📘React.js – Week 1, Day 4

Topic: Default Props & PropTypes

---

1. Why Do We Need Default Props?

- When you build a component, you usually pass **props** from a parent component.
- But sometimes, a parent might forget to pass a prop → this can break your UI.
- **Default props** solve this problem by giving a **fallback value** if no prop is provided.

👉Think of it like saying: *"If no one tells me your name, I'll just call you Guest."*

---

2. Setting Default Props

There are **two ways** to set default props in React:

(a) Inside the component using `defaultProps`

```
function ProfileCard({ name, age, bio, isStudent }) {
  return (
    <>
      <h2>Welcome {name}! Your age is {age}</h2>
      <h3>Bio: {bio}</h3>
      <h4>{isStudent ? "You are a Student" : "You are Graduated"}</h4>
    </>
  );
}

// 👈🏼 Adding default values
ProfileCard.defaultProps = {
  name: "Guest",
  age: 18,
  bio: "No bio provided",
  isStudent: false
};

export default ProfileCard;
```

👉If you now render `<ProfileCard />` without passing any props, React will automatically use these defaults.

(b) ES6 Destructuring with Defaults (modern way)

```
function ProfileCard({ name = "Guest", age = 18, bio = "No bio", isStudent =
false }) {
  return (
    <>
      <h2>Welcome {name}! Your age is {age}</h2>
      <h3>Bio: {bio}</h3>
      <h4>{isStudent ? "You are a Student" : "You are Graduated"}</h4>
    </>
  );
}
```

👉This is **preferred in modern React** because it keeps everything inside the function signature.

---

3. Why Do We Need PropTypes?

- React is **not strongly typed** (unlike TypeScript).
- You could accidentally pass a **string** where a **number** is expected → leading to **bugs**.
- **PropTypes** is a way to add **runtime type-checking** for your props.

👉Think of it like saying: *"This box only accepts numbers, don't put text in it."*

---

4. Using PropTypes

(a) Install PropTypes

```
npm install prop-types
```

(b) Import and Define

```
import PropTypes from "prop-types";

function ProfileCard({ name, age, bio, isStudent }) {
  return (
    <>
      <h2>Welcome {name}! Your age is {age}</h2>
      <h3>Bio: {bio}</h3>
      <h4>{isStudent ? "You are a Student" : "You are Graduated"}</h4>
    </>
  );
}
```

```
// 👈 Define prop types
ProfileCard.propTypes = {
  name: PropTypes.string,     // must be string
  age: PropTypes.number,      // must be number
  bio: PropTypes.string,      // must be string
  isStudent: PropTypes.bool   // must be true/false
};

// 👈 Default props for safety
ProfileCard.defaultProps = {
  name: "Guest",
  age: 18,
  bio: "No bio provided",
  isStudent: false
};

export default ProfileCard;
```

---

5. Common PropTypes Options

- `PropTypes.string` → string only
- `PropTypes.number` → numbers only
- `PropTypes.bool` → true/false
- `PropTypes.array` → array
- `PropTypes.object` → object
- `PropTypes.func` → function
- `PropTypes.node` → anything renderable (string, number, JSX)
- `PropTypes.element` → must be a React element
- `PropTypes.oneOf(["a", "b"])` → restrict values to specific choices
- `PropTypes.arrayOf(PropTypes.number)` → array of numbers only
- `PropTypes.shape({ key: PropTypes.string })` → object with specific shape

---

6. Example in Action

```
<ProfileCard />
// Output:
// Welcome Guest! Your age is 18
// Bio: No bio provided
// You are Graduated

<ProfileCard name="Rehan" age="twenty one" />
```

```
// 👉🏿 Console warning: Invalid prop `age` of type `string` supplied to
`ProfileCard`, expected `number`.
```

👉You still see something rendered, but React warns you in the console → **helps catch bugs early!**

🔗Day 4 Task (15–20 min)

1. Create a new component called `ProductCard.js`
2. Props: `title`, `price`, `inStock`, `tags`
3. Use **default props** so if any value is missing, safe defaults appear.
4. Use **PropTypes** to ensure:
5. `title` must be string
6. `price` must be number
7. `inStock` must be bool
8. `tags` must be an array of strings
9. Render **3 ProductCards** in `App.js`:
10. One with all props provided
11. One missing some props (to test defaults)
12. One with wrong prop type (to see console warning)

👈🏼After this task, you'll understand how **default props and PropTypes** protect your components from bugs and missing data.