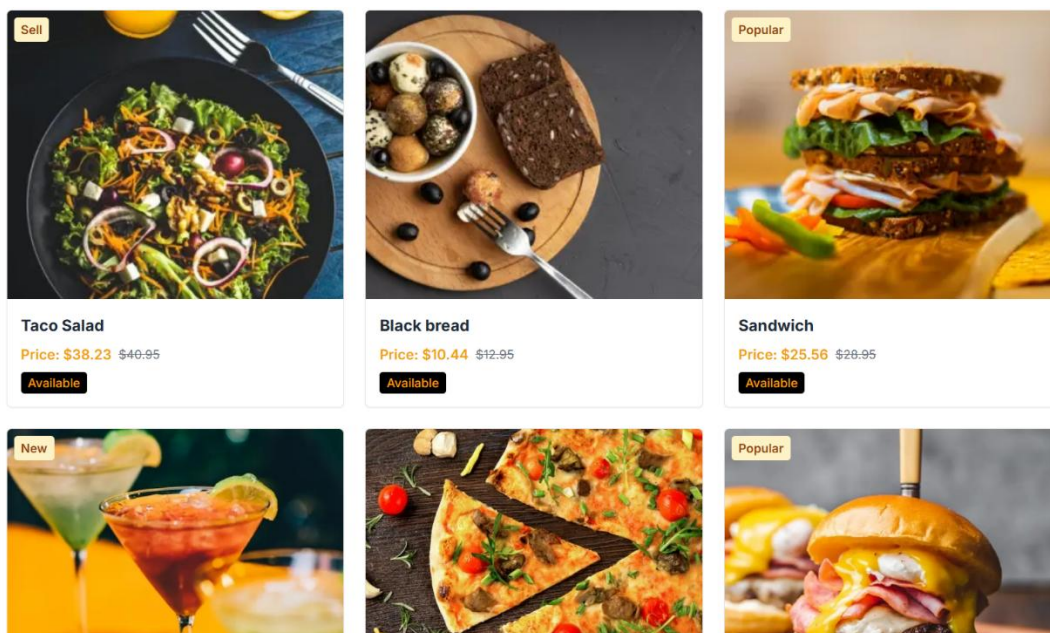


# Day 4 - Dynamic Frontend Components – Food tuck Marketplace

## 1.Product Listing Page:

A grid layout showcasing product data, fetch from sanity include

- Product name.
- Product price.
- Product image.
- Product availability.



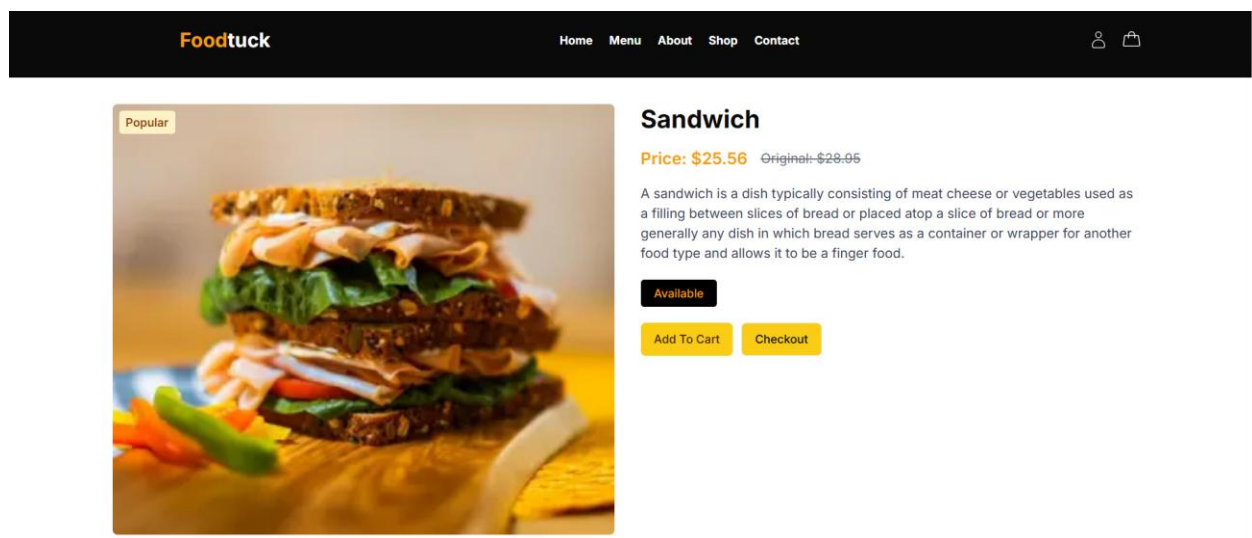
```

src > components > ourshop.tsx > Product
1  import { client } from '@sanity/lib/client';
2  import { urlFor } from '@sanity/lib/image';
3  import Image from 'next/image';
4  import Link from 'next/link';
5
6  type Product = {
7    _id: string;
8    image: string;
9    name: string;
10   price: number;
11   originalPrice: number;
12   available: boolean;
13   tags?: string; // Marked as optional
14 };
15
16 export default async function Ourshop() {
17   try {
18     const products: Product[] = await client.fetch(`
19       *[_type=="food"]{
20         name,
21         image,
22         price,
23         originalPrice,
24         _id,
25         available,
26         tags
27       }
28     `);
29
30     return (
31       <div className="container mx-auto px-4 sm:px-8 py-8">
32         <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-6">
33           {products.map((product) => (
34             <Link key={product._id} href={`/${product._id}`}>
35               <div className="border rounded-md overflow-hidden shadow-sm hover:shadow-md transition cursor-pointer">
36                 <div className="relative w-full aspect-[4/3]">
37                   {/* Conditionally render tags if available */}

```

## 2. Dynamic Product Detail Page Routing:

Used Next.js dynamic routes `/products/[id]/page.tsx` to implement individual product pages.



▼ documentation ●

> node\_modules

> public ●

▼ src ●

▼ app ●

> about ●

> api\shipengi... ●

> contact ●

> login ●

> menu ●

▼ products\[id] ●

⚙️ page.tsx U

▼ shipmentpage ●

⚙️ page.tsx U

> shop ●

> signup ●

> stripe ●

> studio ●

▼ tracking ●

⚙️ page.tsx U

★ favicon.ico

# globals.css M

⚙️ layout.tsx M

⚙️ page.tsx M

> components ●

> lib ●

> sanity ●

> scripts ●

⚙️ .env

\$ .env.local

> OUTLINE

> TIMELINE

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

```
interface Params {
  params: {
    id: string;
  };
}

interface FullProduct {
  _id: string;
  image: any;
  price: number;
  originalPrice: number;
  name: string;
  description: string;
  available: boolean;
  tags?: string;
  price_id: string;
}

export const dynamic = "force-dynamic";

export default async function ProductPage({ params }: Params) {
  const { id } = params;

  try {
    // Fetch product details by `_id`
    const product: FullProduct = await client.fetch(
      `*[_type=="food" && _id == $id][0]{
        name,
        image,
        price,
        origipalPrice,
        _id,
        available,
        description,
        tags,
        price_id
      }`
    );
  } catch {
    // Handle error
  }
}
```

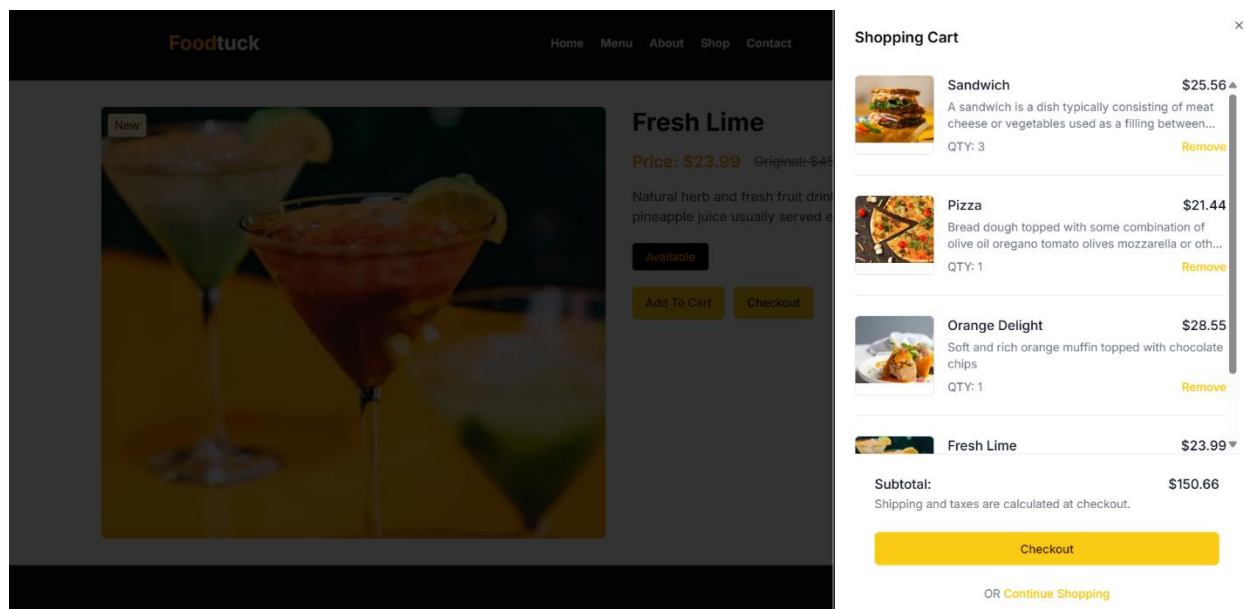
```

src > app > products > [id] > page.tsx > ProductPage
38 export default async function ProductPage({ params }: Params) {
42   // Fetch product details by _id
43   const product: FullProduct = await client.fetch(
44     `[_type=="food" && _id == ${id}][0]{
45     name,
46     image,
47     price,
48     originalPrice,
49     _id,
50     available,
51     description,
52     tags,
53     price_id
54   }`,
55   { id }
56 );
57
58 if (!product) {
59   return (
60     <div className="text-center mt-20 text-gray-500">
61       Product not found.
62     </div>
63   );
64 }
65
66 return (
67   <div className="container mx-auto px-4 py-8">
68     <div className="flex flex-col lg:flex-row gap-8 items-center lg:items-start">
69       <div className="w-full lg:w-1/2 relative">
70         /* Tags displayed on the left side of the image */
71         {product.tags && (
72           <div className="absolute top-2 left-2 bg-amber-100 text-amber-800 text-sm font-semibold px-2 py-1 rounded">
73             {product.tags}
74           </div>
75         )}
76       <div className="aspect-w-1 aspect-h-1 w-full max-w-sm mx-auto lg:max-w-full">
77         <Image

```

### 3.Add to Cart Functionality:

Customer can add product to the cart, view items, and remove items dynamically by the help of useshoppingcart library.



HACKTON3

src

app

tracking

page.tsx

favicon.ico

globals.css

layout.tsx

page.tsx

components

ui

aboutchoose....

aboutmenu.tsx

aboutus.tsx

addtocart.tsx

checkout.tsx

chooseus.tsx

client.tsx

contactform.tsx

dessert.tsx

drink.tsx

footer.tsx

homeabout.tsx

listitem.tsx

maincourse.tsx

menu.tsx

navbar.tsx

ourshop.tsx

partner.tsx

provider.tsx

shoppingcart...

OUTLINE

TIMELINE

src > components > addtocart.tsx > ...

1"use client";

2

3import { Button } from "@components/ui/button";

4import { useShoppingCart } from "use-shopping-cart";

5import { urlFor } from "@sanity/lib/image";

6

7import { useEffect } from "react";

8

9export interface ProductCart {

10 name: string;

11 description: string;

12 price: number;

13 currency: string;

14

15 image: any;

16 price\_id: string;

17 }

18

19export default function AddToCart ({

20 currency,

21 description,

22 image,

23 name,

24 price,

25 price\_id,

26 }: ProductCart) {

27 const { addItem, handleCartClick, cartDetails, clearCart } = useShoppingCart();

28

29 const product = {

30 name: name,

31 description: description,

32 price: price,

33 currency: currency,

34 image: urlFor(image).url(),


35 price\_id: price\_id,

36 };

37


## 4.Checkout Functionality:


Stripe test account and test API key is used for payment gateway.


←  TEST MODE


Pay


\$150.66

 Sandwiches  
Qty 3  
\$25.56 each

 Pizza  
Qty 1  
\$21.44

 Orange Delight  
Qty 1  
\$28.55

 Pina colada  
Qty 1  
\$23.99


Pay with  link


Or pay with card

Email

test@gmail.com

Card information


4242 4242 4242 4242 

04 / 25424 

Cardholder name

nadeem

Billing address

Pakistan 


house no 33 main street

house n0 33 street kalapul

Suburb

karachi75640

☐ Securely save my information for 1-click checkout  
Pay faster on this site and everywhere Link is accepted.

Pay 

HACKT... L+ L+ ○ E

▼ src ●

▼ app ●

▼ tracking ●

⚙️ page.tsx U

★ favicon.ico

# globals.css M

⚙️ layout.tsx M

⚙️ page.tsx M

▼ components ●

> ui ●

⚙️ aboutchoose.... U

⚙️ aboutmenu.tsx U

⚙️ aboutus.tsx U

⚙️ addtocart.tsx U

⚙️ checkout.tsx U

⚙️ chooseus.tsx U

⚙️ client.tsx U

⚙️ contactform.tsx U

⚙️ dessert.tsx U

⚙️ drink.tsx U

⚙️ footer.tsx U

⚙️ homeabout.tsx U

⚙️ listitem.tsx U

⚙️ maincourse.tsx U

⚙️ menu.tsx U

⚙️ navbar.tsx U

⚙️ ourshop.tsx U

⚙️ partner.tsx U

⚙️ provider.tsx U

⚙️ shoppingcart... U

OUTLINE

src > components > checkout.tsx > Checkout

8 export default function Checkout({

9

10 currency,

11 description,

12 image,

13 name,

14 price,

15 price\_id,

16 }): ProductCart) {

17

18 const { checkoutSingleItem } = useShoppingCart();

19 function buyNow(priceId: string) {

20 | checkoutSingleItem(priceId);

21 }

22

23 const product = {

24 | name: name,

25 | description: description,

26 | price: price,

27 | currency: currency,

28 | image: urlFor(image).url(),

29 | price\_id: price\_id,

30 };

31 return (

32 | <Button className="bg-yellow-400"

33 | variant="outline"

34 | onClick={() => {

35 | | buyNow(product.price\_id);

36 | }})

37 | >

38 | Checkout

39 | </Button>

40 | )

41 }

42

43

## 5. ShipEngine Implementation in Next.JS Project:

Shipengine provides multi-carrier shipping functionality used API for rating, tracking and Creating shipping labels.

Foodtuck

HomeMenuAboutShopContact

### Shipping Rates Calculator

Ship To Address

John Doe

+1 555-678-1234

1600 Pennsylvania Avenue NW

Address Line 2

Washington

DC

20500

US

Get Shipping Rates

#### Available Shipping Rates

Carrier: UPS

☐ Service: UPS Next Day Air®

Cost: 379.39 usd

Carrier: UPS

☐ Service: UPS Next Day Air® Early

Cost: 469.39 usd

Carrier: UPS

☐ Service: UPS Next Day Air Saver®

Cost: 342.61 usd

Carrier: UPS

☐ Service: UPS 2nd Day Air®

Cost: 165.3 usd

Carrier: UPS

☐ Service: UPS 2nd Day Air AM®

Cost: 190.09 usd

Carrier: UPS

☐ Service: UPS 3 Day Select®

Cost: 141.79 usd

Carrier: UPS

☒ Service: UPS® Ground

Cost: 49.92 usd

Create Label



Download Label

Tracking thinks (We are using ShipEngine test api key so order will not trace)

tracking number: 1ZXXXXXXXXXXXXXXXXX  
labelId: se-18352403  
carrierCode: ups

Track Order



**Foodtuck**[Home](#)[Menu](#)[About](#)[Shop](#)[Contact](#)

### Track Your Shipment

Enter Label ID or Tracking Number:

Track Shipment

#### Tracking Details

Tracking Number: 1ZXXXXXXXXXXXXXXX

Status: Unknown

Carrier Status: Invalid tracking number

Estimated Delivery: N/A

Actual Delivery: N/A

## BEST PRACETICES FOLLOWED

- Tested the application thoroughly to ensure responsiveness and functionality across devices.
- Modular and reusable component design.
- Clean and maintainable code with proper comments.
- Efficient data fetching with error handling.

