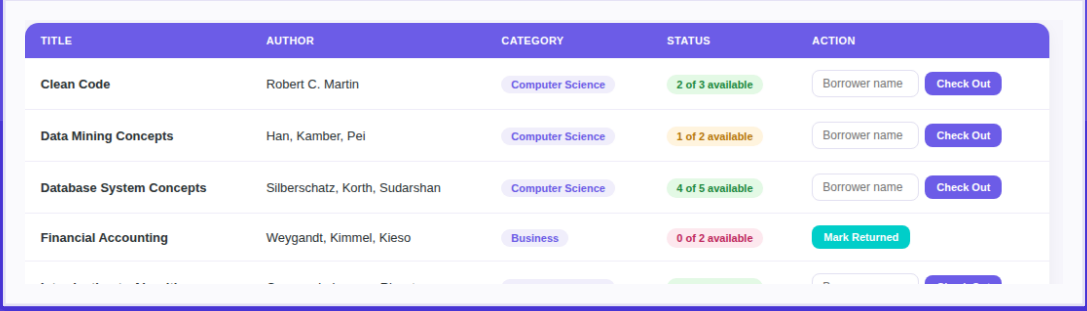
PYTHON | FLASK | SQLITE

Library Management System

A Product Walkthrough

What it is, who it's for, and a tour of every feature



Built by Layan Khayyat

Business Information Technology · Princess Sumaya University for Technology

What Is This System?

The Library Management System is a web application that replaces manual, spreadsheet-based tracking of a book collection with a single, always-up-to-date digital record. It keeps track of every book's total and available copies, records who has borrowed what, and lets staff search and filter the collection instantly instead of scrolling through a spreadsheet.

It was built end-to-end with Python: a Flask web server handles requests and business logic, and a SQLite database stores the books and checkout records permanently, so nothing is lost when the server restarts.

100%	3	2
Real, Working App	Core Actions	Backend Technologies

Who It's For

Library / Resource-Room Staff

Currently tracking inventory in Excel or on paper and need something faster and less error-prone.

Clubs & Student Organizations

Managing a shared collection of books, equipment, or resources that get lent out to members.

Developers Learning Full-Stack

Want a real example of a Python backend (Flask) connected to a database, not just a script.

Why Flask & SQLite

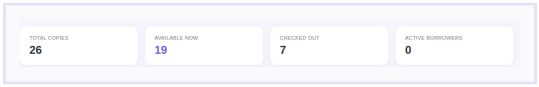
Flask was chosen because it's lightweight and lets the whole system run as a single small Python app with no external services required. SQLite was chosen because it needs no separate database server to install — the entire database lives in one file, making the project simple to set up and reason about.

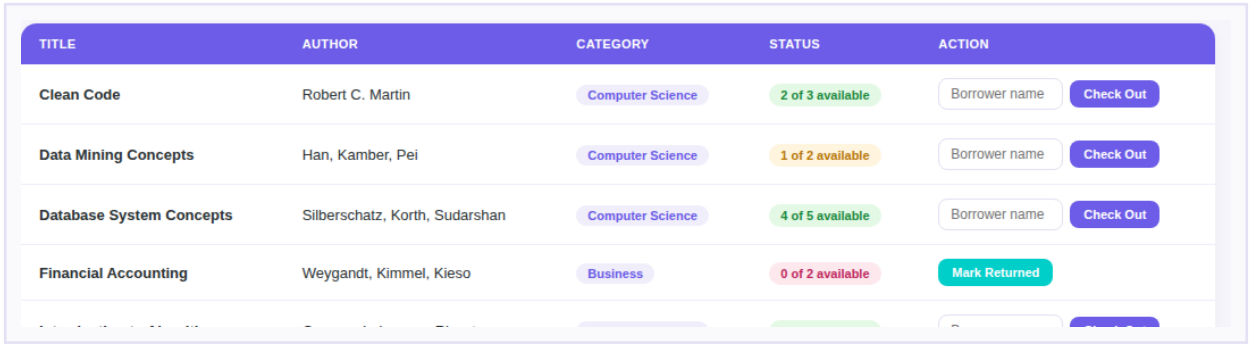
Feature Walkthrough

A tour of the actual, running application — every image below is a real screenshot of the live system, not a mockup.

1. At-a-Glance Dashboard

The home screen opens with four live counters: total copies across the whole collection, how many are available right now, how many are checked out, and how many people currently have something borrowed.

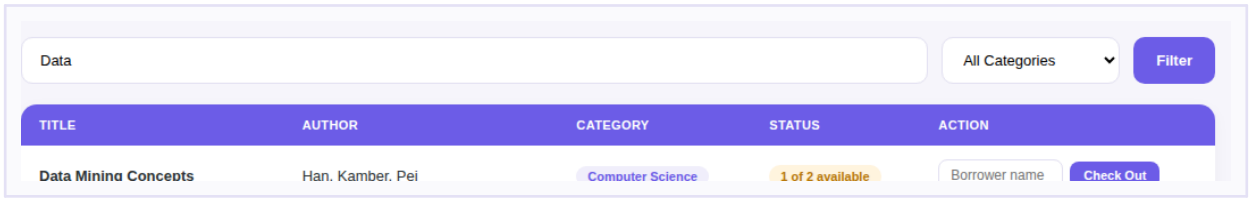




The full catalog, with live status badges: green for well-stocked, amber for low copies, and red/teal when a book is fully checked out.

2. Instant Search

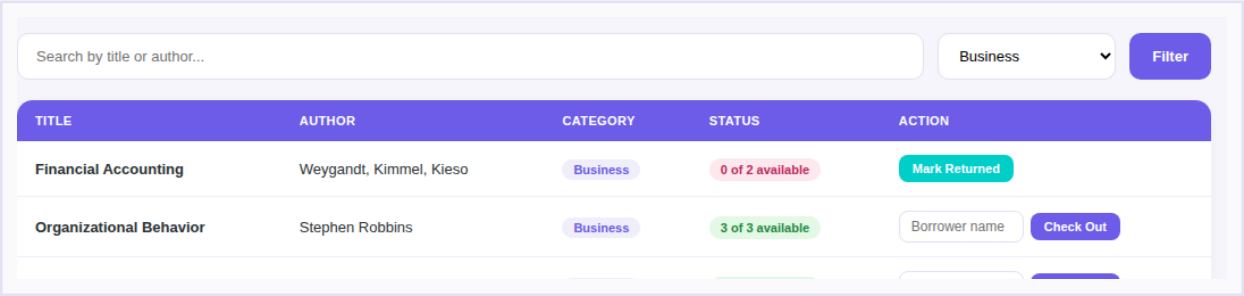
Typing into the search bar filters the book list in real time by title or author — useful the moment someone needs to check if a specific book is on the shelf.



Searching “Data” narrows 8 books down to the 2 matching titles, live as you type.

3. Category Filtering

Books are grouped by category. The dropdown filters the whole list to just that group, helpful for browsing rather than searching a specific title.



Filtering to “Business” shows all 4 business titles, including one fully checked out.

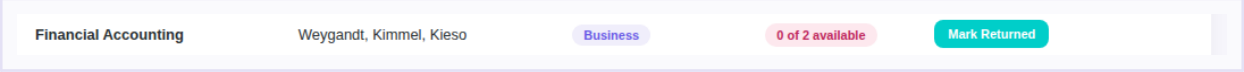
4. Checking a Book Out

Every book with at least one available copy shows a “Check Out” action with a name field. Submitting it immediately reduces that book's available count by one and records the checkout with today's date in the database.

Entering a borrower's name for “Introduction to Algorithms” before checking it out.

5. Returning a Book

When a book has zero available copies, its action column swaps to a “Mark Returned” button. Clicking it closes out that book's oldest active checkout record and adds one copy back to the available count.



“Financial Accounting” has 0 of 2 copies available, so it shows Mark Returned instead of a checkout form.

What's Next

This first version focuses on the core loop — browse, search, check out, return. Natural next steps for a v2:

Accounts & Login	So only authorized staff can check books in and out.
Due Dates & Overdue Alerts	Track when a book is due back and flag it visually when it's late.
Email Reminders	Automatic notices before a book is due, and when it's overdue.
Export to Excel	One-click export of the full catalog or checkout history for reporting.

Built With

Python **Flask** **SQLite** **Jinja2**