Fresher (0-2 years)

HTML & CSS

1. What is the difference between HTML and HTML5?
2. Explain the box model in CSS and how it works.
3. How would you center an element horizontally and vertically in CSS?
4. What are semantic HTML elements, and why are they important?
5. How do you create a responsive layout using CSS?
6. What's the difference between `padding` and `margin`?
7. What is the purpose of the `alt` attribute in images?
8. Explain the difference between `id` and `class` selectors in CSS.
9. What is Flexbox, and when would you use it?
10. How does CSS specificity work?

JavaScript Basics

11. What are JavaScript variables, and how do you declare them?
12. Explain the difference between `let`, `const`, and `var`.
13. What are functions in JavaScript, and how do you define one?
14. What is an event in JavaScript, and how would you handle one?
15. Can you explain what promises are in JavaScript?
16. What is the Document Object Model (DOM)?
17. How would you access an HTML element by its ID in JavaScript?
18. What's the difference between `==` and `===`?
19. How do arrow functions differ from regular functions?
20. What are JavaScript objects, and how do you create one?

Frameworks & Libraries

21. What is React, and why is it popular for frontend development?
22. How do you create a component in React?
23. Explain the concept of props in React.
24. What are some benefits of using a CSS framework like Bootstrap?
25. How would you add a CSS framework to a project?

General Knowledge

26. What are the core differences between client-side and server-side rendering?
27. Explain what a version control system is and why it's useful.
28. How would you optimize a webpage for faster load times?
29. Describe the purpose of using DevTools in a browser.
30. What do you know about cross-browser compatibility?

Intermediate (2-5 years)

HTML & CSS

1. How would you implement a grid layout using CSS Grid?
2. What are CSS preprocessors, and which ones have you worked with?
3. Explain the difference between `em`, `rem`, and `px`.
4. How does `position: absolute` differ from `position: fixed`?
5. Describe how you would approach building a mobile-first design.
6. How do CSS animations work, and when might you use them?
7. What is BEM, and why would you use it?
8. How would you implement lazy loading for images in a webpage?
9. Explain CSS transitions and give an example of how to use them.
10. What is a pseudo-class in CSS? Provide an example.

JavaScript Advanced Concepts

11. What are closures in JavaScript, and why are they useful?
12. Explain the concept of `this` in JavaScript.
13. What are higher-order functions? Provide an example.
14. How does asynchronous JavaScript work, and why is it important?
15. What is destructuring, and how would you use it?
16. Describe the purpose of `async` and `await` in JavaScript.
17. What is a module in JavaScript, and how do you create one?
18. Explain the concept of currying in JavaScript.
19. What is the purpose of `map`, `filter`, and `reduce`?
20. How does JavaScript handle memory management?

React & Modern Frontend Libraries

21. Explain the concept of state in React.
22. How do React hooks work, and why would you use them?
23. What is Redux, and how does it help in managing application state?
24. Explain the React lifecycle methods.
25. Describe what the Context API is in React.
26. How would you handle API requests in React?
27. What are the benefits of server-side rendering with React?
28. How would you optimize a React application for performance?
29. Explain the use of `PropTypes` in React components.
30. What's the difference between controlled and uncontrolled components?

Expert (5+ years)

Advanced JavaScript & Patterns

1. Explain the concept of closures in-depth and provide a complex example.
2. How would you use `Memoization` in JavaScript, and why?
3. Describe design patterns you've used in JavaScript, such as the Singleton or Observer patterns.
4. What are WeakMap and WeakSet, and when would you use them?
5. Explain the JavaScript event loop and how it manages asynchronous events.
6. What is debouncing and throttling, and how are they implemented?
7. How would you implement data caching in JavaScript?
8. What is functional programming, and how have you used it in JavaScript?
9. How does garbage collection work in JavaScript?
10. Explain various types of proxies and their uses.

React, Architecture, and State Management

11. What are React's Concurrent Mode features, and how do they improve performance?
12. Explain the concept of Suspense and how you've used it.
13. How would you implement a custom hook in React, and why?
14. Describe server-side rendering in React and how you've used Next.js or similar frameworks.
15. How do you handle complex state management in large applications?
16. Explain the difference between client-side and server-side data fetching.
17. How do you manage side effects in your React applications?
18. What's the difference between Redux Thunk and Redux Saga?
19. How would you use GraphQL in a React application, and why?
20. Explain how you would handle code splitting and lazy loading in a React application.

Performance Optimization & Testing

21. How do you profile and optimize frontend performance?
22. Describe how you would optimize images, fonts, and assets for a high-performance application.
23. What tools do you use for unit and integration testing in frontend applications?
24. Explain end-to-end testing and tools like Cypress or Selenium.
25. How do you set up CI/CD pipelines for frontend applications?
26. What are some strategies for handling large datasets on the frontend?
27. How would you analyze and optimize for SEO in a React application?
28. How do you handle security concerns, like XSS and CSRF, in frontend applications?
29. What's your approach to managing micro-frontends?
30. Explain Webpack, its configuration, and how you would use it for optimization.