

CODE: MICROPROJECT on MUSIC LIBRARY MANAGEMENT

Submitted by: Evlin Sara Johny
S3 CSE B

```
Activities Text Editor Wed 22:58 vk.c ~/evlin Save
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

struct Song {
    int id;
    char title[100];
    char artist[50];
    char album[50];
    char genre[20];
};

void add_song() {
    struct Song song;
    printf("Enter song ID: ");
    scanf("%d", &song.id);
    printf("Enter song title: ");
    scanf("%[^\n]s", song.title);
    printf("Enter artist: ");
    scanf("%[^\n]s", song.artist);
    printf("Enter album: ");
    scanf("%[^\n]s", song.album);
    printf("Enter genre: ");
    scanf("%s", song.genre);

    FILE *file = fopen("songs.txt", "a");
    if (file == NULL) {
        printf("Error opening file\n");
        return;
    }
    printf(file, "%d %s %s %s\n", song.id, song.title, song.artist, song.album, song.genre);
    fclose(file);
    printf("Song added successfully.\n");
}

void manage_song_details() {
    struct Song song;
    printf("Enter song ID to update: ");
    scanf("%d", &song.id);
    printf("Enter new song title: ");
    scanf("%[^\n]s", song.title);
    printf("Enter new artist: ");
    scanf("%[^\n]s", song.artist);
    printf("Enter new album: ");
    scanf("%[^\n]s", song.album);
    printf("Enter new genre: ");
    scanf("%s", song.genre);

    FILE *file = fopen("songs_temp.txt", "w");
    FILE *original = fopen("songs.txt", "r");
    if (file == NULL || original == NULL) {
        printf("Error opening file\n");
        return;
    }
}
```

```
Activities Text Editor Wed 22:58 vk.c ~/evlin Save
FILE *original = fopen("songs.txt", "r");
if (file == NULL || original == NULL) {
    printf("Error opening file\n");
    return;
}

struct Song temp;
while (fscanf(original, "%d %[^\n] %[^\n] %[^\n] %[^\n]s", &temp.id, temp.title, temp.artist, temp.album, temp.genre) != EOF) {
    if (temp.id == song.id) {
        fprintf(file, "%d %s %s %s\n", song.id, song.title, song.artist, song.album, song.genre);
    } else {
        fprintf(file, "%d %s %s %s\n", temp.id, temp.title, temp.artist, temp.album, temp.genre);
    }
}
fclose(file);
fclose(original);
remove("songs.txt");
rename("songs_temp.txt", "songs.txt");
printf("Song details updated successfully.\n");
}

void send_recommendations() {
    struct Song song;
    FILE *file = fopen("songs.txt", "r");
    if (file == NULL) {
        printf("Error opening file\n");
        return;
    }
    printf("Song Recommendations:\n");
    while (fscanf(file, "%d %[^\n] %[^\n] %[^\n] %[^\n]s", &song.id, song.title, song.artist, song.album, song.genre) != EOF) {
        printf("Recommendation: \"%s\" by %s from the album \"%s\" (Genre: %s)\n", song.title, song.artist, song.album, song.genre);
    }
    fclose(file);
}

void display_library() {
    struct Song song;
    FILE *file = fopen("songs.txt", "r");
    if (file == NULL) {
        printf("Error opening file\n");
        return;
    }
    printf("Music Library:\n");
    while (fscanf(file, "%d %[^\n] %[^\n] %[^\n] %[^\n]s", &song.id, song.title, song.artist, song.album, song.genre) != EOF) {
        printf("ID: %d, Title: \"%s\", Artist: %s, Album: %s, Genre: %s\n", song.id, song.title, song.artist, song.album, song.genre);
    }
    fclose(file);
}

int main() {
    int choice;
    while (1) {
        printf("----- Music Library Management System ----- \n");
        printf("1. Add Song\n");
        printf("2. Manage Song Details\n");
    }
}
```

The screenshot shows a Linux desktop environment with a text editor open, displaying a C program. The desktop background is dark blue. On the left side, there is a vertical dock with icons for Activities, Text Editor, Firefox, a file manager, a terminal, a web browser, a music player, a calendar, a clock, and a system monitor. The text editor window is titled "Text Editor" and shows a file named "vk.c". The code is a C program for a music library management system. It includes headers for `stdio.h` and `stdlib.h`. The program defines a `Song` struct with fields for id, title, artist, album, and genre. It implements functions for adding songs, managing song details, sending recommendations, and displaying the library. The main function uses a menu-driven interface to allow the user to interact with the library.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 struct Song {
5     int id;
6     char title[50];
7     char artist[50];
8     char album[50];
9     char genre[50];
10 };
11
12 void add_song(struct Song *song) {
13     printf("Add song details:\n");
14     scanf("%d %s %s %s %s", &song->id, song->title, song->artist, song->album, song->genre);
15 }
16
17 void manage_song_details(struct Song *song) {
18     printf("Manage song details:\n");
19     scanf("%d %s %s %s %s", &song->id, song->title, song->artist, song->album, song->genre);
20 }
21
22 void send_recommendations(struct Song *song) {
23     printf("Send recommendations:\n");
24     scanf("%d %s %s %s %s", &song->id, song->title, song->artist, song->album, song->genre);
25 }
26
27 void display_library(struct Song *song) {
28     printf("Display library:\n");
29     scanf("%d %s %s %s %s", &song->id, song->title, song->artist, song->album, song->genre);
30 }
31
32 int main() {
33     int choice;
34     while (1) {
35         printf("----- Music Library Management System ----- \n");
36         printf("1. Add Song\n");
37         printf("2. Manage Song details\n");
38         printf("3. Send Recommendations\n");
39         printf("4. Display Library\n");
40         printf("5. Exit\n");
41         scanf("%d", &choice);
42         switch (choice) {
43             case 1:
44                 add_song(song);
45                 break;
46             case 2:
47                 manage_song_details(song);
48                 break;
49             case 3:
50                 send_recommendations(song);
51                 break;
52             case 4:
53                 display_library(song);
54                 break;
55             case 5:
56                 exit(0);
57             default:
58                 printf("Invalid choice\n");
59         }
60     }
61     return 0;
62 }
```

```

Wed 22:58
en
user@user-VivoBook-ASUSLaptop-X415JA-X415JA: ~/evlin$

File Edit View Search Terminal Help
user@user-VivoBook-ASUSLaptop-X415JA-X415JA:~/evlin$ gcc vk.c
user@user-VivoBook-ASUSLaptop-X415JA-X415JA:~/evlin$ gcc vk.c
user@user-VivoBook-ASUSLaptop-X415JA-X415JA:~/evlin$ ./a.out
----- Music Library Management System -----
1. Add Song
2. Manage Song Details
3. Send Recommendations
4. Display Library
5. Exit
1
Enter song ID: 23
Enter song title: single
Enter artist: evlin
Enter album: ruin
Enter genre: pop
Song added successfully.
----- Music Library Management System -----
1. Add Song
2. Manage Song Details
3. Send Recommendations
4. Display Library
5. Exit
2
Enter song ID to update: 23
Enter new song title: game
Enter new artist: evlin
Enter new album: team
Enter new genre: classic
Song details updated successfully.
----- Music Library Management System -----
1. Add Song
2. Manage Song Details
3. Send Recommendations
4. Display Library
5. Exit
3
Song Recommendations:
Recommendation: "game evlin team classic" by ***** from the album "" (Genre: v)
----- Music Library Management System -----
1. Add Song
2. Manage Song Details
3. Send Recommendations
4. Display Library
5. Exit
4
Music Library:
ID: 23, Title: "game evlin team classic", Artist: *****, Album: , Genre: v
----- Music Library Management System -----
1. Add Song
2. Manage Song Details
3. Send Recommendations
4. Display Library
5. Exit
5
user@user-VivoBook-ASUSLaptop-X415JA-X415JA:~/evlin$

```