**ReactJs**

**MODULE: 9 ReactJs Intro**

1. **What is React Js?**

**Ans:**

The React.js framework is an open-source JavaScript framework and library developed by Facebook. It's used for building interactive user interfaces and web applications quickly and efficiently with significantly less code than you would with vanilla JavaScript.

1. **What is NPM in React Js?**

**Ans:**

NPM is short for node package manager, an online directory that contains the various already registered open-source packages.

1. **What is Role of Node Js in react Js?**

**Ans:**

NodeJS is a framework of JavaScript which is mainly used for working with the backend of our application or building the backend using JavaScript, whereas ReactJS is a JavaScript front-end library. It is mainly used for building the user interface or the frontend of our application.

1. **What is CLI command In React Js?**

**Ans:**

CLI is a command line program that accepts text input to execute operating system functions.

1. **What is Components in React Js?**

**Ans:**

Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and return HTML. Components come in two types, Class components and Function components, in this tutorial we will concentrate on Function components.

1. **What is Header and Content Components in React Js?**

**Ans:**

Headers are compositions that extend standard navbar functionalities. They contain additional components like a jumbotron, sub-navbar, or image covers which serve as a containers for extra navigation elements - usually links, forms, or call-to-action buttons.

1. **How to install React Js on Windows, Linux Operating System? How to Install NPM and How to check version of NPM?**

**Ans:**

 Step 1: **Download and install vs code sublime**

**Step 2:** in command promp – checking node – node -v

Step 3: in command promp – checking node – npm –v

Step 4:using node js install the create-node-app-packege

Step 5: cmd: create-react-app –version

Step 6:cd demoreact

Step 7:npm start

1. **How to check version of React Js?**

**Ans:**

Cmd: npm view react version

1. **How to change in components of React Js?**

**Ans:**

1. State: State is used to manage data that can change over time and impact the rendering of your component. To work with state in a functional component, you can use the useState hook, or in a class component, you can define state using this.state. To update state, use the setState function in class components or the function returned by useState in functional components.
2. Props: Props (short for properties) are used to pass data from a parent component to a child component. Props are read-only in the child component, meaning that they cannot be modified by the child component itself. They are set by the parent component.

Here's a basic example of how to create and change state and pass props in a React component:

**import React, { useState } from 'react';**

**// Functional Component**

**function Counter(props) {**

**// Define state using the useState hook**

**const [count, setCount] = useState(0);**

**// Function to increment the count**

**const increment = () => {**

**setCount(count + 1);**

**};**

**// Function to decrement the count**

**const decrement = () => {**

**setCount(count - 1);**

**};**

**return (**

**<div>**

**<h1>Counter: {count}</h1>**

**<button onClick={increment}>Increment</button>**

**<button onClick={decrement}>Decrement</button>**

**<p>Value from parent: {props.valueFromParent}</p>**

**</div>**

**);**

**}**

**// Usage of the Counter component**

**function App() {**

**return (**

**<div>**

**<Counter valueFromParent={42} />**

**</div>**

**);**

**}**

**export default App;**

1. **How to Create a List View in React Js?**

**Ans:**

1. Set Up Your React Application:

If you haven't already set up a React application, you can do so using Create React App or your preferred setup method.

2. Create a Component:

Create a new component that will represent your list view. You can create a file like ListView.js in your src folder.

In this example, the ListView component accepts a data prop, which should be an array of items to display in the list. It uses the map method to iterate over the data and generate list items.

3.Use the Component:

Now, you can use the ListView component in your main application file (e.g., src/App.js) or any other component where you want to display the list view.

4. Style Your List View:

You can apply CSS styles to your list view to make it visually appealing. You can use CSS modules, a CSS framework like Bootstrap, or any other styling method you prefer.

5.Run Your Application:

Start your React application to see the list view in action:

Practical:

npx create-react-app my-list-app

cd my-list-app

// src/components/ListView.js

import React from 'react';

const ListView = ({ data }) => {

return (

<ul>

{data.map((item, index) => (

<li key={index}>{item}</li>

))}

</ul>

);

};

export default ListView;

App.js

import React from 'react';

import './App.css';

import ListView from './components/ListView';

function App() {

const data = ['Item 1', 'Item 2', 'Item 3', 'Item 4'];

return (

<div className="App">

<h1>List View</h1>

<ListView data={data} />

</div>

);

}

export default App;

**11.** **Create Increment decrement state change by button click?**

**Ans:**

<!DOCTYPE html>

<html>

<head>

<title>Increment Decrement Example</title>

<style>

.counter {

font-size: 24px;

margin: 20px;

}

button {

font-size: 18px;

padding: 5px 10px;

margin: 5px;

}

</style>

</head>

<body>

<div class="counter">

<span id="count">0</span>

</div>

<button id="incrementBtn">Increment</button>

<button id="decrementBtn">Decrement</button>

<script>

// Get references to the HTML elements

const countElement = document.getElementById("count");

const incrementBtn = document.getElementById("incrementBtn");

const decrementBtn = document.getElementById("decrementBtn");

// Initialize a counter variable

let counter = 0;

// Function to update the count and display it

function updateCounter() {

countElement.innerText = counter;

}

// Add a click event listener for the increment button

incrementBtn.addEventListener("click", function() {

counter++;

updateCounter();

});

// Add a click event listener for the decrement button

decrementBtn.addEventListener("click", function() {

if (counter > 0) {

counter--;

updateCounter();

}

});

// Initial display

updateCounter();

</script>

</body>

</html>