CHAPTER 4

IMPLEMENTATION

4.1 FRONT-END AND BACK-END USED

In this mini project,front end used is HTML,CSS and back-end used is c#,ASP.NET,SQL and using windows operating system.

Microsoft visual studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio supports different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists.Build-in langauges include c++,VB.net,c#.

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. ... Browsers do not display the HTML tags, but use them to interpret the content of the page.

CSS is a language that describes the style of an HTML document.CSS describes how HTML elements should be displayed. Cascading Style Sheets (***CSS***) is a simple mechanism for adding style (e.g., fonts, colors, spacing) to Web documents.

ASP.NET is an open-source server-side web application framework designed for web development to produce dynamic web pages. It was developed by Microsoft to allow programmers to build dynamic web sites, web applications and web services. The full form of *ASP* is Active Server Pages, and *.NET* is Network Enabled Technologies.

4.2 DISCUSSION OF CODE SEGMENT

FOR LOGIN

```
namespace MusicWebApplication{
   public partial class login1 : System.Web.UI.Page{
      SqlConnection con = new SqlConnection(@"Data Source = (LocalDB)\MSSQLLocalDB;
```

```
AttachDbFilename=C:\Users\Catherine\Desktop\Project\MusicWebApplication\MusicWebA
pplication\App_Data\MusicDatabase.mdf;Integrated Security = True");
    protected void Page_Load(object sender, EventArgs e){}
    //sign in button
    protected void Buttonsignin_Click(object sender, EventArgs e)
    {
        //open session to share among other pages
        string s1 = TextBoxusername.Text;
        Session["username"] = s1;
        con.Open();
        //to check if the username exists in the database
        SqlCommand cmd1 = new SqlCommand("select  [username],[password] from [user]
where [username]=@a and [password]=@b", con);
        cmd1.Parameters.AddWithValue("@a", TextBoxusername.Text);
        cmd1.Parameters.AddWithValue("@b", TextBoxpassword.Text);
        SqlDataReader rd = cmd1.ExecuteReader();
        //if username exists, redirect to mainpage
        if (rd.HasRows){
            rd.Read();
            //cmd.ExecuteNonQuery();
            Response.Redirect("mainpage1.aspx");
        }
        //if username doesn't exist, display the label
        else{
            Label1.Text = "Invalid username or password";
        } con.Close();
    }

    protected void Buttonsignup_Click(object sender, EventArgs e)
    {
        //sign up button
        Response.Redirect("login2.aspx");
    }}}
```

FOR SIGN UP

```
namespace MusicWebApplication{
public partial class login2 : System.Web.UI.Page{
SqlConnection con = new SqlConnection(@&amp;quot;Data Source  (LocalDB)\MSSQLLocalDB;
AttachDbFilename=C:\Users\Catherine\Desktop\Project\MusicWebApplication\MusicWebA
pplic
ation\App_Data\MusicDatabase.mdf;Integrated Security = True&amp;quot;);
protected void Page_Load(object sender, EventArgs e){}
protected void Button1_Click(object sender, EventArgs e){
//create button
SqlCommand cmd2 = new SqlCommand(&amp;quot;insert into [dbo].[user] values
(@a,@password)&amp;quot;, con);
con.Open();
cmd2.Parameters.AddWithValue(&amp;quot;@a&amp;quot;, username.Text);
cmd2.Parameters.AddWithValue(&amp;quot;@password&amp;quot;, password.Text);
//check if username exists in user table
SqlCommand cmd3 = new SqlCommand(&amp;quot;select username from [user] where
username=@a&amp;quot;, con);
cmd3.Parameters.AddWithValue(&amp;quot;@a&amp;quot;, username.Text);
SqlDataReader rd = cmd3.ExecuteReader();
//if username exists, display label
if (rd.HasRows){
message.Text = &amp;quot;Username already exists.&amp;quot;;
message.Visible = true;
}
//if username doesn&amp;#39;t exist, insert the username and password into database
else {
rd.Close();
cmd2.ExecuteNonQuery();
Response.Redirect(&amp;quot;login1.aspx&amp;quot;);
}con.Close()}}}
```

TO VIEW ALL TRACKS AND PLAYLISTS

```
namespace MusicWebApplication{
public partial class music : System.Web.UI.Page{
SqlConnection con = new SqlConnection(@&quot;Data Source =
(LocalDB)\MSSQLLocalDB;
AttachDbFilename=C:\Users\Catherine\Desktop\Project\MusicWebApplication\MusicWebA
pplic
ation\App_Data\MusicDatabase.mdf;Integrated Security = True&quot;);
protected void Page_Load(object sender, EventArgs e){
if (!IsPostBack) { DisplayRecord();}
usr.Text = Session[&quot;username&quot;].ToString();
}
protected void Button1_Click(object sender, EventArgs e){
//log out button
Response.Redirect(&quot;login1.aspx&quot;);
}
protected void ImageButton1_Click(object sender, ImageClickEventArgs e){
//create session for the search text and redirect page
string st = search.Text;
Session[&quot;searchtext&quot;] = st;
Response.Redirect(&quot;search1.aspx&quot;);
}
public DataTable DisplayRecord(){
SqlDataAdapter Adp3 = new SqlDataAdapter(&quot;select
t.track_name,a.artist_name,t.genre from track t,artist a where
a.artist_id=t.artist_id&quot;, con);
DataTable Dt = new DataTable();
Adp3.Fill(Dt);
datatable.DataSource = Dt;
datatable.DataBind();
return Dt;
}
protected void datatable_RowCommand(object sender, GridViewCommandEventArgs e){
if (e.CommandName == &quot;Select&quot;){
```

```
//Determine the RowIndex of the Row whose Button was clicked.
int rowIndex = Convert.ToInt32(e.CommandArgument);
//Reference the GridView Row.
GridViewRow row = datatable.Rows[rowIndex];
//Fetch value of Name.
string trackname = (row.FindControl(&quot;trackname&quot;) as Label).Text;
string artistname = (row.FindControl(&quot;artistname&quot;) as Label).Text;
Session[&quot;trackname&quot;] = trackname;
Session[&quot;artistname&quot;] = artistname;
Response.Redirect(&quot;songplay.aspx&quot;);
}}
```

TO ADD OR CREATE PLAYLISTS

```
namespace MusicWebApplication{
   public partial class create : System.Web.UI.Page    {
      SqlConnection con = new SqlConnection(@"Data Source = (LocalDB)\MSSQLLocalDB; AttachDbFilename=C:\Users\Catherine\Desktop\Project\MusicWebApplication\MusicWebApplication\App_Data\MusicDatabase.mdf;Integrated Security = True");
      protected void Page_Load(object sender, EventArgs e)
      {
         usr.Text = Session["username"].ToString();
      }
      protected void Button2_Click(object sender, EventArgs e){
         //log out button
         Response.Redirect("login1.aspx");
      }
      protected void Button1_Click(object sender, EventArgs e){
         //add button
         string name = TextBox1.Text;
         Label3.Text = Session["username"].ToString();
         Label4.Text = Session["trackname"].ToString();
```

```
        //get user_id using username
        SqlCommand cmd = new SqlCommand("select user_id from [user] where username=@a", con);
        cmd.Parameters.AddWithValue("@a", Label3.Text);
        con.Open();
        string id=cmd.ExecuteScalar().ToString();       //executes first
        con.Close();

        //inserting new playlist
        SqlCommand sq = new SqlCommand("insert into playlist values(@a,@b)", con);
        sq.Parameters.AddWithValue("@a", name);
        sq.Parameters.AddWithValue("@b", id);         //executes third
        //check if playlist exists
        SqlCommand sq2 = new SqlCommand("select playlist_name from playlist p where p.user_id=@b and p.playlist_name=@a",con);
        sq2.Parameters.AddWithValue("@a", name);
        sq2.Parameters.AddWithValue("@b", id);
        con.Open();
        SqlDataReader rd = sq2.ExecuteReader();           //executes second
        if (rd.HasRows){
           error.Text = "playlist name already exists";
        }
```

CHANGE PASSWORD

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Data.Sql;
```

```
namespace MusicWebApplication{
   public partial class change : System.Web.UI.Page{
      SqlConnection con = new SqlConnection(@"Data Source = (LocalDB)\MSSQLLocalDB; AttachDbFilename=C:\Users\Catherine\Desktop\Project\MusicWebApplication\MusicWebApplication\App_Data\MusicDatabase.mdf;Integrated Security = True");
      protected void Page_Load(object sender, EventArgs e)  {
         usr.Text = Session["username"].ToString();
      }
         protected void profile_click(object sender, ImageClickEventArgs e){
         Response.Redirect("account.aspx");
      }
      protected void ImageButton1_Click(object sender, ImageClickEventArgs e){
         //create session for the search text and redirect page
         string st = search.Text;
         Session["searchtext"] = st;
         Response.Redirect("search1.aspx");
      }
      protected void Button1_Click(object sender, EventArgs e){
         //log out button
         Response.Redirect("login1.aspx");
      }
      protected void Button1_Click1(object sender, EventArgs e){
         Response.Redirect("mainpage1.aspx");
      }
      protected void Button3_Click(object sender, EventArgs e){
         Response.Redirect("playlist1.aspx");
      }
      protected void music_Click(object sender, EventArgs e){
         Response.Redirect("music.aspx");
      }
      protected void Button1_Click2(object sender, EventArgs e)
```

TO DELETE ACCOUNT AND PLAYLIST

```
namespace MusicWebApplication{
   public partial class account : System.Web.UI.Page{
      SqlConnection con = new SqlConnection(@"Data Source =(LocalDB)\MSSQLLocalDB;
AttachDbFilename=C:\Users\Catherine\Desktop\Project\MusicWebApplication\MusicWebA
pplication\App_Data\MusicDatabase.mdf;Integrated Security = True");

      protected void Page_Load(object sender, EventArgs e)
      {
        if (!IsPostBack) {DisplayRecord();}
         usr.Text = Session["username"].ToString();
         Label1.Text = Session["username"].ToString();
      }
      //button to delete the currently logged in account
      protected void Button2_Click(object sender, EventArgs e) {
         if (Session["username"] != null) {
            Label1.Text = Session["username"].ToString();
            SqlCommand cmd = new SqlCommand("delete from [user] where username=@b",
con);
            cmd.Parameters.AddWithValue("@b", Label1.Text);
            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
            Response.Redirect("login1.aspx");
        }}
     //gridview to display the playlists of the user
      public DataTable DisplayRecord()
      {
       SqlDataAdapter Adp5 = new SqlDataAdapter("select p.playlist_name,count(pt.track_id)
as no_of_tracks from playlist p, [user] u, playlist_tracks pt where u.user_id=( select a.user_id
from [user] a where a.username='"+Session["username"]+"') and u.user_id=p.user_id and
p.playlist_id=pt.playlist_id group by playlist_name", con);
            DataTable Dt = new DataTable();
```

```
            Adp5.Fill(Dt);
            datatable.DataSource = Dt;
            datatable.DataBind();
            return Dt;
        }
    protected void datatable_RowCommand(object sender, GridViewCommandEventArgs
e){
        if (e.CommandName == "Select")
            //Determine the RowIndex of the Row whose Button was clicked.
            int rowIndex = Convert.ToInt32(e.CommandArgument);
```

4.3 APPLICATIONS

- It acts as a music guide.
- It is an music guide.
- Its also a music library.
- On demand music for streaming media.
- It acts as a virtual showcase for a electronic music shop giving easy access to users through log in procedure to interact with the database.

4.4 DISCUSSION OF RESULTS

4.4.1 LOGIN PAGE

The below fig 4.1 shows Login page that allows user to access music library using their username and password,if they have an account.At first user has to enter the username which they had given in the usernsme textbox and then enter password in the password section.User has to remember the username and password.Username is visible whereas password is not visible .If the username and password mismatches,then invalid password will be displayed.If the user doesn’t has account,then user have to create account in signup page.Signup option is there below login option.The user has to use the same username and password which he has given in the sign up page.

Fig 4.1:Login page

4.4.2 SIGN UP PAGE

The below fig 4.2 Sign up page allows new users to create account using some basic details about users like Name,email,phone no.They have to create one username and password which is further used for login purpose.

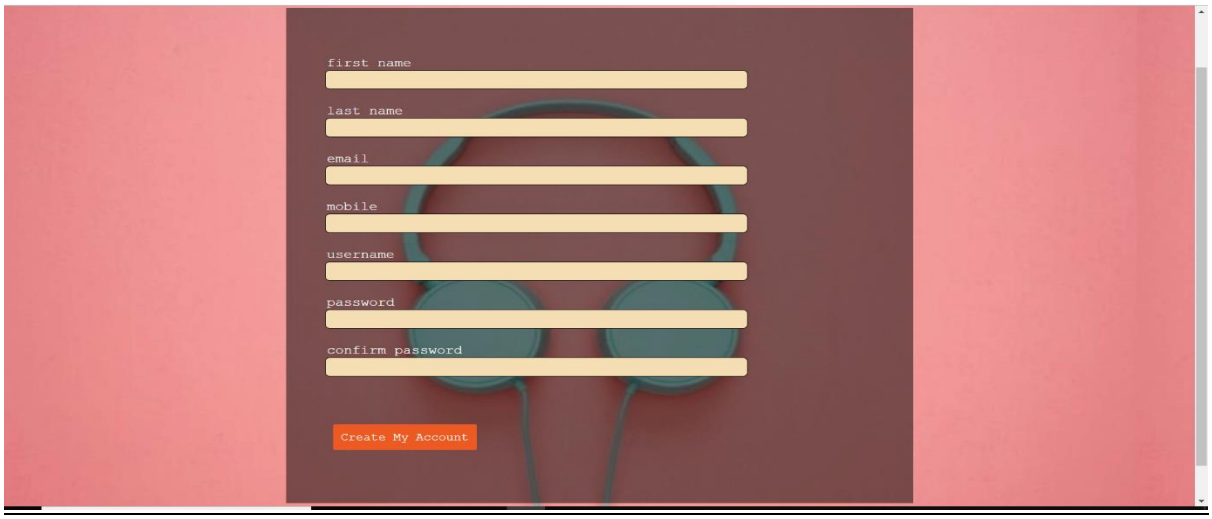

Fig 4.2:Sign up page

4.4.3 MAINPAGE

The below fig 4.3 Main page, contains navigation bar and artist details.Navigation bar consists of username, a search bar which is used to search album name, track name or playlist name. It also consists music, playlists and logout options.Artist details consists of album names and no. of tracks of artist.

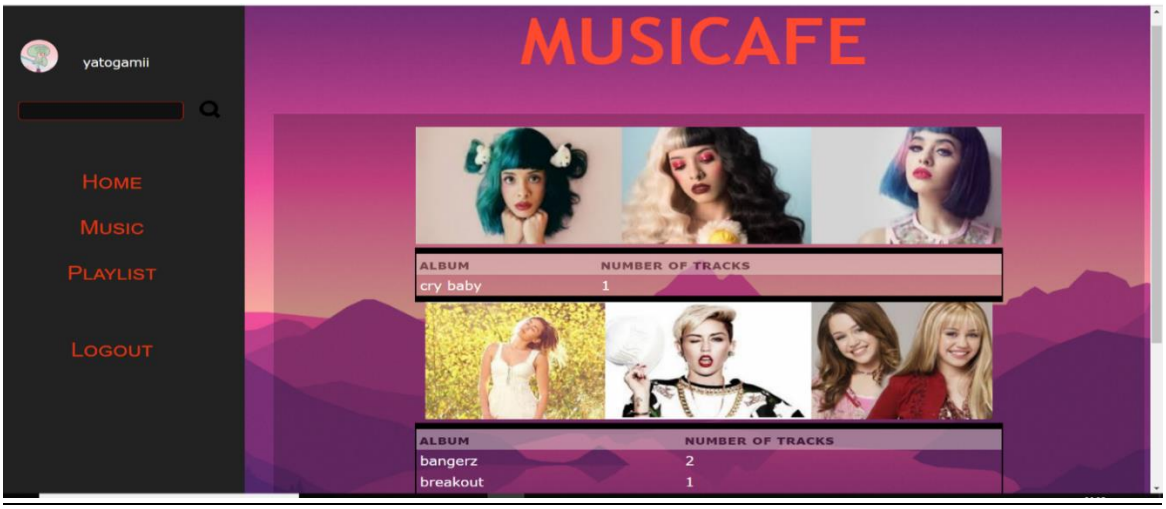

Fig 4.3:Mainpage

4.4.4 MUSIC PAGE

The below fig 4.4 shows music page.This page contains all the track names, artist name of the track and genre of tracks.There is also a "play" logo, on clicking that track info is displayed as shown below.

Fig 4.4:Music page

4.4.5 SONG PAGE

This page displays the current playing track with track name, artist name.Along with this even the album cover is shown.Using add to playlist option, user can add the current playing track into playlist.

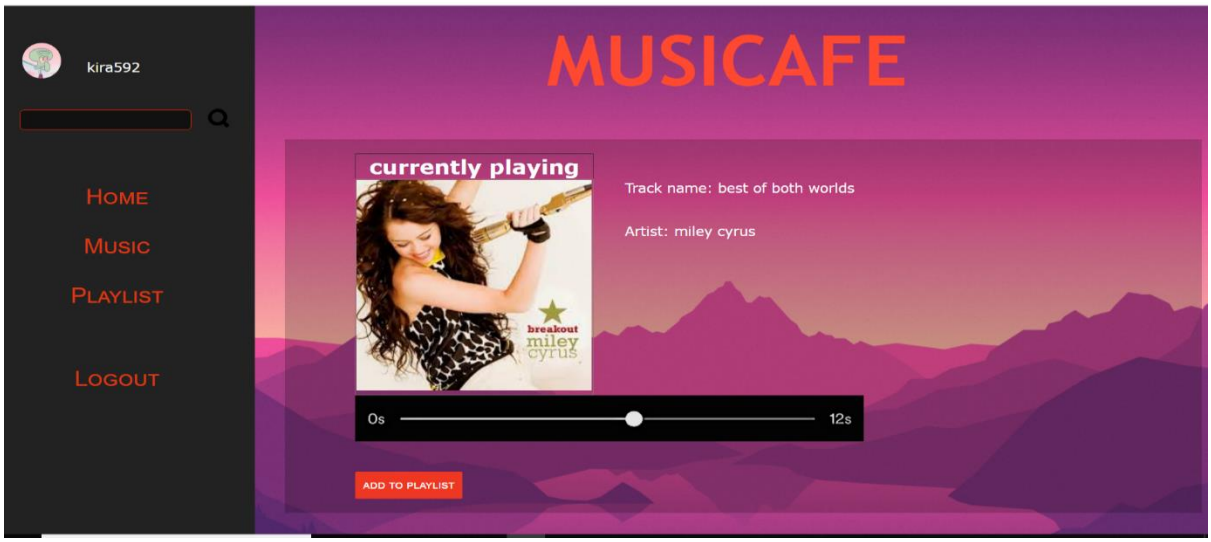

Fig 4.5:Song page

4.4.6 PLAYLIST 1

The below fig 4.6 shows playlist page.This shows the playlists details like playlist's name, no. of tracks

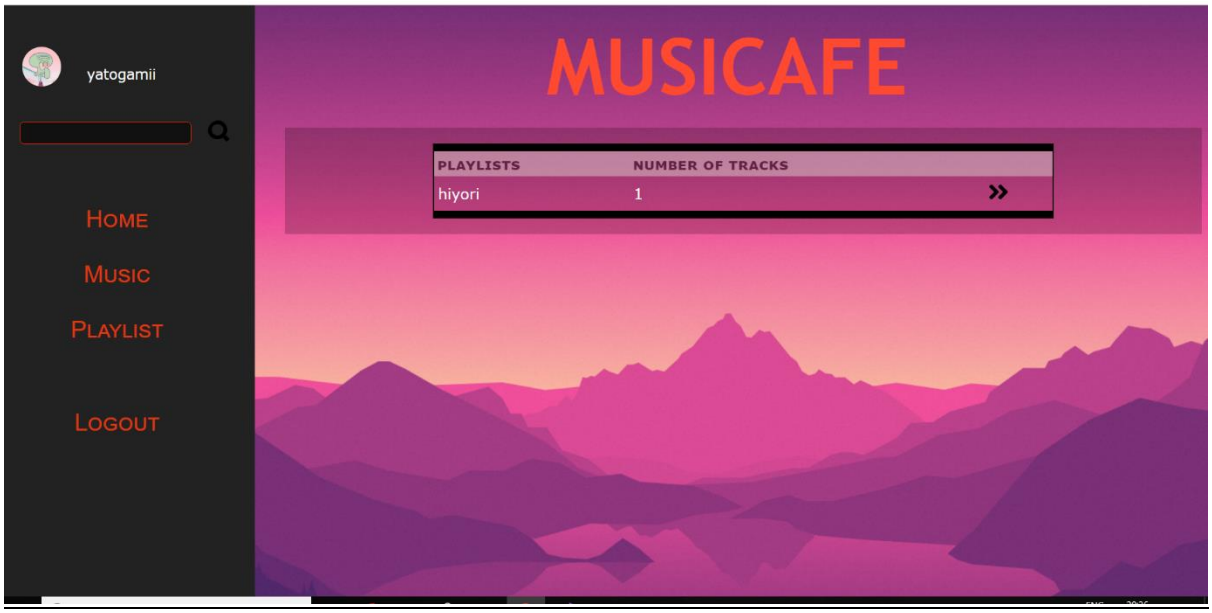

Fig 4.6:Playlist 1

4.4.7 PLAYLIST 2

The below fig 4.7 shows playlist page in which on adding current playing track into the playlist, it displays track names, artist names and genre Of the track along with playlist's name of the user.

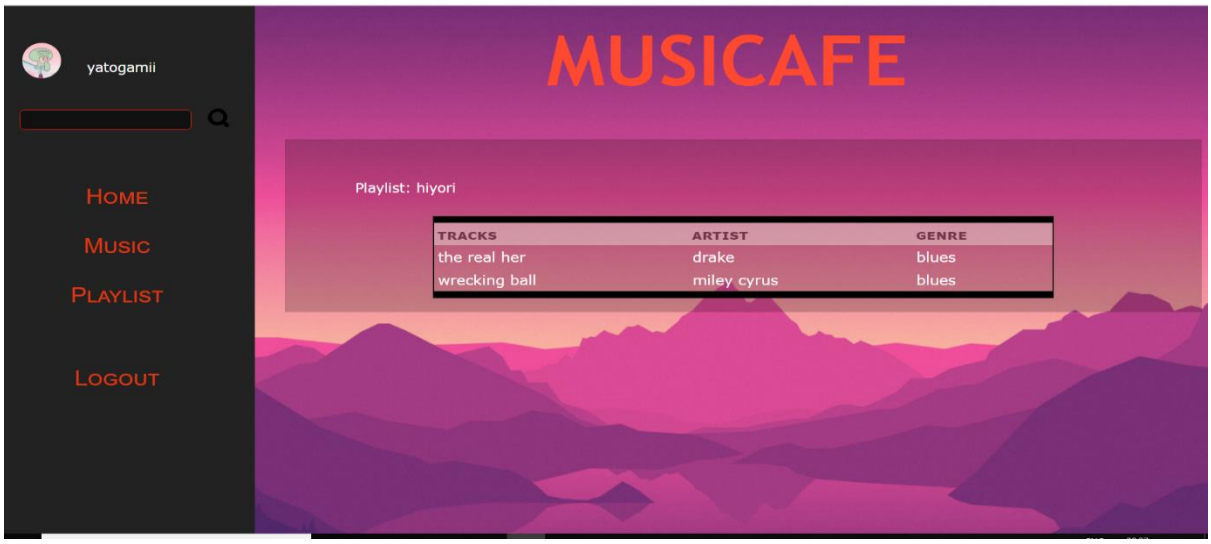

Fig 4.7:Playlist 2

4.4.8 ADD TO PLAYLIST:

This below fig 4.8 shows add to playlist page.If the user has more than one playlist, then user can choose the desired playlist.

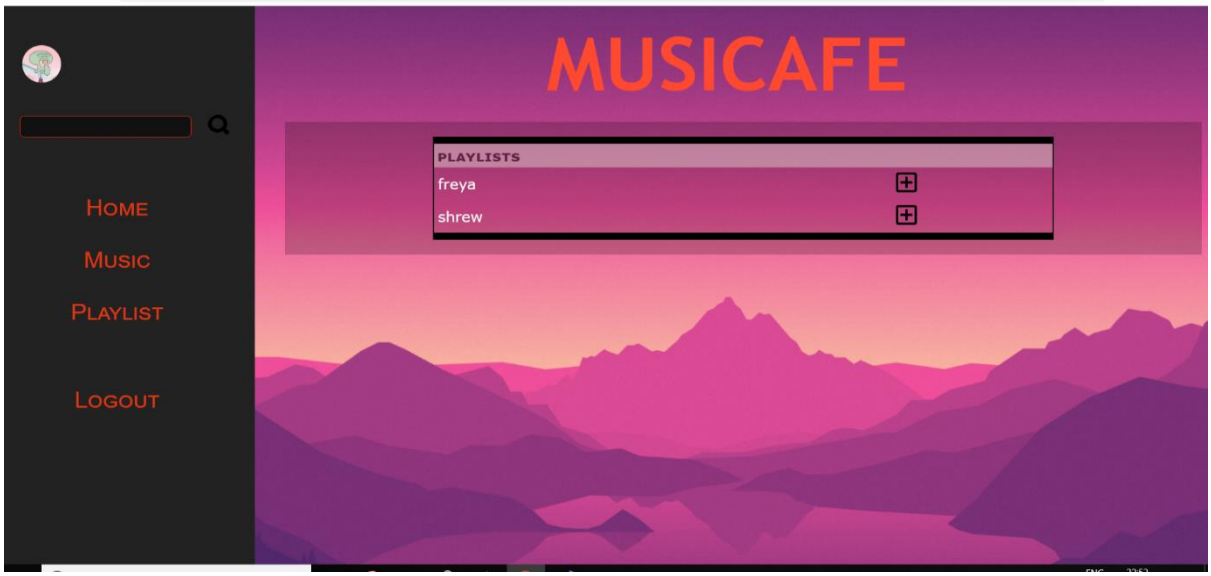

Fig 4.8:Add to playlist

4.4.9 CREATE PLAYLIST:

The below fig 4.9 shows create playlist page.If the user doesn't have a playlist or if user wants to create a new playlist.It is done by entering new playlist name and then clicking on add button.

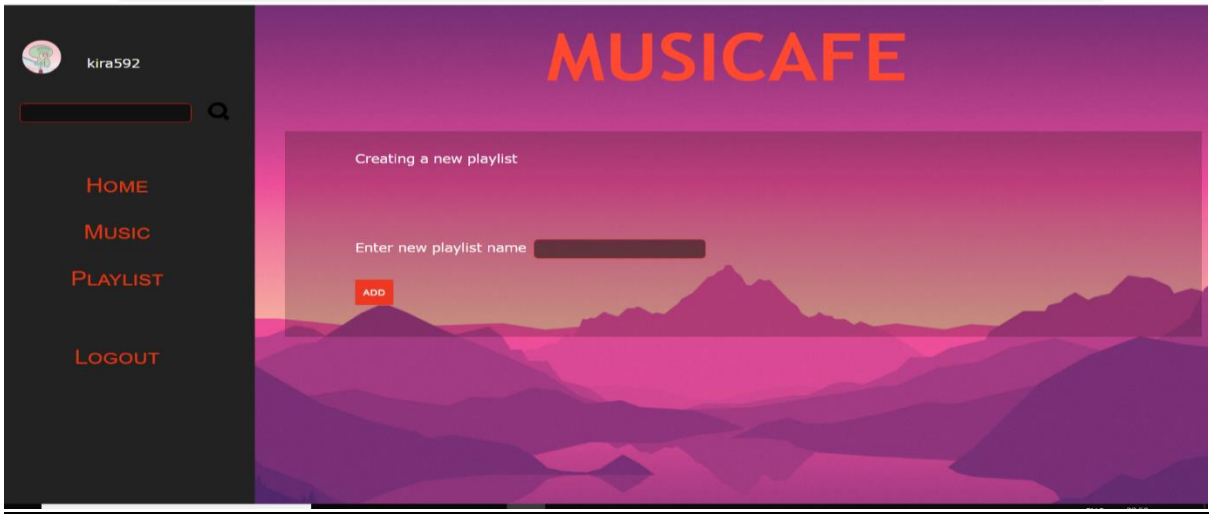

Fig 4.9:Create playlist

4.4.10 ACCOUNT PAGE

The below fig 4.10 shows account Account page enables the user to change password.It also enables the user to delete the account.Along with this even the playlist names of the user is displayed and user can also delete the playlists.

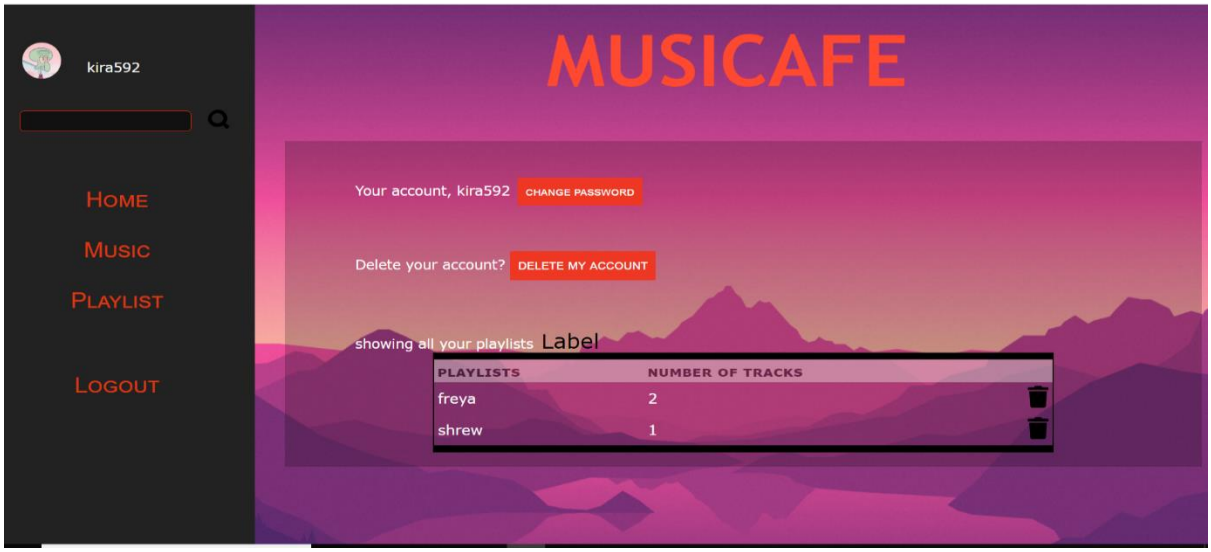

Fig 4.10:Account page

4.4.11 CHANGE PASSWORD

The below fig 4.11 shows change password page.Here, user can change the password giving a new password and then confirm.

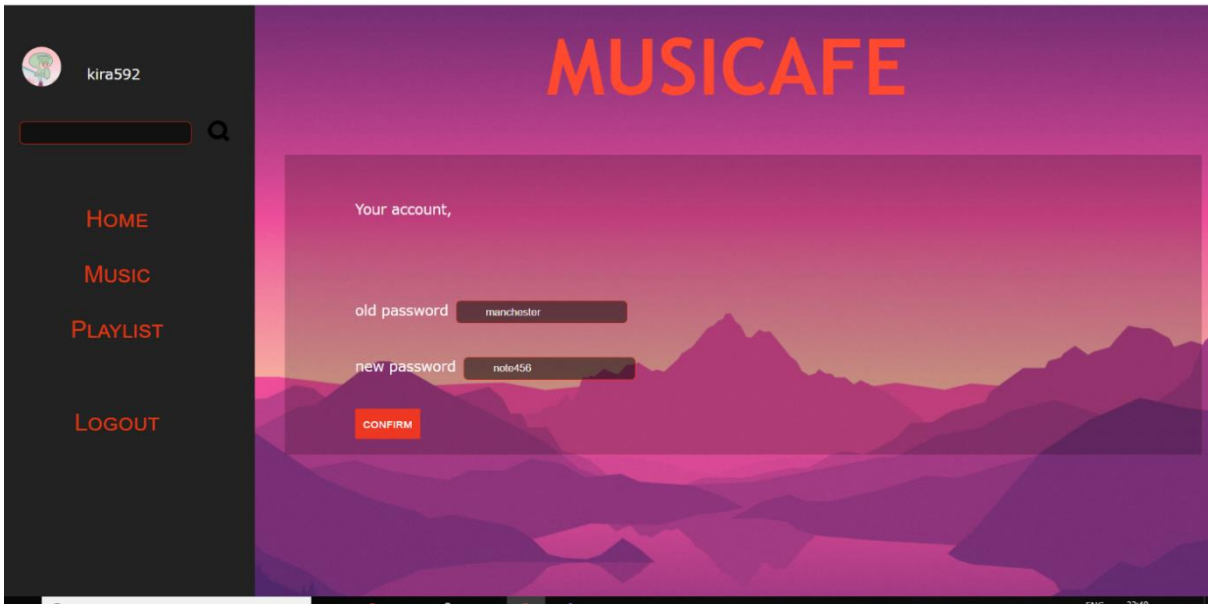

fig 4.11:Change password