**WEEK-6:**

**ReactJS-HOL**

1. Here is the output after setting the environment for reactjs.

* Define SPA and its benefits:

Single page application is a web application or website that interacts with the user and rewrites the current page and will not reload entire server i.e, all files are loaded only once.

* Define React and identify its working:

React is a javascript library and it is component based where the user interface is broken down into reusable pieces called components.

* Identify the differences between SPA and MPA:

SPA:

Single Page Application (SPA) Explained:

Dynamic Content Loading:

SPAs loads initial HTML page and then dynamically update content based on user interactions, without requiring a full page reload.

Enhanced User Experience:

This approach is responsive to user experience, as users don't have to wait for entire pages to reload.

Faster Initial Load:

While the initial load might be slower due to the single page being larger, subsequent interactions are generally faster.

JavaScript Frameworks:

SPAs often utilize JavaScript frameworks like React, Angular, or Vue.js to manage the dynamic content updates and components.

Examples:

Social media platforms like Facebook and Twitter are examples of SPAs.

MPA:

Full Page Reloads:

MPAs load a new HTML page for each user interaction, resulting in full page reloads from the server.

Traditional Web Applications:

MPAs represent the traditional web application structure, where each URL maps to a distinct HTML page.

Slower Navigation:

The full page reloads can lead to a slower perceived performance and longer loading times, especially with larger pages.

Server-Side Rendering:

MPAs rely on server-side rendering to generate new pages in response to user requests.

Examples:

Many news websites and blogs are examples of MPA structure.

* Explain Pros & Cons of Single-Page Application

Pros:

Fast and responsive ,improved user interface can be used in any device maybe mobile or a desktop, code reusability and eaiser debugging like react.

Cons:

Initial time for load is longer, JavaScript must be enabled for proper functionality, can be complex for large app size.

* Explain about React

React is an open-source JavaScript library used for building user interfaces and React handles the rendering efficiently.

* Define virtual DOM

The Virtual DOM is a lightweight JavaScript representation of the actual DOM (Document Object Model). React uses it to improve performance.

* Explain Features of React:

**Component-Based Architecture**: Build encapsulated components that manage their own state.

**JSX**: JavaScript + XML syntax that makes writing UI code easier and more readable. **Virtual**

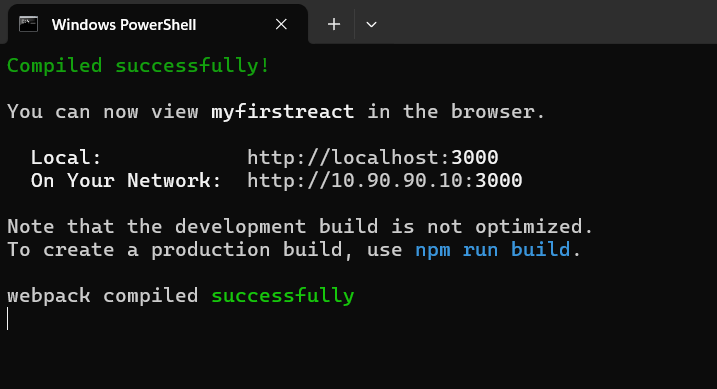
**DOM**: Faster and more efficient UI updates.

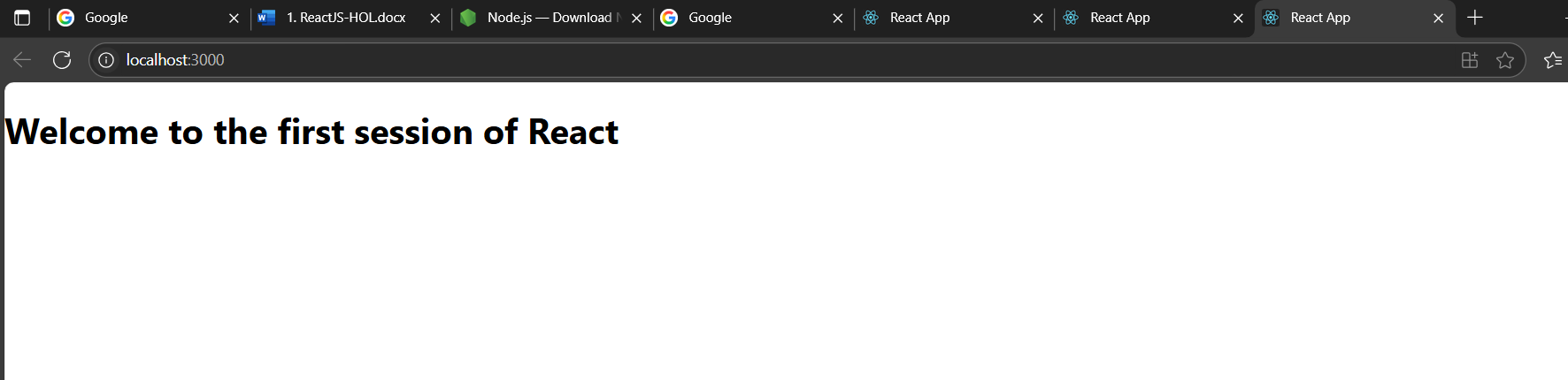
**One-Way Data Binding**: Predictable data flow for better control.

**Hooks**: Functional way to manage state and lifecycle.

**React Router**: Manage navigation in SPAs.

**High Performance**: Efficient DOM updates and re-rendering.

**Output: **



2.

* Explain React components

React components are the building blocks of any react application. A reusable piece of code that returns a react element is known as component. It breaks User interface into manageable pieces.

* Identify the differences between components and JavaScript functions:

Components:

UI-Centric Purpose:

Components are primarily designed to build and manage user interface elements. They encapsulate a part of the UI and its associated logic.

Return UI Elements (JSX):

Components, whether functional or class-based, are expected to return a description of the UI, often in the form of JSX (JavaScript XML.

Lifecycle and State Management (with Hooks for Functional Components):

* + Class Components: Traditionally, class components in React offered built-in lifecycle methods
  + Functional Components with Hooks: With the introduction of React Hooks, functional components gained the ability to manage state.

JavaScript:

General Purpose:

A standard JavaScript function is a block of reusable code designed to perform a specific task. It can accept parameters and return a value.

No UI Rendering Intent:

A typical JavaScript function does not inherently aim to render user interface elements. It focuses on data manipulation, calculations, or other logical operations.

No Built-in Lifecycle or State Management:

Plain JavaScript functions do not have inherent concepts of component lifecycle (e.g., mounting, updating, unmounting) or built-in mechanisms for managing internal state that persists across re-renders.

* Identify the types of components:

They are of two types :class components and function components

* Explain class component:

Class component is an ES6 class ,extends react component. It can manage its own state,can also make use of lifecycle methods. Must use render to return JSX.

* Explain function component:

Function component is a plain JavaScript function that takes props as arguments and return JSX. It is mostly useful when Hooks is used in a program.

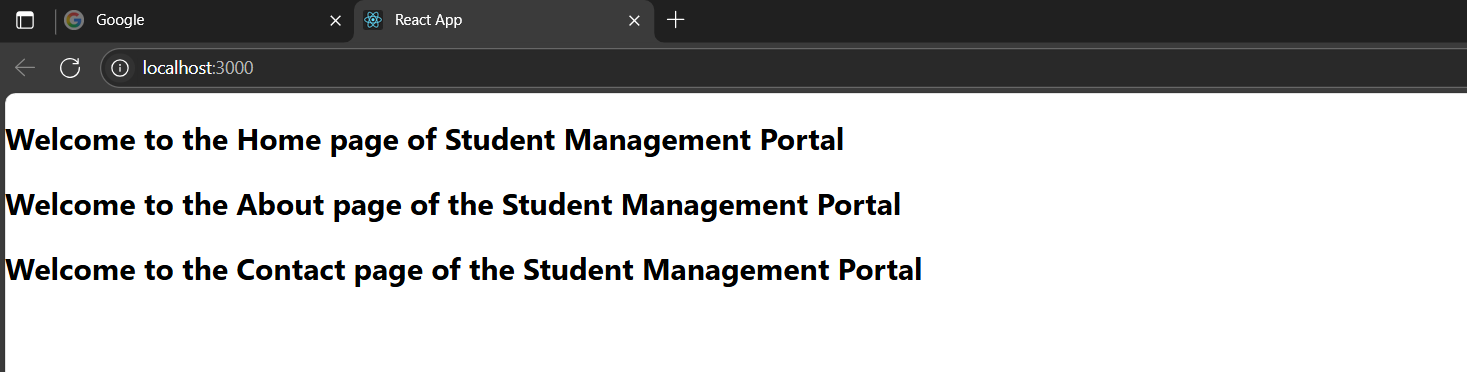
* Define component constructor:

Class component, constructor is a special method used to initialize local state using this.state ,binds event handlers, receives and pass props to parent class using super.

* Define render() function:

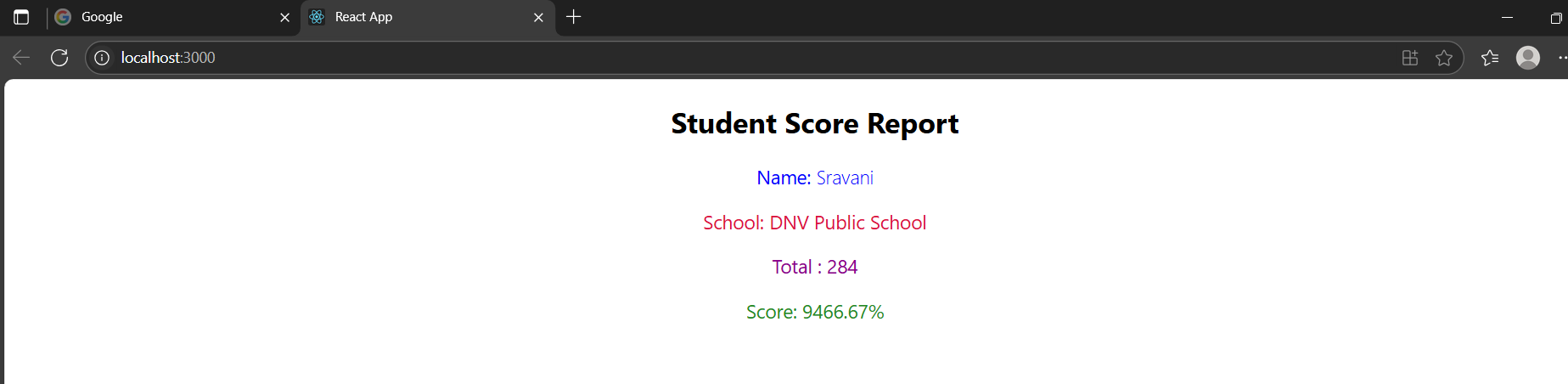
It is a mandatory method in a class components and returns JSX that should be displayed on the screen. It should return only a single parent.

Output:



3.

Output:



4.

* Explain the need and Benefits of component life cycle:

In general every component in react goes through a lifecycle from creation to unmounting. So, it performs actions at specific points, controls rendering behavior based on state or prop changes, cleans up resources, Optimizes performance using specific update conditions.

Benefits:

Better code organization and modularity, Improved performance through controlled updates, Easier to debug and manage state transitions, Enables side-effects handling like API calls, event listeners, etc.

Identify various life cycle hook methods:

constructor(),staticgetDerivedStateFromProps(props,state),render(),

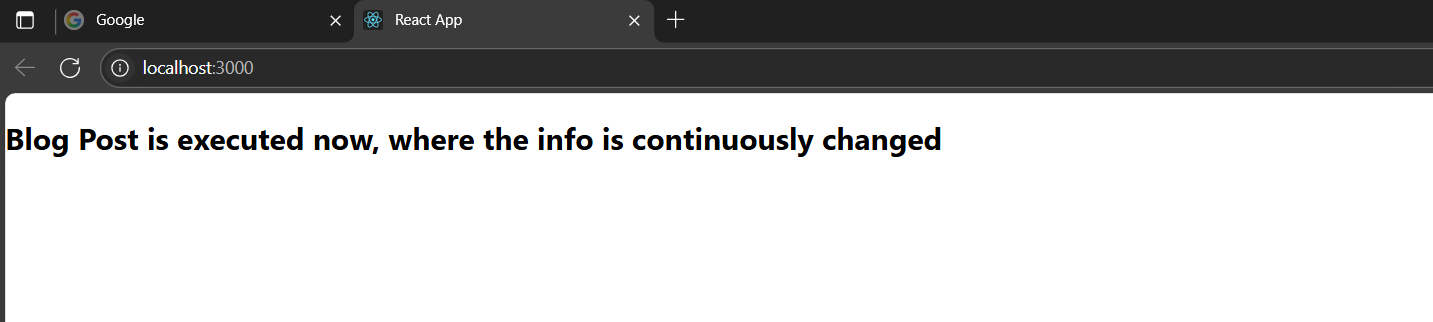
componentDidMount(),staticgetDerivedStateFromProps(),

shouldComponentUpdate(),getSnapshotBeforeUpdate(previousProps,previousProps),etc

* List the sequence of steps in rendering a component:

1. constructor()
2. getDerivedStateFromProps()
3. render()
4. componentDidMount()
5. getDerivedStateFromProps()
6. shouldComponentUpdate()
7. render()
8. getSnapshotBeforeUpdate()
9. componentDidUpdate()
10. componentWillUnmount()

Output:



5.

Output:

