Chapter 04

Talk is Cheap Show me the code

Page No.
Date

→ Building a Food-ordering App

* Is JSX mandatory ⇒ No
* Is Typescript mandatory ⇒ No
* Is ES6 mandatory ⇒ No

3 ways of component composition:-

1. {title()}
2. <Title /> ← used generally
3. <Title></Title>

* JSX expression must have one parent element.

React.Fragment

→ is a component which is exported by React.
[import React from "react"]

→ Group list of Children without adding extra nodes to the DOM.

→ Shorthand syntax <></> is used instead of <React.Fragment></React.Fragment>

Page No. _____
Date

* But you can't pass styles to empty brackets.

Giving style inside React

To give inline styles in react do :-

First Method

```
const styleObj = {
    backgroundColor: "red"
};
```

styleObj is Normal JS Object.

```
const jsx = (
    <div style = {styleObj}>
        <h1> Hello </h1>
        <h2> World </h2>
    </div>
);
```

inside this parenthesis you can write any piece of JS code

→ In React, style is given using JS object.

Alternative way :-

```
const jsx = (
    <div style = {{
        backgroundColor: "yellow";
    }}>
        <h1> Hello </h1>
        <h2> World </h2>
    </div>
);
```

Page No.
Date

Second method

- Give class name to the div (or whatever tag) & write CSS inside CSS file

* const jsx = (
```
<div classname = "JSX">
<h1> Hello </h1>
<h2> World </h2>
</div>);
```

CSS file ↓

```
JSX {
backgroundColor: "blue"
}
```

Third Method

- using external library like TailwindCSS, Bootstrap, materialUI etc.

HW

Can I use a 'React.fragment' inside my 'React.fragment'?

A) Yes, you can nest 'React.fragment' component inside other 'React.fragment' components.

Eg:-

```
import React from 'react';
const jsx = (
  <>
    <ChildA />
    <>
      <ChildB />
      <ChildC />
    </>
    <ChildD />
  </>
);
```

ChildB & C are siblings & will be grouped together without adding an extra node to the DOM.

Config Driven UI

All the UI (let say, swiggy, home page) is driven by a config which is send by backend (api)

Optional chaining

- Allows us to access an object properties without having to check if the object or its properties exist
- represented by ?
- new feature introduced in js ES2020

So after taking real data from swiggy our "Restaurant card" function looks like this

PROPS

- Shorthand for properties
- "passed props" mean I'm passing some data or properties into my functional or class component.
- rcstaurant = {restaurantList[3]}
  This means react wraps up all these properties into a variable known as 'props' I can call it anything

→ So our Restaurant card function will now look like ↓

```
* const RestaurantCard = (props) => {
    return (
      <div className = "card">
        <h1> {props.restaurant.data.name} </h1>
      </div>
    );
  };
```

→ Object Destructuring

```
(props) => ({restaurant})
  (restaurant.data.?name)
```

This has all the props like name, cuisines, time etc & I may want to pass all of these props so instead of writing each prop individually I will do: ↓

```
{...restaurantList[1].data}
```

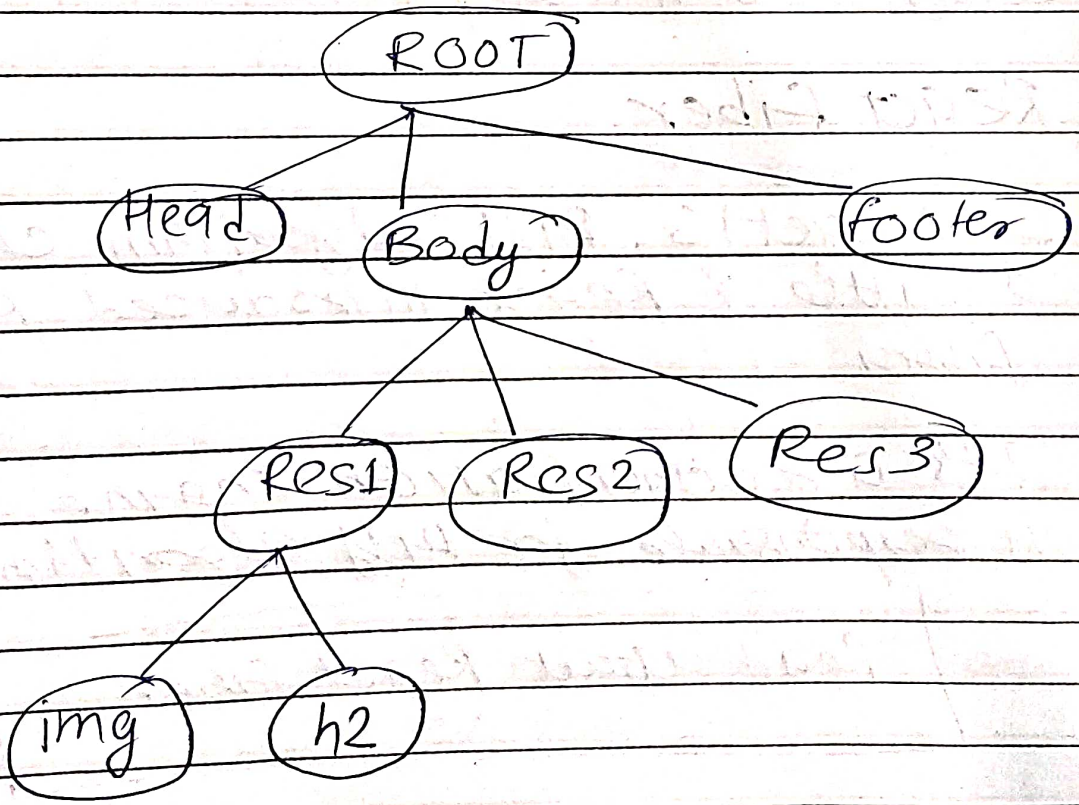
Page No.
Date

Virtual DOM

Let the structure of our DOM look like

```
<head>
<body>
<Rest 1>
<Rest 2> <img ... />
<Rest 3>
</body>
```

⇒ We keep a representation DOM with us, which is known as virtual DOM.



Page No.
Date

→ We need virtual DOM for Reconciliation

→ Reconciliation is an algorithm that react uses to diff one tree from other.

→ It use Diff Algorithm & in determination what need to change & what does not in UI.

[To find out Difference between one tree (Actual DOM) & other (Virtual DOM)]

→ Diff Algorithm then finds out what need to be updated & it change only that small portion.

React fiber

→ In React 16, Diff algorithm changed a little & React introduced React fibres

→ It's a reconciliation engine which is responsible for Diff algorithm.

[Read about React fiber]