1. What are let, const and var?

**var**: Used to declare a variable, but it has function scope and can be redeclared.

**let**: Used to declare a variable with block scope, but it can be updated.

**const**: Used to declare a variable with block scope, but it cannot be updated or redeclared.

1. What is hoisting. in all kind of variables?
2. What is closure?

A function inside another function that remembers the variables from the outer function even after the outer function has finished running.

1. What is Promise?

A special JavaScript object that handles asynchronous operations. It has three states: **pending**, **resolved (fulfilled)**, and **rejected**.

1. What are difference between promise and async/await?
2. **Promise**: Uses .then() and .catch() to handle success and errors.

**async/await**: A simpler way to write promises using async and await, making code easier to read.

1. What is ReactJS?

ReactJS is a JavaScript library used to build user interfaces, mainly for single-page applications. It makes UI updates fast and easy using components.

1. How does ReactJS work?

React creates a **virtual DOM** (a copy of the real DOM).

When something changes, React updates only the changed part in the real DOM, making it fast and efficient.

1. What is DOM?

DOM (Document Object Model) is a tree-like structure that represents HTML elements in a web page. JavaScript can change the DOM to update content dynamically.

1. What is hooks?

Hooks are special functions in React that let you use state and other React features without writing a class. Example: useState, useEffect.

1. What is useState and useEffect?

**useState:** Helps to create and manage variables (state) in a functional component.

**useEffect:** Runs code when the component loads or when a specific value changes (used for API calls, event listeners, etc.).

1. What is props-drilling and how to stop it?

**Props drilling** happens when you pass data (props) from a parent component to multiple child components, making the code harder to manage.

**To stop it**, you can use **Context API** or **Redux** to pass data directly to components without going through every layer of the component tree.

1. Explain process of Context-API?
2. What is Redux?

**Redux** is a state management library for JavaScript applications, usually used with React. It helps manage the app's global state in one place, making it easier to share data across multiple components.

1. What is life-cycle of component?
2. Write a program to reverse a string?
3. var let and const difference
4. settimeout
5. promise
6. async await
7. why we use react
8. what is state why we use
9. what is props and drilling solution of it
10. clouser
11. introduction
12. usestate & useeffect
13. Var let and me different
14. What is Hoisting
15. What is promise
16. What is async await
17. What is event loop
18. what Asynchronous
19. What is Typescript
20. What is call bind and apply method
21. What is clouser
22. What is React js
23. What is props Drilling
24. What is Context API
25. What is use memo & use callback
26. Find the accurence of any number
27. Introduction
28. What is a closure in JavaScript? Provide an example.
29. Explain the difference between var, let, and const in JavaScript.
30. What is hoisting in JavaScript? Provide an example.
31. Write a program to find the maximum and minimum values in an array.
32. . 5. Explain the concept of async/await in JavaScript.
33. difference async/await and promise
34. What is a Promise in JavaScript? Provide an example.
35. Explain the syntax of a Promise in JavaScript.
36. How do you handle errors in async/await and Promises?
37. What is the useState hook in React? Provide an example.
38. Explain the concept of useEffect in React.
39. Explain the concept of lifecycle methods in React.
40. introduction
41. project explaination
42. basic react application program
43. 16.difference between virtual dom and actual dom
44. 17.what is diffing and reconcilation
45. 18.what is dom
46. What is html
47. What is tag and element and attribute and difference
48. difference between html and html 5
49. What is css
50. css selectectors
51. difference between class and id selector
52. what is mern
53. why node js
54. difference between js and nodejs
55. how nodejs works
56. event -driven architecture
57. what is api
58. types of api
59. What is restful apis
60. bootstrap is necessary? if yes then why?
61. what are the pros and cons of using bootstarp
62. what is cdn
63. why mern is popular than full stack develeopment
64. why do we need expresss when we already have node
65. What is MERN
66. How mongobd express or react work
67. Class or function components
68. How to create rest api in express
69. How we can handle state
70. Event driven architecture
71. Event loop
72. Asynchronous or synchronous operation
73. Node pakage manager
74. Difference between SQL and noSql
75. Let var const diffrence
76. Hooks in react
77. Context api
78. Diffrence between Local storage , session storage and cookies
79. Middleware