**2. ReactJS-HOL-Documentation**

**Objective :**

* Create a class component
* Create multiple components
* Render a component

**Prerequisties :**

The following is required to complete this hands-on lab:

* Node.js
* NPM
* Visual Studio Code

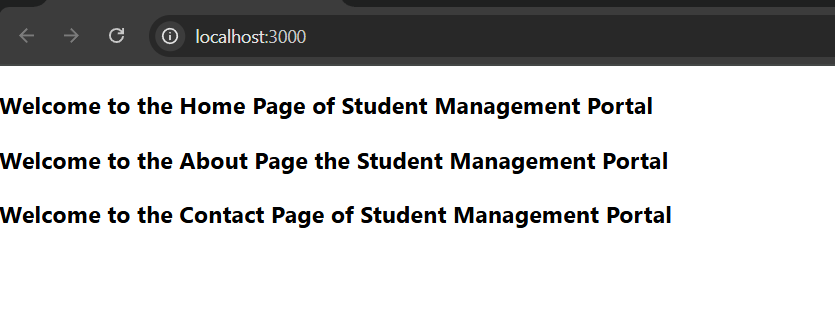
**Project Folder Name :**

* studentapp

**Procedure :**

(Follow up the steps given in hands on)

**Output Snapshot :**



**Knowledge Gain :**

Ques. Explain React components ?

React components are the building blocks of any React application. Think of them like custom HTML elements created using JavaScript:

* They help break the UI into reusable, manageable pieces.
* Each component can have its own logic, structure, and styling.
* They can also handle state (data) and user interactions.

Ques. Identify the differences between components and JavaScript functions ?

| Feature | React Component | JavaScript Function |
| --- | --- | --- |
| Purpose | Builds UI and controls behavior | Performs calculations or actions |
| Syntax | Can be a class or a special function (with JSX) | Plain JavaScript function |
| Uses JSX | Yes | No |
| Lifecycle Methods | Yes (in class components) | No |
| State & Props Handling | Supports state and props | No native concept of state or props |

Ques. Identify the types of components?

React offers two main types of components:

1. Class Components
2. Function Components

Ques. Explain class component ?

A class component is created using the class keyword and extends React.Component. It supports lifecycle methods and internal state.

Ques. Explain function component ?

Function components are simpler and written as plain JavaScript functions. They became powerful with the introduction of Hooks (like useState, useEffect).

Ques. Define component constructor ?

The constructor is a method in a class component used to:

* Initialize state
* Bind event handlers
* Receive props

It runs once when the component is created.

Ques. Define render() function ?

The render() method is essential in a class component. It:

* Returns JSX, which defines the component's UI
* Is called whenever the component needs to re-render (e.g., on state/prop updates)
* State Management – Control dynamic data in components
* Hooks – Functional components can use state and lifecycle
* Virtual DOM – Speeds up rendering
* Ecosystem Support – Integrates with Redux, Router, and more