Personal Reading Tracker Web App

Personal Reading Tracker Web App – BookStack: A Private, Local-First Reading Log

Introduction / Background

Readers who track their reading finish more books — but the popular tracking apps are social networks first, with feeds, ads and data harvesting attached. Students and focused readers often want the opposite: a private, fast, local-first log of what they're reading, how far they've got, and what they thought — with the motivational layer (goals, streaks, stats) and none of the surveillance.

BookStack is a web app built on that premise. Books live on shelves, progress is logged by slider and session, notes anchor to page numbers, and a stats dashboard turns the log into a year-in-reading. All state lives in the browser's IndexedDB: private by architecture, offline-capable, with export/import for backup. Covers and metadata come from the Open Library API, so adding a book is a search, not data entry.

Problem Statement

Existing reading trackers trade privacy and focus for social features, and manual logs (spreadsheets, notebooks) provide no goals, stats or search. Readers need a private, local-first tracker that makes logging effortless and progress visible.

Objectives

1. Organise books on shelves (currently reading, finished, want to read) with one-action status moves.

2. Track progress via page slider and session logging (pages + minutes), deriving pace and estimated finish dates.

3. Attach page-anchored notes and highlights, searchable library-wide.

4. Motivate with a yearly goal ring, monthly bars, streaks and ratings.

5. Keep everything local-first (IndexedDB), offline-capable and exportable — no account, no server.

Proposed Methodology

Static web app (HTML/CSS/JS, no build step). IndexedDB is the system of record with a small data-access layer; the stats engine derives goal progress, monthly aggregates, streaks and averages from sessions and finish events. The Open Library API supplies cover/metadata search with local caching. JSON export/import provides backup and portability. Pace estimation uses recent-session pages-per-day projected over remaining pages, presented as a range for irregular readers rather than a false-precise date.

System Architecture / Working

UI: shelves view, book detail (slider, sessions, notes), stats dashboard (Chart.js). Data: IndexedDB stores (books, sessions, notes, goals). Integration: Open Library search module with cover cache. The offline story: fully functional after first load; covers cached on add. Export/import serialises all stores to one JSON file.

Hardware / Software Requirements

Software: any modern browser; a static host or file:// for deployment; Open Library API access (public, no key). Hardware: any device with a browser — desktop or mobile. No special hardware required.

Expected Outcomes

A working web app managing a personal library with progress, sessions, notes, ratings and a stats dashboard including the yearly goal ring — demonstrated with real book covers from Open Library. Expected secondary outcomes: the IndexedDB schema notes and the honest pace-estimation design.

Applications

Students tracking course and leisure reading; book clubs logging shared reads; researchers maintaining reading notes; anyone wanting a private alternative to social reading apps.

Future Scope

Multi-device sync via the export format; EPUB progress import; reading-challenge modes; annual ‘year in reading’ shareable summary; and lending tracking (who borrowed which book).