

1) What is React Js?

Ans :

=> React.js is a powerful and popular JavaScript library specifically designed for building user interfaces (UIs).

It's open-source and maintained by Meta (formerly Facebook) and a large community of developers.

Core Concept: Components:

- React focuses on building UIs from reusable components. Imagine each component as a self-contained block responsible for a specific part of the interface, like a button, a form, or a navigation bar. You can combine these components to create complex and dynamic UIs.
- JSX: React uses JSX, a syntax extension that blends JavaScript and HTML, making code more readable and maintainable.
- Virtual DOM: React employs a virtual DOM, an in-memory representation of the real DOM. This allows for efficient updates, only rendering changes to the actual DOM when necessary, leading to faster and smoother performance.

2) What is NPM in React Js?

Ans :

=> In React.js, **NPM (Node Package Manager)** plays a crucial role in managing project dependencies.

It allows you to install, update, and manage these packages in your React project.

It comes bundled with Node.js, meaning you automatically get it when you install Node.js.

- NPM provides commands for interacting with packages:

```
npm install <package-name>
```

When you run `npm install`, NPM downloads and installs those packages in a folder called `node_modules`.

NPM is an essential tool for managing the dependencies in your React.js projects

3) What is Role of Node Js in react Js?

Ans:

=> Node.js is a runtime environment that allows you to run JavaScript code on the server side.

The role of Node.js in a React.js application typically involves

1) Setting up a development environment:

Node.js is often used with tools like npm (Node Package Manager) or yarn to manage dependencies and build scripts for the React application.

2) Server-side rendering (SSR):

Node.js can be used to perform server-side rendering of React components. This means that the initial HTML

content is generated on the server and sent to the client, which can improve performance and SEO.

3) Deployment:

Node.js is often used to deploy React applications to production servers. Tools like Express.js, a web application framework for Node.js, can be used to serve the React application files and handle incoming HTTP requests.

4) What is CLI command In React Js?

Ans:

=> In React.js, CLI stands for Command Line Interface.

The most commonly used CLI tool for React.js is create-react-app.

This tool allows you to quickly create a new React.js project with a predefined folder structure, build configuration, and development server setup without having to manually configure everything yourself.

1) First, ensure that you have Node.js and npm (Node Package Manager) installed on your system. You can download and install them from the official Node.js website if you haven't already.

```
npx create-react-app my-react-app
```

2) You can then start the development server by running:

```
npm start
```

5) What is Components in React Js?

Ans :

=> Components are independent and reusable bits of code.

There are two main types of components in React.js:

1) Functional Components:

Also known as stateless components or presentational components.

2) Class Components:

Also known as stateful components or container components.

Defined as ES6 classes that extend the `React.Component` class.

Overall, components are a fundamental concept in React.js that promotes modularity, reusability, and maintainability in building user interfaces.

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

Ans:

=> In React.js, the term "Header" and "Content" components are not specific components provided by React itself, but rather generic terms used to describe parts of a user interface.

Header Component:

The Header component typically refers to the top section of a web page or application. It often contains branding elements like a logo or site name, navigation menus, search bars, or other components that are meant to be consistently displayed across different pages or views within the application.

Content Component:

The Content component generally refers to the main section of a web page or application where the primary content is displayed. This can include text, images, forms, or any other components that are specific to the current page.

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 on Windows, Linux, or any other operating system, you'll first need to have Node.js and npm (Node Package Manager) installed. Here's a step-by-step guide:

- 1) Download Installer: Go to the official Node.js website, download the Windows installer, and run it.
- 2) Install Node.js: Follow the installer instructions to install Node.js and npm.
- 3) Verify Installation : `node -v` , `npm -v`

8) How to check version of React Js?

Ans:

=> To check the version of React.js that you're using in your project, you can typically find this information in your package.json file, which is located in the root directory of your React project.

And command : `type package.json | findstr "react"`

9) How to change in components of React Js?

Ans:

=> To change components in a React.js application, you need to locate the component you want to modify and then make the necessary adjustments to its JSX code or logic.

Functional Components:

If the component is a functional component, you can modify its JSX directly within the function body. You can change text content, add or remove elements, apply styles, etc.

Class Components:

If the component is a class component, you'll need to modify its `render()` method. You can update the JSX code within the `render()` method to make the desired changes.