Ques:-1 What is Single Page Application (SPA)?

A Single Page Application (SPA) is a web application or website that interacts with the user by dynamically updating the current page rather than loading entire new pages from the server. In a traditional multi-page application, clicking on a link or submitting a form would typically result in a full page reload, fetching a new HTML page from the server.

In contrast, SPAs use JavaScript to dynamically update the content of the current page, often by making asynchronous requests to the server to fetch data (typically in JSON format) and updating the page without requiring a full reload. This can provide a more seamless and responsive user experience, as the transition between different views or states of the application happens more quickly.

Ques:-2 What is DNS and IP Address?

DNS is the domain Name System which are name to searching the web browser because it to in previous days it so difficult to learn the Bit code. Example(www.google.com).

IP Address is the internet protocol address which are numerical value which are given by individual value to all the computer device because the connect to the internet.

Ques:-3 What Is DOM in JS?

DOM stands for Document Object Model.  DOM is a programming interface for HTML and XML documents.

When the browser tries to render an HTML document, it creates an object based on the HTML document called DOM. Using this DOM, we can manipulate or change various elements inside the HTML document.

Ques:- 4 What is React and why we use React instead of JS?

React is a JavaScript library used for building user interfaces, especially for web applications with dynamic and interactive content. It's not a replacement for JavaScript; rather, it's a library that works with JavaScript to simplify and enhance the process of creating complex user interfaces.

React JS is a popular JS library used for building user interfaces. It's maintained by Facebook and a community of developers. React simplifies the process of creating interactive UIs by breaking them down into reusable components.

To clarify, React is not used 'in place of JS'. Rather, It's a library that works with JS to make it easier to create complex UIs by managing the DOM efficiently and providing a component based architecture.

Ques:- 5 What is JSX and why do we use it instead of js?

JSX, or JavaScript XML, is a syntax extension for JavaScript often used with React. It allows you to write HTML elements and components in a syntax that looks similar to XML or HTML directly within JavaScript code. JSX is not a separate language; it's a syntactic sugar that makes it more convenient to work with React components.

JSX stands for JS and XML. It's a syntax extension for JS, often used with React, that allows developers to write HTML like code directly within JS. JSX simplifies the process of writing React components by enabling the mixing of HTML like code with JS logic. It provides a more readable and concise way to describe the structure of UI components. Under the hood, JSX gets transpited into regular JS by tools before it's interpreted by the browser.

Ques:- 6 What is package.json?

package.json is a file in a JavaScript project that serves as a manifest or configuration file. It's used to manage project metadata, dependencies, scripts, and other settings. The package.json file is a crucial part of Node.js and JavaScript projects, and it helps in the organization, versioning, and sharing of code.

package.json is a file used in Node.js projects to manage metadata and dependencies. It contains various details about the project, such as its name, version, entry point, and a list of dependencies required for the project to run.

Ques:- 7 What is the Difference between react and react native? Which one is library or framework?

React:

Type: Library

Description: React, also known as React.js or ReactJS, is a JavaScript library for building user interfaces. Developed and maintained by Facebook, it allows developers to create reusable UI components and build dynamic and efficient single-page applications (SPAs).

React Native:

Type: Framework

Description: React Native is an extension of React that enables the development of mobile applications for iOS and Android platforms. It allows developers to use React concepts and components to build native mobile apps, sharing a significant portion of the codebase between different platforms.

Differences:

Platform:

React: Primarily used for building web applications and SPAs that run in web browsers.

React Native: Specifically designed for building mobile applications for iOS and Android.

Output:

React: Outputs HTML-based user interfaces.

React Native: Outputs native mobile components for iOS and Android.

Code Reusability:

React: Components are typically reusable within web applications.

React Native: Components are highly reusable between iOS and Android applications, allowing for a significant amount of code sharing.

Development Environment:

React: Requires a web browser to run and test applications during development.

React Native: Requires a simulator/emulator or a physical device to test mobile applications during development.

Rendering:

React: Uses the browser's DOM (Document Object Model) for rendering.

React Native: Uses native components, not HTML, for rendering on mobile devices.

Access to Native Features:

React: Limited access to native device features; relies on browser capabilities.

React Native: Provides direct access to native features like the camera, GPS, and more, allowing developers to create truly native mobile experiences.

Dependency on Web Browsers:

React: Needs a web browser to display and interact with components.

React Native: Does not depend on web browsers and directly interacts with native APIs on mobile devices.

Ques:- 8 What is difference between class and functional component?

In React, both class components and functional components are used to define UI elements and behaviour, but they have some differences in terms of syntax, features, and use cases.

Class Components:

Syntax:

Defined using ES6 class syntax.

Extends the React.Component class.

Requires the use of this to refer to the component's instance.

State:

Can have local state using this.state.

Can use lifecycle methods (e.g., componentDidMount, componentDidUpdate, etc.).

Allows the use of setState for updating state.

Props:

Accessible through this.props.

Can be used to receive and handle external data.

Lifecycle Methods:

Can use lifecycle methods to perform actions at different stages of a component's life.

Use Cases:

Suitable for components with complex logic, state management, and lifecycle needs.

Historically used in React before the introduction of hooks.

Functional Components:

Syntax:

Defined as regular JavaScript functions.

Introduced in React 16.8 with the advent of hooks.

State:

Initially, functional components didn't have state (prior to hooks).

With the introduction of hooks, functional components can now use state and other features.

Props:

Receive props as function parameters.

Can use destructuring for cleaner prop access.

Hooks:

Use hooks like useState, useEffect, and others to manage state and perform side effects.

Hooks provide a way to use state and lifecycle features in functional components.

Use Cases:

Suitable for simpler components that don't require state, lifecycle methods, or complex logic.

With hooks, functional components can now handle more advanced features, making them more versatile.