly, and responsive frontend. Key features such as real-time data fetching, pagination, search functionality, and a fully functional cart system were successfully implemented. The integration of **Context API** for state management and **local storage** for persistent cart data ensured a seamless user

experience. Additionally, the use of **Tailwind CSS** allowed for a responsive design that adapts to various screen sizes, enhancing usability across devices.

The project also provided valuable learning opportunities, particularly in handling dynamic data, optimizing frontend performance, and improving user experience through features like related products and search filters. The checkout page, built with form validation and order placement notifications, further streamlined the purchasing process.

Overall, this project not only delivered a robust e-commerce platform but also deepened my understanding of modern web development practices, including dynamic routing, responsive design, and efficient state management. The skills and insights gained will undoubtedly be beneficial for future projects, ensuring the ability to build scalable and user-centric web applications.