## **Objectives**

* Explain React components

React components are the building blocks of any React application. They allow you to split the UI into independent, reusable pieces, and think about each piece in isolation. A component returns a portion of JSX (JavaScript XML) code that defines what should be rendered on the screen.

* Identify the differences between components and JavaScript functions

|  |  |  |
| --- | --- | --- |
| Feature | JavaScript Function | React Component |
| Purpose | Used for general logic or calculations | Used to build UI elements |
| Return Value | Returns data or values | Returns JSX (UI code) |
| Lifecycle Methods | Not applicable | Can use lifecycle methods (in class components) |
| JSX Support | Not inherently supported | Designed to return and handle JSX |
| React Integration | Not automatically recognized by React | Recognized by React and used in rendering |

* Identify the types of components

Types of Components:

1. Class Components
2. Function Components

* Explain class component

**Class Component:**

A class component is a JavaScript class that extends React.Component. It can hold and manage its own state and can use lifecycle methods such as componentDidMount, componentDidUpdate, and componentWillUnmount.

**Example:**

import React, { Component } from 'react';

class Welcome extends Component {

render() {

return <h1>Hello, {this.props.name}</h1>;

}

}

* Explain function component

**Function Component:**

A function component is a simpler way to write components. It is just a plain JavaScript function that accepts props as an argument and returns JSX. With the introduction of Hooks (like useState and useEffect), function components can now use state and other React features.

**Example:**

import React from 'react';

function Welcome(props) {

return <h1>Hello, {props.name}</h1>;

}

* Define component constructor

The constructor is a special method used in class components for initializing state and binding methods. It is called before the component is mounted.

**Example:**

constructor(props) {

super(props);

this.state = { count: 0 };

}

* Define render() function

The render() function is required in every class component. It returns the JSX that defines what should be displayed on the screen. It is automatically called during the component lifecycle whenever the component needs to re-render.

**Example:**

render() {

return <div>{this.state.count}</div>;

}

## **Notes**

Estimated time to complete this lab: **30 minutes.**

Create a react app for Student Management Portal named scorecalculatorapp and create a function component named “CalculateScore” which will accept Name, School, Total and goal in order to calculate the average score of a student and display the same.

1. Create a React project named “scorecalculatorapp” type the following command in terminal of Visual studio:



1. Create a new folder under Src folder with the name “Components”. Add a new file named “CalculateScore.js”
2. Type the following code in CalculateScore.js





1. Create a Folder named Stylesheets and add a file named “mystyle.css” in order to add some styles to the components:



1. Edit the App.js to invoke the CalculateScore functional component as follows:



1. In command Prompt, navigate into scorecalculatorapp and execute the code by typing the following command:



1. Open browser and type “localhost:3000” in the address bar:



HANDS-ON:





