

Day 4 - building dynamic frontend component for your Marketplace

Features to Implement

Search Menu

- . Implement a search bar to allow users to find products quickly.
- . Include suggestions or autocomplete functionality.



Slug Handling

- . Create SEO-friendly URLs for products and categories.

```
 1 import React from "react";
 2 import { client } from "@sanity/lib/client";
 3 import { groq } from "next-sanity";
 4 import Image from "next/image";
 5
 6 // Define the structure of the food data
 7 interface Food {
 8   _id: string;
 9   name: string;
10   _type: string;
11   image?: string; // Resolved image URL
12   price: number;
13   description?: string; // Added description field
14   tags?: string[]; // Added tags field
15 }
16
17 // Define the params structure for the dynamic route
18 interface FoodPageProps {
19   params: { slug: string };
20 }
21
22 // Fetch a specific food item based on the slug
23 async function getFood(slug: string): Promise<Food | null> {
24   return client.fetch(
25     groq `!_type == "food" & slug.current == ${slug}[0] {
26       _id,
27       name,
28       _type,
29       "image": image.asset->url, // Resolved URL
30       price,
31       description,
32       tags // Fetch tags
33     },
34     { slug }
35   );
36 }
```

BLOGS COMPONENTS

Design a section to display blog posts related to marketplace trends, product guides, and user stories.

Implement features like blog categorization, tags, and search functionality within the blog section.

Enable dynamic content loading for better performance and user engagement.

Development Ste

OUR LATEST BLOGS



Veggie Chicken Burger

2/2/2025

The Flavor Bomb: A Burger That Exploses with Deliciousness masterpiece, a symphony of flavors and textures

[Learn more](#)



Spanish sweet sauce pasta

2/1/2025

Spanish Pasta is a celebration of sweet and savory flavors, a vibrant dish inspired by the sunny spirit of Spain. Imagine perfectly...

[Learn more](#)



BLOGS DYNAMIC PAGE

Published on 2/1/2025



A burger is a culinary masterpiece, a symphony of flavors and textures nestled between two halves of a bun. At its heart lies a patty, typically crafted from seasoned ground meat – most often beef, but also featuring options like chicken, turkey, or even plant-based alternatives for vegetarians and vegans. This patty can be grilled to juicy perfection, pan-fried for a crispy sear, or even smashed thin for a caramelized crust. The Flavor Bomb so special? It's not just one thing; it's a symphony of carefully curated ingredients working in perfect harmony.

The Patty: We start with a premium blend of [mention specific type of beef, e.g., Angus, Wagyu] ground fresh daily. The patty isn't just cooked; it's seared to lock in the juices, creating a caramelized crust that crackles with every bite.

The Cheese: Forget those flimsy slices of processed cheese. The Flavor Bomb boasts a generous helping of [mention specific cheese, e.g., aged cheddar, smoked gouda], melted to perfection and oozing with creamy goodness.

The Toppings: This is where the magic happens. We're not talking about your average lettuce and tomato. Imagine [mention unique toppings, e.g., caramelized onions, crispy bacon, avocado crema, spicy aioli]. Each ingredient is

BLOGS CODE

The image shows a code editor interface with a sidebar on the left displaying a file tree and the main area showing the content of a selected file.

File Tree:

- src
- app
 - admin
 - page.tsx
- blog\slug
 - page.tsx
- blogs
 - page.tsx
- cart
 - page.tsx
- checkout
 - page.tsx
- chef
 - page.tsx
- components
 - about.tsx
 - abouthome.tsx
 - aboutus.tsx
 - addtocartbutton.tsx
 - blog.tsx
 - blogcard.tsx
 - bloghome.tsx
 - blogscard2.tsx
 - blogtitle.tsx

Selected File: src/app/blog\slug/page.tsx

```
1 import Image from "next/image";
2 import { notFound } from "next/navigation";
3 import { PortableText } from "@portabletext/react";
4 import { client } from "@/sanity/lib/client";
5
6 // ✅ Static Params Generate Karne Ka Function
7 export async function generateStaticParams() {
8   const query = `*[_type == "blogPost"]{ "slug": slug.current }`;
9   const slugs = await client.fetch(query);
10
11   return slugs.map((post: { slug: string }) => ({ slug: post.slug }));
12 }
13
14 // ✅ TypeScript Interface Define Kiya
15 interface BlogPostPageProps {
16   params: { slug: string };
17 }
18
19 // ✅ BlogPost Fetch Karne Ka Function
20 async function getBlogPost(slug: string) {
21   const query = `*[_type == "blogPost" && slug.current == $slug][0]{
22     _id,
23     title,
24     publishedAt,
25     body,
26     "image": image.asset->url
27   }`;
28 }
```

ADD TO CART COMPONENTS

Design a dynamic cart icon to show the number of items in the cart.

Allow users to update quantities or remove items directly from the cart.

Save the cart state in local storage or backend for persistence.



Featured

iltilian pizza

 3 Rating (1 Reviews)

iltilian pizza sauce is a white or brown sauce prepared using mushrooms as its primary ingredient. It can be prepared in different styles using various ingredients, and is used to top a variety of foods.

\$799.00

- 1 +

Add to Cart

Category: Arabian pizza

Share: [f](#) [t](#) [i](#) [p](#)

ADD TO CART DYNAMIC CODE

```
  ⚡ page.tsx      9   name: string
  ✓ blog\[slug]  10  price: number
  ⚡ page.tsx    11  quantity?: number
  ✓ blogs        12 }
  ✓ cart          13
  ✓ checkout     14  export default function AddToCartButton({ product }: { product: Product }) {
  ✓ chef          15    const addItem = useCartStore((state) => state.addItem)
  ✓ components   16
  ⚡ about.tsx    17    const handleAddToCart = () => {
  ⚡ abouthome.tsx 18      addItem({
  ⚡ aboutus.tsx   19        _id: product._id,
  ✓ addtocartbutton.tsx 20        name: product.name,
  ✓ blog.tsx       21        price: product.price,
  ✓ blogcard.tsx  22        quantity: product.quantity || 1,
  ✓ bloghome.tsx  23      })
  ✓ blogcard2.tsx 24      toast.success(`Added ${product.name} to cart!`)
  ✓ blogtitle.tsx 25    }
  ✓
  ✓ return (
  ✓   <button
  ✓     onClick={handleAddToCart}
  ✓     className="w-full py-3 px-4 bg-green-800 text-white rounded-lg hover"
  ✓   >
  ✓     Add to Cart
  ✓   </button>
```

Search Menu

Design the search input field and dropdown suggestion list.

Filter By Menu

Search menu...

Chicken Fry	26
Burger Food	46
Pizza	16
Fresh Fruits	36
Vegetables	16
Shawarma	10
Chicken Rice	8
Beef Burger	6
Vegetables Rice	2
Fries	4
Chicken Tikka	9

```
import { useState } from 'react';
import { FaFacebook, FaTwitter, FaInstagram, FaYoutube } from 'react-icons/fa';

function BlogCard2() {
  const [searchQuery, setSearchQuery] = useState('');

  // Menu items
  const menuItems = [
    { name: 'Chicken Fry', count: 26 },
    { name: 'Burger Food', count: 46 },
    { name: 'Pizza', count: 16 },
    { name: 'Fresh Fruits', count: 36 },
    { name: 'Vegetables', count: 16 },
    { name: 'Shawarma', count: 10 },
    { name: 'Chicken Rice', count: 8 },
    { name: 'Beef Burger', count: 6 },
    { name: 'Vegetables Rice', count: 2 },
    { name: 'Fries', count: 4 },
    { name: 'Chicken Tikka', count: 9 },
  ];

  // Filter menu items based on the search query
  const filteredMenuItems = menuItems.filter(item =>
    item.name.toLowerCase().includes(searchQuery.toLowerCase())
  );
}
```

CATEGORY SEARCH COMPONENT

The screenshot shows a mobile application interface for a food delivery service. At the top, there's a header with the text "Our Shop" and "Home > Checkcard". Below the header, on the left, is a sidebar titled "Categories" with a list of items: All, Pizza, Burgers, Desserts, Drinks, and Snacks. The "All" item is currently selected, highlighted with a white background. The main content area displays four food items in a grid:

- Pizza**: Delicious cheesy pizza with toppings of your choice. Price: \$12. Add to Cart button.
- Burger**: Juicy beef burger with crispy lettuce and tomato. Price: \$8. Add to Cart button.
- Pasta**: Classic Italian pasta with rich marinara sauce. Price: \$10. Add to Cart button.

Below the first row, there are two more items partially visible:

- A dish of flatbread or pizza.
- A bowl of salad with bread on the side.

CHECKOUT PAGE



CHECKOUT CODE

```
page.tsx          32   name: "",  
orders           33   email: "",  
[orderId]        34   phone: "",  
page.tsx         35   address: "",  
});  
const [showOrderSlip, setShowOrderSlip] = useState(false);  
const [orderDetails, setOrderDetails] = useState<OrderDetails | null>(null);  
  
const handleInputChange = (e: React.ChangeEvent<HTMLInputElement>) => {  
  const { name, value } = e.target;  
  setCustomerDetails((prev) => ({ ...prev, [name]: value }));  
};  
  
const handleSubmit = (e: React.FormEvent) => {  
  e.preventDefault();  
  const order: OrderDetails = placeOrder(customerDetails); // Ensure placeOrder  
  setOrderDetails(order);  
  setShowOrderSlip(true);  
};  
  
const handleFinishOrder = () => {  
  // Logic to handle order completion  
};
```

CHECKOUT PAGE



CHECKOUT CODE

```
page.tsx          32   name: "",  
orders           33   email: "",  
[orderId]        34   phone: "",  
page.tsx         35   address: "",  
});  
const [showOrderSlip, setShowOrderSlip] = useState(false);  
const [orderDetails, setOrderDetails] = useState<OrderDetails | null>(null);  
  
const handleInputChange = (e: React.ChangeEvent<HTMLInputElement>) => {  
  const { name, value } = e.target;  
  setCustomerDetails((prev) => ({ ...prev, [name]: value }));  
};  
  
const handleSubmit = (e: React.FormEvent) => {  
  e.preventDefault();  
  const order: OrderDetails = placeOrder(customerDetails); // Ensure placeOrder  
  setOrderDetails(order);  
  setShowOrderSlip(true);  
};  
  
const handleFinishOrder = () => {  
  // Logic to handle order completion  
};
```