**React JS**

1. **What is React JS?**

**A.** React JS is a declarative, efficient, and flexible JavaScript library for building reusable UI components. It is an open-source, component-based frontend library responsible only for the view layer of the application.

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

**A.** NPM stands for Node Package Manager. In the context of React.js, NPM plays a crucial role in managing dependencies, libraries, and packages used in React applications. It is the default package manager for Node.js and is widely used in the JavaScript ecosystem.

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

**A.** Node JS provides NPM to install open-source packages and also people prefer to use JSX format while writing React JS. So Node JS provides efficient environment for creating React by providing useful tools.

1. **What is CLI command In React JS?**

**A.** Command Line Interface (CLI) is a set of tools that streamline the process of creating, building, and managing React applications.

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

**A.** In React.js, components are the fundamental building blocks used to create user interfaces. They are like JavaScript functions or classes responsible for rendering UI elements to the screen. Components allow you to break down the UI into smaller, reusable pieces, making development more modular and manageable.

1. **What is Header and Content Components in React JS?**

**A.** The Header and Content components typically represent parts of the user interface responsible for displaying the header section and the main content area, respectively. These components are often used to structure the layout of a web page or application.

1. **How to install React JS on Windows, Linux Operating System? How to install NPM and How to check version of NPM?**

**A. Installing React.js on Windows and Linux:**

**Step 1: Install Node.js and npm**

**Windows:**

● Visit the Node.js official website and download the Windows installer.

● Run the installer and follow the prompts to install Node.js and npm.

**Linux:**

On Linux, you can install Node.js and npm using a package manager like **‘apt’** (for Debian/Ubuntu-based systems) or **‘yum’** (for Red Hat/Fedora-based systems).

**Step 2: Create a React App**

After installing Node.js and npm, you can create a new React app using Create React App.

**●** Open a terminal or command prompt.

**●** To create a new React app, run:

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

Replace **‘my-react-app’** with your preferred app name.

**●** Navigate into your newly created app directory:

**cd my-react-app**

**Checking the npm Version:**

To check the version of npm installed on your system:

● Open a terminal or command prompt.

● Type the following command:

**npm -v**

This will display the installed npm version.

1. **How to check version of React JS?**

**A.** Open the **‘package.json’** file in the root directory of your project and look for the **‘react’** dependency.

1. **How to change in components of React JS?**

**A.** By using Props we can actually change in components of React JS. We can define props as an arguments in child component.

Now we can change that child component in another component by using there arguments as a property while defining child component.

**10. How to Create a List View in React JS?**

**A.** In this example:

● **‘ListView’** is a functional component.

● **‘items’** is an array of objects representing items.

● The ‘**map’** function is used to iterate through the '**items’** array and generate a list **(<li>)** element for each item.

● Each item's **‘id’** is used as the **‘key’** prop to help React efficiently update the list when items are added, removed, or reordered.

**Input:**

import React from 'react';

const ListView = () => {

  const items = [

    { id: 1, name: 'Item 1' },

    { id: 2, name: 'Item 2' },

    { id: 3, name: 'Item 3' },

    // Add more items here...

  ];

  return (

    <div>

      <h1>List View</h1>

      <ul>

        {items.map(item => (

          <li key={item.id}>{item.name}</li>

        ))}

      </ul>

    </div>

  );

};

export default ListView;