

🔧 Problem Solving: Complete Menu Categories

Initial Problem

The original implementation only displayed one menu category instead of all available categories from the API response.

Original Code (Limited Display)

```
// ❌ Only showed one category
const itemCards = resInfo?.cards?.[4]?.groupedCard?.cardGroupMap?.REGULAR?.cards?.[2]?.card?.card?.itemCards || [];
```

Solution: Complete Menu Display

To resolve the issue of displaying all menu categories, we need to enhance the code to dynamically fetch and render all available categories from the API response. This involves:

1. API Structure Analysis

```
const fetchMenu = async() => {
  const data = await fetch(RESTAURANT_MENU_API + resId);
  const json = await data.json();

  // Debug logging to understand API structure
  console.log("Full API Response:", json.data);
  console.log("Cards structure:", json.data?.cards?.[4]?.groupedCard?.cardGroupMap?.REGULAR?.cards);

  setResInfo(json.data);
};
```

2. Enhanced Category Filtering

```
// ✅ Filter all menu category cards
const categories = resInfo?.cards?.[4]?.groupedCard?.cardGroupMap?.REGULAR?.cards?.filter(
  (card) =>
    card.card?.card?.["@type"] === "type.googleapis.com/swiggy.presentation.food.v2.ItemCategory" ||
    card.card?.card?.["@type"] === "type.googleapis.com/swiggy.presentation.food.v2.NestedItemCategory"
) || [];
```

3. Dynamic Category Rendering

```

return (
  <div className="menu">
    <h1>{name}</h1>
    <h2>{cuisines.join(", ")}</h2>
    <h3>{costForTwoMessage}</h3>

    {categories.length === 0 ? (
      <h2>No menu items available</h2>
    ) : (
      categories.map((category, index) => {
        const categoryTitle = category.card?.card?.title || `Category ${index + 1}`;

        // Handle both regular and nested categories
        let items = [];

        if (category.card?.card?.itemCards) {
          // Regular category with direct items
          items = category.card.card.itemCards;
        } else if (category.card?.card?.categories) {
          // Nested category - flatten all subcategories
          items = category.card.card.categories.reduce((acc, subCategory) => {
            return acc.concat(subCategory.itemCards || []);
          }, []);
        }

        return (
          <div key={index} className="menu-category">
            <h2>{categoryTitle} ({items.length} items)</h2>
            <ul>
              {items.map((item) => (
                <li key={item.card.info.id}>
                  <strong>{item.card.info.name}</strong> - ₹{(item.card.info.price / 100) ||
(item.card.info.defaultPrice / 100)}
                  {item.card.info.description && (
                    <p style={{fontSize: '12px', color: '#666', margin: '5px 0'}}>
                      {item.card.info.description}
                    </p>
                  )}
                </li>
              )}
            </ul>
          </div>
        );
      })
    )}
  </div>
);

```