**3. ReactJS-HOL**

**Objectives:**

**1. Explain React components**  
React components are the building blocks of a React application. A component is a reusable piece of UI that can have its own logic and appearance. Components can be either class-based or function-based and help in creating modular and maintainable code.

**2. Identify the differences between components and JavaScript functions**  
While both React components and JavaScript functions can take inputs and return outputs, React components return **JSX** (React's syntax for UI), and they can manage state and lifecycle methods. JavaScript functions, by default, do not manage UI or state unless defined within the React environment.

**3. Identify the types of components**  
There are two main types of React components:

* **Class components** – Use ES6 class syntax and support lifecycle methods and state.
* **Function components** – Use regular JavaScript functions and can use Hooks to manage state and side effects.

**4. Explain class component**  
A class component in React is defined using the class keyword and extends React.Component. It includes a render() method that returns JSX and can hold local state and lifecycle methods.

**Example:**

class Welcome extends React.Component {

render() {

return <h1>Hello, Class Component!</h1>;

}

}

**5. Explain function component**  
A function component is a simpler way to write components using a JavaScript function. It returns JSX and can use Hooks (like useState, useEffect) to handle state and logic.

**Example:**

function Welcome() {

return <h1>Hello, Function Component!</h1>;

}

**6. Define component constructor**  
The constructor in a class component is a special function used to initialize state and bind methods. It is called once when the component is created.

**Example:**

constructor(props) {

super(props);

this.state = { count: 0 };

}

**7. Define render() function**  
The render() function is required in class components. It describes what UI the component should display and returns JSX. This method is automatically called by React to update the DOM when needed.

**Code:**

**CalculateScore.js**  
import '../Stylesheets/mystyle.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">

<font color="brown"><h1>Student Details:</h1></font>

<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>Marks: {Total}</span>

</div>

<div className="Score">

<b>Score:</b>

<span>

{calcScore(

Total,

Goal

)}

</span>

</div>

</div>

)

**mystyle.css**

.Name {

font-weight: 300;

color: blue;

}

.School {

color: crimson;

}

.Total {

color: darkmagenta;

}

.formatstyle {

text-align: center;

font-size: large;

}

.Score {

color: forestgreen;

}

**App.js**  
import { CalculateScore } from './components/CalculateScore';

function App() {

return (

<div>

<CalculateScore Name={"Steeve"}

School={"DNV Public School"}

Total={284}

Goal={3}

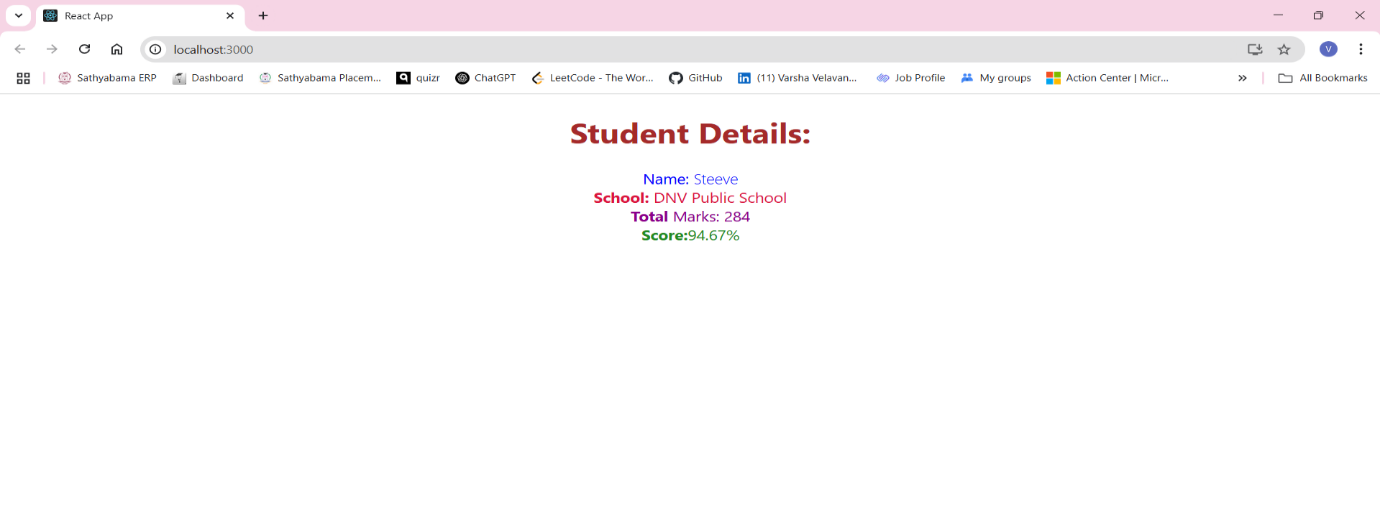
/>

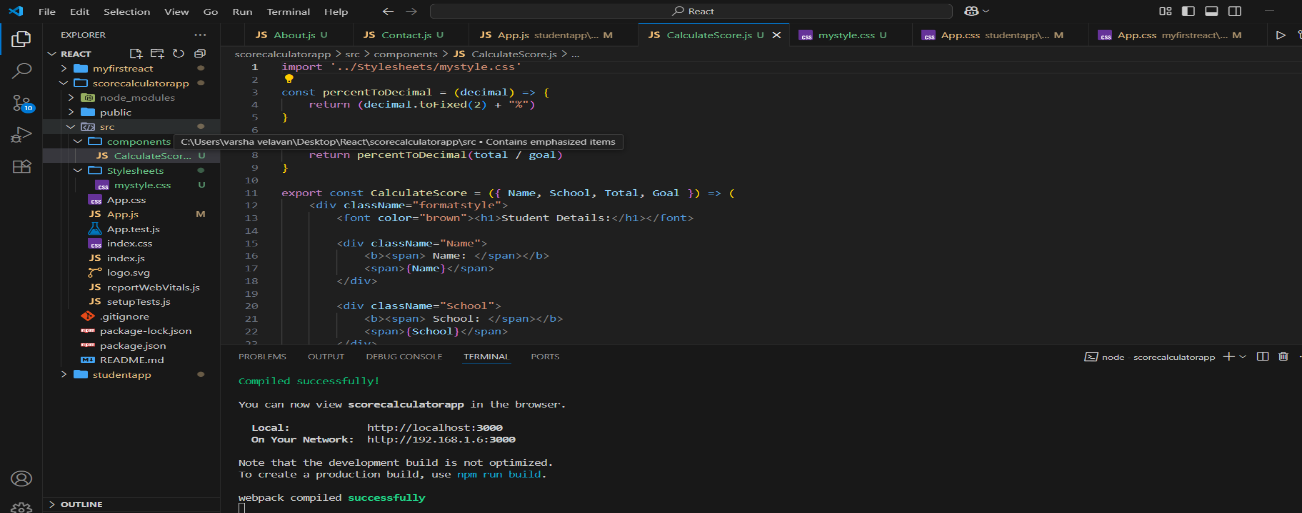
</div>

);

}

export default App;

**Output:  
**

****