Lec 4

Everything is hard coded

Step 1:

```jsx
const AppLayout = () => {
  return(
    <div className="app">
      <Header/>
      <Body/>
    </div>
  );
};

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

root.render(<AppLayout/>)
```

Step 2:

```jsx
const Header = () => {
  return (
    <div className="header">
      <div className="logo-container">
        <img
        className="logo"
        src="https://img.freepik.com/premium-vector/good-food-logo-template_79169-17.jpg?w=740"
        />
      </div>
      <div className="nav-items">
        <ul>
          <li>Home</li>
          <li>About Us</li>
          <li>Contact Us</li>
          <li>Cart</li>
        </ul>
      </div>
    </div>
  );
};
```

Step 3:

```jsx
const Body = () => {
  return(
    <div className="body">
      <div className="search">Search</div>
      <div className="res-container">
        <RestaurantCard/>
        <RestaurantCard/>
      </div>
    </div>
  );
};
```

Step 4:

```
const RestaurantCard = () => {
  return (
    //<div className="res-card" style={styleCard}>
     <div className="res-card" style={{backgroundColor: "grey"}}>
      <img
        className="res-logo"
        alt="res-logo"
        src="https://img.freepik.com/premium-vector/good-food-logo-template_79169-17.jpg?w=740"
      />
      <h3>Meghana Foods</h3>
      <h4>Biryani, North Indian, Asian</h4>
      <h4>4.4 Stars</h4>
      <h4>38 Minutes</h4>
    </div>
  );
};
```

Passing props to the component
Step 1:

```
const Body = () => {
  return(
    <div className="body">
      <div className="search">Search</div>
      <div className="res-container">
        <RestaurantCard
          resName="Meghana Foods"
          cuisine="Biryani, North Indian, Asian"
        />
        <RestaurantCard
          resName="KFC"
          cuisine="Burger, Fast Food"
        />
      </div>
    </div>
  );
};
```

Step 2:

```jsx
const RestaurantCard = (props) => {
  console.log(props);
  return (
    <div className="res-card" style={styleCard}>
      <img
        className="res-logo"
        alt="res-logo"
        src="https://img.freepik.com/premium-vector/good-food-logo-template_79169-17.jpg?w=740"
      />
      <h3>{props.resName}</h3>
      <h4>{props.cuisine}</h4>
    </div>
  );
};
```

Destructuring object

```jsx
const RestaurantCard = ({resName, cuisine}) => {
  console.log(props);
  return (
    <div className="res-card" style={styleCard}>
      <img
        className="res-logo"
        alt="res-logo"
        src="https://img.freepik.com/premium-vector/good-food-logo-template_79169-17.jpg?w=740"
      />
      <h3>{resName}</h3>
      <h4>{cuisine}</h4>
    </div>
  );
};
```

OR

```jsx
const RestaurantCard = ({props}) => {
  const {resName, cuisine} = props;
  return (
    <div className="res-card" style={styleCard}>
      <img
        className="res-logo"
        alt="res-logo"
        src="https://img.freepik.com/premium-vector/good-food-logo-template_79169-17.jpg?w=740"
      />
      <h3>{resName}</h3>
      <h4>{cuisine}</h4>
    </div>
  );
};
```

Passing value from API

```
const restaurantList = {
  type: "restaurant",
  data: {
    name: "Domino's Pizza",
    cuisines: ["Pizzas"],
    avgRating: "4.0",
    costForTwo: 40000,
    deliveryTime: 45,
  }
}
```

Step 1

```
const Body = () => {
  return(
    <div className="body">
      <div className="search">Search</div>
      <div className="res-container">
        <RestaurantCard
          resData={restaurantList}
        />
      </div>
    </div>
  );
```

Step 2

```
const RestaurantCard = (props) => {
  const {resData} = props;
  return (
    <div className="res-card" style={styleCard}>
      <img
        className="res-logo"
        alt="res-logo"
        src={
          "https://res.cloudinary.com/swiggy/image/upload/fl_lossy,f_auto,q_auto,w_508,h_320,c_fill/" +
          resData.data.cloudinaryImageId
        }
      />
      <h3>{resData.data.name}</h3>
      <h4>{resData.data.cuisines.join(",")}</h4>
      <h4>{resData.data.avgRating} stars</h4>
      <h4>{resData.data.costForTwo} / 100</h4>
      <h4>{resData.data.deliveryTime} minutes</h4>
    </div>
  );
};
```

```
<h4>{resData.data.cuisines.join(",")}</h4>
```

Join words with ,

If data is coming in array

Step1

```
const Body = () => {
  return(
    <div className="body">
      <div className="search">Search</div>
      <div className="res-container">
        <RestaurantCard
          resData={restaurantList[0]}
        />
        <RestaurantCard
          resData={restaurantList[1]}
        />
      </div>
    </div>
  );
};
```

Step 2 will be same

Clean & good practice

```
const RestaurantCard = (props) => {
  const {resData} = props;
  const {
    cloudinaryImageId,
    name,
    cuisines,
    avgRating,
    costForTwo,
    deliveryTime
  } = resData?.data;
  return (
    <div className="res-card" style={styleCard}>
      <img
        className="res-logo"
        alt="res-logo"
        src={
          "https://res.cloudinary.com/swiggy/image/upload/fl_lossy,f_auto,q_auto,w_508,h_320,c_fill/" +
          cloudinaryImageId
        }
      />
      <h3>{name}</h3>
      <h4>{cuisines.join(",")}</h4>
      <h4>{avgRating} stars</h4>
      <h4>{costForTwo} / 100</h4>
      <h4>{deliveryTime} minutes</h4>
    </div>
  );
};
```

Looping

```
const Body = () => {
  return(
    <div className="body">
      <div className="search">Search</div>
      <div className="res-container">
        {
          restaurantList.map((restaurant) => (
          <RestaurantCard resData={restaurant}/>
        ))}
      </div>
    </div>
  );
};
```

Looping with key

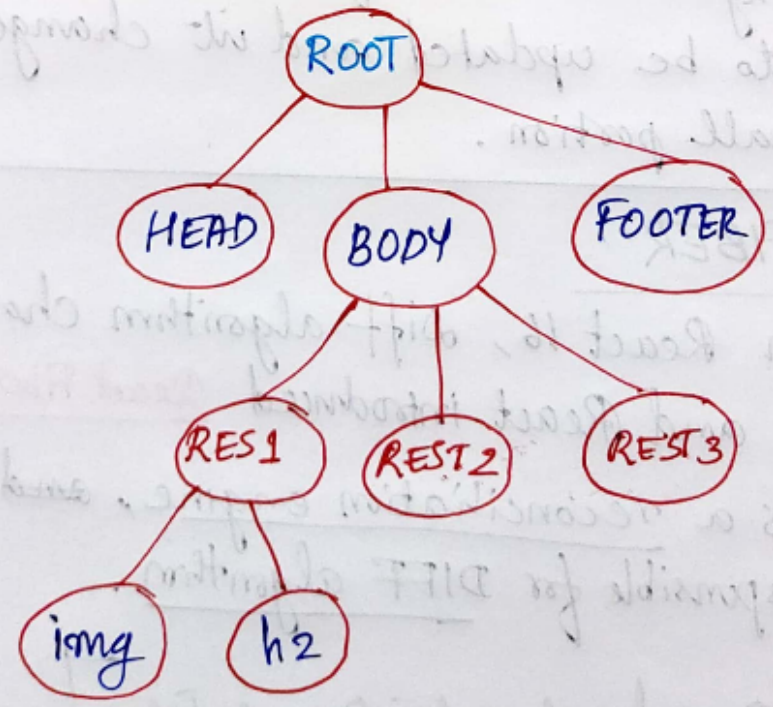
```jsx
const Body = () => {
  return(
    <div className="body">
      <div className="search">Search</div>
      <div className="res-container">
        {
          restaurantList.map((restaurant) => (
          <RestaurantCard key={restaurant.data.id}resData={restaurant}/>
        ))}
      </div>
    </div>
  );
};
```

VIRTUAL DOM

→ Let the structure of our DOM look like ↓

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

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



→ We need Virtual DOM for Reconciliation

* → Reconciliation is an algorithm that React uses to diff one tree from other. It uses Diff Algorithm and in determines what needs to change and what does not in UI.

[To find out DIFFERENCE between one tree (Actual DOM) and other (VIRTUAL DOM)]

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

REACT FIBER

→ In React 16, Diff algorithm changed a little and React introduced React Fiber

→ It's a reconciliation engine, ~~and~~ which is responsible for DIFF algorithm.

[Read about React fiber]