**React HOL-3**

**Explain React Components**

**React components** are the building blocks of a React application. Each component represents a part of the user interface (UI). Components let you split the UI into independent, reusable pieces.

* Components **accept inputs** called **props**
* They **return elements** that describe what should appear on the screen
* Components can be either **class-based** or **function-based**

**Identify the differences between components and JavaScript functions**

|  |  |  |
| --- | --- | --- |
| **Feature** | **React Components** | **JavaScript Functions** |
| Purpose | Build and render UI | Perform logic or computation |
| Return Value | Returns JSX (UI elements) | Returns a value (number, string, etc.) |
| Used In | React environment | Any JavaScript environment |
| Lifecycle Methods | Only in class components | Not applicable |
| React Features (e.g., Hooks) | Available in function components | Not available |

**Identify the types of components**

There are mainly two types of React components:

1. **Class Components**
2. **Function Components**

Additionally, components can be categorized as:

* **Presentational (Stateless)**: Focused on how things look
* **Container (Stateful)**: Focused on how things work (logic and data handling)

**Explain class component**

A **Class Component** is a React component defined using an ES6 JavaScript class that extends React.Component.

#### **Key Features:**

* Has access to **lifecycle methods** (e.g., componentDidMount)
* Uses this.state for internal state management
* Uses render() method to return JSX

**Explain function component**

A **Function Component** is a simpler way to write components using JavaScript functions.

**Key Features:**

* Easier to read and write
* Can use **React Hooks** (like useState, useEffect) for state and lifecycle logic
* More concise than class components

**Define Component Constructor**

In a **class component**, the constructor() method is a special function used to:

* Initialize **state**
* Bind **event handlers** to the class context

**Define render() function**

The render() function is required in class components. It tells React what to display on the screen.

* The render() method **returns JSX**
* React calls render() **whenever the component’s state or props change**

**App.jsx**

import { CalculateScore } from './Components/CalculateScore';

import './App.css';

function App() {

return (

<div>

<CalculateScore Name={"Steeve"}

School = {"DNV Public School"}

total = {284}

goal = {3}></CalculateScore>

</div>

);

}

export default App;

**CalculateScore.jsx**

import '../Stylesheets/mystyles.css'

const percentToDecimal = (decimal)=>{

return (decimal.toFixed(2)+'%')

}

const calcScore = (total,goal)=>{

return percentToDecimal(total/goal)

}

export const CalculateScore = ({Name,School,total,goal})=>(

<div className='formatstyle'>

<h1><font color='Brown'> Student Details</font></h1>

<div className="Name">

<b><span>Name: </span></b>

<span>{Name}</span>

</div>

<div className="School">

<b><span>School: </span></b>

<span>{School}</span>

</div>

<div className="Total">

<b><span>Total: </span></b>

<span>{total}</span>

<span>Marks</span>

</div>

<div className="Score">

<b>Score: </b>

<span>

{calcScore(total,goal)}

</span>

</div>

</div>

)

**Mystyles.css**

.Name{

font-weight: 300;

color: blue;

}

.School{

color: crimson;

}

.Total{

color: darkmagenta;

}

.formatstyle{

text-align: center;

font-size: large;

}

.Score{

color: forestgreen;

}

**Output**

**A screen shot of a student details

AI-generated content may be incorrect.**