1. Define SPA and Its Benefits

A Single Page Application (SPA) is a type of web application that loads a single HTML page and dynamically updates content as the user interacts with the app, without requiring a full page reload. This is achieved using JavaScript to handle routing and data updates.  
  
SPAs often fetch data asynchronously from the server using APIs (like REST or GraphQL) and then render the content on the client side using frameworks like React, Angular, or Vue.js.  
  
**Benefits of SPA:**- Faster navigation and better performance after initial load.  
- Improved user experience with no full page reloads.  
- Efficient use of server resources.  
- Easier to build mobile-like applications.  
- Can be developed and maintained as component-based systems.

2. Define React and Identify Its Working

React is an open-source JavaScript library created by Facebook for building fast and interactive user interfaces. React uses a component-based architecture, meaning the UI is split into reusable components that manage their own state.  
  
React applications are composed of components, which can be class-based or functional. Modern React favors functional components with hooks. It also uses a virtual DOM to optimize rendering performance.  
 **Working of React:**- Uses a virtual DOM to detect changes.  
- Performs a diffing algorithm to find updates.  
- Re-renders only the changed parts of the UI.  
- Provides hooks to manage state and lifecycle of components.  
- Encourages one-way data binding for predictable state flow.

3. Differences Between SPA and MPA

Single Page Applications (SPAs) and Multi Page Applications (MPAs) differ in how content is served and navigated.  
  
**SPA:**- Uses a single HTML page.  
- JavaScript handles routing and content updates.  
- Faster and more responsive.  
- Great for dynamic apps (e.g., Gmail, Trello).  
 **MPA:**- Loads a new HTML page from the server for each request.  
- Traditional web model using server-side rendering.  
- Better for SEO and static content.  
- Suitable for websites like news portals, blogs, etc.

4. Pros and Cons of Single Page Applications

**Pros:**- Improves user experience by avoiding full-page reloads.  
- Highly responsive and performs well after the first load.  
- Can work offline with service workers.  
- Easier development with modular architecture.  
  
**Cons:**  
- Poor SEO unless server-side rendering or pre-rendering is used.  
- Requires JavaScript to be enabled.  
- Browser history management can be complex.  
- Initial page load can be slower due to loading of all assets.

5. Explain About React

React is not a full-fledged framework but a library focused solely on the view layer of an application. It is declarative, making code more predictable and easier to debug. React encourages building user interfaces through composable components.  
  
React can be extended with additional libraries such as React Router (for routing), Redux or Context API (for state management), making it suitable for complex front-end applications.  
  
React promotes the use of hooks, such as useState and useEffect, to manage component state and side-effects.

6. Define Virtual DOM

The Virtual DOM is an in-memory representation of the real DOM elements. React creates a virtual DOM to track changes and updates only the necessary parts of the UI, rather than re-rendering the entire DOM tree.  
  
React’s rendering process:  
- On state or prop changes, a new virtual DOM is created.  
- It is compared with the previous virtual DOM using a diffing algorithm.  
- Only the differences are applied to the real DOM.  
  
Benefits:  
- Better performance and efficiency.  
- Abstracts away direct DOM manipulation.  
- Enables smoother UI interactions.

7. Features of React

- JSX: Combines JavaScript and HTML-like syntax in one file for better readability and development.  
- Component-Based: Encourages the use of isolated, reusable components.  
- Virtual DOM: Efficient updating mechanism for UI performance.  
- One-Way Data Binding: Data flows from parent to child, making debugging easier.  
- Hooks: Introduced in React 16.8, allows use of state and other features in functional components.  
- Lifecycle Methods: Enable actions at specific points in a component’s life .  
- Context API: Provides a way to share data across components without props drilling.  
- Server-Side Rendering (SSR): Can be integrated with tools like Next.js for SEO optimization.  
- React Developer Tools: Browser extension for debugging React components.  
- Cross-Platform: Use React Native to build mobile applications with the same component model.

**App.js**

import React from 'react';  
  
function App() {  
 return (  
 <div>  
 <h1 style={{ textAlign: 'center' }}>Welcome to the first session of React</h1>  
 </div>  
 );  
}  
  
export default App;

**Output:**



