

L2 Standards - Web Development

Benchmarks

#	Benchmark	What to look for
1	Independent Contributions	Has shipped multiple features with minimal mentor guidance. Ask how much direction they needed and check PR history.
2	Code Review	Has reviewed multiple PRs with meaningful, specific feedback. Check the review comments — approvals without comment do not count.
3	Feature Scoping	Given a task description, can break it into steps and ship it. Ask them to walk through how they planned their most recent contribution.
4	Code Quality	Writes code others can read — consistent with project conventions, well-named, not needlessly complex. Check the diff directly.
5	Debugging	Has investigated and resolved a problem they did not introduce. Ask them to describe a specific debugging session in detail.
6	System Understanding	Can describe how their contributions fit into the larger codebase — what a feature touches and what depends on it. Test with specific questions about their own PRs.
7	Implementation Tradeoffs	Can explain what they built, what they considered and rejected, and why. Look for judgment, not just execution.
8	Portfolio Growth	L2 work is meaningfully more complex or more independent than L1 work. Compare the two directly — if the bar is ambiguous, it is not a pass.
9	JavaScript Fluency	Handles async patterns, module/component structure, and data flow without looking up fundamentals. Test live if there is any doubt.

Assessment Process

1. Portfolio Review

Ask the student to share their portfolio before the conversation. At L2, this should include at least two merged contributions showing independence and growth over L1 work, plus links to PRs where they left code review feedback.

Look for:

- At least two contributions, each more than minor changes
- At least one completed with minimal mentor involvement
- Multiple PRs reviewed as reviewer, each with substantive feedback
- Code quality consistent across contributions — not polished in one PR, messy in another

For members — how to maintain your portfolio: Include your merged PRs and links to PRs you reviewed. For each contribution, add a brief note on what it was and how much guidance you had. Your code review comments are part of your portfolio.

2. Structured Conversation (15–20 min)

Walk through their most complex contribution and their review history. The bar at L2 is explaining what their code does and the reasoning behind the choices they made.

Sample prompts:

- "Walk me through your most recent contribution. What guidance did you get, and what did you figure out on your own?"
- "You handled X this way — what did you consider and reject before landing here?"
- "Tell me about a PR you reviewed. What feedback did you leave and what happened?"
- "How does [their feature] fit into the project? What does it depend on? What depends on it?"
- "Describe a problem you debugged that you did not introduce. How did you approach it?"
- "What would break if you changed X?"

A student who cannot explain the reasoning behind their own structural choices is not at L2.

3. Peer Signal

Ask 2–3 teammates who have worked with this student directly:

- Has this person reviewed your code? Was the feedback useful?
- Have you worked through a problem together? Can they carry their own weight?

- Can you name something they helped you understand or improve?
-

Outcome

Pass: student meets all 9 benchmarks and peer signal is positive. Record in the shared assessment sheet (assessor, date, level, notes) and in Basecamp when that feature is available.

Not yet: name the specific gaps with actionable feedback. Common ones: code review participation (assign them to review an open PR now), independence (needs one more unguided contribution), system understanding (schedule an architecture walkthrough first).

Revision #1

Created 11 June 2026 17:50:19 by Drew Beamer

Updated 11 June 2026 17:51:10 by Drew Beamer