Chapter 3: Components and Props

1 Introduction

Is chapter mein humne **React components** aur **props** ke baare mein seekha. Components React ke building blocks hote hain, aur props ka use data pass karne ke liye hota hai taaki components dynamic aur reusable ho sakein.

2 3.1 What are Components?

Component ek function ya class hota hai jo JSX return karta hai aur UI ka ek hissa represent karta hai. Humne **functional components** pe focus kiya kyunki yeh modern aur asaan hain.

2.1 Code Example: Functional Component

```
function WelcomeMessage() {
  return (
    <div className="text-center p-4">
      <h1 className="text-2xl font-bold text-green-600">Welcome to Our Restaurant!</h1>
      <p className="text-lg">Enjoy the best food in town!</p>
    </div>
  );
}
```

2.2 Explanation

- Yeh ek **functional component** hai jo JSX return karta hai.
- `className` se Tailwind CSS ke saath styling ki.
- Component reusable hai, kisi bhi app mein use kar sakte hain.

2.3 Real-World Example

Ek food delivery app mein `WelcomeMessage` component har page pe use ho sakta hai.

3 3.2 Using Components in Other Components

Components ko ek doosre ke andar use kar sakte hain, jaise lego blocks, taaki app modular ho.

3.1 Code Example: Nested Components

```
function MenuItem() {
  return (
    <div className="border p-4 m-2 rounded-lg">
      <h2 className="text-xl font-bold">Samosa</h2>
      <p>Price: Rs 20</p>
    </div>
  );
}

function App() {
  return (
    <div className="text-center p-4">
      <h1 className="text-3xl font-bold text-blue-600">Our
         Menu</h1>
      <MenuItem />
      <MenuItem />
    </div>
  );
}
```

3.2 Output

- Heading: **Our Menu**
- Do cards: **Samosa, Price: Rs 20**

3.3 Explanation

- `MenuItem` ek component hai jo ek menu item dikhata hai.
- `App` ne `MenuItem` ko do baar call kiya.
- `<MenuItem />` syntax component ko call karta hai.

4 3.3 What are Props?

Props components ke liye input hote hain, jisse data pass karte hain taaki component dynamic ho.

4.1 Code Example: Using Props

```
function MenuItem(props) {
  return (
    <div className="border p-4 m-2 rounded-lg">
      <h2 className="text-xl font-bold">{props.itemName}</h2>
      <p>Price: Rs {props.price}</p>
    </div>
  );
}

function App() {
  return (
    <div className="text-center p-4">
      <h1 className="text-3xl font-bold text-blue-600">Our
         Menu</h1>
      <MenuItem itemName="Chai" price={10} />
      <MenuItem itemName="Samosa" price={20} />
      <MenuItem itemName="Idli" price={30} />
    </div>
  );
}
```

4.2 Output

- Heading: **Our Menu**
- Card 1: **Chai, Price: Rs 10**
- Card 2: **Samosa, Price: Rs 20**
- Card 3: **Idli, Price: Rs 30**

4.3 Explanation

- `props` ek object hai jo `itemName` aur `price` rakhta hai.
- `{props.itemName}` aur `{props.price}` se props ka data JSX mein use hota hai.

4.4 Real-World Example

Ek restaurant menu mein har item ka naam aur price alag hota hai. Props ke sa2344ath ek hi `MenuItem` component ko alag-alag data ke saath use kar sakte hain.

5 3.4 Destructuring Props

Props ko function parameter mein destructure kar sakte hain taaki code chhota aur readable ho.

5.1 Code Example: Destructuring Props

```
function MenuItem({ itemName, price }) {
  return (
    <div className="border p-4 m-2 rounded-lg">
      <h2 className="text-xl font-bold">{itemName}</h2>
      <p>Price: Rs {price}</p>
    </div>
  );
}

function App() {
  return (
    <div className="text-center p-4">
      <h1 className="text-3xl font-bold text-blue-600">Our
         Menu</h1>
      <MenuItem itemName="Chai" price={10} />
      <MenuItem itemName="Samosa" price={20} />
    </div>
  );
}
```

5.2 Explanation

- `{ itemName, price }` se props ko directly destructure kiya.
- `props.itemName` likhne ki zaroorat nahi, code clean ho gaya.

6 Common Mistakes

- **Props ka Naam Galat**: `<MenuItem name="Chai" />` likha aur `props.itemName` use kiya toh undefined milega.
- **Forgetting Parent Element**: Ek hi parent element hona chahiye.
- **Static Components**: Bina props ke component reusable nahi hota.

7 Interview Tips

- **Components kya hote hain?**
  - React ke reusable UI blocks, jaise functional ya class components.
  - Functional components modern hain.
- **Props kya hain?**
  - Data pass karne ka tareeka, read-only hote hain.

8 Assignment: Practice Time

8.1 Task 1: Create a Profile Component

- `Profile` component banao jo `name` aur `age` props le.
- Card mein name aur age dikhaye, Tailwind CSS se styling karo.
- `App` mein 2 baar call karo alag-alag data ke saath.

8.2 Task 2: Nested Components

- `ItemCard` component banao jo `itemName` aur `price` props le.
- `Menu` component banao jo 3 `ItemCard` call kare.
- Heading add karo, jaise "Todays Special".

8.3 Task 3: Destructuring Props

- `Product` component banao jo `productName`, `price`, aur `category` props le.
- Props ko destructure karo.
- Card banao aur `App` mein 2 products ke liye call karo.