



UNIVERSITY INSTITUTE *of*  
**COMPUTING**  
*Asia's Fastest Growing University*

**NAAC**  
**GRADE A+**  
ACCREDITED UNIVERSITY

# Mini Project

## LIBRARY-MANAGEMENT-SYSTEM

**Student Name:** Kartikey Gupta

**UID:** 22BCA10176

**Branch:** UIC BCA

**Section/Group:** 3-B

**Semester:** 5th

**Date of Performance:** 08/11/24

**Subject Name:** Computing Aptitude

**Subject Code:** 22CAH-301

## Aim/Overview of the Practical

Create a Library Management System using C++ to manage book records in a library. This console-based application allows users to add, view, and search for books.

## Objective:

The objective of this project is to develop a basic library management tool that:

- Helps store information about books.
- Facilitates the addition, display, and extraction of book details.
- Demonstrates file handling in C++.

## Technologies Covered:

- **Programming Language:** C++
- **File Handling:** Using <fstream> for storing and retrieving book data.
- **Console Interface:** A simple text-based UI for user interactions.

## Task to be Done:

- **Add Book (Admin):**
  - Take inputs for the book's ID, name, and author's name.
  - Store the book data in a file.
- **Show All Books:**
  - Retrieve and display all books currently stored in the system.
- **Extract a Book:**
  - Search for a book by ID and display its details if found.

## Algorithm/Flowchart:

### Main Algorithm:

#### Display Menu:

1. Options include viewing all books, extracting a book, adding a book, and exiting.

#### Choice Handling:

1. Based on the user's choice, call the corresponding function.

#### Functions:

1. **Add Book:** Collects book details from the user and writes them to a file.
2. **Show All Books:** Reads from the file and displays book details.
3. **Extract Book:** Searches the file for a specific book ID and displays the matching book information.

## Dataset:

The dataset comprises book records stored in a text file (bookData.txt), each line containing:

- **Book ID**
- **Book Name**
- **Author's Name**

-----  
1-Show All Books  
2-Extract Book  
3-Add books(ADMIN)  
4-Exit  
-----

Enter Your Choice ::

1-Show All Books  
2-Extract Book  
3-Add books(ADMIN)  
4-Exit  
-----

Enter Your Choice :: 1

Book Id	Book Name	Author's Name
1	who kill naina	ak arun
1	Dairy of a young girl	Anne frank
2	The Lords of the Rings	J.R.R. Tolkien

PS C:\Users\kkart\Downloads\c++ project\Library management system> |

## Code for Experiment/Practical:

```
Library management system > C++ library.cpp
1  #include<iostream>
2  #include<fstream>
3  using namespace std;
4
5  class temp{
6
7      string id,name,author,search;
8      fstream file;
9  public:
10     void addBook();
11     void showAll();
12     void extractBook();
13
14 }obj;
15
16 int main(){
17
18     char choice;
19     cout<<"-----"<<endl;
20     cout<<"1-Show All Books"<<endl;
21     cout<<"2-Extract Book"<<endl;
22     cout<<"3-Add books(ADMIN)"<<endl;
23     cout<<"4-Exit"<<endl;
24     cout<<"-----"<<endl;
25     cout<<"Enter Your Choice :: ";
26     cin>>choice;
27
28     switch(choice){
29         case '1':
30             cin.ignore();
31             obj.showAll();
32             break;
33         case '2':
34             cin.ignore();
35             obj.extractBook();
36             break;
37         case '3':
```

```
37     case '3':
38         cin.ignore();
39         obj.addBook();
40     break;
41     case '4':
42         return 0;
43     break;
44     default:
45         cout<<"Invalid Choice...!";
46 }
47
48
49 return 0;
50 }
51 void temp :: addBook(){
52     cout<<"\nEnter Book ID :: ";
53     getline(cin,id);
54     cout<<"Enter Book Name :: ";
55     getline(cin,name);
56     cout<<"Enter Book's Author name :: ";
57     getline(cin,author);
58
59     file.open("bookData.txt",ios :: out | ios :: app);
60     file<<id<<"*"<<name<<"*"<<author<<endl;
61     file.close();
62 }
63 void temp :: showAll(){
64     file.open("bookData.txt",ios :: in);
65     getline(file,id,'*');
66     getline(file,name,'*');
67     getline(file,author,'\n');
68     cout<<"\n\n";
69     cout<<"\t\t Book Id \t\t\t Book Name \t\t\t Author's Name"<<endl;
70     while(!file.eof()){
```

```
69     cout<<"\t\t Book Id \t\t\t Book Name \t\t\t Author's Name"<<endl;
70     while(!file.eof()){
71         cout<<"\t\t "<<id<<" \t\t\t\t "<<name<<" \t\t\t "<<author<<endl;
72
73         getline(file,id,'*');
74         getline(file,name,'*');
75         getline(file,author,'\n');
76     }
77     file.close();
78 }
79 void temp :: extractBook(){
80
81     showAll();
82     cout<<"Enter Book Id :: ";
83     getline(cin,search);
84
85     file.open("bookData.txt",ios :: in);
86     getline(file,id,'*');
87     getline(file,name,'*');
88     getline(file,author,'\n');
89
90     cout<<"\n\n";
91     cout<<"\t\t Book Id \t\t\t Book Name \t\t\t Author's Name"<<endl;
92     while(!file.eof()){
93         if(search == id){
94             cout<<"\t\t "<<id<<" \t\t\t\t "<<name<<" \t\t\t "<<author<<endl;
95             cout<<"Book Extracted Successfully...!";
96         }
97         getline(file,id,'*');
98         getline(file,name,'*');
99         getline(file,author,'\n');
100     }
101     file.close();
102 }
```

## Result/Output/Writing Summary:

- **Expected Result:** The program allows users to add new books, view all stored books, and search for specific books by ID. Book data persists in bookData.txt.
- **User Interaction:** Users navigate a text menu to perform actions like viewing, adding, and searching for books.



## Learning Outcomes (What I Have Learned):

- **File Handling in C++:** Learned how to store, retrieve, and search data from files.
- **Console UI Design:** Developed a simple, interactive text-based interface.
- **Class and Functions:** Practiced using classes and functions to structure code efficiently.

## Evaluation Grid:

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Demonstration and Performance (Pre Lab Quiz)		5
2.	Worksheet		10
3.	Post Lab Quiz		5



