**Library Management System**

**SHANZAY ALI KHAN**

**CODE:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include<stdbool.h>

struct Book {

int bookId;

char title[100];

char author[100];

int available;

};

void displayMenu() {

printf("\nLibrary Management System\n");

printf("1. Add Book\n");

printf("2. Display Books\n");

printf("3. Issue Book\n");

printf("4. Search Book\n");

printf("5. Return Book\n");

printf("6. Update Book Name\n");

printf("7. Delete Book\n");

printf("8. Exit\n");

printf("Enter your choice: ");

}

void addBook(struct Book\* library, int\* bookCount) {

if (\*bookCount >= 5) {

printf("Maximum limit of books reached. Cannot add more books.\n");

return;

}

printf("Enter Book Title: ");

scanf("%s", library[\*bookCount].title);

printf("Enter Author Name: ");

scanf("%s", library[\*bookCount].author);

library[\*bookCount].bookId = \*bookCount + 1;

library[\*bookCount].available = 1;

(\*bookCount)++;

printf("Book added successfully.\n");

// Write the updated library to the file

FILE\* file = fopen("library.txt", "a");

if (file != NULL) {

fprintf(file, "%d %s %s %d\n", library[\*bookCount - 1].bookId, library[\*bookCount - 1].title,

library[\*bookCount - 1].author, library[\*bookCount - 1].available);

fclose(file);

}

else {

printf("Error writing to file.\n");

}

}

void displayBooks(const struct Book \*library, int bookCount) {

printf("\nLibrary Books:\n");

printf("%-5s %-20s %-20s %s\n", "ID", "Title", "Author", "Availability");

for (int i = 0; i < bookCount; i++) {

printf("%-5d %-20s %-20s %s\n", library[i].bookId, library[i].title,

library[i].author, library[i].available ? "Available" : "Not Available");

}

}

void issueBook(struct Book \*library, int bookCount) {

int bookId;

printf("Enter the Book ID to issue: ");

scanf("%d", &bookId);

if (bookId >= 1 && bookId <= bookCount) {

if (library[bookId - 1].available) {

library[bookId - 1].available = 0;

printf("Book issued successfully.\n");

} else {

printf("Book is not available for issuance.\n");

}

} else {

printf("Invalid Book ID.\n");

}

}

void searchBook(const struct Book \*library, int bookCount) {

char searchQuery[100];

printf("Enter the book title or author to search: ");

scanf("%s", searchQuery);

bool found = false;

printf("\nSearch Results:\n");

printf("%-5s %-20s %-20s %s\n", "ID", "Title", "Author", "Availability");

for (int i = 0; i < bookCount; i++) {

// Check if the search matches the title or author (case-insensitive)

if (strstr(strupr(library[i].title), strupr(searchQuery)) != NULL ||

strstr(strupr(library[i].author), strupr(searchQuery)) != NULL) {

printf("%-5d %-20s %-20s %s\n", library[i].bookId, library[i].title,

library[i].author, library[i].available ? "Available" : "Not Available");

found = true;

}

}

if (!found) {

printf("No matching books found.\n");

}

}

void returnBook(struct Book \*library, int bookCount) {

int bookId;

printf("Enter the Book ID to return: ");

scanf("%d", &bookId);

if (bookId >= 1 && bookId <= bookCount) {

if (!library[bookId - 1].available) {

library[bookId - 1].available = 1;

printf("Book returned successfully.\n");

} else {

printf("Book is already available.\n");

}

} else {

printf("Invalid Book ID.\n");

}

}

void updateBook(struct Book \*library, int bookCount) {

int bookId;

printf("Enter the Book ID to update: ");

scanf("%d", &bookId);

if (bookId >= 1 && bookId <= bookCount) {

printf("Enter new title: ");

scanf("%s", library[bookId - 1].title);

printf("Enter new author: ");

scanf("%s", library[bookId - 1].author);

printf("Book updated successfully.\n");

// Update the file with the modified library

FILE \*file = fopen("library.txt", "w");

if (file != NULL) {

for (int i = 0; i < bookCount; i++) {

fprintf(file, "%d %s %s %d\n", library[i].bookId, library[i].title,

library[i].author, library[i].available);

}

fclose(file);

} else {

printf("Error writing to file.\n");

}

} else {

printf("Invalid Book ID.\n");

}

}

void deleteBook(struct Book \*library, int \*bookCount) {

int bookId;

printf("Enter the Book ID to delete: ");

scanf("%d", &bookId);

if (bookId >= 1 && bookId <= \*bookCount) {

// Shift the array elements to overwrite the deleted book

for (int i = bookId - 1; i < \*bookCount - 1; i++) {

library[i] = library[i + 1];

}

(\*bookCount)--;

printf("Book deleted successfully.\n");

// Update the file with the modified library

FILE \*file = fopen("library.txt", "w");

if (file != NULL) {

for (int i = 0; i < \*bookCount; i++) {

fprintf(file, "%d %s %s %d\n", library[i].bookId, library[i].title,

library[i].author, library[i].available);

}

fclose(file);

} else {

printf("Error writing to file.\n");

}

} else {

printf("Invalid Book ID.\n");

}

}

void loadLibraryFromFile(struct Book \*library, int \*bookCount) {

FILE \*file = fopen("library.txt", "r");

if (file != NULL) {

while (fscanf(file, "%d %s %s %d", &library[\*bookCount].bookId, library[\*bookCount].title,

library[\*bookCount].author, &library[\*bookCount].available) != EOF) {

(\*bookCount)++;

}

fclose(file);

}

}

int main() {

struct Book library[5];

int bookCount = 0;

int choice;

// Load existing library data from file

loadLibraryFromFile(library, &bookCount);

do {

displayMenu();

scanf("%d", &choice);

switch (choice) {

case 1:

addBook(library, &bookCount);

break;

case 2:

displayBooks(library, bookCount);

break;

case 3:

issueBook(library, bookCount);

break;

case 4:

searchBook(library,bookCount);

break;

case 5:

returnBook(library, bookCount);

break;

case 6:

updateBook(library, bookCount);

break;

case 7:

deleteBook(library, &bookCount);

break;

case 8:

printf("Exiting the program. Goodbye!\n");

break;

default:

printf("Invalid choice. Please enter a valid option.\n");

}

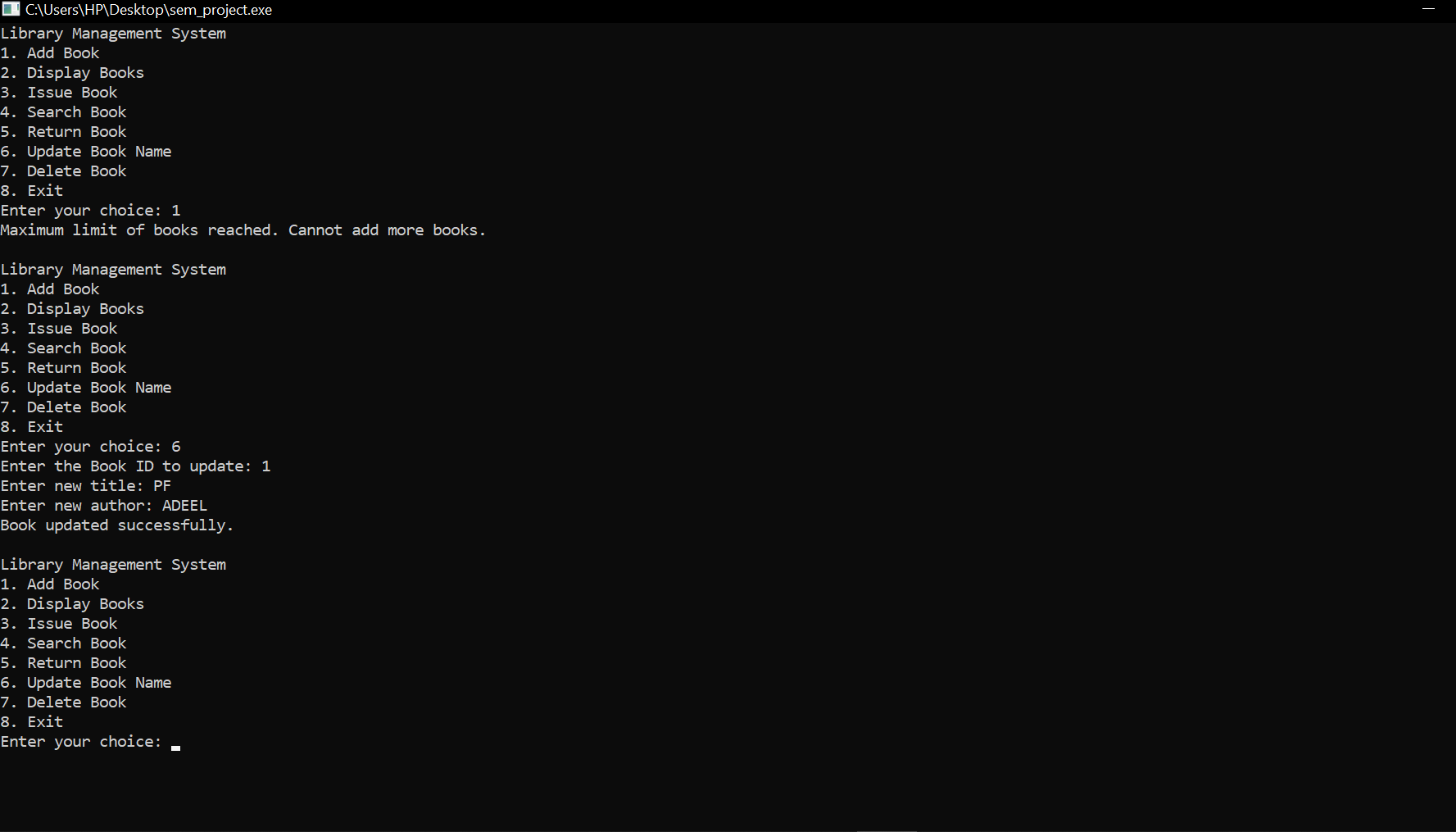
} while (choice != 8);

return 0;

}

**OUTPUTS**

OUTPUTS FOR ADDING AND UPDATING BOOKS:

****

OUTPUTS FOR DISPLAYING BOOKS:

