#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

#define MAX\_ENTRIES 100

// Structure to store a record

typedef struct {

int id;

char name[50];

int age;

char occupation[50];

} Record;

Record database[MAX\_ENTRIES];

int recordCount = 0;

// Function prototypes

void addRecord();

void displayRecords();

void searchRecordById();

void deleteRecordById();

void updateRecordById();

void sortRecordsByName();

void displayStatistics();

void exportRecordsToFile();

void importRecordsFromFile();

void displayMenu();

int main() {

int choice;

while (1) {

displayMenu();

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

addRecord();

break;

case 2:

displayRecords();

break;

case 3:

searchRecordById();

break;

case 4:

deleteRecordById();

break;

case 5:

updateRecordById();

break;

case 6:

sortRecordsByName();

break;

case 7:

displayStatistics();

break;

case 8:

exportRecordsToFile();

break;

case 9:

importRecordsFromFile();

break;

case 10:

printf("Exiting program. Goodbye!\n");

exit(0);

default:

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

}

}

return 0;

}

void displayMenu() {

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

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

printf("2. Display Records\n");

printf("3. Search Record by ID\n");

printf("4. Delete Record by ID\n");

printf("5. Update Record by ID\n");

printf("6. Sort Records by Name\n");

printf("7. Display Statistics\n");

printf("8. Export Records to File\n");

printf("9. Import Records from File\n");

printf("10. Exit\n");

}

void addRecord() {

if (recordCount >= MAX\_ENTRIES) {

printf("Database is full. Cannot add more records.\n");

return;

}

Record newRecord;

newRecord.id = recordCount + 1;

printf("Enter name: ");

scanf("%s", newRecord.name);

printf("Enter age: ");

scanf("%d", &newRecord.age);

printf("Enter occupation: ");

scanf("%s", newRecord.occupation);

database[recordCount++] = newRecord;

printf("Record added successfully!\n");

}

void displayRecords() {

if (recordCount == 0) {

printf("No records to display.\n");

return;

}

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

for (int i = 0; i < recordCount; i++) {

printf("ID: %d\n", database[i].id);

printf("Name: %s\n", database[i].name);

printf("Age: %d\n", database[i].age);

printf("Occupation: %s\n", database[i].occupation);

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

}

}

void searchRecordById() {

int id;

printf("Enter ID to search: ");

scanf("%d", &id);

for (int i = 0; i < recordCount; i++) {

if (database[i].id == id) {

printf("Record found:\n");

printf("ID: %d\n", database[i].id);

printf("Name: %s\n", database[i].name);

printf("Age: %d\n", database[i].age);

printf("Occupation: %s\n", database[i].occupation);

return;

}

}

printf("Record with ID %d not found.\n", id);

}

void deleteRecordById() {

int id;

printf("Enter ID to delete: ");

scanf("%d", &id);

for (int i = 0; i < recordCount; i++) {

if (database[i].id == id) {

for (int j = i; j < recordCount - 1; j++) {

database[j] = database[j + 1];

}

recordCount--;

printf("Record with ID %d deleted successfully.\n", id);

return;

}

}

printf("Record with ID %d not found.\n", id);

}

void updateRecordById() {

int id;

printf("Enter ID to update: ");

scanf("%d", &id);

for (int i = 0; i < recordCount; i++) {

if (database[i].id == id) {

printf("Enter new name: ");

scanf("%s", database[i].name);

printf("Enter new age: ");

scanf("%d", &database[i].age);

printf("Enter new occupation: ");

scanf("%s", database[i].occupation);

printf("Record with ID %d updated successfully.\n", id);

return;

}

}

printf("Record with ID %d not found.\n", id);

}

void sortRecordsByName() {

if (recordCount < 2) {

printf("Not enough records to sort.\n");

return;

}

for (int i = 0; i < recordCount - 1; i++) {

for (int j = 0; j < recordCount - i - 1; j++) {

if (strcmp(database[j].name, database[j + 1].name) > 0) {

Record temp = database[j];

database[j] = database[j + 1];

database[j + 1] = temp;

}

}

}

printf("Records sorted by name successfully.\n");

}

void displayStatistics() {

if (recordCount == 0) {

printf("No records available for statistics.\n");

return;

}

int totalAge = 0;

for (int i = 0; i < recordCount; i++) {

totalAge += database[i].age;

}

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

printf("Total Records: %d\n", recordCount);

printf("Average Age: %.2f\n", (double)totalAge / recordCount);

}

void exportRecordsToFile() {

FILE \*file = fopen("records.txt", "w");

if (file == NULL) {

printf("Error opening file for writing.\n");

return;

}

for (int i = 0; i < recordCount; i++) {

fprintf(file, "%d,%s,%d,%s\n", database[i].id, database[i].name, database[i].age, database[i].occupation);

}

fclose(file);

printf("Records exported successfully to records.txt\n");

}

void importRecordsFromFile() {

FILE \*file = fopen("records.txt", "r");

if (file == NULL) {

printf("Error opening file for reading.\n");

return;

}

recordCount = 0;

while (fscanf(file, "%d,%49[^,],%d,%49[^\n]\n", &database[recordCount].id, database[recordCount].name, &database[recordCount].age, database[recordCount].occupation) == 4) {

recordCount++;

if (recordCount >= MAX\_ENTRIES) {

printf("Database limit reached while importing.\n");

break;

}

}

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_BOOKS 100

#define MAX\_MEMBERS 50

// Structures

typedef struct {

int id;

char title[100];

char author[100];

int year;

} Book;

typedef struct {

int id;

char name[100];

int age;

char membershipType[20];

} Member;

// Global Variables

Book books[MAX\_BOOKS];

Member members[MAX\_MEMBERS];

int bookCount = 0;

int memberCount = 0;

// Function Prototypes

void addBook();

void viewBooks();

void searchBookByTitle();

void addMember();

void viewMembers();

void displayMenu();

// Main Function

int main() {

int choice;

while (1) {

displayMenu();

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

addBook();

break;

case 2:

viewBooks();

break;

case 3:

searchBookByTitle();

break;

case 4:

addMember();

break;

case 5:

viewMembers();

break;

case 6:

printf("Exiting program. Goodbye!\n");

exit(0);

default:

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

}

}

return 0;

}

// Display Menu

void displayMenu() {

printf("\n===== LIBRARY MANAGEMENT MENU =====\n");

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

printf("2. View All Books\n");

printf("3. Search Book by Title\n");

printf("4. Add Member\n");

printf("5. View All Members\n");

printf("6. Exit\n");

}

// Add Book

void addBook() {

if (bookCount >= MAX\_BOOKS) {

printf("Book limit reached. Cannot add more books.\n");

return;

}

Book newBook;

newBook.id = bookCount + 1;

printf("Enter book title: ");

getchar(); // Clear newline

fgets(newBook.title, 100, stdin);

newBook.title[strcspn(newBook.title, "\n")] = '\0'; // Remove trailing newline

printf("Enter book author: ");

fgets(newBook.author, 100, stdin);

newBook.author[strcspn(newBook.author, "\n")] = '\0';

printf("Enter publication year: ");

scanf("%d", &newBook.year);

books[bookCount++] = newBook;

printf("Book added successfully!\n");

}

// View All Books

void viewBooks() {

if (bookCount == 0) {

printf("No books in the library.\n");

return;

}

printf("\n===== BOOK LIST =====\n");

for (int i = 0; i < bookCount; i++) {

printf("ID: %d\n", books[i].id);

printf("Title: %s\n", books[i].title);

printf("Author: %s\n", books[i].author);

printf("Year: %d\n", books[i].year);

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

}

}

// Search Book by Title

void searchBookByTitle() {

char title[100];

printf("Enter book title to search: ");

getchar(); // Clear newline

fgets(title, 100, stdin);

title[strcspn(title, "\n")] = '\0'; // Remove trailing newline

for (int i = 0; i < bookCount; i++) {

if (strcasecmp(books[i].title, title) == 0) {

printf("Book Found:\n");

printf("ID: %d\n", books[i].id);

printf("Title: %s\n", books[i].title);

printf("Author: %s\n", books[i].author);

printf("Year: %d\n", books[i].year);

return;

}

}

printf("Book not found.\n");

}

// Add Member

void addMember() {

if (memberCount >= MAX\_MEMBERS) {

printf("Member limit reached. Cannot add more members.\n");

return;

}

Member newMember;

newMember.id = memberCount + 1;

printf("Enter member name: ");

getchar(); // Clear newline

fgets(newMember.name, 100, stdin);

newMember.name[strcspn(newMember.name, "\n")] = '\0';

printf("Enter member age: ");

scanf("%d", &newMember.age);

printf("Enter membership type (e.g., Regular, Premium): ");

getchar(); // Clear newline

fgets(newMember.membershipType, 20, stdin);

newMember.membershipType[strcspn(newMember.membershipType, "\n")] = '\0';

members[memberCount++] = newMember;

printf("Member added successfully!\n");

}

// View All Members

void viewMembers() {

if (memberCount == 0) {

printf("No members in the library.\n");

return;

}

printf("\n===== MEMBER LIST =====\n");

for (int i = 0; i < memberCount; i++) {

printf("ID: %d\n", members[i].id);

printf("Name: %s\n", members[i].name);

printf("Age: %d\n", members[i].age);

printf("Membership Type: %s\n", members[i].membershipType);

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

}

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_BOOKS 100

#define MAX\_MEMBERS 50

// Structures

typedef struct {

int id;

char title[100];

char author[100];

int year;

} Book;

typedef struct {

int id;

char name[100];

int age;

char membershipType[20];

} Member;

// Global Variables

Book books[MAX\_BOOKS];

Member members[MAX\_MEMBERS];

int bookCount = 0;

int memberCount = 0;

// Function Prototypes

void addBook();

void viewBooks();

void searchBookByTitle();

void addMember();

void viewMembers();

void displayMenu();

// Main Function

int main() {

int choice;

while (1) {

displayMenu();

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

addBook();

break;

case 2:

viewBooks();

break;

case 3:

searchBookByTitle();

break;

case 4:

addMember();

break;

case 5:

viewMembers();

break;

case 6:

printf("Exiting program. Goodbye!\n");

exit(0);

default:

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

}

}

return 0;

}

// Display Menu

void displayMenu() {

printf("\n===== LIBRARY MANAGEMENT MENU =====\n");

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

printf("2. View All Books\n");

printf("3. Search Book by Title\n");

printf("4. Add Member\n");

printf("5. View All Members\n");

printf("6. Exit\n");

}

// Add Book

void addBook() {

if (bookCount >= MAX\_BOOKS) {

printf("Book limit reached. Cannot add more books.\n");

return;

}

Book newBook;

newBook.id = bookCount + 1;

printf("Enter book title: ");

getchar(); // Clear newline

fgets(newBook.title, 100, stdin);

newBook.title[strcspn(newBook.title, "\n")] = '\0'; // Remove trailing newline

printf("Enter book author: ");

fgets(newBook.author, 100, stdin);

newBook.author[strcspn(newBook.author, "\n")] = '\0';

printf("Enter publication year: ");

scanf("%d", &newBook.year);

books[bookCount++] = newBook;

printf("Book added successfully!\n");

}

// View All Books

void viewBooks() {

if (bookCount == 0) {

printf("No books in the library.\n");

return;

}

printf("\n===== BOOK LIST =====\n");

for (int i = 0; i < bookCount; i++) {

printf("ID: %d\n", books[i].id);

printf("Title: %s\n", books[i].title);

printf("Author: %s\n", books[i].author);

printf("Year: %d\n", books[i].year);

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

}

}

// Search Book by Title

void searchBookByTitle() {

char title[100];

printf("Enter book title to search: ");

getchar(); // Clear newline

fgets(title, 100, stdin);

title[strcspn(title, "\n")] = '\0'; // Remove trailing newline

for (int i = 0; i < bookCount; i++) {

if (strcasecmp(books[i].title, title) == 0) {

printf("Book Found:\n");

printf("ID: %d\n", books[i].id);

printf("Title: %s\n", books[i].title);

printf("Author: %s\n", books[i].author);

printf("Year: %d\n", books[i].year);

return;

}

}

printf("Book not found.\n");

}

// Add Member

void addMember() {

if (memberCount >= MAX\_MEMBERS) {

printf("Member limit reached. Cannot add more members.\n");

return;

}

Member newMember;

newMember.id = memberCount + 1;

printf("Enter member name: ");

getchar(); // Clear newline

fgets(newMember.name, 100, stdin);

newMember.name[strcspn(newMember.name, "\n")] = '\0';

printf("Enter member age: ");

scanf("%d", &newMember.age);

printf("Enter membership type (e.g., Regular, Premium): ");

getchar(); // Clear newline

fgets(newMember.membershipType, 20, stdin);

newMember.membershipType[strcspn(newMember.membershipType, "\n")] = '\0';

members[memberCount++] = newMember;

printf("Member added successfully!\n");

}

// View All Members

void viewMembers() {

if (memberCount == 0) {

printf("No members in the library.\n");

return;

}

printf("\n===== MEMBER LIST =====\n");

for (int i = 0; i < memberCount; i++) {

printf("ID: %d\n", members[i].id);

printf("Name: %s\n", members[i].name);

printf("Age: %d\n", members[i].age);

printf("Membership Type: %s\n", members[i].membershipType);

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

}

}