A Project

Report On

Library Management System

CAM206-2C Programming using C++

Department of Computer Science,

School of Science and Technology,

BCA Programme

VANITA VISHRAM WOMEN’S UNIVERSITY,

SURAT

2024-2025

SUPERVISOR:-Dr Chirag Mehta.

Sr. No.	Title	Page No.
1.	Introduction	
2.	Aim and Objective	
3.	Existing System	
4.	Proposed System	
5.	Project Details	
6.	Features of Project	
7.	Implementation	
8.	Interface of Output Screen	
9.	Conclusion	

INTRODUCTION

This project is a **Library Management System** developed using **C++** to manage book lending and member records efficiently. The system allows administrators to handle books, keep track of members, and monitor borrowing and returning of books, thereby simplifying library operations.

Key Features:

1. **Login System**: Secure login feature for accessing the system using a pre-set username and password.
2. **New Member Registration**: Allows the administrator to add new member records with details like member ID, name, book ID, book name, issue date, and due date.
3. **Book Management**:
   - **View Book Details**: The system displays detailed information about a book, including the title, author, year, and available copies.
   - **View All Books**: A complete list of all available books in the library with their respective details.
4. **Book Borrowing**: Members can borrow a book if it is available, and the system updates the number of available copies accordingly.
5. **Book Returning**: Members can return a borrowed book, and the system will update the availability status of the book.
6. **Member Records**: Administrators can view all member records, including the member ID, name, book details, and issue/return dates.
7. **Logout**: Option to securely log out of the system after managing operations.

Technologies Used:

- **C++**: Core language used to build the system.
- **Standard I/O and String Handling**: For user input/output and string manipulations.
- **Data Structures**: Arrays are used to store books and member records.
- **Screen Clearing Functionality**: Platform-specific screen clearing using system commands (cls for Windows, clear for Unix-based systems).

PROJECT OBJECTIVE

The **Library Management System** project aims to simplify and automate the management of library operations. The primary objective is to create a user-friendly system that efficiently handles book-related activities and member records, reducing manual tasks and minimizing errors. The system helps to:

1. **Enhance Library Operations**: Streamline processes such as book borrowing, returning, and record-keeping, allowing for smooth functioning of the library.
2. **Improve Record Accuracy**: Maintain accurate and up-to-date records of available books and registered members, ensuring data integrity.
3. **Simplify Member and Book Management**: Provide an easy-to-use interface for administrators to register new members, view book details, and track the status of borrowed books.
4. **Reduce Manual Work**: Minimize the need for manual record maintenance, reducing human error and saving time in daily operations.
5. **Efficient Resource Utilization**: Ensure the efficient management of library resources by keeping track of book availability and ensuring timely returns.
6. **Secure Access**: Implement a login system to restrict unauthorized access, ensuring only authorized personnel can manage library operations.

This project is designed to cater to small to medium-sized libraries, offering a solution that is both practical and scalable. It provides a foundation for future enhancements, such as adding more books, members, or advanced functionalities like fine calculation and report generation.

Existing System

In traditional libraries, management processes are often carried out manually, involving a lot of paperwork and manual record-keeping. The existing system in most small libraries includes the following challenges:

1. **Manual Record-Keeping**: Librarians rely on physical registers or spreadsheets to manage book and member records. This can lead to errors, data redundancy, and difficulty in maintaining accurate records.
2. **Time-Consuming Operations**: Borrowing, returning, and registering books and members require significant manual intervention, slowing down the overall process and increasing the workload on library staff.
3. **Inefficient Book Tracking**: Manually tracking book availability can lead to misplaced records, making it difficult to determine how many copies are available or whether a book is currently borrowed.
4. **Limited Accessibility**: The system often lacks the ability for remote access, limiting the operations to specific times when the librarian is present.
5. **Inadequate Security**: Since manual processes do not typically incorporate security measures, unauthorized users may access or tamper with records, leading to data inconsistencies.
6. **Limited Reporting**: There is a lack of efficient reporting mechanisms, making it difficult to generate insights such as most borrowed books, overdue books, or member activity.

Proposed System

The proposed Library Management System is designed to overcome the limitations of the existing manual system by automating key processes such as book borrowing, returning, and member management. This system will streamline operations, improve accuracy, and ensure secure data handling.

Key features of the proposed system include:

1. **Automated Member and Book Management**:
   - The system allows for the efficient creation of new member records, storing details such as member ID, name, book details, and due dates.
   - Book details including title, author, publication year, and available copies are maintained, ensuring up-to-date and accurate records.
2. **Book Borrowing and Returning**:
   - Members can borrow books by providing the book ID, and the system automatically updates the available copies.
   - A book borrowed by one member is flagged as unavailable until it is returned.
   - The system allows members to return books, updating the availability in real-time.
3. **Real-Time Book Availability**:
   - The system tracks and displays the number of available copies for each book, ensuring transparency regarding book availability.
4. **Member Record Management**:
   - All member records are stored in the system, allowing library staff to easily view, edit, and manage member details.
   - The system supports the viewing of all current members, their borrowed books, issue dates, and due dates.
5. **User Authentication**:
   - Secure login credentials are required to access the system, ensuring that only authorized users can perform operations such as borrowing or returning books.
6. **Enhanced User Interface**:
   - The system features a simple and intuitive user interface, allowing library staff to quickly navigate through the different functions such as adding members, viewing books, borrowing, and returning.

  - Clear prompts and error messages help guide users through the process, ensuring smooth operation.

7. **Efficient Data Storage**:
   - All book and member data are stored electronically, reducing the risk of data loss or errors associated with manual record-keeping.
   - The system prevents data redundancy and ensures consistency across records.

8. **Improved Security and Data Integrity**:
   - The system ensures that records are updated in real-time and that only authorized users can access sensitive operations such as modifying book or member details.
   - It also prevents accidental data loss by incorporating checks before deletion or modification of records.

9. **Streamlined Reporting**:
   - The system can generate basic reports, such as the list of all available books or the list of all members along with their borrowed books, facilitating decision-making for library staff.

Benefits of the Proposed System:

- **Efficiency**: Reduces the time required to perform routine library operations, such as book borrowing, returning, and member registration.
- **Accuracy**: Eliminates the errors that often occur with manual record-keeping.
- **Transparency**: Real-time updates on book availability and member records ensure transparency.
- **Security**: User authentication and data integrity checks ensure that sensitive data is protected.
- **Scalability**: The system can easily be extended to accommodate more books and members as needed.
- This proposed system will significantly enhance the overall functionality and efficiency of library operations, improving both user experience and management processes.

Project Details

In the table below, you'll find a list of the key project data and technologies used in this system. The **Library Management System** is a C++ based project designed for efficient library operations.

ABOUT PROJECT	**PROJECT DETAILS**
Project Name:	Library Management System
Project Platform:	C++
Programming Language Used:	C++ Programming Language
Developer Name:	[Your Name]
Software Requirements:	
Operating System:	Windows
Front End:	C++ (Command Line Interface)
Back End:	N/A
Project Type:	Desktop Application
Team Size:	6
Guide Name:	[Guide Name]

This project is designed to manage the operations of a library, including book management, member registration, book borrowing, and return functionalities. It uses basic C++ structures and classes, along with simple data handling, to simulate a library system.

Features of the Project:

The **Library Management System** includes several key features that ensure efficient management of library resources and member activities. These features include:

1. New Record
2. Book Details
3. View All Books
4. Borrow a Book
5. Return a Book
6. Display Member Records
7. Logout

Implementation:

```
#include<stdio.h>
#include<string.h>
#include<iostream>
using namespace std;

char usrname[10];
char passwd[10];
int choice;

struct Book
{
        char title[34];
        char author[34];
        int year;
        int availablecopies;
        bool borrowed; // Track if the book is borrowed
};

Book books[5] = {
        {"The Great Gatsby","F.Scott Fitzgerald",1925,3, false},
        {"To Kill a Mockingbird","Harper Lee",1960,2, false},
        {"1984","George Orwell",1949,5, false},
        {"Pride and Prejudice","Jane Austen",1813,4, false},
        {"The Catcher in the Rye","J.D. Salinger",1951,1, false}
};
```

```
struct Member
{
        int member_id;
        char member_name[50];
        int book_id;
        char book_name[30];
        char issue_date[30];
        char return_date[30];
        char due_date[30];
};

Member members[5];  // Array to store member records
int memberCount = 0; // Count of members

class member
{
public:
        void hdisp()
        {
                cout <<  "\n\n\n\n\n\n\n\n\t\t\t
*********************************************";
                cout << "\n\t\t\t LIBRARY  MANAGEMENT SYSTEM ";
                cout << "\n\t\t\t *********************************************";
        }

        void login()
```

```
{
        int flag1=0, flag2=0;
        cout<<"\n \t\t\t User Name =";
        cin>>(usrname);
        cout <<"\n\t\t\t Password =";
        cin>>(passwd);
        flag1=strcmp(usrname,"vvwu");
        flag2=strcmp(passwd,"vvwu");
        if(flag1==0 && flag2==0)
                menu();
        else
        {
                cout <<"\n\n\n \t\t\t Re-enter your details ....";
                fdisp();
        }
}

void fdisp()
{
        cout<< "\n\t\t\t ****************************************";
        cout <<"\n\t\t\t      All Rights Reserved ";
}

void menu()
{
        hdisp();
```

```
do
{
cout<<"\n\t\t\t1. \t New Member Record";
cout<<"\n\t\t\t2. \t Book Details";
cout<<"\n\t\t\t3. \t View All Books";
cout<<"\n\t\t\t4. \t Borrow a Book";
cout<<"\n\t\t\t5. \t Return a Book";
cout<<"\n\t\t\t6. \t Display Member Records";
cout<<"\n\t\t\t7. \t Logout";
cout<<"\n\n\n\t\t\t Enter Your Choice =";
cin>>choice;

switch(choice)
{
        case 1:
                New_member();
                break;
        case 2:
                Book_details();
                break;
        case 3:
                View_books();
                break;
        case 4:
                Borrow_book();
                break;
```

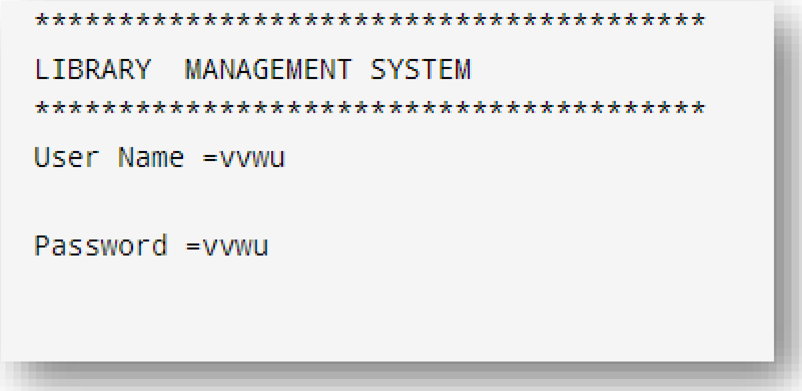
```
            case 5:
                Return_book();
                break;
            case 6:
                Display_members();
                break;
            case 7:
                Logout();
                break;
            default:
                cout <<"Sorry, wrong choice!";
                break;
    }
  }while(choice != 7);
    }

    void New_member()
    {
        if(memberCount >= 5)
        {
            cout << "Cannot add more members. Maximum limit reached.\n";
            return;
        }

        cout << "Enter Member ID: ";
        cin >> members[memberCount].member_id;
    cout << "Enter Member Name: ";
    cin >> members[memberCount].member_name;
    cout << "Enter Book ID: ";
    cin >> members[memberCount].book_id;
    cout << "Enter Book Name: ";
    cin >> members[memberCount].book_name;
    cout << "Enter Issue Date: ";
    cin >> members[memberCount].issue_date;
    cout << "Enter Due Date: ";
    cin >> members[memberCount].due_date;

    cout << "Member record added successfully.\n";
    memberCount++;
}

void Display_members()
{
    if(memberCount == 0)
    {
        cout << "No member records available.\n";
        return;
    }

    cout << "\n\t\t\t--- Member Records ---\n";
    for(int i = 0; i < memberCount; i++)
    {
```

```
                cout << "Member ID: " << members[i].member_id << endl;
                cout << "Member Name: " << members[i].member_name << endl;
                cout << "Book ID: " << members[i].book_id << endl;
                cout << "Book Name: " << members[i].book_name << endl;
                cout << "Issue Date: " << members[i].issue_date << endl;
                cout << "Due Date: " << members[i].due_date << endl;
                cout << "----------------------------------------\n";
            }
    }

    void Book_details()
    {
        int id;
        cout << "Enter the Book ID to view details: ";
        cin >> id;
        if (id >= 0 && id < 5)
        {
            cout << "TITLE: " << books[id].title << endl;
            cout << "AUTHOR: " << books[id].author << endl;
            cout << "YEAR: " << books[id].year << endl;
            cout << "AVAILABLE COPIES: " << books[id].availablecopies << endl;
            cout << "----------------------------------------------------" << endl;
        }
        else
        {
            cout << "Invalid Book ID!\n";
        }
    }

    void View_books()
    {
        cout << "\nAvailable Books in the Library:\n";
        for(int i = 0; i < 5; i++)
        {
            cout << i << ". " << books[i].title << " by " << books[i].author
              << " (" << books[i].year << ") - Copies: " << books[i].availablecopies << endl;
        }
    }

    void Borrow_book()
    {
        int id;
        cout << "Enter the Book ID you want to borrow: ";
        cin >> id;
        if (id >= 0 && id < 5)
        {
            if (books[id].availablecopies > 0 && !books[id].borrowed)
            {
                books[id].availablecopies--;
                books[id].borrowed = true; // Mark the book as
borrowed
                cout << "Book borrowed successfully!\n";
            }
            else
            {
                cout << "Sorry, no copies available or the book is
already borrowed!\n";
            }
        }
        else
        {
            cout << "Invalid Book ID!\n";
        }
    }

    void Return_book()
    {
        int id;
        cout << "Enter the Book ID you want to return: ";
        cin >> id;
        if (id >= 0 && id < 5)
        {
            if (books[id].borrowed)  // Check if the book was
borrowed
            {
                books[id].availablecopies++;
                books[id].borrowed = false; // Mark the book as returned
                cout << "Book returned successfully!\n";
            }
            else
            {
                cout << "This book wasn't borrowed!\n";
            }
        }
        else
        {
            cout << "Invalid Book ID!\n";
        }
    }

    void Logout()
    {
        hdisp();
        cout << "\n\t\t\t    Thank you for using the system...";
        exit(0);
    }
};

int main()
{
    member m1;
```

```
        m1.hdisp();

        m1.login();

        return 0;

}
```

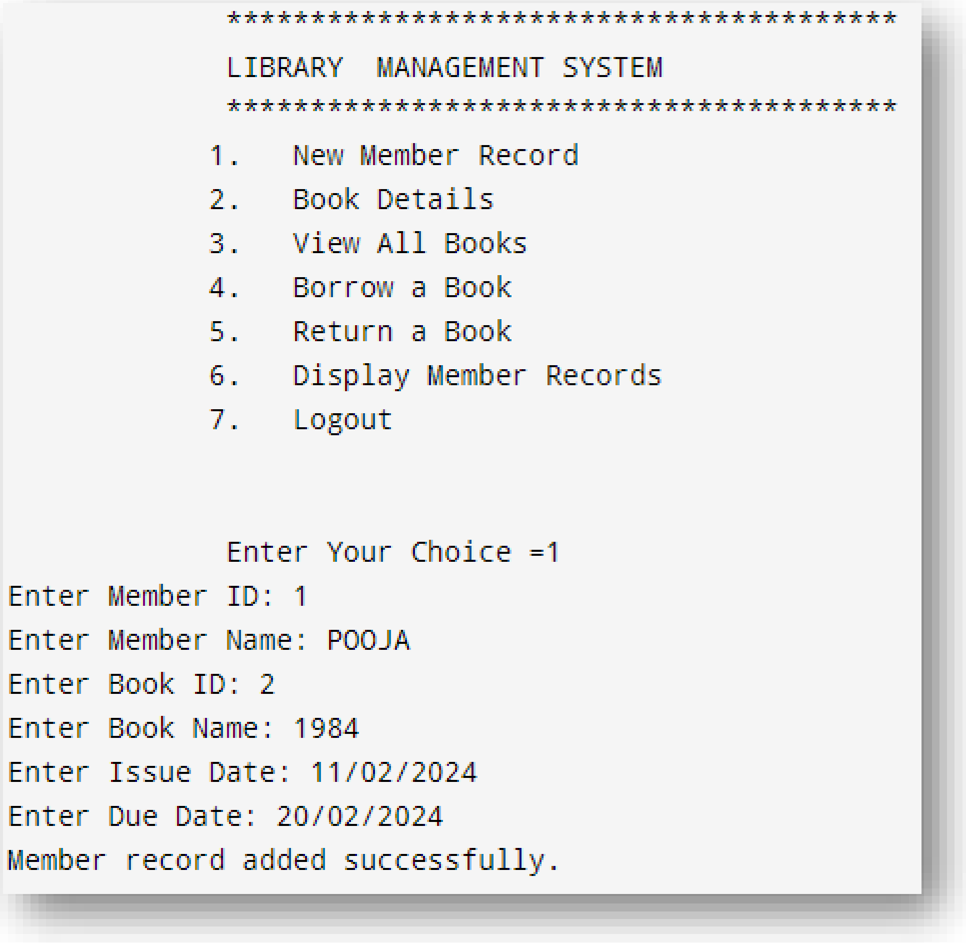
Interface of Output Screen:

1) Login



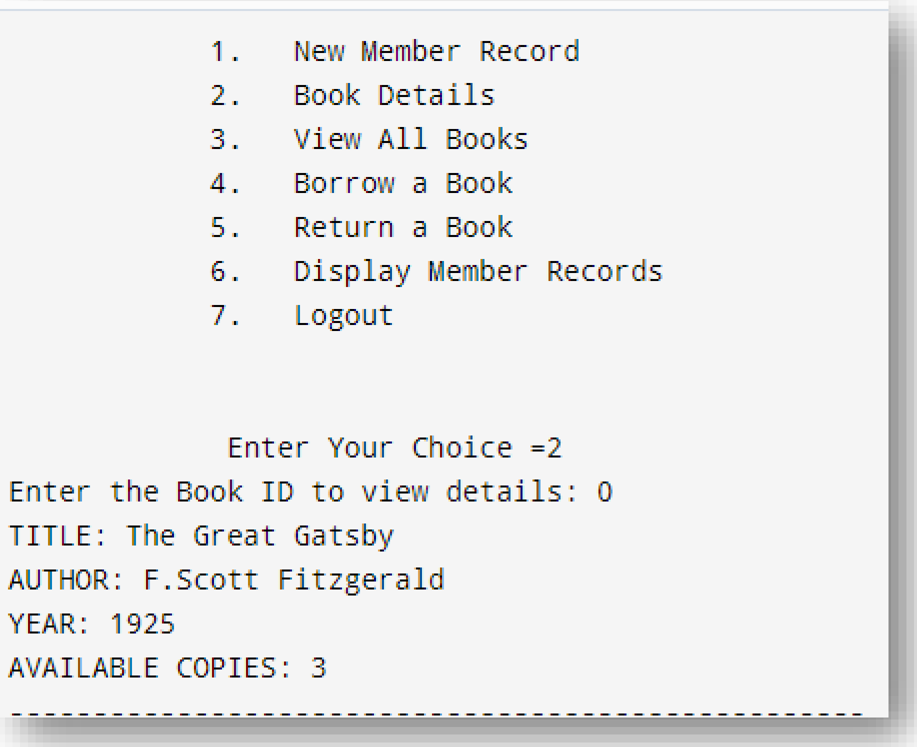
```
****************************************
LIBRARY  MANAGEMENT SYSTEM
****************************************
User Name =vvwu

Password =vvwu
```

2) New Record



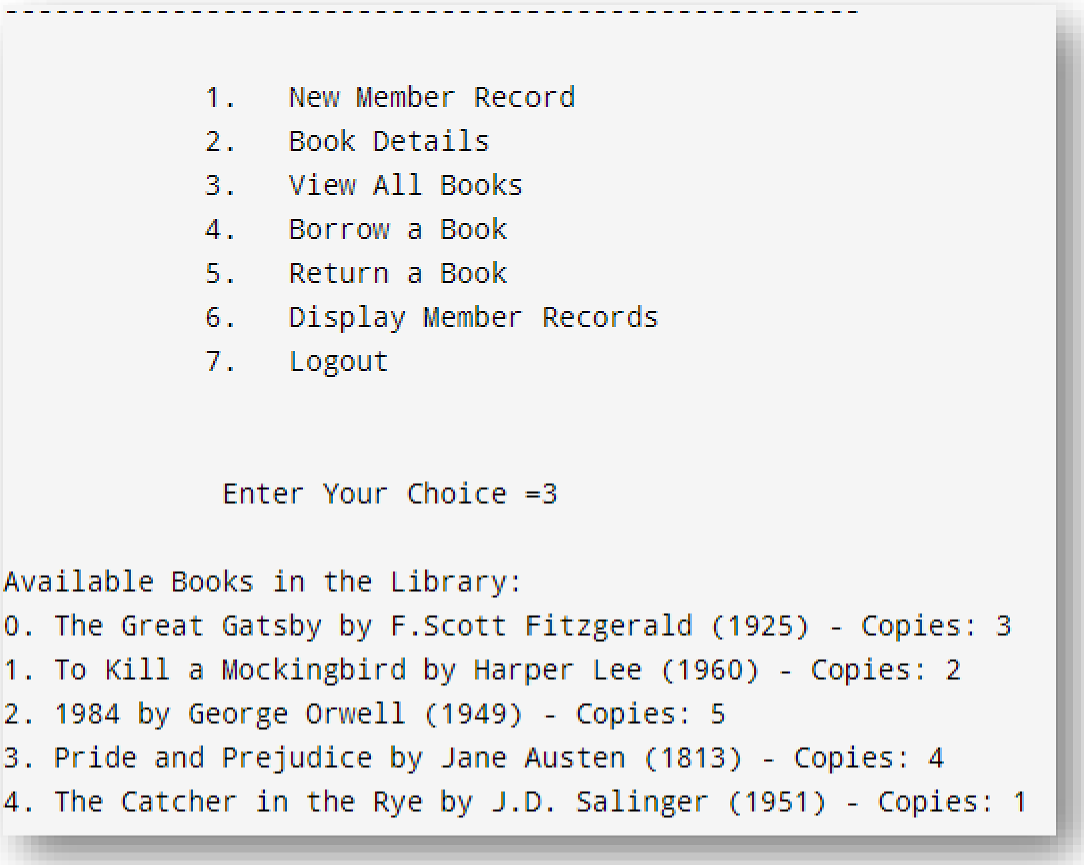
```
             ****************************************
             LIBRARY  MANAGEMENT SYSTEM
             ****************************************
            1.   New Member Record
            2.   Book Details
            3.   View All Books
            4.   Borrow a Book
            5.   Return a Book
            6.   Display Member Records
            7.   Logout


             Enter Your Choice =1
Enter Member ID: 1
Enter Member Name: POOJA
Enter Book ID: 2
Enter Book Name: 1984
Enter Issue Date: 11/02/2024
Enter Due Date: 20/02/2024
Member record added successfully.
```

3)Book Details



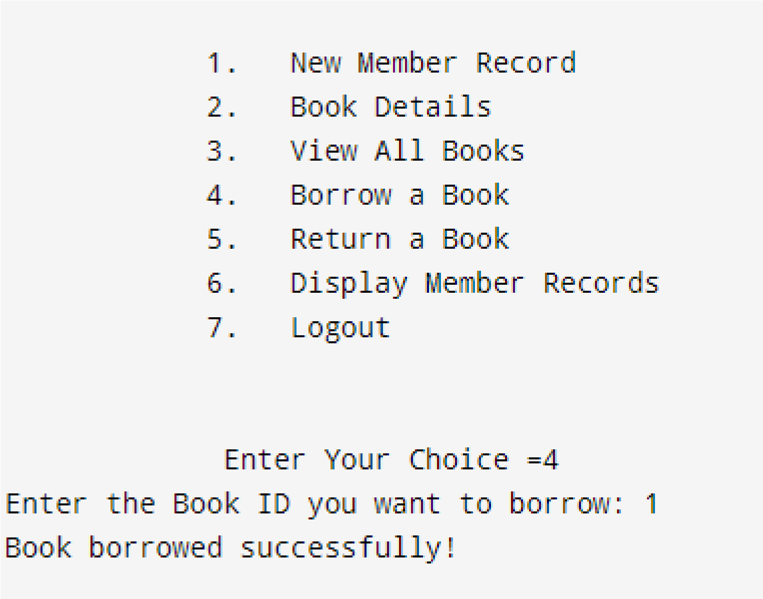
```
                1.    New Member Record
                2.    Book Details
                3.    View All Books
                4.    Borrow a Book
                5.    Return a Book
                6.    Display Member Records
                7.    Logout


                 Enter Your Choice =2
Enter the Book ID to view details: 0
TITLE: The Great Gatsby
AUTHOR: F.Scott Fitzgerald
YEAR: 1925
AVAILABLE COPIES: 3
---------------------------------------------------
```

4) View All Books



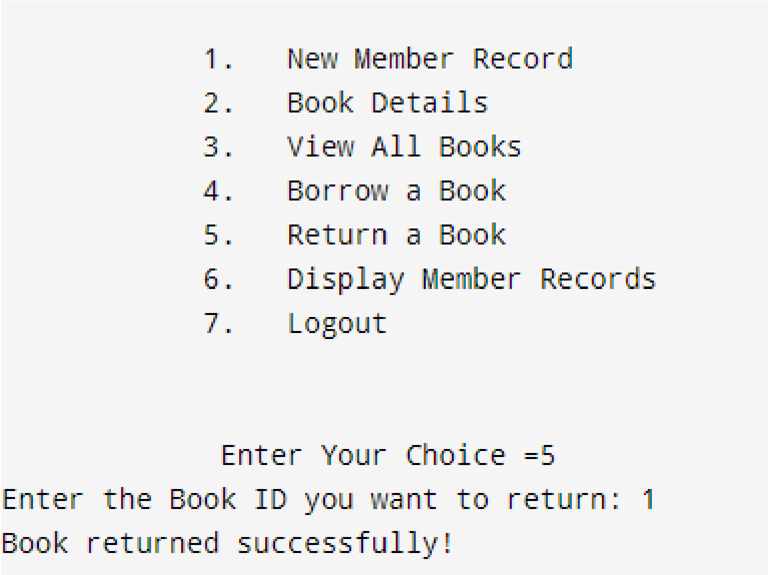
```
---------------------------------------------------


            1.    New Member Record
            2.    Book Details
            3.    View All Books
            4.    Borrow a Book
            5.    Return a Book
            6.    Display Member Records
            7.    Logout



             Enter Your Choice =3

Available Books in the Library:
0. The Great Gatsby by F.Scott Fitzgerald (1925) - Copies: 3
1. To Kill a Mockingbird by Harper Lee (1960) - Copies: 2
2. 1984 by George Orwell (1949) - Copies: 5
3. Pride and Prejudice by Jane Austen (1813) - Copies: 4
4. The Catcher in the Rye by J.D. Salinger (1951) - Copies: 1
```

5) Borrow a Book



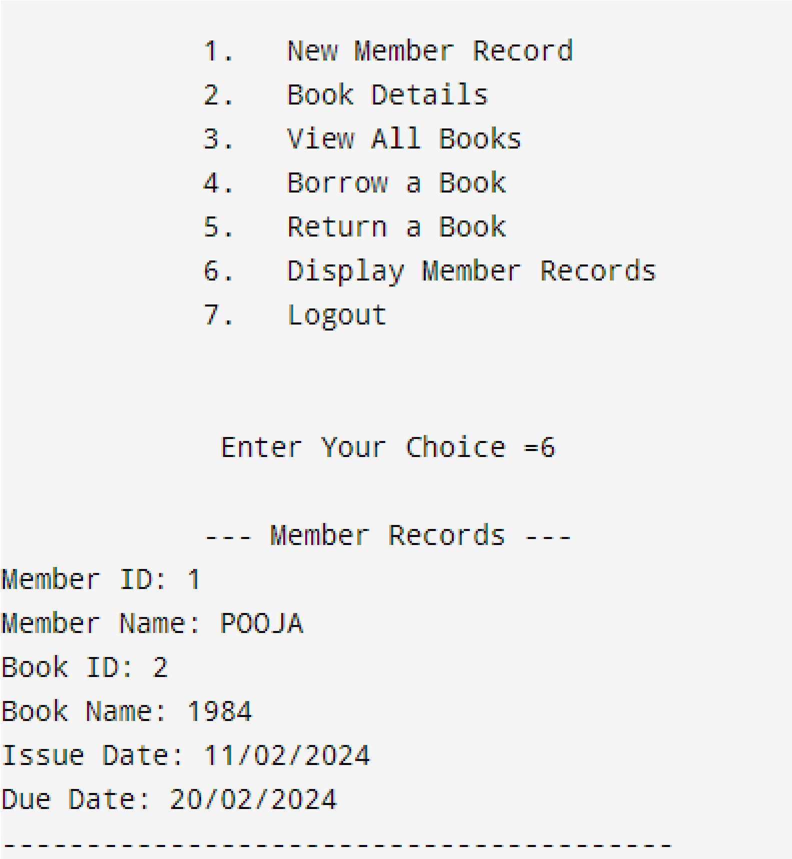
```
            1.   New Member Record
            2.   Book Details
            3.   View All Books
            4.   Borrow a Book
            5.   Return a Book
            6.   Display Member Records
            7.   Logout


             Enter Your Choice =4
Enter the Book ID you want to borrow: 1
Book borrowed successfully!
```

6) Return a Book



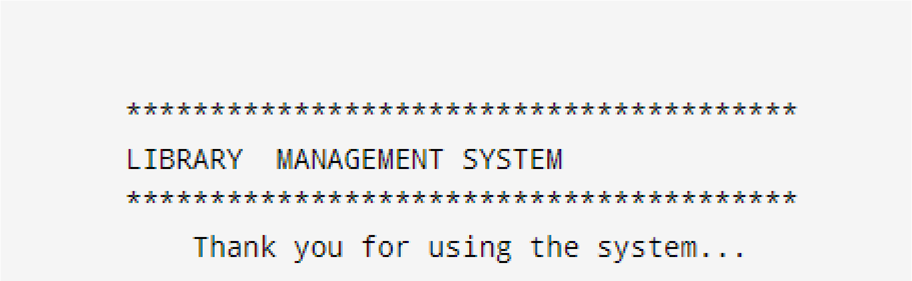
```
            1.   New Member Record
            2.   Book Details
            3.   View All Books
            4.   Borrow a Book
            5.   Return a Book
            6.   Display Member Records
            7.   Logout


             Enter Your Choice =5
Enter the Book ID you want to return: 1
Book returned successfully!
```

7) Display Member Records



```
            1.    New Member Record
            2.    Book Details
            3.    View All Books
            4.    Borrow a Book
            5.    Return a Book
            6.    Display Member Records
            7.    Logout


             Enter Your Choice =6

            --- Member Records ---
Member ID: 1
Member Name: POOJA
Book ID: 2
Book Name: 1984
Issue Date: 11/02/2024
Due Date: 20/02/2024
----------------------------------------
```

8) Logout



```
****************************************
LIBRARY  MANAGEMENT SYSTEM
****************************************
    Thank you for using the system...
```

Conclusion:

The **Library Management System** developed using C++ provides a basic but functional platform for managing library operations such as member registration, book details management, book borrowing, and returning. This system is designed to streamline the essential functions of a library, ensuring that users can efficiently track available books, record member activities, and automate common tasks.

The system is user-friendly with a simple interface, ensuring ease of access for users with minimal technical knowledge. Through the structured implementation of classes, structs, and functions, the system encapsulates key functionalities while maintaining clean and readable code.

By implementing additional features like advanced search options, fine calculations, and integrating more complex databases, this basic library system can be expanded to meet the needs of larger libraries or educational institutions. As it stands, this project demonstrates fundamental programming concepts in C++ while addressing real-world problems in library management.