Part 1

App: Food Ordering App.

The first thing is to create planning.

Website

-> Header

-> logo

-> nav bar

-> body

-> search

-> card container

-> cards

-> Footer

- copyright

-> links

We are going to use App.js only to build this and we will break all above things into components and components within components

Example Header, Body, Navbar, Search, CardContainer, Card, Footer, Copyright and so on.

There are many ways to create app

We will start with basic one

1. Open the App.js file
2. Wirte components with help of css (seprately or inline
3. Render the code

In JSX there is a different way to write inline css.

Html code.

<h1 style="color:red; font-size:2rem">Hello World</h1>

But in jsx style we have to pass like js object

Jsx

<h1 style={{ color: "red", fontSize: '2rem' }}>Hello World</h1>

Double curly braces because 1st show its js code and 2nd is part of object

Or

const AppLayout = () =>{

const headingStyle = {

color: "red",

fontSize: '2rem'

}

return (

<h1 style={headingStyle}>Hello World</h1>

)

}

Now basic layout is ready

import React from 'react';

import ReactDOM from 'react-dom/client';

const Heading = () =>{

return (

<header className="header">

<div className="logo">MyLogo</div>

<nav className="navbar">

<a href="#" className="nav-link">Home</a>

<a href="#" className="nav-link">About</a>

<a href="#" className="nav-link">Services</a>

<a href="#" className="nav-link">Contact</a>

</nav>

</header>

)

}

const RestaurantCard = () =>{

return (

<div className="card-container">

<div className="card">

<img src="https://media-assets.swiggy.com/swiggy/image/upload/fl\_lossy,f\_auto,q\_auto,w\_660/xvyrclhxftulsglktaek" alt="Card Image" className="card-image" />

<div className="card-content">

<h2 className="card-title">Card Title</h2>

<p className="card-description">This is a brief description of the card. It can include details about food, hotel, etc.</p>

<div className="card-details">

<span className="card-rating">Rating: ★★★★☆</span>

<span className="card-cost">Cost: $20</span>

<span className="card-time">Time: 30 mins</span>

</div>

</div>

</div>

</div>

)

}

const AppLayout = () =>{

return (

<div>

<Heading />

<main className="main-content">

<div className="search-bar">

<input type="text" placeholder="Search..." />

<button type="button">Search</button>

</div>

<div className='res-container'>

<RestaurantCard />

<RestaurantCard />

</div>

</main>

<footer className="footer">

<div className="footer-links">

<a href="#" className="footer-link">Privacy Policy</a>

<a href="#" className="footer-link">Terms of Service</a>

</div>

<p className="footer-copyright">&copy; 2024 MyWebsite. All rights reserved.</p>

</footer>

</div>

)

}

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(<AppLayout />);

//Take css also, not mentioned here

Now we know there should be dynamic things

We we need to make components dynamic

In js how we can make things dynamic

For example we have an example to add two numbers but we pass these numbers dynamically

So the output is dependent on the input it will be very based on its input.

Input will be received in parameters and passed as arguments

In jsx we called that argument as props (properties)

When we pass any attribute to any component it will be argument for the function which is going to call in the background.

Lets see how to pass props.

<RestaurantCard resName=”Meghna Food” />

<RestaurantCard resName=”KFC” />

This is called passing props to components.

Whatever you will pass as an argument will be received in the key called props.

You can access all the attributes like props.key1, props.key2.

Like

<RestaurantCard

resName = "Meghna Foods"

cousins = "Biryani, North Indian, South Indian"

rating = "★★★★☆"

cost = "20"

time = "30 minutes"

image = 'https://media-assets.swiggy.com/swiggy/image/upload/fl\_lossy,f\_auto,q\_auto,w\_660/xvyrclhxftulsglktaek'

/>

const RestaurantCard = (props) =>{

return (

<div className="card-container">

<div className="card">

<img src={props.image} alt="Card Image" className="card-image" />

<div className="card-content">

<h2 className="card-title">{props.resName}</h2>

<p className="card-description">{props.cousins}</p>

<div className="card-details">

<span className="card-rating">Rating: {props.rating}</span>

<span className="card-cost">Cost: ${props.cost}</span>

<span className="card-time">Time: {props.time}</span>

</div>

</div>

</div>

</div>

)

}

Or we can also use props in other way (term called de-structuring on the fly)

const RestaurantCard = ({resName, cousins,rating, cost,time, image}) =>{

return (

<div className="card-container">

<div className="card">

<img src={props.image} alt="Card Image" className="card-image" />

<div className="card-content">

<h2 className="card-title">{props.resName}</h2>

<p className="card-description">{props.cousins}</p>

<div className="card-details">

<span className="card-rating">Rating: {props.rating}</span>

<span className="card-cost">Cost: ${props.cost}</span>

<span className="card-time">Time: {props.time}</span>

</div>

</div>

</div>

</div>

)

}

Ultimately what is happening

When you passing attributes

Its creating an object

Const props = {

Key1 ,

Key2

}

Now in the de structuring so we will pass directly

{

Key1 ,

Key2

}

As attributes

So finally it will look like

const RestaurantCard = ({resName, cousins,rating, cost,time, image}) =>{}

This is a feature of js

Const abc = (props) =>{}

Const {resName, cousin} = props;

This above code is correct

Now we can replace like

Const abd = ({resname, cousin}) =>{}

Both are same.

Config-driven UI.

Different design based on different configuration

Like may have different layout in different countries.

How to print array comma separated inline

arrayName.join(“, “);

?. -> search for this.

Const abcd = data?.subdata;

Now in place of data.subdata.abcd

I can use abcd directly.

Now take some dynamic data and use this as array and also we can make dynamic card by map function.

Simply take list of data and appy map method on it and pass one by one the card value to make cards Dynamic.

Final code

import React from 'react';

import ReactDOM from 'react-dom/client';

const restaurantData = {

"data": [

{

"image": "https://media-assets.swiggy.com/swiggy/image/upload/fl\_lossy,f\_auto,q\_auto,w\_660/xvyrclhxftulsglktaek",

"resName": "Restaurant One",

"cousins": "Italian, Mediterranean",

"rating": "★★★★☆",

"cost": 25,

"time": "20 mins"

},

{

"image": "https://media-assets.swiggy.com/swiggy/image/upload/fl\_lossy,f\_auto,q\_auto,w\_660/xvyrclhxftulsglktaek",

"resName": "Restaurant Two",

"cousins": "Chinese, Thai",

"rating": "★★★☆☆",

"cost": 15,

"time": "25 mins"

}

]

};

const Heading = () =>{

return (

<header className="header">

<div className="logo">MyLogo</div>

<nav className="navbar">

<a href="#" className="nav-link">Home</a>

<a href="#" className="nav-link">About</a>

<a href="#" className="nav-link">Services</a>

<a href="#" className="nav-link">Contact</a>

</nav>

</header>

)

}

// const RestaurantCard = ({resName,cousins,rating,cost,time,image}) =>{

const RestaurantCard = (props) => {

// console.log(props.resData.resName);

props = props?.resData;

return (

<div className="card-container">

<div className="card">

<img src={props.image} alt="Card Image" className="card-image" />

<div className="card-content">

<h2 className="card-title">{props.resName}</h2>

<p className="card-description">{props.cousins}</p>

<div className="card-details">

<span className="card-rating">Rating: {props.rating}</span>

<span className="card-cost">Cost: ${props.cost}</span>

<span className="card-time">Time: {props.time}</span>

</div>

</div>

</div>

</div>

);

}

const AppLayout = () =>{

return (

<div>

<Heading />

<main className="main-content">

<div className="search-bar">

<input type="text" placeholder="Search..." />

<button type="button">Search</button>

</div>

<div className='res-container'>

{restaurantData.data.map((res, index) => (

<RestaurantCard

key={index}

resData = {res}

/>

))}

{/\* <RestaurantCard

resName = "Meghna Foods"

cousins = "Biryani, North Indian, South Indian"

rating = "★★★★☆"

cost = "20"

time = "30 minutes"

image = 'https://media-assets.swiggy.com/swiggy/image/upload/fl\_lossy,f\_auto,q\_auto,w\_660/xvyrclhxftulsglktaek'

/>

<RestaurantCard

resName = "KFC"

cousins = "Burger, Pizza"

rating = "★★★☆☆"

cost = "10"

time = "20 minutes"

image = 'https://media-assets.swiggy.com/swiggy/image/upload/fl\_lossy,f\_auto,q\_auto,w\_660/xvyrclhxftulsglktaek'

/> \*/}

</div>

</main>

<footer className="footer">

<div className="footer-links">

<a href="#" className="footer-link">Privacy Policy</a>

<a href="#" className="footer-link">Terms of Service</a>

</div>

<p className="footer-copyright">&copy; 2024 MyWebsite. All rights reserved.</p>

</footer>

</div>

)

}

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(<AppLayout />);

If we not add key in listing in react it will give error

Each child in a list should have a unique key.

What is this?

As we know react works on components

We are using same component and elements for different things multiple times.

So how react will get which data we are going to use here?

For that purpose react make a rule

For any kind of listing we have to pass unique key with fixed keyword Key.

Each key should be unique on the page.

Example:

{restaurantData.data.map((res, index) => (

<RestaurantCard

key={index}

resData = {res}

/>

))}

The index is the parameter in map. We can use it if noting is unique there.

If supposed key is not given the it will render all the data on the same level.

When we give unique id then it will check react check other keys and wherever it will get difference it will render the data.

When we apply map functionality on the array

Its first argument is each element, 2nd is index of the element,

React says

We DO NOT recommend to USE INDEX AS KEY.

[robinpokorny.com/blog/index-as-a-key-is-an-anti-pattern](http://robinpokorny.com/blog/index-as-a-key-is-an-anti-pattern)

Search and read the article.