A Group Project Report on

**Library management system**

Submitted to

The Department of Computer Science and Engineering

By

1. VELUGURI SRIJA (22311A6217)
2. PITTALA PRAVALIKA (22311A6222)
3. KARNE SHYAM SUNDHAR (22311A6231)
4. DEVANDLA VISHNU TEJA (22311A6240)
5. BURAM RAMAKANTH (22311A6250)
6. BHUPELLI TEJASWINI (22311A6256)

Under the Guidance of

Prof. Abdul Khalandar



**Sreenidhi Institute of Science and Technology**

***(An Autonomous Institution)***

Yamnampet, Ghatkesar, R.R. District, Hyderabad - 501301

**CONTENTS**

**S.NO TOPIC NAME PAGE NO.**

1. Introduction 03

2. Source Code 05

3. Screen Shots (Output Screen) 17

Introduction:

A library management system is software that is designed to manage all the functions of a library. It helps librarian to maintain the database of new books and the books that are borrowed by members along with their due dates.

This system completely automates all your library’s activities. The best way to maintain, organize, and handle countless books systematically is to implement a library management system software.

A library management system is used to maintain library records. It tracks the records of the number of books in the library, how many books are issued, or how many books have been returned or renewed or late fine charges, etc.

You can find books in an instant, issue/reissue books quickly, and manage all the data efficiently and orderly using this system. The purpose of a library management system is to provide instant and accurate data regarding any type of book, thereby saving a lot of time and effort.

The purpose of a library management system is to operate a library with efficiency and at reduced costs. The system being entirely automated streamlines all the tasks involved in operations of the library. The activities of book purchasing, cataloging, indexing, circulation recording and stock checking are done by the software. Such software eliminates the need for repetitive manual work and minimizes the chances of errors.  
  
The library management system software helps in reducing operational costs. Managing a library manually is labor intensive and an immense amount of paperwork is involved. An automated system reduces the need for manpower and stationery. This leads to lower operational costs.  
  
The system saves time for both the user and the librarian. With just a click the user can search for the books available in the library. The librarian can answer queries with ease regarding the availability of books. Adding, removing or editing the database is a simple process. Adding new members or cancelling existing memberships can be done with ease.  
  
Stock checking and verification of books in the library can be done within a few hours. The automated system saves a considerable amount of time as opposed system.  
  
The library management system software makes the library a smart one by organizing the books systematically by author, title and subject. This enables users to search for books quickly and effortlessly.  
  
Students need access to authentic information. An advanced organized library is an integral part of any educational institution. In this digital age a web based library management system would be ideal for students who can access the library’s database on their smartphones.

Source Code:

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include<string.h>

struct books {

int id;

char \*bookName;

char \*authorName;

char \*date;

};

struct student {

int id;

char \*sName;

char \*sClass;

int sRoll;

char \*bookName;

char \*date;

};

FILE \*fp;

void addBook();

void booksList();

void del();

void issueBook();

void issueList();

int main() {

int ch;

while (1) {

system("cls");

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

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

printf("2. Books List\n");

printf("3. Remove Book\n");

printf("4. Issue Book\n");

printf("5. Issued Book List\n");

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

printf("Enter your choice: ");

scanf("%d", &ch);

switch (ch) {

case 0:

exit(0);

case 1:

addBook();

break;

case 2:

booksList();

break;

case 3:

del();

break;

case 4:

issueBook();

break;

case 5:

issueList();

break;

default:

printf("Invalid Choice...\n\n");

}

printf("Press Any Key To Continue...");

getch();

}

return 0;

}

void addBook() {

char myDate[12];

time\_t t = time(NULL);

struct tm tm = \*localtime(&t);

sprintf(myDate, "%02d/%02d/%d", tm.tm\_mday, tm.tm\_mon + 1, tm.tm\_year + 1900);

struct books b;

b.date = malloc(sizeof(char) \* 12);

strcpy(b.date, myDate);

fp = fopen("books.txt", "ab");

printf("Enter book id: ");

scanf("%d", &b.id);

printf("Enter book name: ");

b.bookName = malloc(sizeof(char) \* 50);

fflush(stdin);

gets(b.bookName);

printf("Enter author name: ");

b.authorName = malloc(sizeof(char) \* 50);

fflush(stdin);

gets(b.authorName);

printf("Book Added Successfully\n");

fwrite(&b, sizeof(struct books), 1, fp);

fclose(fp);

free(b.date);

free(b.bookName);

free(b.authorName);

}

void booksList() {

system("cls");

printf("<== Available Books ==>\n\n");

printf("%-10s %-30s %-20s %s\n\n", "Book id", "Book Name", "Author", "Date");

struct books b;

b.date = malloc(sizeof(char) \* 12);

b.bookName = malloc(sizeof(char) \* 50);

b.authorName = malloc(sizeof(char) \* 50);

fp = fopen("books.txt", "rb");

while (fread(&b, sizeof(struct books), 1, fp) == 1) {

printf("%-10d %-30s %-20s %s\n", b.id, b.bookName, b.authorName, b.date);

}

fclose(fp);

free(b.date);

free(b.bookName);

free(b.authorName);

}

void del() {

int id, f = 0;

system("cls");

printf("<== Remove Books ==>\n\n");

printf("Enter Book id to remove: ");

scanf("%d", &id);

FILE \*ft;

struct books b;

b.date = malloc(sizeof(char) \* 12);

b.bookName = malloc(sizeof(char) \* 50);

b.authorName = malloc(sizeof(char) \* 50);

fp = fopen("books.txt", "rb");

ft = fopen("temp.txt", "wb");

while (fread(&b, sizeof(struct books), 1, fp) == 1) {

if (id == b.id) {

f = 1;

} else {

fwrite(&b, sizeof(struct books), 1, ft);

}

}

if (f == 1) {

printf("\n\nDeleted Successfully.");

} else {

printf("\n\nRecord Not Found !");

}

fclose(fp);

fclose(ft);

remove("books.txt");

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

free(b.date);

free(b.bookName);

free(b.authorName);

}void issueBook() {

struct student \*s = malloc(sizeof(struct student));

char myDate[12];

time\_t t = time(NULL);

struct tm \*tm = localtime(&t);

sprintf(myDate, "%02d/%02d/%d", tm->tm\_mday, tm->tm\_mon + 1, tm->tm\_year + 1900);

s->date = malloc(sizeof(char) \* 12);

strcpy(s->date, myDate);

int f = 0;

system("clear");

printf("<== Issue Books ==>\n\n");

printf("Enter Book id to issue: ");

scanf("%d", &(s->id));

// Check if we have the book of given id

struct books b;

fp = fopen("books.txt", "rb");

while (fread(&b, sizeof(struct books), 1, fp) == 1) {

if (b.id == s->id) {

s->bookName = malloc(sizeof(char) \* 50);

strcpy(s->bookName, b.bookName);

f = 1;

break;

}

}

fclose(fp);

if (f == 0) {

printf("No book found with this id\n");

printf("Please try again...\n\n");

free(s->date);

free(s);

return;

}

fp = fopen("issue.txt", "ab");

printf("Enter Student Name: ");

fflush(stdin);

s->sName = malloc(sizeof(char) \* 50);

fgets(s->sName, 50, stdin);

s->sName[strcspn(s->sName, "\n")] = '\0';

printf("Enter Student Class: ");

fflush(stdin);

s->sClass = malloc(sizeof(char) \* 50);

fgets(s->sClass, 50, stdin);

s->sClass[strcspn(s->sClass, "\n")] = '\0';

printf("Enter Student Roll: ");

scanf("%d", &(s->sRoll));

printf("Book Issued Successfully\n\n");

fwrite(s, sizeof(struct student), 1, fp);

fclose(fp);

free(s->date);

free(s->bookName);

free(s->sName);

free(s->sClass);

free(s);

}

void issueList() {

system("clear");

printf("<== Book Issue List ==>\n\n");

printf("%-10s %-30s %-20s %-10s %-30s %s\n\n", "S.id", "Name", "Class", "Roll", "Book Name", "Date");

struct student s;

fp = fopen("issue.txt", "rb");

while (fread(&s, sizeof(struct student), 1, fp) == 1) {

printf("%-10d %-30s %-20s %-10d %-30s %s\n", s.id, s.sName, s.sClass, s.sRoll, s.bookName, s.date);

}

fclose(fp);

}

Screen Shots (Output Screen):













