

COMPETENCY EVIDENCE BRIEF

Testing Across Multiple Browsers

A consistent browser strategy for Google Chrome, Mozilla Firefox and Microsoft Edge.

Project: Learning Planner **Date:** 14 August 2026 **Status:** Edge session pending

Objective

Show that the same Learning Planner workflows and responsive layouts behave consistently in Google Chrome, Microsoft Edge and Mozilla Firefox, with reproducible developer-tools evidence and compatibility remediation.

Approach

Common scenario: Each browser starts from the same seeded dataset and executes persisted goal creation, task search/status, storage reload and overflow checks. This separates browser differences from test-data differences.

Coverage: Playwright runs the installed Google Chrome channel and a Mozilla Firefox build. A Windows Edge protocol repeats the scenario and captures version, Console, Local Storage and computed Grid evidence.

Work completed

Compatibility design: Used semantic HTML, system fonts, standards-based Grid/Flexbox, visible focus, reduced motion, a `crypto.randomUUID` fallback and validated browser storage.

Developer tools: Console review checks runtime errors; Application/Storage verifies persistence; computed styles expose breakpoint differences; the accessibility tree verifies useful control names and structure.

Regression path: A reproduced issue must record environment and expected/actual behaviour, use the smallest standards-based fix and gain an automated or documented manual retest.

Compatibility concerns Date inputs, font metrics/wrapping, sticky positioning, focus outlines, Grid sizing, downloads/uploads, localStorage and UUID support are explicit checklist items.

Testing performed

Google Chrome: The installed Chrome channel passed the core planning journey, task interaction, overflow check and evidence capture.

Mozilla Firefox: Firefox passed the same functional and overflow checks; its seeded dashboard was captured for direct comparison.

Microsoft Edge: The Windows/manual session remains pending because Edge is not installed on this Linux host. The checklist requires the stable version page, app, Console and Local Storage captures.

Outcome

Chrome and Firefox provide completed, matching automated evidence with no application console errors. Edge is not claimed complete: the document becomes assessor-ready only after the genuine Edge session is logged and its appendix screenshots are added.

Lessons learned

Name browsers precisely: A Chromium engine run helps compatibility, but it is not evidence of the branded Chrome or Edge products. Launching the installed Chrome channel strengthened the result.

Use tools to explain differences: Computed styles, console messages and stored data are more diagnostic than screenshots alone; together they show whether a problem is layout, runtime or persistence.

Prefer progressive enhancement: Feature detection and narrow fallbacks are safer than broad polyfills. Add a prefix/polyfill only for a reproduced support gap and retain a regression test.

Backwards-compatibility response Confirm the support baseline, reproduce in the affected browser, prefer standards and feature detection, add Autoprefixer or a narrowly scoped fallback/polyfill only when needed, then rerun the full scenario.

SCREENSHOT APPENDIX · EXCLUDED FROM TWO-PAGE NARRATIVE LIMIT

BR-01 Google Chrome seeded dashboard

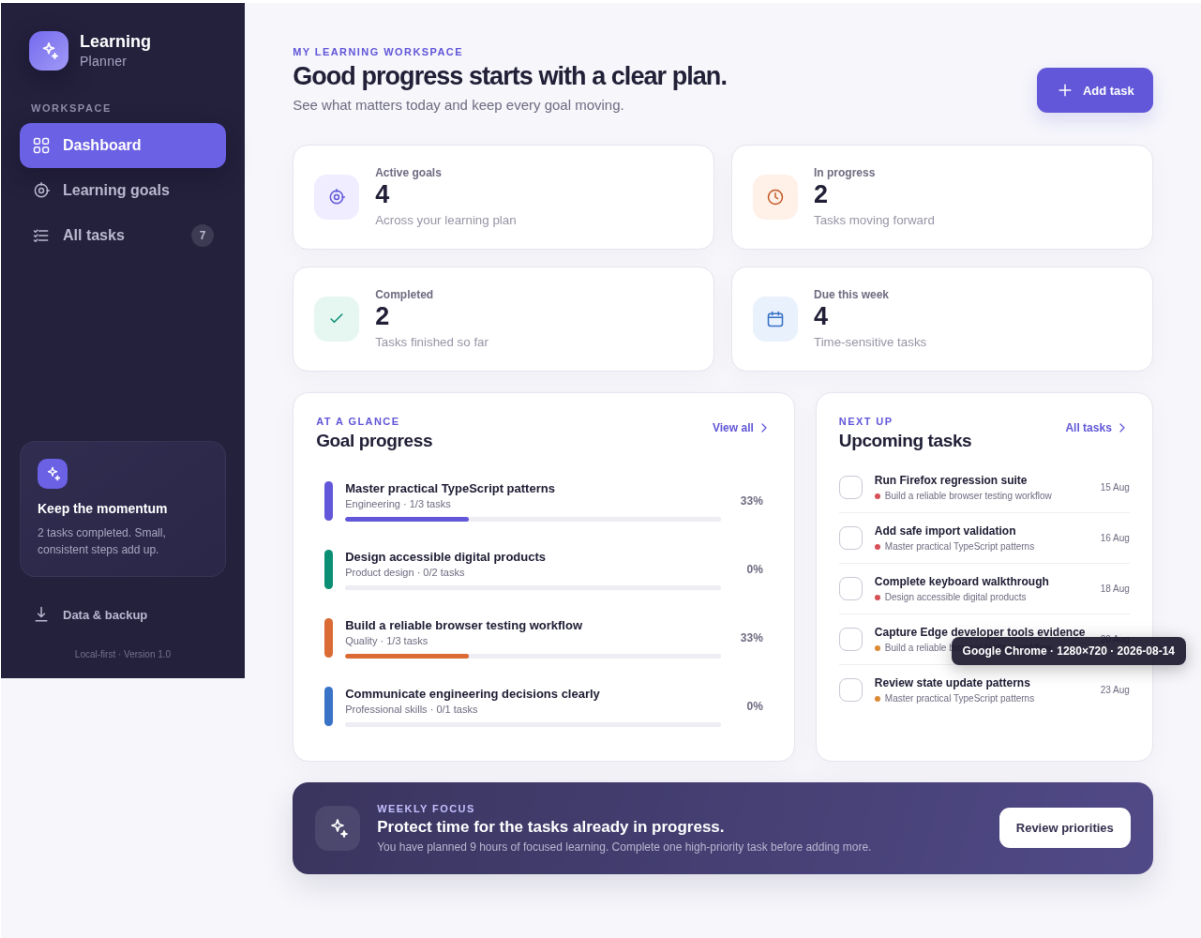


Figure BR-01. Google Chrome seeded dashboard. Generated from non-sensitive sample data.

BR-02 Mozilla Firefox seeded dashboard

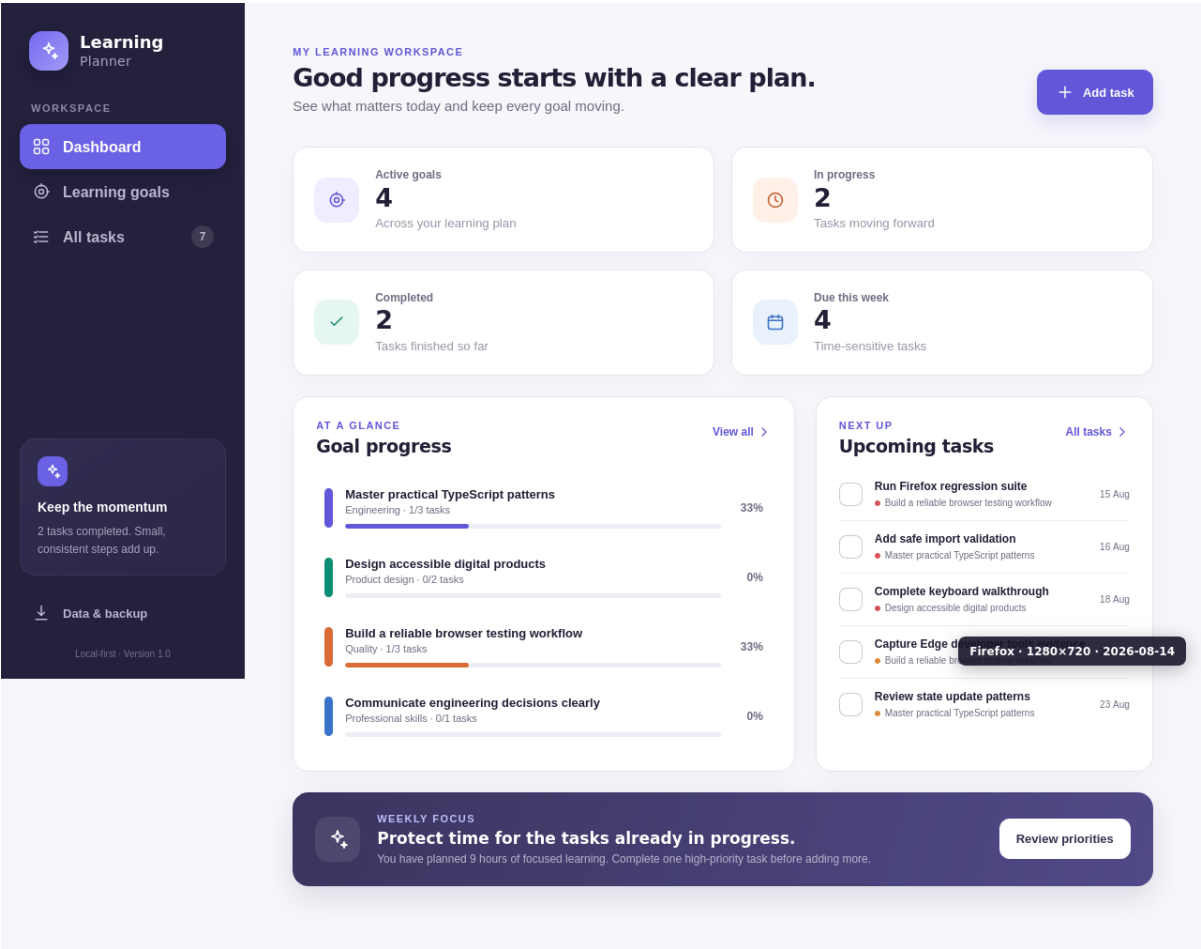


Figure BR-02. Mozilla Firefox seeded dashboard. Generated from non-sensitive sample data.