React Notes: Filtering API Data in UI

Goal

> Fetch data from an API once ☑ Store it 💾 Filter it dynamically based on **user input** (e.g., search text) without re-fetching 🚫

Scenario

You fetch a list of restaurants from an API and want to **filter them** using a search bar.

☑ Requirements:

- Search field for name/cuisine
- Filtered results in real-time
- Preserve **original data** (to avoid re-fetching)

Core Concepts

Concept	Explanation
`useEffect()`	Fetch data on first render
`useState()`	Store API data and filtered data
Controlled Input	Search box bound to React state
Derived State	Filter UI data using `.filter()`

Component Flow

```
Mount ⬇
↓
useEffect() → fetch API → set full data (allRestaurants)
↓
User types in input (onChange)
↓
Filter allRestaurants based on input → setFilteredRestaurants
↓
Render filteredRestaurants to UI ☑
```

Full Example Code

```
import React, { useState, useEffect } from "react";
import RestaurantCard from "./RestaurantCard";
import Shimmer from "./Shimmer";

const Body = () => {
  const [allRestaurants, setAllRestaurants] = useState([]); // full data
  const [filteredRestaurants, setFilteredRestaurants] = useState([]); // filtered
UI data
  const [searchText, setSearchText] = useState(""); // controlled input

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

  const fetchData = async () => {
    const res = await fetch("https://mock-api/restaurants");
    const json = await res.json();

    // Suppose json.data.restaurants is the array
    setAllRestaurants(json.data.restaurants);
    setFilteredRestaurants(json.data.restaurants);
  };

  const handleSearch = () => {
    const filtered = allRestaurants.filter((res) =>
      res.name.toLowerCase().includes(searchText.toLowerCase())
    );
    setFilteredRestaurants(filtered);
  };

  // 🚨 optional chaining and early return
  if (!allRestaurants) return <Shimmer />;

  return (
    <div className="main">
      {/* 🔍 Controlled input */}
      <div className="search-bar">
        <input
          type="text"
          placeholder="Search Restaurants"
          value={searchText}
          onChange={(e) => setSearchText(e.target.value)}
        />
        <button onClick={handleSearch}>Search</button>
      </div>

      {/* 📃 Display filtered results */}
      <div className="restaurant-list">
        {filteredRestaurants.length === 0 ? (
          <h3>😔 No Restaurants Found</h3>
        ) : (
          filteredRestaurants.map((res) => (
            <RestaurantCard key={res.id} {...res} />
```

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

  export default Body;
```

🔍 Why Keep Two States?

State	Purpose
`allRestaurants`	Source of truth from API
`filteredRestaurants`	Derived state shown in UI

☑ This allows **non-destructive filtering** without losing original data.

☑ Best Practices

💡 Tip	☑ Why
Use `.filter()` on full data	So you don't lose original dataset
Lowercase `.includes()`	Case-insensitive search
Use controlled input	Ensures React manages form state
Debounce input (optional)	Improves performance in large data

🚨 Common Mistakes to Avoid

❌ Mistake	🔍 What Happens
Filtering directly on API response state	You lose original data
Updating `searchText` but not triggering filter	UI won't update
Not using `.toLowerCase()`	Search becomes case-sensitive
Fetching API on every input	Inefficient and unnecessary

🛠 Optional: Auto-Filtering on Input

```
useEffect(() => {
  const filtered = allRestaurants.filter((res) =>
    res.name.toLowerCase().includes(searchText.toLowerCase())
  );
```

```
    setFilteredRestaurants(filtered);
  }, [searchText]);
```

☑ This auto-filters **as the user types** (without clicking the button).

🎨 Bonus UX Enhancements

- Add **debounce** using `lodash.debounce()` for better performance
- Show **"No results found"** message when filter is empty
- Highlight matched text using `regex` if needed
- Add dropdown filters (e.g., by cuisine or rating)

🧪 Debugging Tips

```
console.log("All:", allRestaurants);
console.log("Filtered:", filteredRestaurants);
console.log("SearchText:", searchText);
```

📦 Real API to Test

Use https://dummyjson.com/products or JSON server to test dynamic filtering.

📝 Summary

💧 Task	🛠 Tool
Store API data	`useEffect + useState`
Filter it	`.filter()` logic on state
Show results	`map()` over filtered state
Handle input	Controlled component + search handler
