Part 1:

Higher Order Components

It takes a component and returns a component.

Work as enhancer

See this example

export const withPromotedLabel = (RestaurantCard)=>{

return (data)=>{

return(

<>

<label className="absolute p-4 bg-red-400 text-white">Recommended</label>

<RestaurantCard {...data} />

</>

)

}

}

We are not changing the input and over creating new component with help of this

This is called pure function.

import RestaurantCard, {withPromotedLabel} from "./RestaurantCard";

const PromotedRestoCards = withPromotedLabel(RestaurantCard);

{res.info.avgRating > 4.5 ? <PromotedRestoCards resData = {res}/> : <RestaurantCard resData = {res}/> }

Collapse integration

If parent want to control its child

We can not directly do it

We have to pass from chile to parent methods.

Like

Child

import { useState } from "react";

import { ITEM\_URL } from "../utils/constents";

const RestaurantCategory = ({data, showItem, setShowIndexMenu})=>{

const items = data?.card?.card?.itemCards;

// now we will manage state from parents aas we want to make it controlled component

// const [showItem, setShowItem] = useState(true);

const toggleItems = ()=>{

// setAccordianItem(!showItem);

setShowIndexMenu();

}

return (

<>

<div className="w-full bg-gray-50 shadow-lg p-4 mx-auto my-4 ">

<div className="justify-between flex" onClick={toggleItems}>

<span className="font-bold text-lg">{data.card.card.title} ({items.length})</span>

<span> ^ </span>

</div>

{

showItem && <div>

{

items.map(item =>{

return (

<div key={item.card.info.name} className="p-2 m-2 border-b-2 flex justify-between">

<div>

<h2 className="font-bold text-lg">{item.card.info.name}</h2>

<p>

&#8377; {item.card.info.price/100} /-

| {(item.card.info.offerTags !== undefined)?item.card.info.offerTags[0]?.title:''}

| {(item.card.info.offerTags !== undefined)?item.card.info.offerTags[0]?.subTitle:''}

</p>

<p>{item.card.info.description}</p>

<h3>{item.card.info.itemAttribute.vegClassifier}</h3>

</div>

<div>

{

item.card.info.imageId &&

<img src={ITEM\_URL+item.card.info.imageId} className="w-40" />

}

</div>

</div>

)

})

}

</div>

}

</div>

</>

)

}

export default RestaurantCategory;

Parent

import { useEffect, useState } from "react";

import SimmerUIResList from './SimmerUIResList';

import { MENU\_URL } from "../utils/constents";

import { useParams } from "react-router-dom";

import Error from './Error';

import useRestaurantMenu from "../utils/useRestaurantMenu";

import RestaurantCategory from "./RestaurantCategory";

const RestaurantMenu = () => {

const [showIndexMenu, setShowIndexMenu] = useState(0);

const {id} = useParams();

const resData = useRestaurantMenu(id);

if (resData == null) return <SimmerUIResList />;

const restroInfo = resData?.cards[0]?.card?.card;

const dineTypes = resData?.cards[1]?.card?.card?.tabs;

const aboutCard = resData?.cards[2]?.card?.card.info;

const mainMenuCategories = resData?.cards[4]?.groupedCard.cardGroupMap.REGULAR.cards.filter(c=> c.card?.card?.['@type'] === 'type.googleapis.com/swiggy.presentation.food.v2.ItemCategory');

const topPicks = resData?.cards[4]?.groupedCard.cardGroupMap.REGULAR.cards[1].card.card.carousel;

const showThisItem= (e)=>{

console.log('showThisItem');

}

return (

<>

<div className="max-w-[1000px] mx-auto px-4">

<div className="text-left mt-4 mb-4">

<h1 className="text-2xl">{restroInfo.text}</h1>

</div>

{dineTypes &&

<div className="flex justify-around border-b">

<button className={`py-2 px-4 font-semibold focus:outline-none text-blue-500 border-b-2 border-blue-500`} >

{dineTypes[0].title}</button>

<button className={`py-2 px-4 font-semibold focus:outline-none text-gray-500 hover:text-blue-500`} onClick={() => alert("yet to integreate")}>

{(dineTypes[1] !== undefined)?dineTypes[1].title:''}</button>

</div>

}

<div className="mx-auto mt-5 p-4 bg-white rounded-lg shadow-lg border">

<div className="flex items-center justify-between mb-2">

<div className="flex items-center">

<span className="text-green-600 text-xl font-semibold">{aboutCard.avgRating}</span>

<span className="text-gray-500 ml-1 text-sm">{aboutCard.totalRatingsString}</span>

</div>

<span className="text-gray-700">{aboutCard.costForTwoMessage}</span>

</div>

<div className="text-orange-500 text-sm mb-2">

{

aboutCard.cuisines.map((e, indx)=>{

return (<a href="#" key={'cusin'+indx} className="hover:underline"> {e} </a> )

})

}

</div>

<div className="flex items-center mb-2">

<span className="text-gray-700 font-semibold">Outlet</span>

<span className="ml-2 text-gray-500">{restroInfo.areaName}</span>

</div>

<div className="text-gray-500 mb-4">

<span>{aboutCard?.sla?.slaString}-</span>

</div>

<div className="flex items-center text-gray-500 text-sm">

<svg className="w-4 h-4 mr-1" fill="none" stroke="currentColor" viewBox="0 0 24 24" xmlns="http://www.w3.org/2000/svg">

<path strokeLinecap="round" strokeLinejoin="round" strokeWidth={2} d="M13 16V6a1 1 0 011-1h3a1 1 0 011 1v6a1 1 0 001 1h2M5 9l7 7m0 0l7-7m-7 7V3" />

</svg>

<span>{aboutCard?.feeDetails?.message}</span>

</div>

</div>

<div className="text-center mt-4 mb-4">

<hr/>

<h2 className="text-lg">~ Menu ~</h2>

<hr/>

</div>

<div className="w-full mt-6">

<h2 className="text-2xl font-semibold mb-4">Top Picks</h2>

<div className="flex space-x-4 overflow-x-scroll p-4">

{topPicks.map((item, index) => (

<div key={index} className="w-64 bg-white rounded-lg shadow-lg overflow-hidden flex-shrink-0" >

<img src={'https://media-assets.swiggy.com/swiggy/image/upload/fl\_lossy,f\_auto,q\_auto,w\_292,h\_300/'+item.creativeId} alt={item.title} className="w-full h-40 object-cover" />

<div className="p-4">

<h3 className="text-lg font-bold">{item.title}</h3>

<p className="text-gray-600 text-sm mt-2">{item.dish.info.description}</p>

<div className="mt-4 flex justify-between items-center">

<span className="text-lg font-semibold">Rs.{item.dish.info.price/100}/-</span>

<button className="bg-green-500 text-white px-4 py-2 rounded-full">

ADD

</button>

</div>

</div>

</div>

))}

</div>

</div>

{

mainMenuCategories.map((cat, index)=>{

return(

<RestaurantCategory

key={index+1}

data={cat}

showItem={index == showIndexMenu && true}

setShowIndexMenu={()=> setShowIndexMenu(index)}

/>

)

})

}

{/\* Main Div \*/}

</div>

</>

)

}

export default RestaurantMenu;

This is called LIFTING THE STATE UP.

We have covered

Higher order component

Lifting the state up

Controller and uncontrolled component

Accordions

Props Drilling

useContext

createContext from react

Lets create

import { createContext } from "react";

const UserContext = createContext({

"name":"Default User"

});

export default UserContext;

Lets use

import { useContext } from "react";

const {name} = useContext(UserContext);

log(name)

But there is different way to call this in class based component

Lets see

import { Component } from "react";

import UsersClass from "./UsersClass";

import Users from "./Users";

import UserContext from "../utils/UserContext";

class About extends Component {

constructor(props){

super(props);

console.log("About us Parent Constructor Called");

}

componentDidMount(){

console.log("Parent Component Did Mount method Called");

this.timer = setInterval(() => {

console.log("set interval functionality");

}, 1000);

}

componentWillUnmount(){

clearInterval(this.timer);

}

render(){

console.log("Parent Render Method Called");

return(

<div>

<UserContext.Consumer>

{

(data)=>{

console.log(data);

}

}

</UserContext.Consumer>

<h1> ABout us Class Component</h1>

<UsersClass name={'first'} location={'First'} />

<Users name={'Second'} />

{/\* <UsersClass name={'Thirt'} location={'Thirt'} /> \*/}

</div>

)

}

}

export default About;

.consumer to use this.

To update and use updated value we have to use .Provider

And it will be only effective for the section we wrapped it. If we wrapped a section i tiwll reflect in that only if whole body then i twill reflect in whole body.

Example

See example

import React, { lazy, Suspense, useContext, useEffect, useState } from 'react';

import ReactDOM from 'react-dom/client';

import Footer from './components/Footer';

import Header from './components/Header';

import HomeRestaurants from './components/HomeRestaurants';

import { createBrowserRouter, Outlet, RouterProvider } from 'react-router-dom';

import About from './components/About';

import Contact from './components/Contact';

import Error from './components/Error';

import RestaurantMenu from './components/RestaurantMenu';

import UserContext from './utils/UserContext';

const Grocery= lazy(()=> import('./components/Grocery') );

const AppLayout = () =>{

const {name} = useContext(UserContext);

const [userName, setUserName] = useState(name);

useEffect(()=>{

// suppose we called and api and get data like

const json = {"name":"Akshay Saini"};

setUserName(json.name);

},[]);

return (

<div>

<UserContext.Provider value={{name:userName}}>

<Header />

</UserContext.Provider>

<Outlet />

<Footer />

</div>

)

}

Or

<UserContext.Provider value={{name:userName}}>

<Header />

<Outlet />

<Footer />

</UserContext.Provider>

It will be accessible to all three components

This is also valid, nested providers

<UserContext.Provider value={{name:userName}}>

<Header />

<UserContext.Provider value={{name:"new"}}>

<Outlet />

</UserContext.Provider>

<Footer />

</UserContext.Provider>

const {name, setUserName} = useContext(UserContext);

<label>Welcome :

<input type="text" className="border-lime-100 p-1 shadow-xl" value={name} onChange={(e)=>{

setUserName(e.target.value);

}}/>

</label>