**EXERCISE 7 REACTJS-HOL**

**Define Props**

Props (short for *properties*) in React are used to pass data from a parent component to a child component. They are read-only attributes that help in creating dynamic and reusable components. By using props, you can customize a component’s behavior or appearance without altering its code. Props are passed as attributes in JSX and can be accessed inside the child component using this.props (in class components) or directly as function parameters (in functional components).

**Explain Default Props**

Default Props in React are used to define default values for props in case no value is provided by the parent component. This ensures that the component behaves correctly even when certain props are missing. You can set default props by assigning an object to ComponentName.defaultProps. This feature helps in making components more robust and prevents errors due to undefined props.

**Identify the differences between State and Props**

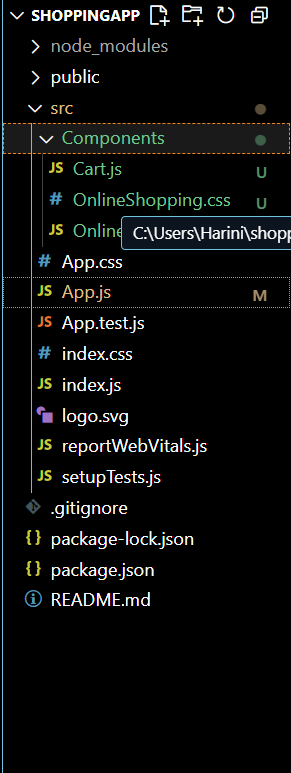
The main difference between **state** and **props** is that props are immutable and controlled by the parent component, whereas state is mutable and managed within the component itself. Props are used to pass data down the component tree, while state is used to store and manage data that may change over time within a component. Props make a component reusable and configurable, whereas state represents the local data that can change based on user actions or internal logic.

**Explain reactDOM.render()**

The reactDOM.render() method is responsible for rendering a React element or component into the DOM. It takes two arguments: the React element (usually created using JSX) and the target DOM node where the component should be mounted. This method is usually called once in the application to render the root component.

**HANDSON EXERCISE:**

**STRUCTURE:**



**CODE:**

**App.js**

import React from 'react';

import OnlineShopping from './Components/OnlineShopping';

function App() {

  return (

    <div>

      <OnlineShopping />

    </div>

  );

}

export default App;

**Cart.js**

import React from 'react';

class Cart extends React.Component {

  render() {

    const { itemname, price } = this.props;

    return (

      <tr>

        <td>{itemname}</td>

        <td>{price}</td>

      </tr>

    );

  }

}

export default Cart;

**OnlinShopping.css**

.container {

  text-align: center;

  margin-top: 50px;

}

h1 {

  color: green;

}

table {

  margin: 0 auto;

  border-collapse: collapse;

  width: 300px;

  font-family: Arial, sans-serif;

}

th, td {

  border: 1px solid grey;

  padding: 10px;

  text-align: center;

  color: seagreen;

}

th {

  background-color: #f2f2f2;

  font-weight: bold;

}

**OnlineShopping.js**

import React from 'react';

import Cart from './Cart';

import './OnlineShopping.css';

class OnlineShopping extends React.Component {

  constructor(props) {

    super(props);

    this.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">

        <h1>Items Ordered :</h1>

        <table>

          <thead>

            <tr>

              <th>Name</th>

              <th>Price</th>

            </tr>

          </thead>

          <tbody>

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

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

            ))}

          </tbody>

        </table>

      </div>

    );

  }

}

export default OnlineShopping;

OUTPUT:

