Swiggy-style dynamic routing example in React

--

🧠 Concept Flow (with emojis!)

🗺️ Page Structure

```
/               → 🍽 Restaurant Listing Page (cards)
/restaurant/47589  → 📄 Restaurant Menu Page (dynamic route)
```

🔁 Data Flow

```
Home.jsx → List of cards → click on Link/route → dynamic menu
              ↓
App.jsx  → Define Route path="/restaurant/:resId"
              ↓
RestaurentMenu.jsx → useParams() → fetch menu for that ID → render
```

📦 1. `App.jsx` – Setup Routes

```jsx
import React from "react";
import { BrowserRouter, Routes, Route } from "react-router-dom";
import Home from "./Home";
import RestaurentMenu from "./RestaurentMenu";

const App = () => {
  return (
    <BrowserRouter>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/restaurant/:resId" element={<RestaurentMenu />} />
      </Routes>
    </BrowserRouter>
  );
};

export default App;
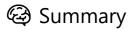```

---

## 🏠 2. `Home.jsx` – Show Restaurant Cards

```jsx
import React, { useEffect, useState } from "react";
import { Link } from "react-router-dom";

const Home = () => {
  const [restaurants, setRestaurants] = useState([]);

  useEffect(() => {
    fetchRestaurants();
  }, []);

  const fetchRestaurants = async () => {
    const data = await fetch("https://www.swiggy.com/dapi/restaurants/list/v5?
lat=23.022505&lng=72.5713621&page_type=DESKTOP_WEB_LISTING");
    const json = await data.json();
    const list = json?.data?.cards?.find(card => card?.card?.card?.id ===
"restaurant_grid_listing")?.card?.card?.gridElements?.infoWithStyle?.restaurants
|| [];
    setRestaurants(list);
  };

  return (
    <div>
      <h1>🍽 Nearby Restaurants</h1>
      <div style={{ display: "flex", flexWrap: "wrap", gap: "20px" }}>
        {restaurants.map((res) => (
          <Link to={`/restaurant/${res.info.id}`} key={res.info.id} style={{
textDecoration: "none", color: "black" }}>
            <div style={{ border: "1px solid #ddd", padding: "10px", borderRadius:
"8px", width: "200px" }}>
              <img
                src=
{`https://res.cloudinary.com/swiggy/image/upload/${res.info.cloudinaryImageId}`}
                alt={res.info.name}
                width="100%"
              />
              <h3>{res.info.name}</h3>
              <p>⭐ {res.info.avgRating}</p>
              <p>{res.info.cuisines.slice(0, 2).join(", ")}</p>
              <p>{res.info.areaName}</p>
            </div>
          </Link>
        ))}
      </div>
    </div>
  );
};

export default Home;
```
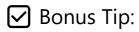

📄 3. `RestaurentMenu.jsx` – Dynamic Route Page

```jsx
import React, { useEffect, useState } from "react";
import { useParams } from "react-router-dom";

const RestaurentMenu = () => {
  const { resId } = useParams(); // 🗝️ Extract ID from route
  const [restInfo, setRestInfo] = useState(null);

  useEffect(() => {
    fetchMenu();
  }, []);

  const fetchMenu = async () => {
    const data = await fetch(
      `https://www.swiggy.com/dapi/menu/pl?page-type=REGULAR_MENU&complete-menu=true&lat=23.022505&lng=72.5713621&restaurantId=${resId}&catalog_qa=undefined&submitAction=ENTER`
    );
    const json = await data.json();
    setRestInfo(json?.data);
  };

  if (!restInfo) return <h2>Loading menu...</h2>;

  const restaurant = restInfo.cards.find((c) =>
c?.card?.card?.info)?.card?.card?.info;

  const menuItems =
    restInfo?.cards
      ?.find((c) => c.groupedCard?.cardGroupMap?.REGULAR)
      ?.groupedCard.cardGroupMap.REGULAR.cards
      ?.find((c) => c.card?.card?.itemCards)?.card?.card?.itemCards || [];

  return (
    <div>
      <h1>📄 {restaurant?.name}</h1>
      <p>{restaurant?.areaName}, {restaurant?.city}</p>
      <p>⭐ {restaurant?.avgRating} | {restaurant?.costForTwoMessage}</p>
      <hr />
      <h2>📋 Menu</h2>
      <ul>
        {menuItems.map(item => (
          <li key={item.card.info.id}>
            {item.card.info.name} - ₹{item.card.info.price / 100 ||
item.card.info.defaultPrice / 100}
          </li>
        ))}
      </ul>
    </div>
  );
};

export default RestaurentMenu;
```

🧠 Summary

Page	Role	Uses
`Home.jsx`	🔍 Shows list of restaurants	`Link to="/restaurant/:id"`
`App.jsx`	🖼️ Sets up dynamic routes	`Route path="/restaurant/:resId"`
`RestaurentMenu.jsx`	🍽️ Fetches & renders menu based on `resId`	`useParams()`

☑️ Bonus Tip:

You can now navigate directly to:

```
http://localhost:3000/restaurant/47589
```

And dynamically see the correct menu 🎉