Coding Interview — Frontend

Coding Interview | Frontend

Live coding scorecard for frontend: component building, state management, and accessibility.

Evaluation Criteria

Problem Decomposition & Approach

Evaluates how the candidate breaks down the problem before writing code.

Rating: □ 1 □ 2 □ 3 □ 4 □ 5

Sample Questions:

- Before you start coding, walk me through how you'd approach this problem.
- What components would you create and how would they interact?
- What edge cases should we consider?

✓ STRONG SIGNAL

Candidate asks clarifying questions, identifies requirements and edge cases upfront, and outlines a component hierarchy before coding. They communicate their plan before diving in.

✕ WEAK SIGNAL

Candidate starts coding immediately without understanding the problem, doesn't ask clarifying questions, or cannot break the UI into logical components.

Component Architecture & State Management

Assesses the quality of component design, prop interfaces, and state management decisions during live coding.

Rating: □ 1 □ 2 □ 3 □ 4 □ 5

Sample Questions:

- Why did you choose to put that state in this component vs. lifting it up?
- How would this component be reused in a different context?
- What would change if we needed to add feature X to this component?

✓ STRONG SIGNAL

Candidate creates well-scoped components with clean prop interfaces, places state at the appropriate level, and separates concerns between presentational and container logic. Components are naturally extensible.

✕ WEAK SIGNAL

Candidate builds one monolithic component, passes excessive props, or puts all state at the top level without justification. No thought given to reusability.

CSS & Layout Implementation

Evaluates ability to implement visual designs using CSS with proper techniques.

Rating: □ 1 □ 2 □ 3 □ 4 □ 5

Sample Questions:

- How would you make this layout responsive?
- Can you implement this design without using absolute positioning?

- How would you handle text overflow in this container?

✓ STRONG SIGNAL

Candidate uses modern CSS confidently (flexbox, grid, custom properties), writes maintainable styles, handles responsive behavior, and considers cross-browser edge cases.

✕ WEAK SIGNAL

Candidate cannot center a div, relies on magic numbers and absolute positioning for everything, or is unable to implement basic responsive behavior.

Code Quality & JavaScript Fluency

Assesses the quality, readability, and correctness of the code written during the interview.

Rating: □ 1 □ 2 □ 3 □ 4 □ 5

Sample Questions:

- Is there a way to simplify this logic?
- How would you name this function to make the code more readable?
- What would happen if the API returns an error here?

✓ STRONG SIGNAL

Candidate writes clean, readable code with meaningful variable names. They handle loading, error, and empty states. Code is well-organized and demonstrates fluency with modern JavaScript/TypeScript.

✕ WEAK SIGNAL

Candidate's code is messy, uses single-letter variable names, ignores error states, or demonstrates fundamental misunderstanding of JavaScript patterns.

Accessibility & User Experience Awareness

Evaluates whether the candidate considers accessibility and UX without being prompted.

Rating: □ 1 □ 2 □ 3 □ 4 □ 5

Sample Questions:

- How would a screen reader user interact with this component?
- What keyboard interactions should this support?
- Are there any ARIA attributes we should add?

✓ STRONG SIGNAL

Candidate uses semantic HTML naturally, considers keyboard navigation, adds ARIA attributes where needed, and thinks about focus management. They mention a11y without being prompted.

✕ WEAK SIGNAL

Candidate uses only div and span elements, ignores keyboard accessibility, or has never heard of ARIA roles.

Debugging & Iteration

Assesses how the candidate handles bugs, feedback, and iterative changes during the coding session.

Rating: □ 1 □ 2 □ 3 □ 4 □ 5

Sample Questions:

- Something's not working as expected — how would you debug this?
- Let's add a new requirement: how would you modify your approach?
- What would you refactor if you had more time?

✓ STRONG SIGNAL

Candidate debugs methodically using browser tools, reads error messages carefully, and adapts to new

✕ WEAK SIGNAL

Candidate makes random changes hoping something works, cannot read error messages effectively, or

requirements without panic. They identify refactoring opportunities proactively.

becomes flustered and shuts down when requirements change.

Red Flags

- Cannot build a basic interactive component (button, form, toggle) from scratch
- Copies code patterns without understanding them — cannot modify when requirements change
- Refuses to discuss or consider accessibility as part of frontend development

Notes & Overall Recommendation

☐ Strong Hire ☐ Hire ☐ No Hire ☐ Strong No Hire

Notes: _____