

# WEB APP DEVELOPMENT WITH REACTJS (INT252)

## Lecture 8: Component Composition & Conditional Rendering

### Unit II – Component Design & Styling

#### Syllabus Mapping:

- Component composition
- Conditional rendering in React
- UI control using logic

Target Learner: Beginner (HTML, CSS, JavaScript knowledge)

# Why This Lecture Is Important

Real applications:

- Show or hide UI
- Change layout based on data
- Combine many components together

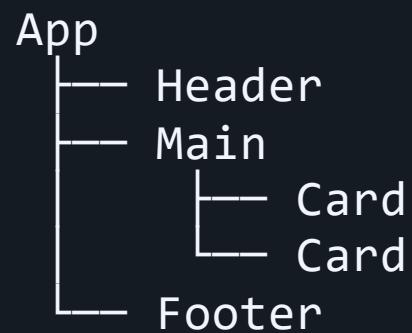
This lecture teaches **UI control logic**.

# What is Component Composition?

Component composition means:

- Using components **inside other components**
- Building complex UI from simple pieces

# Composition Example (Visual)



# Simple Composition Example

```
function Header() {
  return <h2>Header</h2>;
}

function Footer() {
  return <h2>Footer</h2>;
}

function App() {
  return (
    <>
      <Header />
      <Footer />
    </>
  );
}
```

# Why Composition Is Better Than Big Components

- ✗ One huge component
- ✓ Many small components

Benefits:

- Reusability
- Readability
- Easy maintenance

## Problem: Static UI

UI that never changes 

Real apps need:

- Dynamic UI
- Conditional display

# What is Conditional Rendering?

Conditional rendering means:

- Showing different UI
- Based on condition (true/false)

# Conditional Rendering (Concept)

Condition true → Show UI A  
Condition false → Show UI B

## Method 1: if Statement (Outside JSX)

```
function App() {  
  const isLoggedIn = true;  
  
  if (isLoggedIn) {  
    return <h1>Welcome User</h1>;  
  }  
  
  return <h1>Please Login</h1>;  
}
```

# Why `if` Cannot Be Used Inside JSX

JSX allows:

- Expressions ✅ statements

`if` is a statement

## Method 2: Ternary Operator (Most Used)

```
{  
  isLoggedIn  
  ? <h1>Welcome</h1>  
  : <h1>Login</h1>  
}
```

# Ternary Inside Component

```
function App() {  
  const isLoggedIn = false;  
  
  return (  
    <h1>  
      {isLoggedIn ? 'Welcome' : 'Login'}  
    </h1>  
  );  
}
```

## Method 3: Logical AND (`&&`)

```
{isAdmin && <p>Admin Panel</p>}
```

Only shows when true

## When to Use `&&`

Use when:

- Only want to show something
- No else condition

# Conditional Rendering with Components

```
function Message({ isLoggedIn }) {  
  return isLoggedIn ? <Dashboard /> : <Login />;  
}
```

## Real Example: Login UI

```
function App() {  
  const loggedIn = true;  
  
  return (  
    <>  
      {loggedIn && <h2>Dashboard</h2>}  
      {!loggedIn && <h2>Login Page</h2>}  
    </>  
  );  
}
```

## Common Beginner Mistakes

- ✗ Using if inside JSX
- ✗ Returning undefined
- ✗ Forgetting else condition

# Visual Flow: Conditional Rendering

State / Props



Condition Check



Correct UI Renders

## Practice Exercises

1. Show "Welcome" if age > 18
2. Hide message when condition false
3. Toggle component using boolean



## Answers – Practice Exercises

### 1. Age Check

```
{age > 18 ? <h1>Welcome</h1> : <h1>Not Allowed</h1>}
```



## Answers – Continued

### 2. Hide Message

```
{show && <p>Hello</p>}
```



## Answers – Continued

### 3. Toggle Component

```
{isOn ? <On /> : <Off />}
```

## Key Takeaways

- Composition builds complex UI
- Conditional rendering controls UI
- JSX supports expressions only
- `&&` and ternary are core tools

## Next Lecture

Lecture 9: Styling in React (Tailwind CSS)

Unit II – Component Design & Styling