PSP [20ES104] COURSE PROJECT REPORT

On

“PROBLEM SOLVING WITH PROGRAMMING [R20]”

Developed By:

H.T.NO	STUDENT NAME
2203A51047	V.B.GOMATHI
2203A51060	S.PRATHIBHA
2203A51064	S.HARISH
2205A42011	SHRUJAN SINGH

Under the Guidance of

Mr. P PRAMOD KUMAR, M.Tech.(Ph.D)

Senior Assistant Professor

Submitted to

Department Computer Science and Artificial Intelligence

SR University

Ananthasagar(V), Hasanparthy(M), Hanamkonda(Dist.) – 506371

www.sru.edu.in

June 2023

Department of Computer Science and Artificial Intelligence

CERTIFICATE

This is to certify that the PSP course project report entitled **“ PROBLEM SOLVING WITH PROGRAMMING [R20]”** is a record of bonafide work carried out by the V.B. GOMATHI, S.PRATHIBHA, S.HARISH, SHRUJAN SINGH, bearing roll number 2203A51047, 2203 A51060, 2203 A51064 , 2205A42011 of Computer Science and Artificial Intelligence department during the academic year 2022-23.

Supervisor

(P Pramod Kumar)

INDEX

PROBLEM STATEMENT:

Develop a c project based on songs database management system that allows the user to search for a song by the title of the song,genre of song,artist,time period of the song (70s,80s,90s,20s) and also allow the user to add the songs and delete the song if the user want to from the database

MODULES:

In this application all variables and structure are declared globally so that these variables and structure members can be accessed throughout the program at any function call. We can choose any function by using function calls which are declared in switch-case. In order to repeat the loop control statement (do-while) is used with condition. The memory allocation will be done in this program dynamically.

In the given C program project, there are four modules used to implement different functionalities. Let's break down the applications of each module:

1. **Menu Module**: This module is used to display a menu to the user and handle user input. It is responsible for the overall control flow of the program, allowing the user to choose different options and invoking corresponding functions from other modules.

2. **Search Module**: This module is responsible for searching songs based on various criteria such as title, genre, artist, and time period. It interacts with the database module to retrieve the relevant songs and displays the search results to the user.

3. **Database Module**: This module handles the storage and retrieval of songs in the database. It provides functions to add a song, delete a song, and retrieve songs based on different criteria. The database module ensures the integrity of the database and performs operations on the song data.

4. **Validation Module**: This module validates user input to ensure it is correct and within the expected constraints. It is used to validate the time period entered by the user when adding a song to the database. It helps ensure that only valid time periods (70s, 80s, 90s, or 20s) are accepted.So, in total, there are four modules used in this C program project: Menu Module, Search Module, Database Module, and Validation Module. Each module serves a specific purpose and contributes to the overall functionality of the songs database management system.

KNOWLEDGE REQUIRED TO DEVELOP THIS APPLICATION

- Control Statements (if, if-else, switch)
- Loop Statements (while, for)
- Arrays (1D/2D-arrays)
- Strings (Strings and Table of strings) and its functions

  (strcpy, strcmp)
- Functions (Any type of user defined functions)
- Structure (structures and nested structures)

SOURCE CODE [HEADER FILE]:

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<windows.h>
#include<conio.h>
#include<math.h>
struct song {
   char title[100];
   char genre[50];
   char artist[50];
   char time_period[20];
}s[50];
int user_main(void);
int login(void);
int signin(void);
void read(void);
void search(void);
void print(void);
void sorting(struct song s[], int n);
void searching(struct song s[], int n);
void sort_title(struct song s[], int n);
void sort_genre(struct song s[], int n);
void sort_artist(struct song s[], int n);
void sort_tp(struct song s[], int n);
void search_by_title(struct song s[], int n);
void search_by_artist(struct song s[], int n);
void search_by_genre(struct song s[], int n);
void search_by_tp(struct song s[], int n);
void add();
void delete();
int choice;
FILE *fptr;
char a;
int count=0,ch,i,j,p1,x,n;
struct song *pas,*ptr;
```

```
int u,p,f=0,c=0,l_r,p_c;
char uname[20],pwd[20],ch1,ch2,rname[20],runame[20],pass[20],repass[20],profile[5][20],loginpro[5][20];
//int main()
void read(void)
{
printf("\n\n\n******WELCOME TO MUSIC*******\n\n\n");
    int i;
    printf("Enter the number of songs: ");
    scanf("%d", &n);

    for ( i = 0; i < n; i++) {
        printf("Enter the details of song %d:\n", i + 1);
        printf("Title: ");
        scanf("%s", s[i].title);
        printf("Genre: ");
        scanf("%s", s[i].genre);
        printf("Artist: ");
        scanf("%s", s[i].artist);
        printf("Time Period: ");
        scanf("%s", &s[i].time_period);
    }
}
void searching(struct song s[], int n)
{
    abc:
{
    printf("\nEnter your choice:\n");
    printf("1. Search by Song Title\n");
    printf("2. Search by Artist\n");
    printf("3. Search by genre\n");
    printf("4. Search by time period\n");
    scanf("%d", &choice);

    switch (choice) {
        case 1:
         search_by_title(s, n);
         break;
      case 2:
         search_by_artist(s, n);
         break;
      case 3:
         search_by_genre(s, n);
         break;
          case 4:
         search_by_tp(s, n);
         break;
          case 5:
                    add();
                    break;
          case 6:
                    //delete();
                    break;
      default:
         printf("Invalid Choice!\n");
         break;
   }
   char s;
   printf("\n\n WANT TO TRY AGAIN......?\n   PRESS 'Y' OR 'y' FOR YES\n  PRESS 'n' OR 'N' FOR NO\n");
   scanf("%c",&s);
   scanf("%c",&s);

   if(s=='y'||s=='Y')
   goto abc;
}
   //return 0;
}
void sorting(struct song s[], int n)
{
char a;
int ch;
```

```
abc:
    printf("\nEnter your choice:\n");
    printf("1. Sort by Song Title\n");
    printf("2. Sort by Artist\n");
    printf("3. Sort by genre\n");
    printf("4. Sort by time period\n");
    printf("5.add a song\n");
    printf("6.delete a song\n");
    scanf("%d", &ch);
    switch (ch) {
        case 1:
            sort_title(s, n);
            break;
        case 2:
            sort_artist(s, n);
            break;
        case 3:
            sort_genre(s, n);
            break;
             case 4:
            sort_tp(s, n);
            break;
             case 5:
                           add();
                           break;
             case 6:
                           delete();
                           break;
        default:
            printf("Invalid Choice!\n");
            break;
            printf("\n WANT TO TRY AGAIN ?......\n1.PRESS 'Y' or 'y' for yes \n PRESS 'n' or 'N'
for no\n");
            scanf("%c",&a);
            if(a=='y'||a=='Y')
            goto abc;
    }
}
void sort_title(struct song s[], int n) {
    for ( i = 0; i < n - 1; i++) {
        for (j = i + 1; j < n; j++) {
            if (strcmp(s[i].title, s[j].title) > 0) {
                struct song temp = s[i];
                s[i] = s[j];
                s[j] = temp;
            }
        }
    }

    printf("\nList of songs sorted by title:\n");
    for ( i = 0; i < n; i++) {
        printf("%d. %s - %s - %s - %s\n", i + 1, s[i].title, s[i].genre, s[i].artist, s[i].time_period);
    }
}
void sort_artist(struct song s[], int n) {
    for ( i = 0; i < n - 1; i++) {
        for (j = i + 1; j < n; j++) {
            if (strcmp(s[i].artist, s[j].artist) > 0) {
                struct song temp = s[i];
                s[i] = s[j];
                s[j] = temp;
            }
        }
    }

    printf("\nList of songs sorted by artist:\n");
    for ( i = 0; i < n; i++) {
        printf("%d. %s - %s - %s - %s\n", i + 1, s[i].title, s[i].genre, s[i].artist, s[i].time_period);
    }
}
void sort_genre(struct song s[], int n) {
    for ( i = 0; i < n - 1; i++) {
        for (j = i + 1; j < n; j++) {
            if (strcmp(s[i].genre, s[j].genre) > 0) {
                struct song temp = s[i];
                s[i] = s[j];
                s[j] = temp;
            }
        }
    }

    printf("\nList of songs sorted by genre:\n");
    for ( i = 0; i < n; i++) {
        printf("%d. %s - %s - %s - %s\n", i + 1, s[i].title, s[i].genre, s[i].artist, s[i].time_period);
    }
}
void sort_tp(struct song s[], int n) {
    for ( i = 0; i < n - 1; i++) {
        for (j = i + 1; j < n; j++) {
            if (strcmp(s[i].time_period, s[j].time_period) > 0) {
                struct song temp = s[i];
                s[i] = s[j];
                s[j] = temp;
            }
        }
    }

    printf("\nList of songs sorted by time period:\n");
    for ( i = 0; i < n; i++) {
        printf("%d. %s - %s - %s - %s\n", i + 1, s[i].title, s[i].genre, s[i].artist, s[i].time_period);
    }
}
void search_by_title(struct song s[], int n) {
    char title[100];
    printf("Enter the song title: ");
    scanf("%s", title);
    int flag = 0;
    for ( i = 0; i < n; i++) {
        if (strcmp(s[i].title, title) == 0) {
            printf("Song found!\n");
            printf("%s - %s - %s - %s\n", s[i].title, s[i].genre, s[i].artist, s[i].time_period);
            flag = 1;
            break;
        }
    }
    if (!flag) {
        printf("Song not found!\n");
    }
}
//int i;
void search_by_artist(struct song s[], int n) {
    char artist[50];
    printf("Enter the artist name: ");
    scanf("%s", artist);
    int flag = 0;
    for (i = 0; i < n; i++) {
        if (strcmp(s[i].artist, artist) == 0) {
            printf("Song found!\n");
            printf("%s - %s - %s - %s\n", s[i].title, s[i].genre, s[i].artist, s[i].time_period);
            flag=2;
            break;
        }
    }
    if(!flag)
    {
             printf("artist not found!\n");
}
}
void search_by_genre(struct song s[], int n) {
    char genre[50];
    printf("Enter the genre: ");
    scanf("%s", genre);
    int flag = 0;
    for (i = 0; i < n; i++) {
        if (strcmp(s[i].genre, genre) == 0) {
            printf("Song found!\n");
            printf("%s - %s - %s - %s\n", s[i].title, s[i].genre, s[i].artist, s[i].time_period);
            flag=2;
            break;
        }
    }
    if(!flag)
    {
              printf("genre not found!\n");
}
}
void search_by_tp(struct song s[], int n) {
    char tp[50];
    printf("Enter the time period: ");
    scanf("%s", tp);
    int flag = 0;
    for (i = 0; i < n; i++) {
        if (strcmp(s[i].time_period, tp) == 0) {
            printf("Song found!\n");
            printf("%s - %s - %s - %s\n", s[i].title, s[i].genre, s[i].artist, s[i].time_period);
            flag=2;
            break;
        }
    }
    if(!flag)
    {
              printf("time period not found!\n");
}
}
void add()
{
printf("\n ENTER THE DETAILS OF SONG YOU WANT TO ADD : ........\n\n");
        printf("Title: ");
        scanf("%s",s[n].title);
        printf("Genre: ");
        scanf("%s", s[n].genre);
        printf("Artist: ");
        scanf("%s", s[n].artist);
        printf("Time Period: ");
        scanf("%s", s[n].time_period);
    int i,j;
    for ( i = 0; i <n - 1; i++) {
        for (j = i + 1; j < n; j++) {
            if (strcmp(s[i].title, s[j].title) > 0) {
                struct song temp = s[i];
                s[i] = s[j];
                s[j] = temp;
            }
        }
    }

    printf("\nList of songs sorted by title after adding a song :\n");
    for ( i = 0; i <=n; i++) {
        printf("%d. %s - %s - %s - %s\n", i + 1, s[i].title, s[i].genre, s[i].artist, s[i].time_period);
    }
}
void delete()
{
char new_song[20],temp;
printf("\n ENTER THE SONG YOU WANT TO DELETE FROM THE LIST : ......");
scanf("%s",new_song);
for(i=0;i<n;i++)
{
              if(new_song==s[i].title)
              {
              for(j=i;j<n;j++)
              {
                            s[j].title[j]=s[j++].title[j++];
                            s[j].genre[j]=s[j++].genre[j++];
                            s[j].artist[j]=s[j++].artist[j++];
                            s[j].time_period[j]=s[j++].time_period[j++];
```

```
             }
             n--;
             i--;
}
int i,j;
    for ( i = 0; i <n - 2; i++) {
        for (j = i + 1; j < n-1; j++) {
            if (strcmp(s[i].title, s[j].title) > 0) {
                struct song temp = s[i];
                s[i] = s[j];
                s[j] = temp;
            }
        }
    }
}
 printf("\nList of songs sorted by title after deleting a song :\n");
    for ( i = 0; i < n-1; i++) {
        printf("%d. %s - %s - %s - %s\n", i + 1, s[i].title, s[i].genre, s[i].artist, s[i].time_period);
    }
}
int login(){
             login:
             f=0;
             do
{
printf("\n\n Login Form.....");
printf("\n\n Enter User Name : ");
scanf("%s",uname);
printf("\n \nEnter Password : ");
while((ch=getch())!='\r')
{
     pwd[f++]=ch;
     ch='*';
    printf("%c",ch);
}
pwd[f]='\0';
```

```
//scanf("%s",pwd);
u=strcmp(uname,profile[0]);
p=strcmp(pwd,profile[1]);
printf("\n\n Please wait Credential Verification is in Process....");
Sleep(5000);
system("cls");
if(u==0&&p==0)
{
        printf("\n music::..................SCR................\n");
        user_main();
        return(0);
}
else
{
        printf("\n\n Credentials are wrong.....please try again.....");
        c++;
        printf("\n\n Left %d attempts",3-c);
}
}while(c<3);
}
int signin(){
fptr=fopen("reserve.txt","w+");
printf("\n ------------MUSIC---------------");
printf("\n NAME \t\t:");
scanf("%s",rname);
printf("\n USER NAME(no special char's):");
scanf("%s",runame);
strcpy(profile[0],runame);
pw:
printf("\n PASS WORD \t:");
while((ch1=getch())!='\r')
{
   pass[f++]=ch1;
   ch1='*';
  printf("%c",ch1);
}
```

```
pass[f]='\0';
strcpy(profile[1],pass);
printf("\n PASS WORD (re enter as above)");
while((ch2=getch())!='\r')
{
    repass[f++]=ch2;
    ch2='*';
    printf("%c",ch2);
}
repass[f]='\0';
u=strcmp(pass,repass);
if(u!=0)
        {
                printf("\n\n storing the details...........");
                printf("\n\n REGISTRATION DONE SUCCESSFULL.........");
                for(a=0;a<5;a++)
                {
                        strcpy(profile[f],loginpro[f]);
                }
fprintf(fptr,"NAME=%s \n USER NAME=%d ,\n PASSWORD=%d",rname,runame,pass);// to not to understand the data by the others in .txt file i have stored my data in integer format
fclose(fptr);
                login();
                return(0);
        }
else
        {
        printf("\n enter the valid password ");
        goto pw;
        }
}

int user_main()
{
  //menu repetition
  printf("\n enter the no. of song detais you want to store:");
```

```
 scanf("%d",&n);
 ptr=(struct song *)malloc(n*sizeof(struct song));
 pas=ptr;
 do
 {
system("cls");
 printf("\n MENU:\n 1.READ DATA\n 2.SORTING\n 3.SEARCHING\n 4.PRINT DATA");
 printf("\n enter option:");
 scanf("%d",&ch);
 //switch case
 switch(ch)
 {
  case 1: abc:
                     read();//function call
                     count++;
  break;
  case 2:if (count==0)
                     {
                     printf("\nERROR:: to SORT the data first you have to READ the
data\n\n");
                     goto abc;
              }
              else sorting(s,n);//function call
  break;
  case 3:if(count==0)
                     {
                     printf("\nERROR:: to SEARCH the data first you have to READ the
data\n\n");
                     goto abc;
              }
               else searching(s,n);//function call
  break;
  default:printf("\n invalid option");
  break;
 }
 printf("\n enter y or Y to continue,enter n or N to exit:");
 scanf(" %c",&a);
```

```
  }while(a=='y'||a=='Y');
 system("cls");
}
```

SOURCE CODE [.C FILE]:

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include "musicheader.h"
//global variables
int g;
//main starts here
main()
{
    printf("\n welcome to the MUSIC LIST DATA\n\n\t 1.LOGIN\n\n\t 2.REGISTER\n\t 3.EXIT OR QUIT");
    scanf("%d",&g);
    switch(g)
    {
        case 1:login();
        break;
        case 2:signin();
        break;
        case 3: return(0);
        break;
        default:printf("\n option not found...........please enter the valid option........");
    }
return(0);
}
```

OUTPUT:

```
welcome to the MUSIC LIST DATA

        1.LOGIN
        2.REGISTER
        3.EXIT OR QUIT1
 Login Form.....
Enter User Name : musicteam
Enter Password : ----

MENU:
 1.READ DATA
 2.SORTING
 3.SEARCHING
 enter option:1

*****WELCOME TO MUSIC******
Enter the number of songs: 3
Enter the details of song 1:
Title: abc
Genre: sad
Artist: karthik
Time Period: 90
Enter the details of song 2:
Title: lmn
Genre: happy
Artist: chithra
Time Period: 80
Enter the details of song 3:
Title: pqr

Genre: happy
Artist: SPB
Time Period: 90
```

```
 enter y or Y to continue,enter n or N to exit:y
MENU:
 1.READ DATA
 2.SORTING
 3.SEARCHING
 enter option:2

Enter your choice:
1. Sort by Song Title
2. Sort by Artist
3. Sort by genre
4. Sort by time period
5.add a song
6.delete a song
1

List of songs sorted by title:
1. abc - sad - karthik - 90
2. lmn - happy - chithra - 80
3. pqr - happy - SPB - 90

 enter y or Y to continue,enter n or N to exit:y
MENU:
 1.READ DATA
 2.SORTING
 3.SEARCHING
 enter option:2

Enter your choice:
1. Sort by Song Title
2. Sort by Artist
3. Sort by genre
4. Sort by time period
5.add a song
6.delete a song
```

```
2

List of songs sorted by artist:
1. pqr - happy - SPB - 90
2. lmn - happy - chithra - 80
3. abc - sad - karthik - 90

 enter y or Y to continue,enter n or N to exit:y
MENU:
 1.READ DATA
 2.SORTING
 3.SEARCHING
 enter option:2

Enter your choice:
1. Sort by Song Title
2. Sort by Artist
3. Sort by genre
4. Sort by time period
5.add a song
6.delete a song
3

List of songs sorted by genre:
1. pqr - happy - SPB - 90
2. lmn - happy - chithra - 80
3. abc - sad - karthik - 90

 enter y or Y to continue,enter n or N to exit:y
MENU:
 1.READ DATA
 2.SORTING
 3.SEARCHING
 enter option:2

Enter your choice:
```

```
1. Sort by Song Title
2. Sort by Artist
3. Sort by genre
4. Sort by time period
5.add a song
6.delete a song
4

List of songs sorted by time period:
1. lmn - happy - chithra - 80
2. pqr - happy - SPB - 90
3. abc - sad - karthik - 90

 enter y or Y to continue,enter n or N to exit:y
MENU:
 1.READ DATA
 2.SORTING
 3.SEARCHING
 enter option:2

Enter your choice:
1. Sort by Song Title
2. Sort by Artist
3. Sort by genre
4. Sort by time period
5.add a song
6.delete a song
5

 ENTER THE DETAILS OF SONG YOU WANT TO ADD : ........

Title: xyz
Genre: sid
Artist: sid
Time Period: 20
```

```
List of songs sorted by title after adding a song :
1. abc - sad - karthik - 90
2. lmn - happy - chithra - 80
3. pqr - happy - SPB - 90
4. xyz - sad - sid - 20

 enter y or Y to continue,enter n or N to exit:y

MENU:
 1.READ DATA
 2.SORTING
 3.SEARCHING
 enter option:2

Enter your choice:
1. Sort by Song Title
2. Sort by Artist
3. Sort by genre
4. Sort by time period
5.add a song
6.delete a song
6

 ENTER THE SONG YOU WANT TO DELETE FROM THE LIST : ......abc

List of songs sorted by title after deleting a song :
1. lmn - happy - chithra - 80
2. pqr - happy - SPB - 90
3. xyz - sad - sid - 20
enter y or Y to continue,enter n or N to exit:y

MENU:
 1.READ DATA
 2.SORTING
 3.SEARCHING
 enter option:
```

```
3

Enter your choice:
1. Search by Song Title
2. Search by Artist
3. Search by genre
4. Search by time period
1
Enter the song title: pqr
Song found!
pqr - happy - SPB - 90


 WANT TO TRY AGAIN......?
   PRESS 'Y' OR 'y' FOR YES
  PRESS 'n' OR 'N' FOR NO y
MENU:
 1.READ DATA
 2.SORTING
 3.SEARCHING
 enter option:



3

Enter your choice:
1. Search by Song Title
2. Search by Artist
3. Search by genre
4. Search by time period
1
Enter the song title: pqr
```

```
Song found!
pqr - happy - SPB - 90


 WANT TO TRY AGAIN......?
   PRESS 'Y' OR 'y' FOR YES
  PRESS 'n' OR 'N' FOR NO yMENU:
 1.READ DATA
 2.SORTING
 3.SEARCHING
 enter option:



3

Enter your choice:
1. Search by Song Title
2. Search by Artist
3. Search by genre
4. Search by time period
1
Enter the song title: pqr
Song found!
pqr - happy - SPB - 90


 WANT TO TRY AGAIN......?
   PRESS 'Y' OR 'y' FOR YES
  PRESS 'n' OR 'N' FOR NO
y

Enter your choice:
1. Search by Song Title
2. Search by Artist
3. Search by genre
```

```
4. Search by time period
2
Enter the artist name: chithra
Song found!
lmn - happy - chithra - 80



 WANT TO TRY AGAIN......?
   PRESS 'Y' OR 'y' FOR YES
  PRESS 'n' OR 'N' FOR NO
y

Enter your choice:
1. Search by Song Title
2. Search by Artist
3. Search by genre
4. Search by time period
4
Enter the time period: 20
time period not found!



 WANT TO TRY AGAIN......?
   PRESS 'Y' OR 'y' FOR YES
  PRESS 'n' OR 'N' FOR NO
y

Enter your choice:
1. Search by Song Title
2. Search by Artist
3. Search by genre
4. Search by time period
4
Enter the time period: 80
Song found!
lmn - happy - chithra - 8
```

```
WANT TO TRY AGAIN......?
  PRESS 'Y' OR 'y' FOR YES
 PRESS 'n' OR 'N' FOR NO
```