L1 Standards - Web Development

Benchmarks

#	Benchmark	What to look for
1	Functional Contribution	At least one merged PR to a YETI project (Juice, scouting site, tba-sdk, Basecamp, or equivalent) involving JavaScript/TypeScript logic. CSS-only or single-line changes do not count.
2	Code Authorship	Can explain what their contribution does and why it was structured that way. Walk through the code with them — if they cannot explain it, do not pass.
3	JavaScript Fundamentals	Can read and write variables, functions, loops, conditionals, and async/await without looking them up. Test live if there is any doubt.
4	Project Literacy	Understands the structure of the project they worked in — what the major folders and files are for and how the app fits together. Ask them to orient you.
5	Local Development	Can set up a local dev environment and run the project. Ask how they got it running the first time.
6	Code Consistency	Followed existing code style and patterns without repeated correction. Check the PR diff directly.
7	Guided Execution	Completed their contribution with mentor guidance. Independence is not required at L1 — verify that the scope was appropriate for a first contribution.

Assessment Process

1. Portfolio Review

Ask the student to share their portfolio before the conversation. At L1, this is a link to at least one merged PR or deployed feature in a YETI project.

Look for:

- Involves JavaScript/TypeScript logic (not just CSS or markup changes)
- Student is the primary author
- Scope appropriate for a guided first contribution — a feature, a bug fix, a UI change

If the contribution is CSS-only or a single-line change, it does not meet the bar — ask the student to ship something functional before scheduling the assessment.

For members — how to maintain your portfolio: Your portfolio at L1 is a link to your merged PR(s). Keep a note on what you built and what guidance you received. Before your assessment, be ready to walk an evaluator through what you shipped and why.

2. Structured Conversation (15–20 min)

Walk through their contribution together. The conversation serves as the competency check — if they can explain their own code, authorship is confirmed. Syntax recall is not the bar.

Sample prompts:

- "Tell me about the feature you shipped. What problem does it solve?"
- "Walk me through what this code does." (pick a specific section from their PR)
- "Why did you structure it this way? Did you consider anything else?"
- "What would happen if you removed this part?"
- "How did you set up and run the project locally?"
- "How did you know when the feature was working?"

If the student cannot explain code attributed to them, do not pass.

3. Reliability Check

Ask a mentor or student who worked with them directly this season:

- Did they follow through on tasks assigned to them?
- Did they communicate when they were stuck or something slipped?

Outcome

Pass: student meets all 7 benchmarks. 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 concrete feedback. "You hit 6 of 7. Benchmark #1: the CSS change you submitted does not meet the functional scope bar. Ship a feature that involves JavaScript logic and come back."

Revision #3
Created 2026-06-11 17:48:07 UTC by Drew Beamer
Updated 2026-06-11 17:50:10 UTC by Drew Beamer