|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| |  | | --- | | On every key press “render()” will displayed    IF we don’t have to any change any thing then we can use normal local variable |  |  | | --- | | Problem with our search function  After searching a ‘king’, we wont be able other restaurant list.    This is happened because our filter function has changed our  RestaurantList | |

|  |
| --- |
| Monolith vs micro-services |

|  |
| --- |
| **Where to calll an api?**    Above is not the good way to call an api, it will be called after evry key press  Two Ways to call api  1) Load webpage => call API => render page                      (300ms)     (500ms)  2) Load webpage => render some UI => Call API => Update UI                      (100ms)           (300ms)     (500ms)  2nd is better method as it gives better user experience  We will be using **useEffect hook** to implememt 2nd way |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **useEffect Hook**    This will be when {searchText} changes   |  | | --- | | If its dependent array is black then it will be called once | | Example-2 |  |  | | --- | | “render” will be call first if page renders. |  |  | | --- | | This is the best way to call API.      useEffect with empty dependanancy array |  |  | | --- | |  |  |  | | --- | | Load webpage => render some UI => Call API => Update UI                      (100ms)           (300ms)     (500ms)  When reload our UI, it will render page with ‘RestaurantList’ and it re render the page with swiggy api. This happens quckly and called **Reconciliation** |  |  | | --- | | **Shimmer effect** |  |  | | --- | | Now UI will directly remder 20 restaurent list | |

|  |  |
| --- | --- |
| **Conditional rendering**   |  | | --- | | For shimmer UI    Changes made | |

|  |
| --- |
| When we search any Restaurant out of the Restaurant list.    Because restaurants Size become zero and it is the condition to load shimmer UI |

|  |
| --- |
| Making our search functionality work completely    Now maintaining two copy of Restaurants. One will be filtered and other will contain only all Restaurant list.      import RestrauntCard from "./RestruantCard"  import { useState, useEffect } from "react"  import ShimmerUI from "./ShimmerUI"  function filterData(searchInput, restaurants){      const filterData = restaurants.filter((restaurant) =>          restaurant?.info?.name?.toLowerCase()?.includes(searchInput.toLowerCase())      )      return filterData  }  const Body = () => {      const [allRestaurants, setAllRestaurants] = useState([])      const [filteredRestaurants, setFilteredRestaurants] = useState([])      const [searchInput, setsearchInput] = useState("")      useEffect(()=>{          getRestaurant()      }, [])      async function getRestaurant() {          const data = await fetch("https://cors-anywhere.herokuapp.com/https://www.swiggy.com/dapi/restaurants/list/v5?lat=30.29844139999999&lng=77.99313599999999&page\_type=DESKTOP\_WEB\_LISTING" )          const jsonData = await data.json()          console.log(jsonData)          setAllRestaurants(jsonData?.data?.cards[1]?.card?.card?.gridElements?.infoWithStyle?.restaurants)          setFilteredRestaurants(jsonData?.data?.cards[1]?.card?.card?.gridElements?.infoWithStyle?.restaurants)      }        return allRestaurants.length === 0 ? (<ShimmerUI/>) : (          <>          <div className="search-container">              <input                  type="text"                  className="search-input"                  placeholder="Search"                  value={searchInput}                  onChange={(e) =>{                      setsearchInput(e.target.value)                  }}              />                <button className="search-btn"              onClick={() =>{                  //need to filter data from RestaurantList                  //filterData                  const data =  filterData(searchInput, allRestaurants)                  setFilteredRestaurants(data)              }}                >                  Search              </button>          </div>          <div className="retaurant-list">               {                  f**ilteredRestaurant**s.map((restaurant) =>{                      return <RestrauntCard key={restaurant.info.id}  {...restaurant.info}/>                  })               }          </div>          </>      )  }  export default Body  Now we will be able to search restaurent |

|  |
| --- |
| Avoid rendering a component |

|  |  |
| --- | --- |
| It has two problem.   1. Our shimmer UI not displaying, instead no Restraunt found displayed 2. Our search bar went away      |  | | --- | |  | |