Module – 3 ReactJs

1. What is React Js?

* React JS, also known as React, is a free, open-source JavaScript library used for building user interfaces. It's a popular front-end library that allows developers to create interactive UI components for web and native applications. React is widely used in web development and is maintained by Facebook and a community of developers.

1. What is NPM in React Js?

* NPM (Node Package Manager) is the package manager for JavaScript and is the default package manager for the JavaScript runtime environment Node.js. In the context of React JS, NPM is used to manage dependencies and libraries required by the React application.
* Here are some key features of NPM in React JS:
* Package Installation: NPM allows you to easily install and manage packages (libraries and dependencies) required by your React application.
* Dependency Management: NPM helps manage dependencies between different packages, ensuring that your application has the correct versions of each package.
* Script Execution: NPM provides a way to execute scripts, such as building and running your React application.

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

* Node.js plays a crucial role in React JS development, although it's not a requirement for building React applications. Node.js is a JavaScript runtime environment that allows developers to run JavaScript on the server-side, making it a popular choice for building server-side applications .
* Here are some key roles of Node.js in React JS:
  + Server-Side Rendering (SSR): Node.js can be used to render React components on the server-side, improving SEO and providing faster page loads.
  + API Development: Node.js can be used to build RESTful APIs that interact with React applications, providing data and services to the client-side application.
  + Build Tools: Node.js is used by popular build tools like Webpack and Babel, which are essential for building and bundling React applications.
  + Development Server: Node.js can be used to create a development server for React applications, providing features like hot reloading and automatic code refresh.

1. What is CLI command In React Js?

* CLI (Command-Line Interface) commands in React JS are used to interact with the React application and perform various tasks, such as creating a new React project, running the application, and building the application for production.
* Here are some common CLI commands used in React JS:
  + npx create-react-app my-app: Creates a new React project called my-app using the create-react-app tool.
  + npm start: Starts the React development server, which allows you to run the application in development mode.
  + npm run build: Builds the React application for production, creating a optimized and minified version of the application.
  + npm test: Runs the React application's tests using the Jest testing framework.
  + npm run eject: Ejects the React application from the create-react-app tool, allowing you to customize the application's configuration.

1. What is Components in React Js?

* Components are a fundamental concept in React JS, and they play a crucial role in building reusable and modular user interfaces.
* In React, a component is a small, self-contained piece of code that represents a UI element, such as a button, a form, or a navigation menu. Components can contain other components, and they can be reused throughout an application to build complex UIs.
* Here are some key characteristics of components in React:
  + Reusability: Components can be reused throughout an application, reducing code duplication and making it easier to maintain.
  + Modularity: Components are self-contained, making it easier to develop, test, and maintain individual components without affecting the rest of the application.
  + Composability: Components can be composed together to build more complex UIs, making it easier to build and maintain large applications.
* There are two main types of components in React:
* Functional Components: These are simple components that accept props and return JSX elements. They are pure functions and do not have their own state.
* Class Components: These are more complex components that have their own state and lifecycle methods. They are typically used when you need to manage state or perform side effects.
* Here is an example of a simple React component:

import React from 'react';

function HelloComponent(props) {

  return <h1>Hello, {props.name}!</h1>;

}

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

* A header is a section at the top of a page that displays site name and navigation. React is the most popular frontend library for developing component-driven user interfaces. It's used for developing single page, mobile, and server-rendered applications.
* 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. How to install React Js on Windows, linux Operating System? How to install NPM and How to check version of NPM?

* Installing Node.js and NPM:
* Windows: Download the Node.js installer from the official Node .js website [6]. Run the installer and follow the prompts to install Node.js and NPM.
* Linux: Use the package manager to install Node.js and NPM. For example, on Ubuntu, you can use the following command: sudo apt install nodejs
* Installing React JS:
* Windows: Open a command prompt or PowerShell and run the following command to install Create React App: npm install -g create-react-app [2].
* Linux: Run the following command to install Create React App: sudo npm install -g create-react-app
* Checking the Version of NPM:
* Windows: Open a command prompt or PowerShell and run the following command to check the version of NPM: npm -v [6].
* Linux: Run the following command to check the version of NPM: npm -v

1. How to check version of React Js?

* Method 1: Using the npm command
* Open a terminal or command prompt.
* Navigate to the root directory of your React project.
* Run the following command: npm ls react
* This will display the version of React installed in your project.
* Method 2: Using the package.json file
* Open the package.json file in your React project.
* Look for the react and react-dom dependencies.
* Check the version numbers listed next to these dependencies.

1. How to change in components of React Js?

* In React, changing components typically involves updating their state or props. Depending on the context, you can use React hooks for functional components or the setState method for class components.

1. **Using State in Functional Components**

import React, { useState } from 'react';

function Counter() {

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

return (

<div>

<p>Current Count: {count}</p>

<button onClick={() => setCount(count + 1)}>Increment</button>

</div>

);

}

export default Counter;

1. Passing and Changing Props

import React, { useState } from 'react';

// Child Component

function Child({ updateCount }) {

return <button onClick={() => updateCount()}>Increment from Child</button>;

}

// Parent Component

function Parent() {

const [count, setCount] = useState(0)

return (

<div>

<p>Parent Count: {count}</p>

<Child updateCount={() => setCount(count + 1)} />

</div>

);

}

export default Parent;

1. Class Components

import React, { Component } from 'react';

class Message extends Component {

constructor(props) {

super(props);

this.state = { message: 'Hello, World!' };

changeMessage = () => {

this.setState({ message: 'You clicked the button!' });

}

render() {

return (

<div>

<p>{this.state.message}</p>

<button onClick={this.changeMessage}>Change Message</button>

</div>

);

}

}

export default Message;

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

* Creating List Views in React JS
* Rendering lists is a common task in React applications, and the recommended approach aligns with the component-based nature of the framework. Here's a breakdown of the key steps:

1. Prepare Your Data

* Ensure you have an array of items that you want to render in your list view. Each item in the array should represent a single list element.

1. Use Array.map()

* The map() method is the preferred way to iterate over an array and create corresponding React elements. It takes a callback function that receives each item as an argument and returns a new React element.

1. Render Unique Elements

* For each item in the array, create a unique React element. This is crucial for React to efficiently update the list when data changes.

1. Provide a Unique key Prop

* To help React identify and update individual list elements efficiently, assign a unique key prop to each element. This key should be a string that uniquely identifies the item within the list.

1. Avoid Using Array Index as the key

* While it might seem convenient to use the array index as the key, this is generally discouraged. If the order of the items changes or items are added or removed, using the index as the key can lead to unexpected rendering behavior.

import React from 'react';

function MyList({ items }) {

return (

<ul>

{items.map((item) => (

<li key={item.id}>{item.name}</li>

))}

</ul>

);

}

export default MyList;

1. Create Increment decrement state change by button click?

* code

import React, { useState } from 'react';

function Counter() {

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

const handleIncrement = () => {

setCount(count + 1);

};

const handleDecrement = () => {

setCount(count - 1);

};

return (

<div>

<p>Count: {count}</p>

<button onClick={handleIncrement}>Increment</button>

<button onClick={handleDecrement}>Decrement</button>

</div>

);

}

export default Counter;