**Exercise 7**

**Objectives**

1. **Define Props**

Props (short for “properties”) are inputs passed from a parent component to a child component. They allow components to be dynamic and reusable by accepting values externally. Props are read-only — a child component cannot modify them directly. In class components, props are accessed using **this.props**. In functional components, they are accessed via the **props** parameter or object destructuring.

1. **Explain Default Props**

Default props provide fallback values for props not explicitly passed by the parent. They help avoid undefined errors and ensure consistent component behavior. You can define default props using the **defaultProps** property in class components. In functional components, you can also set default values using default parameters. Example: **Component.defaultProps = { name: "Guest" }** will assign "Guest" if no **name** is passed.

1. **Identify the differences between State and Props**

**Props** are passed from parent to child; **State** is managed within the component. Props are **immutable** in the child component; State is **mutable** and managed internally. Props make components **reusable**; State manages **local data** and UI behavior. Props flow **downward** (top-to-bottom); State doesn’t have this flow restriction. Props are for **configuration**; State is for **dynamic behavior** like user interaction.

1. **Explain reactDOM.render()**

**ReactDOM.render()** is the function used to render React components into the DOM. It takes two arguments: the React element and the DOM container.

Syntax: **ReactDOM.render(<App />, document.getElementById('root'));**

It creates a virtual DOM and updates the real DOM efficiently. This is the entry point of most React applications and is typically used in **index.js**.

**Hands On Practice**

1. **Create a React Application named “shoppingapp”**

npx create-react-app shoppingapp

1. **Create a class component named “OnlineShopping”**

**OnlineShopping.js**

import React from 'react';

import Cart from './Cart';

import './OnlineShopping.css';

class OnlineShopping extends React.Component {

  render() {

    const items = [

      { Itemname: 'Shoes', Price: 1500 },

      { Itemname: 'Bag', Price: 800 },

      { Itemname: 'Watch', Price: 2500 },

      { Itemname: 'Shirt', Price: 1200 },

      { Itemname: 'Book', Price: 500 }

    ];

    return (

      <div className='table-container'>

        <h2>Online Shopping Cart</h2>

        <table className='shopping-table'>

          <thead>

            <th>

              Item Name:

            </th>

            <th>

              Price:(₹)

            </th>

          </thead>

          <tbody>

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

              <tr key={index}>

                <td>{items.Itemname}</td>

                <td>{items.Price}</td>

              </tr>

            ))}

          </tbody>

        </table>

      </div>

    );

  }

}

export default OnlineShopping;

1. **Create a class component named “Cart”**

**Cart.js**

import React from 'react';

class Cart extends React.Component {

  render() {

    return (

      <div className="cart">

        <h3>{this.props.Itemname}</h3>

        <p>Price: ₹{this.props.Price}</p>

      </div>

    );

  }

}

export default Cart;

1. **Create a css file for styling the online shopping cart**

**OnlineShopping.css**

.table-container{

  width: 60%;

  margin: auto;

  font-family: Arial, Helvetica, sans-serif;

}

.table-container h2{

  text-align: center;

  margin-bottom: 20px;

}

.shopping-table{

  width: 100%;

  border-collapse: collapse;

}

.shopping-table th, .shopping-table td{

  border: 3px solid grey;

  padding: 10px;

  text-align: left;

}

.shopping-table.th{

  background-color: azure;

}

1. **Modify the App.js file as per this online shopping project**

**App.js**

import React from 'react';

import OnlineShopping from './OnlineShopping';

function App() {

  return (

    <div className="App">

      <OnlineShopping />

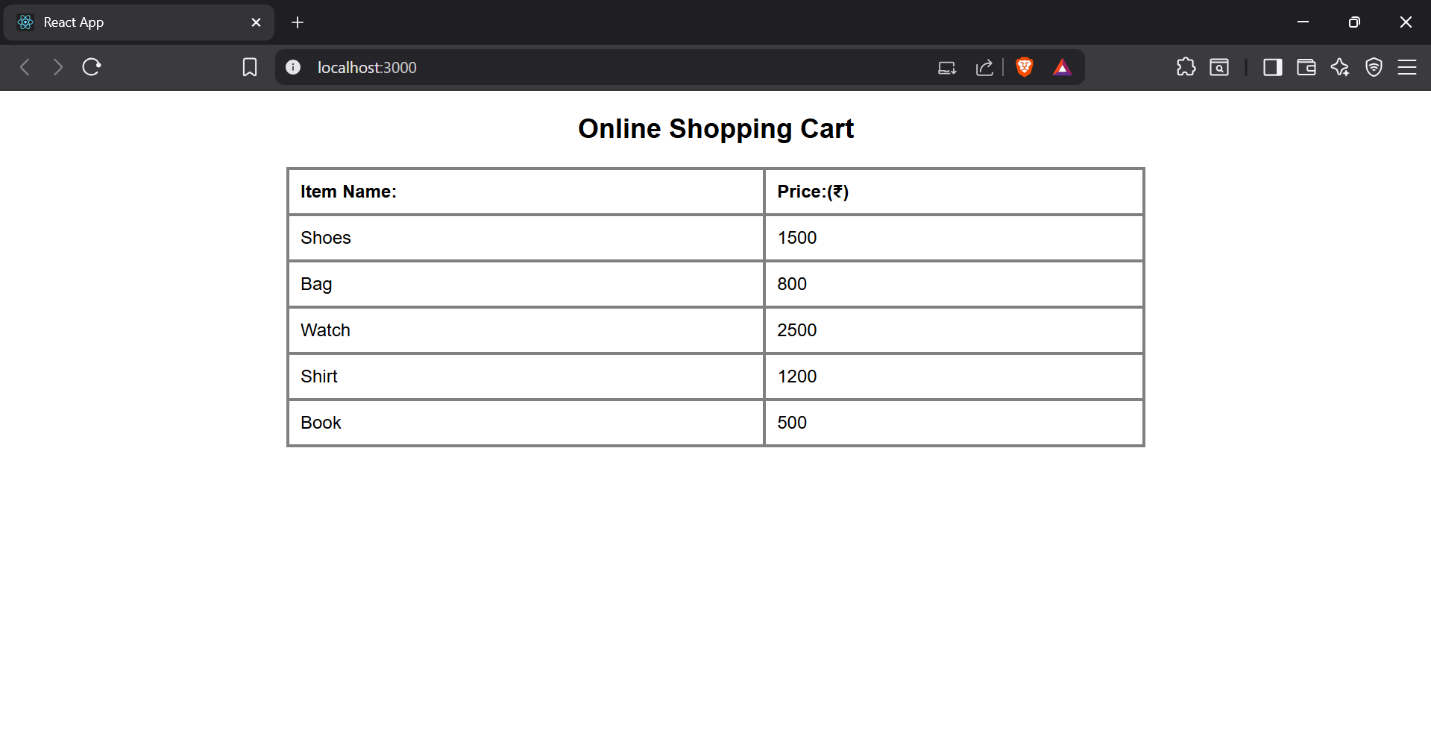
    </div>

  );

}

export default App;

1. **Output:**

****