

Presented by: **Maryam Siddiqui**

**www.linkedin.com/in/maryam-siddiqui-ghani**

Task: **Library Management System**

**Project Report: Library Management System**

**Introduction**

The "Library Management System" project is a console-based application developed in C++ as an internship task for SoftCode. The objective of this project is to create a user-friendly software system for managing a library's collection of books. The system facilitates actions such as adding books, searching for books, issuing books to students, returning books, and managing the library's inventory.

**Project Scope**

The scope of this project involves designing and implementing a software solution that allows both students and librarians to interact with the library's book collection. The system includes features such as inserting new books, searching for books by various criteria, issuing books to students, returning books, listing all books, and sorting books by publication year.

**Features**

User-friendly console interface for students and librarians to interact with the library system.

Adding new books to the library's collection with details such as title, author, publisher, year, and ISBN.

Searching for books by title, author, ISBN, and publisher.

Issuing books to students based on title, ISBN, author, or publisher.

Returning books to the library and calculating fines for late submissions.

Listing all books in the library's collection.

Sorting books by publication year for easy reference.

**Implementation**

The "Library Management System" project is implemented using C++ programming language and structured into functions for modularity and ease of use.

**Functions:**

Functions like insert, issuebytitle, issuebyisbn, searchbytitle, searchbyisbn, searchbyauthor, searchbypublisher, printlist, countnodes, sortbyyear, returnbook, etc., are designed to manage various aspects of the library system.

The code provided implements a basic Library Management System with features such as adding books, issuing books to students, searching for books by various criteria, returning books, and librarian-specific actions like listing all books and sorting them by year.

Let's break down the main components of the code:

**Class book:**

This class represents a book and its attributes, such as title, author, publisher, year, ISBN, and a pointer to the next book in the list.

The constructor book::book initializes the attributes of a book when a new book object is created.

**Function Definitions:**

* **void insert(bookptr& root):**

Allows the librarian to insert a new book into the library's collection by inputting its details.

Creates a new book node and adds it to the linked list of books.

* **void issuebytitle(bookptr& root):**

Allows a student to issue a book based on its title.

Locates the book based on the title and removes it from the list.

* **void issuebyisbn(bookptr& root), void issuebyauthor(bookptr& root), void issuebypublisher(bookptr& root):**

Similar to issuebytitle, these functions allow students to issue books based on ISBN, author, and publisher.

* **bookptr locatenode(bookptr temp, string titl), bookptr locatenodepub(bookptr temp, string pub), bookptr locatenodeisbn(bookptr temp, string isb):**

Locates a specific book node in the linked list based on title, publisher, or ISBN, respectively.

* **void searchbyisbn(bookptr temp), void searchbyauthor(bookptr temp), void searchbytitle(bookptr temp), void searchbypublisher(bookptr temp):**

Allows students to search for books based on ISBN, author, title, or publisher, respectively.

* **void printlist(bookptr temp), void printlist1(bookptr temp):**

Displays the details of all books in the library. printlist1 displays sorted books by year.

* **int countnodes(bookptr temp):**

Counts the number of books in the linked list.

* **void sortbyyear(bookptr& root):**

Sorts the books in the linked list based on the publication year.

* **void returnbook(bookptr& root):**

Allows students to return a book, which is added back to the library collection.

Calculates and displays a fine for late submissions.

* **Main Function:**

The main function contains the user interface for the Library Management System.

It offers options for logging in as a student or librarian and provides corresponding functionalities.

**Object-Oriented Approach:**

The book class is used to define the attributes of a book, and each book is represented as an object of this class.

The linked list structure (bookptr) is utilized to organize and manage the collection of books in the library.

**User Interaction:**

The project offers a user-friendly console interface for both students and librarians to perform various actions such as searching for books, issuing books, returning books, and managing the library's inventory.

**Conclusion**

The "Library Management System" project demonstrates the application of programming concepts to create a functional software solution for managing a library's book collection. By offering features like book insertion, searching, issuing, returning, and sorting, the project aims to enhance the efficiency of library operations and provide a user-friendly experience for both students and librarians.

**Acknowledgments**

Maryam Siddiqui as part of an internship task successfully completed this project for SoftCode. Grateful acknowledgment is extended to mentors and guides who provided valuable support throughout the development process.

**Contact Information**

For inquiries or feedback related to this project, please contact:

Note: This project report template serves as a starting point and can be customized further to match your specific project's details and requirements.

Feel free to expand and tailor this project report template according to your project's specifics.

This project report template provides a detailed overview of your Library Management System project, covering its purpose, features, implementation details, and acknowledgments. You can further customize it to include more specific information about your project's challenges, successes, and any additional features or considerations.

**Source code:**

#include <iostream>

#include <string>

using namespace std;

class book

{

public:

string title;

string author;

string publisher;

int year;

string isbn;

book\* next;

book(string, string, string, int, string, book\*);

};

book::book(string temptitle, string tempauthor, string temppublisher, int tempyear, string tempisbn, book\* tempnext)

{

title=temptitle;

author=tempauthor;

publisher=temppublisher;

year=tempyear;

isbn=tempisbn;

next=tempnext;

}

typedef book\* bookptr;

void getline(istream &stream, int &num, char delimiter)

{ int temp;

stream >> temp;

stream.ignore(500, delimiter);

num= temp;

}

void insert (bookptr &root);

void issuebyisbn(bookptr &root);

void issuebytitle(bookptr &root);

void issuebyauthor(bookptr &root);

void issuebypublisher(bookptr &root);

bookptr locatenodeisbn(bookptr temp, string isb);

bookptr locatenode(bookptr temp, string titl);

void searchbyisbn(bookptr temp);

void searchbyauthor(bookptr temp);

void searchbytitle(bookptr temp);

void searchbypublisher(bookptr temp);

void printlist(bookptr temp);

int countnodes(bookptr temp);

void sortbyyear(bookptr &root);

void printlist1(bookptr temp);

void insert (bookptr &root)

{

string titl, aut, pub, isb;

int yea;

cout << "\t\tTitle:\t\t";

cin.ignore(500,'\n');

getline(cin, titl, '\n');

cout << "\t\tAuthor:\t\t";

getline(cin, aut, '\n');

cout << "\t\tPublisher:\t";

getline(cin,pub, '\n');

cout << "\t\tYear:\t\t";

getline(cin,yea, '\n');

cout << "\t\tIsbn:\t\t";

getline(cin, isb, '\n');

root = new book (titl, aut, pub, yea, isb, root);

cout<<"\n\t\tBook is added in the library list successfully."<<endl;

}

void issuebytitle(bookptr &root)

{

string titl;

cout << "\t\tBook title:\t";

cin.ignore(500,'\n');

getline(cin, titl, '\n');

bookptr p = locatenode(root, titl);

if (p == NULL){

cout << "\n\tThis book is not found in the list.\n\t please search another book .\n\n";

}

else if (root == p){

cout<<"\t\tThe book "<<p->title<< " is issued by a student. \n";

root = p->next;

}

else

{

bookptr q = root;

while (q->next != p){

q = q->next;

}

q->next = p->next;

cout<<"\t\tThe book "<<p->title<< " is issued by a student. \n";

}

delete p;

}

bookptr locatenode(bookptr temp, string titl)

{

while (temp != NULL)

{

if (temp->title == titl)

{

return temp;

}

temp = temp->next;

}

return NULL;

}

bookptr locatenodepub(bookptr temp, string pub)

{

while (temp != NULL)

{

if (temp->publisher == pub)

{

return temp;

}

temp = temp->next;

}

return NULL;

}

void issuebyisbn(bookptr &root)

{

string isb;

cout << "\t\tBook isbn:\t\t\t";

cin.ignore(500,'\n');

getline(cin, isb, '\n');

bookptr p = locatenodeisbn(root, isb);

if (p == NULL){

cout << "\n\t\tThis book is not found in the list. please search another book ..\n\n";

}

else if (root == p){

cout<<"\t\tThe book "<<p->isbn<< " is issued by a student. \n";

root = p->next;

}

else

{

bookptr q = root;

cout<<"\t\tThe book "<<p->isbn<< " is issued by a student. \n";

while (q->next != p)

q = q->next;

q->next = p->next;

}

delete p;

}

void issuebyauthor(bookptr &root)

{

string aut;

cout << "\t\tBook isbn:\t\t\t";

cin.ignore(500,'\n');

getline(cin, aut, '\n');

bookptr p = locatenodeisbn(root, aut);

if (p == NULL){

cout << "\n\t\tThis book is not found in the list. please search another book ..\n\n";

}

else if (root == p){

cout<<"\t\tThe book "<<p->author<< " is issued by a student. \n";

root = p->next;

}

else

{

bookptr q = root;

cout<<"\t\tThe book "<<p->author<< " is issued by a student. \n";

while (q->next != p)

q = q->next;

q->next = p->next;

}

delete p;

}

void issuebypublisher(bookptr &root)

{

string pub;

cout << "\t\tBook publisher name:\t\t\t";

cin.ignore(500,'\n');

getline(cin, pub, '\n');

bookptr p = locatenode(root, pub);

if (p == NULL){

cout << "\n\t\tThis book is not found in the list. please search another book ..\n\n";

}

else if (root == p){

cout<<"\t\tThe book "<<p->publisher<<" is issued by a student. \n";

root = p->next;

}

else

{

bookptr q = root;

cout<<"\t\tThe book "<<p->publisher<< " is issued by a student. \n";

while (q->next != p)

q = q->next;

q->next = p->next;

}

delete p;

}

bookptr locatenodeisbn(bookptr temp, string isb)

{

while (temp != NULL)

{

if (temp->isbn == isb)

{

return temp;

}

temp = temp->next;

}

return NULL;

}

void searchbyisbn(bookptr temp)

{

string isb;

int c=0;

cout << "\t\tBook isbn:\t";

cin.ignore(500,'\n');

getline(cin, isb, '\n');

while (temp != NULL)

{

if (isb == temp->isbn)

{

cout << "\t\tTitle of the book: "<<temp->title << "\n\t\tIsbn of the book: "<<temp->isbn<<endl<<" BOOK IS FOUND \n";

c++;

}

temp = temp->next;

}

if(c==0){

cout<<"\t\tNo book is available with this title."<<endl;

}

cout << "\n";

}

void searchbytitle(bookptr temp)

{

string titl;

int c=0;

cout << "\t\tEnter book title:\t";

cin.ignore(500,'\n');

getline(cin, titl, '\n');

while (temp != NULL){

if (temp->title==titl)

{

cout << "\t\tTitle of the book: "<<temp->title <<endl<<" BOOK IS FOUND \n";

c++;

}

temp = temp->next;

}

if(c==0){

cout<<"\t\tThe book is not found with this title. "<<endl;

}

cout << "\n";

}

void printlist(bookptr temp){

if(temp==NULL){

cout<<"\t\tThere is no book in the list of library.\n";

}

else{

while (temp != NULL)

{

cout<<"\t\tTiltle of the book:\t"<<temp->title << "\n";

cout<<"\t\tAuthor of the book:\t"<<temp->author << "\n";

cout<<"\t\tPublisher of the book: "<<temp->publisher <<"\n";

cout<<"\t\tyear\t\t\t"<< temp->year << "\n";

cout<<"\t\tIsbn\t\t\t"<< temp->isbn << "\n\n";

temp = temp->next;

}

}

}

void printlist1(bookptr temp){

if(temp==NULL){

cout<<"\t\tThere is no book in the list of library.\n";

}

else{

cout<<"\n\t\t<<>>====list of sorted books by year====<<>>\n\n";

while (temp != NULL)

{

//for(int i=1;i<temp->year;i++){

cout<<"\t\tBOOK INFORMATION : "<<"YEAR: " <<temp->year<<"\t"<<"TITLE: "<<temp->title <<"\t"<<"AUTHOR: "<<temp->author<<"\t PUBLISHER: "<<temp->publisher<<"\t"<<"ISBN:"<<temp->isbn<< "\n";

temp = temp->next;

}

}

}

void sortbyyear(bookptr &root){

struct book \*current = root;

struct book \*index = NULL;

int temp;

if(root == NULL) {

return;

}

else {

while(current != NULL) {

index = current->next;

while(index != NULL) {

if(current->year> index->year) {

temp = current->year;

current->year = index->year;

index->year = temp;

}

index = index->next;

}

current = current->next;

}

}

}

void searchbyauthor(bookptr temp)

{

string aut;

int c=0;

cout << "\t\tAuthor name:\t\t";

cin.ignore(500,'\n');

getline(cin, aut, '\n');

while (temp != NULL)

{

if (temp->author == aut)

{

cout << "\t\tBook title: "<<temp->title << "\t\t" <<"Author name: " <<temp->author<<endl<<" BOOK IS FOUND \n" ;

c++;

}

temp = temp->next;

}

if(c==0){

cout<<"\t\tNo book is available with this author name."<<endl;

}

cout << "\n";

}

void searchbypublisher(bookptr temp)

{

string pub;

int c=0;

cout << "\t\tPublisher name:\t\t";

cin.ignore(500,'\n');

getline(cin, pub, '\n');

while (temp != NULL)

{

if (temp->publisher == pub)

{

cout << "\t\tBook title: "<<temp->title << "\t\t" <<"Publisher name: " <<temp->publisher<<endl<<" BOOK IS FOUND \n" ;

c++;

}

temp = temp->next;

}

if(c==0){

cout<<"\t\tNo book is available with this publisher name."<<endl;

}

cout << "\n";

}

int countnodes(bookptr temp)

{

int countb = 0;

while (temp != NULL)

{

countb++;

temp = temp->next;

}

return countb;

}

void returnbook(bookptr &root)

{

string titl, aut, pub, isb;

int yea;

cout << "\t\tTitle of the book:\t";

cin.ignore(500, '\n');

getline(cin, titl, '\n');

cout << "\t\tAuthor of the book:\t";

getline(cin, aut, '\n');

cout << "\t\tPublisher of the book:\t";

getline(cin, pub, '\n');

cout << "\t\tYear of the book:\t";

cin >> yea;

cout << "\t\tISBN of the book:\t";

cin.ignore(500, '\n');

getline(cin, isb, '\n');

bookptr newBook = new book(titl, aut, pub, yea, isb, root);

cout << "\n\t\tBook has been returned and added back to the library list successfully.\n";

// Add the returned book to the beginning of the list

newBook->next = root;

root = newBook;

}

//MAIN FUCTIONS

int main()

{

int choice;

int ch,Days,Fine;

int c ,a; string p;

//bookptr root = NULL;

bookptr root = new book("Book 1 Title", "Author 1", "Publisher 1", 2020, "111", NULL);

root->next = new book("Book 2 Title", "Author 2", "Publisher 2", 2018, "222", NULL);

root->next->next = new book("Book 3 Title", "Author 3", "Publisher 3", 2015, "333", NULL);

cout <<endl<<endl<< "\t\t\t\t \t=========================================== ";

cout <<endl<< "\n\t\t\t\t\t\t>>LIBRARY MANAGMENT SYSTEM<< ";

cout<<endl << " \n\t\t\t \t\t===========================================\n ";

do{

cout <<" \n\t\t LOGIN THROUGH : \n";

cout<<"\n\t\t 1.TO INSERT BOOK. "<< " \n\t\t 2.STUDENT \n\t\t 3.LIBRARIAN \n"<<"\n\t\tSELECT THE OPTION:\t";

cin>>ch;

if(ch==1){

do{

cout<<"\n\t\tENTER 1 TO INSERT THE BOOK. ";

cout<<"\n\t\tENTER 2 FOR EXITING FROM THIS SUB-MENU . \n\n\t\tSELECT OPTIONS IN THE SUB MENU:";

cin>>a;

switch(a){

case 1:

insert(root);

break;

case 2:

cout<<"\n\t\tEXIT.\n";

break;

default:

cout<<"\t\tINVALID CHOICE.\n";

break;

}

}while(a!=2);

}

else if(ch==2){

cout<<"\n\t\tYOU ARE LOGIN AS A STUDENT.\n\n \t\t\t\t>>======WELCOME TO LIBRARY======<<\n\n";

cout<<"\t\t1).SEARCH BOOK BY TITLE:\n";

cout<<"\t\t2).SEARCH BOOK BY AUTHOR:\n";

cout<<"\t\t3).SEARCH BOOK BY ISBN:\n";

cout<<"\t\t4).SEARCH BOOK BY PUBLISHER:\n";

cout<<"\t\t5).ISSUE A BOOK BASED ON TITLE\n";

cout<<"\t\t6).ISSUE A BOOK BASED ON ISBN\n";

cout<<"\t\t7).ISSUE A BOOK BASED ON AUTHOR\n";

cout<<"\t\t8).ISSUE A BOOK BASED ON PUBLISHER\n";

cout<<"\t\t9).LIST OF ALL BOOKS.\n";

cout<<"\t\t10).Return Book \n";

cout<<"\t\t11).FOR EXITING FROM THE SUB MENU SELECT 11: \n";

do{

cout<<"\t\t ENTER YOUR CHOICE :\t" ;

cin>>c;

switch(c)

{

case 1:

searchbytitle(root);

break;

case 2:

searchbyauthor(root);

break;

case 3:

searchbyisbn(root);

break;

case 4:

searchbypublisher(root);

break;

case 5:

issuebytitle(root);

break;

case 6:

issuebyisbn(root);

break;

case 7:

issuebyauthor(root);

break;

case 8:

issuebypublisher(root);

break;

case 9:

printlist(root);

break;

case 10:

{returnbook(root);

cout<<"how many days you keep it \t";

cin>>Days;

if(Days<=7)

{

cout<<"your good enough to not cross the submission days Thanks!!!!"<<endl;

}

else

{

cout<<"your late so your fined on your late submission"<<endl;

int fine;int days;

days=Days-7;

fine=days\*150;

cout<<"Amount of fine is per day from 7 days onward is 150/- rupess \n Fine="<<fine<<endl;

}

}

break;

case 11:

cout<<"\t\t YOU ARE EXIT FROM SUB-MENU \n\n";

break;

}

}while (c!=11);

}

else if(ch==3)

{

string pas;

int i=3;

do{

int attempts=0;

cout<<"\n\t\tAS YOU LOGIN AS A LIBRARIAN SO ENTER PASSWORD:\t";

cin>>pas;

while ( ((pas != "pass")&&(pas!=p))&& attempts < 2 ){

cout << "\t\tPLEASE TRY AGAIN YOU ENTER WRONG PASSWORD:\n\t\tEnter again :";

cin>>pas;

attempts++;

}

if(attempts<2){

if(pas=="pass" || pas==p){

cout<<"\n\t\t>>======WELCOME TO LIBRARY======<<\n";

cout<<"\t\t\t\t1.ADD THE BOOK TO THE LIST:\n";

cout<<"\t\t\t\t2.LIST OF ALL BOOKS: \n";

cout<<"\t\t\t\t3.LIST OF SORTED BOOKS BY YEAR: \n";

cout<<"\t\t\t\t4.UPDATE YOUR PASSWORD\n";

cout<<"\t\t\t\t5.LOGOUT:\n";

do{

cout<<"\t\t ENTER YOUR CHOICE :\t";

cin>>c;

switch(c){

case 1:

insert(root);

break;

case 2:

printlist(root);

break;

case 3:

sortbyyear(root);

cout<<"LIST OF SORTED BOOK BY YEAR: ";

printlist1(root);

break;

case 4:

cout<<"ENTER YOUR NEW PASSWORD\n";

cin>>p;

cout<<"YOUR PASSWORD UPDATED SUCCESSFULLY\n";

pas=p;

break;

case 5:

cout<<"\t\tLOGOUT\n";

break;

default:

cout<<"Invalid choice. please enter correct option: "<<endl;

break;

} }while(c!=5);

}

}

else{

cout << "\t\tSORRY, ONLY ALLOWED 3 ATTEMPTS\n";

}

}while(i!=3);

}

} while (ch!=4);

return 0;

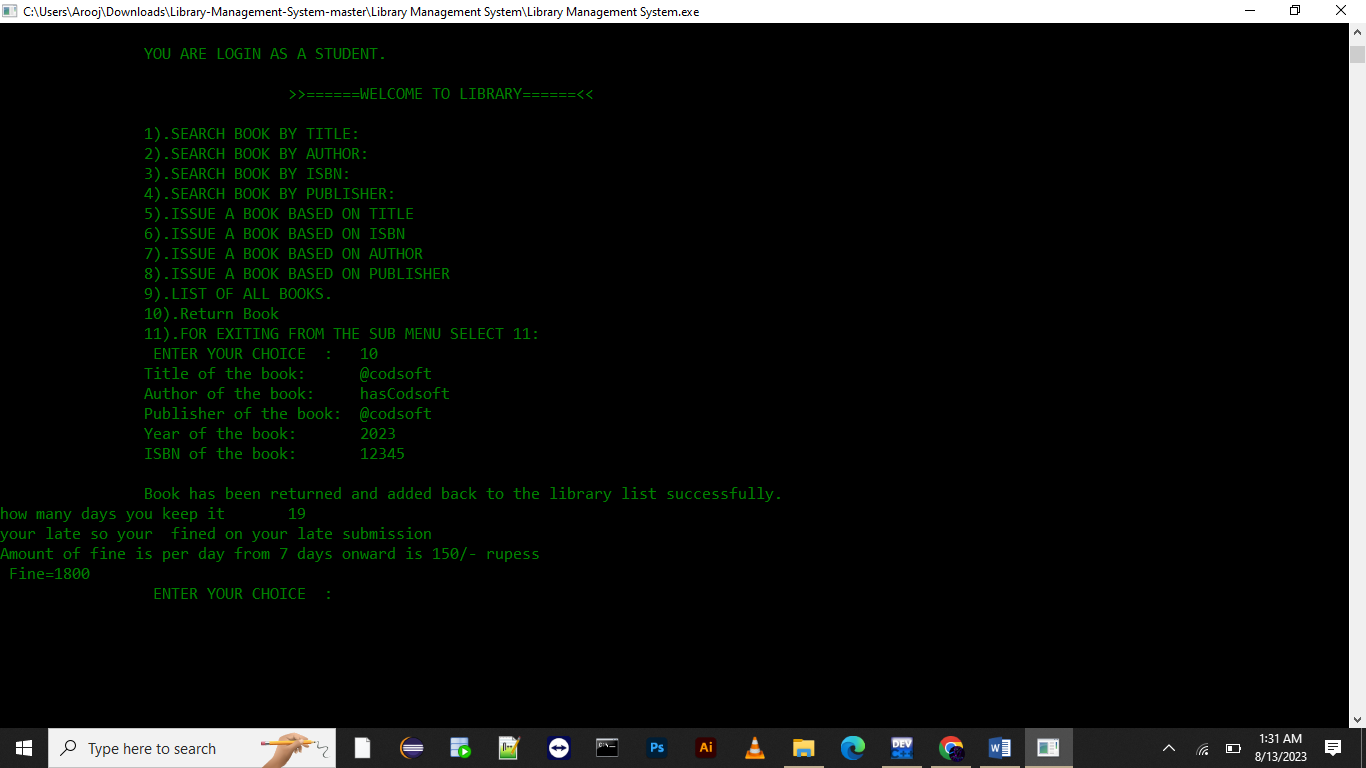
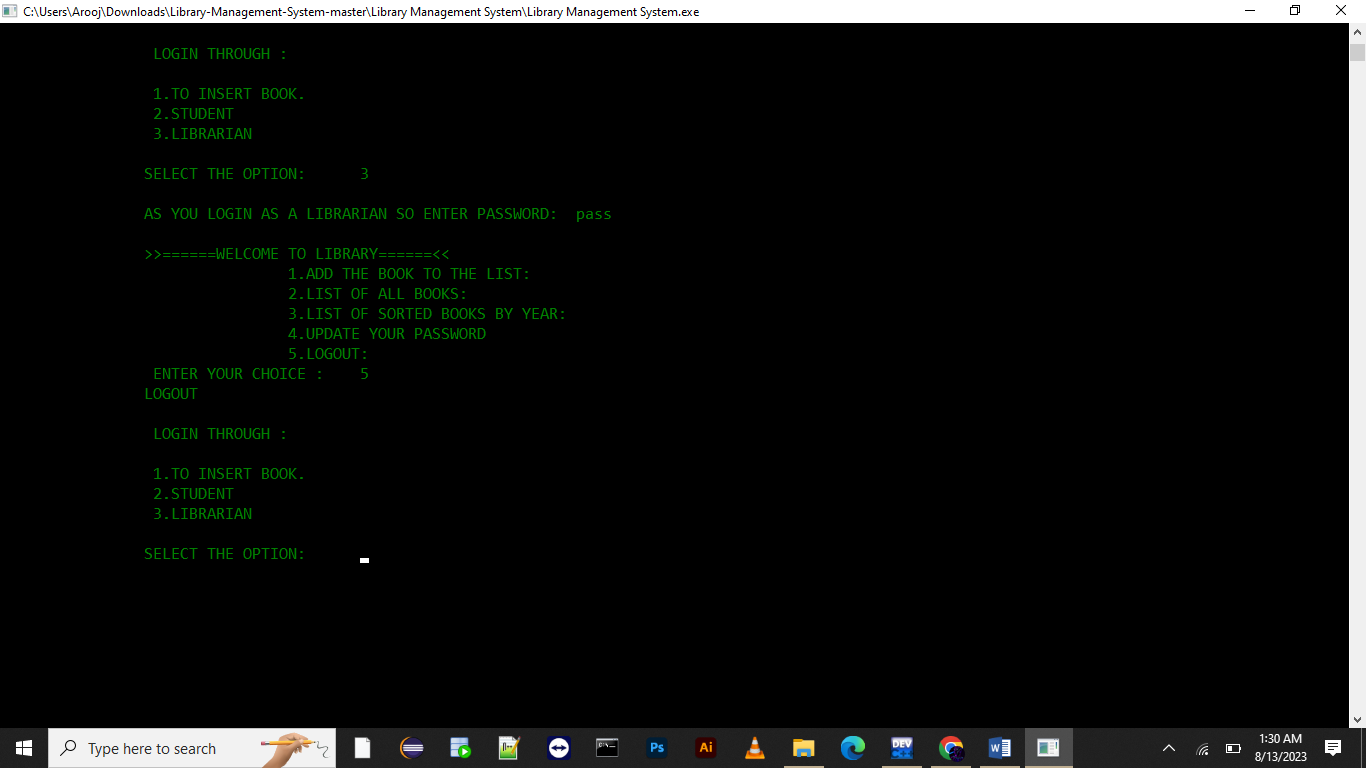
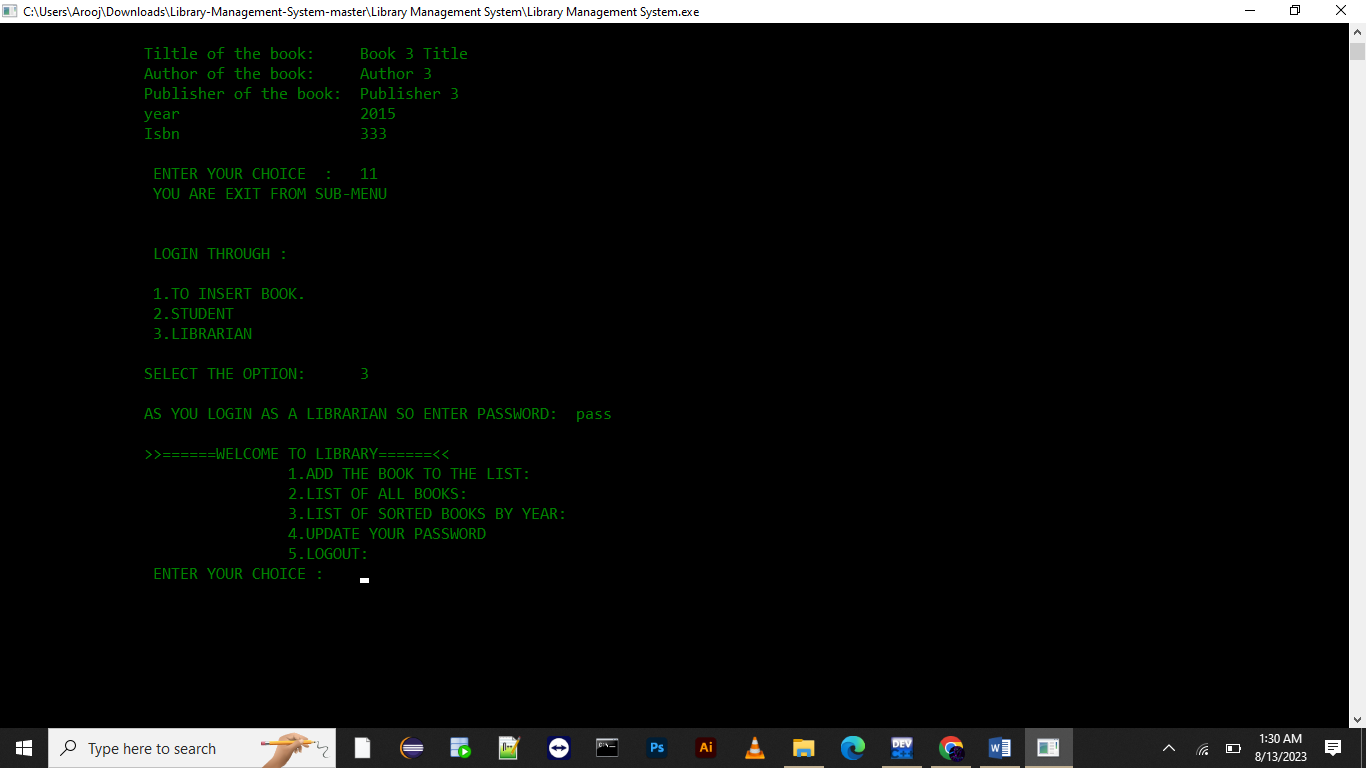
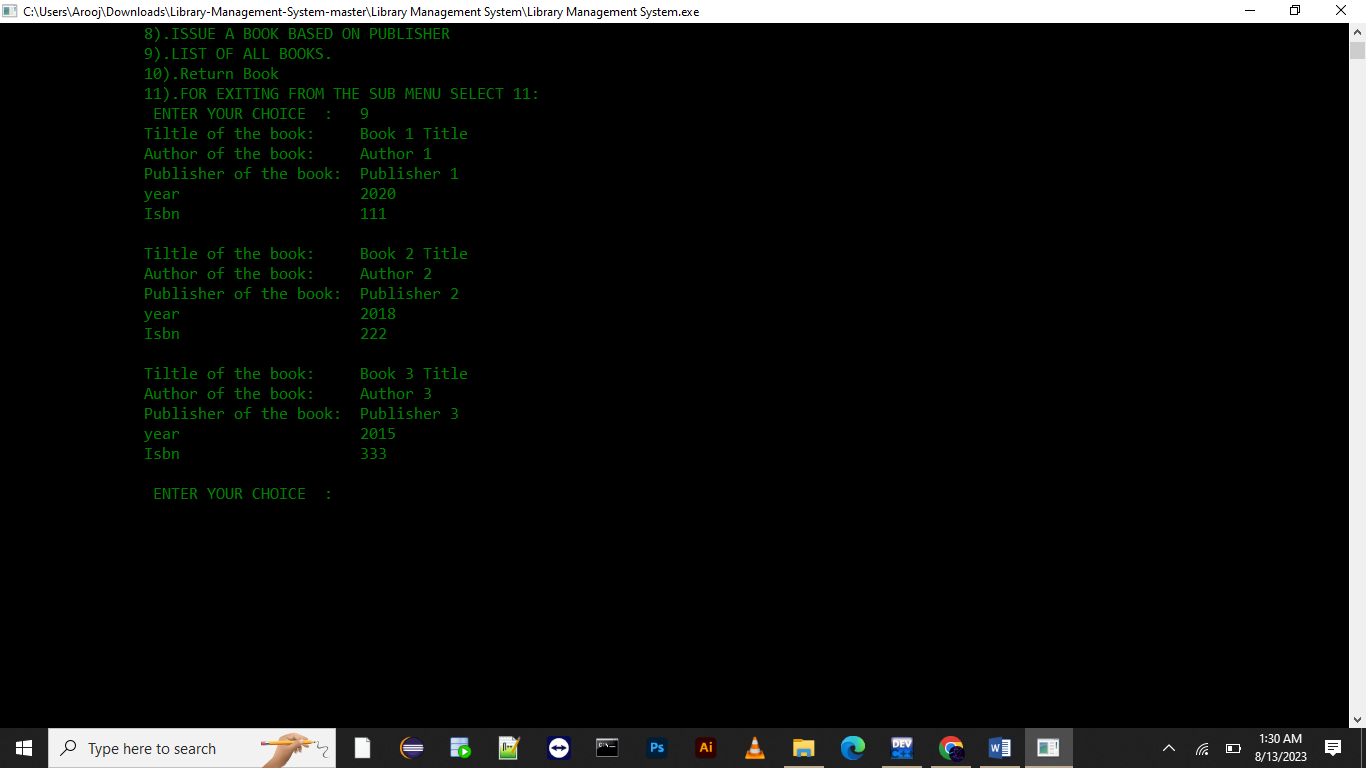
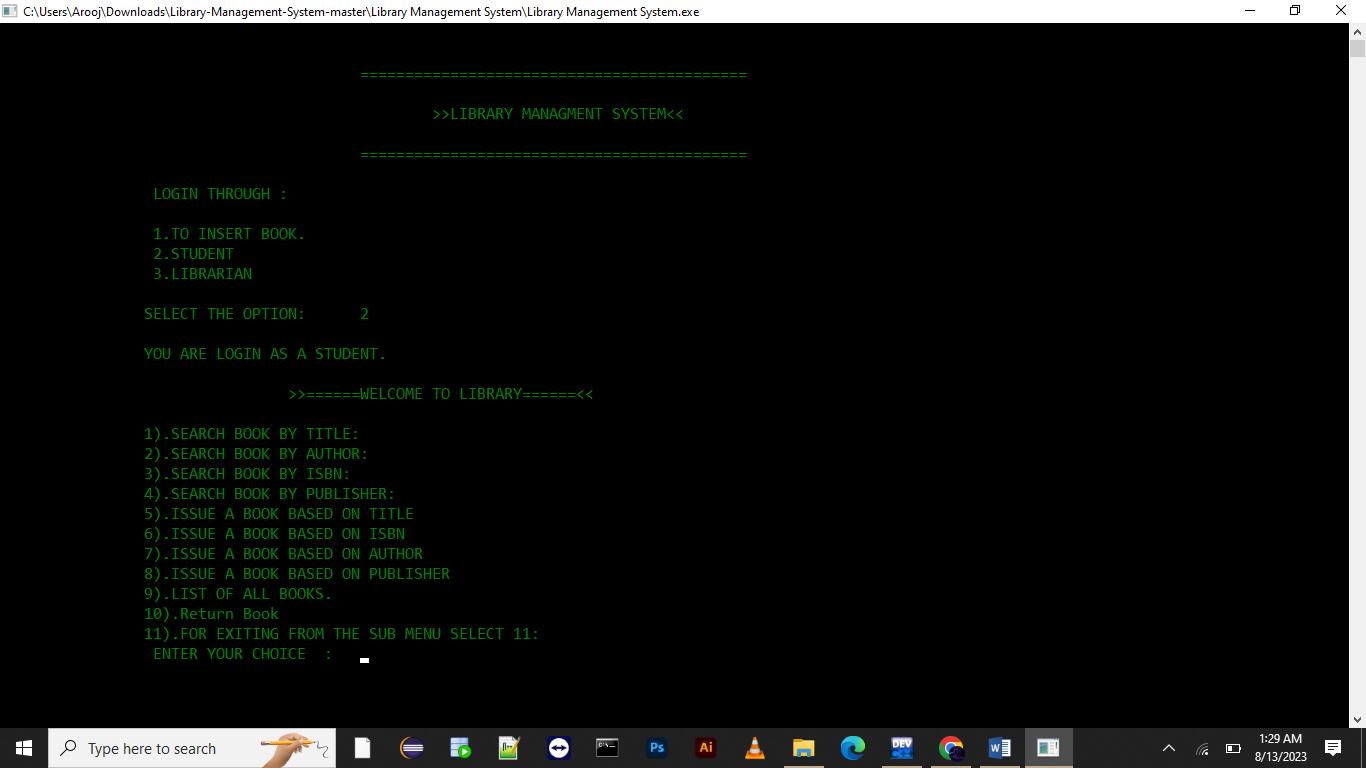
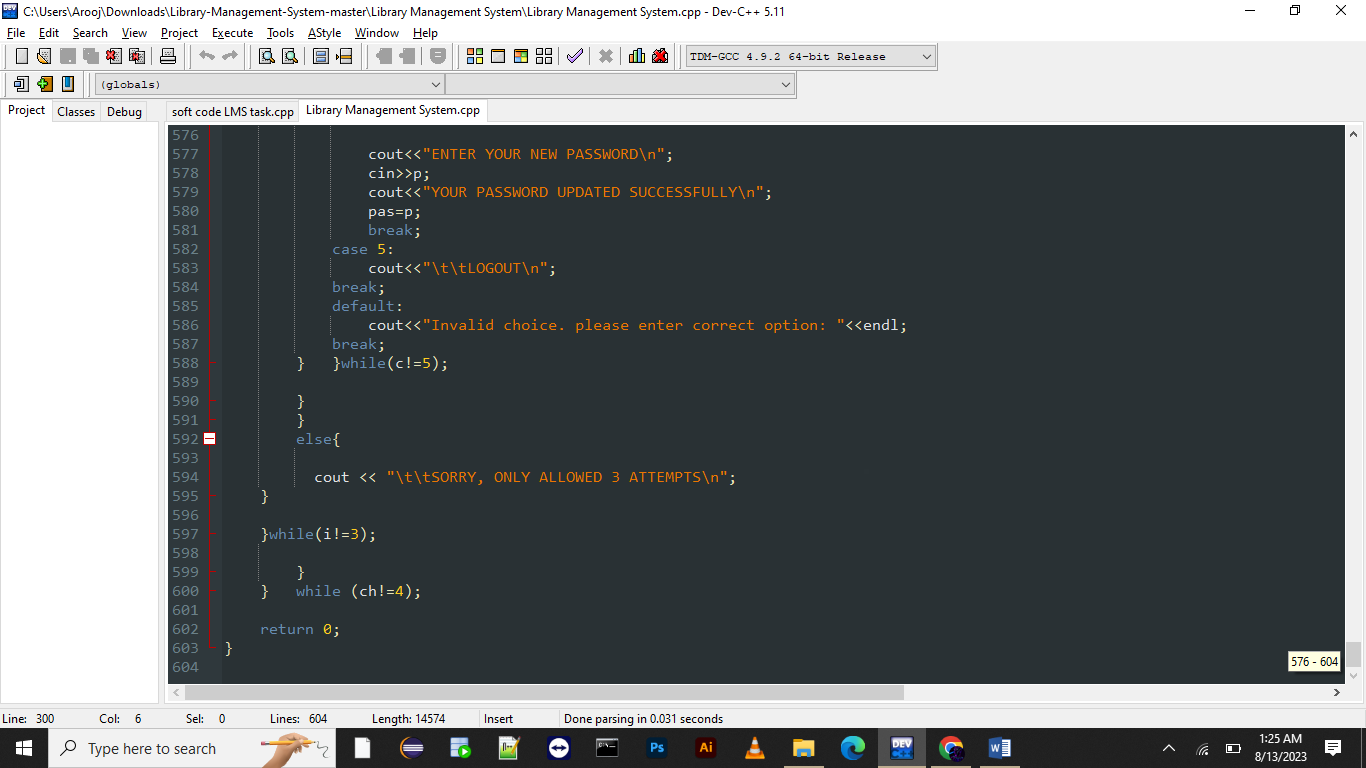
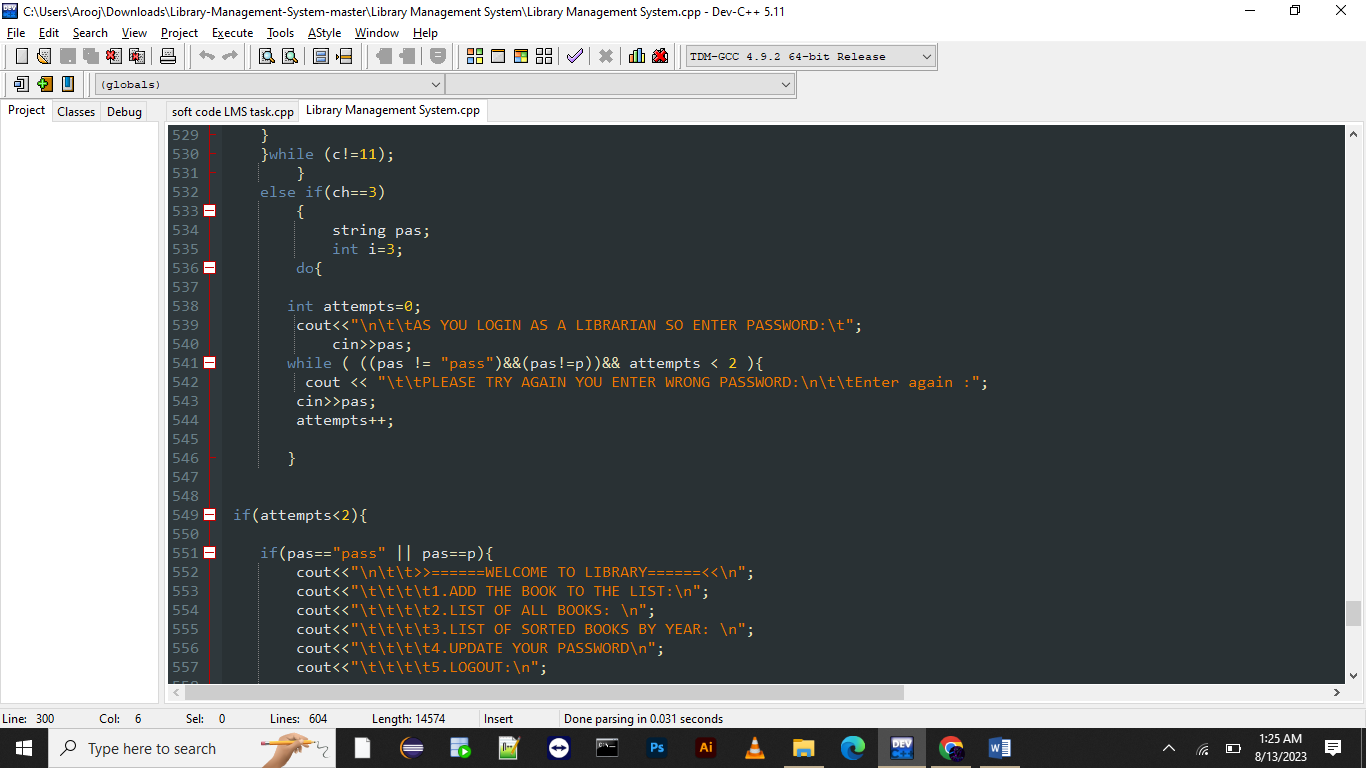
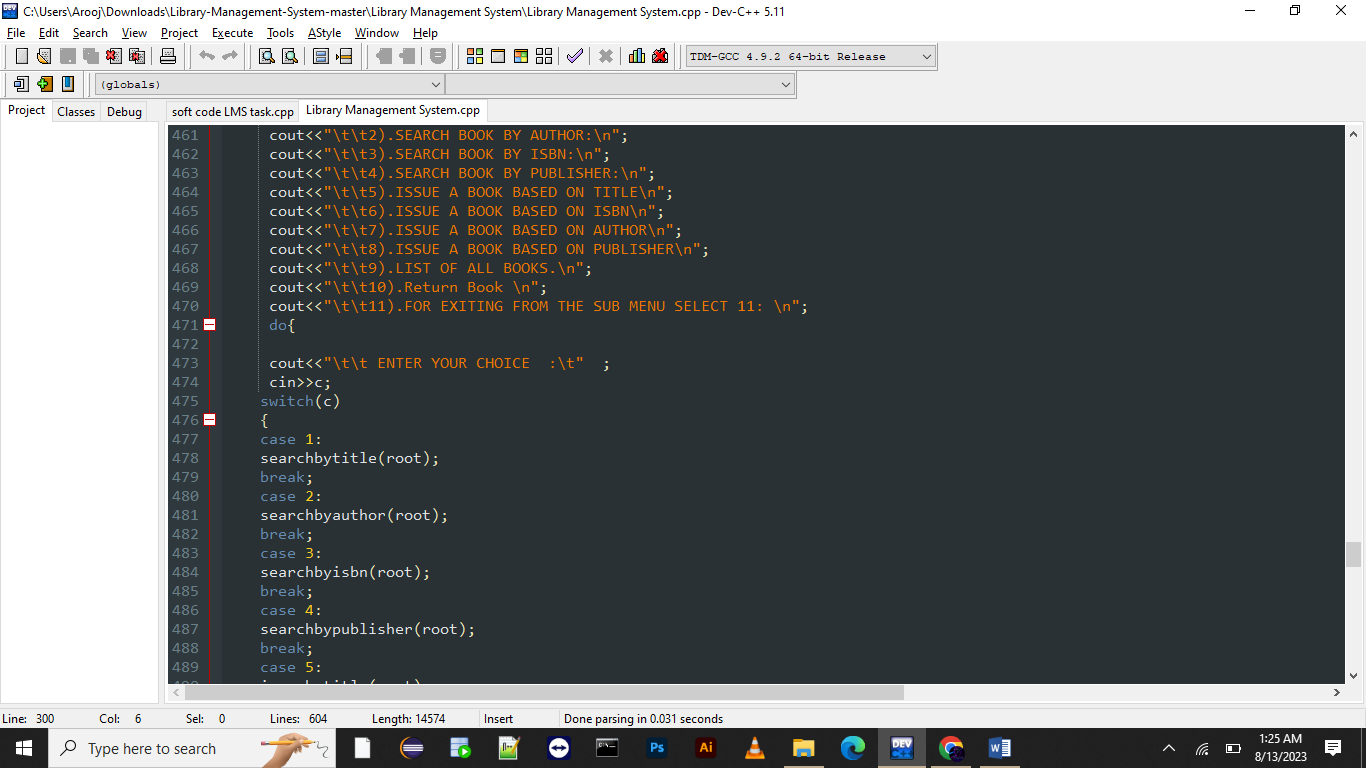
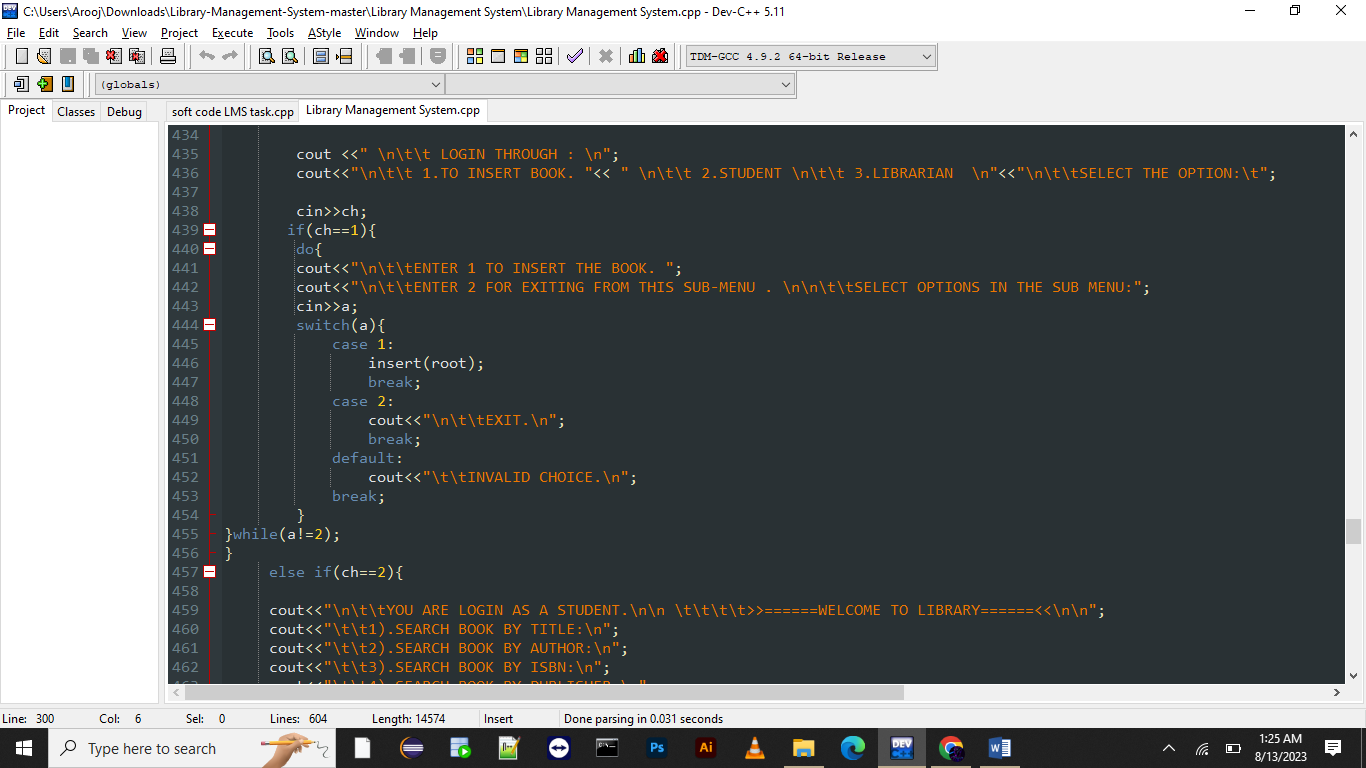
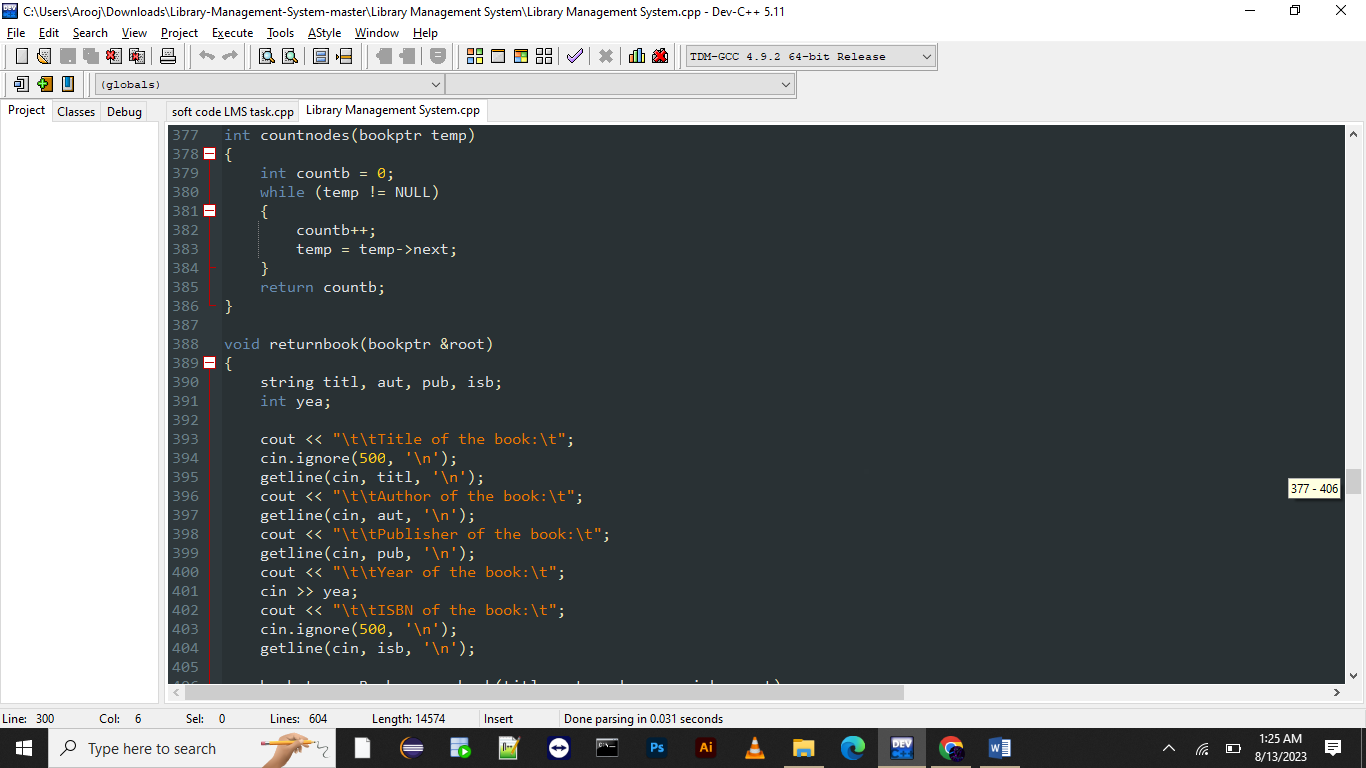
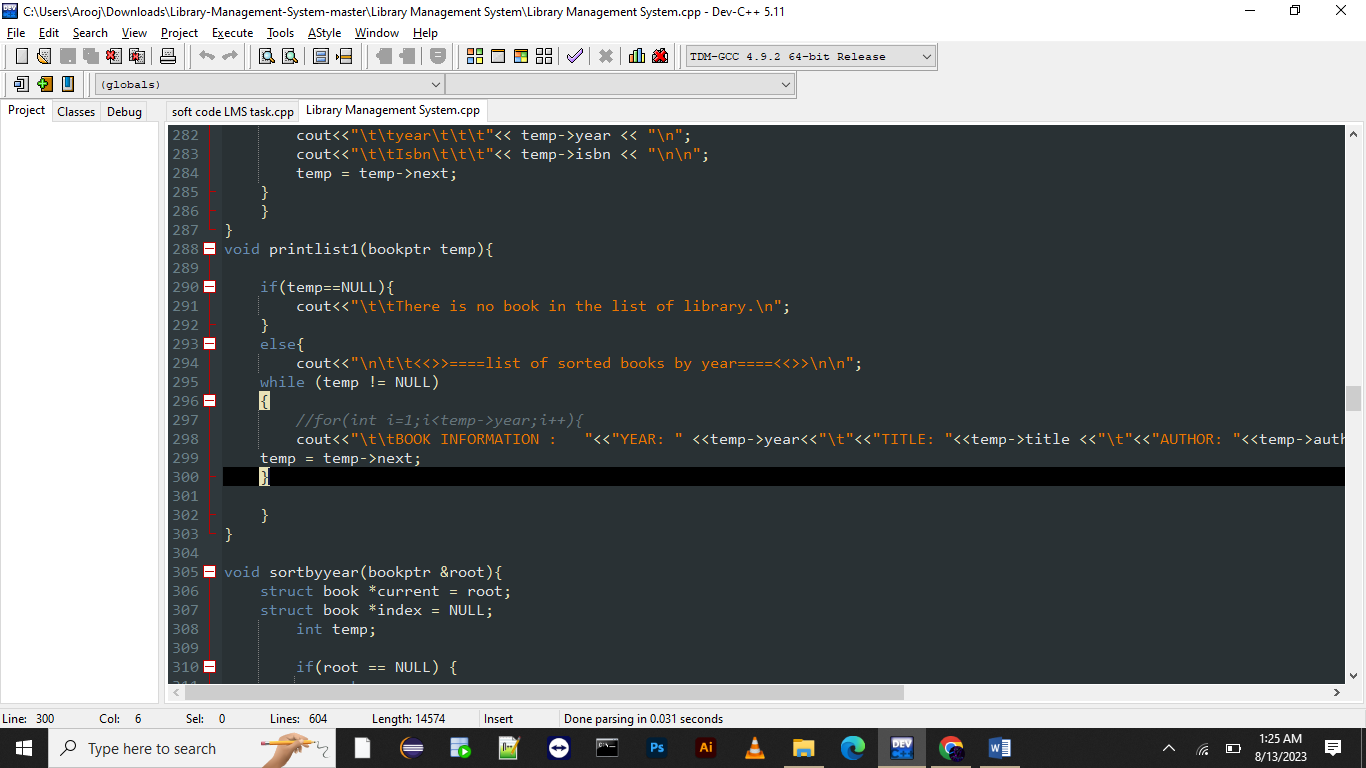
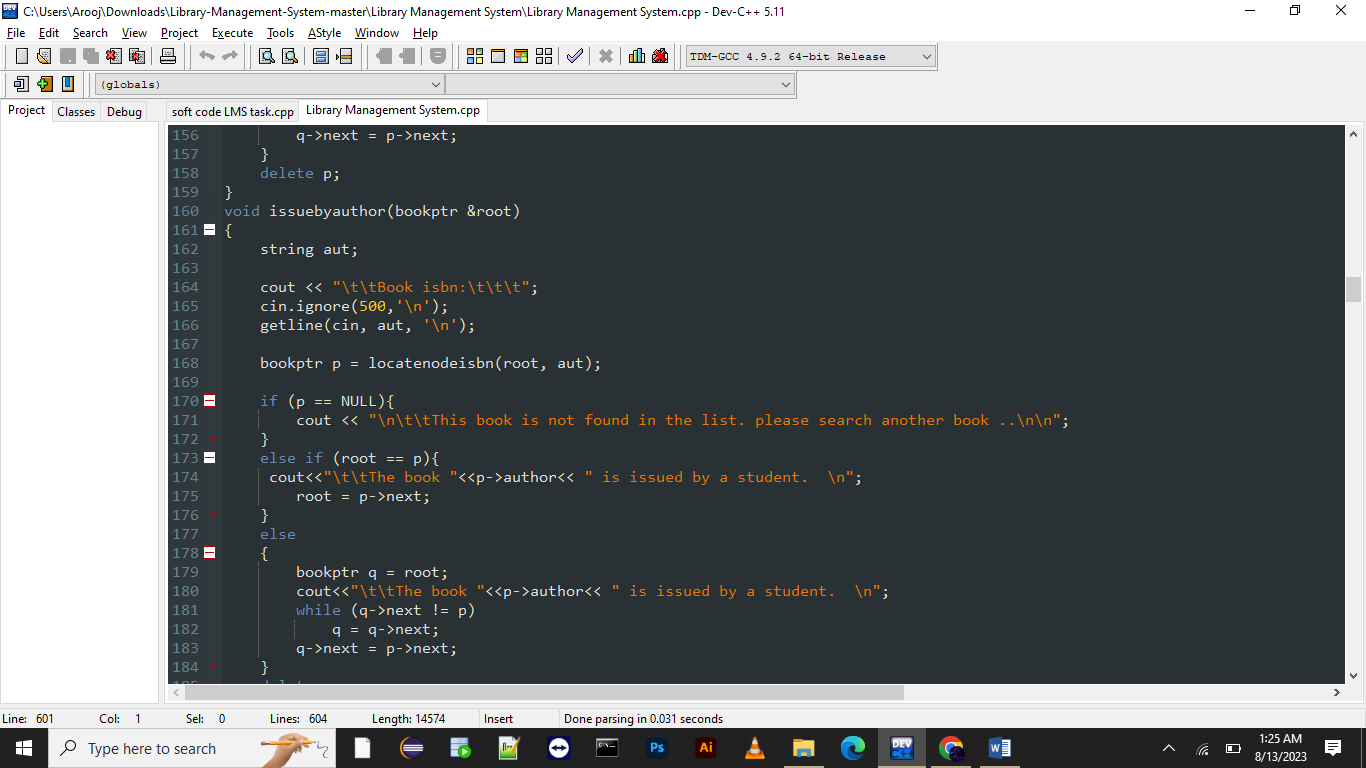
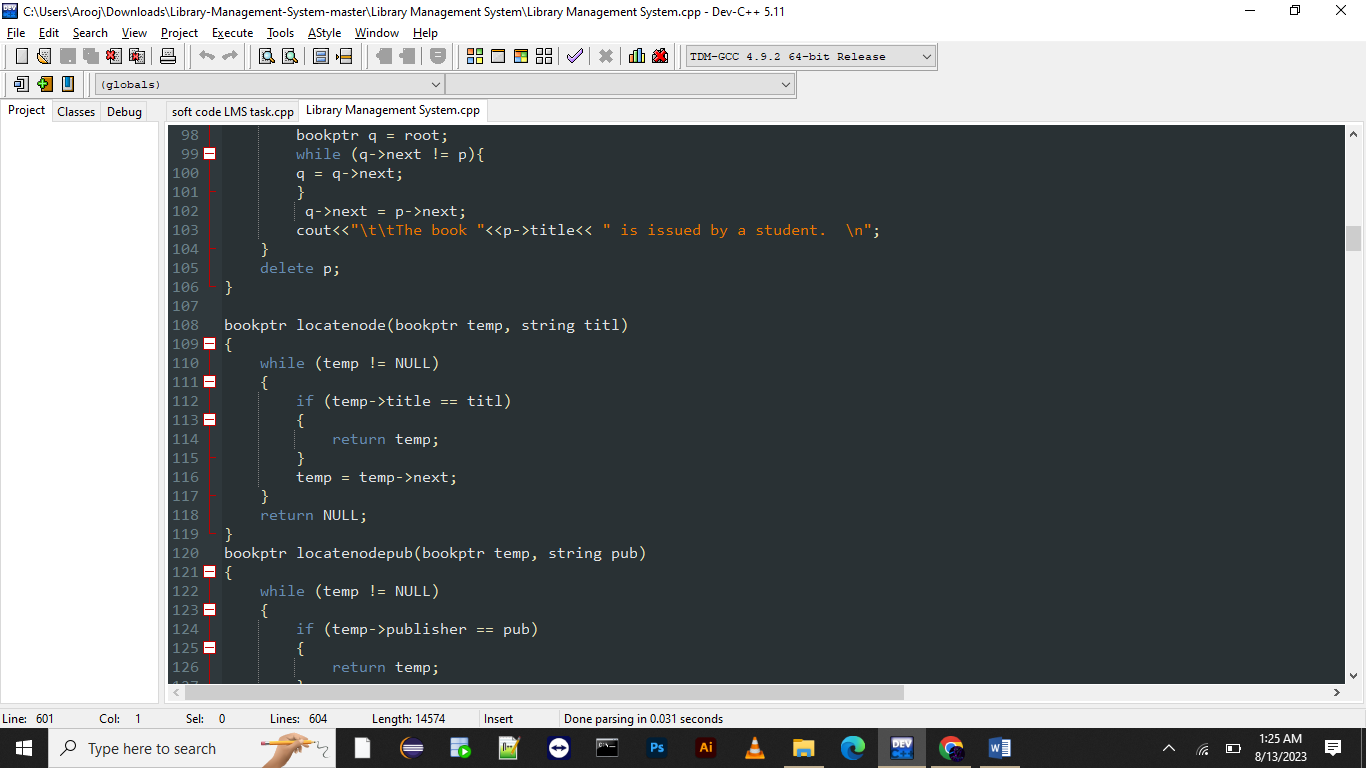
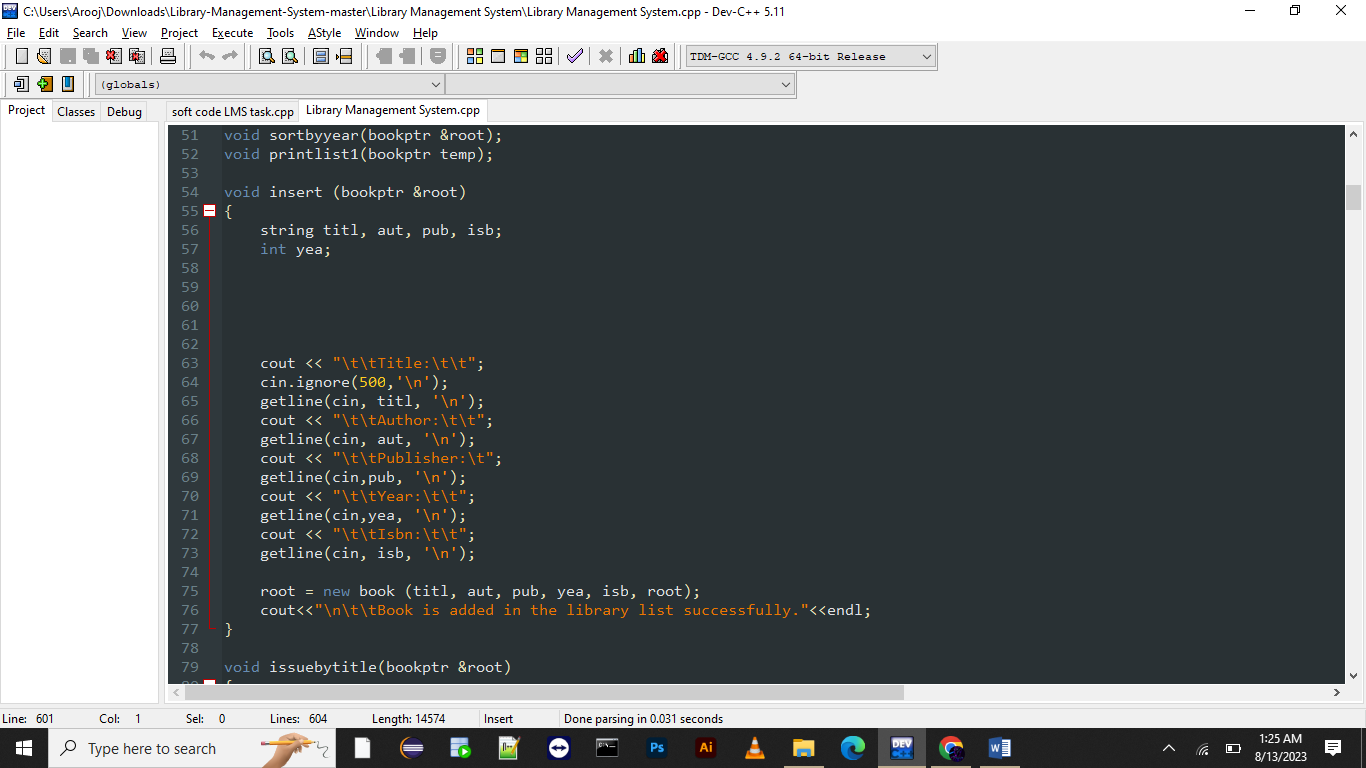
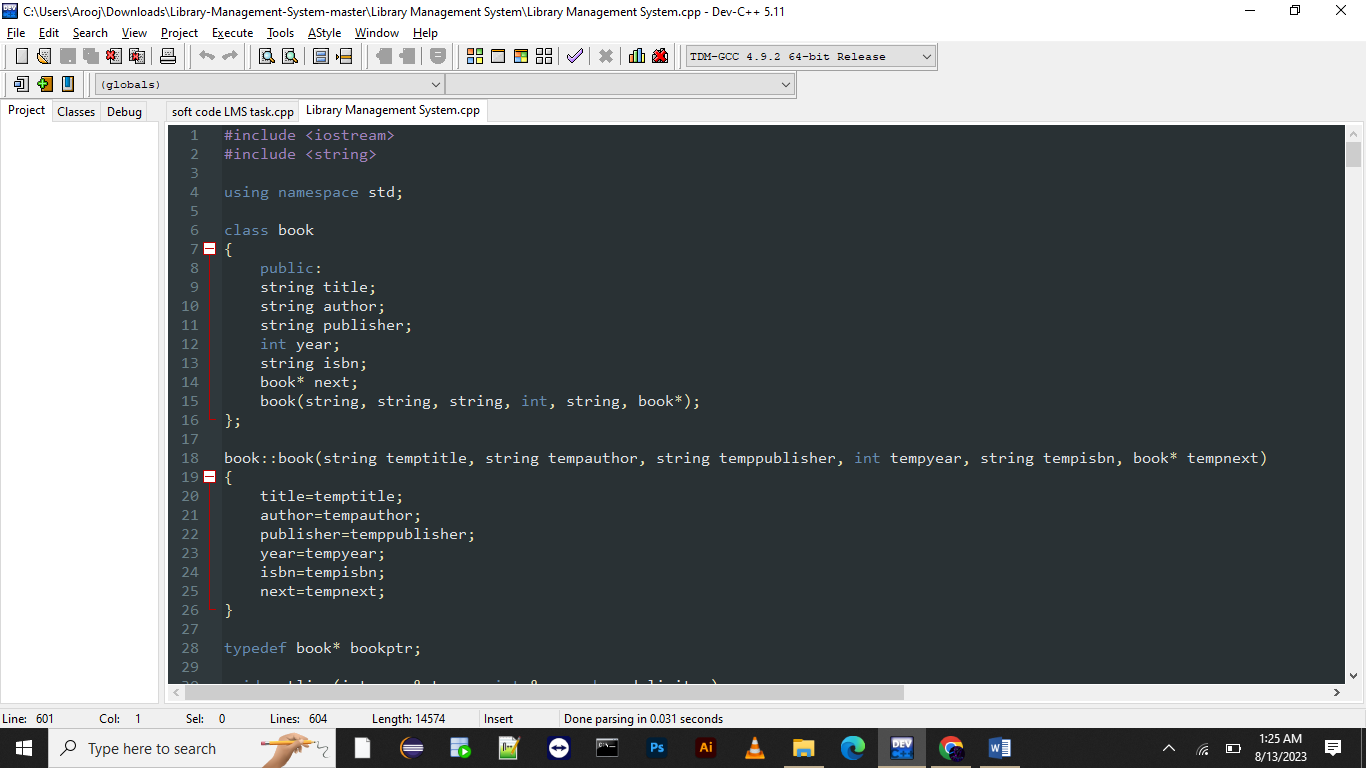
}

**//Created by Maryam Siddiqui**

**//For Internship Task To-do manager**

**//For @softcode**

**Output:**

****