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

#define MAX_SONGS 100

// Structure to store song details
typedef struct {
    int id;
    char title[50];
    char artist[50];
    float duration; // in minutes
} Song;

// Global Variables
Song playlist[MAX_SONGS];
int songCount = 0;

// Function Prototypes
void addSong();
void displayPlaylist();
void deleteSong();
void searchSong();

int main() {
    int choice;
    while (1) {
        printf("\n=== Music Playlist Manager ===\n");
        printf("1. Add Song\n");
        printf("2. Display Playlist\n");
        printf("3. Delete Song\n");
        printf("4. Search Song\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1: addSong(); break;
            case 2: displayPlaylist(); break;
            case 3: deleteSong(); break;
            case 4: searchSong(); break;
            case 5: exit(0);
            default: printf("Invalid choice! Please try again.\n");
        }
    }
    return 0;
}

// Function to Add a Song to the Playlist
void addSong() {
    if (songCount >= MAX_SONGS) {
        printf("Playlist is full! Cannot add more songs.\n");
        return;
    }
    printf("Enter Song ID: ");
    scanf("%d", &playlist[songCount].id);
    getchar(); // Consume newline character
    printf("Enter Song Title: ");
    fgets(playlist[songCount].title, 50, stdin);
    playlist[songCount].title[strcspn(playlist[songCount].title, "\n")] = '\0'; // Remove newline
    printf("Enter Artist Name: ");
    fgets(playlist[songCount].artist, 50, stdin);
    playlist[songCount].artist[strcspn(playlist[songCount].artist, "\n")] = '\0'; // Remove
newline
    printf("Enter Duration (in minutes): ");
    scanf("%f", &playlist[songCount].duration);
    songCount++;
    printf("Song added successfully!\n");
}

// Function to Display the Playlist
void displayPlaylist() {
    if (songCount == 0) {
        printf("Playlist is empty!\n");
        return;
    }
    printf("\n=== Playlist ===\n");
    for (int i = 0; i < songCount; i++) {
        printf("ID: %d, Title: %s, Artist: %s, Duration: %.2f mins\n",
                playlist[i].id, playlist[i].title, playlist[i].artist, playlist[i].duration);
    }
}

// Function to Delete a Song from the Playlist
void deleteSong() {
    if (songCount == 0) {
        printf("Playlist is empty! No song to delete.\n");
        return;
    }
    int id;
    printf("Enter the Song ID to delete: ");
    scanf("%d", &id);

    int found = 0;
    for (int i = 0; i < songCount; i++) {
        if (playlist[i].id == id) {
            found = 1;
            for (int j = i; j < songCount - 1; j++) {
                playlist[j] = playlist[j + 1];
            }
            songCount--;
            printf("Song deleted successfully!\n");
            break;
        }
    }
    if (!found) {
        printf("Song with ID %d not found!\n", id);
    }
}

// Function to Search for a Song by Title
void searchSong() {
    if (songCount == 0) {
        printf("Playlist is empty!\n");
        return;
    }
    char searchTitle[50];
    getchar(); // Consume newline character
    printf("Enter the Song Title to search: ");
    fgets(searchTitle, 50, stdin);
    searchTitle[strcspn(searchTitle, "\n")] = '\0'; // Remove newline

    int found = 0;
    printf("\n=== Search Results ===\n");
    for (int i = 0; i < songCount; i++) {
        if (strstr(playlist[i].title, searchTitle) != NULL) {
            printf("ID: %d, Title: %s, Artist: %s, Duration: %.2f mins\n",
                   playlist[i].id, playlist[i].title, playlist[i].artist, playlist[i].duration);
            found = 1;
        }
    }
    if (!found) {
        printf("No song found with title containing \"%s\".\n", searchTitle);
    }
}
```