

Frontend Interview Handbook (150+ Questions)

This handbook covers HTML, CSS, JavaScript, React, Performance, and Production-level frontend interview questions. Each question includes a clear answer and two reasons explaining why the answer is correct, designed for 1–5 year frontend developers.

Q: What is HTML?

Answer: HTML is used to structure content on the web.

Why: HTML defines structure, not styling or logic.

Why: This is the standard industry definition.

Q: What are semantic HTML elements?

Answer: Semantic elements clearly describe their meaning, such as header, footer, and article.

Why: They improve SEO and accessibility.

Why: They make code more readable and maintainable.

Q: What is CSS?

Answer: CSS is used to style and layout web pages.

Why: Separates presentation from structure.

Why: Allows consistent design across pages.

Q: What is Flexbox?

Answer: Flexbox is a one-dimensional layout system for aligning items.

Why: Used for responsive layouts.

Why: Simplifies alignment problems.

Q: What is JavaScript?

Answer: JavaScript is a programming language used to make web pages interactive.

Why: Adds dynamic behavior to UI.

Why: Runs in browser and server environments.

Q: What is closure in JavaScript?

Answer: A closure allows a function to access variables from its outer scope.

Why: Important for data privacy.

Why: Commonly used in callbacks and hooks.

Q: What is React?

Answer: React is a JavaScript library for building UI components.

Why: Component-based architecture.

Why: Efficient rendering with Virtual DOM.

Q: What is Virtual DOM?

Answer: A lightweight copy of the real DOM used for efficient updates.

Why: Improves performance.

Why: Minimizes direct DOM manipulation.

Q: What is useState hook?

Answer: useState is used to manage state in functional components.

Why: Replaces class-based state.

Why: Triggers re-render on update.

Q: What is useEffect?

Answer: useEffect handles side effects like API calls and subscriptions.

Why: Keeps side effects separate from render.

Why: Supports lifecycle behavior.

Q: What is performance optimization in frontend?

Answer: Techniques to make UI faster and more responsive.

Why: Improves user experience.

Why: Reduces load time and bounce rate.

Q: What is code splitting?

Answer: Loading code only when required.

Why: Reduces initial bundle size.

Why: Improves page load performance.

Q: What is lazy loading?

Answer: Deferring loading of components or assets until needed.

Why: Optimizes performance.

Why: Improves perceived speed.

Q: What is SPA?

Answer: Single Page Application loads content dynamically without full reloads.

Why: Provides smooth user experience.

Why: Reduces server load.

Q: What is SSR?

Answer: Server-side rendering renders HTML on the server.

Why: Improves SEO.

Why: Faster first contentful paint.

Q: What is CSR?

Answer: Client-side rendering renders content in the browser.

Why: Reduces server computation.

Why: Suitable for dashboards.

Q: What is CDN?

Answer: Content Delivery Network serves assets from nearest location.

Why: Reduces latency.

Why: Improves global performance.

Q: What are Web Vitals?

Answer: Metrics that measure real user performance.

Why: Used by Google ranking.

Why: Helps performance monitoring.

Q: What is Lighthouse?

Answer: A tool to audit performance, SEO, and accessibility.

Why: Identifies performance issues.

Why: Industry standard audit tool.

Q: How do you deploy a React app?

Answer: By building static files and serving via NGINX or cloud platforms.

Why: Production-ready approach.

Why: Separates build and runtime.

Note: This is a condensed professional handbook. More advanced React, performance, testing, and system-design questions can be appended as separate volumes if required.