**Module-3) React - Component, State, Props**

**What is React JS?**

React JS, also known as React or React.js, is a popular JavaScript library for building user interfaces (UIs) and single-page applications (SPAs).

Developed by Facebook in 2011, React allows you to create dynamic and interactive web applications.

**What is NPM in React JS?**

NPM (Node Package Manager) is a package manager for JavaScript and Node.js.

In the context of React JS, NPM is used to manage and install third-party packages (such as React itself, libraries, and tools) that enhance the development process.

**Role of Node JS in React JS?**

Node.js is a JavaScript runtime environment that allows you to execute JavaScript code outside the browser (i.e., on the server side).

In React JS development, Node.js is essential for:

Running build tools (like Webpack or Babel) to bundle and transpile React code.

Setting up development servers.

Managing dependencies using NPM.

Server-side rendering (if needed).

**CLI Commands in React JS?**

React projects can be created and managed using the React Command-Line Interface (CLI).

Common CLI commands include:

npx create-react-app my-app: Creates a new React project.

npm start: Starts the development server.

npm build: Builds the production-ready bundle.

npm test: Runs tests.

These commands streamline the development workflow and help manage React applications.

**Components in React JS?**

React applications are built using components.

A component is a reusable piece of UI that encapsulates logic, styling, and behavior.

Components can be functional (using functions) or class-based (using classes).

Examples of components: Header, Button, Card, etc.

Components allow you to create modular and maintainable code.

**Header and Content Components in React JS?**

In a typical React application, you might have a Header component and a Content component.

The Header component usually contains navigation links, branding, and other top-level UI elements.

The Content component displays the main content of your application.

By breaking down your UI into smaller components, you can manage them independently and compose them into larger views.

**Installing React JS on Windows and Linux?**

To install React JS, follow these steps:

Install Node.js: React requires Node.js. Download and install Node.js from the official website.

Create a React Project: Open your terminal and run:

npx create-react-app my-app

Navigate to the Project Directory:

cd my-app

Start the Development Server:

npm start

To check the version of NPM:

npm -v

To check the version of React:

npm view react version

**Changing Components in React JS:**

To modify components in React:

Edit the relevant component files (usually .js or .jsx files).

Update the component’s logic, styling, or behavior.

Save the changes, and React will automatically re-render the affected parts of your UI.