**SDM COLLEGE OF ENGINEERING AND**

**TECHNOLOGY, Dharwad-580002**

**(An autonomous institution affiliated to**

**Visvesvaraya Technological University, Belgaum – 590018)**



**Department of Information Science and Engineering**

**4TH SEMESTER B.E ACADEMIC YEAR: 2022-23**

**INTRODUCTORY PROJECT**

**21UISL406**

**UNDER THE GUIDANCE OF**

Dr Sachidanand S. Joshi

Submitted by

Ananya R Upadhya 2SD21IS006

Priya Mannur 2SD21IS036

Vaishnavi Hegde 2SD21IS057

Trupti Hasabi 2SD21IS052

Aditi Kulkarni 2SD21IS003

Nisargaa Kulkarni 2SD21IS028

**S. D. M. College of Engineering & Technology Dharwad– 580002**

 **Department of Information Science and Engineering**

**CERTIFICATE**

This is to certify that the project work [IP] entitled “Library Management System” is a bonafied work carried out Ananya R Upadhya - 2SD21IS006, Priya Mannur - 2SD21IS036, Vaishnavi Hegde - 2SD21IS057, Trupti Hasabi - 2SD21IS052, Aditi Kulkarni – 2SD21IS003, Nisargaa Kulkarni - 2SD21IS028 completed the Introductory Project for IV Semester B.E. Degree in Information Science and Engineering of S.D.M. College of Engineering, Autonomous Institution under Visvesvaraya Technological University, Belagavi during the year 2021–2022. It is certified that all necessary suggestions indicated for internal assessment have been incorporated. The project work has been approved as it has successfully satisfied the academic requirements as prescribed for the Bachelor of Engineering Degree.

**Guide: Dr Sachidanand S. Joshi**

**Project Coordinator: Dr Vandana S Bhat**

**HOD-ISE: Dr Jagadeesh D. Pujari**

**Principal: Dr K. Gopinath**

**PHASE 1**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TITLE** | **PAGENO** |
| 1 | Abstract | 4 |
| 2 | Introduction | 5 |
| 3 | Literature Survey | 6-8 |
| 4 | Survey Of Different Methodologies | 9-11 |
| 5 | Conclusion | 12 |
| 6 | References | 13 |

1. **ABSTRACT**

Library management systems (LMSs) are software applications that help libraries manage their collections, circulation, and other resources. LMSs can track the number of books in a library, the number of books that have been issued and the number of books that have been returned or renewed. They can also help libraries to quickly identify books, issue and reissue books, and manage their data in an efficient and organized manner. LMSs are essential tools for modern libraries, and they can help libraries to operate more efficiently and effectively, and they can provide patrons with a better experience.

1. **INTRODUCTION**

Libraries serve as vital gateways to knowledge dissemination and the cultivation of a conducive learning milieu. Effective administration of their resources and operations is paramount for seamless functionality. Considering this, this project introduces an innovative Library Management System that capitalizes on data structures to adeptly store and swiftly retrieve book-related data. By incorporating linked lists into its framework, the system strives to revolutionize book organization and amplify the efficiency of borrowing and returning processes. In an era where unfettered access to information is pivotal, libraries assume an irreplaceable role. Nonetheless, conventional methods of resource management and operational oversight have encountered constraints that hinder their fluid operation. This project responds with a visionary solution that holds the potential to reshape the landscape of library management.

At its heart, the Library Management System revolves around the infusion of advanced data structures, particularly linked lists. This integration is more than just technological; it signifies a strategic move towards optimizing the arrangement of books. Linked lists facilitate the construction of dynamic book lists, ensuring resourceful storage and expedited retrieval. By doing so, they replace the conventional linear search with a more efficient method, thereby drastically cutting down the time it takes to locate and manage books. The ramifications of this undertaking are extensive. Libraries are poised to experience heightened operational efficiency as the arduous administrative procedures are streamlined. The accessibility of resources stands to witness a notable elevation, ultimately enriching the library experience for patrons. The project's success hinges on a meticulously designed user interface and rigorous testing to authenticate its functionality, reliability, and security.

In summation, the Library Management System project addresses a pivotal need within the domain of library management. Through the judicious integration of linked lists, it reimagines the modus operandi of book organization and management, ushering libraries into an era of heightened effectiveness. This project not only upholds the pivotal role of libraries but also enhances their capacity to facilitate knowledge dissemination and foster learning.

1. **LITERATURE SURVEY**

[1] Describes the automation of library, it provides facilities to student or member to search for the required books and it allows the administrator or librarian to Issue & return books to students can create & delete membership of students. The implementation is described as follows, it initially has a login page so that user can login into online library by filling correct username and password. This login renders a welcome page with various options such as view profile, edit profile, change password, Book Availability. The first option 'view profile’, results in a page with the content of information of the user(profile). User can also change her password through the third option. In Book availability service, user can search for a particular book and issue it on her name. This way it eases the manual errors usually caused. In the admin part of the code, he has options like, issuing a book, add a book, remove a book. the combination of all the above web pages result in a web application named Library Management System, which works as online library.

[2] Describes about ELMS, Enhanced library management system, an app. ELMS is an application which refers to library systems which are generally small/medium in size. It helps the librarian to manage the library with the computerized system, by use of this system she can record various transaction like issue of books, return of books, adding of new student and adding of new student and more. The users are administrator, Librarian and the Student who has entered into the library system by use of their own login page and access the library information, in this system. When the user has entered in the application and request for information it will fetch from the library database and provide to the user which they required. There are three major modules, Admin module, Librarian module, Student module, each with their different logins resulting in different pages. The admin has to control and manage the library management system; hence they have different set of services that are required. A librarian has a page that can maintain book transaction by the process of adding the book, view the issued book, Return book details. A student after registration can view the details of available book, return date and due amount from library. This paper has presented the details of this library application which is named as Enhanced Library management system. It is the windows application development; we use the Eclipse Neon IDE tool and the MySQL Database to develop this project.

[3] Describes about developing a Library Management System, certain modules

of the system should be critically taken into consideration as these modules forms the fundamental part of the system. These modules include the admin login, Student login, Staff login, Search book and public access. Frontend of this design was developed using html, php, and css, while the backend was developed with MySQL. Summarily, HTML was used to write the whole codes for the development of the web pages, PHP for server-side scripting and CSS for the web pages styling. External styling was used. Also, MySQL was employed as the database technology. The software engineering model employed for this design was waterfall model. Waterfall model is often referred to as linear sequential life cycle model. The model develops systematically from one phase to another in a downward fashion, just like water fall as the name implies. Output of a phase will be input of the next phase. It consists of different pages such as Home Page, Registration Page, Login Page, Add Books Page, Search Books Page. The Backend part consists of the following attributes Borrow Details Table, User Table, Book Table. The designed Robust Library Management System was developed to automate library activities, from registration of users, charging and discharging of books, online research among other packages in the design.

[4] Describes about the computerised system mainly, this paper's targets to develop a computerised system that can accomplish the activities such as providing easy access to library usage for librarians and users of library. The focal task of library is to collect, organise knowledge. The implementations include a login page where user has to login using username and password and register for acquiring the facilities of the library. It also includes adding member, adding book, transaction modules etc., This paper describes the advantage of using proper management in the information system and sustainability of library systems. Conclusion: The mission to make life easier and processing faster led to the computerised of various processes. To foster technology-driven education, a web-based library management system has been developed to manage all library operations such as updating user's record and process of borrowing books etc., It is safely determined that the system is an efficient usable and reliable LMS.

[5] Describes about the development of library management system, With the enhancement of technology it is imperative to exalt all the systems into a user-friendly manner. The LMS acts as a tool to the transfer traditional libraries into digital libraries. To evoke the library into the technological era, it represents a system called the LMS. It is an automatic system that reduces the work of staff through a single click. The system is customisable and user-configurable one which causes it to use in different organisations. The LMS contains an admin module where it demonstrates the activities of admin. The admin is considered as the authorised person to access LSM. At the time of login, the system is loaded and opens the homepage where the users enter their id and password and get logged in.

Conclusion: The hindrance and issues of the traditional library are identified and promote it to easy access for the libraries. Since every student has a unique id for accessing books from the library, the librarian can check the user details, fine payments etc.,

[6] Describes about the automated library system. In the existing system, all the transactions are done either manually or partially automated, which takes longer on searching the book, calculating the fine and every other transaction. The project LMS aims at developing a fully functional computerised system to maintain day to day activities of the library. This project included different modules such as staff module, search module, dispatch module. These modules are much like the student module which include several facilities to all.

In conclusion, the automated version of the library management system which will benefit the students as well as staff. The library is not an exception to the emerging technology; hence it comes with a good number of advantages when all the activities that take place in it are automated.

1. **SURVEY OF DIFFERENT METHODOLOGIES**

Survey on different methodologies on LMS using the data structures BST, Linked List and Hashing is given below.

**4.1 Binary Search Trees (BST):**

A Library Management System implemented using Binary Search Trees (BST) efficiently organizes and manages library resources. A BST is a hierarchical data structure that ensures quick retrieval and manipulation of data while maintaining sorted order.

Using a BST for the Library Management System provides several benefits. The hierarchical nature of the BST ensures that operations like insertion, deletion, and searching have an average time complexity of O(log n), where n is the number of books. This offers efficient management of large amounts of data. However, keeping the tree balanced is important to prevent degeneration into a skewed tree, which could degrade performance.

**4.2 Linked List:**

The linked list's inherent structure facilitates seamless insertion, deletion, and traversal of data, allowing for intuitive management of the library's inventory. When a new book is acquired, a new node is created and seamlessly integrated into the linked list, ensuring straightforward additions. Deletions are equally straightforward – locating the book's node and adjusting pointers effectively removes it. This dynamic nature allows for an unlimited number of books without the constraints of a fixed array size.

Searching for books is streamlined by traversing the linked list and matching search criteria while displaying the library's contents is achieved through straightforward node traversal. The flexibility of linked lists caters to tracking the availability of books for efficient borrowing and returning operations. By leveraging the benefits of linked lists, the Library Management System optimizes resource organization, access, and overall operational efficiency, enhancing the library experience for both staff and users.

**4.3 Hashing:**

Hashing is a technique commonly employed in computer science and software engineering, including library management systems, to efficiently manage and retrieve data. In the context of a library management system, hashing can be a valuable tool for optimizing data storage and retrieval, enhancing security, and improving overall system performance. Here's how hashing can be utilized in a library management system:1. \*Data Storage and Retrieval Efficiency: \*: Hashing involves using a hash function to convert data (such as book titles, author names, or ISBN numbers) into fixed-size hash codes. These hash codes are used as indices to store and retrieve the corresponding data in a data structure called a hash table. This allows for fast and constant-time access to data, as the system can quickly determine where to find or store information based on the calculated hash code.

2. \*Faster Search Functionality\* and \* Data Security: \*: Hashing enables rapid searches by allowing the system to quickly identify the location of data within the hash table. This is particularly useful for finding books, managing borrower information, and tracking due dates efficiently. Hashing can contribute to data security by applying cryptographic hash functions to sensitive information such as passwords or personal details. Hashing converts this information into a fixed-size hash value, making it difficult for unauthorized users to reverse-engineer the original data. This technique helps protect user privacy and ensures that sensitive information is not stored in plain text.

3 \*Preventing Data Duplication: \* and \*Performance Optimization: \*: Hashing can be used to identify duplicate entries in the system. When new books are added or user records are created, the hash function can generate hash codes for these entries. If a duplicate hash code is generated, the system can quickly identify and handle duplicate data, reducing redundancy in the system. Hashing can significantly improve the performance of data retrieval operations, especially when dealing with large datasets. With a well-designed hash function and hash table structure, the system can provide quick access to information, enhancing user experience and operational efficiency.

Ultimately, a well-designed library management system might utilize a combination of these structures based on different functionalities. The choice should align with the system's specific needs, data access patterns, and trade-offs, ensuring efficient data management, quick searches, and optimal user experiences.

1. **CONCLUSION**

The proposed Library Management System utilizing data structures provides an efficient solution to overcome the limitations of traditional manual systems. By implementing linked lists, hash tables, and binary search trees, the system enhances book organization, enables quick retrieval of book information, and streamlines borrowing and returning operations. The project aims to optimize overall operational efficiency and improve the user experience in the library, ultimately fostering a more effective knowledge-sharing environment.

1. **REFERENCE**

[1]. Tsega Weldu Araya, Ass. Pro. Adhana Mengsteab, International Journal of Engineering Research & Technology (IJERT), (This work is licensed under a Creative Commons Attribution 4.0 International License.), Vol. 9 Issue 10, October-2020

[2]. A.Thendral Mary, S.Ramya, Mr.S.Krishna Murthy, Dr.A.Valarmathi, IJCRT | Volume 5, Issue 4 October 2017 | ISSN: 2320-2882 IJCRT1704024 International Journal of Creative Research Thoughts (IJCRT), ENHANCED LIBRARY MANAGEMENT SYSTEM. 2017

[3]. Ayodeji Iwayemi , Sulaimon Oyeniyi Adebayo, International Journal of Computer Applications (0975 – 8887) Volume 178 – No. 12, May 2019 Development of a Robust Library Management System.

[4] Prof. B.A. Jadhawar, Komal A. Bhosale, Research Paper on Java Interactional Development Environment Programming Tool, International Advanced Research Journal in Science, Engineering and Technology, Volume 4, Special Issue 4, January 2017.

[5] Ashutosh Tripathi & Ashish Srivastava, The Online Library Management System IOSR Journal of Engineering (IOSRJEN), Vol. 2 Issue 2, Feb.2012

[6] A.D. Liwen Liu, Management of an academic e-libray project, International Conference on Information Management, Innovation Management and Idustrial Engineering, vol.54(8),pp.62-71,2011.

**PHASE 02**

**LIBRARY MANAGEMENT SYSTEM**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TITLE** | **PAGENO** |
| 1 | Abstract | 15 |
| 2 | Introduction | 16 |
| 3 | Problem Statement | 17 |
| 4 | Objectives | 18 |
| 5 | Methodology, Algorithm and Flowchart | 19-20 |
| 6 | Implementation | 21-44 |
| 7 | Results Coding Input and Output | 45-51 |
| 8 | Conclusion and future enhancement | 52 |

1. **ABSTRACT**

Library management systems (LMSs) are software applications that help libraries manage their collections, circulation, and other resources. LMSs can track the number of books in a library, the number of books that have been issued, and the number of books that have been returned or renewed. They can also help libraries quickly identify books, issue and reissue books, and manage their data in an efficient and organized manner. LMSs are essential tools for modern libraries, and they can help libraries to operate more efficiently and effectively, and they can provide patrons with a better experience.

1. **INTRODUCTION**

Libraries serve as vital gateways to knowledge dissemination and the cultivation of a conducive learning milieu. Effective administration of their resources and operations is paramount for seamless functionality. Considering this, this project introduces an innovative Library Management System that capitalizes on data structures to adeptly store and swiftly retrieve book-related data. By incorporating linked lists into its framework, the system strives to revolutionize book organization and amplify the efficiency of borrowing and returning processes. In an era where unfettered access to information is pivotal, libraries assume an irreplaceable role. Nonetheless, conventional methods of resource management and operational oversight have encountered constraints that hinder their fluid operation. This project responds with a visionary solution that holds the potential to reshape the landscape of library management.

At its heart, the Library Management System revolves around the infusion of advanced data structures, particularly linked lists. This integration is more than just technological; it signifies a strategic move towards optimizing the arrangement of books. Linked lists facilitate the construction of dynamic book lists, ensuring resourceful storage and expedited retrieval. By doing so, they replace the conventional linear search with a more efficient method, thereby drastically cutting down the time it takes to locate and manage books. The ramifications of this undertaking are extensive. Libraries are poised to experience heightened operational efficiency as the arduous administrative procedures are streamlined. The accessibility of resources stands to witness a notable elevation, ultimately enriching the library experience for patrons. The project's success hinges on a meticulously designed user interface and rigorous testing to authenticate its functionality, reliability, and security.

In summation, the Library Management System project addresses a pivotal need within the domain of library management. Through the judicious integration of linked lists, it reimagines the modus operandi of book organization and management, ushering libraries into an era of heightened effectiveness. This project not only upholds the pivotal role of libraries but also enhances their capacity to facilitate knowledge dissemination and foster learning.

1. **PROBLEM STATEMENT**

The traditional manual systems for managing library resources often suffer from inefficiencies, including time-consuming book organization, difficulties in book retrieval, and delays in the borrowing and returning processes. The lack of a centralized system leads to errors, loss of books, and challenges in maintaining accurate records. Therefore, there is a need for an automated Library Management System that utilizes appropriate data structures to address these issues.

1. **OBJECTIVES**

1. Develop a Library Management System that efficiently organizes book information using data structures such as linked lists.

2. Implement a robust borrowing and returning module that ensures seamless transactions and tracks book availability in real-time.

3. To provide facilities to the students who are under specific categories.

4. Enhance overall operational efficiency, reduce manual efforts, and improve the user experience in the library.

1. **METHODOLOGY, ALGORITHM AND FLOWCHART**

5.1 Pseudo Code:

1.Start

2.Choose from

1. Issue of Books

2. Return of Books

3. Display Student Details

4. Exit

3.If the option is other than these print invalid message.

4.If choice is 1, (Issue Book)

\* Display The list of Books

\* Ask for the book ID

\* Ask for Student Details

\* Ask Reservation and Category

\* Displays “Books Issued” message

5.If the choice is 2, (Return Book)

\* Ask for the book ID

\* The Books will be returned

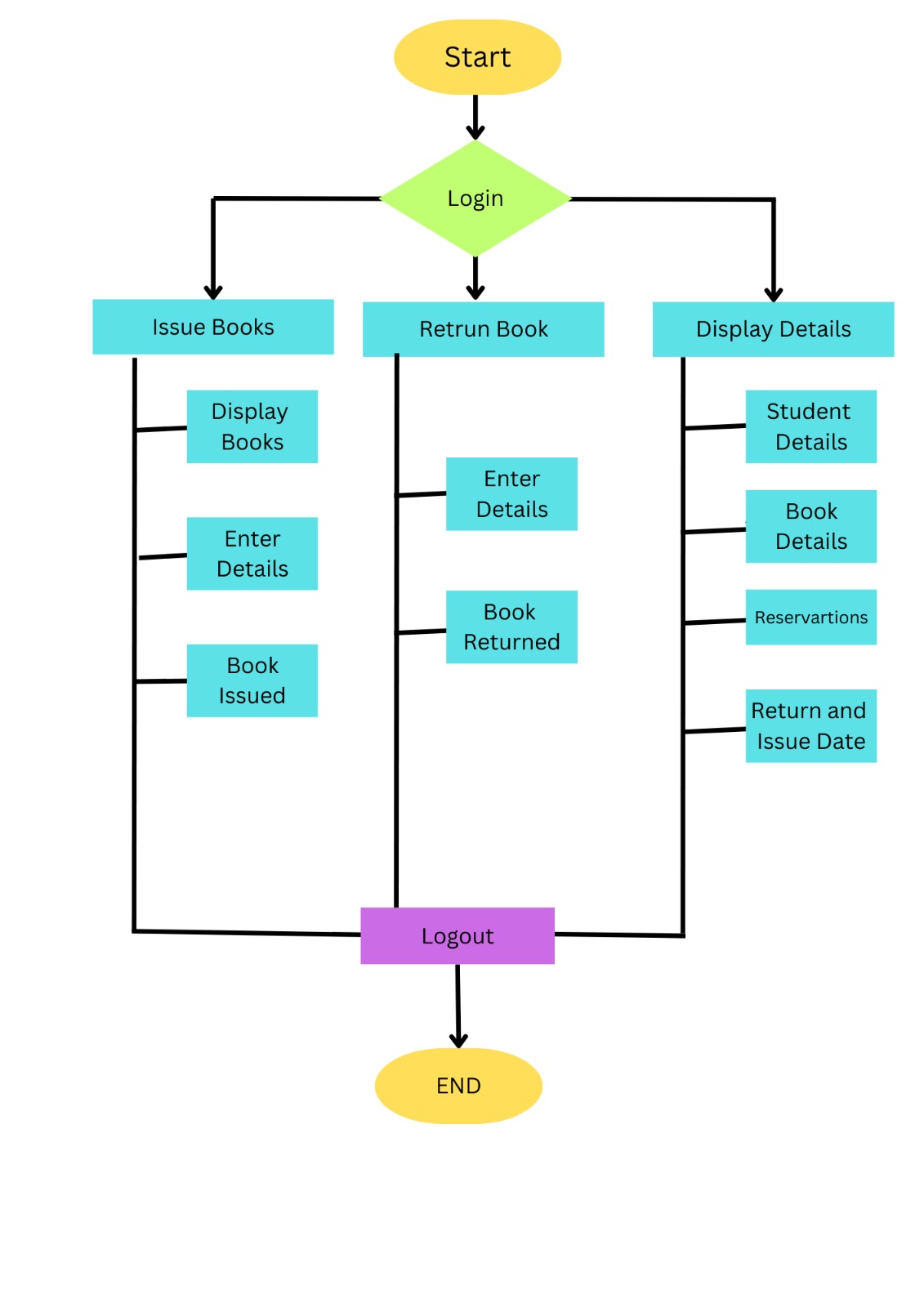
6.If the choice is 3, (Display Student Details)

\* Displays the details of the students who have received the books

7.If the choice is 4,

\* Exit from the program

5.2 Flowchart:



1. **IMPLEMENTATION**

Implementation of the Library Management System using C language is using the Data Structure Linked List. The code is shown below.

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

#include <malloc.h>

#include <string.h>

#include<time.h>

#include<stdbool.h>

struct book{

char name[30];

char author[30];

int id;

int copies;

struct book \*next;

};

struct student{

char name[30];

char email[20];

char book[20];

char a[30];

int id;

int issue\_day ,issue\_mon , issue\_year ;

char reservation;

char category[4];

struct student \*next;

};

struct book \*start\_lib=NULL;

struct student \*start=NULL;

struct book \*initialize\_lib(struct book \*);

struct student \*book\_issue(struct student \*);

struct student \*book\_return(struct student \*);

struct book \*diplay\_lib(struct book \*);

struct book \*delete\_book(int id, struct book \*);

struct book \*add\_book(char [],char [],int);

void display(struct student \*);

void greetings();

void main\_menu();

int is\_leap\_year(int);

int date\_to\_noOfDays(int, int, int);

void book\_return\_date(int , int , int,struct student \*);

void delay(int, int , int,struct student \*);

int main(){

start\_lib=initialize\_lib(start\_lib);

greetings();

main\_menu();

return 0;

}

void greetings(){

printf("\n\n");

printf("\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\t\t\t \* \*\n");

printf("\t\t\t \* \*\n");

printf("\t\t\t \* ---------------------------- \*\n");

printf("\t\t\t \* WELCOME TO STUDENT LIBRARY \*\n");

printf("\t\t\t \* ---------------------------- \*\n");

printf("\t\t\t \* \*\n");

printf("\t\t\t \* \*\n");

printf("\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\n\n");

printf("\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\t\t\t \* \*\n");

printf("\t\t\t \* ------------------------ \*\n");

printf("\t\t\t \* STUDENT LIBRARY \*\n");

printf("\t\t\t \* ------------------------ \*\n");

printf("\t\t\t \* \*\n");

printf("\t\t\t \* \*\n");

printf("\t\t\t \* Dharwad, Karnataka,India \*\n");

printf("\t\t\t \* Email: studentlib@gmail.com \*\n");

printf("\t\t\t \* Contact:8800991010,8800992020 \*\n");

printf("\t\t\t \* \*\n");

printf("\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\n\n\t\t\t Press any key to continue: ");

getch();

}

void main\_menu(){

int choice;

do{

printf("\n\n");

printf("\n\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\n\t\t\t\t MAIN MENU: ");

printf("\n\t\t\t\t 1.ISSUE OF BOOKS ");

printf("\n\t\t\t\t 2.RETURN OF BOOKS ");

printf("\n\t\t\t\t 3.DISPLAY STUDENT DETAILS ");

printf("\n\t\t\t\t 4.EXIT\n ");

printf("\n\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\n\t\t\t\t Enter your choice: ");

scanf("%d",&choice);

switch(choice){

case 1:{

start=book\_issue(start);

break;

}

case 2:{

start=book\_return(start);

break;

}

case 3:{

display(start);

break;

}

case 4:{

exit(1);

}

default:{

printf("\n\t\t\t\t ...Invalid Option!...\n");

printf("\n\t\t\t\t Press any key to try again: ");

getch();

}

}

}while(choice!=4);

}

struct book \*initialize\_lib(struct book \*start){

struct book \*ptr,\*new\_book1,\*new\_book2,\*new\_book3,\*new\_book4,\*new\_book5;

new\_book1=(struct book \*)malloc(sizeof(struct book));

new\_book1->next=NULL;

start\_lib=new\_book1;

strcpy(new\_book1->name,"The Kite Runner");

strcpy(new\_book1->author,"Khaled Hosseini");

new\_book1->id=101;

new\_book1->copies = 4;

ptr=new\_book1;

new\_book2=(struct book\*)malloc(sizeof(struct book));

new\_book2->next=NULL;

strcpy(new\_book2->name,"To Kill A Mockingbird");

strcpy(new\_book2->author,"Harper Lee");

new\_book2->id=102;

new\_book2->copies = 4;

ptr->next=new\_book2;

ptr=new\_book2;

new\_book3=(struct book\*)malloc(sizeof(struct book));

new\_book3->next=NULL;

strcpy(new\_book3->name,"The Alchemist");

strcpy(new\_book3->author,"Paulo Coelho");

new\_book3->id=103;

new\_book3->copies = 4;

ptr->next=new\_book3;

ptr=new\_book3;

new\_book4=(struct book\*)malloc(sizeof(struct book));

new\_book4->next=NULL;

strcpy(new\_book4->name,"Pride And Prejudice");

strcpy(new\_book4->author,"Jane Austen");

new\_book4->id=104;

new\_book4->copies = 4;

ptr->next=new\_book4;

ptr=new\_book4;

new\_book5=(struct book\*)malloc(sizeof(struct book));

new\_book5->next=NULL;

strcpy(new\_book5->name,"A Tale Of Two Cities");

strcpy(new\_book5->author,"Charles Dickens");

new\_book5->id=105;

new\_book5->copies = 4;

ptr->next=new\_book5;

return start\_lib;

}

struct student \*book\_issue(struct student \*start){

struct book \*ptr;

struct student \*ptr2,\*new\_student;

int i=1,id,flag=0;

int count = 0;

if(start\_lib==NULL){

printf("\n\t\t\t\t No books left in the library to issue!\n\t\t\t\t Sorry for the inconvenience!\n");

printf("\n\t\t\t\tTotal number of books available are:%d\n",count);

}else{

system("cls");

ptr=start\_lib;

printf("\n\t\*\*\*\*\*\* Books Available: \*\*\*\*\*\*\*\n");

while(ptr!=NULL){

printf("\n\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\n\t Book %d",i);

printf("\n\t Book Title: %s",ptr->name);

printf("\n\t Name of Author: %s",ptr->author);

printf("\n\t Book ID: %d",ptr->id);

printf("\n\t No of copies available : %d",ptr->copies);

printf("\n\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

ptr=ptr->next;

i++;

count++;

}

printf("\n\n\t Total number of Books available are:%d",count);

printf("\n\n\t Enter the Book ID: ");

scanf("%d",&id);

ptr=start\_lib;

while(ptr!=NULL){

if(ptr->id==id){

flag=1;

break;

}

ptr=ptr->next;

}

if(flag==1){

ptr=start\_lib;

while(ptr->id!=id){

ptr=ptr->next;

}

new\_student=(struct student \*)malloc(sizeof(struct student));

printf("\n\t Enter Student Details:\n ");

printf("\n\t Enter your Name: ");

scanf("%s",new\_student->name);

printf("\n\t Enter your Email: ");

scanf("%s",new\_student->email);

printf("\n\t Enter 'Y/y' if reservation, else 'N/n': ");

scanf(" %c",&new\_student->reservation);

printf("\n\t Enter you category (APL/BPL)\*case sesnsitive(ALL CAPS): ");

scanf(" %s",new\_student->category);

strcpy(new\_student->book,ptr->name);

strcpy(new\_student->a,ptr->author);

new\_student->id=ptr->id;

new\_student->next=NULL;

time\_t t;

t = time(NULL);

struct tm tm = \*localtime(&t);

new\_student->issue\_day = tm.tm\_mday;

new\_student->issue\_mon = tm.tm\_mon+1;

new\_student->issue\_year = tm.tm\_year+1900;

printf("\n\t Issue of Book ID %d done successfully!\n",new\_student->id);

printf("\n\n\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

if(start==NULL){

start=new\_student;

}else{

ptr2=start;

while(ptr2->next!=NULL){

ptr2=ptr2->next;

}

ptr2->next=new\_student;

}

struct book\* temp = ptr;

start\_lib=delete\_book(new\_student->id,temp);

printf("\n\n\t Press any key to go to the main menu: ");

getch();

system("cls");

}else{

printf("\n\t\t ...Invalid Option!...\n");

printf("\n\t\t Press any key to try again: ");

getch();

system("cls");

}

}

return start;

}

struct student \*book\_return(struct student \*start){

struct student \*ptr,\*preptr;

char bookname[30],authorname[30];

int flag=0,id,identity,c=0,d=1;

printf("\n\n\t\*\*\*\*\*\* Books Submission: \*\*\*\*\*\*\*\n");

printf("\n\n\t Enter your Book ID: ");

scanf("%d",&identity);

ptr=start;

while(ptr!=NULL){

if(ptr->id==identity){

flag=1;

break;

}

ptr=ptr->next;

}

if(flag==1){

ptr=start;

while(ptr!=NULL){

c++;

ptr=ptr->next;

}

ptr=start;

while(ptr->id!=identity){

d++;

ptr=ptr->next;

}

ptr=start;

if( d==1 ){

printf("\n\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\n\t Student Name: %s",start->name);

printf("\n\t Student Email: %s",start->email);

printf("\n\t Name of Book Issued: %s",start->book);

printf("\n\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\n\n\t Return of Book ID %d done successfully!\n",identity);

printf("\n\n\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

strcpy(bookname,start->book);

strcpy(authorname,start->a);

id=start->id;

start=start->next;

free(ptr);

add\_book(bookname,authorname,id);

}else{

ptr=start;

while(ptr->id!=identity){

preptr=ptr;

ptr=ptr->next;

}

printf("\n\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\n\t Student Name: %s",ptr->name);

printf("\n\t Student Email: %s",ptr->email);

printf("\n\t Name of Book Issued: %s",ptr->book);

printf("\n\t Book ID: %d",ptr->id);

printf("\n\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

strcpy(bookname,ptr->book);

strcpy(authorname,ptr->a);

id=ptr->id;

preptr->next=ptr->next;

free(ptr);

add\_book(bookname,authorname,id);

}

printf("\n\t Thank you! \n\t Do visit again! ");

printf("\n\n\t Press any key to go to the main menu: ");

getch();

system("cls");

}else{

printf("\n\tSorry the book doesn't exist! Please recheck the entered ID");

printf("\n\t\t\t\t Press any key to try again: ");

getch();

system("cls");

}

return start;

}

void display(struct student \*start){

struct student \*ptr;

ptr=start;

while(ptr!=NULL){

printf("\n\t\*\*\*\*\* Details of Students: \*\*\*\*\*\*\n");

printf("\n\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\n\t\t Student Name: %s",ptr->name);

printf("\n\t\t Student Email: %s",ptr->email);

printf("\n\t\t Name of Book Issued: %s",ptr->book);

printf("\n\t\t Book ID: %d",ptr->id);

printf("\n\t\t Reservation (Y/y - yes,N/n-No): %c",ptr->reservation);

printf("\n\t\t Category : %s",ptr->category);

printf("\n\t\t Issue date ID: %d/%d/%d",ptr->issue\_day,ptr->issue\_mon, ptr->issue\_year);

book\_return\_date(ptr->issue\_day,ptr->issue\_mon, ptr->issue\_year,ptr);

delay(ptr->issue\_day,ptr->issue\_mon, ptr->issue\_year, ptr);

printf("\n\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\n\n\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

ptr=ptr->next;

}

if (ptr == NULL)

printf("\n\n\t\t No BOOKS are issued");

printf("\n\n\t Press any key to go to the main menu: ");

getch();

system("cls");

}

struct book \*delete\_book(int id, struct book \*del){

struct book \*ptr,\*preptr;

int c=0;

ptr=start\_lib;

while(ptr!=NULL){

c++;

ptr=ptr->next;

}

if(del->copies==1){

if(c==1){

ptr=start\_lib;

start\_lib=NULL;

free(ptr);

}else if(start\_lib->id==id){

ptr=start\_lib;

start\_lib=start\_lib->next;

free(ptr);

}else{

ptr=start\_lib;

while(ptr->id!=id){

preptr=ptr;

ptr=ptr->next;

}

preptr->next=ptr->next;

free(ptr);

}

}

else{

del->copies= del->copies-1;

}

return start\_lib;

}

struct book \*add\_book(char bookname[30],char authorname[30],int ide){

struct book \*ptr,\*new\_book;

ptr = start\_lib;

while(ptr->id!=ide && ptr!=NULL)

ptr=ptr->next;

if(ptr!=NULL){

ptr->copies++;

}

else{

new\_book=(struct book \*)malloc(sizeof(struct book));

strcpy(new\_book->name,bookname);

strcpy(new\_book->author,authorname);

new\_book->id=ide;

new\_book->copies = 4;

new\_book->next=NULL;

if(start\_lib==NULL){

start\_lib=new\_book;

}else{

ptr=start\_lib;

while(ptr->next!=NULL){

ptr=ptr->next;

}

ptr->next=new\_book;

}

}

return start\_lib;

}

int is\_leap\_year(int year){

int flag= 0;

if(year % 400 == 0)

flag =1;

else if (year % 100 == 0)

flag= 0;

else if (year % 4 == 0)

flag = 1;

else

flag = 0;

return flag;

}

int date\_to\_noOfDays(int day , int mon ,int year){

int date=day;

switch (mon){

case 1: break;

case 2: date+=31;

break;

case 3: date+=59;

break;

case 4: date+=90;

break;

case 5: date+=120;

break;

case 6: date+=151;

break;

case 7: date+=181;

break;

case 8: date+=212;

break;

case 9: date+=243;

break;

case 10: date+=273;

break;

case 11: date+=304;

break;

case 12: date+=334;

break;

if(is\_leap\_year(year)==1){

if (mon != 1 && mon !=2)

date++;

}

}

return date;

}

void book\_return\_date(int date , int mon , int year, struct student\* ptr){

int var;

if((ptr->reservation=='N'||ptr->reservation=='n')&& (strcmp(ptr->category,"APL")==0)){

var = 15;

}

else if(ptr->reservation=='y' || ptr->reservation=='Y'|| strcmp(ptr->category,"BPL")==0){

var= 30;

}

if(mon == 1 || mon == 3 || mon == 5|| mon == 7 ||mon == 8 ||mon == 10){

date = date+var;

while(date > 31){

mon++;

date = date - 31;

}

}

else if (mon == 4||mon == 6||mon == 9||mon == 11){

date=date+var;

while(date > 30){

date = date-30;

mon++;

}

}

else if (mon==2){

if(is\_leap\_year(year)==1){

date=date+var;

while(date > 29){

date = date-29;

mon++;

}

}

else {

date=date+var;

while(date > 28){

date = date-28;

mon++;

}

}

}

else {

if(date+var > 31){

date = date+var-31;

mon=1;

year++;

}

else

date+=var;

}

printf("\n\t\t Return date = %d/%d/%d" , date , mon , year);

}

void delay (int date , int mon , int year,struct student \*ptr){

int var;

if(ptr->reservation=='N'||ptr->reservation=='n'){

var = 15;

}

else if(ptr->reservation=='y' || ptr->reservation=='Y'){

var= 30;

}

time\_t t;

t = time(NULL);

struct tm tm = \*localtime(&t);

int delay\_date = tm.tm\_mday;

int delay\_mon = tm.tm\_mon+1;

int delay\_year = tm.tm\_year+1900;

int issuedate = date\_to\_noOfDays(date , mon , year);

int todaydate = date\_to\_noOfDays(delay\_date ,delay\_mon, delay\_year);

int difference;

int y = delay\_year - year;

difference = (365\*y) + todaydate - issuedate ;

if(difference > var){

printf("\n\t\t The book is due from %d days" ,difference-var);

}

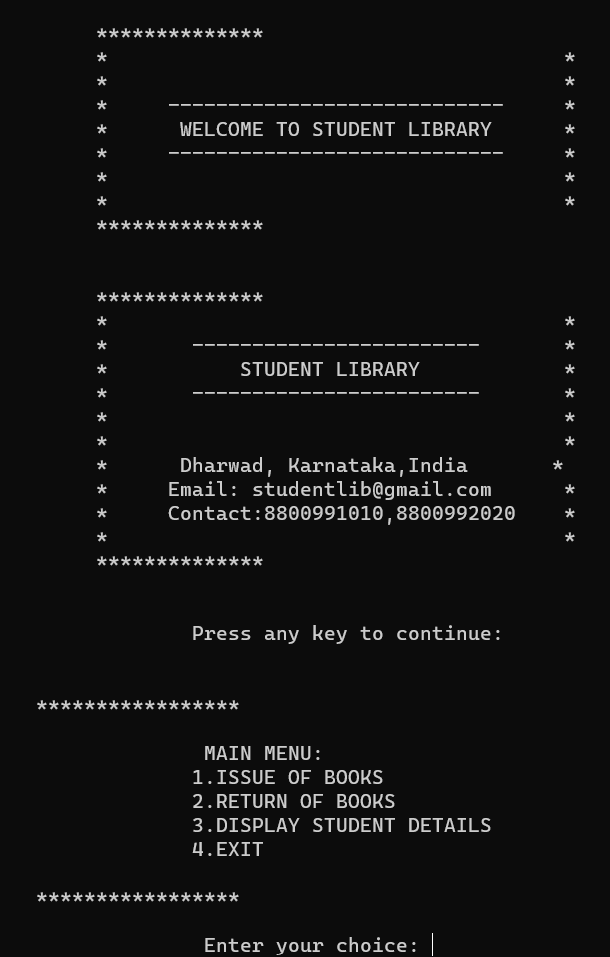
else

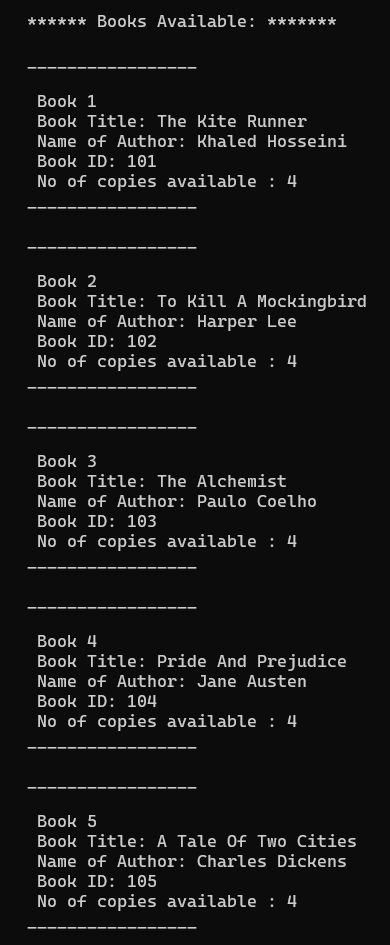
printf("\n\t\t No dues");

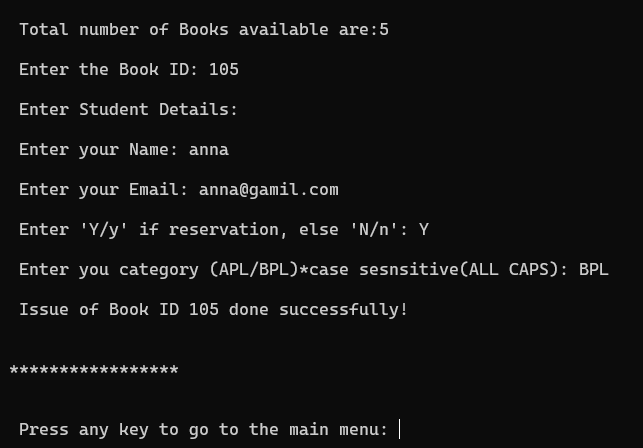
}

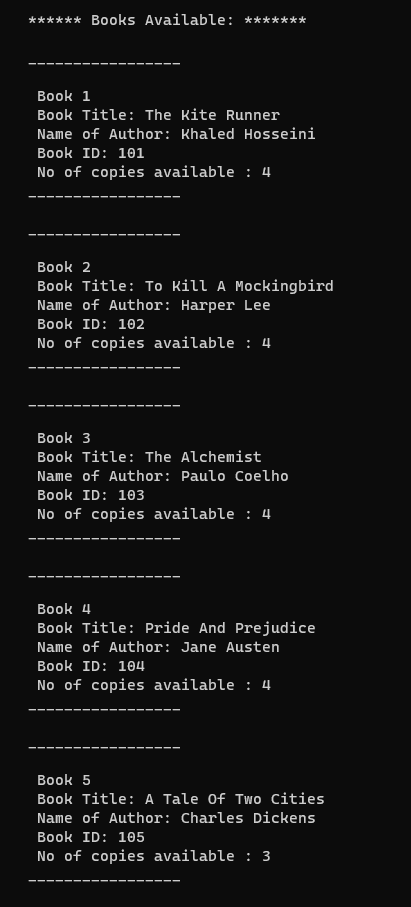
1. **RESULTS CODING INPUT AND OUTPUT**

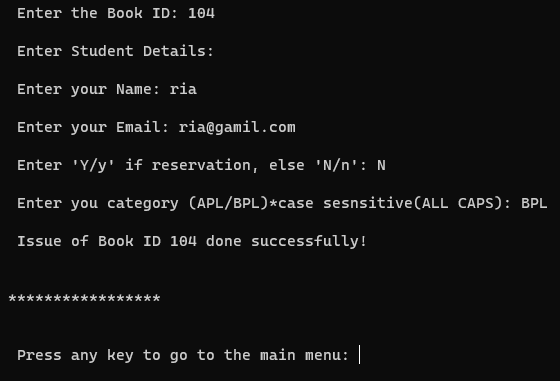
Output of the implementation of the Library Management System is with different sets of inputs is shown below.

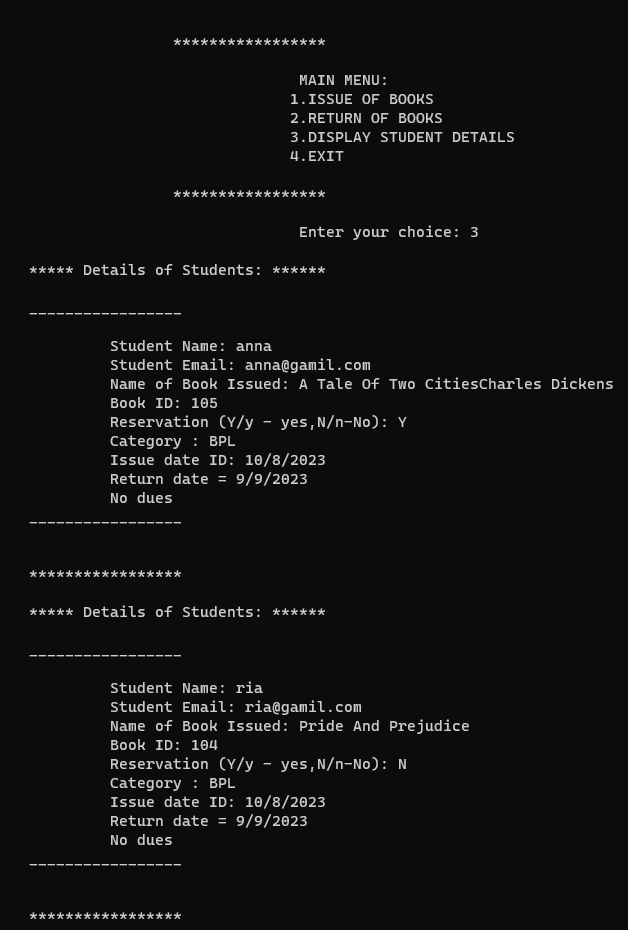


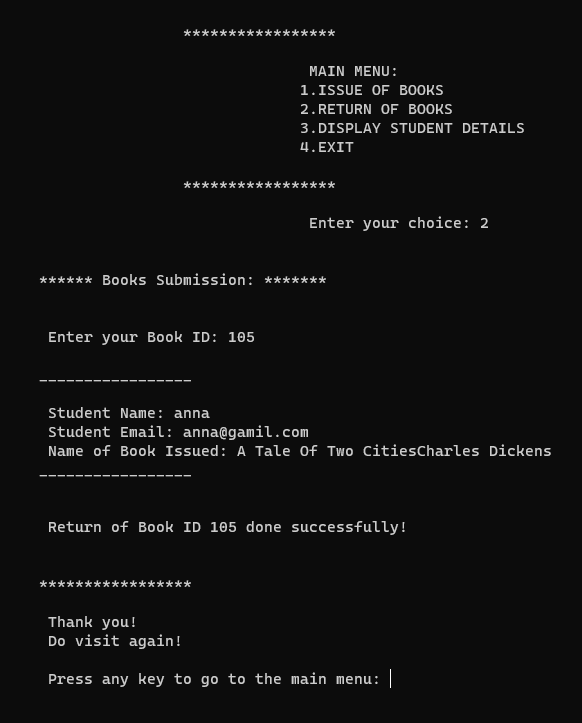


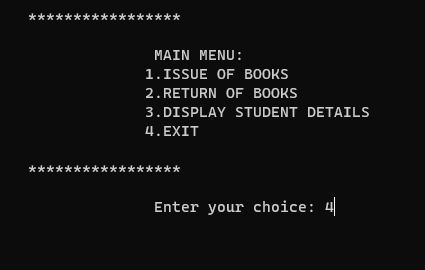












1. **CONCLUSION**

A Library Management System is designed to make the management of books along with student’s data such as issuing of books, number of available books, editing the books in a easy and efficient manner. The Library Management System project is created for meeting the ongoing needs of digitalization of library. It helps to keep track of number of books available, deleting books, searching for a particular book based on book-id, book name or author name. This library management system is an easy way to find and store information.