ReactJs

MODULE: 9 ReactJs Intro

1) What is React Js?

Ans: React, is an open-source JavaScript library primarily used for building user interfaces (UIs) and single-page applications.

- React was created by Jordan Walke, a software engineer at Facebook.
- It was developed by Facebook and released to the public in 2013. React allows developers to create reusable UI components and manage the state of those components efficiently.
- **Key features and concepts of React.js include:**
- **Component-based architecture:** React encourages building applications as a collection of reusable components, each responsible for rendering a part of the UI. Components can be composed together to create complex UIs.
- **Virtual DOM (Document Object Model):** React uses a virtual DOM to optimize the updating process of the actual DOM. Instead of directly manipulating the DOM for every state change, React updates a virtual representation of the DOM and then calculates the most efficient way to update the actual DOM. This results in better performance and a smoother user experience.
- **JSX (JavaScript XML):** JSX is a syntax extension for JavaScript that allows developers to write HTML-like code within JavaScript. It enables the composition of React components in a more intuitive and concise manner.

React Hooks: Introduced in React 16.8, hooks are functions that enable developers to use state and other React features without writing class components. Hooks allow for a more functional approach to component logic and can greatly simplify code readability and maintenance

2) What is NPM in React Js?

Ans: NPM stands for **Node Package Manager.**

- NPM is a package manager for JavaScript that allows developers to easily install, share, and manage dependencies for their projects. It is commonly used in React.js development to install and manage third-party libraries, tools, and utilities.
- When working with React.js projects, developers often use NPM to install React itself, as well as other packages such as React Router for routing, Redux for state management, Axios for making HTTP requests, and many more.
- NPM works with a file named package.json, which lists all the dependencies required for a project. Developers can use the npm install command to install all the dependencies listed in the package.json file, making it easy to set up a development environment or share code with others.
- In summary, NPM is an essential tool in React.js development for managing dependencies and simplifying the process of integrating third-party libraries and tools into projects.

3) What is Role of Node Js in react Js?

Ans: Remember, React renders UI components on the frontend, while Node. js manages server-side data communication. The synergy between React and Node. js creates a cohesive and responsive web application.

Server-side rendering (SSR): Node.js can be used to implement server-side rendering for React.js applications. SSR improves initial page load performance by rendering the React components on the server and sending the pre-rendered HTML to the client. This can lead to faster perceived load times and better search engine optimization (SEO).

Build tools and development environment: Node.js powers many build tools and development environments commonly

used in React.js development. For example, tools like Webpack and Babel, which are essential for bundling, transpiling JSX, and other modern JavaScript features, are built with Node.js. Additionally, development servers like webpack-dev-server or Create React App's development server are also powered by Node.js.

API servers: In React.js applications, Node.js can be used to create API servers that handle backend logic and communicate with databases or external services. These API servers can serve as the backend for React applications, allowing them to fetch data and perform server-side operations.

Middleware: Node.js can act as a middleware layer in React.js applications, intercepting and processing HTTP requests before they reach the application's routes or API endpoints. Middleware can perform tasks like authentication, logging, error handling, and request parsing, enhancing the security and robustness of the application.

4)What is CLI command In React Js?

Ans: CLI is a command line program that accepts text input to execute operating system functions. In the 1960s, using only computer terminals, this was the only way to interact with computers.

- A command line interface (CLI) is a text-based interface where you can input commands that interact with a computer's operating system. The CLI operates with the help of the default shell, which is between the operating system and the user.
- **Installation. npm install -g react-cli react.**
- **Usage. Usage:** react-cli [source file] Options: -h, --help output usage information -V, --version output the version number -p, --props [props] Props to pass into the component.

- A command-line interface (CLI) is a text-based user interface (UI) used to run programs, manage computer files and interact with the computer. Command-line interfaces are also called command-line user interfaces, console user interfaces and character user interfaces.

5)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.
There are two main types of components in React:

- **Functional Components:**

- Functional component is just a simple javascript function; it accepts the data in the form of props and returns the react element.

- **Class Components:** React Class components have a built-in state object. You might have noticed that we used state earlier in the component constructor section. The state object is where you store property values that belongs to the component. When the state object changes, the component re-renders.

6)What is Header and Content Components in React Js?

Ans: In React.js, the term "Header" and "Content" components are not specific components provided by React itself, but rather generic terms used to describe parts of a user interface.
Header Component:
5
The Header component typically refers to the top section of a web page or application. It often contains branding

elements like a logo or site name, navigation menus, search bars, or other components that are meant to be consistently displayed across different pages or views within the application.

Content Component:

The Content component generally refers to the main section of a web page or application where the primary content is displayed. This can include text, images, forms, or any other components that are specific to the current page.

7) How to install React Js on Windows, Linux OperatingSystem? How to Install NPM and How to check version of NPM?

Ans: To install React.js on Windows, Linux, or any other operating system, you'll first need to have Node.js and npm (Node Package Manager) installed. Here's a step-by-step guide:

1)Download Installer: Go to the official Node.js website, download the Windows installer, and run it.

2)Install Node.js: Follow the installer instructions to install Node.js and npm.

3)Verify Installation :node -v,npm -v

8)How to check version of React Js?

Ans:To check which React version is your project using you need to open the package. json. Take a look under the dependencies section. It should list all of the dependencies of your project and one of those should be React.

Checking package. json file

- Locate the root directory of your React Native project.
- Open the package. json file in a text editor.
- Look for the "react-native" key within the "dependencies" section.
- The value corresponding to "react-native" key represents the React Native version used in your project.

9)How to change in components of React Js?

Ans: To change components in React.js, you typically need to modify the code within the component files. Here's a general guide on how to make changes to React components:

Locate the Component: First, find the component you want to change. In a typical React application, components are organized into separate files within a directory structure.

Edit the Component File: Open the file corresponding to the component you want to change using a code editor.

Identify What Needs to Change: Determine what specific aspect of the component you want to modify. This could include updating the component's state, changing its props, modifying its rendering logic, or updating its styling.

Make Changes: Edit the code within the component file to implement the desired changes. This might involve adding, removing, or modifying JSX elements, updating state variables using setState(), adding event handlers, or making other adjustments as needed.

Save the Changes: Once you've made your modifications, save the file.