

Project Report

Project Title:
Library Management System

Submitted by:
Mahajabin Sultana Joty
Student ID: 11230321245
Email Address: mehjabinjoty8@gmail.com

Course Title: Software Development I
Course Code: CSE 2216
Section: 4B

Submitted to:
Anindya Nag
Lecturer,
Northern University of Business and Technology Khulna

Date of Submission: 17.08.2025

Contents

2. Abstract:.....	1
3. Introduction:.....	1
4. Literature Review:	1
5. Methodology:.....	2
6. Implementation:	3
7. Results and Analysis:.....	4
8. Discussion:.....	6
9. Conclusion:	7
10. References:.....	7

Table of Figure:

Figure 1: Library Management Flowchart.....	2
Figure 2: Create display books method	3
Figure 3: Output of Library Management System Step 1.....	4
Figure 4:Output of Library Management System Step 2 &3.....	5

Library Management System

2. Abstract:

This project details the development of a web-based Library Management System using front-end technologies: HTML, CSS, and JavaScript. The problem addressed is the inefficiency and disorganization associated with manual book management. The system provides a user-friendly interface to perform essential library functions, including adding, searching, deleting, and tracking the status of books (borrowed/returned). All book data is stored persistently in the browser's local storage.

3. Introduction:

- **Background:** The need for an efficient and simple system to manage a personal or small-scale library collection has grown with the rise of digital technologies. Manual methods are prone to errors and are difficult to scale.

- **Problem Statement:** This project addresses the challenge of creating a quick and easy-to-use digital system for managing books without the need for a complex back-end database.

- **Objectives:** The primary objectives are to build a functional front-end application that allows users to:

- Add new book entries with title, author, and ISBN information.
- Display a list of all books in the collection.
- Search for books in real-time.
- Delete books from the system.
- Change a book's status from "borrowed" to "returned".

- **Scope:** The project is a client-side web application. It uses browser-based local storage for data persistence and does not include a back-end server, user authentication, or a full-fledged database.

4. Literature Review: This is a project report for an academic course. Therefore, it does not include an extensive literature review. However, the technologies used—HTML, CSS, and JavaScript—are well-documented and are the standard for modern front-end web development.

5. Methodology:

- Approach: The project follows an agile and iterative development approach, focusing on a front-end solution. The user interface was designed first using HTML and CSS, followed by the implementation of logic with JavaScript.

- Algorithms: [Detail any algorithms or models implemented.]

- Tools and Technologies:

- HTML5: For creating the project's structure.
- CSS3: For styling the interface and providing a clean, modern look.
- JavaScript (ES6): For all application logic, including DOM manipulation and data handling.
- Visual Studio Code: The integrated development environment used for coding.
- Web Browser: Used to run and debug the application.

- Experimental Design:

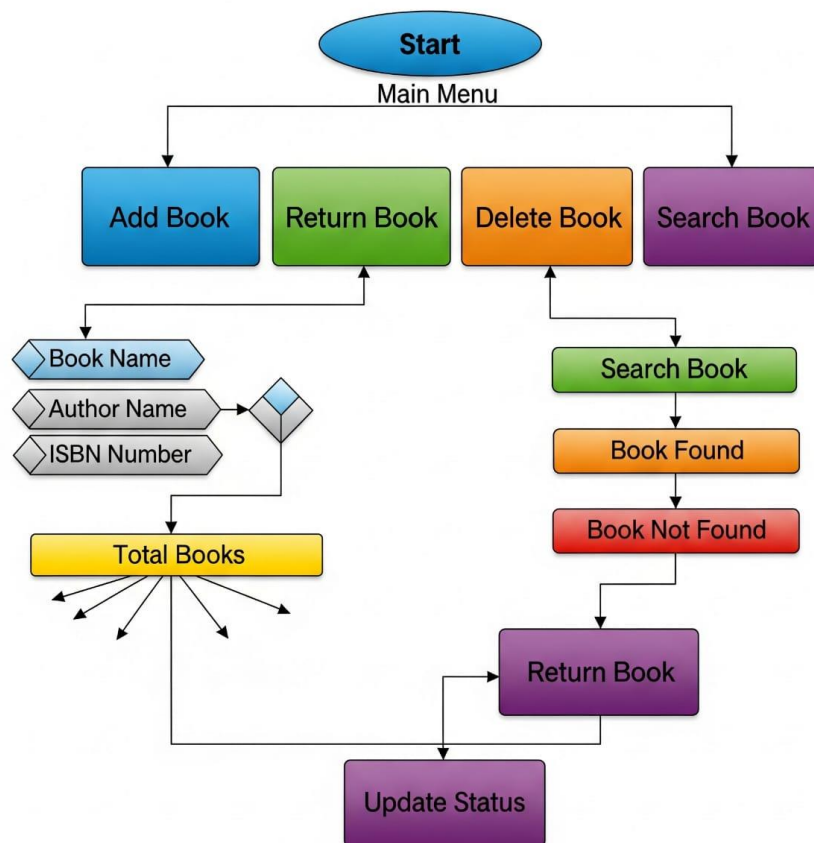


Figure 1: Library Management flowchart

Description: This flowchart illustrates the workflow of a Library Management System.

- Add Book: After entering the book details (name, author, ISBN), the system updates the "Total Books" count and then proceeds to the "Update Status" process.
- Search Book: The search process leads to two possible outcomes: "Book Found" or "Book Not Found." If a book is found, it can be returned, which in turn leads to an "Update Status" process.
- System Updates: The "Total Books" display is a central point, showing that the total count is affected by adding, deleting, and returning books. All main processes (Add, Return, Delete, and Search) eventually lead back to or update the system's status.⁶

Implementation:

- System Architecture: The system architecture is client-side. The index.html file acts as the single page application, the style.css file handles all presentation, and the script.js file handles all logic. Data is stored locally in the browser's local Storage.

- Components:

- HTML Form: The main user input component for adding books.
- Book List : A dynamic list that displays all book entries.
- Search Input: A text field that filters the displayed books in real-time.

- Code Excerpts: The following is a key JavaScript function that handles the dynamic display of books, including the new "returned" status and unique book number.

```
const displayBooks = (bookArray) => {
  bookList.innerHTML = '';
  bookArray.forEach(book => {
    const li = document.createElement('li');
    li.className = 'book-item';
    li.dataset.id = book.isbn;
    if (book.isReturned) {
      li.classList.add('returned');
    }
    li.innerHTML = `
      <div class="book-info">
        <h3>${book.title}</h3>
        <p>By ${book.author}</p>
        <p>ISBN: ${book.isbn}</p>
        <p>Book No: ${book.bookNumber}</p>
        <p class="status">${book.isReturned ? 'Status: Returned' : 'Status: Borrowed'}</p>
      </div>
      <div>
        <button class="return-btn">${book.isReturned ? 'Un-return' : 'Return'}</button>
        <button class="delete-btn">Delete</button>
      </div>
    `;
    bookList.appendChild(li);
  });
};
```

Figure 2: Create display books method

Description: JavaScript (script.js)

This is the brain of the project. It handles all the interactive logic and data management. It does the following:

- Listens for User Actions: It waits for you to click the "Add Book" button, the "Delete" button, or type in the search bar.
- Adds and Manages Books: It takes the data from the form, creates a new book object, and adds it to an array. It also assigns a unique book number and keeps track of whether a book is "borrowed" or "returned."
- Saves Data: It uses the browser's Local Storage to save all the book information. This means the data will not be lost when you refresh the page or close the browser.
- Filters the List: It takes your search input and filters the book list in real time, showing only the books that match your search query.
- Updates the Display: It dynamically creates and updates the HTML to show the current list of books, the total book count, and the status of each book.

7. Results and Analysis:

- Results: Results: The project successfully created a fully functional web application. The system correctly adds and displays new books, updates the total book count, and allows for real-time searching. The "Return" functionality also works as intended, visually updating the book's status.

- Analysis: The local storage solution proved effective for a simple, single-user system. The real-time search and dynamic updates provide a smooth user experience. The modular design with separate HTML, CSS, and JavaScript files made the project easy to manage and extend.

- Snippet:

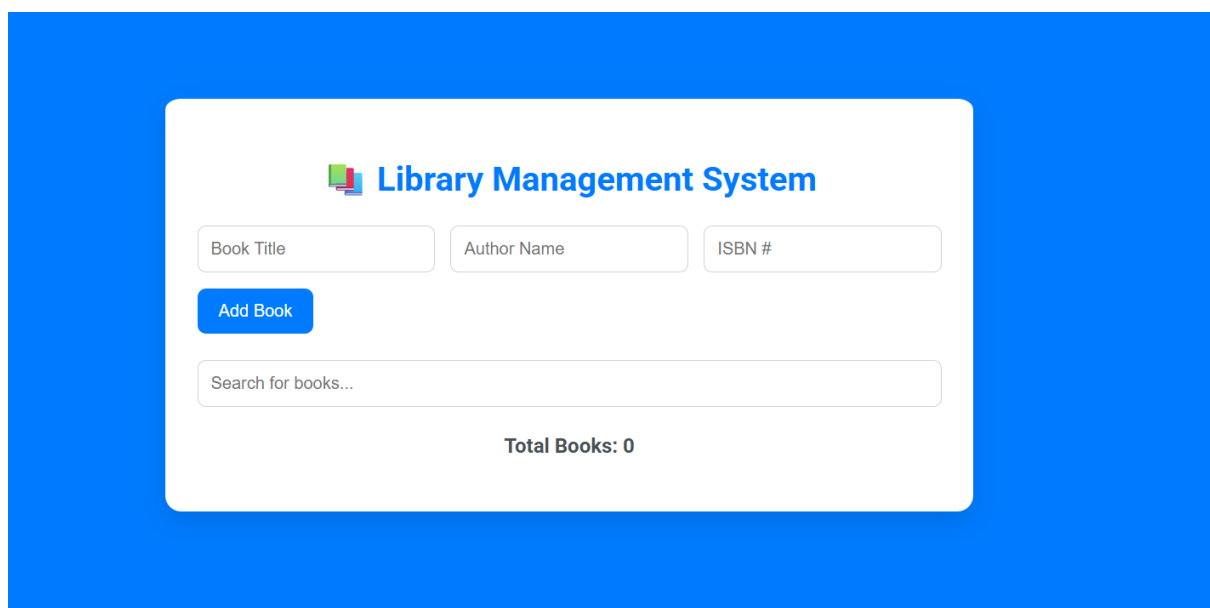
The image shows a web application interface for a Library Management System. It features a blue background with a white rounded rectangle in the center. Inside the rectangle, there is a logo with three colored squares (green, red, blue) to the left of the text "Library Management System". Below the logo, there are three input fields: "Book Title", "Author Name", and "ISBN #". Under the "Book Title" field is a blue button with the text "Add Book". Below the input fields is a search bar with the placeholder text "Search for books...". At the bottom of the white rectangle, it says "Total Books: 0".

Figure 3: Output of Library Management System Step 1.

Description: The screenshot displays a web interface for managing books. It includes input fields for "Book Title," "Author Name," and "ISBN," a button to "Add Book," and a search bar labeled "Search for books..." Below these elements, it shows the "Total Books: 0," indicating the current count in the system.

Library Management System

Book Title: Author Name: ISBN #:

Add Book

Search for books...

Total Books: 3

The C Programming Language

By Thomas H.Cormen
ISBN: 9780262033848
Book No: 5
Status: Borrowed

Return **Delete**

Introduction to Algorithm

By Ronald L.Rivest
ISBN: 978001235568
Book No: 6
Status: Borrowed

Return **Delete**

Anci C

Add Book

Anci C

Total Books: 3

The C Programming Language

By Thomas H.Cormen
ISBN: 9780262033848
Book No: 5
Status: Borrowed

Return **Delete**

Introduction to Algorithm

By Ronald L.Rivest
ISBN: 978001235568
Book No: 6
Status: Borrowed

Return **Delete**

Anci C

By B.Balagurusuami
ISBN: 98753220975
Book No: 7
Status: Returned

Un-return **Delete**

Figure 4: Output of Library Management System step 2 & 3

Description:

The screenshot represents the interface of a Library Management System (LMS), developed to manage books in a digital environment. The system provides an easy-to-use interface where users can add, search, issue, return, and delete books.

At the top of the interface, there are three input fields for Book Title, Author Name, and ISBN Number, followed by an “Add Book” button. This feature allows users to enter new books into the system. A search bar is also provided to quickly locate books by title, author, or ISBN.

Below the search section, the system displays the total number of books available. Each book entry is shown with the following details:

- Book Title
- Author Name
- ISBN Number
- Book Number (unique ID)
- Current Status (e.g., Available or Borrowed)
- In addition, each entry includes two action buttons:
 - Return (Green Button): Updates the status of a borrowed book to returned.
 - Delete (Red Button): Removes the selected book from the system.

The system design uses a blue gradient sidebar with a white content area, making it visually clear and user-friendly. The screenshot also shows example books such as The C Programming Language and Introduction to Algorithm, both marked with the status Borrowed. A sample test entry titled “Anci C” is also visible, likely added for testing purposes. This project demonstrates the core functionalities of a digital library system, including book management, tracking borrowed/returned books, and dynamic display of the collection.

8. Discussion:

- Interpretation: The project fulfills all its core objectives, providing a practical demonstration of front-end development skills. The choice of local storage was appropriate for the scope of a school project, proving effective for data persistence.

- Challenges: A key challenge was ensuring that the data in local storage was correctly synchronized with the on-screen display. Another challenge was creating a robust search function that could filter by multiple book properties.

- Limitations: The system is limited to a single user on a single browser, as local storage is not shared across devices or users. It also lacks a back-end, which would be necessary for a production-level application.

9. Conclusion:

-Summary: This project successfully developed a simple but effective Library Management System. It is a strong example of how HTML, CSS, and JavaScript can be combined to build a dynamic and interactive web application.

-Contributions: The main contribution is a working application that solves a real-world problem on a small scale. It demonstrates proficiency in front-end development, including event handling, DOM manipulation, and data persistence with local Storage.

-Future Work: Future improvements could include-Implementing a back-end with a database to support multiple users and devices. Adding user authentication and login features. Expanding book details to include genres, cover images, and more. Adding an "overdue" status for books that have not been returned on time.

10. References:

1. *W3Schools.com*. (n.d.). https://www.w3schools.com/java/java_ref_hashmap.asp
2. GeeksforGeeks. (2023). Library Management System – Introduction and Design. Available at: <https://www.geeksforgeeks.org/library-management-system-introduction-and-design>
3. ResearchGate. (2022). Design and Implementation of a Library Management System. Retrieved from: <https://www.researchgate.net>