DAY-4 DYNAMIC FRONTEND COMPONENTS – [FOOD-HUB]

INTRODUCTION

Welcome to Gul-Kashi Food Hub

The **ultimate destination** for high-quality, delicious, and freshly prepared food. We believe in serving not just meals, but **memorable experiences** filled with taste and satisfaction.

Why Choose Food Hub?

✅ **Fresh Ingredients** – We use only the freshest and finest ingredients to prepare our Dishes, ensuring every Bite is perfect.

✅ **Fast Delivery** – Enjoy quick and reliable delivery to your doorstep, making your dining experience Hassle-free.

✅ **Customer Satisfaction** – With a focus on customer satisfaction, we strive to make every experience Delightful and fulfilling.

What Makes Us Special?

✨ **Convenient Payment Options** – Pay easily through **Easy paisa and Jazz Cash**.

✨ **Perfect for Everyone** – Specially crafted meals for **students, offices, and families**.

✨ **Customizable Meals** – Choose according to your specific taste and preferences.

✨ **24/7 Customer Support** – Always here to assist you anytime.

✨ **Wide Variety** – From **traditional flavors to international cuisines**, we have it all!

Experience the Best in Food!

At **Food Hub**, we don't just serve food; **we serve happiness**. Join us and explore a world of flavors, convenience, and excellence.

I have built my project, Gul-Kashi Food Hub, as a **quick commerce platform**, keeping in mind the needs of the modern era. It provides all the conveniences that allow people to **order food easily, anytime, even in the middle of the night, from the comfort of their homes, without any hassle**.

In addition, whether it's a **picnic spot, office meetings, students, or families**, our platform ensures the **fastest delivery with fresh ingredients**.

Keeping all these factors in mind, I have incorporated the following pages into my project to **connect with as many users/customers as possible and provide them with the best experience**:

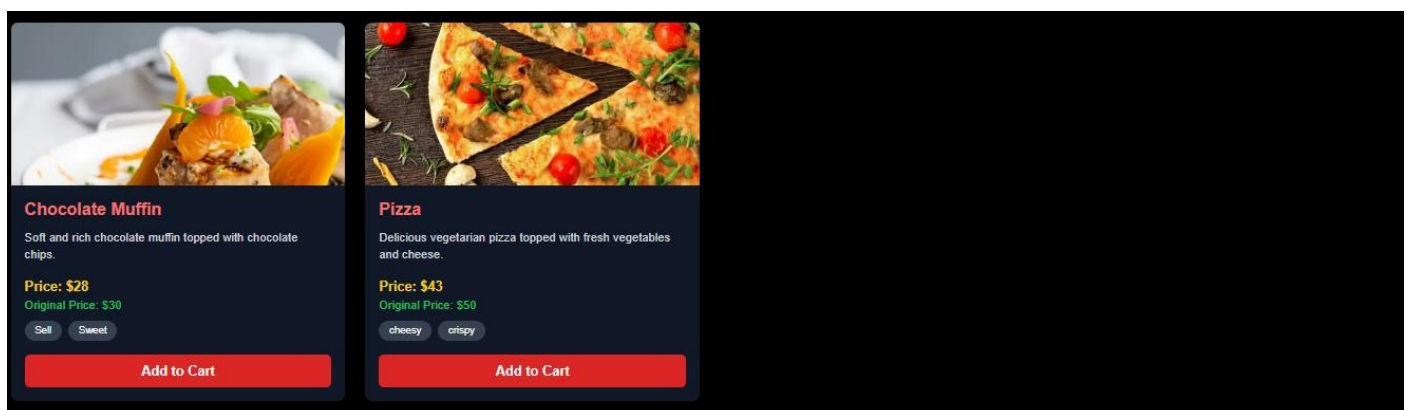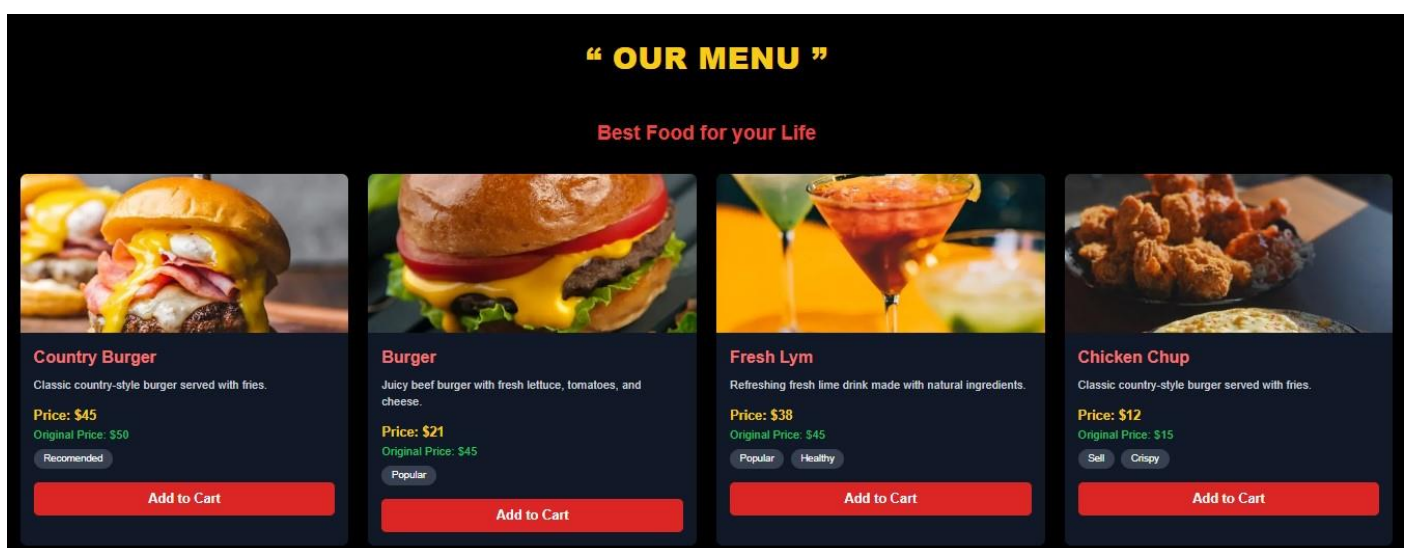**Home, About, Chefs, Menu, Shop, FAQ, and Contact Details.**

Because….!

**We don't just serve food; we serve happiness.** ☺🍽️🚀

1-API INTEGRATION & DATA MIGRATION:-

As per below screen shots, I have import the data from **Template-9 API** of Sir **Mubashir Imtiaz** through installing the Sanity studio, Schema Type and Script Importdata.mjs code. Below are mine fetched data from Sanity screen shot. On-which I have implemented the **Dynamic routing**.

" OUR MENU "

Best Food for your Life

Country Burger
Classic country-style burger served with fries.
Price: $45
Original Price: $50
Recomended
Add to Cart

Burger
Juicy beef burger with fresh lettuce, tomatoes, and cheese.
Price: $21
Original Price: $45
Popular
Add to Cart

Fresh Lym
Refreshing fresh lime drink made with natural ingredients.
Price: $38
Original Price: $45
Popular   Healthy
Add to Cart

Chicken Chup
Classic country-style burger served with fries.
Price: $12
Original Price: $15
Sell   Crispy
Add to Cart

Chocolate Muffin
Soft and rich chocolate muffin topped with chocolate chips.
Price: $28
Original Price: $30
Sell   Sweet
Add to Cart

Pizza
Delicious vegetarian pizza topped with fresh vegetables and cheese.
Price: $43
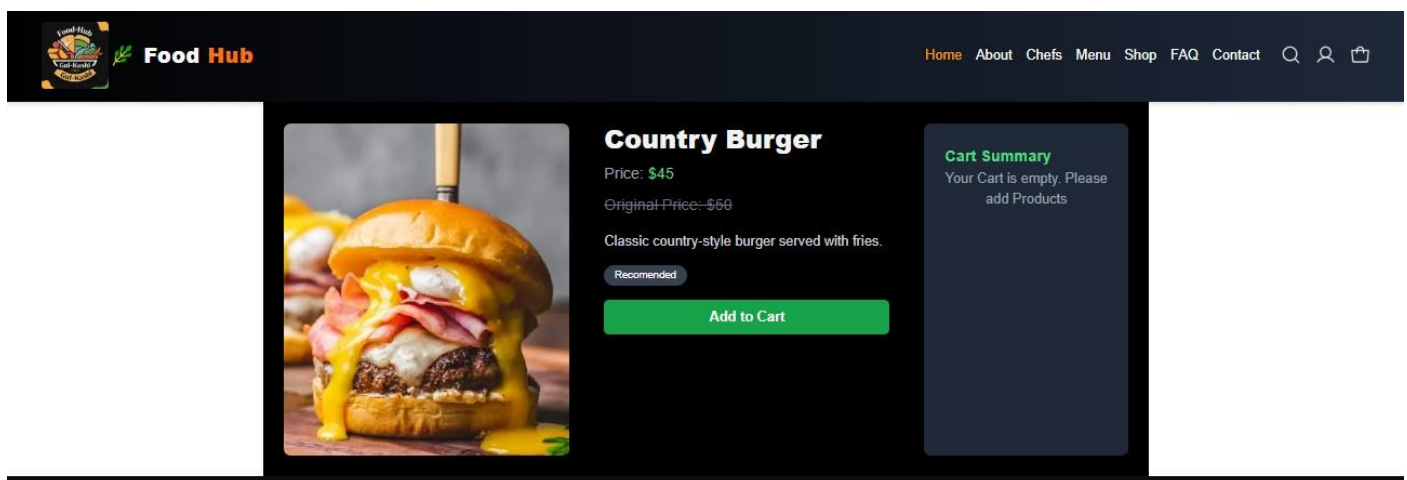Original Price: $50
cheesy   crispy
Add to Cart

2-Dynamic Routing of Products:-

On-Click of every product, user will find the separate page of this product or he can also open the chosen product on the new page dynamically without any delay or loading.
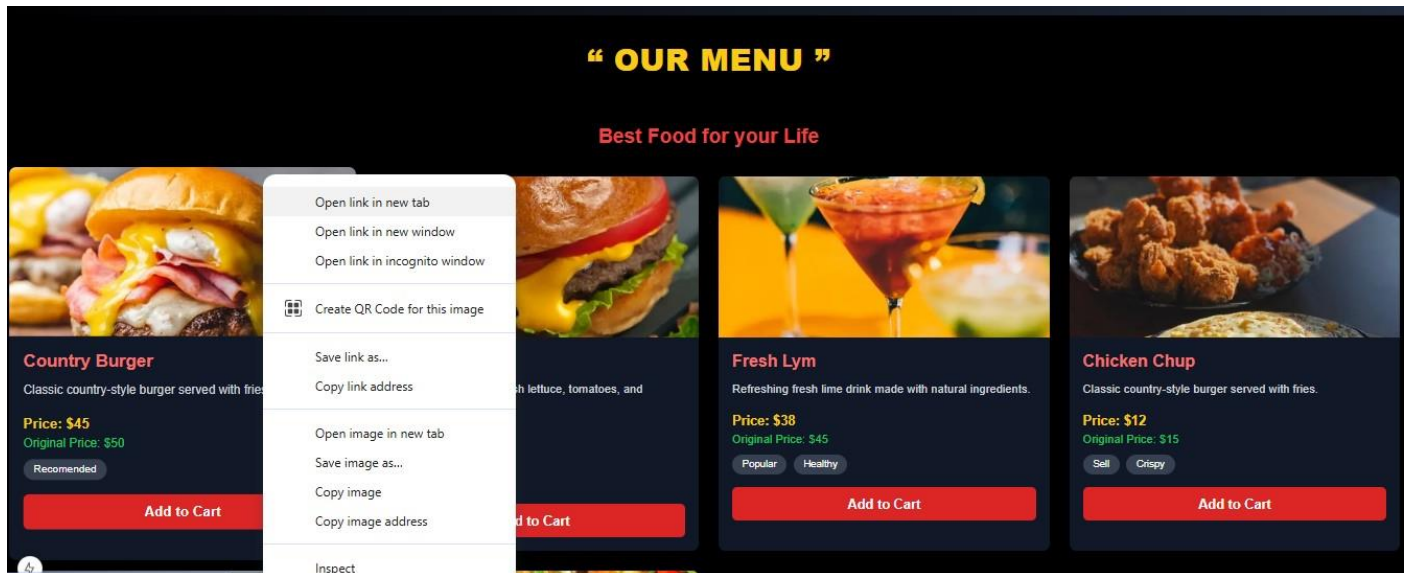
Product 1

Food Hub

Home About Chefs Menu Shop FAQ Contact

Fresh Lym

Price: $38

Original Price: $45

Refreshing fresh lime drink made with natural ingredients.

Popular   Healthy

Add to Cart

Cart Summary
Your Cart is empty. Please add Products

Food Hub

Home About Chefs Menu Shop FAQ Contact

Chicken Chup

Price: $12

Original Price: $15

Classic country-style burger served with fries.

Sell   Crispy

Add to Cart

Cart Summary
Your Cart is empty. Please add Products

Food Hub

Home About Chefs Menu Shop FAQ Contact

Chocolate Muffin

Price: $28

Original Price: $30

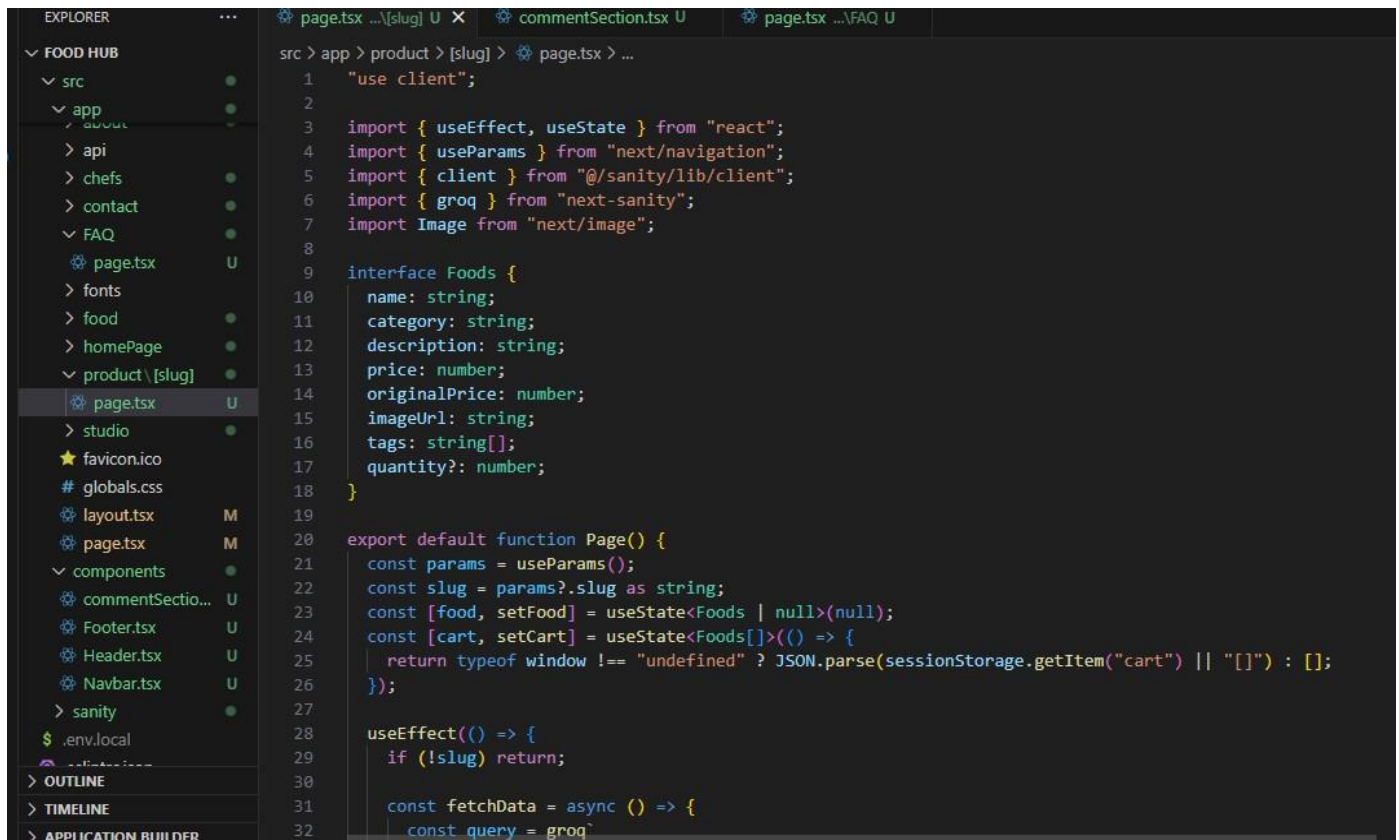Soft and rich chocolate muffin topped with chocolate chips.

Sell   Sweet

Add to Cart

Cart Summary
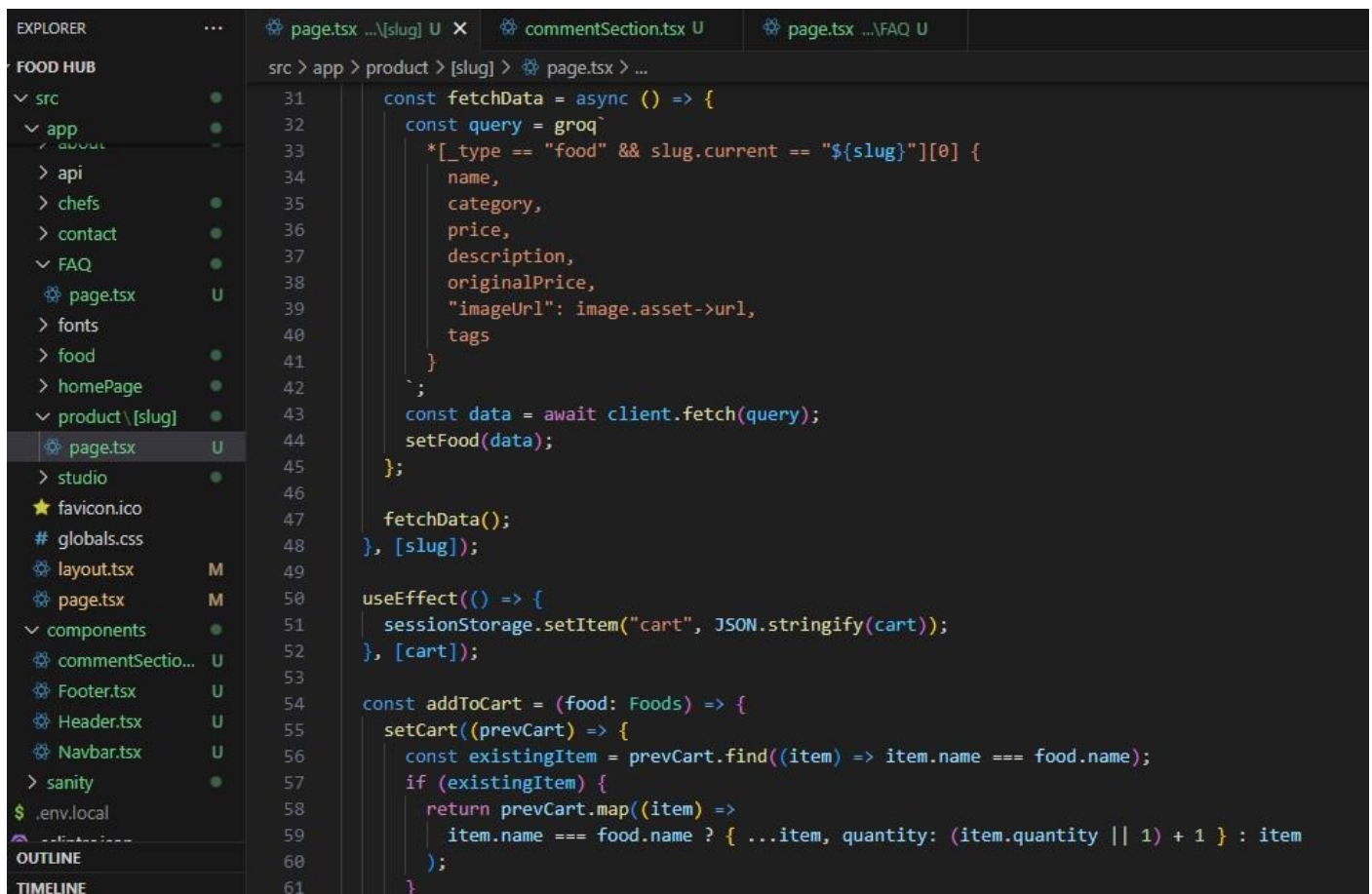Your Cart is empty. Please add Products

Dynamic Routing Code:-

I have made the every product page dynamically through the [Slug] and GROQ Query, which code I am sharing.

```tsx
"use client";

import { useEffect, useState } from "react";
import { useParams } from "next/navigation";
import { client } from "@/sanity/lib/client";
import { groq } from "next-sanity";
import Image from "next/image";

interface Foods {
  name: string;
  category: string;
  description: string;
  price: number;
  originalPrice: number;
  imageUrl: string;
  tags: string[];
  quantity?: number;
}

export default function Page() {
  const params = useParams();
  const slug = params?.slug as string;
  const [food, setFood] = useState<Foods | null>(null);
  const [cart, setCart] = useState<Foods[]>(() => {
    return typeof window !== "undefined" ? JSON.parse(sessionStorage.getItem("cart") || "[]") : [];
  });

  useEffect(() => {
    if (!slug) return;

    const fetchData = async () => {
      const query = groq`
```

```tsx
    const fetchData = async () => {
      const query = groq`
        *[_type == "food" && slug.current == "${slug}"][0] {
          name,
          category,
          price,
          description,
          originalPrice,
          "imageUrl": image.asset->url,
          tags
        }
      `;
      const data = await client.fetch(query);
      setFood(data);
    };

    fetchData();
  }, [slug]);

  useEffect(() => {
    sessionStorage.setItem("cart", JSON.stringify(cart));
  }, [cart]);

  const addToCart = (food: Foods) => {
    setCart((prevCart) => {
      const existingItem = prevCart.find((item) => item.name === food.name);
      if (existingItem) {
        return prevCart.map((item) =>
          item.name === food.name ? { ...item, quantity: (item.quantity || 1) + 1 } : item
        );
      }
```

3-Tags :-

I have enclosed the Tags along with the products, which screen shots along with code are given below.
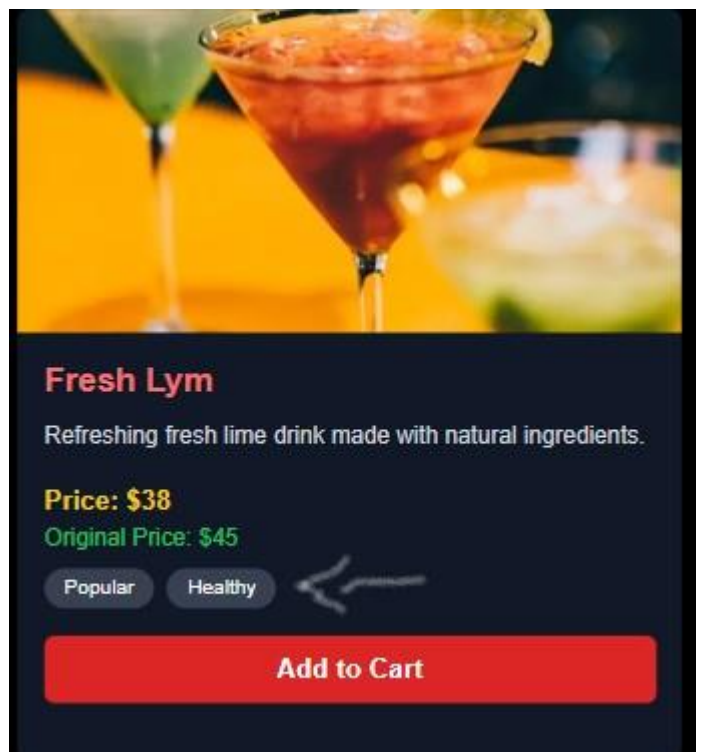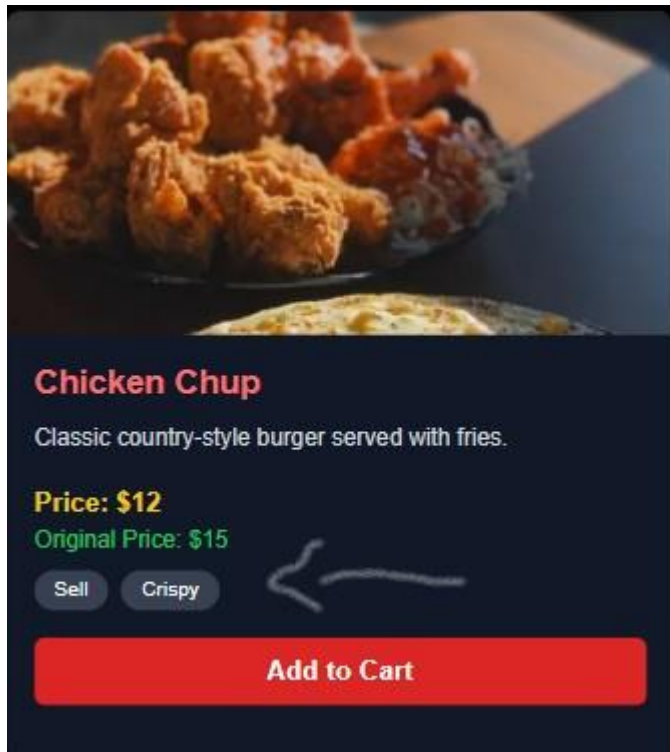
Tags Code:-

```
110
111        <div className="mt-2 flex flex-wrap gap-2">
112          {food.tags.map((tag, index) => (
113            <span
114              key={index}
115              className="text-xs ☐bg-gray-700 ◼text-white rounded-full px-3 py-1"
116            >
117              {tag}
118            </span>
119          ))}
120        </div>
121
```

4-Add to Cart with Functionality:-

Another Functionality I have added in my Food Hub website, which is "**Add to Cart**". When user visit on the menu page for choosing the products so he can select or buy any product via Add to Cart button.
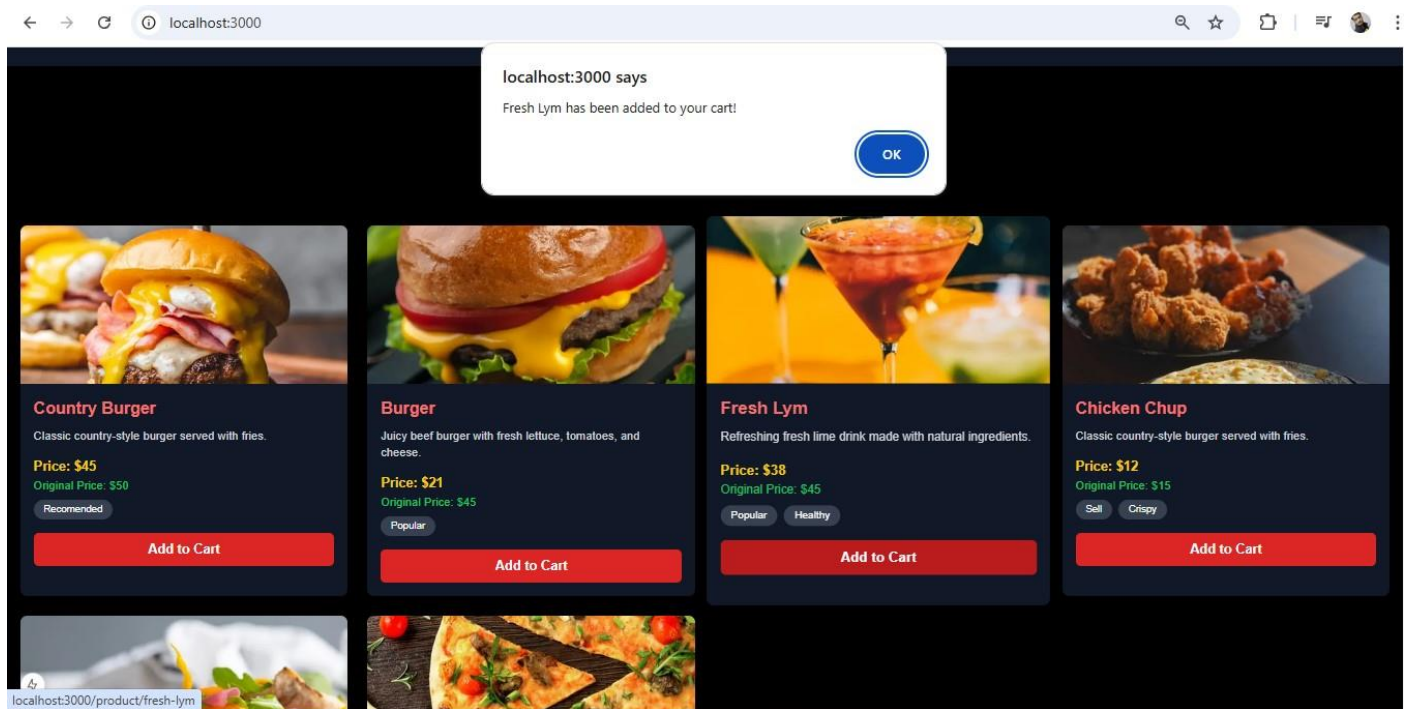
Ordered Product Cart Summary.

After click on add to cart button, product will be add in his cart list. After that he can add multiple products in his list and his cart summary will be show in-front of him with the ordered quantity and total amount.

Like ...

Decrease Product Cart Summary.

User can also decrease the ordered quantity from his cart summary.

Remove Product from Cart Summary.

User can also remove any product / items from his cart list, which he doesn't want to buy.

Code:-

```
61
62    const addToCart = (food: Foods) => {
63      setCart((prevCart) => [...prevCart, food]);
64      alert(`${food.name} has been added to your cart!`);
65    };
66
67    useEffect(() => {
68      fetchProducts();
69    }, []);
```
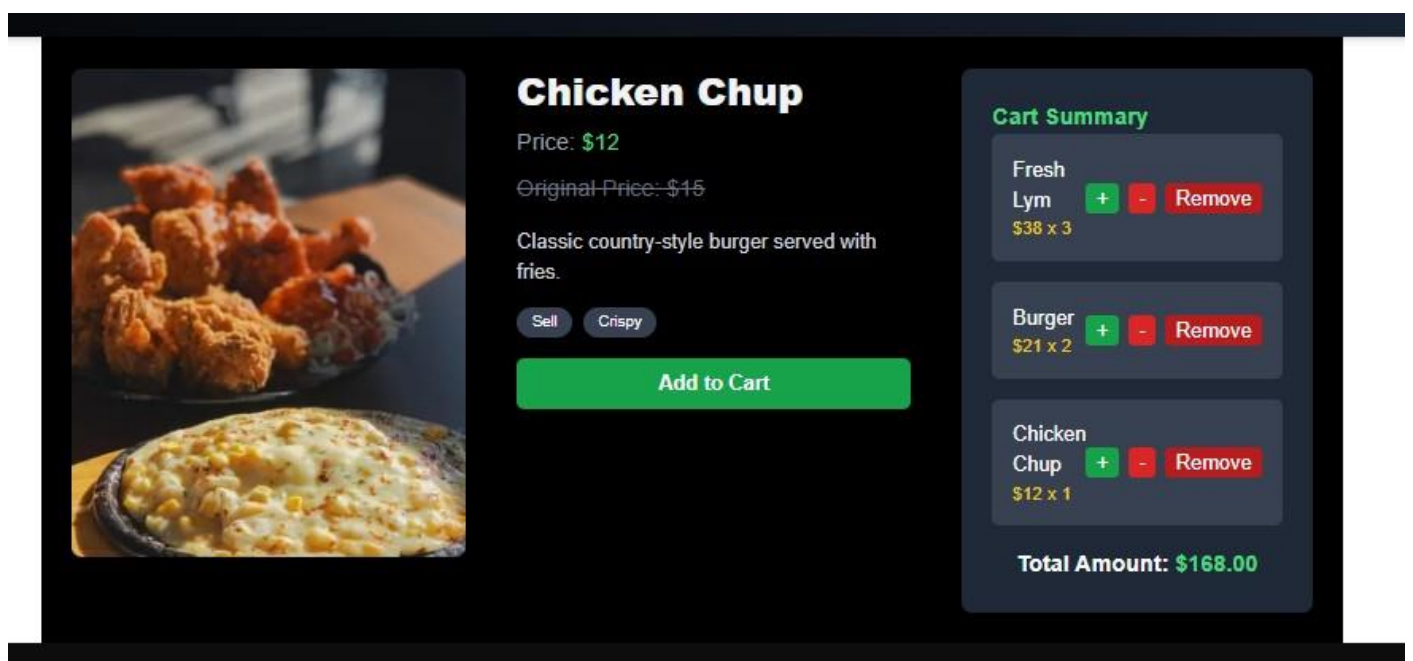
```
123              {/* Add to cart Functionality */}
124              <button
125                className="mt-4 w-full ■bg-red-600 ■text-white py-2 rounded-md ■hover:bg-red-700 transition-all font-
126                onClick={() => addToCart(food)}
127              >
128                Add to Cart
129              </button>
130            </div>
131          </Link>
132        </div>
133      ))}
134    </div>
135
136    {/* Cart Summary */}
137    <div className="mt-8 □bg-gray-800 p-6 rounded-lg shadow-md">
138      <h2 className="text-lg font-bold ■text-red-500">Cart Summary</h2>
139      {cart.length > 0 ? (
140        <ul className="space-y-4">
141          {cart.map((item, index) => (
142            <li
143              key={index}
144              className="flex justify-between items-center □bg-gray-700 shadow-sm p-4 rounded-md"
145            >
146              <div>
147                <p className="font-medium ■text-white">{item.name}</p>
148                <p className="text-sm ■text-yellow-400">
149                  ${item.price}
150                </p>
```

```
151                    <Image
152                      src={item?.imageUrl}
153                      alt={item.name}
154                      width={50}
155                      height={50}
156                      className="rounded-md"
157                    />
158                  </div>
159                </li>
160              ))}
161            </ul>
162          ) : (
163            <p className="■text-gray-400 text-center">
164              Your Cart is empty. Please add Products
165            </p>
166          )}
167        </div>
168      </div>
169    );
170  };
171
172  export default ProductCards;
173
```

5-FAQ Page:-

I have created the FAQ page in my website, where mostly asked users questions
of answers are given already or they can ask more questions from us regarding
to our services.

Frequently Asked Questions

Your queries, our answers. Find what you need below.

How do we serve food? — We serve high-quality food to ensure you have a wonderful experience.	**How can we get in touch with you?** +
How is our food quality? +	**What will be delivered? And when?** +
How do we give home delivery? +	**Is this restaurant 24 hours open?** +

Frequently Asked Questions

Your queries, our answers. Find what you need below.

How do we serve food? +	**How can we get in touch with you?** +
How is our food quality? +	**What will be delivered? And when?** +
How do we give home delivery? +	**Is this restaurant 24 hours open?** — Yes, absolutely. We are open 24/7

Same like above screen shots, users will find the answers of most asked questions into my FAQ Page.

FAQ Code:-

Tabs: page.tsx ...\[slug] U | page.tsx ...\food U | commentSection.tsx U | page.tsx ...\FAQ U ✕

src > app > FAQ > page.tsx > [∅] FAQ

```tsx
1    "use client";
2
3    import CommentSection from "@/components/commentSection";
4    import React, { useState } from "react";
5
6    const FAQ = () => {
7      const [openIndex, setOpenIndex] = useState<number | null>(null);
8
9      const faqs = [
10       {
11         question: "How do we serve food?",
12         answer:
13           "We serve high-quality food to ensure you have a wonderful experience.",
14       },
15       {
16         question: "How can we get in touch with you?",
17         answer:
18           "You can contact us via our website or helpline for any assistance.",
19       },
20       {
21         question: "How is our food quality?",
22         answer: "Our food quality is top-notch, and we use fresh ingredients.",
23       },
24       {
25         question: "What will be delivered? And when?",
26         answer: "Yes, delivery is timely and consistent with your order details.",
27       },
28       {
29         question: "How do we give home delivery?",
30         answer: "Our Riders will deliver your order at your door-steps.",
31       },
32       {
33         question: "Is this restaurant 24 hours open?",
34         answer: "Yes, absolutely. We are open 24/7",
35       },
36     ];
37
38     const toggleFAQ = (index: number) => {
39       setOpenIndex(openIndex === index ? null : index);
40     };
41
42     return (
43       <div className="w-full min-h-screen □bg-black py-16 px-4">
44         <section className="max-w-6xl mx-auto">
45           {/* Header */}
46           <h1 className="text-[48px] md:text-[56px] font-extrabold text-center ■text-white drop-shadow-lg">
47             Frequently Asked Questions
48           </h1>
49           <p className="text-center ■text-gray-400 text-lg md:text-xl mt-4">
50             Your queries, our answers. Find what you need below.
51           </p>
```

```
53              {/* FAQ Section */}
54              <div className="grid grid-cols-1 md:grid-cols-2 gap-8 mt-12">
55                {faqs.map((faq, index) => (
56                  <div
57                    key={index}
58                    className={`transition-transform duration-300 transform ${
59                      openIndex === index ? "scale-105 shadow-lg" : ""
60                    }`}
61                  >
62                    <div
63                      className="bg-[#1a1a1a] border border-gray-700 rounded-lg p-6 shadow-md cursor-pointer hover:shadow-
64                      onClick={() => toggleFAQ(index)}
65                    >
66                      {/* Question */}
67                      <div className="flex justify-between items-center">
68                        <h2 className="text-lg md:text-xl font-semibold text-gray-300">
69                          {faq.question}
70                        </h2>
71                        <span
72                          className={`text-3xl font-bold ${
73                            openIndex === index ? "text-white" : "text-gray-400"
74                          }`}
75                        >
76                          {openIndex === index ? "-" : "+"}
77                        </span>
78                      </div>
```

```
79
80                      {/* Answer */}
81                      {openIndex === index && (
82                        <p className="mt-4 text-gray-400 leading-relaxed">
83                          {faq.answer}
84                        </p>
85                      )}
86                    </div>
87                  </div>
88                ))}
89              </div>
90            </section>
91            <CommentSection />
92          </div>
93        );
94      };
95
96      export default FAQ;
97
```

6-Feed-Back / Comments:-

Users / Customers feed-backs are always value able for any marketplace or any business. That's why I have added the Feed-back / Comments section into my website.

After Feed-back of Customer/User.

Code:

```tsx
"use client";
import { useState } from "react";

const CommentSection = () => {
  const [comments, setComments] = useState<{ name: string; comment: string }[]>(
    []
  );
  const [name, setName] = useState<string>("");
  const [newComment, setNewComment] = useState<string>("");

  const handleAddComment = () => {
    if (name.trim() === "" || newComment.trim() === "") {
      alert("Please enter both your name and comment!");
      return;
    }

    setComments([...comments, { name, comment: newComment }]);
    setName("");
    setNewComment("");
  };

  return (
    <div className="mt-12">
      <h2 className="text-2xl text-slate-500 font-bold mb-4">Your Feed Back is Valueable for us:-</h2>

      {/* Comments List */}
      <ul className="space-y-4">
        {comments.map((comment, index) => (
          <li
            key={index}
            className="bg-gray-100 p-4 rounded shadow-md border border-gray-200"
          >
```

```tsx
            <p className="text-lg font--semibold text-gray-800">{comment.name}</p>
            <p className="text-gray-600 mt-1">{comment.comment}</p>
          </li>
        ))}
      </ul>

      {/* Add Comment Form */}
      <div className="mt-6">
        <div className="space-y-4">
          <div>
            <label
              htmlFor="name"
              className="block text-white font-medium mb-1"
            >
              Your Name
            </label>
            <input
              type="text"
              id="name"
              value={name}
              onChange={(e) => setName(e.target.value)}
              className="w-full border border-gray-300 p-2 rounded focus:ring-2 focus:ring-blue-400 focus:outline-non
              placeholder="Enter your name"
            />
          </div>
          <div>
            <label
              htmlFor="comment"
              className="block text-white font-medium mb-1"
            >
              Your Comment
            </label>
```

```tsx
65              <textarea
66                id="comment"
67                value={newComment}
68                onChange={(e) => setNewComment(e.target.value)}
69                rows={3}
70                className="w-full border █border-gray-300 p-2 rounded focus:ring-2 █focus:ring-blue-400 focus:outline-nor
71                placeholder="Write your comment here..."
72              />
73            </div>
74          </div>
75          <button
76            onClick={handleAddComment}
77            className="mt-4 █bg-blue-500 █text-white px-6 py-2 rounded █hover:bg-blue-600 transition duration-300"
78          >
79            Add Comment
80          </button>
81        </div>
82      </div>
83    );
84  };
85
86  export default CommentSection;
87
```
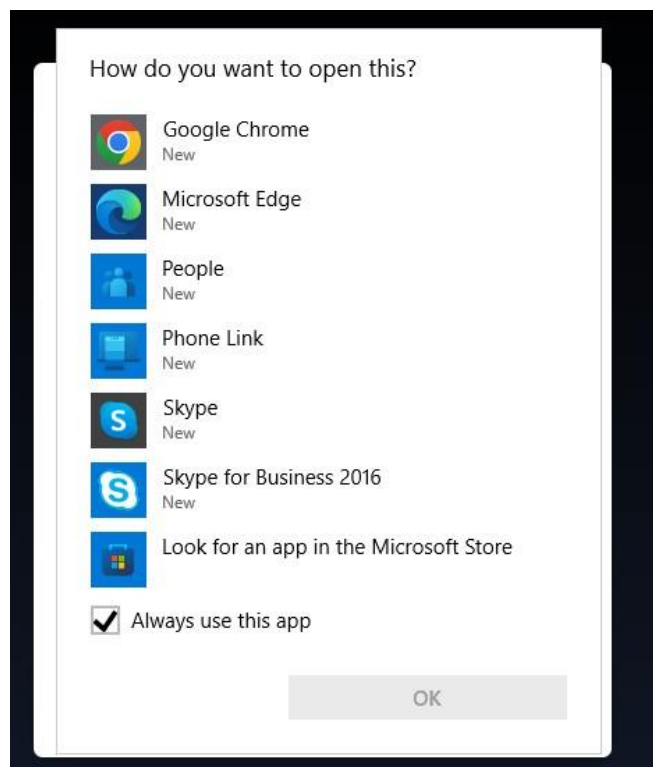
7-Contact Details:-

I have added the Contact details page separately for the users to reach & place the order to us.
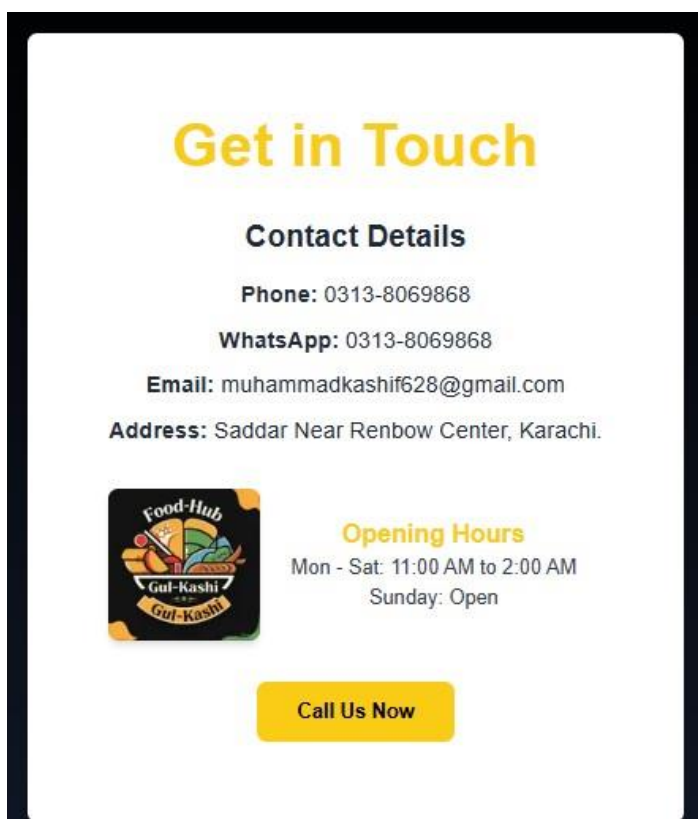
Upon click on "**Call us Now**" button, alert will show for the users to contact us through the pick an app.
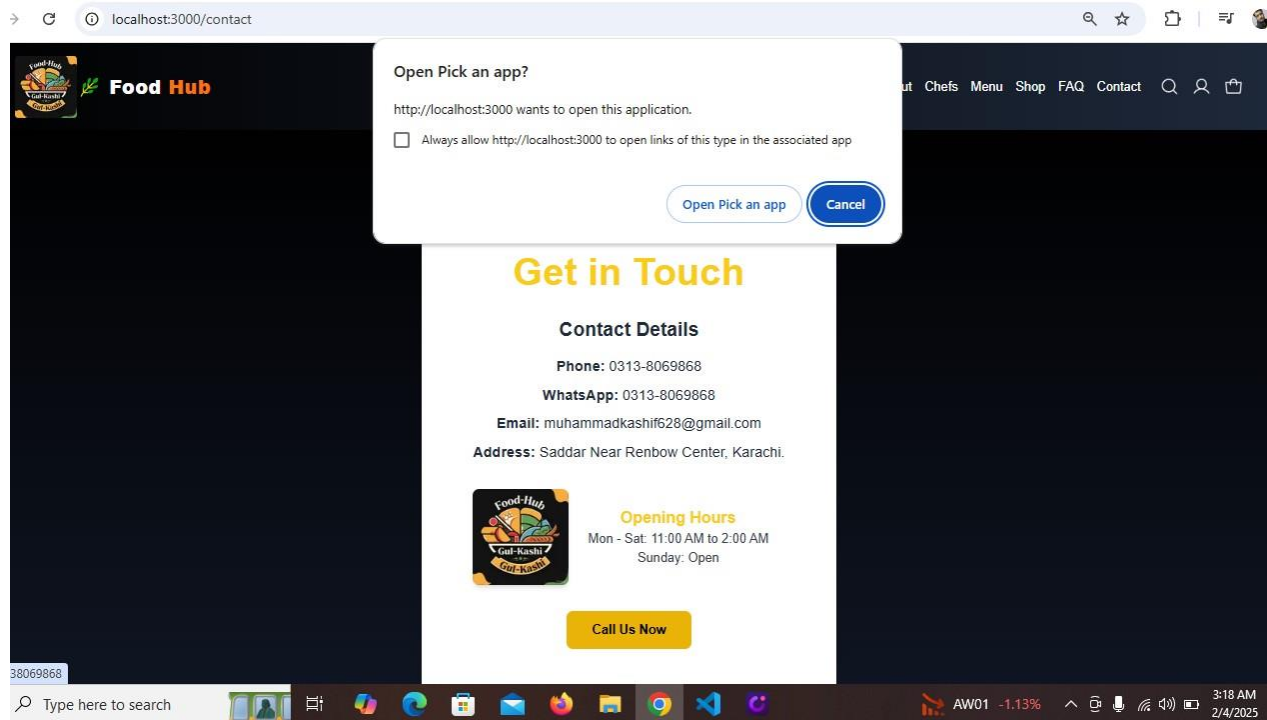

When user will click on pick an app, then another window will open for him to choose or select the app to place the order at our platform.
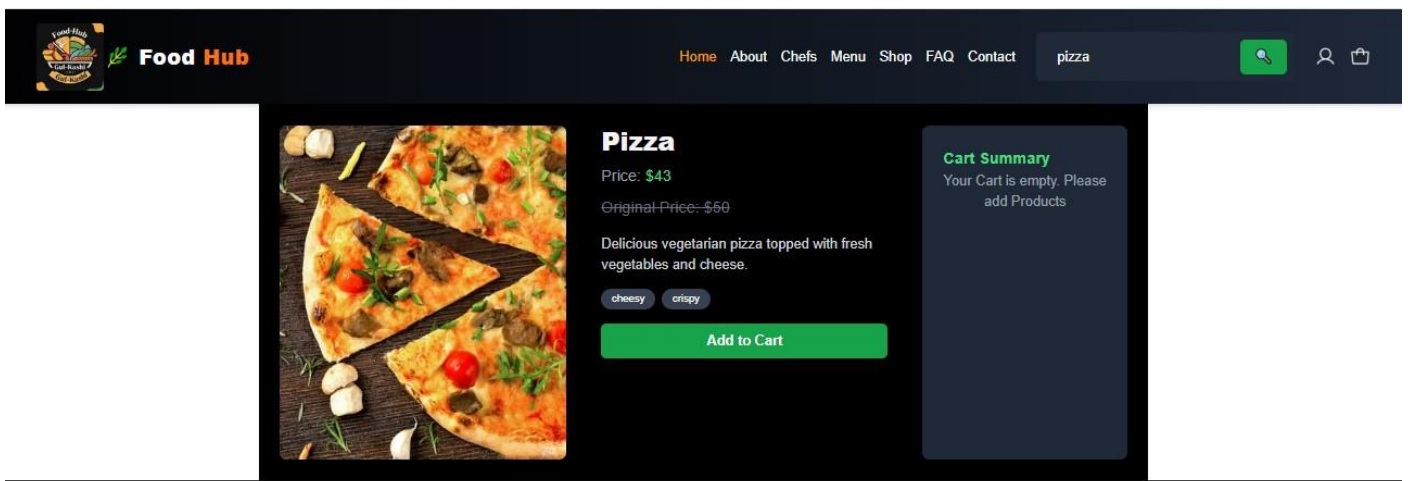
7-Search Bar:-

I have used the Search bar components in my code for the users to search & choose the product quickly. If they don't want to go on menu page and they directly search the product with the name, so this components will directly take them on the chosen product.
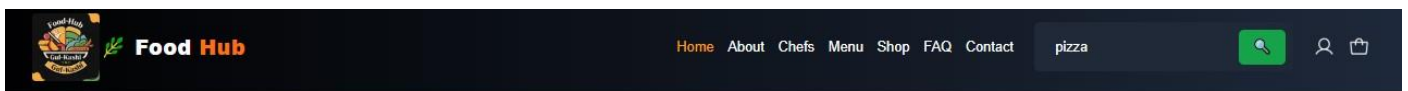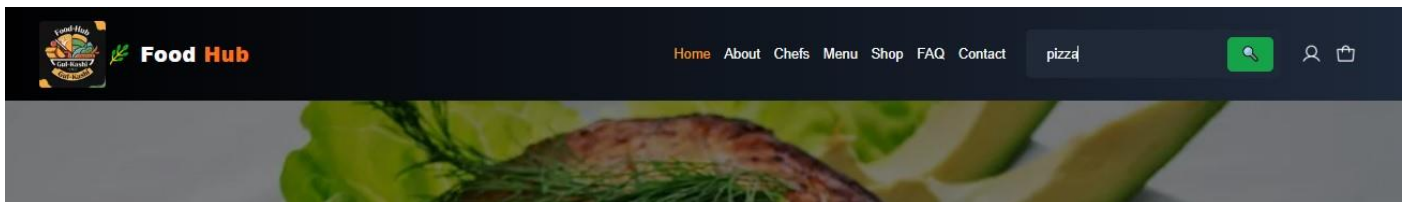
After clicking on the "**View product**" button. The separate page of the product will open, where they can add them in their cart list.

Code:-

Search Bar Components.

```tsx
"use client";

import { useState } from "react";
import { useRouter } from "next/navigation";

export default function SearchBar() {
  const [query, setQuery] = useState("");
  const router = useRouter();

  const handleSearch = (e: React.FormEvent) => {
    e.preventDefault();
    if (!query.trim()) return;
    router.push(`/search?query=${query}`);
  };

  return (
    <form onSubmit={handleSearch} className="flex items-center ☐bg-gray-800 p-2 rounded-md">
      <input
        type="text"
        value={query}
        onChange={(e) => setQuery(e.target.value)}
        placeholder="Search products..."
        className="bg-transparent ☐text-white px-4 py-2 focus:outline-none"
      />
      <button type="submit" className="☐text-white px-4 py-2 ☐bg-green-600 rounded-md ☐hover:bg-green-700 items-end">
        🔍
      </button>
    </form>
  );
}
```

Code:-

Search/page.tsx.

```tsx
      const data = await client.fetch(searchQuery);
      setResults(data);
    };

    fetchResults();
  }, [query]);

  return (
    <div className="max-w-5xl mx-auto p-6 ☐text-white">
      <h1 className="text-2xl font-bold">Search Results for "{query}"</h1>
      {results.length > 0 ? (
        <ul className="mt-4 space-y-4">
          {results.map((item, index) => (
            <li key={index} className="☐bg-gray-800 p-4 rounded-lg shadow-md flex justify-between items-center">
              <div>
                <p className="text-lg font-semibold">{item.name}</p>
                <p className="☐text-green-400">${item.price}</p>
              </div>
              <Link href={`/product/${item.slug}`} className="☐bg-green-600 px-4 py-2 rounded-md ☐text-white ☐hover:b
                View Product
              </Link>
            </li>
          ))}
        </ul>
      ) : (
        <p className="☐text-gray-400 mt-4">No results found.</p>
      )}
    </div>
  );
}
```

Import in Navbar

```tsx
"use client";

import Link from "next/link";
import Image from "next/image";
import SearchBar from "@/components/searchBar";
```

Import in Footer

```tsx
import React from "react";
import Image from "next/image";
import Link from "next/link";
import SearchBar from "@/components/searchBar";
```