

Front End Developer interview questions

careertips.com.au: common questions and what they're really testing

What to expect

Front end developer interviews usually combine a look at your practical build skills with questions about how you work inside a team that includes designers, backend engineers and product managers. Expect at least one live or take-home coding component alongside the conversation.

- **Technical:** Questions on HTML, CSS, JavaScript and React fundamentals, often extended into a live coding exercise or code review of a small component.
- **Process:** Walk-through questions on how you approach performance optimisation, cross-browser testing or API integration in a real build.
- **Behavioural:** Past-experience questions about working with UX designers, handling design changes late in a sprint, or resolving disagreements over implementation.
- **Scenario:** Judgement questions about trade-offs, such as shipping quickly versus fixing a browser compatibility issue, or how you'd handle a slow-loading page under deadline pressure.

Most interviews open with a short discussion of your background and recent projects, move into a technical segment (live coding, pair programming or a code walkthrough of your portfolio), then cover behavioural and scenario questions, and close with your questions about the team's tech stack and delivery process.

1. Walk me through how you'd optimise a web page that's loading slowly on mobile.

Why they ask

Performance optimisation is a listed core task, and interviewers want to see a structured diagnostic approach rather than a single guess.

How to structure your answer

Process walk-through: state how you'd measure the problem first, then list the likely causes in order of impact, then the fixes you'd apply and how you'd confirm improvement.

Example answer

"I'd start by profiling the page in Chrome DevTools to see what's actually slow: network waterfall, render-blocking resources, or JavaScript execution time. If it's asset-heavy, I'd look at image compression and lazy loading first since that's usually the biggest win on mobile. If it's bundle size, I'd apply code-splitting so we're not shipping code the user doesn't need on first load, and check whether caching headers are set correctly for repeat visits. Once I've made changes, I'd re-run the same profiling and check Core Web Vitals to confirm the fix actually moved the numbers, not just that the page feels faster."

2. Tell me about a time a design handed to you wasn't feasible to build exactly as specified. How did you handle it?

Why they ask

Collaborating with UX designers on implementing specifications is a core task, and this checks how you manage the tension between design intent and technical reality.

How to structure your answer

STAR: situation, task, action, result.

Example answer

"On one project the designer had specified a custom scroll animation that didn't perform well on lower-end mobile devices during testing. My task was to keep the interaction feeling similar without

the performance cost. I went back to the designer with a recorded comparison showing the jank on a mid-range phone, and we agreed on a simplified transition using CSS instead of a heavier JavaScript library. The result was an interaction that felt close to the original design and ran smoothly across the devices we tested, and the designer appreciated seeing the trade-off explained rather than just told no."

3. How do you approach testing a new feature across different browsers and devices?

Why they ask

Cross-browser and device testing is explicitly listed as a core task and is a common point of failure for junior developers who only test in one browser.

How to structure your answer

Process walk-through: describe your testing checklist and priority order.

Example answer

"I start with the browsers and devices that match our actual user analytics rather than testing everything equally. I check layout and functionality in Chrome and Safari first since they cover most mobile and desktop traffic, then Firefox and Edge. For responsive behaviour I use Chrome DevTools device emulation for a first pass, then test on a real phone before calling it done, since emulation misses touch-specific issues. I also check keyboard navigation and screen reader basics as part of the same pass, since accessibility issues often show up alongside layout bugs."

4. You're integrating a new API and the backend response doesn't match what the front end expects. What do you do?

Why they ask

API integration to display dynamic data is a listed core task, and this scenario checks how you handle a common real-world mismatch under time pressure.

How to structure your answer

Scenario judgement: describe your immediate response, who you'd involve, and how you'd avoid blocking the rest of the team.

Example answer

"First I'd check whether it's a documentation issue or an actual bug, by testing the endpoint directly rather than assuming. If it's a mismatch, I'd flag it to the backend developer with the specific fields and expected shape so it's a quick fix on their end rather than a vague bug report. While that's being sorted, I'd build the component against a mocked version of the expected response so I'm not blocked, and swap in the real API call once it's fixed. I'd also raise it in stand-up if it's likely to affect the sprint timeline."

5. What's your experience with React, and how do you decide when to break a UI into separate components?

Why they ask

React is a named tool for this role and this question checks practical, applied knowledge rather than just familiarity.

How to structure your answer

Technical explanation with a concrete example.

Example answer

"I break components down around a single responsibility and reuse, so if a piece of UI like a card or a form input shows up in more than one place, it becomes its own component with clearly defined props. I try to keep state as local as possible and only lift it up when multiple components need to share it, since over-centralising state early makes components harder to test and reuse. On a recent

build I split a large product page into a product image gallery, a details panel and a reviews section, each testable on its own, which made it much easier to debug a rendering issue that turned out to be isolated to the gallery component.”

6. How do you use Git and code review in your day-to-day work?

Why they ask

Git and agile delivery are both listed as core tools and skills, and this checks whether you can work within a team's existing workflow rather than solo.

How to structure your answer

Process walk-through with a specific habit or example.

Example answer

“I work in small, focused branches so each pull request is easy to review, and I write commit messages that explain why a change was made, not just what changed. Before requesting review I check my own diff for anything I've left in by accident, like console logs or commented-out code. When reviewing others' work I try to give specific, actionable feedback rather than just approving, and I've found that catching small inconsistencies early, like a component not matching our existing naming conventions, saves a lot of cleanup later in the sprint.”