**REACT**

1. What is React? What are its features?
2. What are the Advantages of React?
3. What are SOLID principles of React?
4. What is the File structure for React?
5. What are coding standard in React?
6. How is React different from React Native?
7. What is Babel?
8. What is JSX? How does the browser read it?
9. What are Fragments in React?
10. Explain the building blocks of React?

* The five main building blocks of React are:
  1. **Components**: These are reusable blocks of code that return HTML.
  2. **JSX**: It stands for JavaScript and XML and allows you to write HTML in React.
  3. **Props and State**: props are like function parameters and State is similar to variables.
  4. **Context**: This allows data to be passed through components as props in a hierarchy.
  5. **Virtual DOM**: It is a lightweight copy of the actual DOM which makes DOM manipulation easier.

1. What is components in React? How they are different types of components from others?
2. How to create components in React?
3. Why we use keys when creating list of components?
4. What are class components and functional components?
5. Which is better functional or class component? Why?
6. What is functional component why its uses over the class component?
7. What is Pure Component?
8. What is Controlled and Uncontrolled component?
9. What is state-full and state-less component?
10. What are the different phases of the component lifecycle?
11. What are the lifecycle methods of React?
12. How do you implement mount, update and unmount with functional components?
13. What are Higher Order Components(HOC)?
14. What are HOC and functional architecture? Please elaborate and explain it with example?
15. Create a HOC component to reuse it as a reusable table component?
16. In React component how it differs from other component?
17. What will be the benefits of using functional component in dev vs prod?
18. What is States and Props?
19. What is difference b/w useState and Props?
20. How to pass data between Components?
21. What is prop drilling? What is disadvantage and how to fix?
22. Differentiate between real DOM and virtual DOM?
23. What is virtual DOM? How it works? What is the purpose of creating?
24. What is shadow DOM? How it works?
25. What are hooks in React? Give a brief example? What are rules of hooks?
26. What is custom hooks in React?
27. What are life cycle hooks and life cycle methods?
28. What are class based and function based life cycle hooks?
29. What are two most used lifecycle method in hooks?
30. What is State management?
31. What State management we used in react and explain why we use them?
32. What are State Management Libraries and how we use them?
33. What is the use of ref in React?
34. What is useState. Why we can't use let or var instead of useState?
35. What is the difference b/w useState and useReducer?
36. What is difference b/w useRef, createRef in React?
37. What is useEffect in React?
38. What are the use of arguments for useEffect, how they work?
39. What dependency is passed in useEffect?
40. If we pass an object to useEffect how does it behave.
41. What is useMemo? Why we use it? Explain with an example?
42. What is React.Memo? Where we use React.Memo?
43. What is difference b/w useEffect and useMemo?
44. Why we use Memoization in React?
45. What are the techniques uses in Memoization, what problems we face what are their principles?
46. If we are reutilizing the value with Memoization then why we don't use cache?
47. What is useContext in React?
48. In useContext we have variable a, b, c and all these are siblings to each other. after made a change in parent for c only, what will happen on these component, will it re-render?
49. What are interceptors in React?
50. What are synthetic events in React JS?
51. What are error boundaries?
52. How do you implement error boundaries and what are predefined for that?
53. What is encapsulations in React?
54. What is polymorphism in React?
55. What are forms in React?
56. What is the lazy loading in React?
57. What is skeleton loader?
58. What is tree shaking in React? How it will happen?
59. What is Client side rendering and Server side rendering (CSR & SSR)?
60. What is used for SSR?
61. How to optimize SEO in CSR?
62. How to avoid re-rendering?
63. What is Lazy loading (Suspense and fallback)?
64. What is code splitting?
65. What is code coverage? What tools we use?
66. What is React Router?
67. How do you use private protector routes in react with router? Is this a part of react route?
68. How do you implemented authentication in React?
69. How to perform Automatic redirect after login?
70. How do you validate user is logged in OR valid user or not in React?
71. If login to dashboard, will we redirect to login screen if page is refreshed?
72. Draw diagram for login & save account channel functionality (how structure your component).
73. Explain the entire flow for the login and authentication in React? Explain scenarios in brief?
74. What are the roles of JWT tokens?
75. What happen when token gets expired and how user maintain the login? So that user having a seamless experience?
76. What is Web Worker?
77. Shallow copy vs Deep copy?
78. What is Flux architecture in React?
79. What are debugging tools in React?
80. How to use web-flow in React?
81. What are the Various vision or techniques to optimize react code or performance?
82. What are some good engineering practices that we can apply while creating a page in the React?
83. How you will identify which component is taking more time to load?
84. Differences between old react (before 16.8) and new react version (with hook)?
85. On performance basis how we decide this component will be best?
86. What is the process of CI/CD to deploy the code?
87. When you want to migrate the images from local to stage than prod sever how do you configure the any Web-Pack file?
88. We have 1 parent component and inside 1 child called A and inside A we have another child B. Variables are passed from Parent to A and B. In parent we are changing values used in child B but not used in child A, what is behavior, will A re-render and B re-render or both will re-render. vice versa.

**REDUX & API**

1. What is Redux and where & why do we use in React?
2. Explain the core components of react-redux?
3. What are core Principals/Designs or Concepts on which Redux is built up on?
4. What is the execution Process of Redux?
5. Explain the workflow or flow of data through Redux?
6. What is Redux toolkit and the middleware and how it works?
7. What are the use case of Redux uses?
8. What is Context API in React?
9. Difference b/w Context API and Redux?
10. Difference b/w useContext and Redux Toolkit?
11. How store works?
12. Where do wrap the store?
13. What is despatcher in redux?
14. What is fetch and Axios?
15. How do you call API in React?
16. What will be your best approach to fetch the data from API in react? Give an example.
17. What is Thunk & Saga?
18. Is it possible to implement redux thung or saga with the functional approach?
19. Explain one-way data binding in Redux.

2. Hooks in a React Redux App - useSelector, useDispatch

3. connect() - The connect() function connects a React component to a Redux store.

4. connect() Parameters

mapStateToProps?: (state, ownProps?) => Object

If a mapStateToProps function is specified, the new wrapper component will subscribe to Redux store updates. This means that any time the store is updated, mapStateToProps will be called.

mapDispatchToProps?: Object | (dispatch, ownProps?) => Object

If your mapDispatchToProps is declared as a function taking one parameter, it will be given the dispatch of your store.

5. middlewares

6. combineReducers

There are four fundamental concepts of redux in react which decide how the data will flow through components

• Redux Store: It is an object that holds the application state

• Acrtion Creators: These are unctions that return actions (objects)

• Actions: Actions are simple objects which conventionally have two properties- type and payload

• Reducers: Reducers are pure functions that update the state of the application in response to actions

How to pass state data from one component to another component?

• Parent to Child (via props)

Child to Parent (via callback functions)

Sibling to Sibling

Using Context API

**JEST**

1. How we do the unit testing on the code?
2. What is Unit Testing?
3. For Unit Testing in react what we use and elaborate?
4. What is Jest? What are Jest tools?
5. Can you explain what mock functions are? How are they created?
6. What do you understand about snapshot testing in context with Jest?
7. How to check the element is present of the DOM using jest?
8. Jest tool – unit test cases (If function having third party dependency how you write unit test cases for this. Will you use mock/spi)

EXERCISE

Given an api, and fetch title image and rating of the product?

Increment and Decrement of Text size on click in React?

Create Stop Watch in React/

1. What is the process that you will follow if you assign a new task and you don’t have any knowledge regarding that?