**Module – 3 ReactJs**

**1. NPM in React.js:**

* **NPM (Node Package Manager) is the default package manager for Node.js, and it is commonly used in React.js development.**
* **NPM is used to install and manage third-party libraries and packages that can be utilized in a React project.**
* **It simplifies dependency management and allows developers to easily share and distribute their code.**

**2. Role of Node.js in React.js:**

* **Node.js is used as the runtime environment for running JavaScript code outside the browser, on the server side.**
* **In React.js development, Node.js is often used for tools like npm, which is used to manage project dependencies and scripts.**
* **Node.js is not required for React.js development in the browser, but it is commonly used in the development workflow.**

**3. CLI command in React.js:**

* **The CLI (Command Line Interface) for React is typically associated with Create React App (CRA), which is a toolchain for setting up React projects with a predefined and optimized configuration.**
* **Common CLI command for creating a new React app: npx create-react-app my-react-app**
* **After creating the app, you can navigate into the app directory (cd my-react-app) and start the development server using npm start.**

**4. Components in React.js:**

* **Components are the building blocks of a React application.**
* **They are reusable, self-contained units of UI that can be composed together to build complex user interfaces.**
* **Components can be either functional or class-based, and they encapsulate the logic and UI for a specific part of the application.**

**5. Header and Content Components in React.js:**

* **Header and Content components are examples of React components that might be used in a larger application.**
* **The Header component could contain the navigation bar or any other elements at the top of the page.**
* **The Content component might include the main content of the page, such as articles, images, or other interactive elements.**

**6. Installing React.js and NPM on Windows/Linux:**

* **To install React.js, you can use Create React App. Open a terminal and run:**

**npx create-react-app my-react-app**

**cd my-react-app**

**npm start**

* **To install NPM, it comes bundled with Node.js. So, when you install Node.js, NPM is included.**
* **To check the version of NPM, run npm -v in the terminal.**

**7. To check the version of React.js, you can navigate to your React project directory in the terminal and use the following command:**

**npm list react**

**8. For Functional Components:**

1. **Open Your Component File:**
   * **Locate the file that contains your functional component. This is typically a file with a .js or .jsx extension.**
2. **Modify the JSX Code:**
   * **Find the JSX code inside your functional component.**
   * **Update the JSX code to reflect the changes you want. For example, if you want to change the text displayed, update the text content.**

**// Before**

**const MyComponent = () => {**

**return (**

**<div>**

**<h1>Hello from MyComponent!</h1>**

**</div>**

**);**

**};**

**// After**

**const MyComponent = () => {**

**return (**

**<div>**

**<h1>Updated Text!</h1>**

**</div>**

**);**

**};**

**For Class Components:**

1. **Open Your Component File:**
   * **Locate the file that contains your class component. This is typically a file with a .js or .jsx extension.**
2. **Modify the JSX Code Inside the render Method:**
   * **Find the render method inside your class component.**
   * **Update the JSX code within the return statement to reflect the changes you want.**

**// Before**

**class MyClassComponent extends React.Component {**

**render() {**

**return (**

**<div>**

**<h1>Hello from MyClassComponent!</h1>**

**</div>**

**);**

**}**

**}**

**// After**

**class MyClassComponent extends React.Component {**

**render() {**

**return (**

**<div>**

**<h1>Updated Text!</h1>**

**</div>**

**);**

**}**

**}**

**3. Save the File:**

* + **Save the changes you made to the file.**

**4. Revisit or Reuse Component:**

* + **If your component is used in other parts of your application, those parts will automatically reflect the changes you made when you save the file.**