## **Objectives**

* Define Props

Props (short for properties) are read-only inputs passed to React components. They are used to send data from a parent component to a child component.

**Example:**

function Welcome(props) {

return <h1>Hello, {props.name}!</h1>;

}

**Usage:**

<Welcome name="Chaitra" />

* Explain Default Props

Default Props allow you to specify a default value for a prop when it is not provided by the parent component.

**Example:**

function Greeting(props) {

return <h1>Welcome, {props.name}!</h1>;

}

Greeting.defaultProps = {

name: "Guest"

};

**Usage:**

<Greeting /> // Output: Welcome, Guest!

* Identify the differences between State and Props

|  |  |  |
| --- | --- | --- |
| Feature | Props | State |
| Definition | Data passed to a component | Data managed within the component |
| Mutability | Immutable (read-only) | Mutable (can be changed) |
| Accessed via | this.props (class) or props (func) | this.state or useState (in hooks) |
| Purpose | Configure the component | Store local, dynamic data |
| Ownership | Controlled by parent | Controlled by the component itself |

* Explain reactDOM.render()

ReactDOM.render() is used to render a React element into the DOM.

**Syntax:**

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

* <App />: The React component you want to render.
* document.getElementById('root'): The DOM node where the React component will be mounted.

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

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

root.render(<App />);

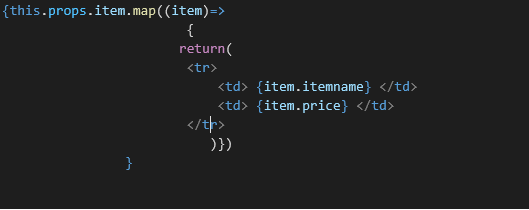
## **Notes**

Estimated time to complete this lab: **60 minutes.**

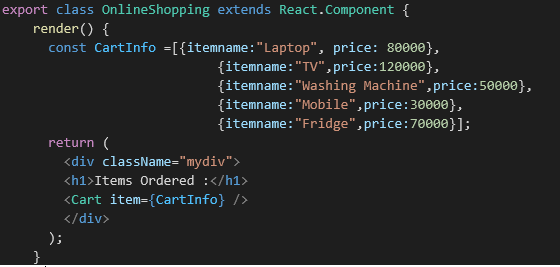
Create a React Application named “shoppingapp” with a class component named “OnlineShopping” and “Cart”.

1. In Cart class, create 2 properties as mentioned below:

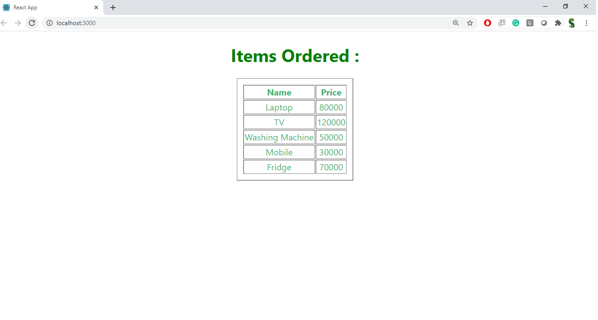
* Itemname
* Price



1. In OnlineShopping class, create an array of Cart and initialize 5 items.



1. Loop through these items and display the data as shown below:



Hands-On:

