Answer 1

ReactJS is the highly used open-source JavaScript Library. It helps in creating impressive web apps that require minimal effort and coding. The main objective of ReactJS is to develop User Interfaces (UI) that improves the speed of the apps. There are important pros and cons of ReactJS given as following:

Advantage of ReactJS

1. Easy to Learn and USe

ReactJS is much easier to learn and use. It comes with a good supply of documentation, tutorials, and training resources. Any developer who comes from a JavaScript background can easily understand and start creating web apps using React in a few days

2. Creating Dynamic Web Applications Becomes Easier

To create a dynamic web application specifically with HTML strings was tricky because it requires a complex coding, but React JS solved that issue and makes it easier.

3. Reusable Components

A ReactJS web application is made up of multiple components, and each component has its own logic and controls. These components are responsible for outputting a small, reusable piece of HTML code which can be reused wherever you need them

4. Performance Enhancement

5. The Support of Handy Tools

6. Known to be SEO Friendly

7. The Benefit of Having JavaScript Library

Disadvantage of ReactJS

1. **The high pace of development**
2. **Poor Documentation**
3. **View port**
4. **JSX as a barrier**

Answer 2

The virtual DOM provides a mechanism that abstracts manual DOM manipulations away from the developer, helping us to write more predictable code. It does so by comparing two render trees to determine exactly what has changed, only updating what is necessary on the actual DOM.

Like React, Vue also employs this strategy. However, Svelte proposes another approach to ensure that an application is optimized, compiling all components into independent, tiny JavaScript modules, making the script very light and fast to run.

Answer 3

Real DOM	Virtual DOM

It is an abstraction of a page's HTML. `	It is an abstraction of an HTML DOM.
It can manipulate on-screen elements.	It cannot manipulate on-screen elements.
Any change updates the entire DOM tree.	Any change only updates the relevant node in the tree.
Updating is slow and inefficient.	Updating is fast and efficient.

Answer 4

React components are independent and reusable code. They are the building blocks of any React application. Components serve the same purpose as JavaScript functions, but work individually to return JSX code as elements for our UI. Components usually come in two types, functional components and class components, but today we will also be talking about pure components and higher-order components.

Types of React Components Discussed

Functional Components

Class Components

Pure Components

Higher-Order Components

Answer 5

Functional Components	Class Components
A functional component is just a plain JavaScript pure function that accepts props as an argument and returns a React element(JSX).	A class component requires you to extend from React. Component and create a render function that returns a React element.
There is no render method used in functional components.	It must have the render() method returning JSX (which is syntactically similar to HTML)
Functional component run from top to bottom and once the function is returned it can’t be kept alive.	The class component is instantiated and different life cycle method is kept alive and is run and invoked depending on the phase of the class component.
Also known as Stateless components as they simply accept data and display them in some form, they are mainly responsible for rendering UI.	Also known as Stateful components because they implement logic and state.

Answer 6

We have seen so far that React web apps are actually a collection of independent components that run according to the interactions made with them. Every React Component has a lifecycle of its own, lifecycle of a component can be defined as the series of methods that are invoked in different stages of the component's existence. The definition is pretty straightforward but what do we mean by different stages? A React Component can go through four stages of its life as follows.

Initialization: This is the stage where the component is constructed with the given Props and default state. This is done in the constructor of a Component Class.

Mounting: Mounting is the stage of rendering the JSX returned by the render method itself.

Updating: Updating is the stage when the state of a component is updated and the application is repainted.

Unmounting: As the name suggests Unmounting is the final step of the component lifecycle where the component is removed from the page.

Answer 7

Prop drilling occurs when you need to pass down data through multiple nested components to reach the ones that require the data. This practice can lead to a complex and tightly coupled component hierarchy, making it challenging to manage state and refactor your application.

Using Context API and Hooks to Avoid Prop Drilling

The Context API, introduced in React 16.3, provides a way to share data between components without having to pass props through every level of the component tree. By combining the Context API with Hooks, you can simplify state management and minimize prop drilling in your application.