React JS

**Module : 9 React JS Intro**

1. **What is React JS ?**

* ReactJS is a JavaScript library used for building reusable UI components. React is a JavaScript library created by a Facebook.

React is a tool for building UI Components.

Instead of manipulating the browser’s DOM directly, React creates a **Virtual DOM** in memory, where it does all the necessary manipulating, before making the changes in the browser DOM.

React is a JavaScript library for building user interfaces.

React is used to build **Single-Page Application**.

React allows us to create Reusable UI Components.

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

* NPM stands for Node Package Manager.

It’s like an online store for code that helps React developers manage and use libraries and tools in their projects.

**Node** : Node.js is a plateform that lets you run JavaScript code outside of a web browser. It’s commonly used in the React ecosystem for building and managing web applications on servers.

**Package** : A package is a bundle of code that serves a specific purpose. In React, you often need additional code to help with tasks like handling dates, making HTTP requests, or managing state. These packages can be easily found on NPM.

**Manager** : NPM is like a manager or organizer for these packages. It keeps track of which packages your React project depends on and makes it easy to install, update and remove them.

NPM is a handy tool that makes it easier for React developers to find, use and manage code from other developers, saving them time and effort in building web applications.

**Example** : if I want to use a special button component in my React app that someone else has created and shared, you can search for it on NPM, “add it to my cart” by installing it, and then use it in my project.

1. **What is Role of Node JS in React JS ?**

* Node JS is like the backstange worker that helps make the show (your React app) run smoothly.
* **Server-Side Logic :** Node JS is a server-side JavaScript runtime environment. In React applications, most of the code runs in the user’s web browser (the client-side). However, there are tasks that are better handled on the server, like processing forms, handling user authentication and intercting with databases. Node JS allows you to write JavaScript code on the server side to manage these tasks.
* **APIs :** React apps often need to communicate with servers to fetch data or send data back. Node JS is excellent for creating APIs (Application Programming Interfaces) that enable your React app to request and exchange data with the server. It acts as the bridge between your React Front-End and the Server.
* **Build Tools** : Node JS used to run various build tools and development servers. For example, tools like Webpack and Babel, which are crucial for building and transpiling React code are typically run using Node JS.
* **Development** **Environment** : Node JS provides a runtime environment for developing and testing your React app locally. You can set up a Node JS server your React app during development, making it easy to see changes in real-time.
* **Dependency Management** : As mentioned in a previous response, Node JS also come with NPM, which is used to manage and install packages and libraries required for building and running React apps.