**PROGRAM CODING:**

#include <iostream>

#include <fstream>

#include <vector>

#include <string>

#include <algorithm>

using namespace std;

class Book {

public:

int id;

string title;

string author;

bool isIssued;

Book() : id(0), title(""), author(""), isIssued(false) {}

Book(int id, const string& title, const string& author, bool isIssued = false)

: id(id), title(title), author(author), isIssued(isIssued) {}

void display() const {

cout << "ID: " << id << " | Title: " << title

<< " | Author: " << author

<< " | Status: " << (isIssued ? "Issued" : "Available") << endl;

}

string serialize() const {

return to\_string(id) + "," + title + "," + author + "," + (isIssued ? "1" : "0");

}

static Book deserialize(const string& line) {

Book b;

size\_t pos = 0, prev = 0;

vector<string> tokens;

while ((pos = line.find(',', prev)) != string::npos) {

tokens.push\_back(line.substr(prev, pos - prev));

prev = pos + 1;

}

tokens.push\_back(line.substr(prev));

if (tokens.size() == 4) {

b.id = stoi(tokens[0]);

b.title = tokens[1];

b.author = tokens[2];

b.isIssued = (tokens[3] == "1");

}

return b;

}

};

class Library {

private:

vector<Book> books;

const string filename = "library\_data.txt";

void loadFromFile() {

ifstream infile(filename);

string line;

while (getline(infile, line)) {

books.push\_back(Book::deserialize(line));

}

infile.close();

}

void saveToFile() const {

ofstream outfile(filename);

for (const auto& book : books) {

outfile << book.serialize() << endl;

}

outfile.close();

}

public:

Library() {

loadFromFile();

}

~Library() {

saveToFile();

}

void addBook() {

int id;

string title, author;

cout << "Enter Book ID: ";

cin >> id;

cin.ignore();

cout << "Enter Title: ";

getline(cin, title);

cout << "Enter Author: ";

getline(cin, author);

books.emplace\_back(id, title, author);

cout << "Book added successfully.\n";

}

void issueBook() {

int id;

cout << "Enter Book ID to issue: ";

cin >> id;

for (auto& book : books) {

if (book.id == id) {

if (!book.isIssued) {

book.isIssued = true;

cout << "Book issued successfully.\n";

} else {

cout << "Book is already issued.\n";

}

return;

}

}

cout << "Book not found.\n";

}

void returnBook() {

int id;

cout << "Enter Book ID to return: ";

cin >> id;

for (auto& book : books) {

if (book.id == id) {

if (book.isIssued) {

book.isIssued = false;

cout << "Book returned successfully.\n";

} else {

cout << "Book was not issued.\n";

}

return;

}

}

cout << "Book not found.\n";

}

void searchBook() const {

int choice;

cout << "Search by:\n1. ID\n2. Title\nEnter choice: ";

cin >> choice;

cin.ignore();

if (choice == 1) {

int id;

cout << "Enter Book ID: ";

cin >> id;

for (const auto& book : books) {

if (book.id == id) {

book.display();

return;

}

}

cout << "Book not found.\n";

} else if (choice == 2) {

string title;

cout << "Enter Title: ";

getline(cin, title);

for (const auto& book : books) {

if (book.title == title) {

book.display();

return;

}

}

cout << "Book not found.\n";

} else {

cout << "Invalid choice.\n";

}

}

void displayStatistics() const {

int total = books.size();

int issued = count\_if(books.begin(), books.end(), [](const Book& b) { return b.isIssued; });

int available = total - issued;

cout << "Total Books: " << total << "\n";

cout << "Issued Books: " << issued << "\n";

cout << "Available Books: " << available << "\n";

}

void displayAllBooks() const {

for (const auto& book : books) {

book.display();

}

}

};

int main() {

Library lib;

int choice;

do {

cout << "\nLibrary Management System\n";

cout << "1. Add Book\n";

cout << "2. Issue Book\n";

cout << "3. Return Book\n";

cout << "4. Search Book\n";

cout << "5. Display Statistics\n";

cout << "6. Display All Books\n";

cout << "7. Exit\n";

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1:

lib.addBook();

break;

case 2:

lib.issueBook();

break;

case 3:

lib.returnBook();

break;

case 4:

lib.searchBook();

break;

case 5:

lib.displayStatistics();

break;

case 6:

lib.displayAllBooks();

break;

case 7:

cout << "Exiting the program.\n";

break;

default:

cout << "Invalid choice. Try again.\n";

}

} while (choice != 7);

return 0;

}