**React JS**

**Module: 9 ReactJS intro**

(1) What is React JS?

Ans :- ReactJS is a javascript library used for building reusable UI Components. React is javascript library created by a Facebook . 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 UI (user interface).

React is used to build Single Page Applications.

React allows us to create reusable UI Components.

(2) What is NPM in React Js?

Ans :- NPM is stand for Node Package Manager. It is like digital store for getting helpful tools and pieces of code that building React apps easier.

Node :- Node.js is a platform that lets you run Javascript code outside of web browser.it’s commonly used in the react ecosystem for building and managing web application on server.

Package :- A package is a bundle of code that serves a specific purpose. In react, You often need additional code to help with task like handling dates, making HTTP request or managing state. This package cane be easily found on NPM.

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

NPM is 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.

(3) What is Role of Node Js in React Js ?

Ans :- Node JS is like the backstange worker that helps make the show in your react app run smoothly.

Server-Side Logic :- Node JS is a server-side Javascript runtime Enviroment. In React applications, most of the code runs in the user’s web browser.there are tasks that are better handled on the server, like processing forms, handling user authentications, and interacting with database.

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 Interface) that enable your React app to request and exchange data with the server.

Build Tools :- Node.js is used to run various build tools and development servers. For example, tools like webpack and Babel, which are crucial for bundling and transpiling React

(4) What is CLI command In React Js?

Ans :- In React.js, CLI stands for Command Line Interface. It's a tool that allows developers to interact with and manage their React projects through the command line. The React CLI provides various commands for creating new projects, starting development servers, building production-ready code, and more. It simplifies the development process by automating common tasks and providing a streamlined workflow for React.js projects.

(5) What is Components in React Js?

Ans : - In React.js, components are the building blocks of a user interface. They are reusable and self-contained pieces of code that encapsulate a specific functionality or UI element. Components can be either functional or class-based. Functional components are simple functions that return JSX, while class-based components are JavaScript classes that extend the React.Component class. Components allow developers to create modular and reusable code, making it easier to manage and maintain complex UI structures in React.js applications.

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

Ans :- In React.js, the Header and Content components are commonly used to structure the layout of a web page or application.

The Header component typically contains elements such as the site logo, navigation menu, and other header-related content. It provides a consistent and easily identifiable section at the top of the page.

The Content component, on the other hand, holds the main content of the page. It can include various components, such as text, images, forms, or other UI elements, depending on the specific content and functionality of the page.

By separating the header and content into distinct components, it becomes easier to manage and update different sections of the page independently, promoting code reusability and maintainability.

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

Ans :-

* To Install react.js requires Node.js to run. You can download and install Node.js from the official website.
* Open a terminal or command prompt
* Install create-react-app. This is a command-line tool that sets up a new React.js project with the necessary files and dependencies. Run thr following Command: npm install-g-create-react-app
* Create a new project React.js Navigate to the directory where you want to create your project and run the following command: npx create-react-app my-app
* Once the project is created, Navigate into the project directory: cd my-app
* Start the development server: Run th following command to start the development server and open your react.js application in the browser: npm start

To install npm, you don’t need to do anything separately as it comes bundled with Node.js

To check the version of npm, you can run the following command in your terminal or command prompt: npm –v

This will display the installed version of npm.

Example: $npm –v

7.24.0

(8 ) How to check version of React Js?

Ans :- To check the version of React.js in your project, you can open the `package.json` file located in the root directory of your React.js project. Inside the `package.json` file, you will find a `"react"` dependency entry with its corresponding version number.

Here's an example of what the `"dependencies"` section in a `package.json` file might look like:

json

"dependencies": {

"react": "17.0.2",

"react-dom": "17.0.2"

}

(9) How to change in components of React Js?

Ans :- To make changes to components in React.js, you can follow these steps:

1. Locate the component you want to change: Components in React.js are typically stored in separate files with a `.js` extension. Find the file that contains the component you want to modify.

2. Open the component file: Use a code editor to open the component file. You can make changes to the component's code within this file.

3. Make your desired changes: Modify the component's code according to the changes you want to make. You can update the component's structure, add or remove elements, update styling, or modify the component's behavior.

4. Save the changes: After making the necessary modifications, save the component file.

5. Check the changes in your application: If you have your React.js application running in development mode (`npm start`), the changes should automatically be reflected in your application. If not, stop and restart the development server (`npm start`) to see the updated component.

By following these steps, you can make changes to components in React.js and see the updated results in your application