

C. V. Raman Global University, Bhubaneswar
Odisha-752054

Department of Computer Science



A Case Study Report
On
Library Management System

Under The Guidance of
Mamatarani Das & Jyoti Ranjan Swain

SUBMITTED BY:-

Name	Regd. No.
ROJALIN BISWAL	2401080034
GOLDI KUMARI	2401080041
PARVINDER KAUR	2401080044
MANYATA BEHRA	2401080056

TABLE OF CONTENT

S No.	Content	Page No.
01	INTRODUCTION	03
02	ABSTRACT	04
03	ADVANTAGES OF PROPOSED SYSTEM	05
04	ALGORITHM	06-07
05	FLOWCHART	08
06	USE OF HEADER FILE	09
07	CODE	10-12
08	OUTPUT	13-17
09	CONCLUSION	18
10	REFERENCES	19

INTRODUCTION

A **Library Management System (LMS)** is software that facilitates the management of library resources and services. This system allows users to borrow, return, and search for books while keeping track of book inventory, borrowers, and transactions. The implementation of this system uses **HTML, CSS, JavaScript** for the front-end, **C Language** for the back-end, and **Oracle** as the database management system.

Libraries play a crucial role in knowledge dissemination, serving as a hub for students, researchers, and book enthusiasts. Traditional library management relied on manual cataloging and tracking, which often led to inefficiencies, misplaced books, and difficulty in managing records. With the advent of digital systems, library management has evolved to ensure seamless accessibility, accuracy, and security.

The primary objective of an LMS is to create an automated system that reduces human effort while increasing efficiency. The system enables users to search for books using keywords, reserve them in advance, and track borrowing history. Librarians can efficiently manage book stock, issue/return transactions, and generate insightful reports. Additionally, integrating a database ensures that large volumes of data are stored securely and accessed with ease.

By leveraging **C Language** for backend processing, the system offers a fast and robust mechanism to handle user requests and maintain database transactions. The **HTML, CSS, and JavaScript** front-end ensures an intuitive user experience, while **Oracle Database** guarantees secure and structured data management.

ABSTRACT

A **Library Management System (LMS)** is a digital solution designed to streamline and enhance the management of library resources, transactions, and services. Traditional library management relied heavily on manual record-keeping, which often led to inefficiencies such as misplaced books, difficulties in tracking borrowing history, and challenges in maintaining an accurate book inventory. With the advancement of technology, the implementation of an automated LMS has become essential in modern libraries to ensure efficient resource allocation, data accuracy, and an improved user experience.

This case study explores the **design, implementation, and functionality** of a Library Management System developed using **HTML, CSS, and JavaScript** for the front-end, **C Language** for back-end processing, and **Oracle Database** for data storage and management.

The LMS offers multiple **key advantages**, including automation of manual processes, improved data security, faster processing, fine management for overdue books, multi-user accessibility, and remote access capabilities. Librarians benefit from real-time inventory updates, streamlined book issue/return processes, and analytical reporting that helps in effective resource allocation. The system also ensures **data security and integrity**, preventing unauthorized access and loss of critical information.

The **technical implementation** of the system follows a structured **algorithm**, ensuring efficient handling of library transactions. The system architecture is designed to handle multiple users simultaneously, providing role-based access to students, faculty members, and librarians.

Additionally, the case study presents a **flowchart** illustrating the step-by-step working mechanism of the LMS, from user authentication to book issue, return, and inventory updates. The **C program implementation** demonstrates the system's ability to handle library operations dynamically through structured data storage and retrieval techniques. The program effectively utilizes **header files**, including `<stdio.h>`, `<stdlib.h>`, and `<string.h>`, to perform input-output operations, memory allocation, and string manipulations.

ADVANTAGES OF PROPOSED SYSTEM

- ✚ A **Library Management System** is a digital solution designed to organize and manage books, users, and transactions efficiently, ensuring smooth library operations and an improved user experience.
- **Automation:** Eliminates manual bookkeeping and reduces paperwork, making the system more efficient and error-free.
 - **Efficient Book Management:** The system tracks book availability, issue, and return records efficiently, ensuring that users can quickly find and borrow books.
 - **User-friendly Interface:** A web-based interface built with HTML, CSS, and JavaScript makes the system easy to use for both librarians and users.
 - **Data Security:** The Oracle database provides robust security, ensuring that user and book data is protected from unauthorized access and data loss.
 - **Faster Processing:** By utilizing C language for backend processing, the system ensures quick execution of operations such as searching, issuing, and returning books.
 - **Inventory Control:** Librarians can manage book stocks more effectively by adding, removing, or updating book details in real time.
 - **Fine Management:** The system can automatically calculate fines for overdue books, reducing manual effort and ensuring accountability.
 - **Multi-User Access:** The system supports multiple user roles such as students, librarians, and administrators, ensuring access control and security.
 - **Report Generation:** The system provides detailed reports on book transactions, defaulters, and inventory, helping librarians manage resources effectively.
 - **Remote Access:** A web-based system allows users to access library services from anywhere, making it more convenient for students and faculty members.

ALGORITHM

Algorithm for E-Library Management System

Step 1: Start

Step 2: Initialize

- **Declare a structure library to store book details:**
 - **book_name (String)**
 - **author (String)**
 - **pages (Integer)**
 - **price (Float)**
- **Create an array lib[100] to store multiple book records.**
- **Declare variables:**
 - **ar_nm[100] for author name search**
 - **count = 0 to keep track of the number of books**
 - **input to store user choices**

Step 3: Display Menu

- **Continuously display the menu until the user chooses to exit.**
- **Prompt the user for input:**

Menu Options:

- 1. Add book information**
- 2. Display book information**
- 3. List all books by a specific author**
- 4. Show total count of books**
- 5. Exit**

Step 4: Process User Choice

- **Case 1: Add a Book**
 - **Check if the count is less than 100 (library storage limit).**
 - **Prompt user to enter:**
 - **Book name**

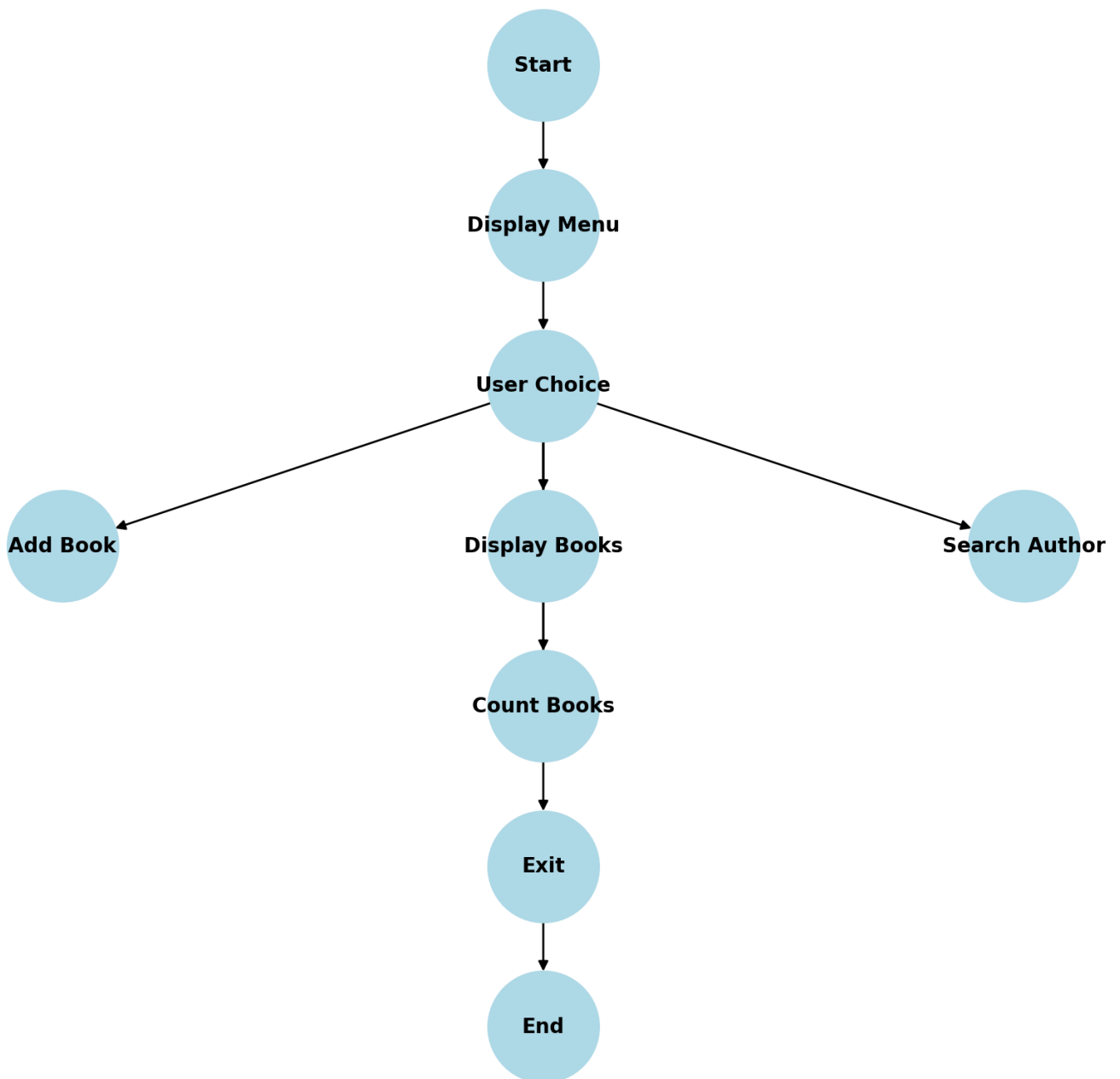
- Author name
- Number of pages
- Price
- Store the book details in lib[count].
- Increment count by 1.
- Display "Book added successfully!".
- **Case 2: Display All Books**
 - If count == 0, display "No books available".
 - Otherwise, iterate over the lib[] array and print book details.
- **Case 3: Search Books by Author**
 - If count == 0, display "No books available to search".
 - Prompt the user to enter an author's name.
 - Iterate through lib[] and compare the stored author names using strcmp().
 - If a match is found, print the book details.
 - If no match is found, display "No books found by this author".
- **Case 4: Count Books**
 - Print the value of count.
- **Case 5: Exit the Program**
 - Print "Exiting program. Thank you!" and terminate execution.
- **Default Case: Invalid Input**
 - Print "Invalid choice! Please enter a valid option."

Step 5: Repeat Until User Chooses to Exit

Step 6: End

FLOWCHART

Flowchart of Library Management System



USE OF HEADER FILE

❖ Here's a brief explanation of the header files used in our C program:

 `#include <stdio.h> :`

Definition:

This is one of the standard input/output header files in C. It stands for "standard input/output" and includes functions like `printf()` for output formatting and `scanf()` for input reading. It is a fundamental header file for basic input and output operations in C.

 `#include <stdlib.h> :`

Definition:

The header file `<stdlib.h>` stands for "standard library." It includes functions involving memory allocation (`malloc()`, `calloc()`, `free()`), random number generation (`rand()`), and other general utility functions. It is an integral part of the C Standard Library.

 `#include <string.h> :`

Definition:

The `<string.h>` header file is used for string manipulation operations. It includes functions like `strcpy()` (copying strings), `strlen()` (calculating string length), `strcat()` (concatenating strings), and others. It is a key component for working with character arrays and strings in C.

CODE :-

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// Structure to store book details
struct library {
    char book_name[100];
    char author[100];
    int pages;
    float price;
};

int main() {
    struct library lib[100]; // Array to store books
    char ar_nm[100]; // For searching author
    int input, count = 0;

    while (1) {
        printf("\n\n*****##### WELCOME TO E-LIBRARY
        *****\n");
        printf("1. Add book information\n2. Display book
        information\n");
        printf("3. List all books of a given author\n");
        printf("4. List the count of books in the library\n");
        printf("5. Exit\n");

        // Get user input
        printf("\nEnter your choice: ");
        scanf("%d", &input);
        while (getchar() != '\n'); // Clear newline character from
        input buffer

        switch (input) {
            case 1:
                if (count >= 100) {
                    printf("Library storage is full!\n");
                    break;
                }

                printf("Enter book name: ");
                fgets(lib[count].book_name, 100, stdin);
```

```

        lib[count].book_name[strcspn(lib[count].book_name,
"\n")] = '\0'; // Remove newline

        printf("Enter author name: ");
        fgets(lib[count].author, 100, stdin);
        lib[count].author[strcspn(lib[count].author, "\n")]
= '\0'; // Remove newline

        printf("Enter number of pages: ");
        scanf("%d", &lib[count].pages);

        printf("Enter price: ");
        scanf("%f", &lib[count].price);
        while (getchar() != '\n'); // Clear buffer

        count++;
        printf("Book added successfully!\n");
        break;

    case 2:
        if (count == 0) {
            printf("No books available in the library.\n");
        } else {
            printf("Book Information:\n");
            for (int i = 0; i < count; i++) {
                printf("\nBook Name: %s\n",
lib[i].book_name);

                printf("Author: %s\n", lib[i].author);
                printf("Pages: %d\n", lib[i].pages);
                printf("Price: $%.2f\n", lib[i].price);
                printf("-----\n");
            }
        }
        break;

    case 3:
        if (count == 0) {
            printf("No books available to search.\n");
            break;
        }

        printf("Enter author name: ");
        fgets(ar_nm, 100, stdin);

```

```

        ar_nm[strcspn(ar_nm, "\n")] = '\0'; // Remove
newline

        printf("Books by %s:\n", ar_nm);

        int found = 0;
        for (int i = 0; i < count; i++) {
            if (strcmp(lib[i].author, ar_nm) == 0) {
                printf("Book Name: %s | Pages: %d | Price:
$%.2f\n", lib[i].book_name, lib[i].pages, lib[i].price);
                found = 1;
            }
        }
        if (!found) {
            printf("No books found by this author.\n");
        }
        break;

    case 4:
        printf("Total number of books in the library:
%d\n", count);
        break;

    case 5:
        printf("Exiting program. Thank you!\n");
        return 0;

    default:
        printf("Invalid choice! Please enter a valid
option.\n");
    }
}
return 0;
}

```

OUTPUT :-

C:\Users\amish\Desktop\Unti X

+ v

```
*****##### WELCOME TO E-LIBRARY *****
1. Add book information
2. Display book information
3. List all books of a given author
4. List the count of books in the library
5. Exit
```

❖ Add Book's Information :-

```
*****##### WELCOME TO E-LIBRARY *****
1. Add book information
2. Display book information
3. List all books of a given author
4. List the count of books in the library
5. Exit
```

```
Enter your choice: 1
Enter book name: The Story Of My Experiments With The Truth
Enter author name: Mahatma Gandhi
Enter number of pages: 145
Enter price: 80
Book added successfully!
```

```
*****##### WELCOME TO E-LIBRARY *****
1. Add book information
2. Display book information
3. List all books of a given author
4. List the count of books in the library
5. Exit
```

```
Enter your choice: 1
Enter book name: The Guide
Enter author name: R.K. Narayan
Enter number of pages: 123
Enter price: 90
Book added successfully!
```

*****##### WELCOME TO E-LIBRARY #####*****

1. Add book information
2. Display book information
3. List all books of a given author
4. List the count of books in the library
5. Exit

Enter your choice: 1

Enter book name: A Suitable Boy

Enter author name: Vikram Seth

Enter number of pages: 178

Enter price: 110

Book added successfully!

*****##### WELCOME TO E-LIBRARY #####*****

1. Add book information
2. Display book information
3. List all books of a given author
4. List the count of books in the library
5. Exit

Enter your choice: 1

Enter book name: God of Small Things

Enter author name: Arundhati Roy

Enter number of pages: 220

Enter price: 145

Book added successfully!

*****##### WELCOME TO E-LIBRARY #####*****

1. Add book information
2. Display book information
3. List all books of a given author
4. List the count of books in the library
5. Exit

Enter your choice: 1

Enter book name: Rich Like Us

Enter author name: Nayantara Sehgal

Enter number of pages: 110

Enter price: 250

Book added successfully!

❖ Display Book's Information :-

*****##### WELCOME TO E-LIBRARY #####*****

1. Add book information
2. Display book information
3. List all books of a given author
4. List the count of books in the library
5. Exit

Enter your choice: 2

Book Information:

Book Name: The Story Of My Experiments With The Truth

Author: Mahatma Gandhi

Pages: 145

Price: \$80.00

Book Name: The Guide

Author: R.K. Narayan

Pages: 123

Price: \$90.00

Book Name: A Suitable Boy

Author: Vikram Seth

Pages: 178

Price: \$110.00

Book Name: God of Small Things

Author: Arundhati Roy

Pages: 220

Price: \$145.00

Book Name: Rich Like Us

Author: Nayantara Sehgal

Pages: 110

Price: \$250.00

❖ List all Book's of a given author:-

```
*****##### WELCOME TO E-LIBRARY #####*****
```

1. Add book information
2. Display book information
3. List all books of a given author
4. List the count of books in the library
5. Exit

Enter your choice: 3

Enter author name: Nayantara Sehgal

Books by Nayantara Sehgal:

Book Name: Rich Like Us | Pages: 110 | Price: \$250.00

```
*****##### WELCOME TO E-LIBRARY #####*****
```

1. Add book information
2. Display book information
3. List all books of a given author
4. List the count of books in the library
5. Exit

Enter your choice: 3

Enter author name: Arundhati Roy

Books by Arundhati Roy:

Book Name: God of Small Things | Pages: 220 | Price: \$145.00

❖ List the count of book's in the library.

```
*****##### WELCOME TO E-LIBRARY #####*****
```

1. Add book information
2. Display book information
3. List all books of a given author
4. List the count of books in the library
5. Exit

Enter your choice: 4

Total number of books in the library: 5

❖ Exit Program.

```
*****##### WELCOME TO E-LIBRARY #####*****
```

1. Add book information
2. Display book information
3. List all books of a given author
4. List the count of books in the library
5. Exit

```
Enter your choice: 5
```

```
Exiting program. Thank you!
```

```
-----  
Process exited after 210 seconds with return value 0  
Press any key to continue . . . |
```

CONCLUSION :-

The **Library Management System** provides a modernized, efficient, and automated way of handling library operations. The integration of **HTML, CSS, JavaScript** for the user interface, **C Language** for backend processing, and **Oracle Database** for data management ensures an effective and seamless user experience.

This system not only enhances efficiency by reducing manual efforts but also improves accuracy in book management, user transactions, and record-keeping. The automation of issuing, returning, and searching for books minimizes human errors and ensures books are available to students and researchers without unnecessary delays.

Moreover, the **multi-user access** and **secure authentication** features provide a robust system that can handle multiple transactions simultaneously while ensuring data integrity. The inclusion of **fine calculation, book reservation, and report generation** further adds to the system's usability, making it a comprehensive solution for library management.

From an administrative perspective, the system aids in book inventory management, ensuring an updated and well-maintained catalog. The ability to generate analytical reports on overdue books, issued books, and defaulters provides valuable insights that help in strategic decision-making.

Overall, the **Library Management System** serves as a **cost-effective, user-friendly, and reliable** solution that enhances library operations, improves the user experience, and ensures systematic book management in an educational or institutional setting.

REFERENCES :-

Websites :-

GeeksForGeeks:<https://www.geeksforgeeks.org/e-library-management-system/>

Code_With_C:<https://www.codewithc.com/project-in-c-programming-building-a-library-management-system-2023/?amp=1>

Github:https://github.com/aaneloy/Library_Management_System_C/blob/master/library%20management%20system.c

ChatGPT:<https://chatgpt.com/c/67b8880b-64bc-8006-b117-e89b83a55394>

Book & Docs. :-

The C Programming book (3rd Edition) By Ashok n. kamthane Amit Ashok kamthane.