# Technical Phone Screen — Frontend

Phone Screen | Frontend

Evaluate frontend fundamentals: DOM, CSS, React/Vue patterns, and browser APIs.

## Evaluation Criteria

### HTML & CSS Fundamentals

Evaluates understanding of semantic HTML, CSS layout models, and responsive design principles.

Rating: ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

**Sample Questions:**

- Explain the difference between flexbox and CSS grid — when would you use each?
- How does the CSS specificity system work, and how do you manage it in large codebases?
- What is the purpose of semantic HTML, and how does it affect accessibility?

✓ STRONG SIGNAL

Candidate explains layout models with precision, discusses trade-offs between approaches, and mentions accessibility implications of markup choices. They reference real scenarios from their work.

✕ WEAK SIGNAL

Candidate confuses flexbox and grid, cannot explain specificity beyond 'use !important', or shows no awareness of semantic HTML or accessibility.

### JavaScript Core Concepts

Assesses depth of JavaScript knowledge including closures, async patterns, and the event loop.

Rating: ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

**Sample Questions:**

- Explain how closures work and give a practical example of when you'd use one.
- What is the difference between Promises, async/await, and callbacks? When would you choose each?
- How does the JavaScript event loop work, and why does it matter for UI performance?

✓ STRONG SIGNAL

Candidate demonstrates deep understanding of JS internals, explains concepts with concrete examples, and connects them to real-world UI scenarios like avoiding jank or managing state.

✕ WEAK SIGNAL

Candidate gives textbook definitions without practical application, confuses async patterns, or cannot explain how the event loop relates to rendering.

### Framework & Component Patterns

Evaluates experience with modern frontend frameworks (React, Vue, Angular) and component architecture.

Rating: ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

**Sample Questions:**

- How do you decide when to split a component into smaller pieces?
- Explain your approach to state management — when do you use local state vs. global state?
- How do you handle side effects in your components?

✓ STRONG SIGNAL

Candidate articulates clear principles for component decomposition, understands the trade-offs of different state management approaches, and can discuss lifecycle/effect patterns with nuance.

✕ WEAK SIGNAL

Candidate cannot explain component architecture beyond basic syntax, puts all state in global store without reasoning, or shows no awareness of rendering performance.

### Browser APIs & Performance

Assesses knowledge of browser APIs, rendering pipeline, and frontend performance optimization.

Rating: □ 1 □ 2 □ 3 □ 4 □ 5

**Sample Questions:**

- What strategies do you use to improve page load performance?
- How would you debug a page that feels sluggish when scrolling?
- What browser APIs have you used beyond the basics, and for what purpose?

✓ STRONG SIGNAL

Candidate discusses code splitting, lazy loading, image optimization, and can explain the critical rendering path. They mention specific tools like Lighthouse, DevTools Performance tab, or Web Vitals.

✕ WEAK SIGNAL

Candidate has no strategy beyond 'make fewer API calls', cannot describe browser rendering pipeline, or has never used performance profiling tools.

### Testing & Quality Practices

Evaluates the candidate's approach to testing frontend code and ensuring UI quality.

Rating: □ 1 □ 2 □ 3 □ 4 □ 5

**Sample Questions:**

- How do you decide what to test in a frontend application?
- What's your experience with unit tests vs. integration tests vs. E2E tests for UI?
- How do you handle visual regression testing?

✓ STRONG SIGNAL

Candidate describes a testing strategy with clear rationale for what belongs in unit vs. integration vs. E2E tests. They mention testing user behavior over implementation details.

✕ WEAK SIGNAL

Candidate has no testing experience, tests only implementation details (e.g., checking internal state), or dismisses testing as unnecessary for frontend.

## Red Flags

- Cannot write basic CSS layouts without copy-pasting from Stack Overflow
- No understanding of how the browser renders a page or what the DOM is
- Claims 5+ years of React experience but cannot explain hooks or component lifecycle

## Notes & Overall Recommendation

□ Strong Hire □ Hire □ No Hire □ Strong No Hire

Notes: _____