

Project Title: Library Management

Problem Statement

Tracking and managing book collections manually or through paper records can be slow and error-prone. Individuals such as students, hobbyists, or small libraries often need a lightweight digital solution to register books, quickly view details, and edit information without installing large applications or managing servers. The **Lightweight Library Manager** addresses this challenge by delivering an in-browser CRUD application created entirely with HTML, CSS and JavaScript, using the browser's local storage for persistence.

Requirement Gathering

- Pinpoint the fundamental user actions (create, display, modify and remove book entries).
- Establish which input fields must always be present (Title, Author, Year, ISBN).
- Choose the data storage approach (client-side local storage versus server-side storage).
- Gather expectations for the interface (form layout, data table, action buttons).
- Lay out the file organisation (HTML for content, CSS for styles, JS for logic).
- Specify non-functional aspects such as offline availability and a straightforward UI.

Stakeholder Analysis

- Identify the main audience (students, small-scale libraries, demonstration users).
- List secondary audiences (developers studying CRUD functionality).
- Consider the technical comfort of stakeholders (simple, server-free operation).
- Collect input on potential enhancements (additional fields, modal dialogues).
- Recognise constraints (browser-based, limited dataset size).
- Define a roadmap for potential upgrades (multi-user support, server APIs).

Business Process Mapping

- Illustrate how a new book moves through the system (form submission → JavaScript logic → local storage → on-screen table).
- Explain how the application keeps data intact after reloading (JSON stored locally).
- Detail the update path (select edit → modify data → re-save to storage).
- Describe the removal path (select delete → confirm → erase from storage).
- Set the initial state (empty listing before first entry).
- Note validation steps and user confirmations (field checks, data clearing).

Industry-specific Use Case Analysis

- Relate the app to small library or personal book tracking scenarios.
- Highlight its role as a teaching tool for CRUD operations in classrooms.
- Consider adapting it for inventory or asset tracking beyond books.
- Examine limits of the approach (local storage capacity, single-user mode).
- Discuss privacy and security factors (data confined to the user's device).
- Review how it should display across devices such as phones and desktops.

AppExchange Exploration

- Add extra attributes such as Genre, Publisher or Ratings to enrich records.
- Replace alert prompts with inline editing forms or modal pop-ups.
- Plan migration from browser storage to a lightweight backend or web API.
- Package the app as an open-source starter template for wider reuse.
- Redesign the look and feel with frameworks like Tailwind CSS, Bootstrap or animations.
- Publish or share the tool on platforms such as Salesforce AppExchange or GitHub Pages.