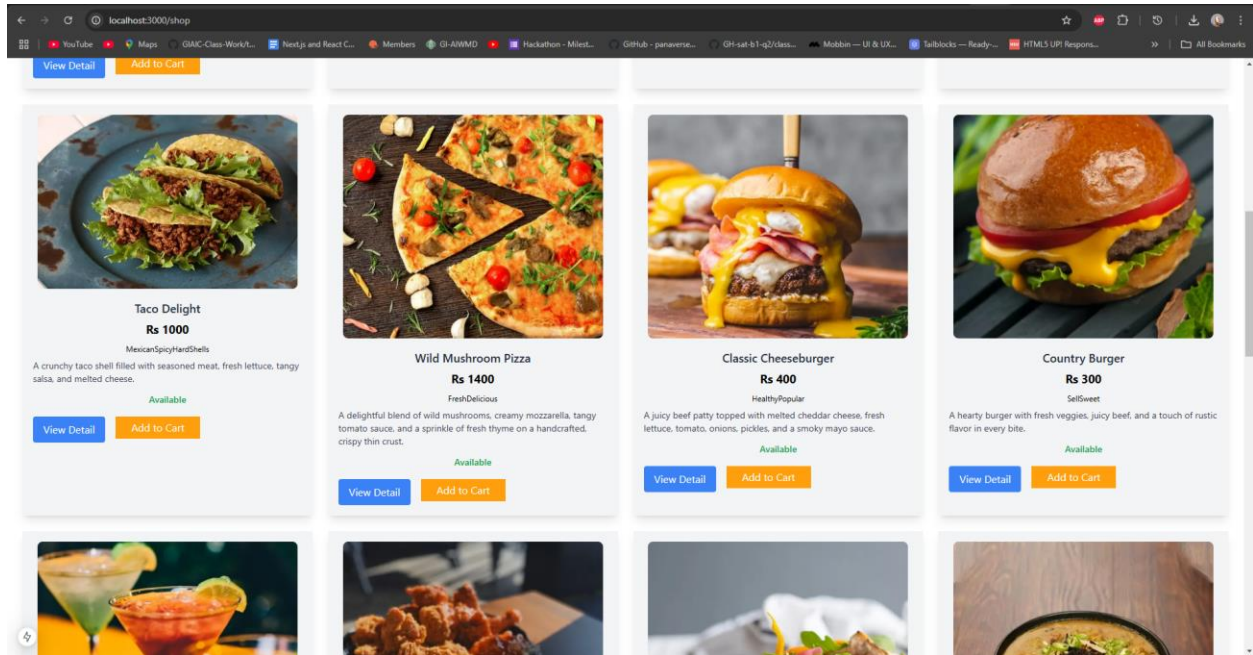


Day 4 - Dynamic Frontend Components –FoodTuck

1. The product listing page with dynamic data.



API INTEGRATION:

[Home](#)
[Menu](#)
[Blog](#)
[Chefs](#)
[About](#)
[Shop](#)
[Contact](#)

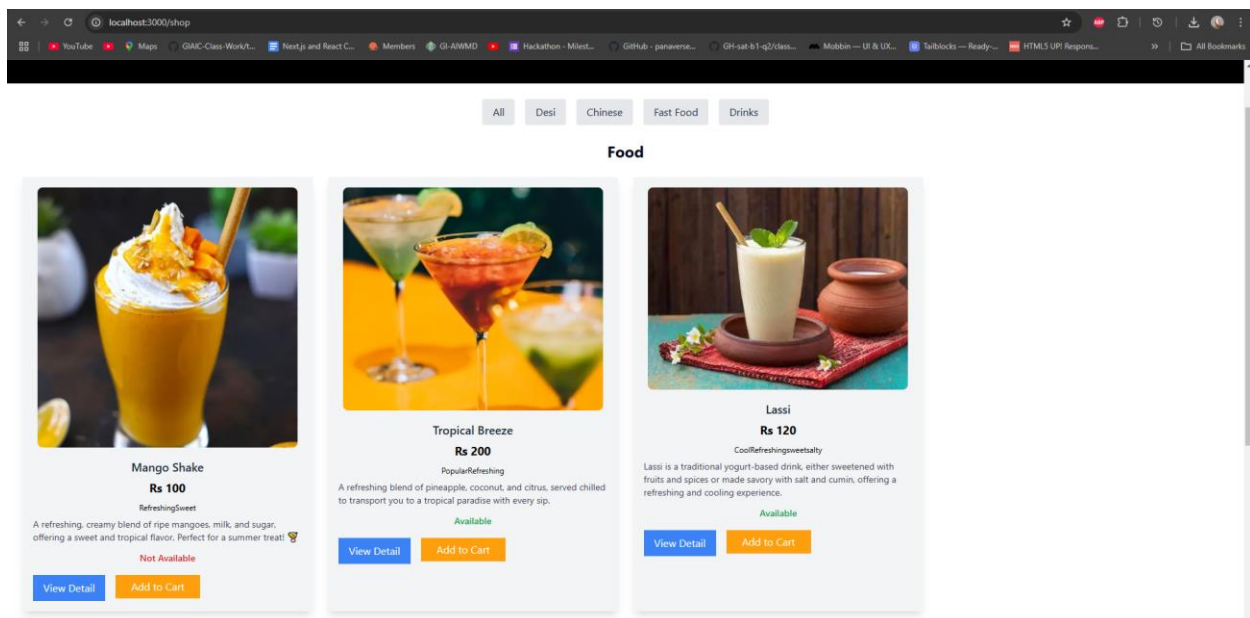
API INTEGRATION:

```

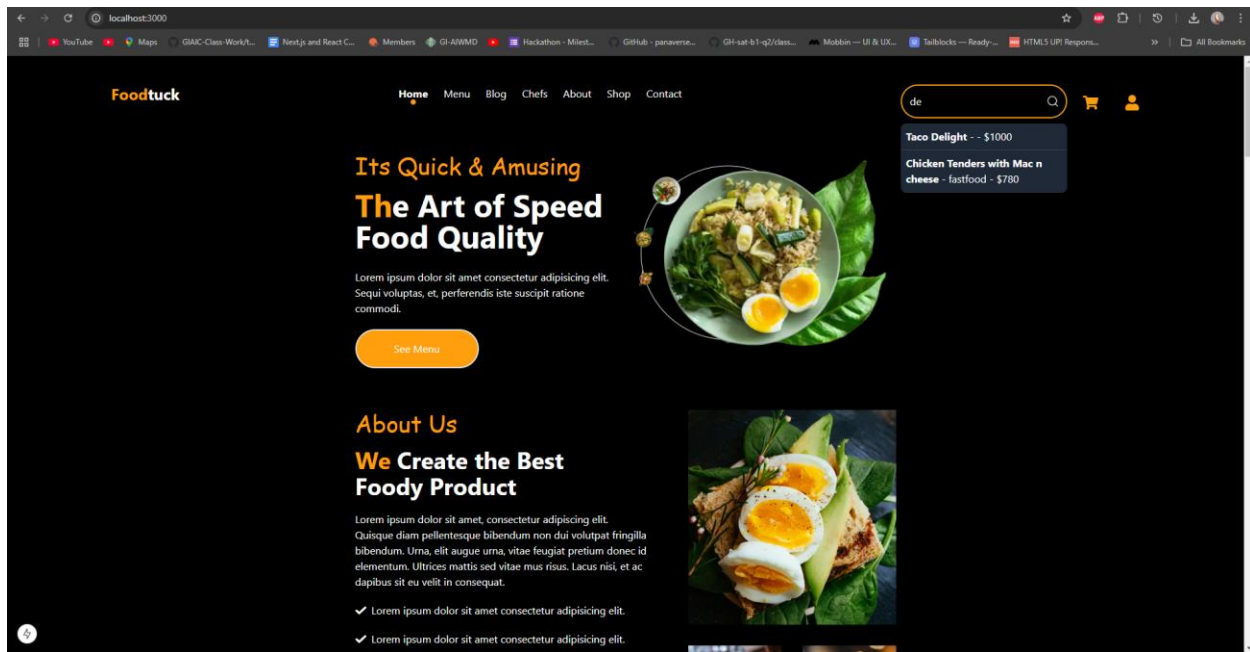
src > app > product > [slug] > page.tsx > Page
1  import { client } from "@sanity/lib/client";
2  import Image from "next/image";
3  import { urlFor } from "@sanity/lib/image";
4  import AddToCartButton from "@app/component/AddToCartButton/AddToCartButton";
5  import Counter from "@app/component/counter";
6  import { FaHeart, FaFacebook, FaInstagram, FaTwitter, FaYoutube } from "react-icons/fa";
7  import Link from "next/link";
8
9  import { FaStar, FaRegStar } from "react-icons/fa";
10 import ProjectStatus from "@app/public/Project Status.png"
11
12 type Review = {
13   user: string;
14   comment: string;
15   rating: number;
16 };
17
18
19 const Page = async ({ params }: { params: { slug: string } }) => {
20   const query = `*[ _type=='food' && slug.current == $slug ] {
21     _id, name, price, tags, image, description, available, category,
22     "imageUrl": image.asset->url,
23     "reviews": reviews[] {rating, comment, user},
24     slug
25   }[0]`;
26
27   const food = await client.fetch(query, { slug: params.slug });
28
29   if (!food) return <div className="text-center text-red-500">Product not found</div>;
30
31   const relatedItemsQuery = `*[ _type == 'food' && category == $category && slug.current != $slug ] {
32     _id, name, price, image, "imageUrl": image.asset->url, slug
33   }`;
34
35   const relatedItems = await client.fetch(relatedItemsQuery, { category: food.category, slug: food.slug.current });
36
37   return <div>
38     <AddToCartButton food={food} slug={slug} />
39   </div>;

```

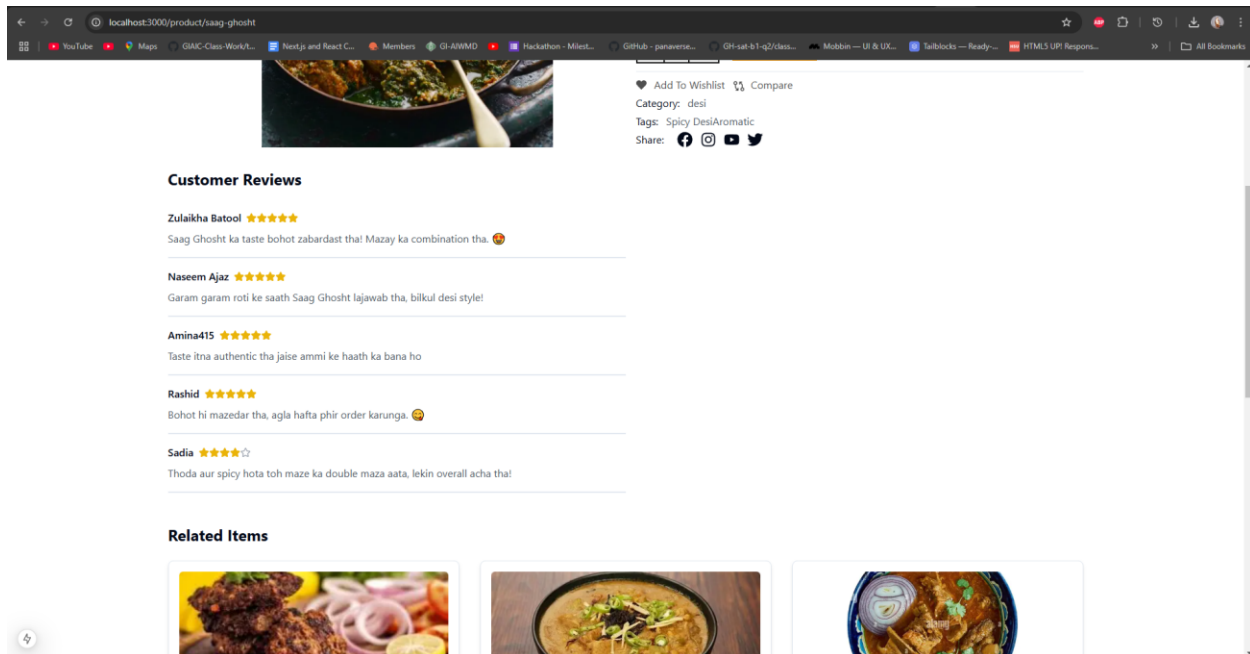
3.Category Component



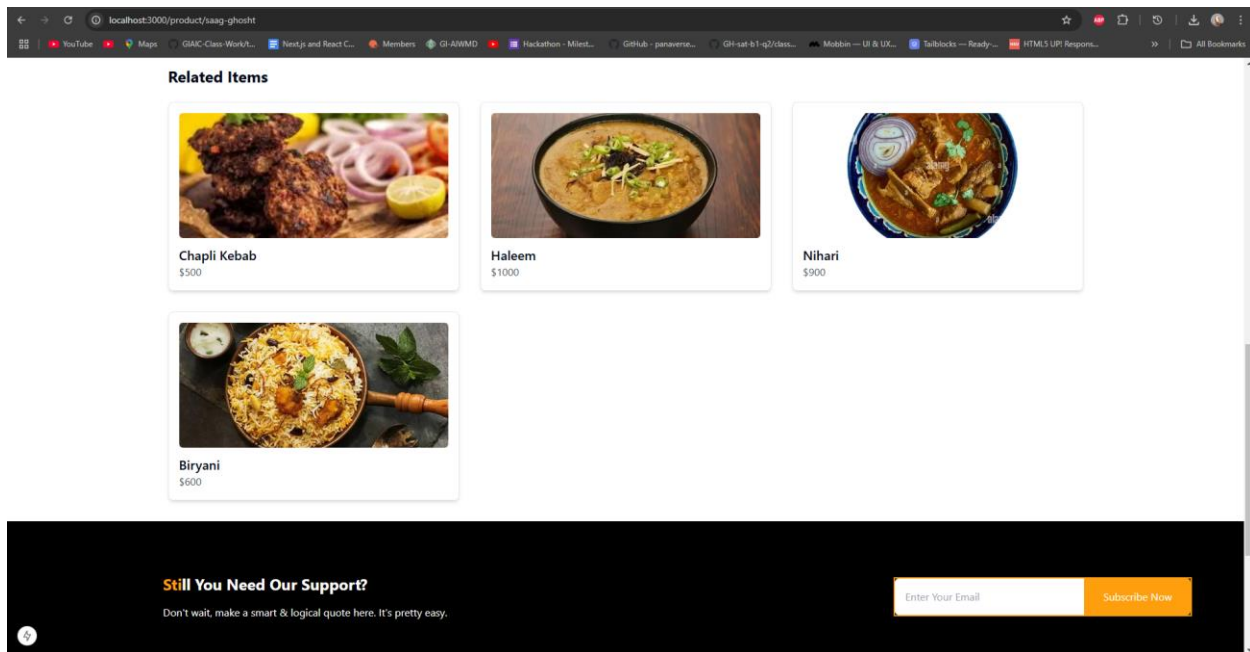
4. Search bar Component



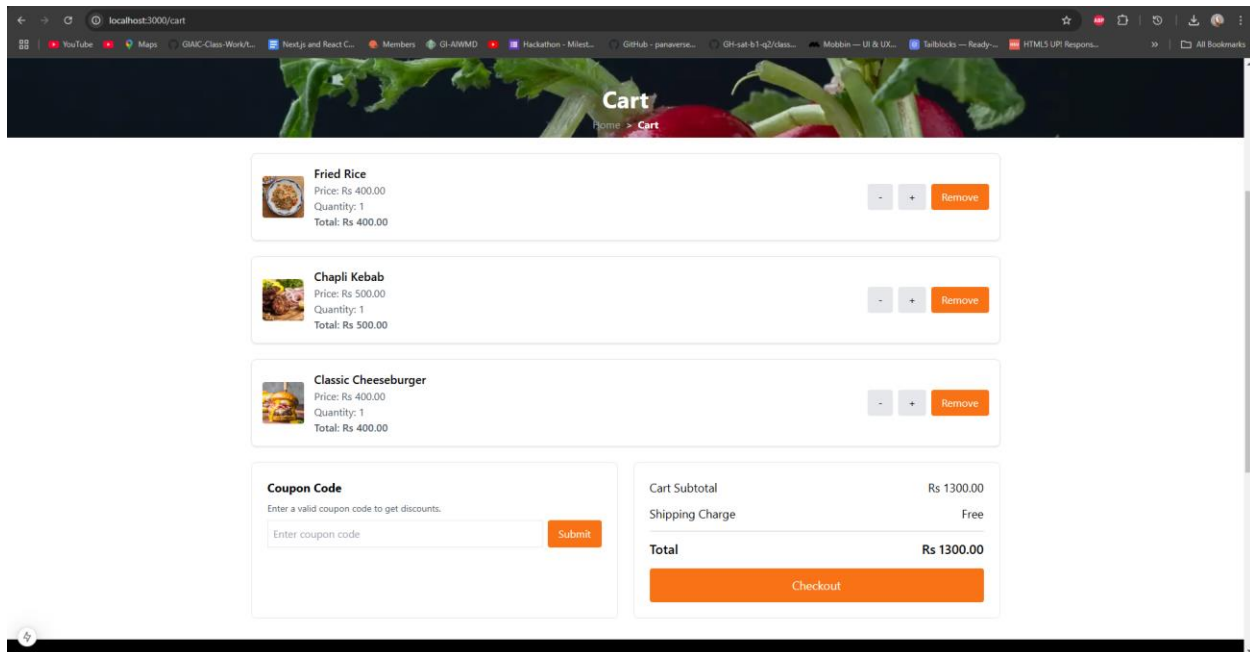
5. Reviews Section



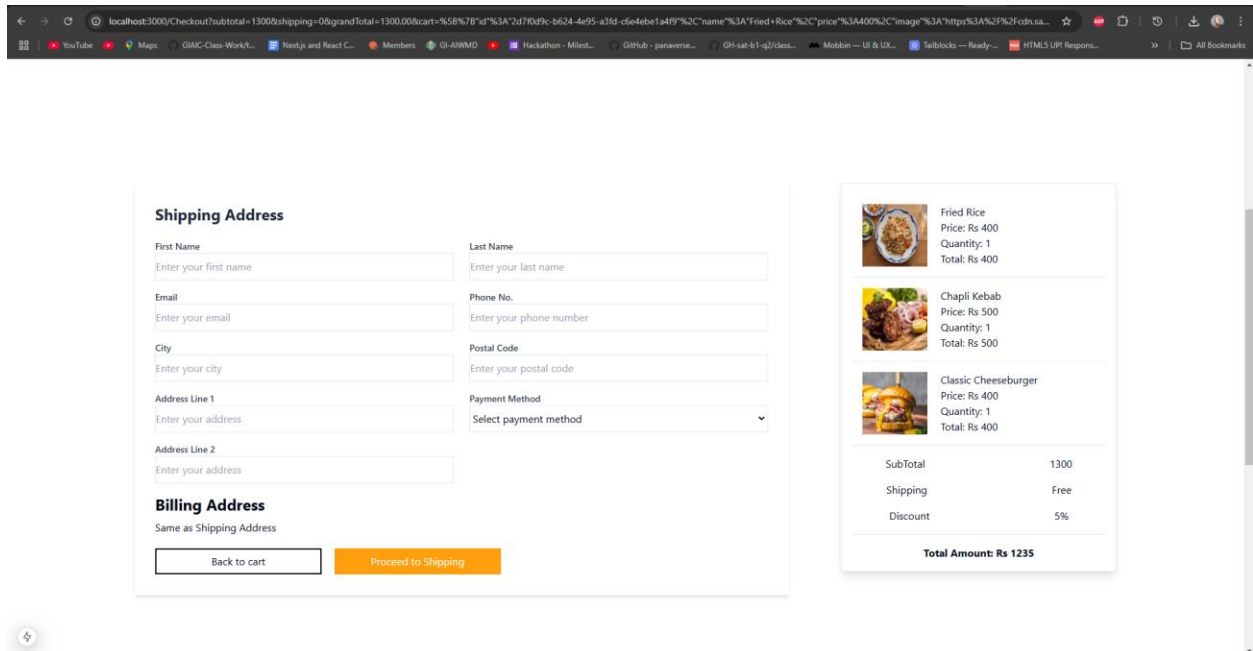
6.Related Component



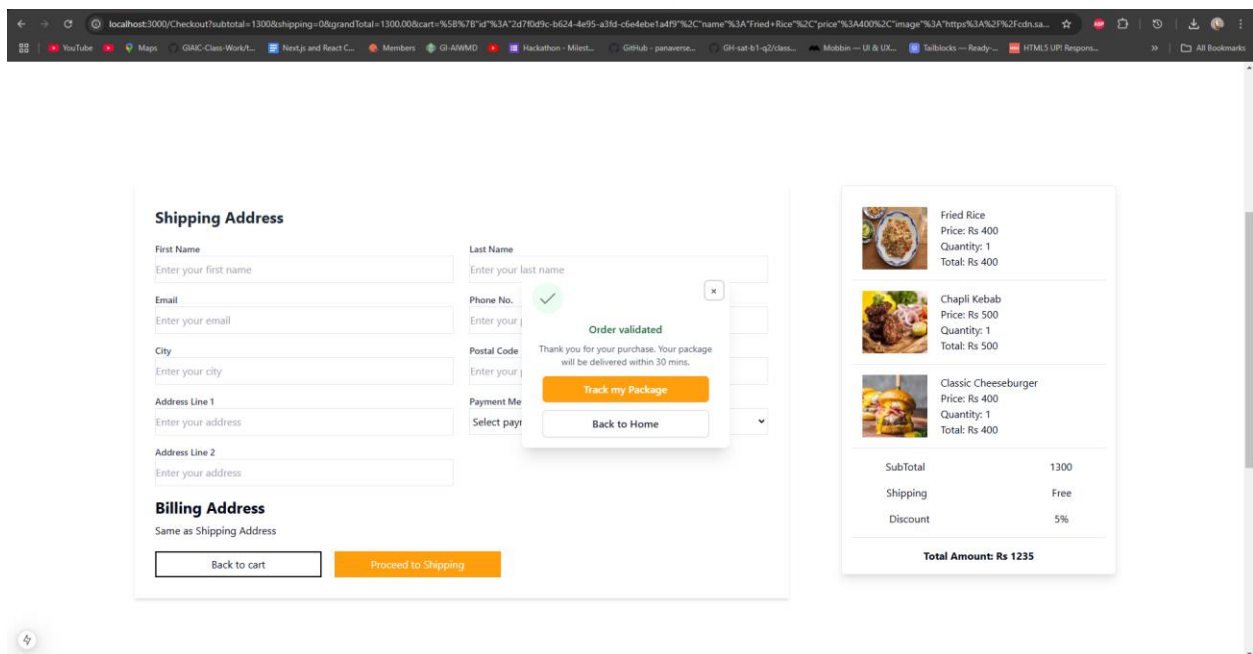
Functional Cart



Checkout Page:



Order Placed Component:



Code snippets for Search bar :

```
import React, { useState, useEffect } from "react";
import Link from "next/link";
import client from "@sanity/lib/client";
```

```

import magnify from '../public/magnify.png';
import Image from "next/image";

interface SearchBarProps {
  searchQuery: string;
  handleSearchChange: (query: string) => void;
  searchResults: { name: string; category: string; description?: string; price:
number; slug: string }[];
}

const Searchbar: React.FC<SearchBarProps> = ({ searchQuery, handleSearchChange,
searchResults }) => {
  const [inputValue, setInputValue] = useState(searchQuery);
  const [foods, setFoods] = useState<{ name: string; category: string;
description?: string; price: number; slug: string }[]>([]);

  // Fetch foods data
  useEffect(() => {
    const fetchFoods = async () => {
      const query = `*[_type == "food"]{
        _id,
        name,
        price,
        description,
        category,
        "slug": slug.current
      }`;

      try {
        const fetchedFoods = await client.fetch(query);
        setFoods(fetchedFoods);
      } catch (error) {
        console.error("Error fetching foods:", error);
      }
    };

    fetchFoods();
  }, []);

  const handleInputChange = (e: React.ChangeEvent<HTMLInputElement>) => {
    const value = e.target.value;
    setInputValue(value);
  };

```



```

    handleSearchChange(value);
  };

  const handleSearchClick = () => {
    handleSearchChange(inputValue);
  };

  const filteredResults = foods.filter(food =>
    food.name.toLowerCase().includes(inputValue.toLowerCase())
  );

  return (
    <div className="relative flex flex-col items-center mt-6">
      { /* Search Input */ }
      <div className="relative w-64">
        <input
          type="text"
          value={inputValue}
          onChange={handleInputChange}
          placeholder="Search food items..."
          className="p-3 pr-10 w-full bg-transparent border-2 border-[#FF9F0D]
text-white rounded-full focus:outline-none focus:ring-2"
        />

        <Image
          src={magnify}
          alt="Search"
          className="absolute right-3 top-1/2 transform -translate-y-1/2 w-5 h-5
cursor-pointer"
          onClick={handleSearchClick}
        />
      </div>

      {inputValue && filteredResults.length > 0 && (
        <ul className="mt-2 bg-gray-800 rounded-lg w-full max-h-40 overflow-y-
auto absolute z-10 top-full left-0">
          {filteredResults.map((food, index) => (
            <Link href={` /product/${food.slug}`} key={index} passHref>
              <li className="text-white p-2 border-b border-gray-700 hover:bg-
gray-600 cursor-pointer">
                <div>

```



```

        <strong>{food.name}</strong> - {food.category} - ${food.price}
      </div>
    </li>
  </Link>
  )}}
</ul>
)}
</div>
);
};

export default Searchbar;

```

Code snippets for ProductListing :

```

"use client";
import { useState, useEffect } from "react";
import Link from "next/link";
import AddToCartButton from "../component/AddToCartButton/AddToCartButton";
import Image from "next/image";
import client from "@sanity/lib/client";
import CategoryBox from "@app/component/categorybox";

const categories = ["All", "Desi", "Chinese", "Fast Food", "Drinks"];

const Shop = () => {
  const [foods, setFoods] = useState<any[]>([]);
  const [filteredFoods, setFilteredFoods] = useState<any[]>([]);
  const [selectedCategory, setSelectedCategory] = useState<string>("All");
  const [loading, setLoading] = useState<boolean>(true);
  const [error, setError] = useState<string | null>(null);

  useEffect(() => {
    const fetchFoods = async () => {
      try {
        const query = `*[_type == "food"]{
          _id,
          name,
          price,
          description,
          tags,

```

```

        available,
        category,
        "imageUrl": image.asset->url,
        "slug": slug.current
    }`;
    const data = await client.fetch(query);
    setFoods(data);
    setFilteredFoods(data);
  } catch (err) {
    setError("Failed to fetch food items.");
    console.error(err);
  } finally {
    setLoading(false);
  }
};

fetchFoods();
}, []));

useEffect(() => {
  console.log("Selected Category:", selectedCategory);
  console.log("All Foods Data:", foods);

  if (selectedCategory === "All") {
    setFilteredFoods(foods);
  } else {
    const filtered = foods.filter((food) =>
      Array.isArray(food.category)
        ? food.category.includes(selectedCategory)
        : (food.category?.toLowerCase() || "") ===
selectedCategory.toLowerCase()
    );

    console.log("Filtered Foods:", filtered);
    setFilteredFoods(filtered);
  }
}, [selectedCategory, foods]); //

return (
  <div className="bg-white p-6 mt-5">
    <CategoryBox categories={categories}
setSelectedCategory={setSelectedCategory} />
    <h1 className="text-2xl font-bold mb-6 text-center">Food</h1>

```

```

    {loading && <p>Loading food items...</p>}
    {error && <p className="text-red-600">{error}</p>}}
    {filteredFoods.length === 0 && !loading && <p className="text-center">No
items found.</p>}}

    <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-
cols-4 gap-6">
      {filteredFoods.map((food) => (
        <div key={food._id} className="bg-gray-100 shadow-lg p-4 flex flex-col
items-center h-full">
          <Link href={` /food/${food.slug}`} className="text-center">
            <Image src={food.imageUrl} alt={food.name} width={400} height={200}
className="rounded-lg" />
            <h2 className="text-lg font-semibold text-gray-800 mt-
4">{food.name}</h2>
          </Link>

          <p className="text-black text-lg font-bold mt-1">Rs {food.price}</p>
          <p className="text-black text-xs mt-2">{food.tags}</p>
          <p className="text-gray-700 text-sm mt-2">{food.description}</p>
          <p className={`text-sm font-medium mt-3 ${food.available ? "text-
green-600" : "text-red-600"}`}>
            {food.available ? "Available" : "Not Available"}
          </p>

          <div className="flex flex-col sm:flex-row gap-4 mt-4 w-full">
            <Link href={` /product/${food.slug}`} className="w-full sm:w-auto">
              <button className="px-4 py-2 bg-blue-500 text-white sm:w-auto
hover:scale-105 transition-transform duration-300">View Detail</button>
            </Link>

            <AddToCartButton
              food={{
                id: food._id,
                name: food.name,
                price: food.price,
                image: food.imageUrl || "/default-image.jpeg",

              }}
            />
          </div>
        </div>
      )}
    </div>

```

```
    })}
    </div>
  </div>
);
};

export default Shop;
```

Student Name: Yusra Fatima

Roll No: 00411279