Library Management System

Code:

// In the name of Allah, the most gracious, the most merciful

#include <bits/stdc++.h>

#include <conio.h>

using namespace std;

// ANSI escape codes for text color

#define RESET "\033[0m"

#define RED "\033[31m"

#define GREEN "\033[32m"

#define YELLOW "\033[33m"

#define BLUE "\033[34m"

#define MAGENTA "\033[35m"

#define CYAN "\033[36m"

#define WHITE "\033[37m"

// login status

bool isUserLoged = false;

bool isAdminLoged = false;

// user class

class User

{

public:

    string name;

    int id;

    string email;

    string password;

    User() : name(""), id(0), email(""), password("") {}

    User(const string &n, int i, const string &e, const string &p)

        : name(n), id(i), email(e), password(p) {}

};

// Function to save users to a file

void saveUser(const User &user)

{

    ofstream file("users.csv", ios::app);

    file << user.id << "," << user.name << "," << user.email << "," << user.password << "\n";

    file.close();

}

// load user

map<int, User> loadUsers()

{

    map<int, User> users;

    ifstream file("users.csv");

    string name, email, password;

    int id;

    string temp;

    while (getline(file, temp, ',') && getline(file, name, ',') && getline(file, email, ',') && getline(file, password))

    {

        try

        {

            id = stoi(temp);

            users[id] = User(name, id, email, password);

        }

        catch (const std::invalid\_argument &e)

        {

            // Skip this line if the ID cannot be converted to an integer

            continue;

        }

    }

    file.close();

    return users;

}

// books class

class Book

{

public:

    string name, isbn, category, publication, author;

    int stock;

    // Constructor

    Book(const string &n, const string &i, const string &c,

         const string &p, const string &a, int s)

        : name(n), isbn(i), category(c), publication(p), author(a), stock(s) {}

};

// log in function

pair<int, string> logIn()

{

    pair<int, string> loginData;

    cout << "Fill the form to get library card :) \n";

    int id;

    cout << "id: ";

    cin >> id;

    char ch = ' ';

    string password;

    cout << "Password: ";

    while (ch != 13)

    {

        ch = getch();

        if (ch == 8 || ch == 13 || ch == 32)

            continue;

        else

        {

            password += ch;

            cout << "\*";

        }

    }

    loginData.first = id;

    loginData.second = password;

    return loginData;

}

// exit function

void exitProgram()

{

    return;

}

// main class

class LMS

{

public:

    vector<User> users;

    vector<Book> books;

    // load books from file

    void loadBooks()

    {

        ifstream inFile("books.csv");

        if (inFile.is\_open())

        {

            string line;

            while (getline(inFile, line))

            {

                stringstream ss(line);

                string name, isbn, category, publication, author;

                int stock;

                getline(ss, name, ',');

                getline(ss, isbn, ',');

                getline(ss, category, ',');

                getline(ss, publication, ',');

                getline(ss, author, ',');

                ss >> stock;

                Book newBook(name, isbn, category, publication, author, stock);

                books.push\_back(newBook);

            }

            inFile.close();

        }

        else

        {

            cout << "Unable to open the file for loading books.\n";

        }

    }

    // Function to save user data to a CSV file

    void saveUserToCSV(const User &user)

    {

        ofstream file("users.csv", ios::app); // Open the file in append mode

        if (file.is\_open())

        {

            // Write user data to the file in CSV format

            file << user.name << "," << user.id << "," << user.email << "," << user.password << "\n";

            cout << "User data saved to library database.\n";

            file.close(); // Close the file

        }

        else

        {

            cout << RED << "Error opening the CSV file.\n"

                 << RESET;

        }

    }

    // sign up function

    void signup()

    {

        cout << "Fill the form to get library card :) \n";

        cout << "Name: ";

        string name;

        cin.ignore();

        getline(cin, name);

        int id;

        cout << "id: ";

        cin >> id;

        cout << "email: ";

        string email;

        cin.ignore();

        getline(cin, email);

        char ch = ' ';

        string password;

        cout << "You don't have permit to backspace\nPassword: ";

        while (ch != 13)

        {

            ch = getch();

            if (ch == 8 || ch == 13 || ch == 32)

                continue;

            else

            {

                password += ch;

                cout << "\*";

            }

        }

        // admin

        if (id == 223311161 && password == "admin")

        {

            User adminUser(name, id, email, password);

            saveUser(adminUser);

            cout << "\nCongrats! Admin user added successfully.\n";

        }

        // normal people

        else

        {

            // Create a new user

            User newUser(name, id, email, password);

            saveUser(newUser);

            cout << "\nCongrats! You have successfully created an account and got a library card :)\n";

        }

    }

    // show all books

    void showBooks()

    {

        ifstream inFile("books.csv");

        if (inFile.is\_open())

        {

            cout << "Book Names:\n";

            string line;

            getline(inFile, line);

            int serial = 1;

            while (getline(inFile, line))

            {

                stringstream ss(line);

                string name;

                getline(ss, name, ',');

                cout << serial << ". " << name << "\n";

                serial++;

            }

            inFile.close();

        }

        else

        {

            cout << "Unable to open the file for reading books.\n";

        }

    }

    // book categories

    void showCategory()

    {

        ifstream inFile("books.csv");

        if (inFile.is\_open())

        {

            set<string> categories;

            string line;

            getline(inFile, line);

            while (getline(inFile, line))

            {

                stringstream ss(line);

                string name, isbn, category;

                getline(ss, name, ',');

                getline(ss, isbn, ',');

                getline(ss, category, ',');

                categories.insert(category);

            }

            inFile.close();

            int list = 1;

            cout << "Book Categories:\n";

            for (const string &category : categories)

            {

                cout << list << ". " << category << "\n";

                list++;

            }

        }

        else

        {

            cout << "Unable to open the file for reading books.\n";

        }

    }

    // show all publications

    void showPublications()

    {

        ifstream inFile("books.csv");

        if (inFile.is\_open())

        {

            set<string> publications;

            string line;

            getline(inFile, line);

            while (getline(inFile, line))

            {

                stringstream ss(line);

                string name, isbn, category, publication;

                getline(ss, name, ',');

                getline(ss, isbn, ',');

                getline(ss, category, ',');

                getline(ss, publication, ',');

                publications.insert(publication);

            }

            inFile.close();

            cout << "Book Publications:\n";

            int listCount = 1;

            for (const string &publication : publications)

            {

                cout << listCount << ". " << publication << "\n";

                listCount++;

            }

        }

        else

        {

            cout << "Unable to open the file for reading books.\n";

        }

    }

    // add book (admin)

    void addBook()

    {

        cout << "Enter book details:\n";

        cout << "Name: ";

        string name;

        cin.ignore();

        getline(cin, name);

        cout << "ISBN: ";

        string isbn;

        char isbnc;

        cin >> isbnc;

        getline(cin, isbn);

        isbn = isbnc + isbn;

        cout << "Category: ";

        string category;

        char categoryC;

        cin >> categoryC;

        getline(cin, category);

        category = categoryC + category;

        cout << "Publication: ";

        string publication;

        char pub;

        cin >> pub;

        getline(cin, publication);

        publication = pub + publication;

        cout << "Author: ";

        string author;

        char auth;

        cin >> auth;

        getline(cin, author);

        author = auth + author;

        cout << "Stock: ";

        int stock;

        cin >> stock;

        // Create a new Book object

        Book newBook(name, isbn, category, publication, author, stock);

        // Add the new book to the vector

        books.push\_back(newBook);

        // Save the new book to the file in append mode

        ofstream outFile("books.csv", ios::app);

        if (outFile.is\_open())

        {

            outFile << newBook.name << "," << newBook.isbn << "," << newBook.category << ","

                    << newBook.publication << "," << newBook.author << "," << newBook.stock << "\n";

            outFile.close();

            cout << "Book added and saved to Library.\n";

        }

        else

        {

            cout << "Unable to open the file for saving books.\n";

        }

    }

    // search isbn or name

    void searchBook()

    {

        cout << "\nenter isbn or name: ";

        string item;

        cin.ignore();

        getline(cin, item);

        ifstream inFile("books.csv");

        if (inFile.is\_open())

        {

            string line;

            getline(inFile, line);

            bool found = false;

            while (getline(inFile, line))

            {

                stringstream ss(line);

                string name, isbn, category, publication, author, stock;

                getline(ss, name, ',');

                getline(ss, isbn, ',');

                getline(ss, category, ',');

                getline(ss, publication, ',');

                getline(ss, author, ',');

                getline(ss, stock, ',');

                if (name == item || isbn == item)

                {

                    cout << GREEN << "Book found: \n"

                         << "Name : " << name << "\n"

                         << "ISBN: " << isbn << "\n"

                         << "Category: " << category << "\n"

                         << "Publication: " << publication << "\n"

                         << "Author: " << author << "\n"

                         << "Stock: " << stock << "\n"

                         << RESET;

                    found = true;

                    break;

                }

            }

            inFile.close();

            if (!found)

            {

                cout << "No book found.\n";

            }

        }

        else

        {

            cout << "Unable to open the file for reading books.\n";

        }

    }

    // about us

    void aboutUs()

    {

        cout << GREEN << "\n===========================\n          About Us\n===========================\n\nBurhan Uddin Ashik => 223311161\nSohag Mia => 223311162\nAfia Akter => 223311163\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\*    LMS Console Project\n\*      Developed by Varendra University       \n\*           31st Batch E Section Students \n\*           Subject: Software Engineering\n\*      Teacher: Delwar Hossain\n\*    Release: November 2023\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"

             << RESET;

    }

    // print all users

    void printAllUsers()

    {

        cout << "List of all users:\n";

        ifstream inFile("users.csv");

        if (inFile.is\_open())

        {

            string line;

            getline(inFile, line);

            int serial = 1;

            while (getline(inFile, line))

            {

                stringstream ss(line);

                string id, name;

                getline(ss, id, ',');

                getline(ss, name, ',');

                cout << serial << ". " << GREEN << name << RESET << "\n";

                serial++;

            }

            inFile.close();

        }

        else

        {

            cout << "Unable to open the file for reading usres.\n";

        }

    }

    // issue books

    void issueBooks()

    {

    }

    // stock out books

    void stockOut()

    {

        ifstream inFile("books.csv");

        if (inFile.is\_open())

        {

            cout << "Books with 0 stock:\n";

            string line;

            getline(inFile, line);

            while (getline(inFile, line))

            {

                stringstream ss(line);

                string name, isbn, category, publication, author;

                int stock;

                getline(ss, name, ',');

                getline(ss, isbn, ',');

                getline(ss, category, ',');

                getline(ss, publication, ',');

                getline(ss, author, ',');

                ss >> stock;

                if (stock == 0)

                {

                    cout << YELLOW << "\n"

                         << "Name : " << name << "\n"

                         << "ISBN: " << isbn << "\n"

                         << "Category: " << category << "\n"

                         << "Publication: " << publication << "\n"

                         << "Author: " << author << "\n"

                         << "Stock: " << stock << "\n"

                         << RESET;

                }

            }

            inFile.close();

        }

        else

        {

            cout << "Unable to open the file for reading books.\n";

        }

    }

    // admin menu

    int aMenuList()

    {

        cout << MAGENTA << "\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n      Admin Panel\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n"

             << RESET << "1.Add Books\n2.Show All Books\n3.Show All Users\n"

             << "4." << YELLOW << "Stock out books\n5."

             << RESET

             << CYAN << "About Us\n"

             << RESET

             << "6." << RED << "Exit\n\n";

        cout << RESET << "Enter your choice: ";

        int adminMenu;

        cin >> adminMenu;

        return adminMenu;

    }

    void adminMenu()

    {

        int adminMenu;

        do

        {

            adminMenu = aMenuList();

            switch (adminMenu)

            {

            case 1:

            {

                addBook();

                break;

            }

            case 2:

            {

                showBooks();

                break;

            }

            case 3:

            {

                printAllUsers();

                break;

            }

            case 4:

            {

                stockOut();

                break;

            }

            case 5:

            {

                aboutUs();

                break;

            }

            case 6:

            {

                exitProgram();

                break;

            }

            default:

                cout << RED << "invalid choice! please choose a valid choice :)\n"

                     << RESET;

            }

        } while (adminMenu != 6);

    }

};

// main function

void mainFunc(LMS lms, function<void(LMS lms)> logedUserMenu)

{

    int menu;

    do

    {

        cout << MAGENTA << "\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\nWelcome Big & Not Found Library :)\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n"

             << RESET << "1.SignUp\n2.LogIn\n3.Show All Books\n4.Show All Category\n5." << CYAN << "About Us\n"

             << RESET

             << "6." << RED << "Exit\n\n";

        cout << RESET << "Enter your choice: ";

        cin >> menu;

        switch (menu)

        {

        case 1:

        {

            lms.signup();

        }

        case 2:

        {

            map<int, User> users = loadUsers();

            pair<int, string> loginDAta = logIn();

            if (users.count(loginDAta.first) && users[loginDAta.first].password == loginDAta.second)

            {

                isUserLoged = true;

            }

            else if (loginDAta.first == 61 && loginDAta.second == "admin")

            {

                lms.adminMenu();

            }

            else

            {

                cout << RED << "\nLogin failed. Invalid ID or password.\nlogin again\n"

                     << RESET;

            }

            if (isUserLoged)

            {

                logedUserMenu(lms);

            }

            break;

        }

        case 3:

        {

            lms.loadBooks();

            lms.showBooks();

            break;

        }

        case 4:

        {

            cout << "\nall books in our Category: \n";

            lms.showCategory();

            break;

        }

        case 5:

        {

            lms.aboutUs();

        }

        case 6:

        {

            exitProgram();

            break;

        }

        default:

            cout << RED << "invalid choice! please choose a valid choice :)\n"

                 << RESET;

        }

    } while (menu != 6);

}

// sub main function

void logedUserMenu(LMS lms)

{

    cin.ignore(numeric\_limits<streamsize>::max(), '\n');

    int logedMenu;

    do

    {

        cout << MAGENTA << "\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\nWelcome Big & Not Found Library :)\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n"

             << RESET << "1.Show All Books\n2.Show All Category\n3.Show All Publications\n4.Search by isbn or name\n5." << CYAN << "About Us\n"

             << RESET

             << "6." << RED << "Exit\n\n"

             << RESET << "Enter your choice: ";

        cin >> logedMenu;

        switch (logedMenu)

        {

        case 1:

        {

            lms.showBooks();

            break;

        }

        case 2:

        {

            lms.showCategory();

            break;

        }

        case 3:

        {

            lms.showPublications();

            break;

        }

        case 4:

        {

            lms.searchBook();

            break;

        }

        case 5:

        {

            lms.aboutUs();

            break;

        }

        case 6:

        {

            exitProgram();

            break;

        }

        default:

            cout << RED << "invalid choice! please choose a valid choice :)\n"

                 << RESET;

        }

    } while (logedMenu != 6);

}

int main()

{

    LMS lms;

    mainFunc(lms, logedUserMenu);

    return 0;

}