#include<iostream>

#include<vector>

#include<fstream>

#include<string>

#include<sstream>

using namespace std;

// global variables

int b, c, i = 0;

string a;

string username, password;

//structure

struct book

{

string id;

string title;

string author;

bool isborrowed = false;

};

//vector

vector<book> library;

book d;

//function declaration

void main\_menu();

void loadbooksfromfile();

void login\_path();

void librarian\_login();

void addbook();

void display\_all();

void search();

void borrow\_book();

void return\_book();

void view\_borrow();

void delete\_book();

void student\_login();

//main function

int main()

{

loadbooksfromfile();

main\_menu();

}

//user-defined function

void loadbooksfromfile()

{

ifstream read("books.txt");

if (!read)

{

cout << "Error! can't open file.";

return;

}

string line;

while (getline(read, line))

{

stringstream ss(line);

string id, title, author, isborrowed;

getline(ss, id, ',');

getline(ss, title, ',');

getline(ss, author, ',');

getline(ss, isborrowed, ',');

book b;

b.id = stoi(id);

b.title = title;

b.author = author;

b.isborrowed = (isborrowed == "1");

library.push\_back(b);

}

read.close();

}

//user\_defined function

void main\_menu()

{

cout << endl << "\t\t\t\t\*Library Management System\*" << endl << endl;

cout << "\t\t\t\t\t\t1. Librarian." << endl;

cout << "\t\t\t\t\t\t2. Student." << endl;

cout << "\t\t\t\t\t\t3. Exit." << endl;

cout << "\t\t\t\t\t\tChoice: ";

cin >> a;

if (a == "1")

{

login\_path();

}

else if (a == "2")

{

student\_login();

}

else if (a == "3")

{

exit(0);

}

else

{

cout << "Error! You enter invalid input.";

main\_menu();

}

}

//user\_defined function

void login\_path()

{

cout << endl << "\t\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* login \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl << endl;

cout << "\t\t\t\t\t\tUsername: ";

cin >> username;

cout << "\t\t\t\t\t\tPassword: ";

cin >> password;

if (username == "admin" && password == "admin123")

{

librarian\_login();

}

else if (username != "admin" || password != "admin123")

{

cout << "\t\t\t\t\t\tWrong credentials!." << endl;

main\_menu();

}

}

//user\_defined function

void librarian\_login()

{

cout << endl << "\t\t\t\t\*Librarian Login\*" << endl << endl;

cout << "\t\t\t\t\t\t1. Add Book." << endl;

cout << "\t\t\t\t\t\t2. Display All." << endl;

cout << "\t\t\t\t\t\t3. Search." << endl;

cout << "\t\t\t\t\t\t4. Borrow." << endl;

cout << "\t\t\t\t\t\t5. Return." << endl;

cout << "\t\t\t\t\t\t6. View Borrow." << endl;

cout << "\t\t\t\t\t\t7. Delete." << endl;

cout << "\t\t\t\t\t\t8. Exit." << endl;

cout << "\t\t\t\t\t\tChoice: ";

cin >> b;

switch (b)

{

case 1:

addbook();

break;

case 2:

i = 1;

display\_all();

break;

case 3:

i = 1;

search();

break;

case 4:

borrow\_book();

break;

case 5:

return\_book();

break;

case 6:

view\_borrow();

break;

case 7:

delete\_book();

break;

case 8:

main\_menu();

break;

default:

cout << "Error! you enter invalid input.";

}

}

//user\_defined function

void addbook()

{

cout << "\t\t\t\t\t\tID: ";

cin >> d.id;

cin.ignore();

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

getline(cin, d.title);

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

getline(cin, d.author);

d.isborrowed = false;

library.push\_back(d);

ofstream write("books.txt", ios::app);

if (!write)

{

cout << "Error! file not found.";

return;

}

write << d.id << "," << d.title << "," << d.author << "," << d.isborrowed << endl;

write.close();

cout << "\t\t\t\t\t\tBook added successfully! " << endl;

librarian\_login();

}

//user\_defined function

void display\_all()

{

ifstream read("books.txt", ios::app);

if (!read)

{

cout << "\t\t\t\t\t\tError! File is not opening";

return;

}

string line;

while (getline(read, line))

{

stringstream ss(line);

string id, title, author, isborrowed;

getline(ss, id, ',');

getline(ss, title, ',');

getline(ss, author, ',');

getline(ss, isborrowed, ',');

cout << "\t\t\t\t\t\tID : " << id << " , " << "Title : " << title << " , " << "Author : " << author << " , " << "Status : " << (isborrowed == "1" ? "Borrowed" : "Available") << endl;

}

read.close();

if (i == 1)

{

librarian\_login();

}

else if (i == 2)

{

student\_login();

}

}

//user\_defined function

void search()

{

int a;

string c;

cout << "\t\t\t\t\t\tPress 1. search by ID. " << endl << "\t\t\t\t\t\t Press 2. search by Title.";

cin >> a;

if (a == 1)

{

cout << "\t\t\t\t\t\t Enter ID : ";

cin >> c;

cin.ignore();

}

else

{

cout << "\t\t\t\t\t\tEnter Title : ";

getline(cin, c);

}

ifstream read("books.txt");

if (!read)

{

cout << "Error! File is not opening.";

return;

}

string line;

while (getline(read, line))

{

stringstream ss(line);

string id, title, author, isborrowed;

getline(ss, id, ',');

getline(ss, title, ',');

getline(ss, author, ',');

getline(ss, isborrowed, ',');

if (id == c || title == c)

{

cout << "\t\t\t\t\t\tID : " << id << " , " << "Title : " << title << " , " << "Author : " << author << " , " << "Status : " << (isborrowed == "1" ? "Borrowed" : "Available") << endl;

}

}

read.close();

if (i == 1)

{

librarian\_login();

}

else if (i == 2)

{

student\_login();

}

}

//user\_defined function

void borrow\_book()

{

string choice, input;

cout << "\t\t\t\t\t\tPress 1. ID" << endl;

cout << "\t\t\t\t\t\tPress 2. Title" << endl;

cout << "\t\t\t\t\t\tChoice: ";

cin.ignore();

getline(cin, choice);

if (choice == "1") {

cout << "\t\t\t\t\t\tEnter ID : ";

getline(cin, input);

}

else if (choice == "2") {

cout << "\t\t\t\t\t\tEnter Title : ";

getline(cin, input);

}

else {

cout << "\t\t\t\t\t\tError! Invalid input." << endl;

librarian\_login();

return;

}

// Read the entire file and prepare updated content

ifstream read("books.txt");

if (!read) {

cout << "\t\t\t\t\t\tError! File not opening." << endl;

return;

}

vector<string> updatedBooks;

bool found = false;

string line;

while (getline(read, line)) {

stringstream ss(line);

string id, title, author, isborrowed;

getline(ss, id, ',');

getline(ss, title, ',');

getline(ss, author, ',');

getline(ss, isborrowed, ',');

if ((choice == "1" && id == input) || (choice == "2" && title == input)) {

found = true;

if (isborrowed == "1") {

cout << "\t\t\t\t\t\tBook is already borrowed." << endl;

}

else {

isborrowed = "1";

cout << "\t\t\t\t\t\tBook borrowed successfully!" << endl;

}

}

updatedBooks.push\_back(id + "," + title + "," + author + "," + isborrowed);

}

read.close();

// Rewriting the entire file

ofstream write("books.txt", ios::trunc); // <---- VERY IMPORTANT: ios::trunc clears the file before writing

if (!write) {

cout << "Error! Could not open file for writing." << endl;

return;

}

for (const string& record : updatedBooks) {

write << record << "\n";

}

write.close();

librarian\_login();

}

//user\_defined function

void return\_book()

{

string choice, input;

cout << "\t\t\t\t\t\tPress 1. ID" << endl;

cout << "\t\t\t\t\t\tPress 2. Title" << endl;

cout << "\t\t\t\t\t\tChoice: ";

cin.ignore();

getline(cin, choice);

if (choice == "1") {

cout << "\t\t\t\t\t\tEnter ID : ";

getline(cin, input);

}

else if (choice == "2") {

cout << "\t\t\t\t\t\tEnter Title : ";

getline(cin, input);

}

else {

cout << "\t\t\t\t\t\tError! Invalid input." << endl;

librarian\_login();

return;

}

// Read the entire file and prepare updated content

ifstream read("books.txt");

if (!read) {

cout << "\t\t\t\t\t\tError! File not opening." << endl;

return;

}

vector<string> updatedBooks;

bool found = false;

string line;

while (getline(read, line)) {

stringstream ss(line);

string id, title, author, isborrowed;

getline(ss, id, ',');

getline(ss, title, ',');

getline(ss, author, ',');

getline(ss, isborrowed, ',');

if ((choice == "1" && id == input) || (choice == "2" && title == input)) {

found = true;

if (isborrowed == "0") {

cout << "\t\t\t\t\t\tBook is not borrowed.It is available." << endl;

}

else {

isborrowed = "0";

cout << "\t\t\t\t\t\tStatus updated successfully!" << endl;

}

}

updatedBooks.push\_back(id + "," + title + "," + author + "," + isborrowed);

}

read.close();

// Rewriting the entire file

ofstream write("books.txt", ios::trunc); // <---- VERY IMPORTANT: ios::trunc clears the file before writing

if (!write) {

cout << "Error! Could not open file for writing." << endl;

return;

}

for (const string& record : updatedBooks) {

write << record << "\n";

}

write.close();

librarian\_login();

}

//user\_defined function

void view\_borrow()

{

ifstream read("books.txt");

if (!read)

{

cout << "\t\t\t\t\t\tError! File is not opening.";

return;

}

string line;

while (getline(read, line))

{

stringstream ss(line);

string id, title, author, isborrowed;

getline(ss, id, ',');

getline(ss, title, ',');

getline(ss, author, ',');

getline(ss, isborrowed, ',');

if (isborrowed == "1")

{

cout << "\t\t\t\t\t\tID : " << id << " , " << "Title : " << title << " , " << "Author : " << author << " , " << "Status : Borrowed" << endl;

}

}

read.close();

librarian\_login();

}

//user\_defined function

void delete\_book()

{

string choice, input;

cout << "\t\t\t\t\t\tPress 1. ID" << endl;

cout << "\t\t\t\t\t\tPress 2. Title" << endl;

cout << "\t\t\t\t\t\tChoice: ";

cin.ignore();

getline(cin, choice);

if (choice == "1") {

cout << "\t\t\t\t\t\tEnter ID : ";

getline(cin, input);

}

else if (choice == "2") {

cout << "\t\t\t\t\t\tEnter Title : ";

getline(cin, input);

}

else {

cout << "\t\t\t\t\t\tError! Invalid input." << endl;

librarian\_login();

return;

}

// Read the entire file and prepare updated content

ifstream read("books.txt");

if (!read) {

cout << "\t\t\t\t\t\tError! File not opening." << endl;

return;

}

vector<string> updatedBooks;

string line;

while (getline(read, line)) {

stringstream ss(line);

string id, title, author, isborrowed;

getline(ss, id, ',');

getline(ss, title, ',');

getline(ss, author, ',');

getline(ss, isborrowed, ',');

int found = 1;

if ((choice == "1" && id == input) || (choice == "2" && title == input))

{

cout << "\t\t\t\t\t\tBook deleted successfully!";

found = 0;

}

if (found == 1)

{

updatedBooks.push\_back(id + "," + title + "," + author + "," + isborrowed);

}

}

read.close();

// Rewriting the entire file

ofstream write("books.txt", ios::trunc); // <---- VERY IMPORTANT: ios::trunc clears the file before writing

if (!write) {

cout << "Error! Could not open file for writing." << endl;

return;

}

for (const string& record : updatedBooks) {

write << record << "\n";

}

write.close();

librarian\_login();

}

//user\_defined function

void student\_login()

{

cout << endl << "\t\t\t\t\*\*Student Login" << endl << endl;

cout << "\t\t\t\t\t\t1. search." << endl;

cout << "\t\t\t\t\t\t2. Display All." << endl;

cout << "\t\t\t\t\t\t3. Exit." << endl;

cout << "\t\t\t\t\t\tChoice: ";

cin >> c;

switch (c)

{

case 1:

i = 2;

search();

break;

case 2:

i = 2;

display\_all();

break;

case 3:

main\_menu();

break;

default:

cout << "Error! You enter invalid input.";

}

}