

## 2.React-HOL

### 1. Explain React components

React components are the building blocks of a React application. They let you split the UI into independent, reusable pieces that can be managed separately.

Each component:

- Accepts inputs (called props)
- Returns React elements (UI)
- Can be a function or a class

### 2. Identify the differences between components and JavaScript functions

Feature	React Components	JavaScript Functions
Purpose	Build reusable UI blocks	Perform a specific task
Return Value	JSX (React elements)	Any data (number, string, etc.)
React Integration	Can use state, props, lifecycle hooks	Cannot natively use React features
Component Types	Functional or Class	Only function type
Rendering	Part of React's rendering system	Not rendered to UI by default

### 3. Identify the types of components

React components are mainly of **two types**:

#### 1. Class Components

- Uses ES6 classes
- Can hold state and use lifecycle methods
- Requires a render() method

#### 2. Function Components

- Uses JavaScript functions
- Can use **React Hooks** (e.g., useState, useEffect) to manage state and lifecycle
- Simpler and more modern approach

### 4. Explain Class Component

A **Class Component** is a React component defined using a JavaScript class. It extends React.Component and must define a render() method which returns JSX.

```
import React, { Component } from 'react';

class Home extends Component {

  render() {

    return <h1>Welcome to the Home page</h1>;

  }

}
```

### Key features:

- Can use state (this.state)
- Can use lifecycle methods (componentDidMount, etc.)
- More boilerplate code than function components

## 5. Explain Function Component

A **Function Component** is a simpler way to write components using JavaScript functions. It returns JSX and can use React Hooks to manage state and side effects.

```
function About() {  
  return <h1>Welcome to the About page</h1>;  
}
```

### Key features:

- Lightweight and concise
- Uses Hooks for state and lifecycle
- Now preferred over class components

## 6. Define Component Constructor

In class components, the **constructor** is a special function used to:

- Initialize the component's state
- Bind event handlers (if needed)
- Set up props using super(props)

```
constructor(props) {  
  super(props);  
  this.state = { message: 'Hello' };  
}
```

## 7. Define render() function

The render() method is **mandatory** in class components. It returns the JSX that defines the UI of the component.

```
render() {  
  return <h1>Hello from render method!</h1>;  
}
```

- Called during mounting and updating phase
- Must return one parent JSX element

## HANDS-ON

### CODE: Home.js

```
import React, { Component } from 'react';

class Home extends Component {

  render() {

    return (

      <div>

        <h2>Welcome to the Home page of Student Management Portal</h2>

      </div>

    );

  }

}

export default Home;
```

### Contact.js

```
import React, { Component } from 'react';

class Contact extends Component {

  render() {

    return (

      <div>

        <h2>Welcome to the Contact page of the Student Management Portal</h2>

      </div>

    );

  }

}

export default Contact;
```

### About.js

```
import React, { Component } from 'react';

class About extends Component {

  render() {

    return (

      <div>

        <h2>Welcome to the About page of the Student Management Portal</h2>

      </div>

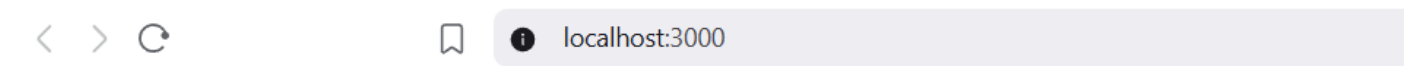
    );

  }

}
```

```
}  
}  
export default About;  
  
App.js  
import React from 'react';  
import Home from './Components/Home';  
import About from './Components/About';  
import Contact from './Components/Contact';  
function App() {  
  return (  
    <div className="App">  
      <Home />  
      <About />  
      <Contact />  
    </div>  
  );  
}  
export default App;
```

## OUTPUT



**Welcome to the Home page of Student Management Portal**

**Welcome to the About page of the Student Management Portal**

**Welcome to the Contact page of the Student Management Portal**