**What is React.js?**

It is open-source JS library for building user-interfaces

It is developed in 2001 by a software engineer at Facebook and it is maintained by the Facebook Meta

**Why React.js Is Used?**

To develop the Sigla Page Application (SPA)

**React.js uses HTML?**

Under the hood it is all HTML, CSS and JS

We write HTML using JS

**When React.js Released?**

In May 2003

**Current Stable version?**

August 2024 React.js 19 is the current version with the new feature “use” API

**Virtual DOM?**

Generated by react comps. It is the representation of the real DOM elements

When state changed the DOM tree is created. React compares the previous DOM and this DOM tree and update the only parts that has been changed.

**Data binding?**

To manage and synchronize the state of the component with the UI

React uses unidirectional (one-way) data binding  
means data flows from parent to child using props or state

Or data flows from child to parent using events

**What is the architecture of the React.js?**

The architecture of React.js is based on a component-based structure that allows developers to build reusable UI components.

**How to embed two comps in one comp?**

Export two child comps in the one comp (parent) and then just render the parent component

**Advantages of React.js?**

Component based architecture

Virtual DOM

One way data binding

Reusable components

Performance optimization

SEO friendly

Friendly for web developer

**Limitations of React.js?**

* It is library not a framework, so we need some other libraries for additional features like redux
* Rapid changes due to large ecosystems
* Dependency on External Tools (webpack, babel etc)

**Latest React Features?**

* Server component: for SEO optimization
* Asset Loading: ensuring asset are loaded in the background and rendered when needed
* DocumentHead: allow to manage meta data
* Ref as a prop: eliminate the forwardRef and functional component can directly access the ref as a prop
* Context Provider: simplify the context API as <context />

**State management?**

Core concept for handling data in react. We use different tools and libraries to manage state.

The basic way to manage state is useState

To handle complex state, use the useReducer hook

For the global state management use context API, any component can access it without prop drilling.

**Different ways to manage the state in React.js and which one is better?**

1. useState: for single component
2. useReducer: for more complex state logic and one state depends on other state
3. context API: global state, data can pass without props (small to medium project)
4. Redux: for large scale project with complex state logic
5. Recoil: new (alternative of redux)

**What are React hooks and why were they introduced?**

Hooks are the special functions that manage react state and lifecycle features from functional components

We can get the power of the class components in functional components using hooks

We provide two powers of class components (state and lifecycle) to the functional components

**What is Conditional rendering?**

We use if else or ternary condition to render the appropriate component

**What is react router?**

Standard library for routing in React.js

It changes the URL and does not refresh the web page but change the component (UI)

**What is the difference between state and props?**

State:

States are immutable

Internal data store

Props:

It is a property passed from parent to child

And immutable

**Explain the lifecycle methods of components?**

There are three phases of the react class components

1. Mounting
2. Updating
3. Unmounting

**Mounting:**

When component is first created and inserted into the DOM. It has some methods

1. constructor
2. static getDerivedStateFromProps (nextProps, prevState): update state based on the props changed
3. render: describe how the UI should look like
4. componentDidMount: run after the component is entered into DOM

**updating:**

when the component is updating and re-rendering. It has some methods

1. static getDerivedStateFromProps (nextProps, prevState)
2. shouldComponentUpdate (nextProps, nextState): return true or false
3. render
4. getSnapshotBeforeUpdate (prevProps, prevState)
5. componentDidUpdate (prevProps, prevState, snapshot)

**Unmounting:**

it runs just before the component is destroyed and unmounted

1. componentWillUnmount

**Explain the methods used in mounting phase of components?**

**What is this. setState function in React?**

To change the state of the component. It is used in class components while in functional components we use setState

**Types of components in React.js?**

**What is the use of ref in React?**

Directly interact with the DOM element. Use to manage focus, handling animations, and integrating with third party libraries

**Explain the useState and useEffect hooks in React?**

**What is the difference between state and props?**

**What is the difference between class and prop functional components?**

**Functional components:**

They are simple JS functions that returns JSX,

they are stateless

**Class components:**

They are ES6 classes that extends the class components and have rendered methods, they have some additional features like lifecycle methods and states

They are stateful

**What is the purpose of the key attribute in React?**

When list the items in the React, we give the identity to each element to identify which element is added, updated, and removed.

**What is Redux, and how does it relate to React?**

**What is the Context API?**

**What is server-side rendering (SSR) in React, and why is it useful?**

**What are Higher-Order Components (HOCs) and how are they used?**

They take a component and return a new component with additional features and props

**What are React portals?**

**What is the difference between useEffect and useLayoutEffect?**

**What are React fibers?**

**How does the useEffect hook work and what are some common use cases?**

**Can you write a custom hook? Provide an example.**

**What is the Context API and how would you use it?**

**Controlled and uncontrolled comps in react? What’s the difference and when to use each?**

**Promise vs to Async/await. What are the differences, which one is better to handle asynchronous operations?**

**What is memoization and why it is important for performance?**

It is a technique to enhance the performance of the functions by caching the result of expensive functions calls and reusing them when the same inputs occurs again.

We use useMemo() hook to avoid recalculation

**Package selection? If you have 5 packages which one you will choose and why?**

**What are pseudo classes in CSS and how the enhance styling?**

**How do you pass data from parent to child comp and child to parent comp?**

We pass data from parent to child using props

And we can pass data from child to parent with event

**How do you handle forms in React?**

onChange, we use handle the submit of the form with the event

like e.preventtDefault

**How do you handle errors in React?**

By using try…catch or other error boundaries

**What is Redux and can connect redux with react?**

It is the state container for JS application and primarily use with the React.js

It provides the centralized store for managing state

We can connect the redux using the library named “react-redux”

**What is Babel?**

Babel is a tool and a JavaScript transpiler,

It converts JSX into regular JS

Converts the modern JS into old version of the JS that can run on the old browser and environment

**What is a spread operator in JSX?**

**What is the purpose of using useReducer in React.js?**

**What is the purpose of using useContext in React.js?**

**What is React developer tool?**

A browser extension to inspect the react components trees and debug react application

It inspects the hierarchies, props, states and hooks

**How does React.js handle prop drilling and how can it be avoided?**

**What is significant of the key prop in react router?**

For navigation and rendering of the appropriate components

**What is the purpose of forwardRef function in react?**

**Explain the concept of the error boundaries in react?**

**How does react handle form?**

**CODE**

**Given a list of items, how would you implement a searchable list in React?**

**How would you implement a debounced input in React?**

**How would you handle form validation in React?**

**How do you handle asynchronous data fetching in React?**

**How would you implement pagination in a React application?**

**Explain the components of a react-router?**

BrowserRouter

Routes

Route

Link

navLink

useNavigate

useParams

useLocation

**What are some best practices for organizing and structuring a React project?**

**How do you handle error boundaries in React?**

**What is the difference between controlled and uncontrolled components in React?**