**7.ReactJs-HOL**

**Define Props**

**Props** (short for “properties”) in React are used to pass data from one component to another, typically from a parent to a child component. Props are **read-only** and help make components reusable by allowing them to accept inputs dynamically.  
For example:

jsx

CopyEdit

<MyComponent name="Pavi" />

In this case, name is a prop passed to MyComponent.

**• Explain Default Props**

**Default props** are used to provide default values for props in a component if no value is passed from the parent. This ensures that the component behaves predictably even if some props are missing.  
In class components:

jsx

CopyEdit

MyComponent.defaultProps = {

name: "Guest"

};

In functional components:

jsx

CopyEdit

function MyComponent({ name = "Guest" }) {

return <h1>Hello, {name}</h1>;

}

**• Identify the Differences between State and Props**

| **Feature** | **Props** | **State** |
| --- | --- | --- |
| Mutability | Immutable (read-only) | Mutable (can be updated) |
| Usage | Passed by parent component | Managed within the component |
| Purpose | Configure component externally | Track changes within component |
| Access | this.props | this.state |

Props allow components to be dynamic through external input, while state enables dynamic changes within a component itself.

**• Explain ReactDOM.render()**

ReactDOM.render() is a method used to render React elements or components into the DOM. It is used only once in the application (typically in index.js) to mount the root component.  
**Syntax:**

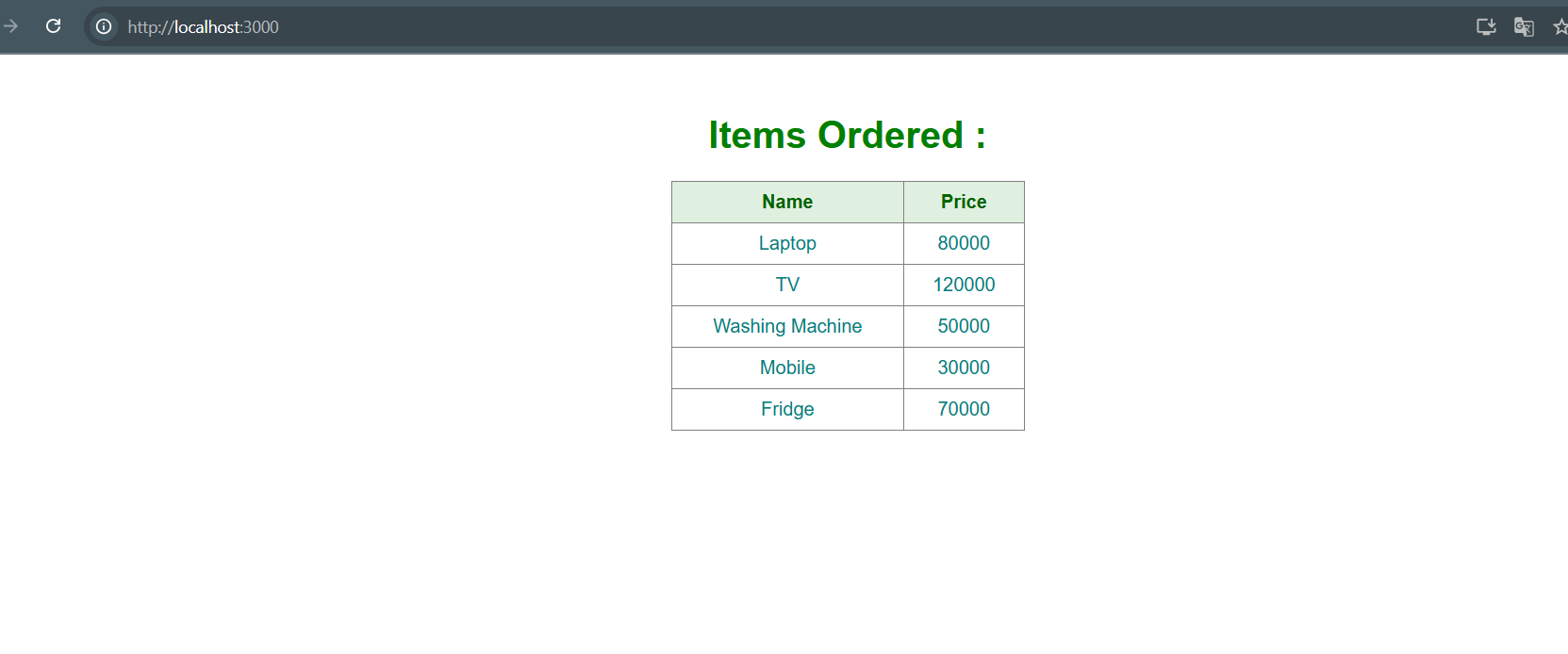
jsx

CopyEdit

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

This tells React to render the App component into the DOM element with the ID 'root'.

**Output:**

****