Real time problem

Problem statement : Managing a music library efficiently

Code:

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

#define MAX_SONGS 50
#define MAX_STRING_LENGTH 50

struct Song {
   char title[MAX_STRING_LENGTH];
   char artist[MAX_STRING_LENGTH];
   char genre[MAX_STRING_LENGTH];
};

void displayMenu() {
   printf("\n--- Music Library ---\n"
          "1. Add a song\n"
          "2. Search by title\n"
          "3. Search by artist\n"
          "4. Search by genre\n"
          "5. Display all songs\n"
          "6. Exit\n"
          "Enter your choice: ");
}

void addSong(struct Song library[], int* numSongs) {
   if (*numSongs < MAX_SONGS) {
      printf("\nEnter song details:\nTitle: ");
      scanf(" %[^\n]s", library[*numSongs].title);
      printf("Artist: ");
      scanf(" %[^\n]s", library[*numSongs].artist);
      printf("Genre: ");
      scanf(" %[^\n]s", library[*numSongs].genre);
      (*numSongs)++;
      printf("Song added successfully!\n");
   } else {
      printf("\nMusic library is full. Cannot add more songs.\n");
   }
```

```
}

void searchByField(const struct Song library[], int numSongs, const char* field) {
    char searchStr[MAX_STRING_LENGTH];
    printf("\nEnter the %s to search for: ", field);
    scanf(" %[^\n]s", searchStr);
    printf("\nSearch Results:\n");

    int found = 0;
    for (int i = 0; i < numSongs; i++) {
        if (strcmp(field, "title") == 0 && strstr(library[i].title, searchStr) != NULL ||
            strcmp(field, "artist") == 0 && strstr(library[i].artist, searchStr) != NULL ||
            strcmp(field, "genre") == 0 && strstr(library[i].genre, searchStr) != NULL) {
            printf("Title: %s\nArtist: %s\nGenre: %s\n\n", library[i].title, library[i].artist, library[i].genre);
            found = 1;
        }
    }

    if (!found) {
        printf("No matching songs found.\n");
    }
}

void displayAllSongs(const struct Song library[], int numSongs) {
    printf("\n--- All Songs ---\n");

    if (numSongs > 0) {
        for (int i = 0; i < numSongs; i++) {
            printf("Title: %s\nArtist: %s\nGenre: %s\n\n", library[i].title, library[i].artist, library[i].genre);
        }
    } else {
        printf("No songs in the library.\n");
    }
}

int main() {
    struct Song musicLibrary[MAX_SONGS];
    int numSongs = 0;
    int choice;

    do {
        displayMenu();
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                addSong(musicLibrary, &numSongs);
                break;
            case 2:
                searchByField(musicLibrary, numSongs, "title");
                break;
            case 3:
                searchByField(musicLibrary, numSongs, "artist");
                break;
            case 4:
                searchByField(musicLibrary, numSongs, "genre");
                break;
            case 5:
                displayAllSongs(musicLibrary, numSongs);
                break;
            case 6:
                printf("Exiting the Music Library.\n");
                break;
            default:
                printf("Invalid choice. Please try again.\n");
        }
    } while (choice != 6);

    return 0;
}
```

Output :

```
--- Music Library ---
1. Add a song
2. Search by title
3. Search by artist
4. Search by genre
5. Display all songs
6. Exit
Enter your choice: 1

Enter song details:
Title: Let Me Love You
Artist: Justin Bieber
Genre: love
Song added successfully!

--- Music Library ---
1. Add a song
2. Search by title
3. Search by artist
4. Search by genre
5. Display all songs
6. Exit
Enter your choice: 1

Enter song details:
Title: Sorry
Artist: Justin Bieber
Genre: depression
Song added successfully!

--- Music Library ---
1. Add a song
2. Search by title
3. Search by artist
4. Search by genre
5. Display all songs
6. Exit
Enter your choice: 1

Enter song details:
Title: Time
```

```
Artist: Hans Zimmer
Genre: instrumental
Song added successfully!

--- Music Library ---
1. Add a song
2. Search by title
3. Search by artist
4. Search by genre
5. Display all songs
6. Exit
Enter your choice: 5

--- All Songs ---
Title: Let Me Love You
Artist: Justin Bieber
Genre: love

Title: Sorry
Artist: Justin Bieber
Genre: depression

Title: Time
Artist: Hans Zimmer
Genre: instrumental


--- Music Library ---
1. Add a song
2. Search by title
3. Search by artist
4. Search by genre
5. Display all songs
6. Exit
Enter your choice: 6
Exiting the Music Library.
```