**ADAPTABLE MUSIC PLAYER FOR WINDOWS**

*A*

*Mini Project Report*

*Submitted in partial fulfilment of the*

*Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

IN

**INFORMATION TECHNOLOGY**

By

**<K.V.S.SIDDARTHA><1602-19-737-058>**

**<N.PRANAY TEJA><1602-19-737-028>**

****

**Department of Information Technology**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Ibrahimbagh, Hyderabad-31**

**2020**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Hyderabad-500 031**

**Department of Information Technology**



**DECLARATION BY THE CANDIDATE**

I, **K.V.S.SIDDARTHA and N.Pranay Teja** bearing hall ticket number, **1602-19-737-058andd 1602-19-737-028**, hereby declare that the project report entitled “**Adaptable Music Player For Windows Systems”** Department of Information Technology, Vasavi College of Engineering, Hyderabad, is submitted in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering** in **Information Technology**

This is a record of bonafide work carried out by me and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

**<K.V.S.SIDDARTHA>**

**<1602-19-737-058>**

**<N.PRANAY TEJA>**

**<1602-19-737-028>**

(Faculty In-Charge) (Head,Dept of IT)

**ACKNOWLEDGEMENT**

It gives us immense pleasure to thank the department of INFORMATION TECHNOLOGY, for introducing the subject “mini project” in BE third semester thatlet us learn and explore more features in “C programming language”. The satisfaction that accompanies from the successful completion of this project would not be in complete without the mention of the people who made it possible, without whose constant guidance and encouragement would have made efforts go in vain. We consider ourselves privileged to express gratitude and respect towards all those who guided us through the completion of this project.I would also like to show my appreciation to our honourable Head of the Department, Information Technology, Dr K. Ram Mohan Rao sir, for supporting us and our beloved mini project lecturer, Mrs Leelavathi mam, for letting us properly understand the process of doing the mini project using C language and for providing insight and expertise that greatly assisted the project.My parents were my first teachers and they have provided me with such a great exposure that has helped me bloom. My family and friends will always be loved for sticking by me through thick and thin. THANK YOU!

**ABSTRACT**

In order to solve the problem of complex functions and large required memory for

Windows music player on the current market, a new music player of simple, convenient, less required memory as well as user-friendly is developed. Using C language and Windows multimedia API lead to design and coding of music player. The new design mainly realizes six core functions including main play interface, .main menu, music file browsing, adding music to library, deleting music from library,timer. This player has merits of high performance, simple operation, and run in all windows operating systems.

**INTRODUCTION**

In 1979,the first personal music player was released by Sony . The Walkman combined an audio cassette player and headphones .The first MP3 player,playing audio files,was released in 1998.Although player eliminated the need for another media to hold music there a many issues which we might face while using the music players on the current market .

.They use complex functions and requires large memory. When the user starts a playlist and forget to off the application it plays all the time until the battery is dead or the user again terminates. In the regular music player the user have to go to the destination of the music file to play the music which would be difficult to change the folders all the time.

In order to terminate the program automatically we have put a timer. The user can enter the time after which he wanted to terminate the program automatically.

All the music files(mp3) in the drives(C drive, F drive, etc) will appear at the same place.

There are 5 modules in our application which are user friendly and provides the user to use according to his requirement. The modules are view main menu ,music library, add music to library, delete files in music library ,timer.

**Our work:**

Developed a music player for windows which works in any windows operating system with key features like adding music to the library, deleting music from library ,view library music, timer(for automatic termination).

Used Pthread.h header file for implementing the timer and most of the UI features were self created in the custom.h header file which had been imported in the main program.WIN API for opening the dialog boxes Windows multimedia API for

Playing the music file(mp3).

**TECHNOLOGY**

To implement any project successfully, there will be technological requirements which can either be software or hardware requirements.

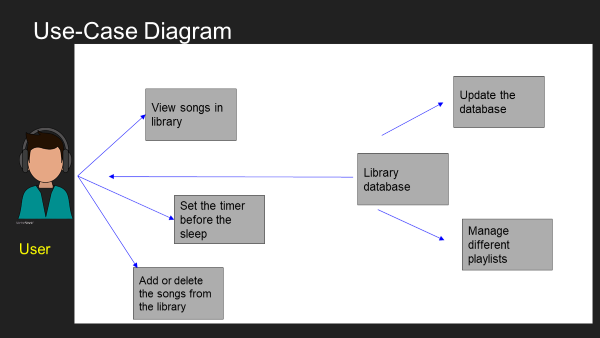
1. Software requirements:

Since our project was supposed to be based on the C programming language, it is a bare necessity to have the knowledge over syntax of the language and a proper compiler and text editor to run and write the programs. We have used a lot of C Libraries such as Pthread.h, Windows WIN API and had to link the – lwinmm to the compiler settings. These libraries are to be downloaded and linked to your compiler. We have developed the application in the Codeblocks latest version. Here most of these libraries are to be downloaded and linked. Mostly the UI features were self-created in the coustom.h header file which had been imported in the main program.

1. Hardware requirement:
2. The hardware requirements are quite low and there is no specific hardware required to run this program. We just need a processor with decent multi – tasking capabilities and minimum of 4GB ram.

**DESIGN**

**Use Case Diagram**

Here is the pictorial representation of all the available features and modules in our application. We have only one type of user and he has access to all the above mentioned modules like viewing the songs in the library, updating and managing the library, setting the timer, etc.

**User**

* View songs in library
* Set timer for termination
* Add or delete the songs from the library.
* Play or puse the music.

**IMPLEMENTATION**

**Important Snippets from Our Source Code**

User Interface for main menu🡪

|  |
| --- |
| int main\_menu() |
|  | { |
|  | int i,x1,y1,x2,y2,n,point,pointx; |
|  | char ch; |
|  |  |
|  | system("cls"); |
|  | system("COLOR BC"); |
|  | console\_title("Github link:https://github.com/siddarthakvs098"); |
|  | Hide\_cursor(); |
|  |  |
|  | adjustWindowSize(0,0,111,42); |
|  | Bk\_color(28,1); |
|  | back\_freame(0,111,0,2); |
|  | Bk\_color(28,15); |
|  | MSGinM("Adaptable Music Player For Windows System",2); |
|  |  |
|  | Bk\_color(180,17); |
|  | drowlineH(0,111,42,219); |
|  | Bk\_color(117,17); |
|  | drowlineV(3,42,1,219); |
|  | drowlineV(3,42,111,219); |
|  |  |
|  | ind\_back\_freame(1,110,40,42,28,180,1); |
|  | ind\_back\_freame(2,38,3,5,117,180,12); |
|  | ind\_back\_freame(40,109,3,5,117,180,12); |
|  |  |
|  | adjustWindowSize(0,0,112,43); |
|  | Bk\_color(180,7); |
|  | main\_menu\_background(); |
|  | working\_area\_background(); |
|  | buffer\_size(116,45); |
|  | gotoxy(2,41); |
|  | Bk\_color(28,15); |
|  | printf("STATUS"); |
|  | gotoxy(17,4); |
|  | Bk\_color(197,15); |
|  | printf("MAIN MENU"); |
|  | x1=3; |
|  | x2=38; |
|  | y1=6; |
|  | pointx=6; |
|  | Notification(); |
|  |  |
|  | getback: |
|  | y2=y1+2; |
|  | point = y1+1; |
|  | Bk\_color(117,1); |
|  | if(x1==3) |
|  | { |
|  | gotoxy(12,7); |
|  | printf("Music Library"); |
|  |  |
|  | gotoxy(12,9); |
|  | printf("Add Music to Library"); |
|  |  |
|  | gotoxy(12,11); |
|  | printf("User Manual"); |
|  |  |
|  | gotoxy(12,13); |
|  | printf("About"); |
|  |  |
|  | gotoxy(12,15); |
|  | printf("Timer"); |
|  |  |
|  | gotoxy(12,17); |
|  | printf("Exit"); |
|  |  |
|  | setcolor(14); |
|  | menu\_option(x1,x2,y1,y2); |
|  | Bk\_color(225,9); |
|  | if(y1+1==7) |
|  | { |
|  | gotoxy(12,7); |
|  | printf("Music Library"); |
|  | } |
|  | else if(y1+1==9) |
|  | { |
|  | gotoxy(12,9); |
|  | printf("Add Music to Library"); |
|  | } |
|  | else if(y1+1 == 11 ) |
|  | { |
|  | gotoxy(12,11); |
|  | printf("User Manual"); |
|  | } |
|  | else if(y1+1 == 13 ) |
|  | { |
|  | gotoxy(12,13); |
|  | printf("About"); |
|  | } |
|  | else if(y1+1 == 15 ) |
|  | { |
|  | gotoxy(12,15); |
|  | printf("Timer"); |
|  | } |
|  | else if(y1+1 == 17) |
|  | { |
|  | gotoxy(12,17); |
|  | printf("Exit"); |
|  | } |
|  | } |
|  | gotoxy(pointx,point); |
|  | printf("%c",62); |
|  | ch = getch(); |
|  | Bk\_color(117,4); |
|  |  |
|  | if(ch == -32) |
|  | { |
|  | ch = getch(); |
|  | if(ch == up || ch == down || ch == left || ch == right) |
|  | { |
|  | switch(ch) |
|  | { |
|  | case up: |
|  | if(y1>6 && y1<= 17) |
|  | { |
|  | gotoxy(pointx,point); |
|  | printf("%c",62); |
|  | setcolor(7); |
|  | menu\_option(x1,x2,y1,y2); |
|  | y1 = y1-2; |
|  | goto getback; |
|  | } |
|  | case down: |
|  | if(y1>=6 && y1<14) |
|  | { |
|  | gotoxy(pointx,point); |
|  | printf("%c",62); |
|  |  |
|  | setcolor(7); |
|  |  |
|  | menu\_option(x1,x2,y1,y2); |
|  | y1=y1+2; |
|  | goto getback; |
|  | } |
|  | else if(y1==14) |
|  | { |
|  | gotoxy(pointx,point); |
|  | setcolor(7); |
|  | menu\_option(x1,x2,y1,y2); |
|  | y1=y1+2; |
|  | goto getback; |
|  | } |
|  | else |
|  | { |
|  | goto getback; |
|  | } |
|  |  |
|  | } |
|  | } |
|  |  |
|  | } |
|  | else if( ch == 13) |
|  | { |
|  |  |
|  | if(point == 7) |
|  | { |
|  |  |
|  | Bk\_color(180,7); |
|  | working\_area\_background(); |
|  | Bk\_color(117,1); |
|  | drowlineV(6,38,96,179); |
|  | ind\_back\_freame(40,109,3,5,117,180,12); |
|  | Bk\_color(197,15); |
|  | gotoxy(75-strlen("Name"),4); |
|  | printf("Name"); |
|  | gotoxy(101,4); |
|  | printf("DATE"); |
|  | m\_library(); |
|  | goto getback; |
|  | } |
|  | else if(point == 11) |
|  | { |
|  | ind\_back\_freame(40,109,3,5,117,180,12); |
|  | Bk\_color(197,15); |
|  | gotoxy(66,4); |
|  | printf("User Manual"); |
|  | User\_manual(); |
|  | goto getback; |
|  |  |
|  | } |
|  | else if(point == 13) |
|  | { |
|  | ind\_back\_freame(40,109,3,5,117,180,12); |
|  | Bk\_color(197,15); |
|  | gotoxy(75-strlen("About"),4); |
|  | printf("About"); |
|  | about(); |
|  | goto getback; |
|  | } |
|  | else if(point == 15) |
|  | { |
|  | Bk\_color(180,7); |
|  | working\_area\_background(); |
|  | ind\_back\_freame(40,109,3,5,117,180,12); |
|  | Bk\_color(197,15); |
|  | gotoxy(75-strlen("Timer"),4); |
|  | printf("Timer"); |
|  | gotoxy(44,7); |
|  | Bk\_color(117,1); |
|  | printf("Enter time in sec : "); |
|  | scanf("%d",&n); |
|  | if(flag ==1) |
|  | pthread\_cancel(ptimer); |
|  | flag = 1; |
|  | // pthread\_t ptimer; |
|  |  |
|  | pthread\_create(&ptimer,NULL,timer,n); |
|  | delay(10); |
|  | flag = 0; |
|  | goto getback; |
|  | } |
|  | else if(point == 17) |
|  | { |
|  | exit(0); |
|  |  |
|  | } |
|  | else if(point == 9) |
|  | { |
|  | Bk\_color(180,7); |
|  | working\_area\_background(); |
|  | ind\_back\_freame(40,109,3,5,117,180,12); |
|  | Bk\_color(197,15); |
|  | gotoxy(75-strlen("Explore"),4); |
|  | printf("Explore"); |
|  | detect\_all\_the\_drives(Drives,&System\_Drive\_Count); |
|  | File\_explore(Drives); |
|  | goto getback; |
|  | } |
|  |  |
|  | } |
|  | else if(ch == 27) |
|  | { |
|  | Bk\_color(180,7); |
|  | working\_area\_background(); |
|  | ind\_back\_freame(40,109,3,5,117,180,12); |
|  | Bk\_color(197,15); |
|  | gotoxy(70,4); |
|  | printf("Notification"); |
|  | Notification(); |
|  | goto getback; |
|  | } |
|  | else{ |
|  | goto getback; |
|  | } |
|  | } |

Files exploring:

|  |
| --- |
| void File\_explore(char \*Disk) |
|  | { |
|  |  |
|  | int length,y1=7,i=0,a1=44,a2=95,b1,b2; |
|  | char drive[50]; |
|  | struct node \*temp; |
|  | restart: |
|  | length = strlen(Disk); |
|  |  |
|  |  |
|  | if(length == 0) |
|  | { |
|  | Bk\_color(117,1); |
|  | gotoxy(60,20); |
|  | printf("NO DRIVES"); |
|  | } |
|  | else if(length ==1 && System\_Drive\_Count==0) |
|  | { |
|  | Bk\_color(117,1); |
|  | gotoxy(72,21); |
|  | printf("NO DRIVES"); |
|  | } |
|  | Bk\_color(117,1); |
|  | drive\_background(45,7); |
|  |  |
|  | if(length>=1) |
|  | { |
|  | Bk\_color(117,2); |
|  | drive\_background(45,15); |
|  | } |
|  | if(length>=2) |
|  | { |
|  | Bk\_color(117,5); |
|  | drive\_background(45,23); |
|  | } |
|  | if(length >=3) |
|  | { |
|  | Bk\_color(117,20); |
|  | drive\_background(45,31); |
|  | } |
|  |  |
|  |  |
|  |  |
|  | Bk\_color(28,15); |
|  | gotoxy(66,10); |
|  | printf("C Drive : System Drive "); |
|  | if(length>=1) |
|  | { |
|  | // detect\_all\_the\_drives(Drives,System\_Drive\_Count); |
|  |  |
|  | Bk\_color(32,15); |
|  | gotoxy(66,18); |
|  | // printf("%c",Drives[i]); |
|  | printf("New voulume(%c:)",Disk[i]); |
|  | i++; |
|  | } |
|  | if(length>=2) |
|  | { |
|  | Bk\_color(82,15); |
|  | gotoxy(66,26); |
|  | printf("New Volume(%c:)",Disk[i]); |
|  | i++; |
|  | } |
|  | if(length>=3) |
|  | { |
|  | Bk\_color(68,15); |
|  | gotoxy(66,34); |
|  | printf("New Volume(%c:)",Disk[i]); |
|  | i++; |
|  | } |
|  | getback: |
|  | setcolor(12); |
|  | drive\_selector(41,y1); |
|  | char ch=getch(); |
|  | if(ch == -32) |
|  | { |
|  | ch = getch(); |
|  | if(ch == up || ch == down) |
|  | { |
|  | if(ch == up) |
|  |  |
|  | { |
|  | if(y1>7 && y1<=(7+(length\*8))) |
|  | { |
|  | Bk\_color(117,7); |
|  | drive\_selector(41,y1); |
|  | y1 = y1-8; |
|  | goto getback; |
|  | } |
|  | else |
|  | { |
|  | goto getback; |
|  | } |
|  | } |
|  | if(ch == down) |
|  | { |
|  | if(y1>=7 && y1<(7+(length\*8))) |
|  | { |
|  |  |
|  | Bk\_color(117,7); |
|  | drive\_selector(41,y1); |
|  |  |
|  | y1=y1+8; |
|  | goto getback; |
|  |  |
|  | } |
|  | else if(y1 == (7+(length\*8))) |
|  | { |
|  | goto getback; |
|  | } |
|  | } |
|  | } |
|  |  |
|  | else |
|  | { |
|  | goto getback; |
|  | } |
|  | } |
|  |  |
|  |  |
|  | else if(ch == 27 ) |
|  | { |
|  | Bk\_color(180,7); |
|  | working\_area\_background(); |
|  | ind\_back\_freame(40,109,3,5,117,180,12); |
|  | Bk\_color(197,15); |
|  | gotoxy(70,4); |
|  | printf("Notification"); |
|  | Notification(); |
|  | return; |
|  |  |
|  | } |
|  | else if(ch ==13 ) |
|  |  |
|  | { |
|  | b1 = 6; |
|  | reset: |
|  | b2 = b1+2; |
|  | Bk\_color(180,7); |
|  | working\_area\_background(); |
|  | Bk\_color(117,1); |
|  | drowlineV(6,38,96,179); |
|  | ind\_back\_freame(40,109,3,5,117,180,12); |
|  | Bk\_color(197,15); |
|  | gotoxy(75-strlen("Name"),4); |
|  | printf("Name"); |
|  | gotoxy(98,4); |
|  | printf("FILE TYPE"); |
|  | gotoxy(50,25); |
|  | song\_selector(44,95,6,8); |
|  | Bk\_color(117,1); |
|  | int i; |
|  | strcpy(current\_drive,"x:\\\\"); |
|  | setcolor(14); |
|  | song\_selector(a1,a2,b1,b2); |
|  | temp = head; |
|  | Bk\_color(117,1); |
|  | if((y1-7)/8 == 0) |
|  | { |
|  | i=7; |
|  |  |
|  | current\_drive\_music\_count=0; |
|  | strcpy(current\_drive,"C:\\\\"); |
|  | gotoxy(50,28); |
|  | Bk\_color(117,1); |
|  | head = NULL; |
|  | tail = NULL; |
|  | Bk\_color(117,1); |
|  | listFilesRecursively("C:\\\\Users\\Sandhya.Raghu\\Downloads"); |
|  | temp = head; |
|  | if(current\_drive\_music\_count == 0) |
|  | { |
|  | gotoxy(55,20); |
|  | Bk\_color(117,1); |
|  | printf("No Music Files"); |
|  | } |
|  | else |
|  | { |
|  | struct node \*temp; |
|  | temp = head; |
|  | Bk\_color(117,1); |
|  | while(temp != NULL) |
|  | { |
|  | gotoxy(44,i); |
|  | printf("%s\n",temp->music\_file); |
|  | gotoxy(100,i); |
|  | printf("mp3"); |
|  | temp = temp->next; |
|  | i=i+2; |
|  | } |
|  | printf("\n"); |
|  | } |
|  | } |
|  | else |
|  | { |
|  | i=7; |
|  | current\_drive\_music\_count=0; |
|  | current\_drive[0] = Drives[(y1-7)/8 -1]; |
|  | head=NULL; |
|  | tail=NULL; |
|  | Bk\_color(117,1); |
|  | listFilesRecursively(current\_drive); |
|  | temp = head; |
|  | if(current\_drive\_music\_count == 0) |
|  | { |
|  | gotoxy(55,20); |
|  | Bk\_color(117,1); |
|  | printf("No Music Files"); |
|  | } |
|  | else |
|  | { |
|  | struct node \*temp; |
|  | temp = head; |
|  | /\* for(int k=1;k<5;k++) |
|  | { |
|  | gotoxy(44,i); |
|  | printf("%s\n",temp->music\_file); |
|  | gotoxy(100,i); |
|  | printf("mp3"); |
|  | temp = temp->next; |
|  | i=i+2; |
|  |  |
|  | }\*/ |
|  | if(current\_drive\_music\_count > 17) |
|  | { |
|  | for(int k=1;k<17;k++) |
|  | { |
|  | gotoxy(44,i); |
|  | printf("%s\n",temp->music\_file); |
|  | gotoxy(100,i); |
|  | printf("mp3"); |
|  | temp = temp->next; |
|  | i=i+2; |
|  | } |
|  | } |
|  | else |
|  | { |
|  | while(temp != NULL) |
|  | { |
|  | gotoxy(44,i); |
|  | printf("%s\n",temp->music\_file); |
|  | gotoxy(100,i); |
|  | printf("mp3"); |
|  | temp = temp->next; |
|  | i=i+2; |
|  | } |
|  | } |
|  | printf("\n"); |
|  | } |
|  |  |
|  |  |
|  | } |
|  | redo: |
|  | if(current\_drive\_music\_count >0) |
|  | { |
|  |  |
|  | setcolor(14); |
|  | song\_selector(a1,a2,b1,b2); |
|  | Bk\_color(225,9); |
|  | gotoxy(a1,b1+1); |
|  | printf("%s",temp->music\_file); |
|  | } |
|  | char ch = getch(); |
|  | if(ch == -32) |
|  | { |
|  | ch = getch(); |
|  | if(ch == up || ch == down) |
|  | { |
|  |  |
|  | if(ch == up) |
|  | { |
|  | if(b1>6 && b1< 6+(16\*2)) |
|  | { |
|  |  |
|  |  |
|  | cord\_changed=1; |
|  | Bk\_color(117,7); |
|  | song\_selector(a1,a2,b1,b2); |
|  | gotoxy(a1,b1+1); |
|  | Bk\_color(117,1); |
|  | printf("%s",temp->music\_file); |
|  | temp = temp->prev; |
|  | b1=b1-2; |
|  | b2 = b1+2; |
|  | goto redo; |
|  | } |
|  | else if(b1 ==6 ) |
|  | { |
|  | page = 1; |
|  | goto reset; |
|  | } |
|  | } |
|  | else if(ch == down) |
|  | { |
|  | //if(b1>=6 && b1<6+((current\_drive\_music\_count-1)\*2)) |
|  | if(b1>=6 && b1<6+(15\*2)) |
|  | { |
|  | if(temp->next == NULL) |
|  | goto redo; |
|  | cord\_changed=1; |
|  | Bk\_color(117,7); |
|  | song\_selector(a1,a2,b1,b2); |
|  | gotoxy(a1,b1+1); |
|  | Bk\_color(117,1); |
|  | printf("%s",temp->music\_file); |
|  | temp = temp->next; |
|  | b1 = b1+2; |
|  | b2 = b1+2; |
|  | goto redo; |
|  | } |
|  | else if(b1 == 6+(15\*2)) |
|  | { |
|  |  |
|  | b1 = 6; |
|  | b2 = b1+2; |
|  | Bk\_color(180,7); |
|  | working\_area\_background(); |
|  | Bk\_color(117,1); |
|  | drowlineV(6,38,96,179); |
|  | gotoxy(50,25); |
|  | song\_selector(44,95,6,8); |
|  | Bk\_color(117,1); |
|  | int i=b1+1; |
|  | setcolor(14); |
|  | song\_selector(a1,a2,b1,b2); |
|  | Bk\_color(117,1); |
|  | struct node \*temp2 =temp; |
|  | while(temp != NULL) |
|  | { |
|  |  |
|  | gotoxy(44,i); |
|  | printf("%s\n",temp->music\_file); |
|  | gotoxy(100,i); |
|  | printf("mp3"); |
|  | temp = temp->next; |
|  | i=i+2; |
|  | } |
|  | temp = temp2; |
|  | goto redo; |
|  | } |
|  | else if(temp == NULL && b1>=6 && b1<6+(15\*2)) |
|  | {printf("hellooooo"); |
|  | goto redo; |
|  | } |
|  | } |
|  | else if(ch == 8) |
|  | { |
|  | goto reset; |
|  | } |
|  | } |
|  | else |
|  | { |
|  | goto redo; |
|  | } |
|  | } |
|  | else if (ch == 27) |
|  | { |
|  |  |
|  | Bk\_color(180,7); |
|  | working\_area\_background(); |
|  | ind\_back\_freame(40,109,3,5,117,180,12); |
|  | Bk\_color(197,15); |
|  | gotoxy(70,4); |
|  | printf("Notification"); |
|  | Notification(); |
|  | return; |
|  |  |
|  | } |
|  | else if(ch == 8) |
|  | { |
|  |  |
|  | Bk\_color(180,7); |
|  | working\_area\_background(); |
|  | ind\_back\_freame(40,109,3,5,117,180,12); |
|  | Bk\_color(197,15); |
|  | gotoxy(75-strlen("Explore"),4); |
|  | printf("Explore"); |
|  | goto restart; |
|  | } |
|  | else if(ch == '+') |
|  | { |
|  | if(is\_music\_file\_already\_exist(temp->music\_file) == 0) |
|  | add\_music\_to\_library(temp->data,temp->music\_file); |
|  | else |
|  | MessageBoxA(NULL,"Music file already exist in library","Warning",16); |
|  | goto redo; |
|  | } |
|  | else if(ch == 'p') |
|  | { |
|  | // close\_musicfile(); |
|  | if(cord\_changed ==0 &&play == 1) |
|  | { |
|  | play =0; |
|  | ind\_back\_freame(1,110,40,42,28,180,1); |
|  | gotoxy(2,41); |
|  | Bk\_color(28,15); |
|  | printf("pause : %s",temp->music\_file); |
|  | Bk\_color(117,1); |
|  | pause\_music(); |
|  | } |
|  | else if(cord\_changed == 0 && play==0) |
|  | { |
|  | play=1; |
|  |  |
|  | int re=play\_music(temp->data,0); |
|  |  |
|  |  |
|  | if(re == 0 && cord\_changed==1) |
|  | { |
|  | MessageBoxA(NULL," Unsupportable mp3 files","Error",64); |
|  | } |
|  | } |
|  | else if(cord\_changed==1) |
|  | { |
|  | cord\_changed = 0; |
|  | play=1; |
|  | ind\_back\_freame(1,110,40,42,28,180,1); |
|  | gotoxy(2,41); |
|  | Bk\_color(28,15); |
|  | printf("playing : %s",temp->music\_file); |
|  | Bk\_color(117,1); |
|  | close\_musicfile(); |
|  | int re=play\_music(temp->data,0); |
|  | if(re == 0) |
|  | { |
|  | MessageBoxA(NULL," Unsupportable mp3 files","Error",64); |
|  | } |
|  | } |
|  | goto redo; |
|  | } |
|  |  |
|  | else |
|  | { |
|  | goto redo; |
|  | } |
|  | } |
|  | else |
|  | { |
|  | goto getback; |
|  | } |
|  |  |
|  | } |

Listing the files recursively 🡪

|  |
| --- |
| void listFilesRecursively(char \*basePath) |
|  | { |
|  |  |
|  | char path[1000]; |
|  | char a[1000]; |
|  | struct dirent \*dp; |
|  | DIR \*dir = opendir(basePath); |
|  | if (!dir) |
|  | { |
|  | return; |
|  | } |
|  |  |
|  | while ((dp = readdir(dir)) != NULL) |
|  | { |
|  | if (strcmp(dp->d\_name, ".") != 0 && strcmp(dp->d\_name, "..") != 0 && strcmp(dp->d\_name,"$RECYCLE.BIN") !=0 && strcmp(dp->d\_name,"$Recycle.Bin")!= 0) |
|  | { |
|  | strcpy(path, basePath); |
|  | strcat(path, "/"); |
|  | strcat(path, dp->d\_name); |
|  | listFilesRecursively(path); |
|  | if(is\_mp3\_file(path) == 1) |
|  | { |
|  | insert(path,dp->d\_name); |
|  | current\_drive\_music\_count++; |
|  |  |
|  | } |
|  |  |
|  | } |
|  | } |
|  |  |
|  | closedir(dir); |
|  | } |

Function for playing music🡪

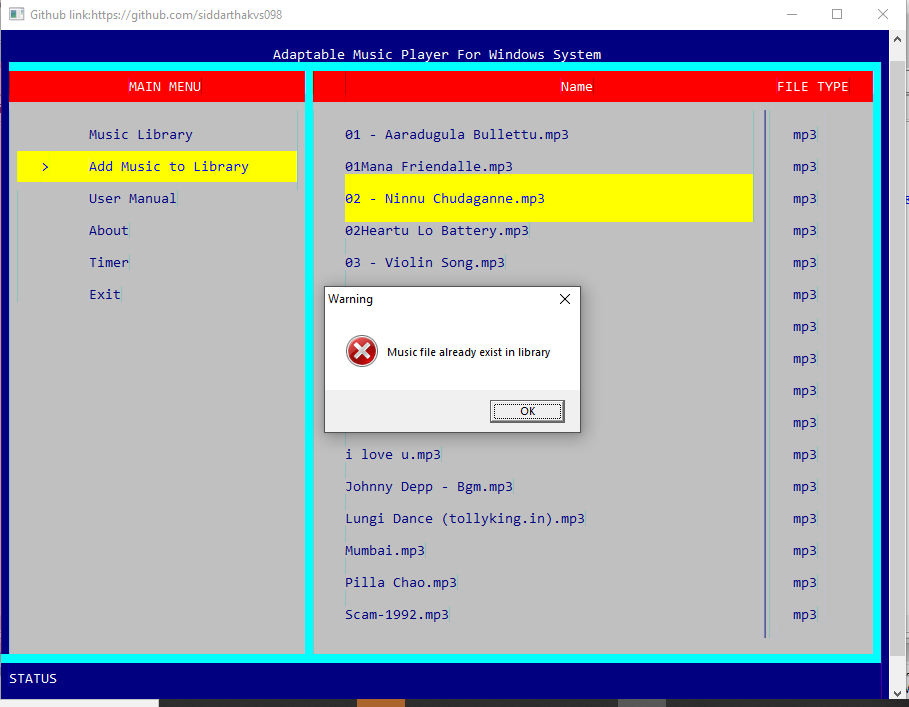
|  |
| --- |
| int play\_music(char \*file\_name,int restart) |
|  | { |
|  | char temp[250],invated\_coma; |
|  | int lenth=0,re1,re2,n; |
|  | temp[0]='\0'; |
|  | invated\_coma=34; |
|  | strcat(temp,"open "); |
|  | lenth=strlen(temp); |
|  | temp[lenth]=invated\_coma; |
|  | lenth++; |
|  | temp[lenth]='\0'; |
|  | strcat(temp,file\_name); |
|  | lenth=strlen(temp); |
|  | temp[lenth]=invated\_coma; |
|  | lenth++; |
|  | temp[lenth]='\0'; |
|  | strcat(temp," type mpegvideo alias mp3"); |
|  | re1=mciSendStringA(temp,NULL,0,NULL); |
|  | re2=mciSendString("play mp3",NULL,0,NULL); |
|  | if(restart == 1) |
|  | { |
|  | mciSendStringA("play mp3 repeat",NULL,0,NULL); |
|  | } |
|  | if(re1==0 && re2==0) |
|  | return 1; |
|  | else |
|  | return 0; |
|  |  |
|  | } |

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

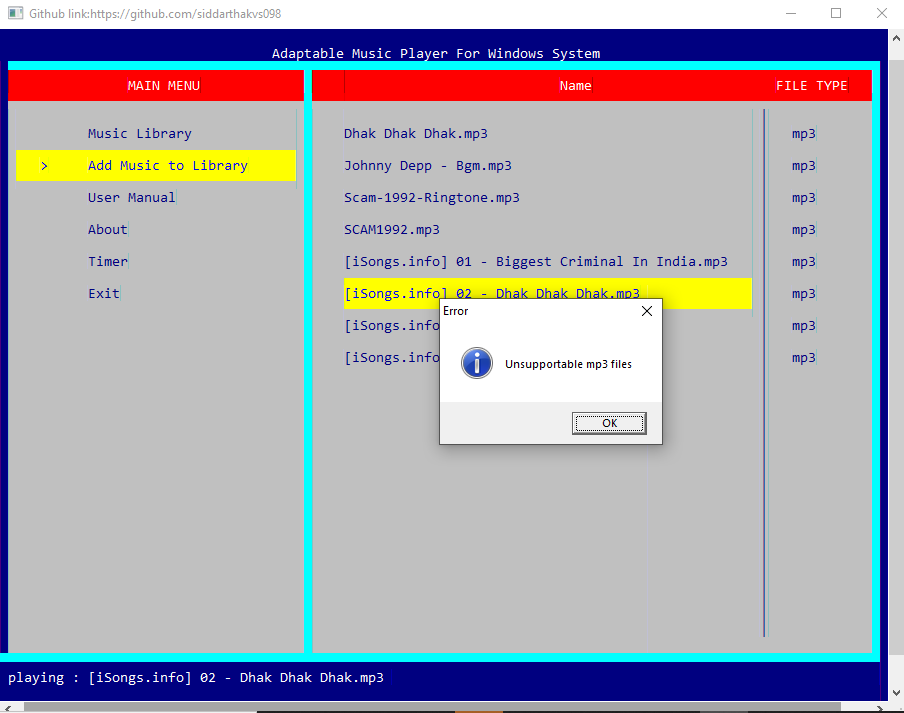
TESTING:

For executing the code first we need to link the –lwinmm to the linker in compiler settings

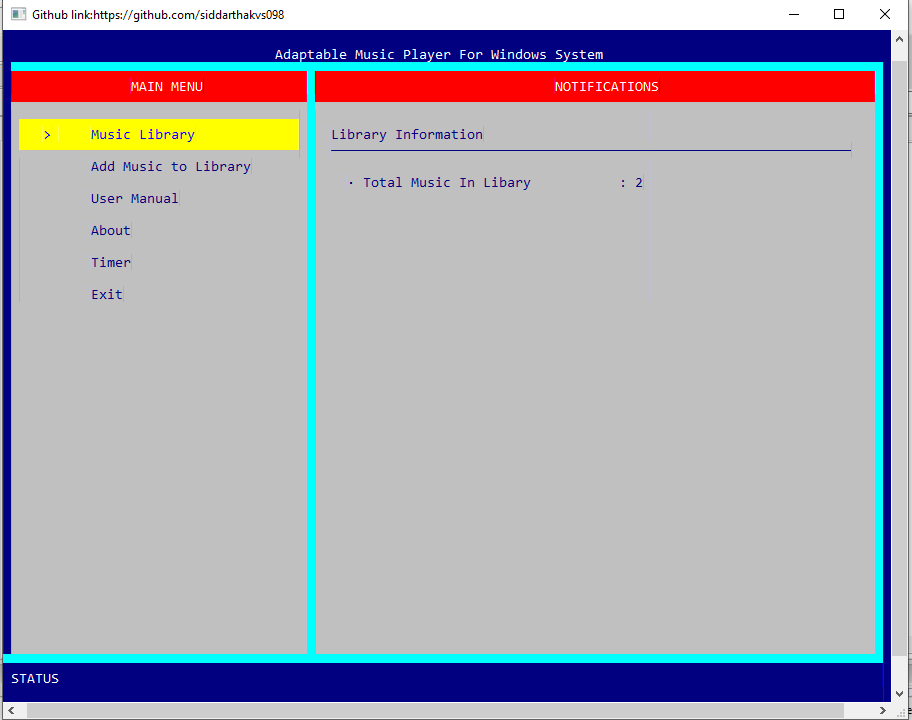
🡪If we try to add the same file into the music library



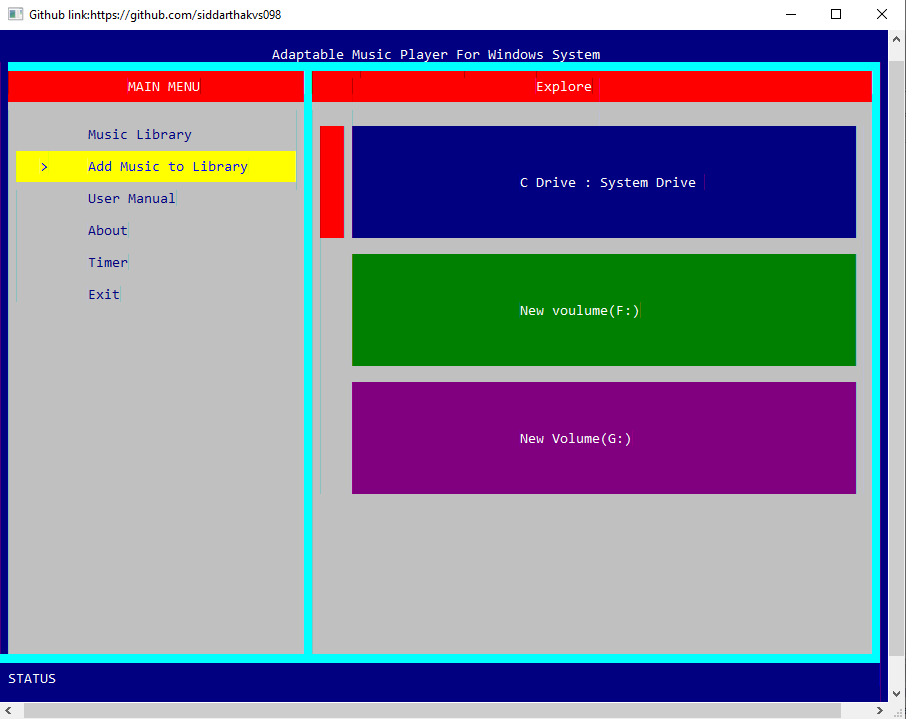
🡪When we try to play a corrupted mp3 file

RESULTS:

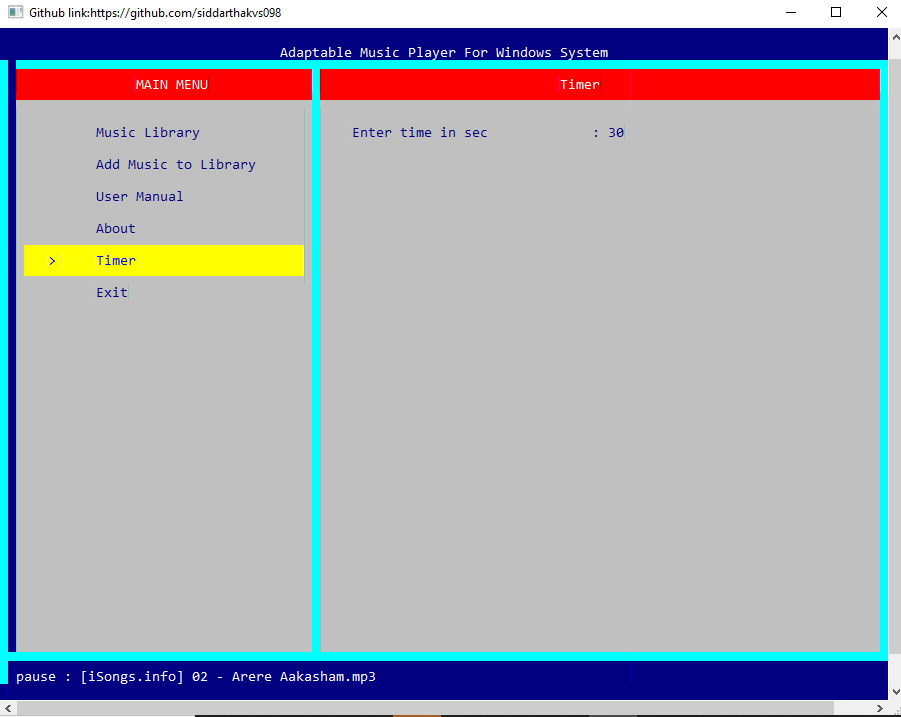
After executing the code by clicking the button build and execute.



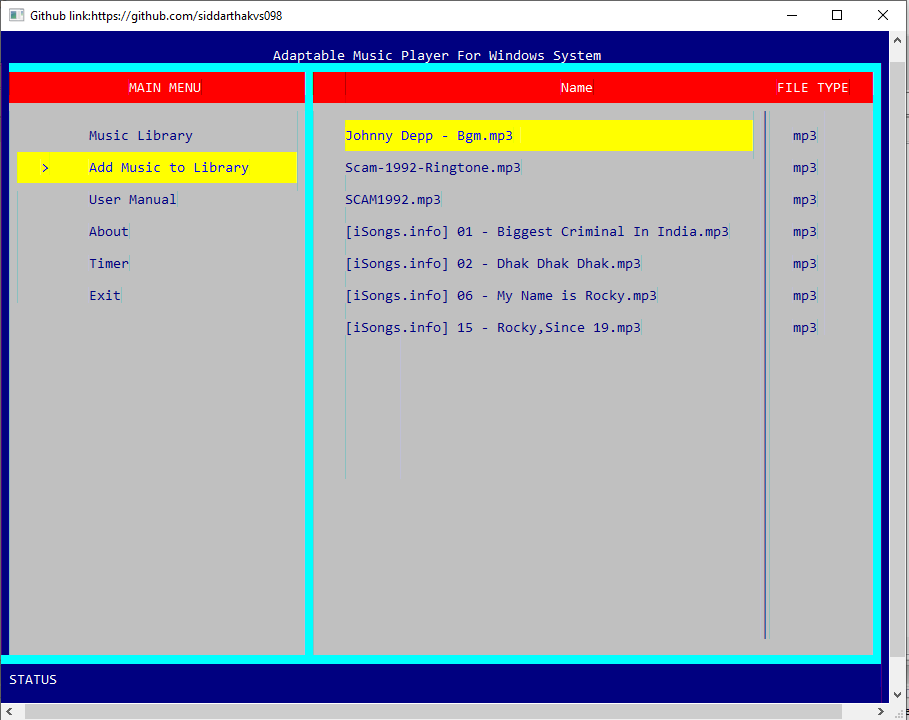
->Pressing the add music to library



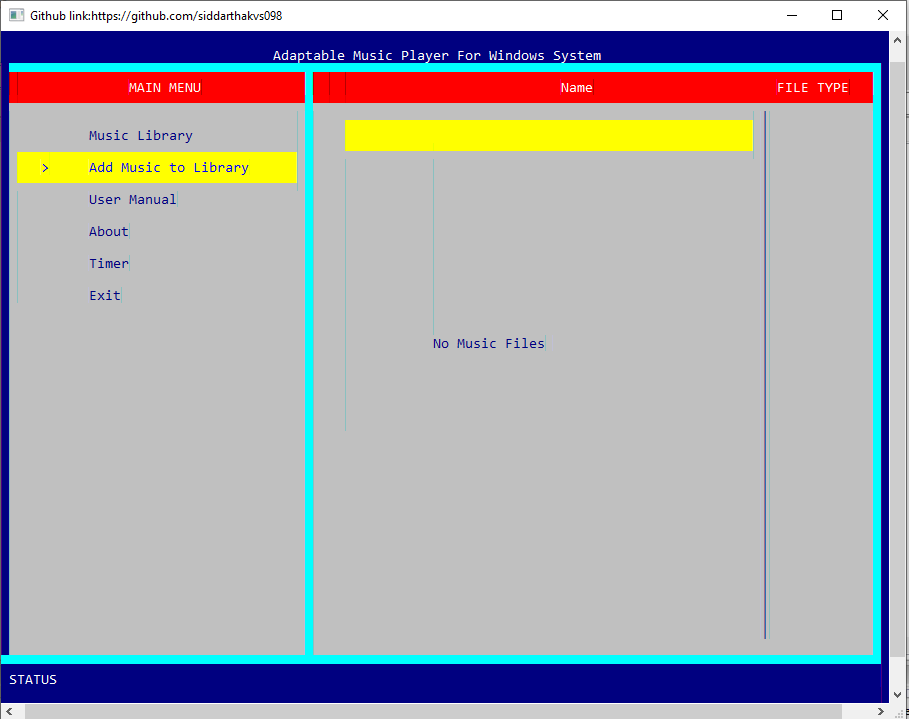
If we select the timer option

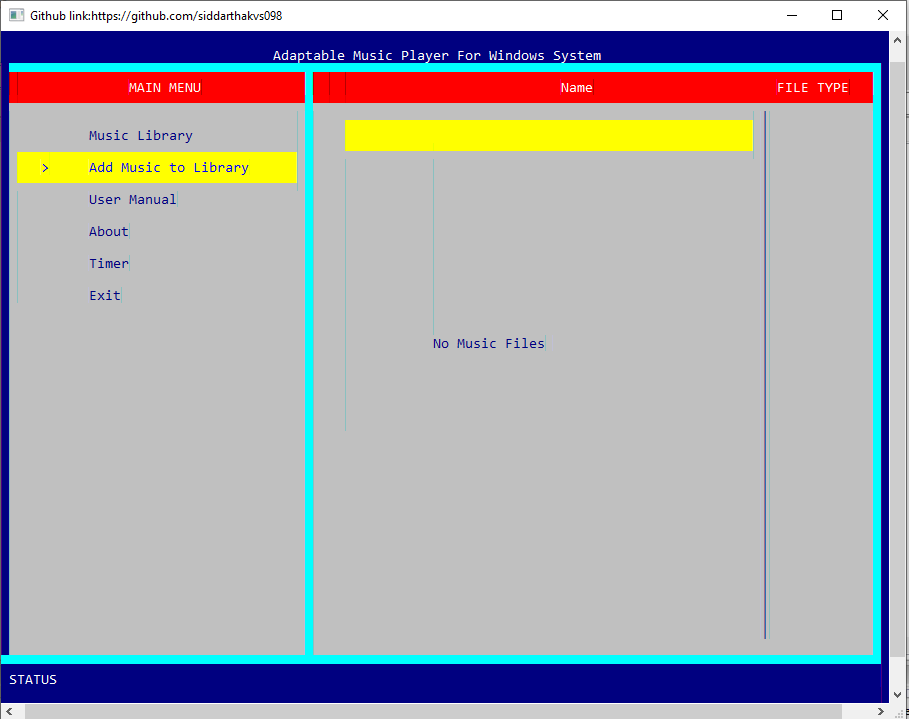


Clicking enter to go into the drive



If there are no music files in library:





ADDITIONAL LEARNINGS:

Apart from the immense understanding and practice of C language and “Adaptable music player for windows” we obtained throughout the course of this project, there were many additional learnings too:

We have learned to work in a team. Each of us have different schedules but had the same goal which made us work towards it together and hence we completed it together. Doing the entire project during these uncertain times of covid was also a challenge. We spent time to coordinate through Microsoft teams and had our discussions.

We have learned to meet deadlines. Throughout these months, we successfully completed and submitted the mini project abstract, the design document, the different modules in the project and not the report too.

Thinking out of the box. We were constantly thinking about how to make this project better than the existing ones and user friendly.

**DISCUSSION AND FUTURE WORK:**

We have developed an adaptable music player which can work in most of the windows operating platforms. We have most of the essential features that a music player needs. Songs can be added to library and managed properly. We have a timer in our application where in you can set it up and enjoy the music even while sleeping.

There’s always a scope of improvement, in our case we intend to improve our application by enhancing the UI and improving the user experience also include cross platform compatibility i.e., run our application in other environments such as the Linux or the mac OS. We also want to make it online where in the application can fetch songs from a cloud and play them

REFERENCES:

<https://stackoverflow.com/>

<https://www.geeksforgeeks.org/>

<https://www.codespeedy.com/>

Github link:

<https://github.com/siddarthakvs098/Adaptable-music-player-for-windows-system/>