

COMPETENCY EVIDENCE BRIEF

Managing a Personal Engineering Project

Planning, delivering and evidencing the Learning Planner web application.

Project: Learning Planner **Date:** 14 August 2026 **Status:** Implemented

Objective

Plan and deliver a realistic solo web application while making scope, dependencies, checkpoints, decisions and progress auditable to an engineering assessor.

Approach

Scope: I selected a local-first learning planner: substantial enough for typed state, CRUD, validation, responsive design and test automation, but bounded by excluding accounts, cloud sync and a backend.

Control: I converted the work into P01–P14, linked dependencies, identified the critical path and defined six milestone exit checks. A versioned board snapshot, roadmap, test log and decision log preserve the evidence.

Work completed

Foundation: Created the React/TypeScript/Vite project, quality scripts, stable domain interfaces, a versioned storage adapter and realistic sample data.

Product: Delivered goal/task CRUD, dashboard summaries, search/filter/sort, validation, confirmation, JSON backup/recovery and accessible responsive interactions.

Governance: Recorded acceptance criteria, dependency mapping, milestone status, risks and the RWD-01 decision. Manual Edge/device items remain visibly pending rather than being reported as complete.

Critical path P01 → P02/P03 → P04 → P05 → P06 → P07 → P10 → P11 → P12 → P13 → P14.
Search/filter and backup proceeded as parallel branches before accessibility and final testing.

Testing performed

Quality gate: Lint, nine unit/component tests, TypeScript compilation and the production build passed.

Browser workflows: Playwright verified persisted goal creation, task search/status changes and horizontal-overflow protection across Google Chrome, Chromium, Firefox and all required automated viewports.

Checkpoint review: I compared implementation against each backlog acceptance criterion and retained named test/evidence IDs. This exposed one visual defect at 1680×1050, which was fixed and retested.

Outcome

The project is runnable from a clean clone, the automated delivery scope is complete and the repository contains a portable audit trail. The assessor-ready release remains gated by genuine Edge and two-physical-device sessions, for which a signed checklist and screenshot naming protocol are supplied.

Lessons learned

Make dependencies explicit: The data model and responsive shell could progress in parallel, but CRUD depended on both; documenting that convergence kept sequencing clear.

Treat evidence as a deliverable: Test IDs, seeded data and screenshot names must be designed early. Retrofitting them would have produced weaker, less repeatable evidence.

Report status truthfully: A viewport emulator is not a physical device and Chromium is not Edge. Visible pending checkpoints are stronger engineering evidence than unsupported completion claims.

Evidence map PM-01 overview and acceptance criteria · PM-02 backlog/dependencies/roadmap · PM-03 test and checkpoint logs · Appendix figures show the delivered application and validated goal workflow.

SCREENSHOT APPENDIX · EXCLUDED FROM TWO-PAGE NARRATIVE LIMIT

PM-04 Evidence-ready application at 1920×1080

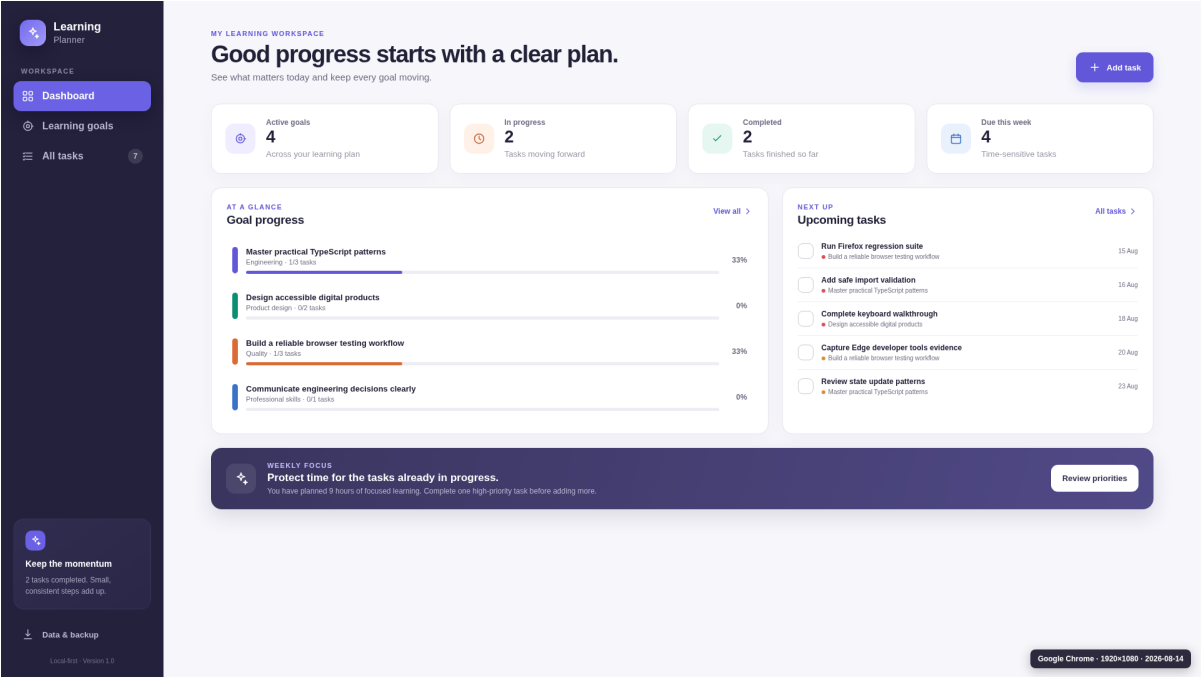


Figure PM-04. Evidence-ready application at 1920×1080. Generated from non-sensitive sample data.

SCREENSHOT APPENDIX · EXCLUDED FROM TWO-PAGE NARRATIVE LIMIT

PM-05 Goal creation workflow at the 1680×1050 checkpoint

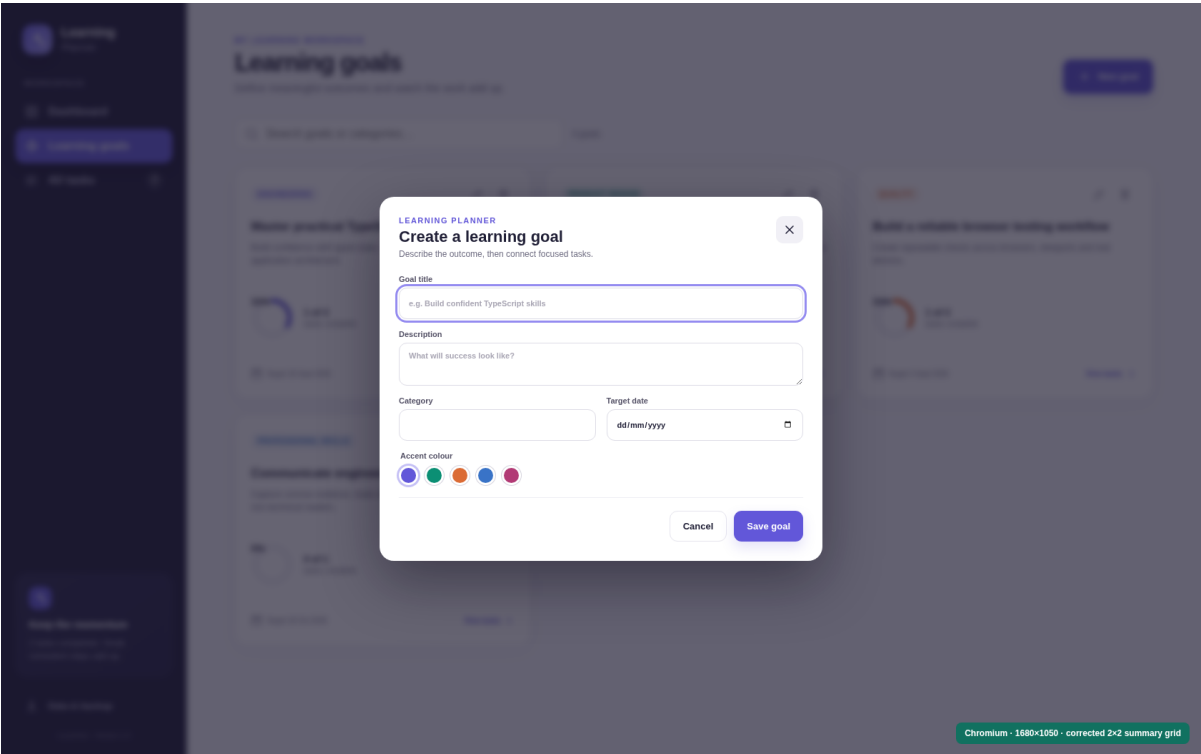


Figure PM-05. Goal creation workflow at the 1680×1050 checkpoint. Generated from non-sensitive sample data.