**Module 10 – Single Page Application framework – React**

**7.ReactJS-HOL**

**Define Props**

Props (short for properties) are used in React to pass data from one component to another, usually from a parent to a child.  
Props are read-only, meaning the receiving component cannot change them.  
They make components dynamic and reusable.

Example:  
A parent can send a name to a child component using props.  
<Welcome name="John" /> — Here, name is a prop.

**Explain Default Props**

Default props are used to set a default value for props in case the parent component doesn't provide one.  
It helps to avoid errors and keeps components predictable.

**Example:**  
If a component expects a prop called time, we can set:

Component.defaultProps = {

time: "Morning"

};

So, even if no time is passed, it will still show "Morning".

**Difference Between State and Props**

| **Props** | **State** |
| --- | --- |
| Passed from parent component | Defined inside the component |
| Read-only | Can be changed/updated |
| Cannot be modified inside | Can be modified using setState |
| Used to pass data to children | Used to track data internally |

So, props are for passing data and state is for managing data inside the component.

**Explain ReactDOM.render()**

In React, ReactDOM.render() is used to display a React component on the screen.  
It connects your component with the real HTML DOM.

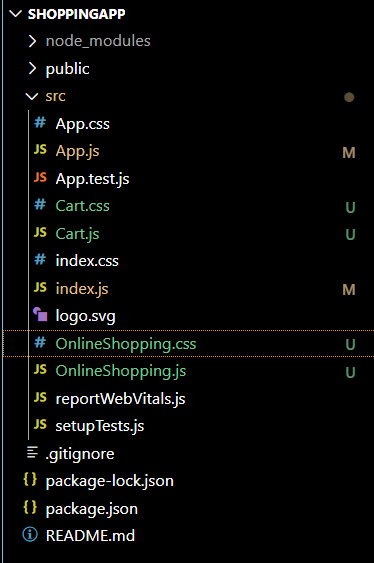
In React 18 and above, the new syntax is:

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(<App />);

It tells React: “Render my App component inside the element with ID 'root'.”

**Structure:**



**App.js:**

import React from 'react';

import OnlineShopping from './OnlineShopping';

function App() {

  return (

    <div>

      <OnlineShopping />

    </div>

  );

}

export default App;

**Cart.js:**

import React from 'react';

import './Cart.css';

class Cart extends React.Component {

  render() {

    return (

      <tr className="green-text">

        <td>{this.props.itemname}</td>

        <td>{this.props.price}</td>

      </tr>

    );

  }

}

export default Cart;

**Cart.css:**

.green-text td {

  color: #2e8b57; /\* or use 'green' \*/

  text-align: center;

  padding: 8px;

}

**index.js:**

import React from 'react';

import { createRoot } from 'react-dom/client';

import App from './App';

const root = createRoot(document.getElementById('root'));

root.render(<App />);

**index.css:**

body {

  margin: 0;

  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',

    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',

    sans-serif;

  -webkit-font-smoothing: antialiased;

  -moz-osx-font-smoothing: grayscale;

}

code {

  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',

    monospace;

}

**OnlineShopping.js:**

import React from 'react';

import Cart from './Cart';

import './OnlineShopping.css';

class OnlineShopping extends React.Component {

  constructor(props) {

    super(props);

    this.state = {

      items: [

        { itemname: "Laptop", price: 80000 },

        { itemname: "TV", price: 120000 },

        { itemname: "Washing Machine", price: 50000 },

        { itemname: "Mobile", price: 30000 },

        { itemname: "Fridge", price: 70000 },

      ]

    };

  }

  render() {

    return (

      <div className="container">

        <h2 className="heading">Items Ordered :</h2>

        <table className="custom-table">

          <thead>

            <tr>

              <th>Name</th>

              <th>Price</th>

            </tr>

          </thead>

          <tbody>

            {this.state.items.map((item, index) => (

              <Cart key={index} itemname={item.itemname} price={item.price} />

            ))}

          </tbody>

        </table>

      </div>

    );

  }

}

export default OnlineShopping;

**OnlineShopping.css:**

.container {

  text-align: center;

  margin-top: 30px;

}

.heading {

  color: green;

  font-size: 28px;

  font-weight: bold;

}

.custom-table {

  margin: 0 auto;

  border-collapse: collapse;

  border: 1px solid #999;

}

.custom-table th {

  border: 2x solid #999;

  padding: 10px 20px;

  color: green;

  font-weight: bold;

}

.custom-table td {

  border: 1px solid #999;

  padding: 10px 20px;

  color: green;

  text-align: center;

  font-weight: normal;

}

**Output:**



