**Module – 3 ReactJs**

**• What is React Js?**

-> React.js is a JavaScript library for building user interfaces, particularly for single-page applications where UI updates are frequent. React allows developers to create reusable UI components and efficiently manage the state of an application. It uses a declarative syntax, making it easier to understand and debug. React is widely used for building interactive and dynamic user interfaces in web applications.

**• What is NPM in React Js?**

-> NPM stands for Node Package Manager, and it is a package manager for JavaScript. NPM is commonly used to install, manage, and distribute third-party libraries (packages) and tools that are needed for building and running React applications.

**• What is Role of Node Js in react Js?**

-> Node.js plays several roles in react. Node.js can be used for server-side rendering to improve initial load times and enhance SEO. Node.js powers the development server, providing features like hot module replacement for faster development. Node.js works with build tools like Webpack to bundle and optimize React code for deployment. Node.js can handle middleware tasks, such as authentication and routing, in the server-side logic. Node.js is used to create a backend server to handle API requests in a full-stack application. Node.js is essential for using npm, the package manager, to manage dependencies and project scripts in React development.

**• What is CLI command In React Js?**

-> CLI (Command Line Interface) commands is used to create a new React application using Create React App.

Example:

“npx create-react-app my-react-app”

This command initializes a new React project named "my-react-app" and sets up the project structure, configuration files, and dependencies, allowing you to start building a React application quickly. The ‘npx’ command is used to run packages from the ‘npm’ registry without installing them globally.

**• What is Components in React Js?**

-> In React.js, components are the building blocks of the user interface. A component is a reusable, self-contained piece of code that encapsulates a specific functionality or part of the user interface.

**• What is Header and Content Components in React Js?**

-> The **Header** component typically contains elements such as navigation menus, branding, and other UI elements that are meant to be displayed at the top of the page.

-> The **Content** component represents the main content area of a page or view. It can dynamically change based on the selected navigation or user interaction.

**• How to install React Js on Windows, linux Operating System?** **How to install NPM and How to check version of NPM?**

-> Open Terminal(Command Propmt or Powershell).

-> Go into your desired location where you want to start react app by using

‘cd’ command in prompt.

-> Use the command ‘npm create-react-app projectname’.

**• How to check version of React Js?**

-> open package.json file.

-> look into dependencies section the latest version is mentioned.

**• How to change in components of React Js?**

-> Props is used to change in components of React Js. props can be defined as an argument in child component.

-> we can change that child component in another component by using their arguments as a property while defining child component.

**• How to create list view in React Js?**

->

**Code:**

[**https://github.com/jaypatel200010/assignment/tree/main/assignment7\_React\_Components\_State\_Props/listview**](https://github.com/jaypatel200010/assignment/tree/main/assignment7_React_Components_State_Props/listview)

**Output:**

****

**• Create Increment decrement state change by button click?**

-> **code:**

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Document</title>

    <style>

      body {

        background-color: wheat;

      }

      div {

        position: absolute;

        top: 50%;

        left: 50%;

        transform: translate(-50% -50%);

      }

      #inc {

        color: blue;

        background-color: greenyellow;

        height: 35px;

        width: 70px;

        border-radius: 20px;

      }

      #dec {

        color: aliceblue;

        background-color: red;

        height: 35px;

        width: 70px;

        border-radius: 20px;

      }

      #reset {

        color: black;

        background-color: grey;

        height: 35px;

        width: 70px;

        border-radius: 20px;

      }

    </style>

  </head>

  <body>

    <div>

      <input type="submit" name="plus" id="inc" value="plus" onclick="plus()" />

      <span id="test">1</span>

      <input

        type="submit"

        name="minus"

        id="dec"

        value="minus"

        onclick="minus()"

      /><br /><br />&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

      <input

        type="submit"

        name="plus"

        id="reset"

        value="reset"

        onclick="reset()"

      />

    </div>

    <script>

      function plus() {

        var value = document.getElementById("test").innerHTML;

        value++;

        document.getElementById("test").innerHTML = value;

      }

      function minus() {

        var value = document.getElementById("test").innerHTML;

        value--;

        document.getElementById("test").innerHTML = value;

      }

      function reset() {

        var value = document.getElementById("test").innerHTML;

        value = 0;

        document.getElementById("test").innerHTML = value;

      }

      // function minus() {

      //     var value = document.getElementById('test').innerHTML;value--;

      //     if (value < 1) {

      //     } else {

      //         document.getElementById('test').innerHTML = value;

      //     }

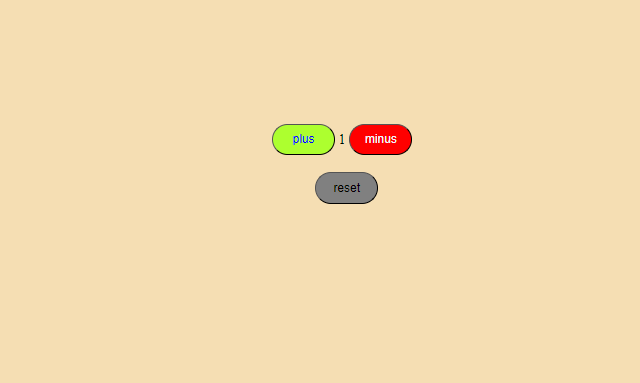
      // }

    </script>

  </body>

</html>

**output:**

****