**DLITHE PROJECT REPORT**

**PROJECT ID :** C22

**PROJECT TITLE :** LIBRARY MANAGEMENT SYSTEM

**TEAM MEMBERS :** Keerthan Marathe (4MT21IS019)

Thrishik Shetty (4MT21IS056) Karthik Narayan Pai(4MT21IS017) Makarand Gore (4MT21IS020) Sudeep (4MT21IS052)

# REPORT

**Abstract :**

The Library Management System is a software application designed to help manage books in the library and streamline inventory management in school or college and public libraries. The system provides login options: ADMIN. The ADMIN login allows authorised personnel to control and manage the Library Functions such as adding books, removing books, searching books and make rentals.

# 

# Introduction :

**Background :**

#### The Library Management System has two panels:

* **Book Management panel:** Authorised personnel responsible for managing the books in the library and performing administrative functions such as Adding books, Modify, Rent books etc.
* **User Management panel:** Authorised personnel responsible for managing the Users or Readers and performing administrative functions such as Adding users, Modify users, List user, List Rentals, Search user, Delete user etc.

# Objectives :

The main objectives of the Library Management System are as follows:

* To maintain an organized information source of users.
* To manage and track the daily work of the library.
* To manage information related to books their names, codes, author names etc.
* To provide the ability to add, display, modify, and delete records of the books and users.
* To help the libraries to organize and manage their materials

# Technologies Used :

The Library Management System is implemented using the following technologies:

* C programming language
* File handling for data storage

# System Architecture :

## Front-End :

The front-end of the system is implemented using the console or command-line interface, where users interact with the system by entering commands and providing input.

## Back-End :

The back-end of the system consists of C programming logic that performs various operations such as adding, displaying, modifying, and deleting records. It also handles file handling to store and retrieve data.

## Database :

The system uses files to act as databases for storing information about books and Users. Each entity has a separate file for data storage.

# File Handling in this Program:

* The program uses file handling to store and retrieve data for books and users.
* Each entity (books and users) has its own separate file for data storage.
* File operations such as opening, reading, writing, modifying, and deleting records are performed using file pointers and functions.

# Project Modules :

The project consists of the following modules:

* Module 1: Book Management Functions
  + Add Books
  + Delete Books
  + Modify Book
  + List book
  + Rent Book
  + Search Book
* Module 2: User Management Functions
  + Add and Modify User
  + User list
  + Search User
  + Delete User

# Design and Implementation :

## Front-End Design :

* The front-end design is based on a command-line interface, where users can select options by entering numbers corresponding to their desired actions.

## Back-End Design :

* The back-end logic is implemented in the C programming language.
* Each module has its functions for adding, displaying, modifying, and deleting records.
* File handling functions are used to perform operations on data files.

## Database Design :

* Data for medicines, suppliers, and customers are stored in separate files.
* Each file contains records in a structured format.

# Features and Functionality:

* Feature 1: **Increased efficiency:**

LMSs can automate many tasks, such as acquisitions, cataloging, circulation, and interlibrary loan. This can free up librarians to focus on other important duties, such as providing customer service and developing new programs.

* Feature 2: **Improved accuracy:**

LMSs can help to improve the accuracy of library records. This is because they are less prone to human error than manual systems.

* Feature 3: **Reduced costs:**

LMSs can help to reduce the costs of library operations. This is because they can help to eliminate the need for manual paperwork and reduce the amount of time that librarians spend on administrative tasks.

* Feature 4: **Enhanced patron experience:**

LMSs can help to enhance the patron experience by making it easier for them to find and borrow library materials. They can also provide patrons with access to online resources, such as e-books, articles, and etc.

# Testing :

## Unit Testing :

* Each module is tested individually to ensure that it performs its functions correctly.
* Test cases are created to cover various scenarios.

## Integration Testing :

* Modules are integrated to test the overall functionality of the system.
* Data flow between modules is tested.

## User Acceptance Testing :

* The system is tested by end-users to ensure it meets their requirements and expectations.
* Any feedback or issues raised by users are addressed and resolved.

# Challenges Faced :

* Implementing file handling for data storage and retrieval.
* Ensuring data consistency and accuracy.
* Handling errors and exceptions gracefully.
* Future Enhancements
* Implementing a graphical user interface (GUI) for a more user-friendly experience.
* Adding features for generating reports and statistics.
* Implementing security measures to protect data.

# Conclusion :

The Library Management System is a valuable tool for storing books and user details in a proper manner. This system helps the admin or librarian to keep the record of the day-to-day activities of the library. It simplifies record keeping and ensures smooth operations. With future enhancements, it can further improve its functionality and usability.

# Appendices:

## Appendix A : Login System

## The login system is an essential part of the Library Management System, Authorised person (ADMIN) can login to the library Management system using password “admin” which offers two types of panels:

### Appendix B : BOOK Management Panel

This Panel provides for managing the books in the library. Only authorised personnel will have access to the Book Management Panel. Once logged in as an ADMIN, the user gains control over stock management of books and price alteration, Modify books, Rent books etc.

### Appendix C : USER Management Panel

The USER login is for regular users or customers of the library. Registered users can log in to access the system to view available books and make book rentals. USER login provides access to administrative functions such as adding users, modifying users, viewing the list of Users and list of Rentals, Search Users and Delete Users.

## Appendix B : **Add Books**

## ADMIN can add new books to the library by providing details such as book name, quantity, author name, Book Publisher, Book ID.

Appendix C : **Delete Books**

ADMIN can remove books that are no longer available in the library by providing details such as book name.

Appendix D : **Modify Book**

ADMIN can modify the details of the book by providing the valid Book name.

Appendix E : **Search Book**

ADMIN can search the book to check the book by entering author name or book name .it displays book name, quantity, author name, Book Publisher, Book ID.

Appendix F : **Add and Modify User**

Can add the details of users (students, lecturers, and librarians) and Modify them.

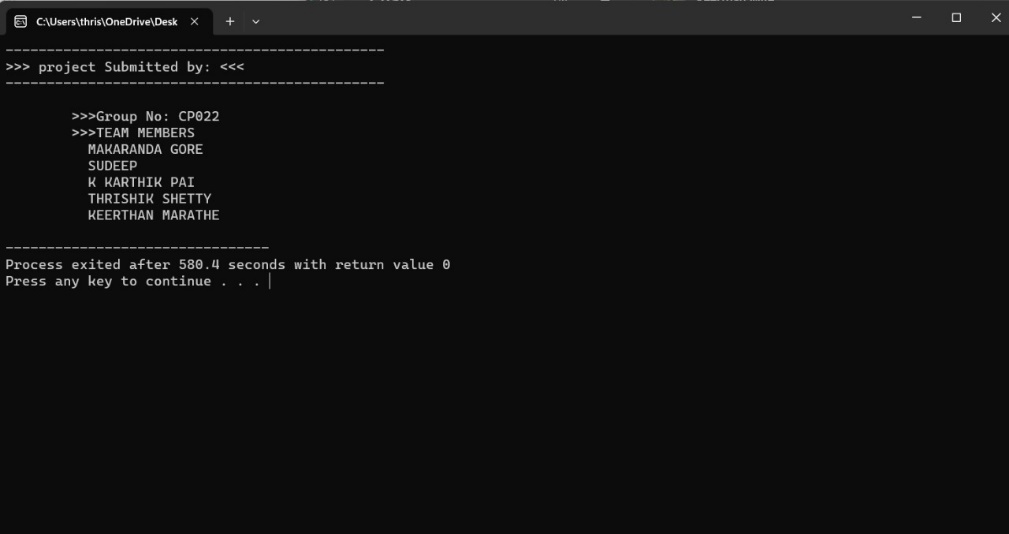
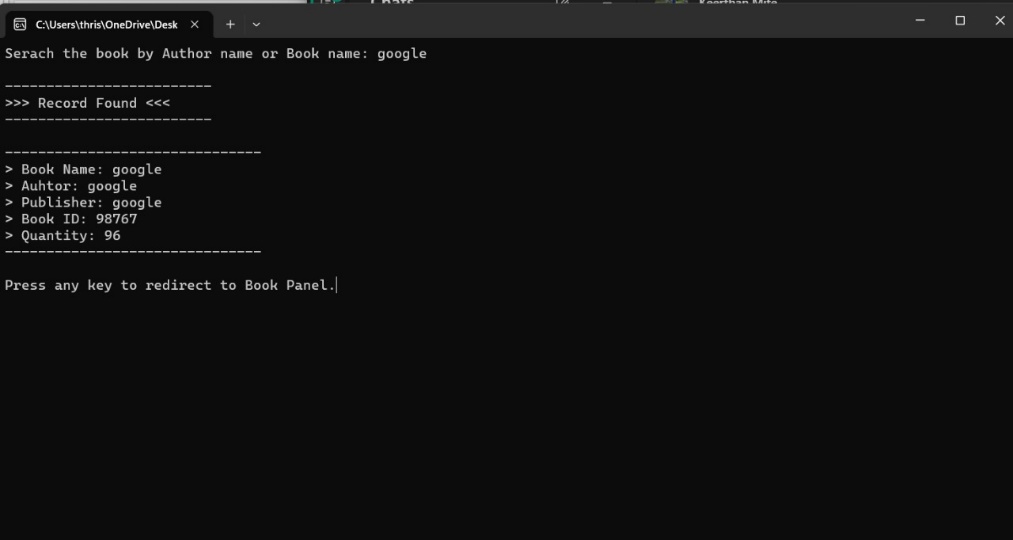
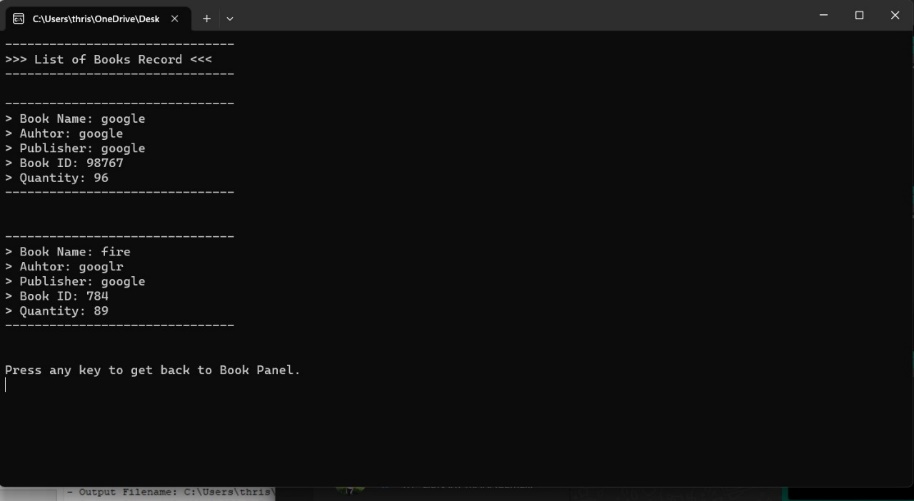
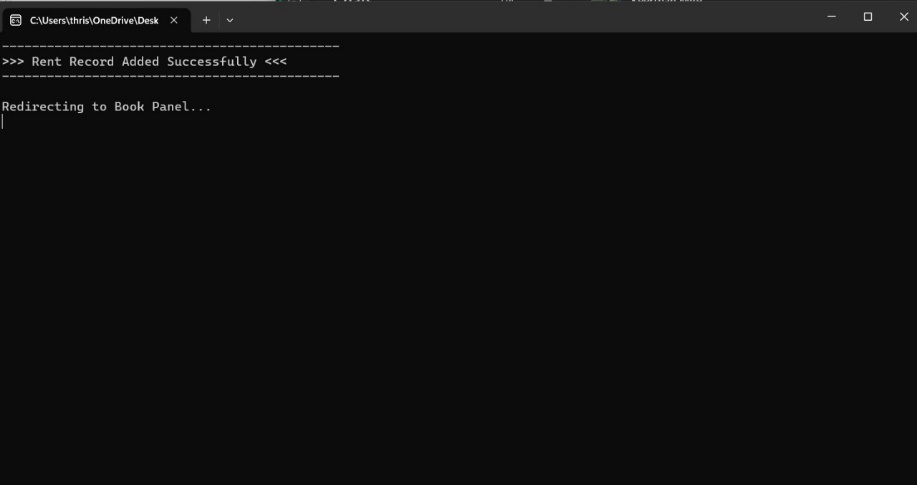
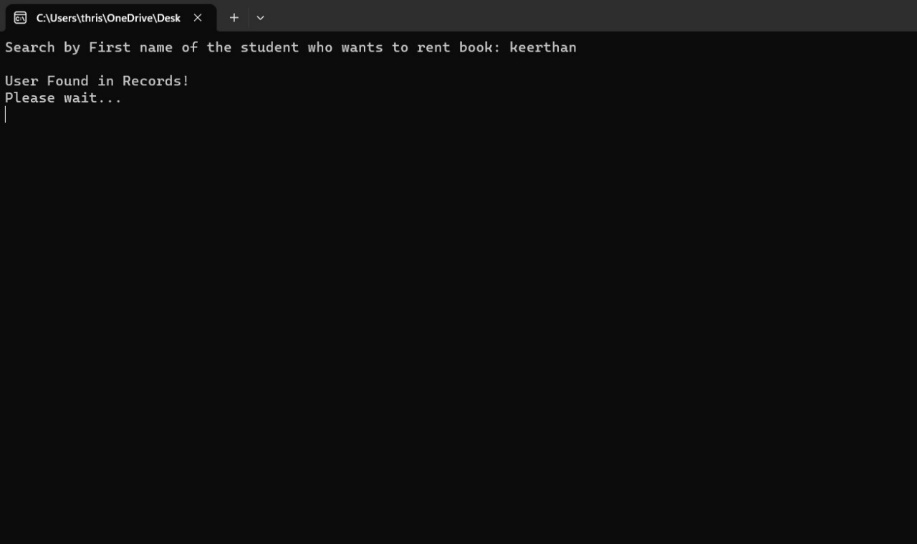
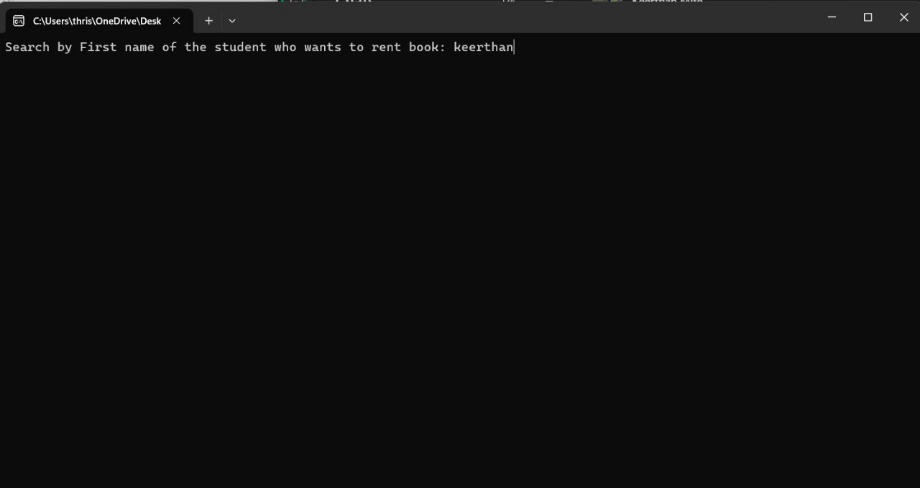
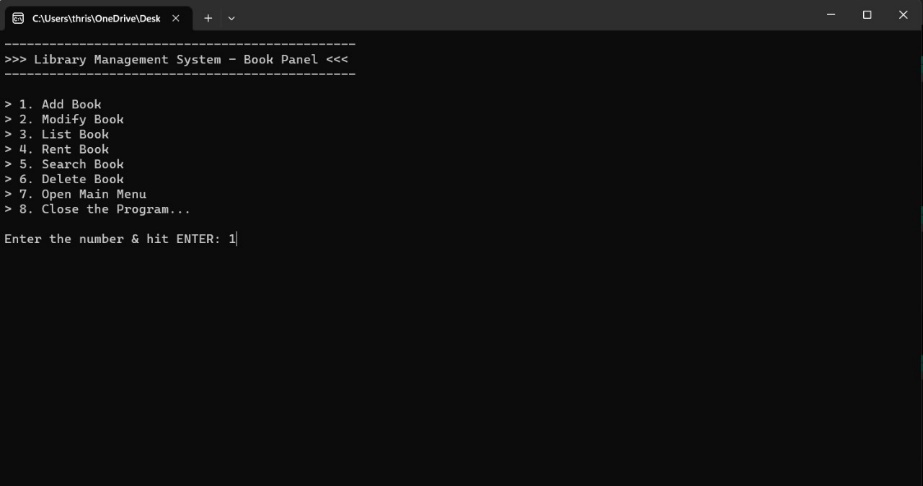
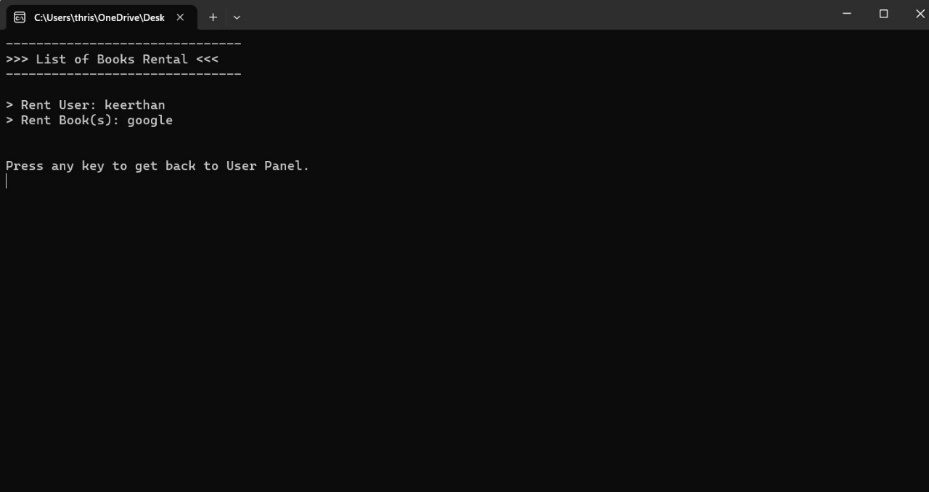
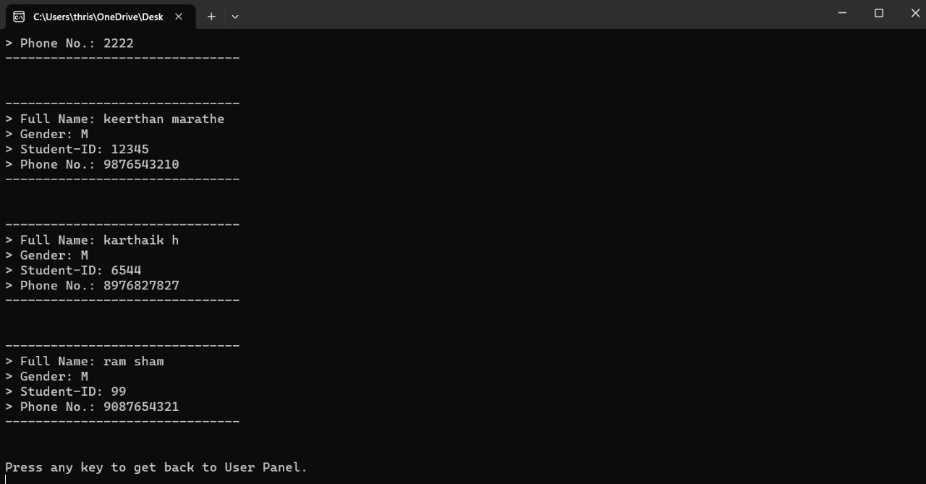
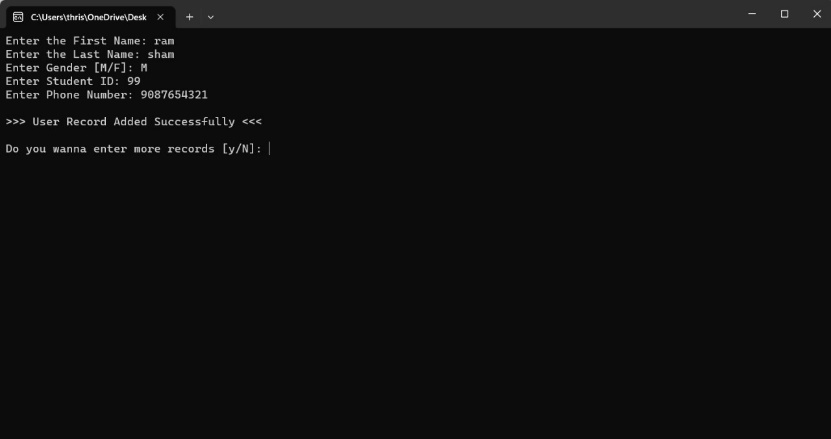
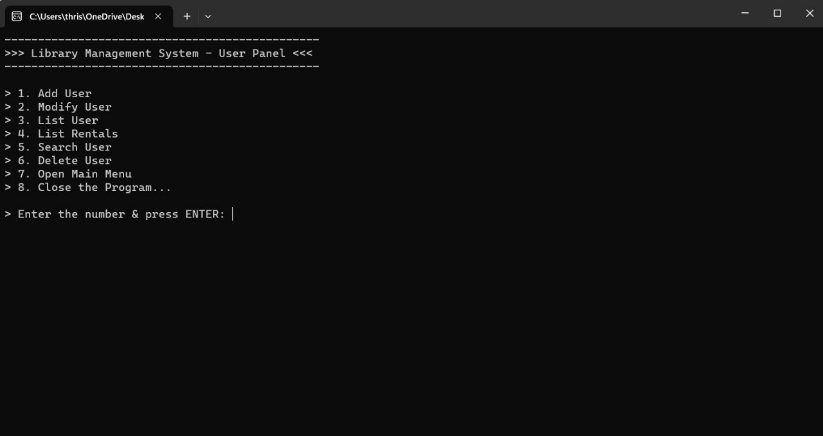
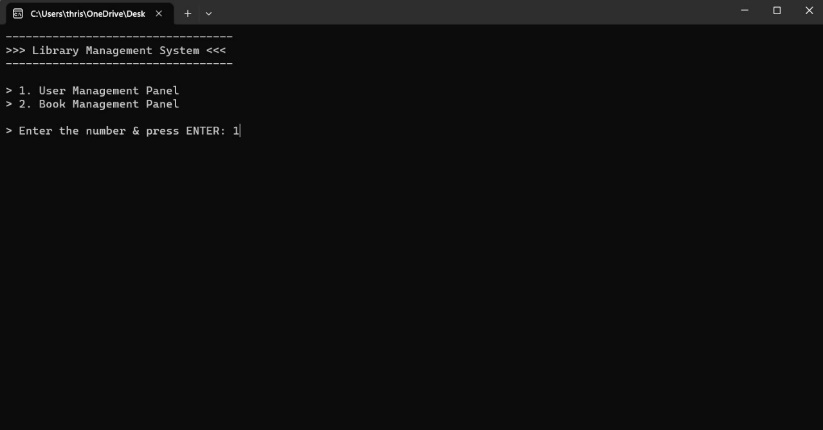
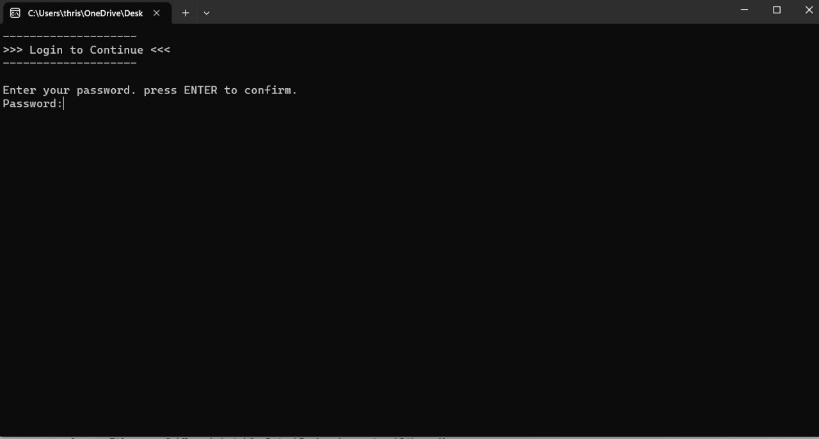
Appendix G : **User list**

View the available users in the stock along with their details.

Appendix H : **Search User**

ADMIN can find a particular piece of information or data regarding the books from a User list.

# Screenshots :



**Code Snippets :**

#include<stdio.h>

#include<conio.h>

#include<string.h>

#include<stdlib.h>

#include<windows.h>

// Define Constant KeyWords

#define ENTER 13

#define TAB 9

#define BKSP 8

// System Operation Functions

void password(void);

void menu(void);

void userPanel(void);

void bookPanel(void);

void endScreen(void);

// User Operation Functions

void addUser(void);

void modifyUser(void);

void listUser(void);

int searchUser(int);

void rentList(void);

void deleteUser(void);

// Book Operation Functions

void addBook(void);

int modifyBook(int);

void listBook(void);

void rentBook(void);

int searchBook(int);

void deleteBook(void);

// Main Function

void main(){

password();

}

// System Functions

int passTerminator = 1;

int bookStock = 0;

char rentName[255], bookName[255];

// If the user enters invaild password 3 times then the program gets terminated.

// Password: ADMIN

void password(){

system("cls");

fflush(stdin);

char pwd[255];

char code[255] = {"ADMIN"};

int i = 0;

char ch;

printf("--------------------\n");

printf(">>> Login to Continue <<<\n");

printf("--------------------\n\n");

printf("Enter your password. press ENTER to confirm. \n");

printf("Password:");

while(1)

{

ch = getch(); // get key

if(ch == ENTER || ch == TAB)

{

pwd[i] = '\0';

break;

}

else if(ch == BKSP)

{

if(i > 0)

{

i--;

printf("\b \b"); // for backspace

}

}

else

{

pwd[i++] = ch;

printf("\* \b"); // to replace password character with \*

}

}

fflush(stdin);

// verifies the password

if(strcmp(code, pwd) == 0)

{

printf("\nCorrect Password!");

Sleep(2000);

menu();

}

else

{

printf("\nInvaild Password!");

(passTerminator == 3) ? exit(0) : passTerminator++;

Sleep(2000);

password();

}

}

// Main Selection menu between User & Book Panel

void menu(){

system("cls");

fflush(stdin);

int number;

printf("----------------------------------\n");

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

printf("----------------------------------\n\n");

printf("> 1. User Management Panel \n");

printf("> 2. Book Management Panel \n\n");

printf("> Enter the number & press ENTER: ");

scanf("%d",&number);

fflush(stdin);

switch(number)

{

case 1:

userPanel();

break;

case 2:

bookPanel();

break;

default:

printf("\n>>> Invaild Input! <<<");

Sleep(2000);

menu();

}

}

// User Panel sub-functions

void userPanel(){

system("cls");

fflush(stdin);

int number;

printf("-----------------------------------------------\n");

printf(">>> Library Management System - User Panel <<< \n");

printf("-----------------------------------------------\n\n");

printf("> 1. Add User \n");

printf("> 2. Modify User \n");

printf("> 3. List User \n");

printf("> 4. List Rentals \n");

printf("> 5. Search User \n");

printf("> 6. Delete User \n");

printf("> 7. Open Main Menu \n");

printf("> 8. Close the Program... \n\n");

printf("> Enter the number & press ENTER: ");

scanf("%d",&number);

fflush(stdin);

switch(number)

{

case 1:

addUser();

break;

case 2:

modifyUser();

break;

case 3:

listUser();

break;

case 4:

rentList();

break;

case 5:

searchUser(0);

break;

case 6:

deleteUser();

break;

case 7:

menu();

break;

case 8:

endScreen();

break;

default:

printf("Invaild Input!");

Sleep(2000);

userPanel();

}

}

// Book Panel sub-functions

void bookPanel(){

system("cls");

fflush(stdin);

int number;

printf("-----------------------------------------------\n");

printf(">>> Library Management System - Book Panel <<< \n");

printf("-----------------------------------------------\n\n");

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

printf("> 2. Modify Book \n");

printf("> 3. List Book \n");

printf("> 4. Rent Book \n");

printf("> 5. Search Book \n");

printf("> 6. Delete Book \n");

printf("> 7. Open Main Menu \n");

printf("> 8. Close the Program... \n");

printf("\nEnter the number & hit ENTER: ");

scanf("%d",&number);

fflush(stdin);

switch(number)

{

case 1:

addBook();

break;

case 2:

modifyBook(0);

break;

case 3:

listBook();

break;

case 4:

rentBook();

break;

case 5:

searchBook(0);

break;

case 6:

deleteBook();

break;

case 7:

menu();

break;

case 8:

endScreen();

break;

default:

printf("Invaild Input!");

Sleep(2000);

bookPanel();

}

}

void endScreen(){

system("cls");

fflush(stdin);

printf("----------------------------------------------\n");

printf(">>> project Submitted by: <<< \n");

printf("----------------------------------------------\n\n");

printf(" >>>Group No: CP022\n");

printf(" >>>TEAM MEMBERS\n");

printf(" MAKARANDA GORE \n");

printf(" SUDEEP\n");

printf(" K KARTHIK PAI\n");

printf(" THRISHIK SHETTY\n");

printf(" KEERTHAN MARATHE\n");

exit(0);

}

// User Functions

// Creates new file if old doesn't exist and saves user records in it

void addUser(){

label1:

system("cls");

fflush(stdin);

char fname[255], lname[255];

char gender;

double sid, phone, balance;

FILE \*uR = fopen("user\_Records.txt", "a+");

if(uR != NULL)

{

printf("Enter the First Name: ");

gets(fname) ;

printf("Enter the Last Name: ");

gets(lname);

printf("Enter Gender [M/F]: ");

scanf("%c",&gender);

printf("Enter Student ID: ");

scanf("%lf",&sid);

printf("Enter Phone Number: ");

scanf("%lf",&phone);

fprintf(uR, "%s %s %c %.0lf %.0lf \n", fname, lname, gender, sid, phone);

printf("\n>>> User Record Added Successfully <<< \n");

}

else

{

printf("Unable to open/locate the file.");

}

fclose(uR);

fflush(stdin);

//retry screen

char input;

printf("\nDo you wanna enter more records [y/N]: ");

scanf("%c",&input);

if(input == 'y' || input=='Y')

{

goto label1;

}

else if(input=='n' || input=='N')

{

printf("\nRedirecting to User Panel.");

Sleep(2000);

userPanel();

}

else

{

printf("\nInvaild input. Redirecting to User Panel.");

Sleep(2000);

userPanel();

}

}

// Edit the user details and saves it

void modifyUser(){

system("cls");

fflush(stdin);

char fname[255], lname[255], gender[5];

char fname1[255], lname1[255], gender1[5];

double sid, sid1, phone, phone1;

int compare, flag=0;

char find[255];

printf("Enter the name of the person you want to modify the detail: ");

gets(find);

fflush(stdin);

FILE \*uR = fopen("user\_Records.txt", "r");

FILE \*pT = fopen("temporary.txt", "a");

while(fscanf(uR, "%s %s %s %lf %lf \n", fname, lname, gender, &sid, &phone) != EOF)

{

compare = strcmp(find, fname);

if(compare == 0)

{

printf("\n---------------------------------------------\n");

printf(">>> Record Found, Allowing Modifications <<<\n");

printf("-----------------------------------------------\n\n");

printf("> Enter First Name: ");

gets(fname1);

printf("> Enter Last Name: ");

gets(lname1);

printf("> Enter Gender: ");

gets(gender1);

printf("> Enter Student ID: ");

scanf("%lf",&sid1);

printf("> Enter Phone Number: ");

scanf("%lf",&phone1);

fprintf(pT, "%s %s %s %.0lf %.0lf \n",fname1, lname1, gender1, sid1, phone1);

printf("\n\nProcessing your changes....");

flag = 1;

}

else

{

fprintf(pT, "%s %s %s %.0lf %.0lf \n",fname, lname, gender, sid, phone);

}

}

fclose(uR);

fclose(pT);

fflush(stdin);

uR = fopen("user\_Records.txt", "w");

fclose(uR);

if(flag == 0)

{

printf("\n\n-------------------------------\n");

printf(">>> Record Not Found <<<\n");

printf("-------------------------------\n\n");

printf("Redirecting to User Panel...");

}

uR = fopen("user\_Records.txt", "a");

pT = fopen("temporary.txt", "r");

while(fscanf(pT, "%s %s %s %lf %lf \n", fname, lname, gender, &sid, &phone) != EOF)

{

fprintf(uR, "%s %s %s %.0lf %.0lf \n", fname, lname, gender, sid, phone);

}

fclose(uR);

fclose(pT);

pT = fopen("temporary.txt", "w");

fclose(pT);

fflush(stdin);

Sleep(2000);

userPanel();

}

// Lists all the user records from user\_Records.txt file

void listUser(){

system("cls");

fflush(stdin);

FILE \*uR = fopen("user\_Records.txt", "r");

char fname[255], lname[255], gender[5];

double sid, phone;

int counter=0;

printf("-------------------------------\n");

printf(">>> List of Users Record <<< \n");

printf("-------------------------------\n\n");

while(fscanf(uR, "%s %s %s %lf %lf \n", fname, lname, gender, &sid, &phone) != EOF)

{

strcat(fname, " ");

strcat(fname, lname);

printf("-------------------------------\n");

printf("> Full Name: %s \n", fname);

//printf("> Last Name: %s \n", lname);

printf("> Gender: %s \n", gender);

printf("> Student-ID: %.0lf \n", sid);

printf("> Phone No.: %.0lf \n", phone);

printf("-------------------------------\n\n\n");

counter++;

}

fclose(uR);

if(counter == 0)

{

printf("-------------------------------------\n");

printf("There is no user records added yet...\n");

printf("--------------------------------------\n\n");

}

printf("Press any key to get back to User Panel.\n");

getch();

userPanel();

}

// this checks if the specified user exists in the records or not

int searchUser(int nameSearcher){

label2:

system("cls");

fflush(stdin);

char fname[255], lname[255], gender[5];

double sid, phone;

int flag=0;

int compare;

char find[255];

(nameSearcher != 3) ? printf("Search by First name of the student: ") : printf("Search by First name of the student who wants to rent book: ");

gets(find);

FILE \*uR = fopen("user\_Records.txt", "r");

while(fscanf(uR, "%s %s %s %lf %lf \n", fname, lname, gender, &sid, &phone) != EOF)

{

//strcmp(variable, variable1) -- if both the strings are equal then it will return 0 otherwise a random number.

compare = strcmp(find, fname);

if(compare == 0)

{

if(nameSearcher != 3)

{

strcat(fname, " ");

strcat(fname, lname);

printf("\n---------------------\n");

printf(">>> Record Found <<< \n");

printf("---------------------\n\n");

printf("-------------------------------\n");

printf("> Full Name: %s \n", fname);

printf("> Gender: %s \n", gender);

printf("> Student-ID: %.0lf \n", sid);

printf("> Phone Number: %.0lf \n", phone);

printf("-------------------------------\n\n");

}

strcpy(rentName, fname);

flag=1;

}

}

fclose(uR);

if(flag == 0)

{

printf("\n>>> Record Not Found <<< \n\n");

}

fflush(stdin);

if(nameSearcher != 3)

{

printf("Press any key to redirect back to Panel.");

getch();

userPanel();

}

else if(nameSearcher == 3 && flag == 1)

{

return 5;

}

}

// deletes the user information from user\_Records.txt file

void deleteUser(){

system("cls");

fflush(stdin);

char fname[255], lname[255], gender[5];

char fname1[255], lname1[255], gender1[5];

double sid, sid1, phone, phone1;

int compare, flag=0;

char find[255];

printf("Enter the name of the person you want to delete the detail: ");

gets(find);

fflush(stdin);

FILE \*uR = fopen("user\_Records.txt", "r");

FILE \*pT = fopen("temporary.txt", "a");

while(fscanf(uR, "%s %s %s %lf %lf \n", fname, lname, gender, &sid, &phone) != EOF)

{

compare = strcmp(find, fname);

if(compare == 0)

{

printf("\n---------------------------------------------\n");

printf(">>> Record Deleted <<<\n");

printf("-----------------------------------------------\n\n");

printf("Redirecting to User Panel...");

flag = 1;

}

else

{

fprintf(uR, "%s %s %s %.0lf %.0lf \n",fname, lname, gender, sid, phone);

}

}

fclose(uR);

fclose(pT);

fflush(stdin);

uR = fopen("user\_Records.txt", "w");

fclose(uR);

if(flag == 0)

{

printf("\n\n-------------------------------\n");

printf(">>> Record Not Found <<<\n");

printf("-------------------------------\n\n");

printf("Redirecting to User Panel...");

}

uR = fopen("user\_Records.txt", "a");

pT = fopen("temporary.txt", "r");

while(fscanf(pT, "%s %s %s %lf %lf \n", fname, lname, gender, &sid, &phone) != EOF)

{

fprintf(uR, "%s %s %s %.0lf %.0lf \n", fname, lname, gender, sid, phone);

}

fclose(uR);

fclose(pT);

pT = fopen("temporary.txt", "w");

fclose(pT);

fflush(stdin);

Sleep(2000);

userPanel();

}

// Book Functions

// add the books record in book\_Records.txt file

void addBook(){

label3:

system("cls");

fflush(stdin);

char name[255], author[255], publisher[255];

double bookid, quantity;

FILE \*uR = fopen("book\_Records.txt", "ab+");

if(uR != NULL)

{

printf("Enter Book Name: ");

gets(name);

printf("Enter Book Author: ");

gets(author);

printf("Enter Book Publisher: ");

gets(publisher);

fflush(stdin);

printf("Enter Book ID: ");

scanf("%lf",&bookid);

printf("Enter Book Quantity: ");

scanf("%lf",&quantity);

fprintf(uR, "%s %s %s %.0lf %.0lf \n", name, author, publisher, bookid, quantity);

printf("\n>>> Book Record Added Successfully <<< \n");

}

else

{

printf("Unable to open/locate the file.");

}

fclose(uR);

fflush(stdin);

char input;

printf("\nDo you want enter more records [y/N]: ");

scanf("%c",&input);

if(input == 'y' || input=='Y')

{

goto label3;

}

else if(input=='n' || input=='N')

{

printf("\nRedirecting to Book Panel.");

Sleep(2000);

bookPanel();

}

else

{

printf("\nInvaild input. Redirecting to Book Panel.");

Sleep(2000);

bookPanel();

}

}

// edits the book details according to you and saves it again

int modifyBook(int rentModifier){

system("cls");

fflush(stdin);

char name[255], author[255], publisher[255];

double bookid, quantity;

char name1[255], author1[255], publisher1[255];

double bookid1, quantity1;

int flag=0;

int compare;

char find[255];

if (rentModifier != 5)

{

printf("Enter the name of the book you want to see the detail: ");

gets(find);

fflush(stdin);

}

else

{

strcpy(find, bookName);

}

FILE \*uR = fopen("book\_Records.txt", "r");

FILE \*pT = fopen("temporary.txt", "a");

while(fscanf(uR, "%s %s %s %lf %lf \n", name, author, publisher, &bookid, &quantity) != EOF)

{

compare = strcmp(find, name);

if(compare == 0)

{

if(rentModifier != 5)

{

printf("\n---------------------------------------------\n");

printf(">>> Record Found, Allowing Modifications <<<\n");

printf("-----------------------------------------------\n\n");

printf("> Enter Book Name: ");

gets(name1);

printf("> Enter Authour: ");

gets(author1);

printf("> Enter Publisher: ");

gets(publisher1);

fflush(stdin);

printf("> Enter Book ID: ");

scanf("%lf",&bookid1);

printf("> Enter Quantity: ");

scanf("%lf",&quantity1);

fprintf(pT, "%s %s %s %.0lf %.0lf \n", name1, author1, publisher1, bookid1, quantity1);

printf("\n\nProcessing your changes....");

}

else

{

quantity = bookStock;

fprintf(pT, "%s %s %s %.0lf %.0lf \n", name, author, publisher, bookid, quantity);

}

flag = 1;

}

else

{

fprintf(pT, "%s %s %s %.0lf %.0lf \n", name, author, publisher, bookid, quantity);

}

}

fclose(uR);

fclose(pT);

fflush(stdin);

uR = fopen("book\_Records.txt", "w");

fclose(uR);

if(flag == 0)

{

printf("\n\n-------------------------------\n");

printf(">>> Record Not Found <<<\n");

printf("-------------------------------\n\n");

printf("Redirecting to Book Panel...");

}

uR = fopen("book\_Records.txt", "a");

pT = fopen("temporary.txt", "r");

while(fscanf(pT, "%s %s %s %lf %lf \n", name, author, publisher, &bookid, &quantity) != EOF)

{

fprintf(uR, "%s %s %s %.0lf %.0lf \n", name, author, publisher, bookid, quantity);

}

fclose(uR);

fclose(pT);

pT = fopen("temporary.txt", "w");

fclose(pT);

if(rentModifier != 5)

{

Sleep(2000);

bookPanel();

}

}

// lists all the book record from the book\_Records.txt file

void listBook(){

system("cls");

fflush(stdin);

char name[255], author[255], publisher[255];

double quantity, bookid;

int counter=0;

FILE \*uR = fopen("book\_Records.txt", "r");

printf("-------------------------------\n");

printf(">>> List of Books Record <<< \n");

printf("-------------------------------\n\n");

while(fscanf(uR, "%s %s %s %lf %lf \n", name, author, publisher, &bookid, &quantity) != EOF)

{

printf("-------------------------------\n");

printf("> Book Name: %s \n", name);

printf("> Auhtor: %s \n", author);

printf("> Publisher: %s\n", publisher);

printf("> Book ID: %.0lf \n", bookid);

printf("> Quantity: %.0lf \n", quantity);

printf("-------------------------------\n\n\n");

counter++;

}

fclose(uR);

if(counter == 0)

{

printf("-------------------------------------\n");

printf("There is no book records added yet...\n");

printf("--------------------------------------\n\n");

}

printf("Press any key to get back to Book Panel.\n");

getch();

bookPanel();

}

// Book Rental function

// first it checks whether the user is already registered in the user\_Records.txt or not

// second it checks whether the book exists in the book\_Records.txt or not

// third it checks if the book quantity is atleast 1, throws error if the book is OUT OF STOCK -- it also reduces book quantity by 1 from book\_Recprds.txt

// fourth it registers the user name & book name in a new file rent\_Records.txt and saves it :)

void rentBook(){

int terminator=1, nameFound, bookFound;

label5:

fflush(stdin);

// check if user exists

nameFound = searchUser(3);

if(nameFound != 5 && terminator != 4)

{

printf("Press any key to re-enter the name. \n");

getch();

(terminator == 3) ? bookPanel() : terminator++;

goto label5;

}

else if(nameFound == 5)

{

printf("\nUser Found in Records! \nPlease wait... \n");

terminator = 1;

Sleep(2000);

}

label6:

fflush(stdin);

// check if book exists

bookFound = searchBook(3);

if(bookFound != 5 && terminator != 4)

{

printf("Press any key to re-enter the name. \n");

getch();

(terminator == 3) ? bookPanel() : terminator++;

goto label6;

}

else if(bookFound == 5)

{

// check if book quantity is > 0

if(bookStock > 0)

{

printf("\nBook Found & In-Stock! \nPlease wait... \n");

}

else

{

printf("\nSorry, Out of Stock! \nPlease wait... ");

Sleep(2000);

(terminator == 3) ? bookPanel() : terminator++;

goto label6;

}

}

fflush(stdin);

/\* printf("\n---------------------------------------------\n");

printf("User Searcher: %d \n", nameFound);

printf("Book Searcher: %d \n", bookFound);

printf("User Name: %s \n", rentName);

printf("Book Name: %s \n", bookName);

printf("---------------------------------------------\n\n"); \*/

if(nameFound == 5 && bookFound == 5)

{

// Adding record in rent\_Records.txt file

FILE \*uR = fopen("rent\_Records.txt", "ab+");

if(uR != NULL)

{

fprintf(uR, "%s %s \n", rentName, bookName);

}

else

{

printf("Unable to open/locate the file.");

}

fclose(uR);

// reducing quantity of book by 1

bookStock--;

modifyBook(5);

printf("---------------------------------------------\n");

printf(">>> Rent Record Added Successfully <<< \n");

printf("---------------------------------------------\n");

printf("\nRedirecting to Book Panel...\n");

Sleep(3500);

bookPanel();

}

}

// lists all the username & booknames which are rented to someone in registered files

void rentList(){

system("cls");

fflush(stdin);

char rentListUser[255], rentListBook[255];

int counter = 0;

FILE \*uR = fopen("rent\_Records.txt", "r");

printf("-------------------------------\n");

printf(">>> List of Books Rental <<< \n");

printf("-------------------------------\n\n");

while(fscanf(uR, "%s %s \n", rentListUser, rentListBook) != EOF)

{

printf("> Rent User: %s \n", rentListUser);

printf("> Rent Book(s): %s \n\n", rentListBook);

counter++;

}

fclose(uR);

fflush(stdin);

if(counter == 0)

{

printf("-------------------------------------\n");

printf("There is no rent records added yet...\n");

printf("--------------------------------------\n\n");

}

printf("\nPress any key to get back to User Panel.\n");

getch();

userPanel();

}

// searches for the book details by book name from the txt file

int searchBook(int bookSearcher){

label4:

system("cls");

fflush(stdin);

char name[255], author[255], publisher[255];

double bookid, quantity;

int flag=0;

int compare;

char find[255];

(bookSearcher != 3) ? printf("Serach the book by Author name or Book name: ") : printf("Serach the book by Author name or Book name: ");

gets(find);

FILE \*uR = fopen("book\_Records.txt", "r");

while(fscanf(uR, "%s %s %s %lf %lf \n", name, author, publisher, &bookid, &quantity) != EOF)

{

compare = strcmp(find, name);

compare = strcmp(find, author);

if(compare == 0)

{

if(bookSearcher != 3)

{

printf("\n-------------------------\n");

printf(">>> Record Found <<< \n");

printf("-------------------------\n\n");

printf("-------------------------------\n");

printf("> Book Name: %s \n", name);

printf("> Auhtor: %s \n", author);

printf("> Publisher: %s\n", publisher);

printf("> Book ID: %.0lf \n", bookid);

printf("> Quantity: %.0lf \n", quantity);

printf("-------------------------------\n\n");

}

strcpy(bookName, name);

bookStock = quantity;

flag=1;

}

}

fclose(uR);

fflush(stdin);

if(flag == 0)

{

printf("\n>>> Record Not Found <<< \n\n");

}

if(bookSearcher != 3)

{

printf("Press any key to redirect to Book Panel.");

getch();

bookPanel();

}

else if(bookSearcher == 3 && flag == 1)

{

return 5;

}

}

// deletes the book records from .txt file

void deleteBook(){

system("cls");

fflush(stdin);

char name[255], author[255], publisher[255];

double bookid, quantity;

char name1[255], author1[255], publisher1[255];

double bookid1, quantity1;

int flag=0;

int compare;

char find[255];

printf("Enter the name of the book you want to delete the detail: ");

gets(find);

fflush(stdin);

FILE \*uR = fopen("book\_Records.txt", "r");

FILE \*pT = fopen("temporary.txt", "a");

while(fscanf(uR, "%s %s %s %lf %lf \n", name, author, publisher, &bookid, &quantity) != EOF)

{

compare = strcmp(find, name);

if(compare == 0)

{

printf("\n---------------------------------------------\n");

printf(">>> Record Deleted <<<\n");

printf("-----------------------------------------------\n\n");

printf("Redirecting to Book Panel...");

flag = 1;

}

else

{

fprintf(pT, "%s %s %s %.0lf %.0lf \n", name, author, publisher, bookid, quantity);

}

}

fclose(uR);

fclose(pT);

fflush(stdin);

uR = fopen("book\_Records.txt", "w");

fclose(uR);

if(flag == 0)

{

printf("\n\n-------------------------------\n");

printf(">>> Record Not Found <<<\n");

printf("-------------------------------\n\n");

printf("Redirecting to Book Panel...");

}

uR = fopen("book\_Records.txt", "a");

pT = fopen("temporary.txt", "r");

while(fscanf(pT, "%s %s %s %lf %lf \n", name, author, publisher, &bookid, &quantity) != EOF)

{

fprintf(uR, "%s %s %s %.0lf %.0lf \n", name, author, publisher, bookid, quantity);

}

fclose(uR);

fclose(pT);

pT = fopen("temporary.txt", "w");

fclose(pT);

Sleep(2000);

bookPanel();

}