Library Management System in C

Abstract

This program is a simple Library Management System implemented in C. It consists of structs and functions
that enable the addition, removal, and display of books. The program presents a menu to the user, allowing them
to perform these actions. The Library struct maintains a dynamic collection of books and provides functions
for managing them. The program offers basic functionality for managing books in a library.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct {
    char title[100];
    char author[100];
} Book;

typedef struct {
    Book* books;
    int count;
    int capacity;
```

```
} Library;

void initLibrary(Library* library) {
    library->capacity = 10;
    library->count = 0;
    library->books = (Book*)malloc(library->capacity * sizeof(Book));
}

void addBook(Library* library, const char* title, const char* author) {
    if (library->count == library->capacity) {
        library->capacity *= 2;
         library->books = (Book*)realloc(library->books, library->capacity
* sizeof(Book));
    }
    strcpy(library->books[library->count].title, title);
    strcpy(library->books[library->count].author, author);
    library->count++;
    printf("Book added successfully!\n");
}

void removeBook(Library* library, const char* title) {
    int found = 0;
    for (int i = 0; i < library->count; i++) {
        if (strcmp(library->books[i].title, title) == 0) {
            found = 1;
            for (int j = i; j < library->count - 1; j++) {
                library->books[j] = library->books[j + 1];
            }
            library->count--;
            printf("Book removed successfully!\n");
            break;
        }
    }
    if (!found) {
        printf("Book not found in the library.\n");
    }
}

void displayBooks(Library* library) {
    if (library->count == 0) {
        printf("No books in the library.\n");
    } else {
        printf("Books in the library:\n");
        for (int i = 0; i < library->count; i++) {
            printf("Title: %s\n", library->books[i].title);
            printf("Author: %s\n", library->books[i].author);
        }
    }
}

void cleanupLibrary(Library* library) {
    free(library->books);
```

```
}


int main() {
    Library library;
    initLibrary(&library);
    int choice;
    char title[100], author[100];

    while (1) {
        printf("\nLibrary Management System\n");
        printf("1. Add Book\n");
        printf("2. Remove Book\n");
        printf("3. Display Books\n");
        printf("4. Exit\n");
        printf("Enter your choice (1-4): ");
        scanf("%d", &choice);
        getchar();  // to consume the newline character after scanf

        switch (choice) {
            case 1:
                printf("Enter book title: ");
                fgets(title, sizeof(title), stdin);
                     title[strcspn(title, "\n")] = 0;  // remove newline character
                printf("Enter author name: ");
                fgets(author, sizeof(author), stdin);
```

```
                    author[strcspn(author, "\n")] = 0;  // remove newline
character
                addBook(&library, title, author);
                break;
            case 2:
                if (library.count == 0) {
                    printf("No books in the library.\n");
                } else {
                    printf("Enter book title to remove: ");
                    fgets(title, sizeof(title), stdin);
                      title[strcspn(title, "\n")] = 0;  // remove newline
character
                    removeBook(&library, title);
                }
                break;
            case 3:
                displayBooks(&library);
                break;
            case 4:
                printf("Pleasure helping you..! This is Vighnesh, over and
out\n");
                cleanupLibrary(&library);
                return 0;
            default:
                  printf("Invalid choice! Please enter a number from 1 to
4.\n");
```

```
        }
    }
}
```