React

1. **What is react-js? Tell us about the advantages and disadvantages of using react js.**

* React is an open-source front-end JavaScript library for building user interfaces.
* My preference is React Js because it has an open platform to learn and use easily.
* Creating a Dynamic web application is easier than any other.
* Reusable components are made by React.
* SEO friendly and have a helpful developer toolset.

1. **What is JSX? How does it work?**

Ans: JSX stands for JavaScript XML. Writing HTML code with JavaScript is known as JSX. JSX provides the syntactic sugar for the react.createElement (). JSX doesn’t return multiple parent elements. After compilation, JSX expressions become regular JavaScript function calls and evaluate JavaScript objects. In JSX JavaScript code can be written with curly braces.

* JSX - syntactic sugar by Facebook [ not real js ]
* to convert JSX into vanilla js WEBPACK library used [ bundle.js - vanilla js run by the browser ]
* JSX always allows a single-line Code for JavaScript.
* JSX allows declaring variables with Html.

**3. What is Virtual Dom? What are the differences between virtual and real Dom?**

**Or what is the diff algorithm?**

Ans: DOM stands for Document Object Model. Dom is very much expensive. When to change one element into the DOM. It changes the whole structure of the model. So, DOM always supervises the changes.

Virtual Dom makes a copy of real DOM and compares it. The Virtual Dom compares with the real Dom and identifies only the possible changes.

* Virtual dom - faster & efficient
* reconciliation algorithm

**Diff Algorithm:** A diff algorithm outputs the set of differences between two inputs. These algorithms are the basis of a number of commonly used developer tools.

**4. Difference between props and state.**

Ans:

* Props are used for passing the data from one component to another.
* The state is something that passed the data within the component only.
* Props are immutable and can not be modified,
* where the state is mutable and can be updated or modified.
* Props can be used with state and functional components.
* **State - Dynamic storage of the database.**
* Both of them hold the information.

**5. What is the purpose of the use state? When and why will you use it?**

**Or manage State**

Ans: The useState() hook allows storage of the state variables in functional components. By using state we can state the variables in some places. useState holds the variable information and may change over the lifetime of the component.

* **hooks** - are just functions,
* contains **value** and **setter function**
* state - dynamic storage
* hooks - destructure by two array element

**6. What is prop drilling?**

   Or What is the best way to pass data 4-5 layers down?

   Or What is a context API? How does it work?

Ans: In React sometimes the props are being sent to the 3rd,4th, and nth, components. But the intermediate component has no connection with the props. It’s called prop drilling. To solve this problem **context API** is introduced.

* **Context API** is a way to produce global variables effectively. This is the process of alternating prop drilling. These are also mentioned as props transferred from grandparents to children to parents and so on.

**7. Difference between useEffect and useState?**

  Or why do we need to inject dependency for useEffect?

Ans: **useState** is used for holding the variables, array, or object from inside of the elements.

* useEffect is used to load information outside of the element or code.
* useEfeect just loads the information by default one single time.

We need to use the injected dependency to load the information multiple times under some conditions. **[trigger]**

**8. What other hooks have you used other than useState and use effect?**

Ans: There are a few hooks that are being used in React. These are **useRef**, **useCallback**, **useParam**, **useMemo**, **useContext**, **useReducer**, etc.

**9. Tell us about react component lifecycle.**

Ans: Three phases introduce the react component life cycle. The phases are **Mounting**, **Updating**, and **Unmounting**.

* **Mounting** means adding elements to the DOM. Four built-in methods react have been called for Mounting. These are constructor (), getDerivedStateFromProps (), render (), and componentDidMount ().
* The next phase in the lifecycle is **updating**. A react component is updated when its state or props are being updated.
* The last phase of react life cycle is **Unmounting**. Unmounting happens when a component is removed from the virtual DOM.

**10. What is the purpose of a custom hook? How will you create a custom hook? Give us an example.**

Ans: The custom hook is created by the user. The custom hook is a JavaScript function that is created by the programmer or developer. To share logic among some components and other JavaScript functions custom hook is very useful. Custom hook allows sharing a piece of code to share different levels of several parts of the app.

**useAuth is one kind of example for a custom hook. It is created for using the context API.**

**11. What is the most challenging task you have accomplished in react?**

Ans: In React I’ve faced the most challenging task is to create the context API and to create the private Routing content.

**12. What is Redux and its uses?**

Ans: Redux is a predictable state container for JavaScript. Redux stores the state in a different way and sent it to multiple components.

Action > Reducer > State

**13. Do You Know about React Native?**

Ans: React Native is a platform for building Android and iOS apps.

**14. What is a higher-order component? Give us an example.**

* The higher-order component is a type in which the selected component can be reused in various components.
* Besides this, the higher-order component takes another component as a function and returns the **new** component.

**15. How would you optimize a React Js Application?**

Ans: There have a few techniques to optimize the React Js Application.

* **Using immutable Data Structure:** revolves around a strict unidirectional dataflow. Immutable data objects are simpler to create, test and use.
* **Use React. Fragments to avoid additional Element wrapper**: react. fragment allows a group of children without adding an extra node.

**16. Explain react hooks.**

Ans: React hooks are mainly used in **functional** components. Hooks are one kind of function that lets us hook into react state and lifecycle features.

* **hooks** - are just functions,
* contains **value** and **setter function**
* hooks - destructure by two array element

**17. How to pass data between react components.**

Ans: By using **props** we can pass data between react components. Using context API is an alternative method for passing data between react components.

**18. What are error boundaries in React?**

Ans: The components which use the lifecycle methods are considered the error boundaries.

Three phases where the boundary detects an error. These are:

* Render phases,
* inside a lifecycle method,
* including the constructor.

**19. What are lifting states in react?**

Ans: When the same state is used in several components or sharing the same state in different components a lifting state is used.

Sharing the same state in the parent element and passing to the child element lifting upstate is more useful.

**20. Why react using class Name instead of class attribute?**

Ans: React uses className because react has a **keyword** named **class**. So, removing the contradiction className is used instead of class.

**21. What is a fragment in React?**

Ans: React fragment is used to return multiple elements in the React components. Inside the fragment, a list of children is introduced.

**22. Why fragments are better than container divs?**

Ans:

* React fragment is lightweight and takes fewer memory spaces than div.
* Some mechanisms in CSS like flexbox or grid have a parent-child relationship and using the div inside this relationship makes it harder to create the layout.

**23. What are stateless Components?**

Ans: The independent state is called the stateless component.

**24. What are stateful components?**

Ans: The component which behaves like a dependent component is known as a stateful component. **The class component is stateful.**

### 25. What is the purpose of push () and replace () methods of history?

### Ans: The history is used for visiting locations.

### The push () method is used to add a new location and

### the replace() method is used to replace the current location with the new one.