**WRITEUP**

*Library Management System*

A Library Management System is a comprehensive software solution designed to streamline and automate the tasks associated with managing a library's resources, including books, students, and administrative functions. It provides an efficient way to organize, store, and retrieve information, making library operations more effective and user-friendly. The primary objective of an LMS is to enhance the overall management and accessibility of library resources.

Login System:

1. Users are prompted to enter a username and password.
2. User credentials are compared with data stored in the users.txt file.
3. If login is successful, users can access the main menu.

Main Menu:

1. Users have the option to navigate between the User Module and Book Module.
2. They can also choose to exit the program.

***User Module***:

Register User:

1. Users can register new accounts with a unique username and password.
2. User data is stored in the users.txt file.

Update User:

1. Users can update their passwords.
2. The users.txt file is updated with the new information.

Search User:

1. Users can search for a specific user by entering a username.
2. If found, the user's details are displayed.

Delete User:

1. Users can delete their accounts by entering their username.
2. The corresponding entry in the users.txt file is removed.

Show User List:

Displays a list of all registered users.

***Book Module:***

Add Book:

1. Users can add new books to the system.
2. Checks if the book ID already exists; if so, updates the quantity.

Book List:

Displays a list of all available books.

Remove Book:

1. Allows users to remove a book by entering the Book ID.
2. The book is removed from the system.

Update Book:

Users can update the details of a specific book using its ID.

Issue Book:

1. Users can issue a book to a student.
2. Updates the book quantity and records the issuance in issuedBooks.txt.

Display Issued Books:

Displays a list of books that have been issued to students.

Return Book:

1. Allows students to return a book.
2. Updates the book quantity and removes the book from the issued list.

**Program Flow:**

The main() function invokes the login process.

After successful login, users are presented with a main menu.

Users navigate between the User Module and Book Module.

Subsequently, users can perform various operations such as adding books, issuing books, updating user details, etc.

The program keeps running until the user chooses to exit

***FURTHER WORK TO BE ON PROJECT***

1. Multilayer Authentication**:** Using of authentication to enter in the user module and book modules
2. Implement a feature to search books by various criteria (e.g., author, publication date, genre).
3. Include more book details (e.g., ISBN, genre, language).

**CODE**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <string.h>**

**struct User {**

**char username[50];**

**char password[50];**

**};**

**struct Book {**

**int book\_id;**

**char book\_name[100];**

**char book\_auth[100];**

**char book\_pub[100];**

**int quantity;**

**char date[12];**

**};**

**struct Student {**

**int student\_id;**

**char student\_name[100];**

**char branch[50];**

**int semester;**

**char section;**

**char email[50];**

**char phone[15];**

**int book\_id;**

**char book\_name[100];**

**char book\_auth[100];**

**char book\_pub[100];**

**char date[12];**

**char dueDate[12];**

**};**

**void libmsLOGO(){**

**printf("\n###############################################################################\n");**

**printf("###############################################################################\n");**

**printf("###############################################################################\n");**

**printf("########################## #####################\n");**

**printf("########################## LIBRARY MANAGEMENT SYSTEM #####################\n");**

**printf("########################## 2023-2024 #####################\n");**

**printf("###############################################################################\n");**

**printf("########################## SESSION- 2022-2026 ############################\n");**

**printf("###############################################################################\n");**

**printf("Ankit Singh|BTech CSE IIyr|Univ.ROLL:2200290100025|KIET GROUP OF INSTITUTIONS\n");**

**printf("###############################################################################\n");**

**}**

**// Function prototypes**

**int login();**

**void bookModule();**

**void userModule();**

**void registerUser();**

**void updateUser();**

**void searchUser();**

**void deleteUser();**

**void showUserList();**

**void addBook();**

**void bookList();**

**void removeBook();**

**void updateBook();**

**void issueBook();**

**void displayIssuedBooks();**

**void returnBook();**

**void bookModule() {**

**libmsLOGO();**

**int choice;**

**do {**

**printf("\nMenu:\n");**

**printf("\t\t\t\t1. Add Book \n");**

**printf("\t\t\t\t2. Book List \n");**

**printf("\t\t\t\t3. Remove Book\n");**

**printf("\t\t\t\t4. Update Book\n");**

**printf("\t\t\t\t5. Issue Book\n");**

**printf("\t\t\t\t6. Display Issued Books\n");**

**printf("\t\t\t\t7. Return Book\n");**

**printf("\t\t\t\t8. Exit\n\n\n");**

**printf("\t\t\t\tEnter your choice: ");**

**scanf("%d", &choice);**

**switch (choice) {**

**case 1:**

**addBook();**

**break;**

**case 2:**

**bookList();**

**break;**

**case 3:**

**removeBook();**

**break;**

**case 4:**

**updateBook();**

**break;**

**case 5:**

**bookList();**

**issueBook();**

**break;**

**case 6:**

**displayIssuedBooks();**

**break;**

**case 7:**

**returnBook();**

**break;**

**case 8:**

**printf("\t\t\t\tExiting the program.\n");**

**break;**

**default:**

**printf("\t\t\t\tInvalid choice. Please enter a valid option.\n");**

**}**

**} while (choice != 8);**

**}**

**void addBook() {**

**FILE \*file = fopen("books.txt", "rb+");**

**if (file == NULL) {**

**printf("Error opening books.txt.\n");**

**return;**

**}**

**struct Book newBook;**

**system("cls");**

**printf("\t\t\t\tEnter book details:\n");**

**printf("\t\t\t\tBook ID: ");**

**scanf("%d", &newBook.book\_id);**

**// Check if the book already exists**

**FILE \*checkFile = fopen("books.txt", "rb");**

**if (checkFile == NULL) {**

**printf("Error opening books.txt for checking.\n");**

**fclose(file);**

**return;**

**}**

**struct Book tempBook;**

**while (fread(&tempBook, sizeof(struct Book), 1, checkFile) == 1) {**

**if (tempBook.book\_id == newBook.book\_id) {**

**printf("Book with ID %d already exists. Enter quantity to add: ", newBook.book\_id);**

**scanf("%d", &newBook.quantity);**

**tempBook.quantity += newBook.quantity;**

**fseek(file, -sizeof(struct Book), SEEK\_CUR);**

**fwrite(&tempBook, sizeof(struct Book), 1, file);**

**fclose(checkFile);**

**fclose(file);**

**printf("Quantity added successfully.\n");**

**return;**

**}**

**}**

**fclose(checkFile);**

**printf("\t\t\t\tBook Name: ");**

**scanf("%s", newBook.book\_name);**

**printf("\t\t\t\tAuthor: ");**

**scanf("%s", newBook.book\_auth);**

**printf("\t\t\t\tPublisher: ");**

**scanf("%s", newBook.book\_pub);**

**printf("\t\t\t\tQuantity: ");**

**scanf("%d", &newBook.quantity);**

**printf("\t\t\t\tDate (DD/MM/YYYY): ");**

**scanf("%s", newBook.date);**

**fseek(file, 0, SEEK\_END); // Move the file pointer to the end of the file**

**fwrite(&newBook, sizeof(struct Book), 1, file);**

**fclose(file);**

**printf("\t\t\t\tBook added successfully.\n");**

**}**

**void bookList() {**

**system("cls");**

**FILE \*file = fopen("books.txt", "rb");**

**if (file == NULL) {**

**printf("No books available.\n");**

**return;**

**}**

**struct Book book;**

**printf("\nBook List:\n");**

**printf("| %s | %s | %s | %s | %s | %s |\n", "ID", "Name", "Author", "Publisher", "Quantity", "Date");**

**while (fread(&book, sizeof(struct Book), 1, file)) {**

**printf("| %d | %s | %s | %s | %d | %s |\n",**

**book.book\_id, book.book\_name, book.book\_auth, book.book\_pub, book.quantity, book.date);**

**}**

**fclose(file);**

**}**

**void removeBook() {**

**system("cls");**

**FILE \*file = fopen("books.txt", "rb");**

**if (file == NULL) {**

**printf("No books available.\n");**

**return;**

**}**

**FILE \*tempFile = fopen("temp.txt", "wb");**

**int bookId;**

**printf("Enter the Book ID to remove: ");**

**scanf("%d", &bookId);**

**struct Book book;**

**int found = 0;**

**while (fread(&book, sizeof(struct Book), 1, file)) {**

**if (book.book\_id == bookId) {**

**found = 1;**

**printf("Book with ID %d removed successfully.\n", bookId);**

**} else {**

**fwrite(&book, sizeof(struct Book), 1, tempFile);**

**}**

**}**

**fclose(file);**

**fclose(tempFile);**

**if (!found) {**

**printf("Book with ID %d not found.\n", bookId);**

**remove("temp.txt");**

**return;**

**}**

**remove("books.txt");**

**rename("temp.txt", "books.txt");**

**}**

**void updateBook() {**

**system("cls");**

**FILE \*file = fopen("books.txt", "rb");**

**if (file == NULL) {**

**printf("No books available.\n");**

**return;**

**}**

**FILE \*tempFile = fopen("temp.txt", "wb");**

**int bookId;**

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

**scanf("%d", &bookId);**

**struct Book book;**

**int found = 0;**

**while (fread(&book, sizeof(struct Book), 1, file)) {**

**if (book.book\_id == bookId) {**

**found = 1;**

**printf("Enter new details for Book ID %d:\n", bookId);**

**printf("Book Name: ");**

**scanf("%s", book.book\_name);**

**printf("Author: ");**

**scanf("%s", book.book\_auth);**

**printf("Publisher: ");**

**scanf("%s", book.book\_pub);**

**printf("Quantity: ");**

**scanf("%d", &book.quantity);**

**printf("Date (DD/MM/YYYY): ");**

**scanf("%s", book.date);**

**}**

**fwrite(&book, sizeof(struct Book), 1, tempFile);**

**}**

**fclose(file);**

**fclose(tempFile);**

**if (!found) {**

**printf("Book with ID %d not found.\n", bookId);**

**remove("temp.txt");**

**return;**

**}**

**remove("books.txt");**

**rename("temp.txt", "books.txt");**

**printf("\t\t\tBook with ID %d updated successfully.\n", bookId);**

**}**

**void issueBook() {**

**system("cls");**

**FILE \*booksFile = fopen("books.txt", "rb+");**

**if (booksFile == NULL) {**

**printf("No books available.\n");**

**return;**

**}**

**FILE \*issuedFile = fopen("issuedBooks.txt", "ab");**

**if (issuedFile == NULL) {**

**fclose(booksFile);**

**printf("Error opening issuedBooks.txt.\n");**

**return;**

**}**

**struct Book book;**

**int bookId, found = 0;**

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

**scanf("%d", &bookId);**

**while (fread(&book, sizeof(struct Book), 1, booksFile)) {**

**if (book.book\_id == bookId) {**

**found = 1;**

**struct Student student;**

**printf("Enter student details:\n");**

**printf("Student ID: ");**

**scanf("%d", &student.student\_id);**

**printf("Student Name: ");**

**scanf("%s", student.student\_name);**

**printf("Branch: ");**

**scanf("%s", student.branch);**

**printf("Semester: ");**

**scanf("%d", &student.semester);**

**printf("Section: ");**

**scanf(" %c", &student.section);**

**printf("Phone: ");**

**scanf("%s", student.phone);**

**printf("Email: ");**

**scanf("%s", student.email);**

**student.book\_id = book.book\_id;**

**sprintf(student.book\_name, "%s", book.book\_name);**

**sprintf(student.book\_auth, "%s", book.book\_auth);**

**sprintf(student.book\_pub, "%s", book.book\_pub);**

**printf("Enter issue date (DD/MM/YYYY): ");**

**scanf("%s", student.date);**

**sprintf(student.dueDate, "%s", "AFTER\_15\_days");**

**// Update quantity in books file**

**book.quantity -= 1;**

**fseek(booksFile, -sizeof(struct Book), SEEK\_CUR);**

**fwrite(&book, sizeof(struct Book), 1, booksFile);**

**// Write issued book data to file**

**fwrite(&student, sizeof(struct Student), 1, issuedFile);**

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

**break;**

**}**

**}**

**fclose(booksFile);**

**fclose(issuedFile);**

**if (!found) {**

**printf("Book with ID %d not found.\n", bookId);**

**}**

**}**

**void displayIssuedBooks() {**

**system("cls");**

**FILE \*file = fopen("issuedBooks.txt", "rb");**

**if (file == NULL) {**

**printf("No books issued.\n");**

**return;**

**}**

**struct Student student;**

**printf("\n\t\t\tIssued Book List:\n\n\n");**

**printf("| %s | %s | %s | %s | %s | %s | %s | %s | %s |\n", "Stu ID", "Name", "Branch", "Sem", "Sec", "Phone", "Email", "Book ID", "Issue Date", "Due Date");**

**while (fread(&student, sizeof(struct Student), 1, file)) {**

**printf("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");**

**printf("| %d | %s | %s | %d | %c | %s | %s | %d | %s | %s |\n",**

**student.student\_id, student.student\_name, student.branch, student.semester, student.section, student.phone, student.email,**

**student.book\_id, student.date, student.dueDate);**

**}**

**fclose(file);**

**}**

**void returnBook() {**

**system("cls");**

**FILE \*issuedFile = fopen("issuedBooks.txt", "rb");**

**if (issuedFile == NULL) {**

**printf("No books issued.\n");**

**return;**

**}**

**FILE \*tempFile = fopen("temp.txt", "wb");**

**int bookId, studentId;**

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

**scanf("%d", &bookId);**

**printf("Enter the Student ID who is returning the book: ");**

**scanf("%d", &studentId);**

**struct Student student;**

**struct Book book;**

**int found = 0;**

**while (fread(&student, sizeof(struct Student), 1, issuedFile)) {**

**if (student.book\_id == bookId && student.student\_id == studentId) {**

**found = 1;**

**// Update quantity in books file**

**FILE \*booksFile = fopen("books.txt", "rb+");**

**if (booksFile != NULL) {**

**while (fread(&book, sizeof(struct Book), 1, booksFile)) {**

**if (book.book\_id == bookId) {**

**book.quantity += 1;**

**fseek(booksFile, -sizeof(struct Book), SEEK\_CUR);**

**fwrite(&book, sizeof(struct Book), 1, booksFile);**

**fclose(booksFile);**

**break;**

**}**

**}**

**} else {**

**printf("Error opening books.txt.\n");**

**}**

**printf("Book with ID %d returned successfully.\n", bookId);**

**} else {**

**fwrite(&student, sizeof(struct Student), 1, tempFile);**

**}**

**}**

**fclose(issuedFile);**

**fclose(tempFile);**

**if (!found) {**

**printf("Book with ID %d issued to Student ID %d not found.\n", bookId, studentId);**

**remove("temp.txt");**

**return;**

**}**

**remove("issuedBooks.txt");**

**rename("temp.txt", "issuedBooks.txt");**

**}**

**int main() {**

**label:**

**libmsLOGO();**

**if (login()) {**

**int choice;**

**do {**

**printf("\n<== Library Management System ==>\n");**

**printf("1. User Module\n");**

**printf("2. Book Module\n");**

**printf("0. Exit\n");**

**printf("Enter your choice: ");**

**scanf("%d", &choice);**

**switch (choice) {**

**case 1:**

**userModule(); //**

**break;**

**case 2:**

**bookModule();**

**break;**

**case 0:**

**printf("Exiting the program.\n");**

**break;**

**default:**

**printf("Invalid choice. Please try again.\n");**

**}**

**} while (choice != 0);**

**}**

**else {**

**printf("Login failed. ENTER CORRECT DETAILS.\n");**

**goto label;**

**}**

**return 0;**

**}**

**int login() {**

**char inputUsername[50];**

**char inputPassword[50];**

**struct User user;**

**FILE \*userFile;**

**printf("\n\n\n\n\t\t\t\tEnter your username: ");**

**scanf("%s", inputUsername);**

**printf("\t\t\t\tEnter your password: ");**

**scanf("%s", inputPassword);**

**userFile = fopen("users.txt", "r");**

**if (userFile == NULL) {**

**printf("Error opening user file. Exiting the program.\n");**

**exit(1);**

**}**

**while (fscanf(userFile, "%s %s\n", user.username, user.password) == 2) {**

**if (strcmp(inputUsername, user.username) == 0 && strcmp(inputPassword, user.password) == 0) {**

**printf("\t\t\t\tLogin successful!\n");**

**fclose(userFile);**

**return 1;**

**}**

**}**

**fclose(userFile);**

**return 0;**

**}**

**void userModule() {**

**libmsLOGO();**

**int choice;**

**do {**

**printf("\t\t\t\t\n<== User Module ==>\n");**

**printf("1. Register User\n");**

**printf("2. Update User\n");**

**printf("3. Search User\n");**

**printf("4. Delete User\n");**

**printf("5. Show User List\n");**

**printf("0. Back to Main Menu\n");**

**printf("Enter your choice: ");**

**scanf("%d", &choice);**

**switch (choice) {**

**case 1:**

**registerUser();**

**break;**

**case 2:**

**updateUser();**

**break;**

**case 3:**

**searchUser();**

**break;**

**case 4:**

**deleteUser();**

**break;**

**case 5:**

**showUserList();**

**break;**

**case 0:**

**printf("Returning to the main menu.\n");**

**break;**

**default:**

**printf("Invalid choice. Please try again.\n");**

**}**

**} while (choice != 0);**

**}**

**void registerUser() {**

**struct User user;**

**FILE \*userFile;**

**system("cls");**

**printf("Enter new username: ");**

**scanf("%s", user.username);**

**printf("Enter new password: ");**

**scanf("%s", user.password);**

**userFile = fopen("users.txt", "a");**

**if (userFile == NULL) {**

**printf("Error opening user file. Exiting the program.\n");**

**exit(1);**

**}**

**fprintf(userFile, "\n%s %s\n", user.username, user.password);**

**fclose(userFile);**

**printf("\t\t\t\tUser registered successfully!\n");**

**}**

**void updateUser() {**

**char targetUsername[50];**

**struct User user;**

**FILE \*userFile, \*tempFile;**

**system("cls");**

**printf("Enter the username to update: ");**

**scanf("%s", targetUsername);**

**userFile = fopen("users.txt", "r");**

**tempFile = fopen("temp\_users.txt", "w");**

**if (userFile == NULL || tempFile == NULL) {**

**printf("Error opening user files. Exiting the program.\n");**

**exit(1);**

**}**

**int found = 0;**

**while (fscanf(userFile, "%s %s\n", user.username, user.password) == 2) {**

**if (strcmp(targetUsername, user.username) == 0) {**

**found = 1;**

**printf("Enter new password for %s: ", targetUsername);**

**scanf("%s", user.password);**

**}**

**fprintf(tempFile, "%s %s\n", user.username, user.password);**

**}**

**fclose(userFile);**

**fclose(tempFile);**

**remove("users.txt");**

**rename("temp\_users.txt", "users.txt");**

**if (found) {**

**printf("User updated successfully!\n");**

**} else {**

**printf("User not found.\n");**

**}**

**}**

**void searchUser() {**

**char targetUsername[50];**

**struct User user;**

**FILE \*userFile;**

**system("cls");**

**printf("Enter the username to search: ");**

**scanf("%s", targetUsername);**

**userFile = fopen("users.txt", "r");**

**if (userFile == NULL) {**

**printf("Error opening user file. Exiting the program.\n");**

**exit(1);**

**}**

**int found = 0;**

**while (fscanf(userFile, "%s %s\n", user.username, user.password) == 2) {**

**if (strcmp(targetUsername, user.username) == 0) {**

**found = 1;**

**printf("User found!\n");**

**printf("\n===================================\n");**

**printf("Username: %s\n", user.username);**

**printf("Password: %s\n", user.password);**

**printf("\n===================================\n");**

**break;**

**}**

**}**

**fclose(userFile);**

**if (!found) {**

**printf("User not found.\n");**

**}**

**}**

**void deleteUser() {**

**char targetUsername[50];**

**struct User user;**

**FILE \*userFile, \*tempFile;**

**system("cls");**

**printf("Enter the username to delete: ");**

**scanf("%s", targetUsername);**

**userFile = fopen("users.txt", "r");**

**tempFile = fopen("temp\_users.txt", "w");**

**if (userFile == NULL || tempFile == NULL) {**

**printf("Error opening user files. Exiting the program.\n");**

**exit(1);**

**}**

**int found = 0;**

**while (fscanf(userFile, "%s %s\n", user.username, user.password) == 2) {**

**if (strcmp(targetUsername, user.username) == 0) {**

**found = 1;**

**printf("User deleted successfully!\n");**

**continue;**

**}**

**fprintf(tempFile, "%s %s\n", user.username, user.password);**

**}**

**fclose(userFile);**

**fclose(tempFile);**

**remove("users.txt");**

**rename("temp\_users.txt", "users.txt");**

**if (!found) {**

**printf("User not found.\n");**

**}**

**}**

**void showUserList() {**

**struct User user;**

**FILE \*userFile;**

**system("cls");**

**userFile = fopen("users.txt", "r");**

**if (userFile == NULL) {**

**printf("Error opening user file. Exiting the program.\n");**

**exit(1);**

**}**

**printf("\n<== User List ==>\n");**

**while (fscanf(userFile, "%s %s\n", user.username, user.password) == 2) {**

**printf("\t\t\t==============================================\n");**

**printf("\t\t\tUsername: %s\n", user.username);**

**printf("\t\t\tpassword: %s\n", user.password);**

**printf("\t\t\t==============================================\n");**

**}**

**fclose(userFile);**

**}**

**OUTPUT SCREEN**



































