



PROJECT

IN

COMPUTER SCIENCE (083)

Academic Year 2024-25

SPECTRUM MUSIC APP

NAME:

CLASS:

ROLL.NO:

REG. NO:

ACADEMIC YEAR:



CERTIFICATE

This is to certify that this project report entitled **Spectrum Music**
App is a bonafide record of the project work done by _____
of _____ class
_____ Reg.No... in the
academic year 2023 – 2024. The project has been submitted in partial fulfilment of
AISSCE for. practical held at Christ Academy CBSE School

Date:.....

Teacher in Charge

Internal Examiner

PRINCIPAL

External Examiner

ACKNOWLEDGEMENT

I solemnly take the opportunity to thank all the helping hands who made me to complete this project. First of all I thank the Almighty for keeping me hale and healthy in order to successfully complete my work.

I wish to express my sincere gratitude to Fr. Lebin, Principal of Christ Academy CBSE school, for permitting me to carry out the project and for the facilities provided for the fulfilment of this project work.

I am greatly indebted to **Mrs. Subhashini Thota**, Teacher in Computer Science who gave me immense support and guidance throughout the completion of this project. I express my gratitude towards my team members for bringing out collective productivity.

Last but not the least, I express my heartiest thanks to my lovable parents and friends for their prayers, suggestions and encouragement for the successful completion of the project.

CONTENTS

ABSTRACT

SYSTEM REQUIREMENTS

PROJECT DESIGN

DATABASE TABLES

SOURCE CODE

SAMPLE OUTPUT

CONCLUSION

BIBLIOGRAPHY

ABSTRACT

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace and indent levels. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small- and large-scale purposes.

This Project, titled **Spectrum Music App** is implemented using the Python programming language. It intends to provide a personalized music experience, modelled around popular music streaming services, but with a vision of improved aesthetic appeal and an immersive experience with curated music suggestions.

The Project contains all the basic user functionality – account creation, login as well as playing, pausing and searching for music – coupled with added features that improve user experience, such as curated playlists for both the top songs language-wise, and based on user interests, and as well a recommendation system that suggests u songs based on your music taste.

The project involves **Graphical User Interface (GUI)** using **Tkinter** and the use of **Spotify API**.

GUI enables users to interact using various visual indicators such as buttons, windows, entry boxes, etc.

Tkinter is the standard GUI library for Python.

API stands for application programming interface. It's a software component, facilitating digital products and services to communicate and interact with each other in order to share data.

The code is designed in such a way that it provides complete control to the user. It handles the common mistakes and exceptions, thus being user-friendly.

SYSTEM REQUIREMENTS

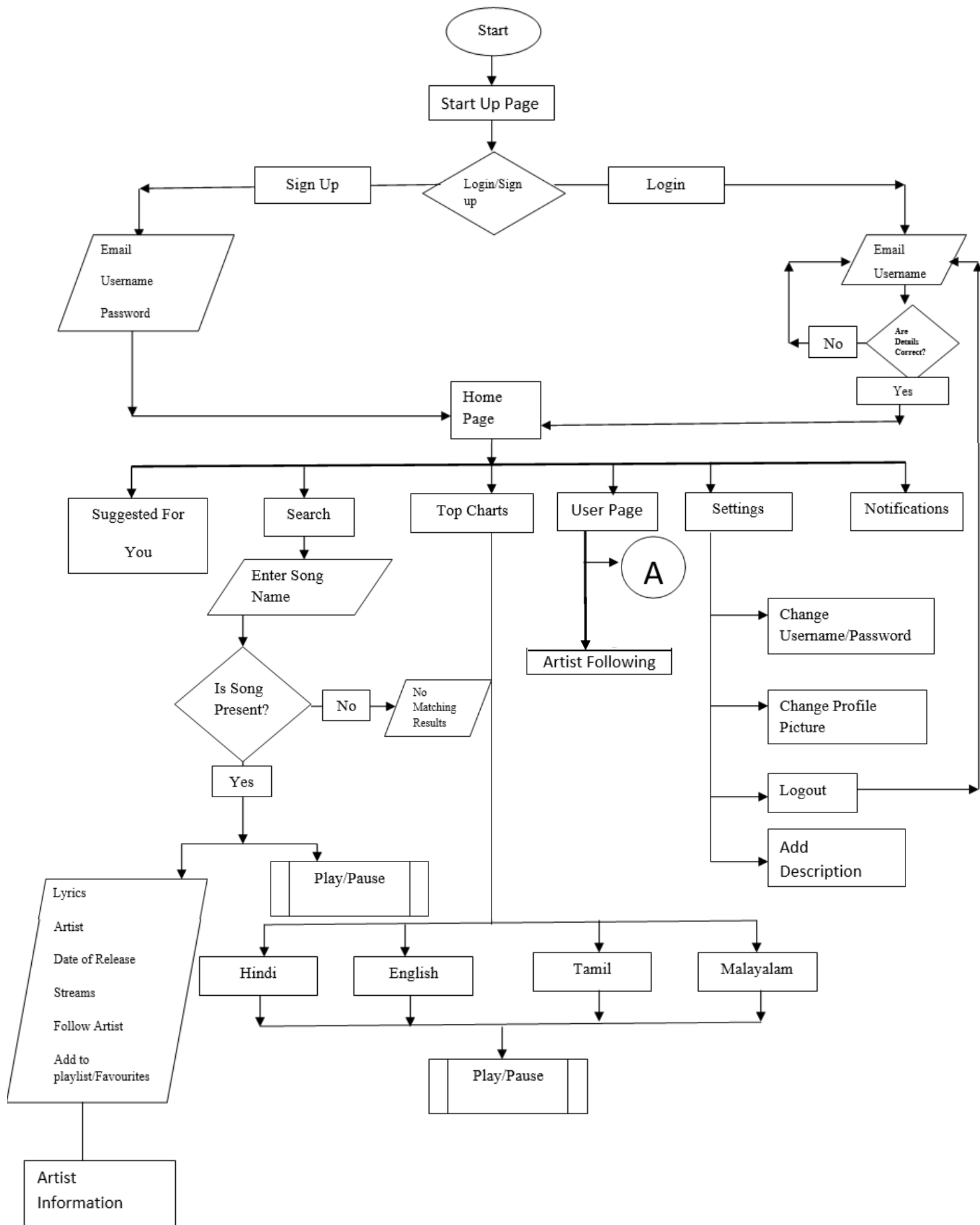
HARDWARE COMPONENTS:

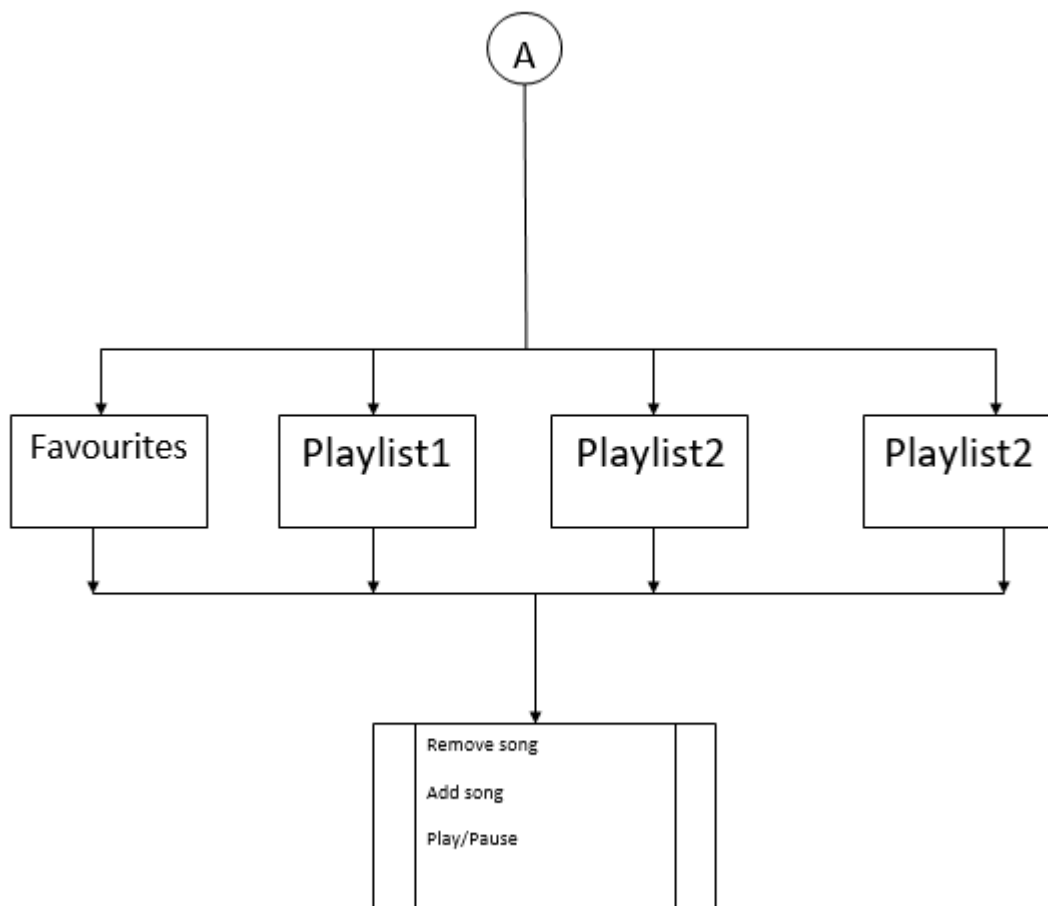
RAM	:	2 GB (Minimum) 4 GB (Recommended)
Operating System	:	32 bit x86 64 bit x64 (Recommended)
Hard Disk Memory	:	Minimum 250 MB Free
Processor	:	Dual Core 2.80 GHz or Greater
Screen Resolution	:	1366 x 768 (Optimal)
Graphics Card	:	Minimum 64 MB

SOFTWARE COMPONENTS:

Platform	:	Windows 7/8/10 with SP1
Python Version	:	Python 3.0 or Greater

PROJECT DESIGN





DATABASE TABLES

MySQL 8.0 Command Line Client

Enter password: *****

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 253

Server version: 8.0.35 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use spectrum;

Database changed

mysql> desc userinfo;

Field	Type	Null	Key	Default	Extra
username	varchar(30)	NO	PRI	NULL	
email	varchar(30)	NO		NULL	
password	varchar(30)	NO		NULL	
pfp	varchar(1000)	YES		NULL	
description	varchar(500)	YES		NULL	
playlist1_image	varchar(1000)	YES		NULL	
playlist2_image	varchar(1000)	YES		NULL	
playlist3_image	varchar(1000)	YES		NULL	

8 rows in set (0.00 sec)

mysql> desc user1234

-> ;

Field	Type	Null	Key	Default	Extra
favourite	varchar(100)	YES		NULL	
playlist1	varchar(100)	YES		NULL	
playlist2	varchar(100)	YES		NULL	
playlist3	varchar(100)	YES		NULL	
following	varchar(100)	YES		NULL	

5 rows in set (0.01 sec)

SOURCE CODE

```
from tkinter import*
import customtkinter
from tkinter import messagebox
from tkinter import PhotoImage
from PIL import Image,ImageTk
from random import*
import random
import os
from tkinter import filedialog
import tkinter.ttk as ttk
import time
from mutagen.mp3 import MP3
import pygame
import smtplib
import ssl
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from email.mime.base import MIMEBase
from email import encoders
import lyricsgenius
import spotipy
from spotipy.oauth2 import SpotifyClientCredentials
import requests
from io import BytesIO
import re
from verify_email import verify_email
import mysql.connector as mysql
import argparse
from src.stream_analyzer import Stream_Analyzer
import threading

my=mysql.connect(host="localhost",user="root",password="Spectrum24",database="spectrum")
if my.is_connected():
    print("MySQL is connected!")
cur=my.cursor()

client_id = 'e0f2b6f4d89f4a98a1ab233180f7a45c'
client_secret = '2d7e928703064ab188d8945280c90d8f'
client_credentials_manager = SpotifyClientCredentials(client_id=client_id, client_secret=client_secret)
sp = spotipy.Spotify(client_credentials_manager=client_credentials_manager)

customtkinter.set_appearance_mode("system")
customtkinter.set_default_color_theme("blue")
```

```

root=Tk()
root.title("Spectrum Music")
root.resizable(False, False)
root.geometry('925x576')
pygame.mixer.init()

image0=ImageTk.PhotoImage(file= r"F:\Spectrum\computerproject\images\startpage.png")
bg_image0=Label(root,image=image0)
bg_image0.place(relheight=1, relwidth=1)

playlist_id = '37i9dQZEVXbMDoHDwVN2tF'
top_tracks_recom = sp.playlist_tracks(playlist_id, limit=5)

song_list_recom=[]
for idx, track in enumerate(top_tracks_recom['items'], start=1):
    song_list_recom.append(f'{track["track"]["name"]} by {' '.join([artist["name"] for artist in
track["track"]["artists"]])}')")

#=====SIGNUP PAGE=====
def Signup_page():
    global image2
    image2=PhotoImage(file= r"F:\Spectrum\computerproject\images\signup page.png")
    bg_image2=Label(root,image=image2)
    bg_image2.place(relheight=1, relwidth=1)

    have_account=customtkinter.CTkLabel(root,text="Already have an account?",
        font=("Segoe Print",16),text_color="Black",bg_color="white")
    have_account.place(x=600,y=540)
    loginin=customtkinter.CTkButton(master=root,text="Login in",width=90,height=25,fg_color="#780090",
        font=("Segoe Print",16),hover_color="Blue",command=Login_page)
    loginin.place(x=810,y=540)

    def on_enter(e):
        Username.delete(0,'end')
        Username.configure(fg="black")
        Username=Entry(root,width=25,fg='#4D4D4D', border=2, bg='white',font=("Segoe Print",16))
        Username.place(x=60,y=220)
        Username.insert(0,'Enter Username')
        Username.bind("<FocusIn>",on_enter)

    def on_enter(e):
        emailid.delete(0,'end')
        emailid.configure(fg="black")
        emailid=Entry(root,width=25,fg='#4D4D4D', border=2, bg='white',font=("Segoe Print",16))
        emailid.place(x=60,y=325)
        emailid.insert(0,'Enter Email Id')
        emailid.bind("<FocusIn>",on_enter)

    def on_enter(e):
        global hide_pswrd
        global unhide_pswrd

```

```
hide_pswrd=PhotoImage(file=r"F:\Spectrum\computerproject\images\hide.png")
unhide_pswrd=PhotoImage(file=r"F:\Spectrum\computerproject\images\unhide.png")
Password.delete(0,'end')
Password.configure(fg="black")
Password.configure(show='●')
def show():
    Password.configure(show='')
def hide():
    Password.configure(show='●')
    hide_label=Button(root,image=hide_pswrd,bg="white",command=show)
    hide_label.place(x=400, y=425)
    unhide_label=Button(root,image=unhide_pswrd,bg="white",command=hide)
    unhide_label.place(x=400, y=425)
hide_label=Button(root,image=hide_pswrd,bg="white",command=show)
hide_label.place(x=400, y=425)

Password=Entry(root,width=25,fg='#4D4D4D', border=2, bg='white',font=("Segoe Print",16))
Password.place(x=60,y=425)
Password.insert(0,'Enter Password')
Password.bind("<FocusIn>",on_enter)

def signup_info():
    global email
    global password
    global username
    email=(emailid.get()).strip()
    password=(Password.get()).strip()
    username=(Username.get()).strip()
    emailcheck=verify_email(email)
    q1="select username,email from userinfo where email='{ }' or username='{ }'".format(email,username)
    cur.execute(q1)
    query1=cur.fetchone()

if len(username)==0 or len(email)==0 or len(password)==0:
    messagebox.showinfo(title="Entry Incomplete",message="Please Enter Data to all the fields")

elif len(username)!=0 and len(email)!=0 and len(password)!=0 and emailcheck==False:
    messagebox.showinfo(title="Email Error",message="Please Enter valid Email id")

elif len(username)!=0 and len(email)!=0 and len(password)!=0 and len(password)<8:
    messagebox.showinfo(title="Password Error",message="Password must be minimum 8 characters")

elif query1!=None:
    if query1[1]==email:
        messagebox.showinfo(title="Email error",message="Email already in use")
    elif query1[1]!=email and query1[0]==username:
        messagebox.showinfo(title="Username error",message="Username already in use")
else:
    messagebox.showinfo(title="Successfull signup",message="You Have successfully created a Spectrum account")
```

```

q="insert into userinfo(username,email,password,playlist1_image,playlist2_image,playlist3_image)
values('{{','{{','{{','F:/Spectrum/computerproject/images/playlist1.png','F:/Spectrum/computerproject
/images/playlist2.png','F:/Spectrum/computerproject/images/playlist3.png'))"
.format(username,email,password)
q0="create table {{}(favourite varchar(100),playlist1 varchar(100), playlist2 varchar(100),
playlist3 varchar(100),following varchar(100))".format(username)
cur.execute(q)
cur.execute(q0)
my.commit()

smtp_port = 587
smtp_server = "smtp.gmail.com"
email_from = "spectrummusicapp@gmail.com"
email_list = [email]
pswd = "jmqpbbsnaokpaoq"
subject = "Welcome to Spectrum Music"

for person in email_list:
    body="welcome to Spectrum Music App, a place to relax, refresh and
        reflect with music.\nIt's a place where you can get to know more about the music
        you listen to and discover new songs."
    msg = MIMEMultipart()
    msg['From'] = email_from
    msg['To'] = person
    msg['Subject'] = subject
    msg.attach(MIMEText(body, 'plain'))

    filename = "F:\Spectrum\computerproject\spectrum music.png"
    attachment= open(filename, 'rb')

    attachment_package = MIMEBase('application', 'octet-stream')
    attachment_package.set_payload((attachment).read())
    encoders.encode_base64(attachment_package)
    attachment_package.add_header('Content-Disposition', "attachment; filename= " + filename)
    msg.attach(attachment_package)

    text = msg.as_string()

    print("Connecting to server...")
    TIE_server = smtplib.SMTP(smtp_server, smtp_port)
    TIE_server.starttls()
    TIE_server.login(email_from, pswd)
    print("Succesfully connected to server")
    print()

    print(f'Sending email to: {person}...')
    TIE_server.sendmail(email_from, person, text)
    print(f'Email sent to: {person}')
    print()

global notify
notify=0
Main_page()

```

```

signup_button=customtkinter.CTkButton(master=root,text="SIGN UP",width=180,
height=50,fg_color="#a900cb",font=("Bahnschrift SemiBold SemiCondens",32),
hover_color="#780090",command=signup_info)
signup_button.place(x=200,y=500)

```

#=====LOGIN PAGE=====

```

def Login_page():
    global image1
    image1=ImageTk.PhotoImage(file= r"F:\Spectrum\computerproject\images\Login page.png")
    bg_image1=Label(root,image=image1)
    bg_image1.place(relheight=1, relwidth=1)

```

```

def on_enter(e):
    email2.delete(0,'end')
    email2.configure(fg="black")
    email2=Entry(root,width=25,fg='#4D4D4D', border=2, bg='white',font=("Segoe Print",16))
    email2.place(x=60,y=250)
    email2.insert(0,'Enter Email Id')
    email2.bind("<FocusIn>",on_enter)

```

```

def on_enter(e):
    Pswrd.delete(0,'end')
    Pswrd.configure(fg="black")
    Pswrd.configure(show='●')
    global hide_pswrd
    global unhide_pswrd
    hide_pswrd=PhotoImage(file= r"F:\Spectrum\computerproject\images\hide.png")
    unhide_pswrd=PhotoImage(file= r"F:\Spectrum\computerproject\images\unhide.png")
    def show():
        Pswrd.configure(show="")
    def hide():
        Pswrd.configure(show='●')
        hide_label=Button(root,image=hide_pswrd,bg="white",command=show)
        hide_label.place(x=400,y=375)
        unhide_label=Button(root,image=unhide_pswrd,bg="white",command=hide)
        unhide_label.place(x=400,y=375)
        hide_label=Button(root,image=hide_pswrd,bg="white",command=show)
        hide_label.place(x=400,y=375)

```

```

Pswrd=Entry(root,width=25,fg='#4D4D4D', border=2, bg='white',font=("Segoe Print",16))
Pswrd.place(x=60,y=375)
Pswrd.insert(0,'Enter Password')
Pswrd.bind("<FocusIn>",on_enter)

```

```

def reset_pswrd(event):
    global reset1
    reset1=ImageTk.PhotoImage(file= r"F:\Spectrum\computerproject\images\reset1.png")
    bg_reset1=Label(root,image=reset1)
    bg_reset1.place(relheight=1, relwidth=1)
    global otp_num
    otp_num=randint(1000000,9999999)

```

```

def on_enter(e):
    email3.delete(0,'end')
    email3.configure(fg="black")
email3=Entry(root,width=25,fg='#4D4D4D', border=2, bg='white',font=("Segoe Print",16))
email3.place(x=260,y=300)
email3.insert(0,'Enter Email Id')
email3.bind("<FocusIn>",on_enter)

def closing(event):
    Login_page()
cross=Label(root,text="X",font=("Segoe Print",14),bg="#D7D1D1",fg="black",anchor='w')
cross.place(x=642,y=140)
cross.bind("<Button-1>", closing)

def continue2(otp_entry,email013):
    global otp_num
    if str(otp_num)!=otp_entry:
        messagebox.showinfo(title="OTP Error",message="Wrong OTP")
    else:
        global reset3
        reset3=ImageTk.PhotoImage(file= r"F:\Spectrum\computerproject\images\reset3.png")
        bg_reset3=Label(root,image=reset3)
        bg_reset3.place(relheight=1, relwidth=1)

        cross=Label(root,text="X",font=("Segoe Print",14),bg="#D7D1D1",fg="black",anchor='w')
        cross.place(x=642,y=141)
        cross.bind("<Button-1>", closing)

def on_enter(e):
    new.delete(0,'end')
    new.configure(fg="black")
new=Entry(root,width=25,fg='#4D4D4D', border=2, bg='white',font=("Segoe Print",16))
new.place(x=260,y=240)
new.insert(0,'Enter New Password')
new.bind("<FocusIn>",on_enter)

def on_enter(e):
    new1.delete(0,'end')
    new1.configure(fg="black")
new1=Entry(root,width=25,fg='#4D4D4D', border=2, bg='white',font=("Segoe Print",16))
new1.place(x=260,y=340)
new1.insert(0,'Confirm New Password')
new1.bind("<FocusIn>",on_enter)

def reset_final_pswrd(event):
    new00=(new.get()).strip()
    new01=(new1.get()).strip()
    if new00==new01=="" or new00!=new01:
        messagebox.showinfo(title="Password Error",message="Incorrect Entry")
    elif new00==new01 and len(new01)<8:
        messagebox.showinfo(title="Password Error",message="Password Must
            be minimum 8 characters")

```

```

elif new00==new01 and len(new01)>8:
    q2="update userinfo set password='{ }' where email='{ }'".format(new01,email013)
    cur.execute(q2)
    my.commit()
    messagebox.showinfo(title="Password Reset",message="Password Reset Successfull")
    Login_page()
reset=Label(root,text="Reset",font=("Segoe Print",14),bg="#D7D1D1",fg="black",anchor='w')
reset.place(x=545,y=403)
reset.bind("<Button-1>", reset_final_pswrd)

```

```

def continuel(event):

```

```

    email013=(email3.get()).strip()
    emailcheck=verify_email(email013)
    q1="select username,email from userinfo where email='{ }'".format(email013)
    cur.execute(q1)
    query1=cur.fetchone()

```

```

if len(email013)==0:

```

```

    messagebox.showinfo(title="Entry Incomplete",message="Please Enter Email ID")

```

```

elif emailcheck==False or query1==None:

```

```

    messagebox.showinfo(title="Email Error",message="Please Enter valid Email id")

```

```

else:

```

```

    global otp_num
    smtp_port = 587
    smtp_server = "smtp.gmail.com"
    email_from = "spectrummusicapp@gmail.com"
    email_to = email013
    pswd = "jmqpbbsnaokpaoq"
    message = "The OTP-"+str(otp_num)
    simple_email_context = ssl.create_default_context()

```

```

try:

```

```

    TIE_server = smtplib.SMTP(smtp_server, smtp_port)
    TIE_server.starttls(context=simple_email_context)
    TIE_server.login(email_from, pswd)
    print()
    TIE_server.sendmail(email_from, email_to,message)
    print(f'Email successfully sent to - {email_to}')

```

```

except Exception as e:

```

```

    print(e)

```

```

finally:

```

```

    TIE_server.quit()

```

```

global reset2

```

```

reset2=ImageTk.PhotoImage(file= r"F:\Spectrum\computerproject\images\reset2.png")

```

```

bg_reset2=Label(root,image=reset2)

```

```

bg_reset2.place(relheight=1, relwidth=1)

```

```

def on_enter(e):

```

```

    otp.delete(0,'end')

```

```

    otp.configure(fg="black")

```

```

otp=Entry(root,width=25,fg='#4D4D4D', border=2, bg='white',font=("Segoe Print",16))

```



```
otp.place(x=260,y=300)
otp.insert(0,'Enter OTP')
otp.bind("<FocusIn>",on_enter)
```

```
cross=Label(root,text="X",font=("Segoe Print",14),bg="#D7D1D1",fg="black",anchor='w')
cross.place(x=642,y=142)
cross.bind("<Button-1>", closing)
```

```
def continue02(event):
    otp_entry=(otp.get()).strip()
    continue2(otp_entry,email013)
```

```
continue_2=Label(root,text="Next",font=("Segoe Print",14),bg="#D7D1D1",fg="black",anchor='w')
continue_2.place(x=545,y=397)
continue_2.bind("<Button-1>", continue02)
```

```
continue_1=Label(root,text="Next",font=("Segoe Print",14),bg="#D7D1D1",fg="black",anchor='w')
continue_1.place(x=545,y=394)
continue_1.bind("<Button-1>", continue1)
```

```
forgot=Label(root,text="Forgot Password?",font=("Segoe Print",10),bg="white",fg="#B02727",anchor='w')
forgot.place(x=318,y=417)
forgot.bind("<Button-1>", reset_pswrd)
```

```
def login_info():
    global email
    global password
    email=(email2.get()).strip()
    password=(Pswrd.get()).strip()
    q2="select password from userinfo where email='{ }'".format(email)
    cur.execute(q2)
    query2=cur.fetchone()
    if len(email)==0 or len(password)==0:
        messagebox.showinfo(title="Entry Incomplete",message="Please Enter Data to all the fields")
    elif len(email) !=0 and len(password)!=0 and email.endswith("@gmail.com")==False:
        messagebox.showinfo(title="Email Error",message="Please Enter valid Email id")
    elif query2==None:
        messagebox.showinfo(title="Email Error",message="Incorrect Email_id")
    elif query2[0]!=password:
        messagebox.showinfo(title="Password Error",message="Incorrect Password")
    elif query2[0]==password:
        messagebox.showinfo(title="Successfull login",message="You Have successfully  
logged into your account")
```

```
smtp_port = 587
smtp_server = "smtp.gmail.com"
email_from = "spectrummusicapp@gmail.com"
email_to = email
pswd = "jmqpbbsnaokpaoq"
message = "You have logged in to your Spectrum Account"
simple_email_context = ssl.create_default_context()
```

```

try:
    TIE_server = smtplib.SMTP(smtp_server, smtp_port)
    TIE_server.starttls(context=simple_email_context)
    TIE_server.login(email_from, pswd)
    print()
    TIE_server.sendmail(email_from, email_to,message)
    print(f'Email successfully sent to - {email_to}')

```

```

except Exception as e:

```

```

    print(e)

```

```

finally:

```

```

    TIE_server.quit()

```

```

global notify

```

```

notify=0

```

```

Main_page()

```

```

login_button=customtkinter.CTkButton(master=root,text="LOGIN",width=180,height=50,
    fg_color="#a900cb",font=("Bahnschrift SemiBold SemiCondens",32),
    hover_color="#780090",command=login_info)
login_button.place(x=200,y=500)

```

```

No_account=customtkinter.CTkLabel(root,text="Dont have an account?",
    font=("Segoe Print",16),text_color="Black",bg_color="white")

```

```

No_account.place(x=630,y=540)

```

```

signin=customtkinter.CTkButton(master=root,text="Sign in",width=90,height=25,fg_color="#780090",
    font=("Segoe Print",16),hover_color="Blue",command=Signup_page)

```

```

signin.place(x=825,y=540)

```

```

LOGIN_button=customtkinter.CTkButton(master=root,text="LOGIN",width=180,height=50,
    fg_color="#a900cb",font=("Bahnschrift SemiBold SemiCondens",32),
    hover_color="#780090",command=Login_page)

```

```

LOGIN_button.place(x=170,y=430)

```

```

SIGNUP_button=customtkinter.CTkButton(master=root,text="SIGN UP",width=180,height=50,
    fg_color="#a900cb",font=("Bahnschrift SemiBold SemiCondens",32),
    hover_color="#780090",command=Signup_page)

```

```

SIGNUP_button.place(x=575,y=430)

```

```

#=====HOME PAGE=====

```

```

def Main_page():

```

```

    global homebg

```

```

    homebg=ImageTk.PhotoImage(file= r"F:\Spectrum\computerproject\images\homebg1.png")

```

```

    homebg1=Label(root,image=homebg)

```

```

    homebg1.place(relheight=1, relwidth=1)

```

```

    global tablename

```

```

    q3="select * from userinfo where email='{ }'.format(email)

```

```

    cur.execute(q3)

```

```

    query3=cur.fetchone()

```

```

    tablename=query3[0]

```

```

def home_icons():
    global image_path3
    image_path3=PhotoImage(file= r"F:\Spectrum\computerproject\images\sidebar.png")
    bg_image3=Label(root,image=image_path3,bg="#272727")
    bg_image3.place(x=0, y=0)

    global menu1
    menu1=PhotoImage(file=r"F:\Spectrum\computerproject\images\home.png")
    logo1=Button(root,image=menu1,bg='#7200a3',borderwidth=0,command=Main_page)
    logo1.place(x=20,y=100,)

    global menu2
    menu2=PhotoImage(file=r"F:\Spectrum\computerproject\images\search.png")
    logo2=Button(root,image=menu2,bg='#7200a3',borderwidth=0,command=search_page)
    logo2.place(x=20,y=150,)

    global menu3
    menu3=PhotoImage(file=r"F:\Spectrum\computerproject\images\settings.png")
    logo3=Button(root,image=menu3,bg='#7200a3',borderwidth=0,command=settings_page)
    logo3.place(x=20,y=500)

    global menu5
    menu5=PhotoImage(file=r"F:\Spectrum\computerproject\images\side2.png")
    logo5=Button(root,image=menu5,bg='#7200a3',borderwidth=0,command=user_page)
    logo5.place(x=6,y=11)

def notifications():
    global notify
    global menu4
    notify=0
    messagebox.showinfo(title="Notifications",message=notify_text)
    def notify_done():
        messagebox.showinfo(title="Notifications",message=notify_text)
    menu4=PhotoImage(file=r"F:\Spectrum\computerproject\images\notifications.png")
    logo4=Button(root,image=menu4,bg='#7200a3',command=notify_done,borderwidth=0)
    logo4.place(x=875,y=0)

titlebar=Label(root,bg="#151515",width=120,height=3)
titlebar.place(x=102,y=1)
global menu4
global notify
menu4=PhotoImage(file=r"F:\Spectrum\computerproject\images\notifications1.png")
if notify==1:
    notify_text="Your Password was changed"
elif notify==2:
    notify_text="Your Email_ID was changed"
else:
    notify_text="No New Notifications"
    menu4=PhotoImage(file=r"F:\Spectrum\computerproject\images\notifications.png")
    logo4=Button(root,image=menu4,bg='#7200a3',command=notifications,borderwidth=0)
    logo4.place(x=875,y=0)
Quote=["Music is the universal language of mankind","Music is the raw energy that feeds the soul",
"Where words fail, music speaks","Music is the divine way to tell poetic things to your heart",

```

```
"Music is the divine way to tell poetic things to your heart","Music is what emotions sound like",  
"A great Song always lights your Heart","Sometimes all you need to do is listen to Music"]
```

```
q=randint(0,7)
```

```
s=Quote[q]
```

```
quote=customtkinter.CTkLabel(root,text=s,font=("Segoe Print",18),text_color="white",  
                               bg_color="#151515")
```

```
quote.place(x=107,y=5)
```

```
def get_song_player0():
```

```
    global play1
```

```
    global pause1
```

```
    global stop1
```

```
    global next1
```

```
    global previous1
```

```
    play1=PhotoImage(file=r"F:\Spectrum\computerproject\images\play0.png")
```

```
    pause1=PhotoImage(file=r"F:\Spectrum\computerproject\images\pause0.png")
```

```
    stop1=PhotoImage(file=r"F:\Spectrum\computerproject\images\stop0.png")
```

```
    next1=PhotoImage(file=r"F:\Spectrum\computerproject\images\next.png")
```

```
    previous1=PhotoImage(file=r"F:\Spectrum\computerproject\images\previous.png")
```

```
    bar=Label(root,bg="#890385",width=500,height=5)
```

```
    bar.place(x=102,y=497)
```

```
    my_slider=ttk.Scale(root,from_=0,to=100,orient=HORIZONTAL,value=0,length=400)
```

```
    my_slider.place(x=300,y=538)
```

```
    status_bar1=Label(root,text='00:00',font=("Segoe Print",14),bg="#890385",fg="white")
```

```
    status_bar1.place(x=705,y=532)
```

```
    status_bar2=Label(root,text='00:00',font=("Segoe Print",14),bg="#890385",fg="white")
```

```
    status_bar2.place(x=228,y=532)
```

```
    bar2=Label(root,bg="#890385",width=57,height=3)
```

```
    bar2.place(x=300,y=497)
```

```
    bar3=Label(root,bg="#890385",width=57,height=2)
```

```
    bar3.place(x=300,y=555)
```

```
    play_btn=Button(root,image=play1,bg="#890385",borderwidth=0)
```

```
    play_btn.place(x=485,y=500)
```

```
    stop_btn=Button(root,image=stop1,bg="#890385",borderwidth=0)
```

```
    stop_btn.place(x=440,y=500)
```

```
    next_btn=Button(root,image=next1,bg="#890385",borderwidth=0)
```

```
    next_btn.place(x=577,y=500)
```

```
    previous_btn=Button(root,image=previous1,bg="#890385",borderwidth=0)
```

```
    previous_btn.place(x=395,y=500)
```

```
    global loop
```

```
    loop=PhotoImage(file=r"F:\Spectrum\computerproject\images\loop.png")
```

```
    song_loop=Button(root,image=loop,bg="#890385",borderwidth=0)
```

```
    song_loop.place(x=530,y=500)
```

```
def music_visualizer():
```

```
    def parse_args():
```

```
        parser = argparse.ArgumentParser()
```

```
        parser.add_argument('--device', type=int, default=None, dest='device',
```

```
                             help='pyaudio (portaudio) device index')
```

```

parser.add_argument('--height', type=int, default=450, dest='height',
                    help='height, in pixels, of the visualizer window')
parser.add_argument('--n_frequency_bins', type=int, default=400, dest='frequency_bins',
                    help='The FFT features are grouped in bins')
parser.add_argument('--verbose', action='store_true')
parser.add_argument('--window_ratio', default='50/35', dest='window_ratio',
                    help='float ratio of the visualizer window. e.g. 50/35')
parser.add_argument('--sleep_between_frames', dest='sleep_between_frames', action='store_true',
                    help='when true process sleeps between frames to reduce CPU usage
                        (recommended for low update rates)')
return parser.parse_args()

def convert_window_ratio(window_ratio):
    if '/' in window_ratio:
        dividend, divisor = window_ratio.split('/')
        try:
            float_ratio = float(dividend) / float(divisor)
        except:
            raise ValueError('window_ratio should be in the format: float/float')
        return float_ratio
    raise ValueError('window_ratio should be in the format: float/float')

global running
global analyzer_thread
analyzer_thread = None
running = False

def run_FFT_analyzer():
    args = parse_args()
    window_ratio = convert_window_ratio(args.window_ratio)

    ear = Stream_Analyzer(
        device = args.device,                # Pyaudio (portaudio) device index, defaults to first mic input
        rate = None,                        # Audio samplerate, None uses the default source settings
        FFT_window_size_ms = 60,            # Window size used for the FFT transform
        updates_per_second = 500,          # How often to read the audio stream for new data
        smoothing_length_ms = 50,          # Apply some temporal smoothing to reduce noisy features
        n_frequency_bins = args.frequency_bins,
        visualize = 1,                      # Visualize the FFT features with PyGame
        verbose = args.verbose,             # Print running statistics (latency, fps, ...)
        height = args.height,              # Height, in pixels, of the visualizer window,
        window_ratio = window_ratio)        # Float ratio of the visualizer window. e.g. 24/9

    fps = 60 #How often to update the FFT features + display
    last_update = time.time()
    print("All ready, starting audio measurements now...")
    fft_samples = 0
    while running:
        if (time.time() - last_update) > (1./fps):
            last_update = time.time()
            raw_fftx, raw_fft, binned_fftx, binned_fft = ear.get_audio_features()
            fft_samples += 1

```

```

        elif args.sleep_between_frames:
            time.sleep(((1./fps)-(time.time()-last_update)) * 0.99)
    print("Stopping audio measurements...")

```

```

def stop_analyzer():
    global running
    running = False
    global musicwave3
    musicwave3=PhotoImage(file=r"F:\Spectrum\computerproject\images\wave.png")
    music_wave3=Button(root,image=musicwave,bg="#890385",borderwidth=0,command=start_analyzer)
    music_wave3.place(x=850,y=500)

```

```

def start_analyzer():
    global running
    global analyzer_thread
    global musicwave2
    musicwave2=PhotoImage(file=r"F:\Spectrum\computerproject\images\wave2.png")
    music_wave2=Button(root,image=musicwave2,bg="#890385",borderwidth=0,
                        command=stop_analyzer)
    music_wave2.place(x=850,y=500)
    if not running:
        running = True
        analyzer_thread = threading.Thread(target=run_FFT_analyzer)
        analyzer_thread.start()

```

```

global musicwave
musicwave=PhotoImage(file=r"F:\Spectrum\computerproject\images\wave.png")
music_wave=Button(root,image=musicwave,bg="#890385",borderwidth=0,command=start_analyzer)
music_wave.place(x=850,y=500)

```

```

def search_requirements():
    bar0=Label(root,bg="#151515",width=45,height=3)
    bar0.place(x=107,y=5)
    def on_enter(e):
        searchbox.delete(0,'end')
    searchbox=Entry(root,width=25,fg='white', bg='black',font=("Segoe Print",14))
    searchbox.place(x=352,y=5)
    searchbox.insert(0,'What would u like to listen to?')
    def searchresult01(event):
        if event.keysym == "Return":
            search_result=searchbox.get().lower()
            global song_list_recom
            song_name, artist_name = search_song(search_result)
            if song_name and artist_name:
                get_lyrics(song_name, artist_name)
                get_cover_image_url(song_name)
                get_song_info(song_name)
                get_song_fav(song_name)
                get_song_player(song_name)
                music_visualizer()
            if len(song_list_recom)<5:
                song_list_recom.append(song_name)

```

```

elif len(song_list_recom)==5:
    song_list_recom.remove(song_list_recom[0])
    song_list_recom.append(song_name)
searchbox.bind("<FocusIn>",on_enter)
searchbox.bind("<Return>",searchresult01)
searchtext=customtkinter.CTkLabel(root,text='SEARCH',text_color="white",
                                font=("Segoe Print",22),bg_color="#151515")
searchtext.place(x=252,y=5)

```

```

def searchresult():
    global song_list_recom
    search_result=searchbox.get().lower()
    song_name, artist_name = search_song(search_result)
    if song_name and artist_name:
        get_lyrics(song_name, artist_name)
        get_cover_image_url(song_name)
        get_song_info(song_name)
        get_song_fav(song_name)
        get_song_player(song_name)
        music_visualizer()
    if len(song_list_recom)<5:
        song_list_recom.append(song_name)
    elif len(song_list_recom)==5:
        song_list_recom.remove(song_list_recom[0])
        song_list_recom.append(song_name)

```

```

global icons
icons=PhotoImage(file=r"F:\Spectrum\computerproject\images\icons.png")
searchicon=Button(root,image=icons,bg="#151515",command=searchresult,borderwidth=0)
searchicon.place(x=725,y=10)

```

```

def songsearch():
    search_requirements()
    global search_result
    global play
    global pause
    global add
    global fav1
    play=PhotoImage(file=r"F:\Spectrum\computerproject\images\play0.png")
    pause=PhotoImage(file=r"F:\Spectrum\computerproject\images\pause0.png")
    add=PhotoImage(file=r"F:\Spectrum\computerproject\images\add.png")
    fav1=PhotoImage(file=r"F:\Spectrum\computerproject\images\heart2.png")
    song_name, artist_name = search_song(search_result)
    if song_name and artist_name:
        get_lyrics(song_name, artist_name)
        get_cover_image_url(song_name)
        get_song_info(song_name)
        get_song_fav(song_name)
        get_song_player(song_name)
        music_visualizer()

```

```

def songname(event):
    global search_result
    clicked_label = event.widget

```

```
search_result=clicked_label.cget("text")
songsearch()
```

```
q007="SELECT COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS
      WHERE TABLE_NAME = '{ }'.format(tablename)
```

```
cur.execute(q007)
query007=cur.fetchall()
folder=[]
for i in query007:
    folder.append(i[0])
```

```
##### PLAYLISTS #####
```

```
def playlist0(playlist_name):
    frame1=customtkinter.CTkScrollableFrame(root,width=799,height=435,fg_color="#1b0024")
    frame1.place(x=103,y=52)
    global image20
    image20=PhotoImage(file= r"F:\Spectrum\computerproject\images\playlist.png")
    bg_image20=Label(frame1,image=image20,bg="#1b0024")
    bg_image20.grid(row=1,column=0,)
    playlist_title=Label(frame1,text=playlist_name[0:20].replace("_", " "),font=("Segoe Print",28),
                        bg="#820AA0",fg="white")
    playlist_title.grid(row=1,column=0,sticky='w', padx=5, pady=5)
    home_icons()
    q10="select { } from { }".format(playlist_name,tablename)
    cur.execute(q10)
    query10=cur.fetchall()
```

```
def songplay(event):
    global play1
    global pause1
    global stop1
    global next1
    global previous1
    global song_result
    play1=PhotoImage(file=r"F:\Spectrum\computerproject\images\play0.png")
    pause1=PhotoImage(file=r"F:\Spectrum\computerproject\images\pause0.png")
    stop1=PhotoImage(file=r"F:\Spectrum\computerproject\images\stop0.png")
    next1=PhotoImage(file=r"F:\Spectrum\computerproject\images\next.png")
    previous1=PhotoImage(file=r"F:\Spectrum\computerproject\images\previous.png")
```

```
clicked_label = event.widget
song_result1=clicked_label.cget("text")
song_result2=song_result1.partition(' ')
song_result=song_result2[2]
```

```
def play_time():
    global stopped
    if stopped:
        return
    current_time=pygame.mixer.music.get_pos()/1000
    converted_current_time=time.strftime('%M:%S',time.gmtime(current_time))
    global song_length
    global song0
```



```

song=song0
song_mut=MP3(song)
song_length=song_mut.info.length
converted_song_length=time.strftime('%M:%S',time.gmtime(song_length))

current_time=+1
if int(my_slider.get())==int(song_length):
    status_bar2.config(text=converted_song_length)
    global loop_click1
    if loop_click1==1:
        play()
        loop_click1=0
    elif paused:
        pass
    elif int(my_slider.get())==int(current_time):
        slider_position=int(song_length)
        my_slider.config(to=slider_position,value=int(current_time))
    else:
        slider_position=int(song_length)
        my_slider.config(to=slider_position,value=int(my_slider.get()))

    converted_current_time=time.strftime('%M:%S',time.gmtime(int(my_slider.get())))
    status_bar2.config(text=converted_current_time)

    next_time=int(my_slider.get())+1
    my_slider.config(value=next_time)

status_bar1.config(text=converted_song_length)
status_bar2.after(1000,play_time)

global paused
paused=False
def pause(is_paused):
    global paused
    paused=is_paused
    if paused:
        pygame.mixer.music.unpause()
        paused=False
        pause_btn=Button(root,image=pause1,bg="#890385",borderwidth=0,
            comman=lambda:pause(paused))
        pause_btn.place(x=485,y=500)
    else:
        pygame.mixer.music.pause()
        paused=True
        play_btn=Button(root,image=play1,bg="#890385",borderwidth=0
            ,command=lambda:pause(paused))
        play_btn.place(x=485,y=500)

global stopped
stopped=False
def stop():
    pygame.mixer.music.stop()

```

```

status_bar1.config(text='00:00')
status_bar2.config(text='00:00')
my_slider.config(value=0)
play_btn=Button(root,image=play1,bg="#890385",borderwidth=0,command=play)
play_btn.place(x=485,y=500)

```

```

global loop
loop=PhotoImage(file=r"F:\Spectrum\computerproject\images\loop.png")
song_loop=Button(root,image=loop,bg="#890385",borderwidth=0,command=song_looping1)
song_loop.place(x=530,y=500)
global stopped
stopped=True

```

```

def next_song():
    global song_result
    num=playlist_songs.index(song_result)
    song_result=playlist_songs[num+1]
    play()
    song_display()

```

```

def prev_song():
    global song_result
    num=playlist_songs.index(song_result)
    song_result=playlist_songs[num-1]
    play()
    song_display()

```

```

def play():
    stop()
    global stopped
    global song0
    stopped=False
    song001=f"F:/Spectrum/computerproject/song/[SPOTIFY-DOWNLOADER.COM]
{song_result}.mp3"
    song0=song001.replace('"','_')
    pygame.mixer.music.load(song0)
    pygame.mixer.music.play()
    play_time()
    pause_btn=Button(root,image=pause1,bg="#890385",borderwidth=0,
        comman=lambda:pause(paused))
    pause_btn.place(x=485,y=500)

```

```

def slide(X):
    global song0
    global song_length
    pygame.mixer.music.load(song0)
    pygame.mixer.music.play(start=int(my_slider.get()))

```

```

my_slider=ttk.Scale(root,from_=0,to=100,orient=HORIZONTAL,value=0,length=400,command=slide)
my_slider.place(x=300,y=538)
status_bar1=Label(root,text='00:00',font=("Segoe Print",14),bg="#890385",fg="white")
status_bar1.place(x=705,y=532)
status_bar2=Label(root,text='00:00',font=("Segoe Print",14),bg="#890385",fg="white")
status_bar2.place(x=228,y=532)

```

```

bar2=Label(root,bg="#890385",width=57,height=3)
bar2.place(x=300,y=497)
bar3=Label(root,bg="#890385",width=57,height=2)
bar3.place(x=300,y=555)

play_btn=Button(root,image=play1,bg="#890385",borderwidth=0,command=play)
play_btn.place(x=485,y=500)
stop_btn=Button(root,image=stop1,bg="#890385",borderwidth=0,command=stop)
stop_btn.place(x=440,y=500)
next_btn=Button(root,image=next1,bg="#890385",borderwidth=0,command=next_song)
next_btn.place(x=577,y=500)
music_visualizer()

def song_looping1():
    def song_looping2():
        global loop
        global loop_click1
        loop_click1=0
        loop=PhotoImage(file=r"F:\Spectrum\computerproject\images\loop.png")
        song_loop=Button(root,image=loop,bg="#890385",borderwidth=0,command=song_looping1)
        song_loop.place(x=530,y=500)

    global loop1
    loop1=PhotoImage(file=r"F:\Spectrum\computerproject\images\loop1.png")
    song_loop1=Button(root,image=loop1,bg="#890385",borderwidth=0,command=song_looping2)
    song_loop1.place(x=530,y=500)
    global loop_click1
    loop_click1=1

global loop
loop=PhotoImage(file=r"F:\Spectrum\computerproject\images\loop.png")
song_loop=Button(root,image=loop,bg="#890385",borderwidth=0,command=song_looping1)
song_loop.place(x=530,y=500)

bar=Label(root,text="HHHHHHHHHHHHHHHHHHHHHHHHHHHH",font=("Segoe Print",14),
          bg="#890385",fg="#890385",anchor='w')
bar.place(x=110,y=500)

def song_display():
    if len(song_result)>22:
        string=""
        for i in range(0,17):
            string+=song_result[i]
        label_text=string+"....."
    else:
        label_text=song_result
    label=Label(root,text=label_text,font=("Segoe Print",14),bg="#890385",fg="white",anchor='w')
    label.place(x=110,y=500)
    label.bind("<Button-1>", songname)

    previous_btn=Button(root,image=previous1,bg="#890385",borderwidth=0,command=prev_song)
    previous_btn.place(x=395,y=500)
    song_display()

```

```

def menudot(btn):
    global search_result
    global search_result
    q006="SELECT COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS
        WHERE TABLE_NAME = '{ }'".format(tablename)
    cur.execute(q006)
    query006=cur.fetchall()
    opt=[]
    for i in query006:
        if i[0]!=playlist_name:
            opt.append(i[0])

    grid_info = btn.grid_info()
    index0 = btn.cget("text")
    index=index0[4::]
    search_result=playlist_songs[int(index)-1]
    def option_selected(event):
        selected_option = clicked.get()
        if selected_option == "Go to song":
            songsearch()
        elif selected_option == "Go to Artist":
            song_name, artist_name = search_song(search_result)
            artist(artist_name)
        elif selected_option ==f"Remove from {playlist_name}":
            q13="update { } set { }=NULL where { }='{ }'".
                format(tablename,playlist_name,playlist_name,search_result)
            cur.execute(q13)
            my.commit()
            playlist0(playlist_name)
        else:
            playlist=selected_option[6::]
            q11="insert into { }({ }) values('{ }')".format(tablename,playlist,search_result)
            cur.execute(q11)
            my.commit()
            playlist0(playlist_name)
    clicked=StringVar()
    clicked.set("      Song options      ")
    options=[f"Remove from {playlist_name}",f"Add to {opt[0]}",f"Add to {opt[1]}",f"Add to {opt[2]}",
        "Go to song","Go to Artist"]
    dropdown=OptionMenu(frame1,clicked,*options,command=option_selected)
    dropdown.config(bg="#310d71", fg="WHITE")
    dropdown["menu"].config(bg="#450d71", fg="WHITE")
    dropdown.grid(row=grid_info["row"],column=0,sticky='e', padx=5, pady=5)

i=1
global playlist_songs
playlist_songs=[]
for j in query10:
    if j[0]!=None:
        label=Label(frame1,text=str(i)+'.' +str(j[0]),font=("Segoe Print",16),bg="#1b0024",
            fg="white",anchor='w')
        label.grid(row=8+i,column=0,sticky='w', padx=5, pady=5)
        label.bind("<Button-1>", songplay)
        button=Button(frame1,text=f"●●● {i}",font=("Segoe Print",11),fg="white",

```

```

        bg="#1b0024",borderwidth=0)
    button.config(command=lambda b=button: menudot(b))
    button.grid(row=8+i,column=0,sticky='e', padx=5, pady=5)

    playlist_songs.append(j[0])
    i=i+1
    get_song_player0()

def playlist1():
    playlist_name=folder[1]
    playlist0(playlist_name)
def playlist2():
    playlist_name=folder[2]
    playlist0(playlist_name)
def playlist3():
    playlist_name=folder[3]
    playlist0(playlist_name)

def edit_playlist(playlist_name):
    def on_enter(e):
        playlist_name0.delete(0,'end')
        playlist_name0.configure(fg="black")
    playlist_name0=Entry(root,width=25,fg='#4D4D4D', border=2, bg='white',font=("Segoe Print",16))
    playlist_name0.place(x=295,y=245)
    playlist_name0.insert(0,'Enter Playlist name')
    playlist_name0.bind("<FocusIn>",on_enter)

    def on_enter(e):
        playlist_image.delete(0,'end')
        playlist_image.configure(fg="black")
    playlist_image=Entry(root,width=25,fg='#4D4D4D',border=2, bg='white',font=("Segoe Print",16))
    playlist_image.place(x=295,y=340)
    playlist_image.insert(0,'Enter Playlist image url')
    playlist_image.bind("<FocusIn>",on_enter)

def image_dialog():
    playlist_image.delete(0,'end')
    playlist_image.configure(fg="black")
    image00= filedialog.askopenfilename(initialdir="Downloads",title="Select a picture",
        filetypes=(("png files", "*.png"),("all files", "*.*")))
    if image00=="" or image00==" ":
        playlist_image.insert(0,'No image inserted')
    else:
        playlist_image.insert(0,image00)
    global image_change
    image_change=ImageTk.PhotoImage(file= r"F:\Spectrum\computerproject\images\camera.png")
    image_url=Button(root,image=image_change,bg="white",command=image_dialog)
    image_url.place(x=638,y=340)

def save_changes0():
    name=(playlist_name0.get()).strip()
    image=(playlist_image.get()).strip()

```

```

q0007="SELECT COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS
      WHERE TABLE_NAME = '{ }'.format(tablename)
cur.execute(q0007)
query0007=cur.fetchall()
if playlist_name[-1:-5:-1]=="niaM":
    playlist_name1=playlist_name.replace("Main","")
elif playlist_name[-1:-5:-1]=="resU":
    playlist_name1=playlist_name.replace("User","")
opt00=[]
for i in query0007:
    opt00.append(i[0])
index = opt00.index(playlist_name1)
if index==1:
    index=5
elif index==2:
    index=6
elif index==3:
    index=7

q0008="SELECT COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS
      WHERE TABLE_NAME = 'userinfo'"
cur.execute(q0008)
query0008=cur.fetchall()
opt01=[]
for i in query0008:
    opt01.append(i[0])

if image not in ["" ," ", 'No image inserted', 'Enter Playlist image url']:
    q11="update userinfo set { }='{ }' where username='{ }'.format(opt01[index],image,tablename)
    cur.execute(q11)
    my.commit()
if name not in ["" ," ", 'Enter Playlist name']:
    name001 = name.replace(' ', '_')
    q14="alter table { } change { } { } varchar(100)".format(tablename,playlist_name1,name001)
    cur.execute(q14)
    my.commit()

if playlist_name[-1:-5:-1]=="niaM":
    Main_page()
elif playlist_name[-1:-5:-1]=="resU":
    user_page()

def cancel_close():
    if playlist_name[-1:-5:-1]=="niaM":
        Main_page()
    elif playlist_name[-1:-5:-1]=="resU":
        user_page()
cancel=Button(root,text="Cancel",font=("Acme",13),fg="white",bg="#A048E4",borderwidth=0,
              command=cancel_close)
cancel.place(x=355,y=423)
save_changes=Button(root,text="Save Changes",font=("Acme",13),fg="white",bg="#A048E4",
                   borderwidth=0,command=save_changes0)
save_changes.place(x=525,y=423)

```

```
def edit_playlist01():
    global editbg
    editbg=ImageTk.PhotoImage(file= r"F:\Spectrum\computerproject\images\playlist edit.png")
    editbg1=Label(root,image=editbg)
    editbg1.place(relheight=1, relwidth=1)
    playlist_name=folder[1]+"Main"
    edit_playlist(playlist_name)
```

```
def edit_playlist02():
    global editbg
    editbg=ImageTk.PhotoImage(file= r"F:\Spectrum\computerproject\images\playlist edit.png")
    editbg1=Label(root,image=editbg)
    editbg1.place(relheight=1, relwidth=1)
    playlist_name=folder[2]+"Main"
    edit_playlist(playlist_name)
```

```
def edit_playlist03():
    global editbg
    editbg=ImageTk.PhotoImage(file= r"F:\Spectrum\computerproject\images\playlist edit.png")
    editbg1=Label(root,image=editbg)
    editbg1.place(relheight=1, relwidth=1)
    playlist_name=folder[3]+"Main"
    edit_playlist(playlist_name)
```

```
def edit_playlist001():
    global editbg
    editbg=ImageTk.PhotoImage(file= r"F:\Spectrum\computerproject\images\playlist edit2.png")
    editbg1=Label(root,image=editbg)
    editbg1.place(relheight=1, relwidth=1)
    playlist_name=folder[1]+"User"
    edit_playlist(playlist_name)
```

```
def edit_playlist002():
    global editbg
    editbg=ImageTk.PhotoImage(file= r"F:\Spectrum\computerproject\images\playlist edit2.png")
    editbg1=Label(root,image=editbg)
    editbg1.place(relheight=1, relwidth=1)
    playlist_name=folder[2]+"User"
    edit_playlist(playlist_name)
```

```
def edit_playlist003():
    global editbg
    editbg=ImageTk.PhotoImage(file= r"F:\Spectrum\computerproject\images\playlist edit2.png")
    editbg1=Label(root,image=editbg)
    editbg1.place(relheight=1, relwidth=1)
    playlist_name=folder[3]+"User"
    edit_playlist(playlist_name)
```

#=====USER PAGE=====

```
def user_page():
    global email
    global password
    global username
    global userbg
    userbg=ImageTk.PhotoImage(file= r"F:\Spectrum\computerproject\images\userbg.png")
    userbg1=Label(root,image=userbg)
    userbg1.place(relheight=1, relwidth=1)

    home_icons()
    global pfp
    pfp=PhotoImage(file=r"F:\Spectrum\computerproject\images\pfp.png")
    if query3[3]==None:
        profile=Label(root,image=pfp,width=166,height=166,bg="#1b0024")
        profile.place(x=103,y=52)
    elif query3[3]!=None:
        image = Image.open(query3[3])
        resized=image.resize((166, 166),Image.LANCZOS)
        photo = ImageTk.PhotoImage(resized)
        label = Label(root, image=photo,height=166,width=166,bg="#B932DB")
        label.image = photo
        label.place(x=103,y=52)

    global name0
    name0=PhotoImage(file=r"F:\Spectrum\computerproject\images\user3.png")
    name=customtkinter.CTkLabel(root,text=query3[0],font=("Segoe Print",24),text_color="white",
                                bg_color="#D60CDA")
    name.place(x=350,y=58)
    name1=Label(root,image=name0,bg="#D60CDA")
    name1.place(x=287,y=58)

    if query3[4]!=None:
        text1=(query3[4]).split()
        str2=""
        for i in text1:
            if (text1.index(i))%7==0 and (text1.index(i))!=0:
                str2=str2+i+"\n"
            else:
                str2+=i+" "
        edit0=customtkinter.CTkLabel(root,text="" +str2+ "",font=("Segoe Print",18),text_color="white",
                                    bg_color="#D60CDA")
        edit0.place(x=287,y=110)

    q10="select * from {}".format(tablename)
    cur.execute(q10)
    query10=cur.fetchall()

    k=1
    global following_1
    following_1=[]
    for r in query10:
        if r[4]!=None:
```



```

h=r[4].split(',')
following_1.append(h[0])
k=k+1

def following(event):
    global following_1
    frame10=customtkinter.CTkScrollableFrame(root,width=799,height=510,fg_color="#1b0024")
    frame10.place(x=103,y=52)
    global image21
    image21=PhotoImage(file= r"F:\Spectrum\computerproject\images\playlist.png")
    bg_image21=Label(frame10,image=image21,bg="#1b0024")
    bg_image21.grid(row=1,column=0,)
    following_title=Label(frame10,text=" Artist Following:",font=("Segoe Print",28),
                          bg="#820AA0",fg="white")
    following_title.grid(row=1,column=0,sticky='w', padx=5, pady=5)
    home_icons()

def redirect(event):
    clicked_label = event.widget
    artist_result1=clicked_label.cget("text")
    song_name, artist_name = search_song(artist_result1)
    artist(artist_name)
    p=1
    for t in following_1:
        label=Label(frame10,text=str(p)+'.' +t,font=("Segoe Print",16),bg="#1b0024",
                    fg="white",anchor='w')
        label.grid(row=8+p,column=0,sticky='w', padx=5, pady=5)
        label.bind("<Button-1>", redirect)
        p=p+1

following_=customtkinter.CTkLabel(root,text='Artists Following: '+str(len(following_1)),
                                  font=("Segoe Print",18),text_color="white",bg_color="#D60CDA")
following_.place(x=287,y=185)
following_.bind("<Button-1>",following)

def liked():
    q10="select * from {}".format(tablename)
    cur.execute(q10)
    query10=cur.fetchall()

    frame1=customtkinter.CTkScrollableFrame(root,width=799,height=435,fg_color="#1b0024")
    frame1.place(x=103,y=52)

    global image2
    image2=PhotoImage(file= r"F:\Spectrum\computerproject\images\fav3.png")
    bg_image2=Label(frame1,image=image2,bg="#1b0024")
    bg_image2.grid(row=1,column=0)

def songplay(event):
    global play1
    global pause1

```

```

global stop1
global next1
global previous1
global song_result
play1=PhotoImage(file=r"F:\Spectrum\computerproject\images\play0.png")
pause1=PhotoImage(file=r"F:\Spectrum\computerproject\images\pause0.png")
stop1=PhotoImage(file=r"F:\Spectrum\computerproject\images\stop0.png")
next1=PhotoImage(file=r"F:\Spectrum\computerproject\images\next.png")
previous1=PhotoImage(file=r"F:\Spectrum\computerproject\images\previous.png")

clicked_label = event.widget
song_result1=clicked_label.cget("text")
song_result2=song_result1.partition('. ')
song_result=song_result2[2]

def play_time():
    global stopped
    if stopped:
        return
    current_time=pygame.mixer.music.get_pos()/1000
    converted_current_time=time.strftime('%M:%S',time.gmtime(current_time))

    global song_length
    global song0
    song=song0
    song_mut=MP3(song)
    song_length=song_mut.info.length
    converted_song_length=time.strftime('%M:%S',time.gmtime(song_length))

    current_time+=1
    if int(my_slider.get())==int(song_length):
        status_bar2.config(text=converted_song_length)
        global loop_click1
        if loop_click1==1:
            play()
            loop_click1=0
        elif paused:
            pass
        elif int(my_slider.get())==int(current_time):
            slider_position=int(song_length)
            my_slider.config(to=slider_position,value=int(current_time))
        else:
            slider_position=int(song_length)
            my_slider.config(to=slider_position,value=int(my_slider.get()))

    converted_current_time=time.strftime('%M:%S',time.gmtime(int(my_slider.get())))
    status_bar2.config(text=converted_current_time)

    next_time=int(my_slider.get()+1)
    my_slider.config(value=next_time)

status_bar1.config(text=converted_song_length)
status_bar2.after(1000,play_time)

```

```

global paused
paused=False
def pause(is_paused):
    global paused
    paused=is_paused
    if paused:
        pygame.mixer.music.unpause()
        paused=False
        pause_btn=Button(root,image=pause1,bg="#890385",borderwidth=0,
                           comman=lambda:pause(paused))
        pause_btn.place(x=485,y=500)
    else:
        pygame.mixer.music.pause()
        paused=True
        play_btn=Button(root,image=play1,bg="#890385",borderwidth=0,
                        command=lambda:pause(paused))
        play_btn.place(x=485,y=500)

global stopped
stopped=False
def stop():
    pygame.mixer.music.stop()
    status_bar1.config(text='00:00')
    status_bar2.config(text='00:00')
    my_slider.config(value=0)
    play_btn=Button(root,image=play1,bg="#890385",borderwidth=0,command=play)
    play_btn.place(x=485,y=500)
    global loop
    loop=PhotoImage(file=r"F:\Spectrum\computerproject\images\loop.png")
    song_loop=Button(root,image=loop,bg="#890385",borderwidth=0,command=song_looping1)
    song_loop.place(x=530,y=500)
    global stopped
    stopped=True

def next_song():
    global song_result
    num=favourites.index(song_result)
    song_result=favourites[num+1]
    play()
    song_display()

def prev_song():
    global song_result
    num=favourites.index(song_result)
    song_result=favourites[num-1]
    play()
    song_display()

def play():
    stop()
    global stopped
    global song0

```

```

stopped=False
song001=f"F:/Spectrum/computerproject/song/[SPOTIFY-DOWNLOADER.COM]
        {song_result}.mp3"

song0=song001.replace('"','_')
pygame.mixer.music.load(song0)
pygame.mixer.music.play()
play_time()
pause_btn=Button(root,image=pause1,bg="#890385",borderwidth=0,command=
                    lambda:pause(paused))
pause_btn.place(x=485,y=500)

def slide(X):
    global song0
    global song_length
    pygame.mixer.music.load(song0)
    pygame.mixer.music.play(start=int(my_slider.get()))

my_slider=ttk.Scale(root,from_=0,to=100,orient=HORIZONTAL,value=0,length=400,
                    command=slide)
my_slider.place(x=300,y=538)
status_bar1=Label(root,text='00:00',font=("Segoe Print",14),bg="#890385",fg="white")
status_bar1.place(x=705,y=532)
status_bar2=Label(root,text='00:00',font=("Segoe Print",14),bg="#890385",fg="white")
status_bar2.place(x=228,y=532)

bar2=Label(root,bg="#890385",width=57,height=3)
bar2.place(x=300,y=497)
bar3=Label(root,bg="#890385",width=57,height=2)
bar3.place(x=300,y=555)

play_btn=Button(root,image=play1,bg="#890385",borderwidth=0,command=play)
play_btn.place(x=485,y=500)
stop_btn=Button(root,image=stop1,bg="#890385",borderwidth=0,command=stop)
stop_btn.place(x=440,y=500)
next_btn=Button(root,image=next1,bg="#890385",borderwidth=0,command=next_song)
next_btn.place(x=577,y=500)

def song_looping1():
    def song_looping2():
        global loop
        global loop_click1
        loop_click1=0
        loop=PhotoImage(file=r"F:\Spectrum\computerproject\images\loop.png")
        song_loop=Button(root,image=loop,bg="#890385",borderwidth=0,command=song_looping1)
        song_loop.place(x=530,y=500)

    global loop1
    loop1=PhotoImage(file=r"F:\Spectrum\computerproject\images\loop1.png")
    song_loop1=Button(root,image=loop1,bg="#890385",borderwidth=0,command=song_looping2)
    song_loop1.place(x=530,y=500)
    global loop_click1
    loop_click1=1

```

```

global loop
loop=PhotoImage(file=r"F:\Spectrum\computerproject\images\loop.png")
song_loop=Button(root,image=loop,bg="#890385",borderwidth=0,command=song_looping1)
song_loop.place(x=530,y=500)

bar=Label(root,text="HHHHHHHHHHHHHHHHHHHHHHHHHH",font=("Segoe Print",14),
           bg="#890385",fg="#890385",anchor='w')
bar.place(x=110,y=500)

def song_display():
    if len(song_result)>22:
        string=""
        for i in range(0,17):
            string+=song_result[i]
            label_text=string+"....."
        else:
            label_text=song_result
        label=Label(root,text=label_text,font=("Segoe Print",14),bg="#890385",fg="white",anchor='w')
        label.place(x=110,y=500)
        label.bind("<Button-1>", songname)

    previous_btn=Button(root,image=previous1,bg="#890385",borderwidth=0,command=prev_song)
    previous_btn.place(x=395,y=500)
    song_display()

def menudot(btn):
    global search_result
    global search_result
    q006="SELECT COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS
        WHERE TABLE_NAME = '{ }'".format(tablename)
    cur.execute(q006)
    query006=cur.fetchall()
    opt=[]
    for i in query006:
        opt.append(i[0])

    grid_info = btn.grid_info()

    index0 = btn.cget("text")
    index=index0[4::]
    search_result=favourites[int(index)-1]
    def option_selected(event):
        selected_option = clicked.get()
        if selected_option == "Go to song":
            songsearch()
        elif selected_option == "Go to Artist":
            song_name, artist_name = search_song(search_result)
            artist(artist_name)
        elif selected_option == "Remove from Favourites":
            q13="update { } set favourite=NULL where favourite='{ }'".format(tablename,search_result)
            cur.execute(q13)

```

```

        my.commit()
        liked()
    else:
        playlist=selected_option[6::]
        q11="insert into {{}} values('{{}})".format(tablename,playlist,search_result)
        cur.execute(q11)
        my.commit()
        liked()
    clicked=StringVar()
    clicked.set("      Song options      ")
    options=["Remove from Favourites",f"Add to {opt[1]}",f"Add to {opt[2]}",f"Add to {opt[3]}",
            "Go to song","Go to Artist"]
    dropdown=OptionMenu(frame1,clicked,*options,command=option_selected)
    dropdown.config(bg="#310d71", fg="WHITE")
    dropdown["menu"].config(bg="#450d71", fg="WHITE")
    dropdown.grid(row=grid_info["row"],column=0,sticky='e', padx=5, pady=5)

i=1
global favourites
favourites=[]
for j in query10:
    if j[0]!=None:
        label=Label(frame1,text=str(i)+'.' +str(j[0]),font=("Segoe Print",16),bg="#1b0024",
                    fg="white",anchor='w')
        label.grid(row=8+i,column=0,sticky='w', padx=5, pady=5)
        label.bind("<Button-1>", songplay)
        button=Button(frame1,text=f"●●● {i}",font=("Segoe Print",11),fg="white",
                    bg="#1b0024",borderwidth=0)
        button.config(command=lambda b=button: menudot(b))
        button.grid(row=8+i,column=0,sticky='e', padx=5, pady=5)
        favourites.append(j[0])
    i=i+1
get_song_player0()

global heart
heart=PhotoImage(file=r"F:\Spectrum\computerproject\images\heart1.png")
heart1=Button(root,image=heart,bg='#AA0ABC',borderwidth=0,command=liked)
heart1.place(x=155,y=289)

global playlist_image_1
playlist_image_1=Image.open(query3[5])
resized1=playlist_image_1.resize((120, 120),Image.LANCZOS)
photo1 = ImageTk.PhotoImage(resized1)
playlist_image_01=Button(root,image=photo1,height=120,width=120,borderwidth=7,bg='#AA0ABC',
                        command=playlist1)
playlist_image_01.image=photo1
playlist_image_01.place(x=125,y=430)

global playlist_image_2
playlist_image_2=Image.open(query3[6])
resized2=playlist_image_2.resize((120, 120),Image.LANCZOS)

```

```

photo2 = ImageTk.PhotoImage(resized2)
playlist_image_02=Button(root,image=photo2,height=120,width=120,borderwidth=7,bg='#AA0ABC',
                        command=playlist2)
playlist_image_02.image=photo2
playlist_image_02.place(x=285,y=430)

```

```

global playlist_image_3
playlist_image_3=Image.open(query3[7])
resized3=playlist_image_3.resize((120, 120),Image.LANCZOS)
photo3 = ImageTk.PhotoImage(resized3)
playlist_image_03=Button(root,image=photo3,height=120,width=120,borderwidth=7,bg='#AA0ABC',
                        command=playlist3)
playlist_image_03.image=photo3
playlist_image_03.place(x=450,y=430)

```

```

q0006="SELECT COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS
      WHERE TABLE_NAME = '{ }'.format(tablename)

```

```

cur.execute(q0006)
query0006=cur.fetchall()
opt0=[]

```

```

for i in query0006:

```

```

    opt = i[0].replace('_', ' ')

```

```

    opt0.append(opt)

```

```

bar1=Label(root,text="                                ",font=("Segoe Print",9),bg="#AA0ABC",fg="white")

```

```

bar1.place(x=130,y=535)

```

```

bar2=Label(root,text="                                ",font=("Segoe Print",9),bg="#AA0ABC",fg="white")

```

```

bar2.place(x=290,y=535)

```

```

bar3=Label(root,text="                                ",font=("Segoe Print",9),bg="#AA0ABC",fg="white")

```

```

bar3.place(x=455,y=535)

```

```

if len(opt0[1])>10:

```

```

    playlist1_label=Label(root,text=f"{opt0[1][0:10]}....",font=("Segoe Print",9),
                        bg="#AA0ABC",fg="white")

```

```

    playlist1_label.place(x=130,y=535)

```

```

else:

```

```

    playlist1_label=Label(root,text=opt0[1],font=("Segoe Print",9),bg="#AA0ABC",fg="white")

```

```

    playlist1_label.place(x=130,y=535)

```

```

if len(opt0[2])>10:

```

```

    playlist1_label=Label(root,text=f"{opt0[2][0:10]}....",font=("Segoe Print",9),
                        bg="#AA0ABC",fg="white")

```

```

    playlist1_label.place(x=290,y=535)

```

```

else:

```

```

    playlist1_label=Label(root,text=opt0[2],font=("Segoe Print",9),bg="#AA0ABC",fg="white")

```

```

    playlist1_label.place(x=290,y=535)

```

```

if len(opt0[3])>10:

```

```

    playlist1_label=Label(root,text=f"{opt0[3][0:10]}....",font=("Segoe Print",9),
                        bg="#AA0ABC",fg="white")

```

```

    playlist1_label.place(x=455,y=535)

```

```

else:

```

```

    playlist1_label=Label(root,text=opt0[3],font=("Segoe Print",9),bg="#AA0ABC",fg="white")

```

```

    playlist1_label.place(x=455,y=535)

```

```

global edit02
edit02=PhotoImage(file=r"F:\Spectrum\computerproject\images\edit.png")
edit_playlist1=Button(root,image=edit02,bg='#AA0ABC',borderwidth=0,command=edit_playlist001)
edit_playlist1.place(x=230,y=535)

```

```

edit_playlist2=Button(root,image=edit2,bg='#AA0ABC',borderwidth=0,command=edit_playlist002)
edit_playlist2.place(x=390,y=535)
edit_playlist3=Button(root,image=edit2,bg='#AA0ABC',borderwidth=0,command=edit_playlist003)
edit_playlist3.place(x=555,y=535)

```

TOP CHARTS

```

def topsongs(topsong):
    global homebg
    if topsong=='mallu':
        playlist_id = '37i9dQZF1DX688wU47emR9'
        homebg=ImageTk.PhotoImage(file= r"F:\Spectrum\computerproject\images\malayalam.png")
    elif topsong=='tamil':
        playlist_id = '37i9dQZF1DX1i3hvhZHpCQV'
        homebg=ImageTk.PhotoImage(file= r"F:\Spectrum\computerproject\images\toptm.png")
    elif topsong=='hindi':
        playlist_id = '37i9dQZF1DX0XUfTFmNBRM'
        homebg=ImageTk.PhotoImage(file= r"F:\Spectrum\computerproject\images\tophn.png")
    elif topsong=='english':
        playlist_id = '37i9dQZEVXbMD0HDwVN2tF'
        homebg=ImageTk.PhotoImage(file= r"F:\Spectrum\computerproject\images\topeng.png")

```

```

homebg1=Label(root,image=homebg)
homebg1.place(relheight=1, relwidth=1)
home_icons()

```

```

frame01=customtkinter.CTkScrollableFrame(root,width=710,height=405,fg_color="#1b0024")
frame01.place(x=140,y=135)

```

```

def menudot(btn):
    global search_result
    global tablename
    q006="SELECT COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS
        WHERE TABLE_NAME = '{ }'".format(tablename)
    cur.execute(q006)
    query006=cur.fetchall()
    opt=[]
    for i in query006:
        opt.append(i[0])

    grid_info = btn.grid_info()
    index0 = btn.cget("text")
    index=index0[4::]
    search_result=top_songs_list[int(index)-1]
    def option_selected(event):
        selected_option = clicked.get()
        if selected_option == "Go to Artist":

```



```

        song_name, artist_name = search_song(search_result)
        artist(artist_name)
    elif selected_option == "Add to Favourites":
        q13="insert into {} (favourite) values('{}')".format(tablename,search_result)
        cur.execute(q13)
        my.commit()
        topsongs(topsong)
    else:
        playlist=selected_option[6:]
        q11="insert into {} ({} ) values('{}')".format(tablename,playlist,search_result)
        cur.execute(q11)
        my.commit()
        topsongs(topsong)
clicked=StringVar()
clicked.set("    Song options    ")
options=["Add to Favourites",f"Add to {opt[1]}",f"Add to {opt[2]}",f"Add to {opt[3]}", "Go to Artist"]
dropdown=OptionMenu(frame01,clicked,*options,command=option_selected)
dropdown.config(bg="#310d71", fg="WHITE")
dropdown["menu"].config(bg="#450d71", fg="WHITE")
dropdown.grid(row=grid_info["row"],column=0,sticky='e', padx=5, pady=5)

```

```

i=1
top_songs_list=[]
top_tracks = sp.playlist_tracks(playlist_id, limit=10)
for idx, track in enumerate(top_tracks['items'], start=1):
    track_name = track['track']['name']
    artists = ', '.join([artist['name'] for artist in track['track']['artists']])
    if topsong=='english':
        info=f"{idx}. {track_name} by {artists}"
        if len(info)>50:
            info=f"{info[0:50]}..... "
    else:
        if len(track_name)>50:
            info=f"{idx}. {track_name[0:50]}....."
        else:
            if topsong=='mallu':
                info=f"{idx}. {track_name} "
            else:
                info=f"{idx}. {track_name} "
    label=Label(frame01,text=info,font=("Segoe Print",16),bg="#1b0024",fg="white",anchor='w')
    label.grid(row=i,column=0,sticky='w', padx=5, pady=5)
    label.bind("<Button-1>", songname)
    button=Button(frame01,text=f"●●● {i}",font=("Segoe Print",11),fg="white",
        bg="#1b0024",borderwidth=0)
    button.config(command=lambda b=button: menudot(b))
    button.grid(row=i,column=1,sticky='e', padx=5, pady=5)
    top_songs_list.append(track_name)
    i=i+1

```

```

def topmalayalam():
    global topsong
    topsong='mallu'
    topsongs(topsong)

```

```
def toptamil():
    global topsong
    topsong='tamil'
    topsongs(topsong)
```

```
def topenglish():
    global topsong
    topsong='english'
    topsongs(topsong)
```

```
def tophindi():
    global topsong
    topsong='hindi'
    topsongs(topsong)
```

```
global eng
eng=PhotoImage(file=r"F:\Spectrum\computerproject\images\topenglish.png")
topeng=Button(root,image=eng,bg='#AA0ABC',borderwidth=7,command=topenglish)
topeng.place(x=125,y=115)
```

```
global hindi
hindi=PhotoImage(file=r"F:\Spectrum\computerproject\images\tophindi.png")
tophnd=Button(root,image=hindi,bg='#AA0ABC',borderwidth=7,command=tophindi)
tophnd.place(x=325,y=115)
```

```
global tamil
tamil=PhotoImage(file=r"F:\Spectrum\computerproject\images\toptamil.png")
toptml=Button(root,image=tamil,bg='#AA0ABC',borderwidth=7,command=toptamil)
toptml.place(x=525,y=115)
```

```
global mallu
mallu=PhotoImage(file=r"F:\Spectrum\computerproject\images\topmallu.png")
topmallu=Button(root,image=mallu,bg='#AA0ABC',borderwidth=7,command=topmalayalam)
topmallu.place(x=725,y=115)
```

SUGGESTED FOR YOU

```
def recommendations():
    home_icons()
    global song_list_recom
    frame1=customtkinter.CTkScrollableFrame(root,width=799,height=435,fg_color="#1b0024")
    frame1.place(x=103,y=52)
    global image2
    image2=PhotoImage(file= r"F:\Spectrum\computerproject\images\recom6.png")
    bg_image2=Label(frame1,image=image2,bg="#1b0024")
    bg_image2.grid(row=1,column=1)
    song_list=list()
```

```
for recom in song_list_recom:
    def search_tracks(query):
        results = sp.search(q=query, type='track', limit=10)
        tracks = results['tracks']['items']
        return tracks
```

```

def get_track_features(track_id):
    features = sp.audio_features(tracks=[track_id])
    return features[0]

def get_recommendations(seed_tracks):
    results = sp.recommendations(seed_tracks=seed_tracks, limit=10)
    tracks = results['tracks']
    return tracks
song=recom

if __name__ == "__main__":
    search_query = song
    search_results = search_tracks(search_query)
    if search_results:
        first_track = search_results[0]
        track_id = first_track['id']

        track_features = get_track_features(track_id)
        print("Track Features:")
        print(track_features)

        seed_tracks = [track_id]
        recommendations = get_recommendations(seed_tracks)
        print("\nRecommendations based on the track:")

        for track in recommendations:
            info00=track['name']+ '-' +track['artists'][0]['name']
            song_list.append(info00)
    else:
        print("No tracks found for the search query.")

def songplay(event):
    global play1
    global pause1
    global stop1
    global next1
    global previous1
    global song_result
    play1=PhotoImage(file=r"F:\Spectrum\computerproject\images\play0.png")
    pause1=PhotoImage(file=r"F:\Spectrum\computerproject\images\pause0.png")
    stop1=PhotoImage(file=r"F:\Spectrum\computerproject\images\stop0.png")
    next1=PhotoImage(file=r"F:\Spectrum\computerproject\images\next.png")
    previous1=PhotoImage(file=r"F:\Spectrum\computerproject\images\previous.png")

    clicked_label = event.widget
    song_result1=clicked_label.cget("text")
    song_result2=song_result1.partition(' ')
    song_result=song_result2[2]

def play_time():
    global stopped
    if stopped:
        return

```

```

current_time=pygame.mixer.music.get_pos()/1000
converted_current_time=time.strftime('%M:%S',time.gmtime(current_time))

global song_length
global song0
song=song0
song_mut=MP3(song)
song_length=song_mut.info.length
converted_song_length=time.strftime('%M:%S',time.gmtime(song_length))

current_time+=1
if int(my_slider.get())==int(song_length):
    status_bar2.config(text=converted_song_length)
    global loop_click1
    if loop_click1==1:
        play()
        loop_click1=0
elif paused:
    pass
elif int(my_slider.get())==int(current_time):
    slider_position=int(song_length)
    my_slider.config(to=slider_position,value=int(current_time))
else:
    slider_position=int(song_length)
    my_slider.config(to=slider_position,value=int(my_slider.get()))

converted_current_time=time.strftime('%M:%S',time.gmtime(int(my_slider.get())))
status_bar2.config(text=converted_current_time)
next_time=int(my_slider.get())+1
my_slider.config(value=next_time)

status_bar1.config(text=converted_song_length)
status_bar2.after(1000,play_time)

global paused
paused=False
def pause(is_paused):
    global paused
    paused=is_paused
    if paused:
        pygame.mixer.music.unpause()
        paused=False
        pause_btn=Button(root,image=pause1,bg="#890385",borderwidth=0,command=
            lambda:pause(paused))
        pause_btn.place(x=485,y=500)
    else:
        pygame.mixer.music.pause()
        paused=True
        play_btn=Button(root,image=play1,bg="#890385",borderwidth=0,command=
            lambda:pause(paused))
        play_btn.place(x=485,y=500)

```

```

global stopped
stopped=False
def stop():
    pygame.mixer.music.stop()
    status_bar1.config(text='00:00')
    status_bar2.config(text='00:00')
    my_slider.config(value=0)
    play_btn=Button(root,image=play1,bg="#890385",borderwidth=0,command=play)
    play_btn.place(x=485,y=500)
    global loop
    loop=PhotoImage(file=r"F:\Spectrum\computerproject\images\loop.png")
    song_loop=Button(root,image=loop,bg="#890385",borderwidth=0,command=song_looping1)
    song_loop.place(x=530,y=500)
    global stopped
    stopped=True

def next_song():
    global song_result
    num=song_list.index(song_result)
    song_result=song_list[num+1]
    play()
    song_display()

def prev_song():
    global song_result
    num=song_list.index(song_result)
    song_result=song_list[num-1]
    play()
    song_display()
def play():
    stop()
    global stopped
    global song0
    stopped=False
    song001=f"F:/Spectrum/computerproject/song/[SPOTIFY-DOWNLOADER.COM]
{song_result}.mp3"
    song0=song001.replace('"','_')
    pygame.mixer.music.load(song0)
    pygame.mixer.music.play()
    play_time()
    pause_btn=Button(root,image=pause1,bg="#890385",borderwidth=0,comman=
        lambda:pause(paused))
    pause_btn.place(x=485,y=500)

def slide(X):
    global song0
    global song_length
    pygame.mixer.music.load(song0)
    pygame.mixer.music.play(start=int(my_slider.get()))

my_slider=ttk.Scale(root,from_=0,to=100,orient=HORIZONTAL,value=0,length=400,command=slide)
my_slider.place(x=300,y=538)

```

```

status_bar1=Label(root,text='00:00',font=("Segoe Print",14),bg="#890385",fg="white")
status_bar1.place(x=705,y=532)
status_bar2=Label(root,text='00:00',font=("Segoe Print",14),bg="#890385",fg="white")
status_bar2.place(x=228,y=532)

bar2=Label(root,bg="#890385",width=57,height=3)
bar2.place(x=300,y=497)
bar3=Label(root,bg="#890385",width=57,height=2)
bar3.place(x=300,y=555)

play_btn=Button(root,image=play1,bg="#890385",borderwidth=0,command=play)
play_btn.place(x=485,y=500)
stop_btn=Button(root,image=stop1,bg="#890385",borderwidth=0,command=stop)
stop_btn.place(x=440,y=500)
next_btn=Button(root,image=next1,bg="#890385",borderwidth=0,command=next_song)
next_btn.place(x=577,y=500)

def song_looping1():
    def song_looping2():
        global loop
        global loop_click1
        loop_click1=0
        loop=PhotoImage(file=r"F:\Spectrum\computerproject\images\loop.png")
        song_loop=Button(root,image=loop,bg="#890385",borderwidth=0,command=song_looping1)
        song_loop.place(x=530,y=500)

    global loop1
    loop1=PhotoImage(file=r"F:\Spectrum\computerproject\images\loop1.png")
    song_loop1=Button(root,image=loop1,bg="#890385",borderwidth=0,command=song_looping2)
    song_loop1.place(x=530,y=500)
    global loop_click1
    loop_click1=1

    global loop
    loop=PhotoImage(file=r"F:\Spectrum\computerproject\images\loop.png")
    song_loop=Button(root,image=loop,bg="#890385",borderwidth=0,command=song_looping1)
    song_loop.place(x=530,y=500)

bar=Label(root,text="HHHHHHHHHHHHHHHHHHHHHHHHHHHH",font=("Segoe Print",14),bg="#890385",
          fg="#890385",anchor='w')
bar.place(x=110,y=500)

def song_display():
    if len(song_result)>22:
        label_text=song_result[0:16]+"....."
    else:
        label_text=song_result
    label=Label(root,text=label_text,font=("Segoe Print",14),bg="#890385",fg="white",anchor='w')
    label.place(x=110,y=500)
    label.bind("<Button-1>", songname)

```

```

previous_btn=Button(root,image=previous1,bg="#890385",borderwidth=0,command=prev_song)
previous_btn.place(x=395,y=500)
song_display()

```

```

def menudot(btn):
    global search_result
    global tablename
    q006="SELECT COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS
        WHERE TABLE_NAME = '{}'.format(tablename)
    cur.execute(q006)
    query006=cur.fetchall()
    opt=[]
    for i in query006:
        opt.append(i[0])

    global search_result
    grid_info = btn.grid_info()
    index0 = btn.cget("text")
    index=index0[4::]
    search_result=song_list[int(index)-1]
    def option_selected(event):
        selected_option = clicked.get()
        if selected_option == "Go to song":
            songsearch()
        elif selected_option == "Go to Artist":
            song_name, artist_name = search_song(search_result)
            artist(artist_name)
        elif selected_option == "Add to Favourites":
            q13="insert into {}(favourite) values('{}')".format(tablename,search_result)
            cur.execute(q13)
            my.commit()

        else:
            playlist=selected_option[6::]
            q11="insert into {}({}) values('{}')".format(tablename,playlist,search_result)
            cur.execute(q11)
            my.commit()
    clicked=StringVar()
    clicked.set("      Song options      ")
    options=["Add to Favourites",f"Add to {opt[1]}",f"Add to {opt[2]}",f"Add to {opt[3]}",
            "Go to song","Go to Artist"]
    dropdown=OptionMenu(frame1,clicked,*options,command=option_selected)
    dropdown.config(bg="#310d71", fg="WHITE")
    dropdown["menu"].config(bg="#450d71", fg="WHITE")
    dropdown.grid(row=grid_info["row"],column=1,sticky='e', padx=5, pady=5)

i=1
for recommend_song in song_list:
    if len(recommend_song)>55:
        recommend_song=f"{recommend_song[0:55]}...."
    label=Label(frame1,text=str(i)+' '+str(recommend_song),font=("Segoe Print",16),bg="#1b0024",
        fg="white",anchor='w')

```

```

label.grid(row=8+i,column=1,sticky='w', padx=5, pady=5)
label.bind("<Button-1>", songplay)
button=Button(frame1,text=f"●●● {i} ",font=("Segoe Print",11),fg="white",
               bg="#1b0024",borderwidth=0)
button.config(command=lambda b=button: menudot(b))
button.grid(row=8+i,column=1,sticky='e', padx=5, pady=5)
i=i+1

```

```

get_song_player0()

```

```

global recom00
recom00=Image.open("F:/Spectrum/computerproject/images/recom5.png")
resized=recom00.resize((145, 145),Image.LANCZOS)
photo = ImageTk.PhotoImage(resized)
recom2=Button(root,image=photo,height=145,width=145,borderwidth=7,bg='#AA0ABC',
               command=recommendations)
recom2.image=photo
recom2.place(x=138,y=380)

```

```

global playlist_image_1
playlist_image_1=Image.open(query3[5])
resized1=playlist_image_1.resize((145, 145),Image.LANCZOS)
photo1 = ImageTk.PhotoImage(resized1)
playlist_image_01=Button(root,image=photo1,height=145,width=145,borderwidth=7,
                          bg='#AA0ABC',command=playlist1)
playlist_image_01.image=photo1
playlist_image_01.place(x=335,y=380)

```

```

global playlist_image_2
playlist_image_2=Image.open(query3[6])
resized2=playlist_image_2.resize((145, 145),Image.LANCZOS)
photo2 = ImageTk.PhotoImage(resized2)
playlist_image_02=Button(root,image=photo2,height=145,width=145,borderwidth=7,
                          bg='#AA0ABC',command=playlist2)
playlist_image_02.image=photo2
playlist_image_02.place(x=525,y=380)

```

```

global playlist_image_3
playlist_image_3=Image.open(query3[7])
resized3=playlist_image_3.resize((145, 145),Image.LANCZOS)
photo3 = ImageTk.PhotoImage(resized3)
playlist_image_03=Button(root,image=photo3,height=145,width=145,borderwidth=7,
                          bg='#AA0ABC',command=playlist3)
playlist_image_03.image=photo3
playlist_image_03.place(x=712,y=380)

```

```

q0006="SELECT COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS
      WHERE TABLE_NAME = '{ }'".format(tablename)
cur.execute(q0006)
query0006=cur.fetchall()
opt0=[]

```



```

for i in query0006:
    opt = i[0].replace('_', ' ')
    opt0.append(opt)
bar1=Label(root,text="                ",font=("Segoe Print",12),bg="#AA0ABC",fg="white")
bar1.place(x=340,y=499)
bar2=Label(root,text="                ",font=("Segoe Print",12),bg="#AA0ABC",fg="white")
bar2.place(x=530,y=499)
bar3=Label(root,text="                ",font=("Segoe Print",12),bg="#AA0ABC",fg="white")
bar3.place(x=717,y=499)

```

```

if len(opt0[1])>20:
    playlist1_label=Label(root,text=f"{opt0[1][0:20]}..... ",font=("Segoe Print",12),
                           bg="#AA0ABC",fg="white")
    playlist1_label.place(x=340,y=499)
else:
    playlist1_label=Label(root,text=opt0[1],font=("Segoe Print",12),bg="#AA0ABC",fg="white")
    playlist1_label.place(x=340,y=499)

```

```

if len(opt0[2])>20:
    playlist1_label=Label(root,text=f"{opt0[2][0:20]}..... ",font=("Segoe Print",12),
                           bg="#AA0ABC",fg="white")
    playlist1_label.place(x=530,y=499)
else:
    playlist1_label=Label(root,text=opt0[2],font=("Segoe Print",12),bg="#AA0ABC",fg="white")
    playlist1_label.place(x=530,y=499)

```

```

if len(opt0[3])>20:
    playlist1_label=Label(root,text=f"{opt0[3][0:20]}..... ",font=("Segoe Print",12),
                           bg="#AA0ABC",fg="white")
    playlist1_label.place(x=717,y=499)
else:
    playlist1_label=Label(root,text=opt0[3],font=("Segoe Print",12),bg="#AA0ABC",fg="white")
    playlist1_label.place(x=717,y=499)

```

```

global edit2
edit2=PhotoImage(file=r"F:\Spectrum\computerproject\images\edit.png")
edit_playlist1=Button(root,image=edit2,bg='#AA0ABC',borderwidth=0,command=edit_playlist01)
edit_playlist1.place(x=461,y=500)
edit_playlist2=Button(root,image=edit2,bg='#AA0ABC',borderwidth=0,command=edit_playlist02)
edit_playlist2.place(x=650,y=500)
edit_playlist3=Button(root,image=edit2,bg='#AA0ABC',borderwidth=0,command=edit_playlist03)
edit_playlist3.place(x=840,y=500)

```

#=====SETTINGS PAGE=====

```

def settings_page():
    global email
    global password
    global username
    global notify
    q3="select * from userinfo where email='{ }'".format(email)
    cur.execute(q3)
    query3=cur.fetchone()

```

```

global setbg
setbg=PhotoImage(file=r"F:\Spectrum\computerproject\images\sett1.png")

setbg1=Label(root,image=setbg)
setbg1.place(relheight=1, relwidth=1)
home_icons()

def desc():
    def describe():
        describe1=(descr.get()).strip()
        text1=describe1.split()
        str2=""
        for i in text1:
            if (text1.index(i))%9==0 and (text1.index(i))!=0:
                str2=str2+i+"\n"
            else:
                str2+=i+" "
        q2="update userinfo set description='{ }' where email='{ }' ".format(str2,query3[1])
        cur.execute(q2)
        my.commit()
        settings_page()
    def on_enter(e):
        descr.delete(0,'end')
        descr=Entry(root,width=35,fg='white', border=1, bg='#B932DB',font=("Segoe Print",14))
        descr.place(x=157,y=290)
        if query3[4]==None:
            descr.insert(0,"Add description")
        else:
            descr.insert(0,query3[4])
        descr.bind("<FocusIn>",on_enter)

    global tick1
    tick1=PhotoImage(file=r"F:\Spectrum\computerproject\images\tick1.png")

    edit2=Button(root,image=tick1,bg='#B932DB',borderwidth=0,command=describe)
    edit2.place(x=655,y=350)
    edit=PhotoImage(file=r"F:\Spectrum\computerproject\images\edit.png")
    edit0=customtkinter.CTkLabel(root,text=query3[4],font=("Segoe Print",18),text_color="white",
        bg_color="#B932DB")
    edit0.place(x=157,y=290)
    edit2=Button(root,image=edit,bg='#B932DB',borderwidth=0,command=desc)
    edit2.place(x=655,y=350)

def about_me():
    about=Toplevel()
    about.title("Search")
    about.geometry("925x576")
    about.configure(bg="#1b0024")
    about.resizable(False, False)
    global me
    me=PhotoImage(file=r"F:\Spectrum\computerproject\images\about me.png")
    me1=Label(about,image=me)
    me1.place(relheight=1,relwidth=1)

```

```

global out
out=PhotoImage(file=r"F:\Spectrum\computerproject\images\logout.png")
out1=Button(root,image=out,bg='#B932DB',borderwidth=0,command=Login_page)
out1.place(x=465,y=400)

global more
more=PhotoImage(file=r"F:\Spectrum\computerproject\images\more.png")
more1=Button(root,image=more,bg='#B932DB',borderwidth=0,command=about_me)
more1.place(x=465,y=515)

def delete0():
    global delete_acc
    delete_acc=ImageTk.PhotoImage(file= r"F:\Spectrum\computerproject\images\delete_acc.png")
    delete_acc=Label(root,image=delete_acc)
    delete_acc.place(relheight=1, relwidth=1)
    def delete_account():
        q2="delete from userinfo where email='{ }'".format(query3[1])
        q5="drop table { }".format(query3[0])
        cur.execute(q5)
        cur.execute(q2)
        my.commit()
        Signup_page()
    def cancel_delete():
        settings_page()
    cancel=Button(root,text="Cancel",font=("Acme",13),fg="white",bg="#A048E4",borderwidth=0,
        command=cancel_delete)
    cancel.place(x=355,y=421)
    del_acc=Button(root,text="Delete",font=("Acme",13),fg="white",bg="#A048E4",borderwidth=0,
        command=delete_account)
    del_acc.place(x=550,y=421)
global delete
delete=PhotoImage(file=r"F:\Spectrum\computerproject\images\delete.png")
delete1=Button(root,image=delete,bg='#B932DB',borderwidth=0,command=delete0)
delete1.place(x=465,y=455,)

##### Profile pic #####
global pfp
pfp=PhotoImage(file=r"F:\Spectrum\computerproject\images\pfp.png")
if query3[3]==None:
    profile=Label(root,image=pfp,width=166,height=166,bg="#1b0024")
    profile.place(x=103,y=52)
elif query3[3]!=None:
    image = Image.open(query3[3])
    resized=image.resize((166, 166),Image.LANCZOS)
    photo = ImageTk.PhotoImage(resized)
    label = Label(root, image=photo,height=166,width=166,bg="#B932DB")
    label.image = photo
    label.place(x=103,y=52)

def change_pfp():
    pic= filedialog.askopenfilename(initialdir="Downloads",title="Select a picture",
        filetypes=(("png files", "*.png"),("all files", "*.*")))

```

```

if pic=="":
    pass
else:
    q4="update userinfo set pfp='{ }' where email='{ }'".format(pic,email)
    cur.execute(q4)
    my.commit()
    pfp=Image.open(pic)
    resized=pfp.resize((166, 166),Image.LANCZOS)
    photo = ImageTk.PhotoImage(resized)
    pfp_image_label=Label(root,image=photo,height=166,width=166,bg="#B932DB")
    pfp_image_label.place(x=103,y=52)
    pfp_image_label.image=photo
    edit1=Button(root,image=edit,bg='#1b0024',borderwidth=0,command=change_pfp)
    edit1.place(x=240,y=190)
    bar=customtkinter.CTkLabel(root,text='hihihihihihihihihihihi',font=("Segoe Print",14),
        text_color="#7200a3",bg_color="#7200a3")
    bar.place(x=530,y=190)

```

```

edit1=Button(root,image=edit,bg='#1b0024',borderwidth=0,command=change_pfp)
edit1.place(x=240,y=190)

```

```

global tick2
tick2=PhotoImage(file=r"F:\Spectrum\computerproject\images\tick2.png")

```

```

def change1():
    def change():
        user12=(user.get()).strip()
        if user12==query3[0]:
            messagebox.showerror(title='Error',message='Username already exists')
        elif user12!=query3[0]:
            q2="update userinfo set username='{ }' where email='{ }'".format(user12,query3[1])
            q4="rename table { } to { }".format(query3[0],user12)
            cur.execute(q2)
            cur.execute(q4)
            my.commit()
            messagebox.showinfo(title="Success!",message="Username changed successfully!")
            settings_page()
    def on_enter(e):
        user.delete(0,'end')
        user=Entry(root,width=20,fg='white', border=2, bg='#7200a3',font=("Segoe Print",14))
        user.place(x=350,y=60)
        user.insert(0,query3[0])
        user.bind("<FocusIn>",on_enter)
        name1=Button(root,image=tick2,bg='#7200a3',borderwidth=0,command=change)
        name1.place(x=287,y=58)

```

```

def change2():
    def change():
        global notify
        mail12=(mail.get()).strip()
        q5="select * from userinfo where email='{ }'".format(mail12)
        cur.execute(q5)
        query5=cur.fetchone()

```

```

if mail12.endswith("@gmail.com")==False:
    messagebox.showinfo(title="Email Error",message="Please Enter valid Email id")
elif query5!=None:
    messagebox.showerror(title='Error',message='Email Id already in use')
else:
    q2="update userinfo set email='{ }' where username='{ }'".format(mail12,query3[0])
    cur.execute(q2)
    my.commit()
    messagebox.showinfo(title="Success!",message="Email_ID was changed.\nPlease Sign in again.")
    notify=2
    Login_page()
def on_enter(e):
    mail.delete(0,'end')
mail=Entry(root,width=20,fg='white', border=2, bg='#7200a3',font=("Segoe Print",14))
mail.place(x=350,y=112)
mail.insert(0,query3[1])
mail.bind("<FocusIn>",on_enter)
mail0=Button(root,image=tick2,bg='#7200a3',borderwidth=0,command=change)
mail0.place(x=287,y=112)

def change3():
    def change():
        global notify
        passwd12=(passwd.get()).strip()
        if len(passwd12)<8:
            messagebox.showerror(title='Error',message='Password must be minimum 8 characters')
        elif len(passwd12)>=8 and passwd12==query3[2]:
            messagebox.showerror(title='Error',message='Please keep a different password')
        else:
            q2="update userinfo set password='{ }' where email='{ }'".format(passwd12,query3[1])
            cur.execute(q2)
            my.commit()
            messagebox.showinfo(title="Success!",message="Password changed successfully!")
            notify=1
            settings_page()

def on_enter(e):
    passwd.delete(0,'end')
    passwd.configure(show='●')
    global hide_pswrd
    global unhide_pswrd
    hide_pswrd=PhotoImage(file= r"F:\Spectrum\computerproject\images\hide.png")
    unhide_pswrd=PhotoImage(file= r"F:\Spectrum\computerproject\images\unhide.png")
    def show():
        passwd.configure(show="")
    def hide():
        passwd.configure(show='●')
        hide_label=Button(root,image=hide_pswrd,bg='#7200a3',command=show)
        hide_label.place(x=600,y=165)
        unhide_label=Button(root,image=unhide_pswrd,bg='#7200a3',command=hide)
        unhide_label.place(x=600,y=165)
    hide_label=Button(root,image=hide_pswrd,bg='#7200a3',command=show)
    hide_label.place(x=600,y=165)

```

```

passwd=Entry(root,width=20,fg='white', border=2, bg='#7200a3',font=("Segoe Print",14))
passwd.place(x=350,y=165)
passwd.insert(0,query3[2])
passwd.bind("<FocusIn>",on_enter)
pass3=Button(root,image=tick2,bg='#7200a3',borderwidth=0,command=change)
pass3.place(x=287,y=160)

```

```

def see1():
    def back():
        str=query3[1].replace("gmail.com","")
        str1=""
        for i in str:
            str1+="*"
        email3=customtkinter.CTkLabel(root,text=str1+"@gmail.com",font=("Segoe Print",24),
            text_color="white",bg_color="#7200a3")
        email3.place(x=350,y=110)
        email4=Button(root,image=email0,bg='#7200a3',borderwidth=0,command=see1)
        email4.place(x=287,y=110)
    email3=customtkinter.CTkLabel(root,text=query3[1],font=("Segoe Print",24),
        text_color="white",bg_color="#7200a3")
    email3.place(x=350,y=110)
    email4=Button(root,image=email0,bg='#7200a3',borderwidth=0,command=back)
    email4.place(x=287,y=110)

```

```

global name0
name0=PhotoImage(file=r"F:\Spectrum\computerproject\images\user.png")
name=customtkinter.CTkLabel(root,text=query3[0],font=("Segoe Print",24),
    text_color="white",bg_color="#7200a3")
name.place(x=350,y=58)
name1=Label(root,image=name0,bg='#7200a3',borderwidth=0)
name1.place(x=287,y=58)
edit01=Button(root,image=edit,bg='#D60CDA',borderwidth=0,command=change1)
edit01.place(x=898,y=60)

```

```

global email0
str=query3[1].replace("gmail.com","")
str1=""
for i in str:
    str1+="*"
email0=PhotoImage(file=r"F:\Spectrum\computerproject\images\email.png")
email3=customtkinter.CTkLabel(root,text=str1+"@gmail.com",font=("Segoe Print",24),
    text_color="white",bg_color="#7200a3")
email3.place(x=350,y=110)
email4=Button(root,image=email0,bg='#7200a3',borderwidth=0,command=see1)
email4.place(x=287,y=110)
edit02=Button(root,image=edit,bg='#D60CDA',borderwidth=0,command=change2)
edit02.place(x=898,y=110)

```

```

global pass0
str=""
for i in query3[2]:
    str+="*"

```

```

pass0=PhotoImage(file=r"F:\Spectrum\computerproject\images\pswr.png")
passwd=customtkinter.CTkLabel(root,text=str,font=("Segoe Print",24),
    text_color="white",bg_color="#7200a3")
passwd.place(x=350,y=165)
pass1=Button(root,image=pass0,bg='#7200a3',borderwidth=0)
pass1.place(x=287,y=160)
edit03=Button(root,image=edit,bg='#D60CDA',borderwidth=0,command=change3)
edit03.place(x=898,y=162)

```

#=====SONG details=====

```

def search_song(query):
    results = sp.search(q=query, limit=1)
    if results['tracks']['items']:
        track = results['tracks']['items'][0]
        song_name = track['name']
        artist_name = track['artists'][0]['name']
        print(f"Found: {song_name} by {artist_name}")
        return (song_name, artist_name)
    else:
        print("No song found.")

def get_lyrics(song_name, artist_name):
    genius = lyricsgenius.Genius('-miTWfHrybeOUktjfVmrjIAzvpp4D8QOBKAwS
        8VPZhiTb_LD6dnBldgersVnbYQppebena')
    song = genius.search_song(song_name, artist_name)
    frame1=customtkinter.CTkScrollableFrame(root,width=590,height=435,fg_color="#1b0024")
    frame1.place(x=314,y=52)

    if song:
        lyrics0 = song.lyrics
        lyrics1=lyrics0.partition(song_name+" Lyrics")
        lyrics2 = lyrics1[2]
        endstr = re.findall("\d*Embed$", lyrics2)
        if len(endstr)==0:
            lyricstitle=Label(frame1,text="Lyrics currently unavailable",font=("Segoe Print",16),
                bg="#1b0024",fg="white")
            lyricstitle.pack(pady=10)
        else:
            endindex = lyrics2.find(endstr[0])
            global lyrics_f
            lyrics_f = lyrics2[0:endindex]
            lyricstitle=Label(frame1,text=lyrics_f,font=("Segoe Print",14),bg="#1b0024",fg="white")
            lyricstitle.pack(pady=10)
    else:
        lyrics_f = "Lyrics currently unavailable.\nWill be updated soon"
        lyricstitle=Label(frame1,text=lyrics_f,font=("Segoe Print",16),bg="#1b0024",fg="white")
        lyricstitle.pack(pady=10)

def get_cover_image_url(song_name):
    result = sp.search(q=song_name, type='track', limit=1)
    if result['tracks']['items']:

```

```

def display_image_from_url(url):
    response = requests.get(url)
    image_data = response.content
    image = Image.open(BytesIO(image_data))
    resized=image.resize((220, 220),Image.LANCZOS)
    photo = ImageTk.PhotoImage(resized)
    label = Label(root, image=photo)
    label.image = photo
    label.place(x=102,y=52)
    image_url = result['tracks']['items'][0]['album']['images'][0]['url']
    display_image_from_url(image_url)
else:
    return "Cover image not found."

```

```

def artist(artist_name):
    frame1=customtkinter.CTkScrollableFrame(root,width=300,height=210,fg_color="#1b0024")
    frame1.place(x=102,y=355)
    global image2
    image2=PhotoImage(file= r"F:\Spectrum\computerproject\images\bg2.png")
    bg_image2=Label(frame1,image=image2)
    bg_image2.pack(pady=10)
    frame2=customtkinter.CTkScrollableFrame(root,width=499,height=516,fg_color="#1b0024")
    frame2.place(x=406,y=52)

```

```

def get_artist_image_url(artist_name):
    result = sp.search(q=artist_name, type='artist', limit=1)
    if result['artists']['items']:
        def display_image_from_url(url):
            response = requests.get(url)
            image_data = response.content

            image = Image.open(BytesIO(image_data))
            resized=image.resize((300, 300),Image.LANCZOS)
            photo = ImageTk.PhotoImage(resized)
            label = Label(root, image=photo)
            label.image = photo
            label.place(x=102,y=52)
            artist_url = result['artists']['items'][0]['images'][0]['url']
            display_image_from_url(artist_url)
        else:
            return "Artist image not found."

```

```

def get_artist_info(artist_name):
    result = sp.search(q=artist_name, type='artist', limit=1)
    if result['artists']['items']:
        artist = result['artists']['items'][0]
        artist_id = artist['id']
        top_tracks = sp.artist_top_tracks(artist_id)
        Name= artist['name']
        Genres=artist['genres']
        genres=""

```



```

for i in Genres:
    genres+=i+', '
Followers= artist['followers']['total']
Popularity=artist['popularity']

info1=Label(frame2,text="Artist: "+Name,font=("Segoe Print",15),
            bg="#1b0024",fg="white",anchor='w')
info1.grid(row=2,column=1,sticky='nswe')
info2=Label(frame2,text="Genres: "+genres[0:32],font=("Segoe Print",15),
            bg="#1b0024",fg="white",anchor='w')
info2.grid(row=3,column=1,sticky='nswe')
info3=Label(frame2,text="Followers: "+str(Followers),font=("Segoe Print",15),
            bg="#1b0024",fg="white",anchor='w')
info3.grid(row=5,column=1,sticky='nswe')
info4=Label(frame2,text="Popularity: "+str(Popularity),font=("Segoe Print",15),
            bg="#1b0024",fg="white",anchor='w')
info4.grid(row=6,column=1,sticky='nswe')

bar2=Label(frame2,text="Popularity: "+str(Popularity),font=("Segoe Print",15),
            bg="#1b0024",fg="#1b0024",anchor='w')
bar2.grid(row=7,column=1,sticky='nswe')

Top=Label(frame2,text="Popular songs:",font=("Britannic Bold",20),
            bg="#1b0024",fg="white",anchor='w')
Top.grid(row=8,column=1,sticky='nswe')

def songsearch(event):
    global play
    global pause
    global add
    global fav1
    play=PhotoImage(file=r"F:\Spectrum\computerproject\images\play0.png")
    pause=PhotoImage(file=r"F:\Spectrum\computerproject\images\pause0.png")
    add=PhotoImage(file=r"F:\Spectrum\computerproject\images\add.png")
    fav1=PhotoImage(file=r"F:\Spectrum\computerproject\images\heart2.png")
    clicked_label = event.widget
    search_result1=clicked_label.cget("text")
    search_result2=search_result1.partition(' ')
    search_result=search_result2[2]
    song_name, artist_name = search_song(search_result)
    if song_name and artist_name:
        get_lyrics(song_name, artist_name)
        get_cover_image_url(song_name)
        get_song_info(song_name)
        get_song_fav(song_name)
        get_song_player(song_name)

i=1
for track in top_tracks['tracks']:
    label=Label(frame2,text=str(i)+'.' +track['name'][0:42],font=("Segoe Print",14),bg="#1b0024",
                fg="white",anchor='w')
    label.grid(row=8+i,column=1,sticky='nswe')

```

```

        label.bind("<Button-1>", songsearch)
        i=i+1
    else:
        return "Artist not found."

def artist_fav(artist_name):
    global fav1
    global fav2
    fav2=PhotoImage(file=r"F:\Spectrum\computerproject\images\heart3.png")
    fav1=PhotoImage(file=r"F:\Spectrum\computerproject\images\heart2.png")
    q06="select * from userinfo where email='{ }'".format(email)
    cur.execute(q06)
    query06=cur.fetchone()
    def artist_favourite():
        q6="select * from userinfo where email='{ }'".format(email)
        cur.execute(q6)
        query6=cur.fetchone()
        q7="insert into { }(following) values('{ } )'".format(query6[0],artist_name)
        cur.execute(q7)
        my.commit()
    def remove():
        fav1_btn=Button(frame2,image=fav1,bg="#1b0024",borderwidth=0,command=artist_favourite)
        fav1_btn.grid(row=2,column=1,sticky='ne')
        q8="update { } set following=NULL where following='{ }'".format(query6[0],artist_name)
        cur.execute(q8)
        my.commit()
        fav2_btn=Button(frame2,image=fav2,bg="#1b0024",borderwidth=0,command=remove)
        fav2_btn.grid(row=2,column=1,sticky='ne')
    q9="select*from { } where following='{ }'".format(query06[0],artist_name)
    cur.execute(q9)
    query9=cur.fetchone()

    def remove():
        global fav1
        fav1_btn=Button(frame2,image=fav1,bg="#1b0024",borderwidth=0,command=artist_favourite)
        fav1_btn.grid(row=2,column=1,sticky='ne')
        q8="update { } set following=NULL where following='{ }'".format(query06[0],artist_name)
        cur.execute(q8)
        my.commit()
    if query9==None:
        fav1_btn=Button(frame2,image=fav1,bg="#1b0024",borderwidth=0,command=artist_favourite)
        fav1_btn.grid(row=2,column=1,sticky='ne')
    else:
        fav2=PhotoImage(file=r"F:\Spectrum\computerproject\images\heart3.png")
        fav2_btn=Button(frame2,image=fav2,bg="#1b0024",borderwidth=0,command=remove)
        fav2_btn.grid(row=2,column=1,sticky='ne')

get_artist_image_url(artist_name)
get_artist_info(artist_name)
artist_fav(artist_name)

```

```

def get_song_info(song_name):
    result = sp.search(q=song_name, type='track', limit=1)
    if result['tracks']['items']:
        track = result['tracks']['items'][0]
        song= track['name']
        artist_name= ', '.join([artist['name'] for artist in track['artists']])
        release_date= track['album']['release_date']
        popularity= track['popularity']

    def ARTISTS(event):
        artist(artist_name)

    frame2=customtkinter.CTkScrollableFrame(root,width=205,height=200,fg_color="#1b0024")
    frame2.place(x=102,y=305)
    info1=Label(frame2,text="Song: "+song,font=("Segoe Print",10),bg="#1b0024",
        fg="white",anchor='w')
    info1.grid(row=2,column=1,sticky='nswe')

    info2=Label(frame2,text="Artist: "+artist_name,font=("Segoe Print",10),bg="#1b0024",
        fg="white",anchor='w')
    info2.grid(row=3,column=1,sticky='nswe')
    info2.bind("<Button-1>", ARTISTS)

    info4=Label(frame2,text="Release date: "+str(release_date),font=("Segoe Print",10),bg="#1b0024",
        fg="white",anchor='w')
    info4.grid(row=5,column=1,sticky='nswe')
    info5=Label(frame2,text="Popularity: "+str(popularity),font=("Segoe Print",10),bg="#1b0024",
        fg="white",anchor='w')
    info5.grid(row=6,column=1,sticky='nswe')

    else:
        return "Song not found."

def get_song_fav(song_name):
    global play10
    global pause10
    global add
    global fav1
    global fav2
    bar=Label(root,bg="#1b0024",width=30,height=2)
    bar.place(x=102,y=276)
    add=PhotoImage(file=r"F:\Spectrum\computerproject\images\add.png")
    play10=PhotoImage(file=r"F:\Spectrum\computerproject\images\play0.png")
    pause10=PhotoImage(file=r"F:\Spectrum\computerproject\images\pause0.png")
    fav2=PhotoImage(file=r"F:\Spectrum\computerproject\images\heart3.png")
    fav1=PhotoImage(file=r"F:\Spectrum\computerproject\images\heart2.png")

    q06="select * from userinfo where email='{ }'.format(email)
    cur.execute(q06)
    query06=cur.fetchone()

```

```

def add01():
    global tablename
    q006="SELECT COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS
        WHERE TABLE_NAME = '{}'.format(tablename)

    cur.execute(q006)
    query006=cur.fetchall()
    options=[]
    for i in query006:
        options.append(i[0][0:18])

    def option_selected(event):
        selected_option = clicked.get()
        q11="insert into {}({}) values('{}')".format(query06[0],selected_option,song_name)
        cur.execute(q11)
        my.commit()
        bar=Label(root,bg="#1b0024",width=20,height=2)
        bar.place(x=115,y=279)
        add_btn=Button(root,image=add,bg="#1b0024",borderwidth=0,command=add01)
        add_btn.place(x=240,y=277)

    clicked=StringVar()
    clicked.set("Add Song")
    dropdown=OptionMenu(root,clicked,options[1],options[2],options[3],command=option_selected)
    dropdown.config(bg="#1b0024", fg="WHITE")
    dropdown.place(x=115,y=279)
    add_btn=Button(root,image=add,bg="#1b0024",borderwidth=0,command=add01)
    add_btn.place(x=240,y=277)

def favourite():
    global fav2
    global query6
    fav2=PhotoImage(file=r"F:\Spectrum\computerproject\images\heart3.png")
    fav1=PhotoImage(file=r"F:\Spectrum\computerproject\images\heart2.png")
    q6="select * from userinfo where email='{}'.format(email)
    cur.execute(q6)
    query6=cur.fetchone()

    q7="insert into {}(favourite) values('{}')".format(query6[0],song_name)
    cur.execute(q7)
    my.commit()

    def remove():
        fav1_btn=Button(root,image=fav1,bg="#1b0024",borderwidth=0,command=favourite)
        fav1_btn.place(x=280,y=277)
        q8="update {} set favourite=NULL where favourite='{}'.format(query6[0],song_name)
        cur.execute(q8)
        my.commit()
        fav2_btn=Button(root,image=fav2,bg="#1b0024",borderwidth=0,command=remove)
        fav2_btn.place(x=280,y=277)
    song_name=song_name.replace("","")
    q9="select*from {} where favourite='{}'.format(query06[0],song_name)
    cur.execute(q9)

```

```

query9=cur.fetchone()
def remove():
    global fav1
    fav1_btn=Button(root,image=fav1,bg="#1b0024",borderwidth=0,command=favourite)
    fav1_btn.place(x=280,y=277)

    q8="update { } set favourite=NULL where favourite='{ }' ".format(query06[0],song_name)
    cur.execute(q8)
    my.commit()
if query9==None:
    fav1_btn=Button(root,image=fav1,bg="#1b0024",borderwidth=0,command=favourite)
    fav1_btn.place(x=280,y=277)
else:
    fav2=PhotoImage(file=r"F:\Spectrum\computerproject\images\heart3.png")
    fav2_btn=Button(root,image=fav2,bg="#1b0024",borderwidth=0,command=remove)
    fav2_btn.place(x=280,y=277)

def get_song_player(song_name):
    global play1
    global pause1
    global stop1
    play1=PhotoImage(file=r"F:\Spectrum\computerproject\images\play0.png")
    pause1=PhotoImage(file=r"F:\Spectrum\computerproject\images\pause0.png")
    stop1=PhotoImage(file=r"F:\Spectrum\computerproject\images\stop0.png")
    bar=Label(root,bg="#890385",width=500,height=5)
    bar.place(x=102,y=497)
    global loop_click1
    global loop_click2
    loop_click1=0
    loop_click2=0

def play_time():
    global stopped
    if stopped:
        return
    current_time=pygame.mixer.music.get_pos()/1000
    converted_current_time=time.strftime('%M:%S',time.gmtime(current_time))

    global song_length
    global song0
    song=song0
    song_mut=MP3(song)
    song_length=song_mut.info.length
    converted_song_length=time.strftime('%M:%S',time.gmtime(song_length))

    current_time+=1
    if int(my_slider.get())==int(song_length):
        status_bar2.config(text=converted_song_length)
        global loop_click1
        if loop_click1==1:
            play()
            loop_click1=0

```

```

elif paused:
    pass
elif int(my_slider.get())==int(current_time):
    slider_position=int(song_length)
    my_slider.config(to=slider_position,value=int(current_time))
else:
    slider_position=int(song_length)
    my_slider.config(to=slider_position,value=int(my_slider.get()))

    converted_current_time=time.strftime('%M:%S',time.gmtime(int(my_slider.get())))
    status_bar2.config(text=converted_current_time)

    next_time=int(my_slider.get()+1)
    my_slider.config(value=next_time)

status_bar1.config(text=converted_song_length)
status_bar2.after(1000,play_time)

```

```

global paused
paused=False
def pause(is_paused):
    global paused
    paused=is_paused
    if paused:
        pygame.mixer.music.unpause()
        paused=False
        pause_btn=Button(root,image=pause1,bg="#890385",borderwidth=0,command=
            lambda:pause(paused))
        pause_btn.place(x=445,y=500)
    else:
        pygame.mixer.music.pause()
        paused=True
        play_btn=Button(root,image=play1,bg="#890385",borderwidth=0,command=lambda:pause(paused))
        play_btn.place(x=445,y=500)

```

```

global stopped
stopped=False
def stop():
    pygame.mixer.music.stop()
    status_bar1.config(text='00:00')
    status_bar2.config(text='00:00')
    my_slider.config(value=0)
    play_btn=Button(root,image=play1,bg="#890385",borderwidth=0,command=play)
    play_btn.place(x=445,y=500)
    global loop
    loop=PhotoImage(file=r"F:\Spectrum\computerproject\images\loop.png")
    song_loop=Button(root,image=loop,bg="#890385",borderwidth=0,command=song_looping1)
    song_loop.place(x=530,y=500)
    global stopped
    stopped=True

```

```

def play():
    stop()
    global stopped
    stopped=False
    global song0
    song001=f"F:/Spectrum/computerproject/song/[SPOTIFY-DOWNLOADER.COM]
{song_name}.mp3"
    song0=song001.replace('"','_')
    pygame.mixer.music.load(song0)
    pygame.mixer.music.play()
    play_time()
    pause_btn=Button(root,image=pause1,bg="#890385",borderwidth=0,command=lambda:pause(paused))
    pause_btn.place(x=445,y=500)

def slide(X):
    global song0
    global song_length
    pygame.mixer.music.load(song0)
    pygame.mixer.music.play(start=int(my_slider.get()))

my_slider=ttk.Scale(root,from_=0,to=100,orient=HORIZONTAL,value=0,length=400,command=slide)
my_slider.place(x=300,y=538)
status_bar1=Label(root,text='00:00',font=("Segoe Print",14),bg="#890385",fg="white")
status_bar1.place(x=705,y=532)
status_bar2=Label(root,text='00:00',font=("Segoe Print",14),bg="#890385",fg="white")
status_bar2.place(x=228,y=532)

bar2=Label(root,bg="#890385",width=57,height=3)
bar2.place(x=300,y=497)
bar3=Label(root,bg="#890385",width=57,height=2)
bar3.place(x=300,y=555)

play_btn=Button(root,image=play1,bg="#890385",borderwidth=0,command=play)
play_btn.place(x=445,y=500)
stop_btn=Button(root,image=stop1,bg="#890385",borderwidth=0,command=stop)
stop_btn.place(x=485,y=500)

def song_looping1():
    def song_looping2():
        global loop_click1
        global loop
        loop_click1=0
        loop=PhotoImage(file=r"F:\Spectrum\computerproject\images\loop.png")
        song_loop=Button(root,image=loop,bg="#890385",borderwidth=0,command=song_looping1)
        song_loop.place(x=530,y=500)

    global loop1
    loop1=PhotoImage(file=r"F:\Spectrum\computerproject\images\loop1.png")
    song_loop1=Button(root,image=loop1,bg="#890385",borderwidth=0,command=song_looping2)
    song_loop1.place(x=530,y=500)
    global loop_click1
    loop_click1=1

```

```

global loop
loop=PhotoImage(file=r"F:\Spectrum\computerproject\images\loop.png")
song_loop=Button(root,image=loop,bg="#890385",borderwidth=0,command=song_looping1)
song_loop.place(x=530,y=500)

```

```

#=====SEARCH PAGE=====

```

```

def search_page():
    global searchbg
    searchbg=PhotoImage(file=r"F:\Spectrum\computerproject\images\searchbg.png")
    bgsearch=Label(root,image=searchbg)
    bgsearch.place(relheight=1,relwidth=1)

```

```

global paused

```

```

paused=False

```

```

def play():

```

```

    def choose_random_file(folder_path):

```

```

        files = os.listdir(folder_path)

```

```

        if files:

```

```

            return os.path.join(folder_path, random.choice(files))

```

```

        else:

```

```

            return None

```

```

    def open_random_file():

```

```

        folder_path = 'F:/Spectrum/computerproject/song'

```

```

        print (folder_path)

```

```

        if folder_path:

```

```

            global random_file

```

```

            random_file = choose_random_file(folder_path)

```

```

            print(random_file)

```

```

        else:

```

```

            print("No folder selected.")

```

```

    open_random_file()

```

```

    song=str(random_file)

```

```

    pygame.mixer.music.load(song)

```

```

    pygame.mixer.music.play()

```

```

def pause(is_paused):

```

```

    global paused

```

```

    paused=is_paused

```

```

    if paused:

```

```

        pygame.mixer.music.unpause()

```

```

        paused=False

```

```

    else:

```

```

        pygame.mixer.music.pause()

```

```

        paused=True

```

```

global music

```

```

music=PhotoImage(file=r"F:\Spectrum\computerproject\images\music.png")

```

```

music12=Button(root,image=music,bg='#C325DD',borderwidth=0,command=play)

```

```

music12.place(x=400,y=480)

```



```
global plypause
plypause=PhotoImage(file=r"F:\Spectrum\computerproject\images\playpause10.png")
playpse=Button(root,image=plypause,bg='#C325DD',borderwidth=0,command=lambda:pause(paused))
playpse.place(x=550,y=483)
home_icons()

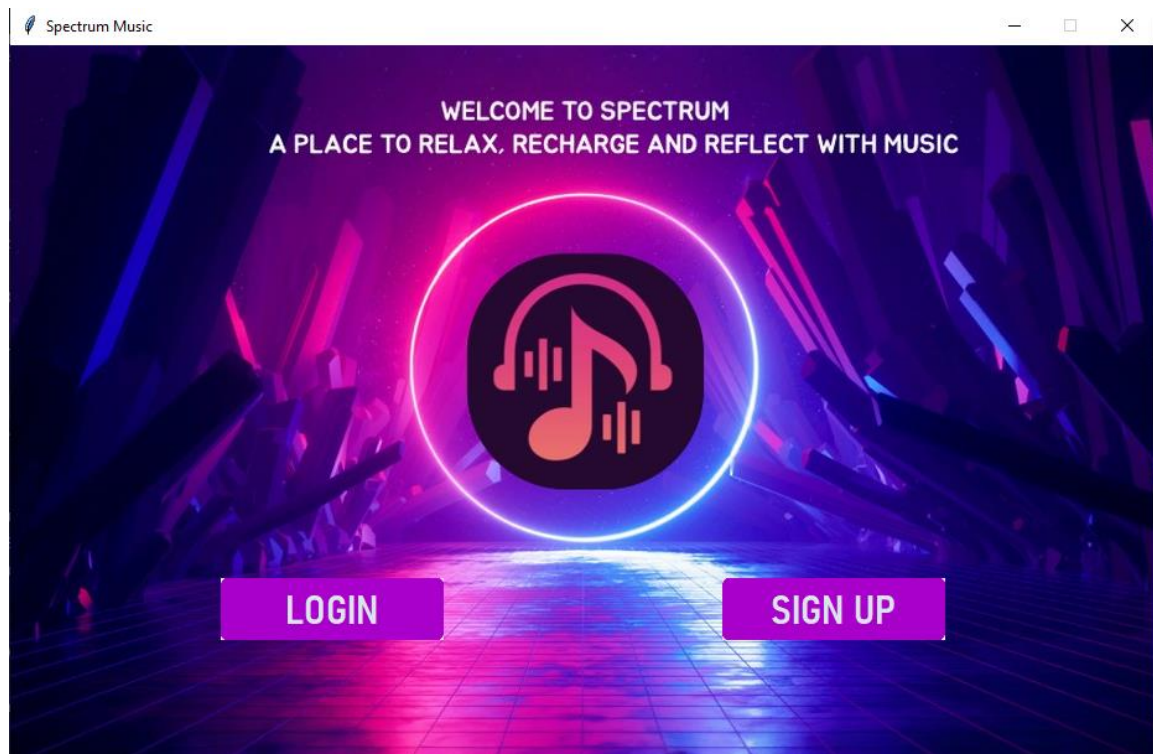
def notifications():
    messagebox.showinfo(title="Notifications",message="No New Notifications")
titlebar=Label(root,bg="#151515",width=120,height=3)
titlebar.place(x=102,y=1)
global menu4
menu4=PhotoImage(file=r"F:\Spectrum\computerproject\images\notifications.png")
logo4=Button(root,image=menu4,bg='#7200a3',command=notifications,borderwidth=0)
logo4.place(x=875,y=0)

search_requirements()
home_icons()
root.mainloop()
```

SAMPLE OUTPUTS

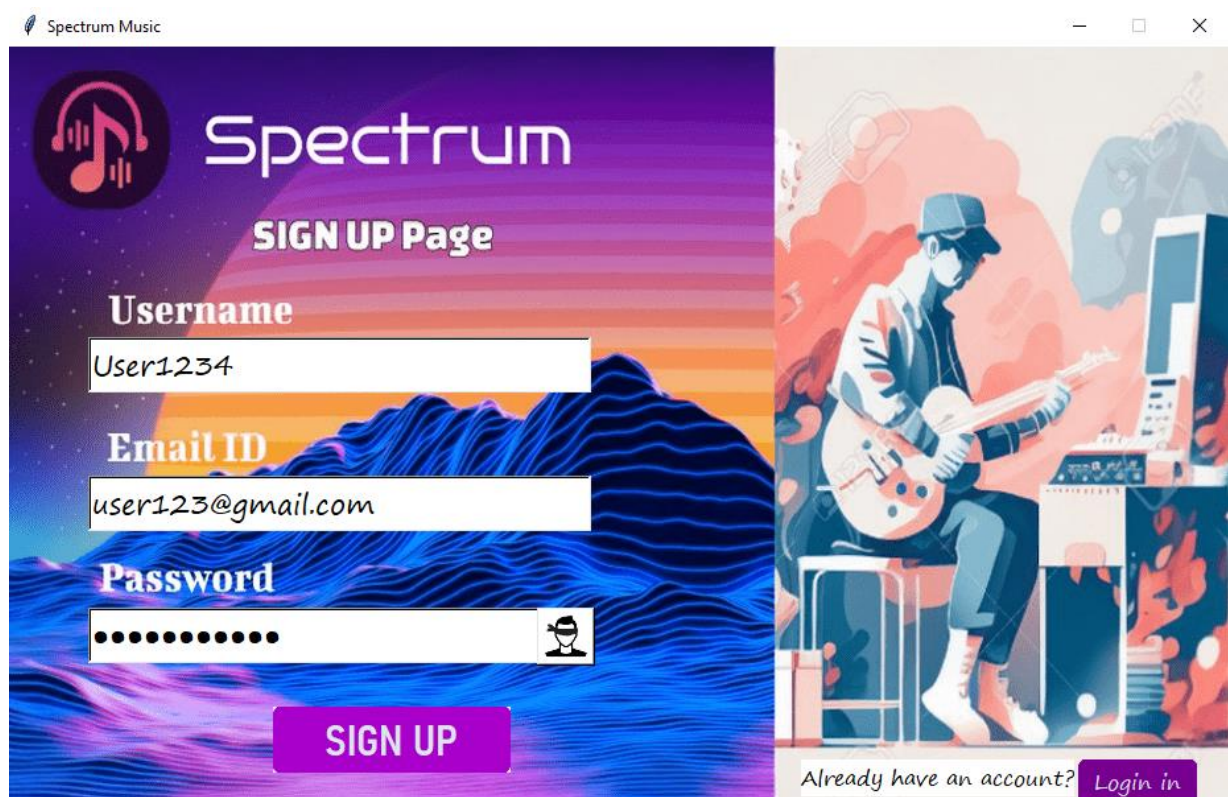
Start page:

When the app opens up, you can either signup and create a new account or log in to your existing spectrum account.

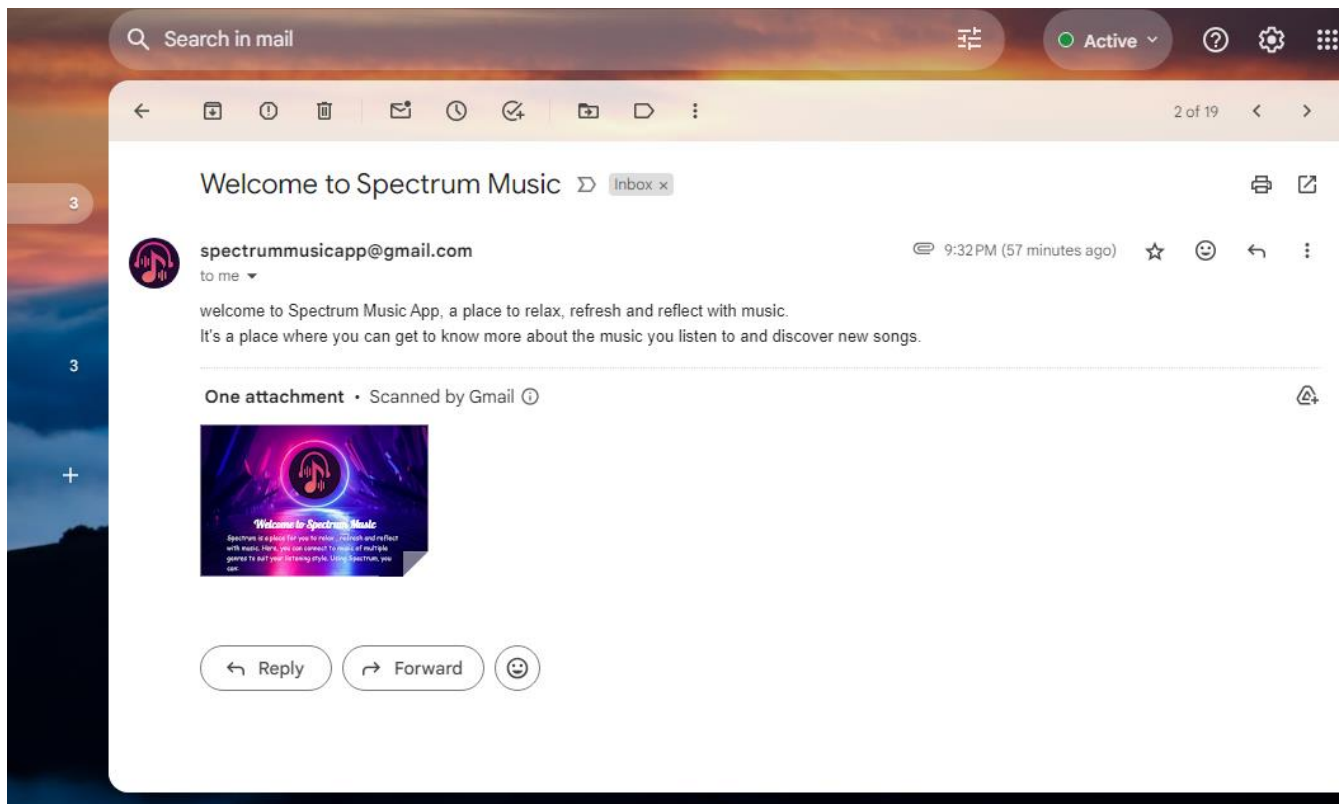


Sign up page:

Sign up to create a spectrum account and listen to music.

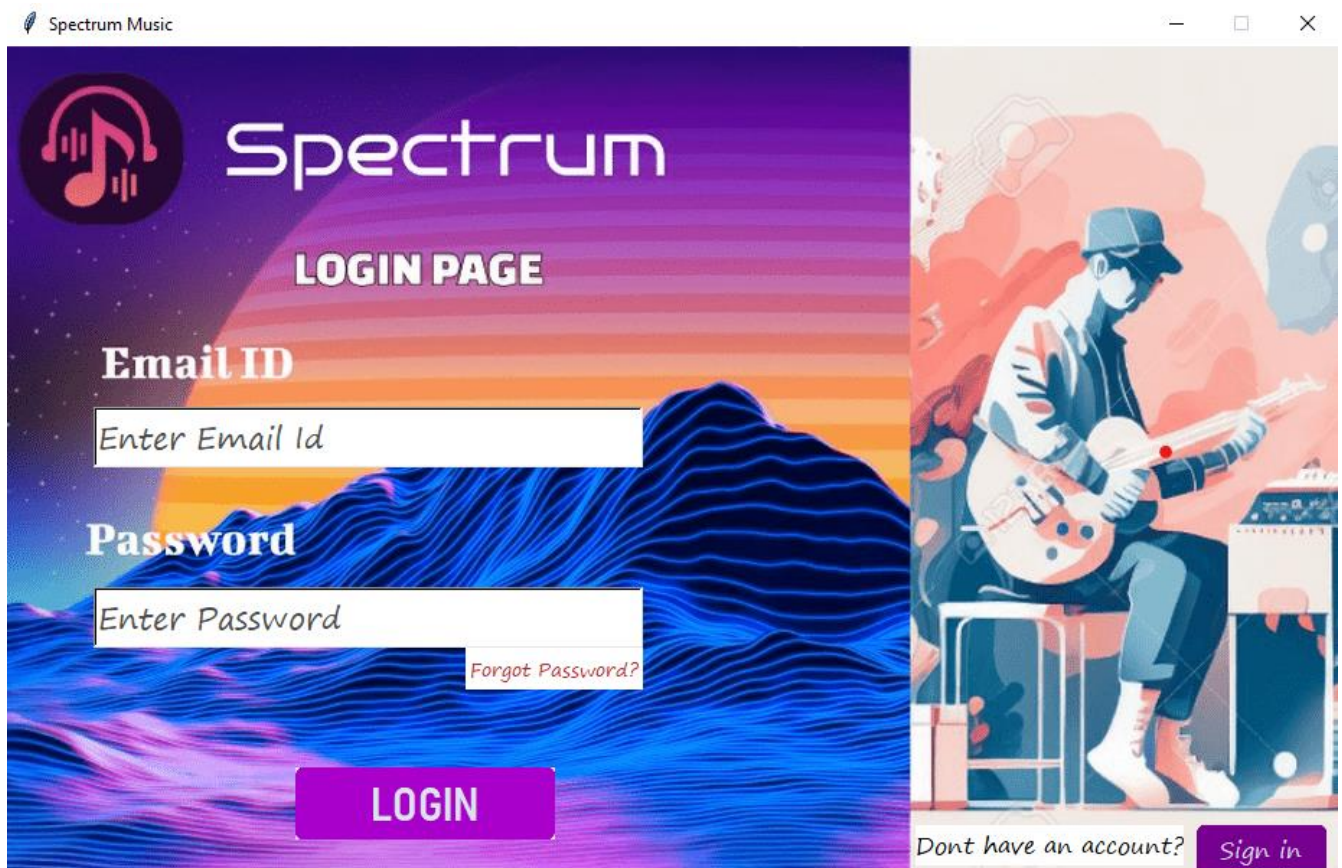


When you sign up into spectrum music, you would receive a welcome message to you email id you used for creating a spectrum music account



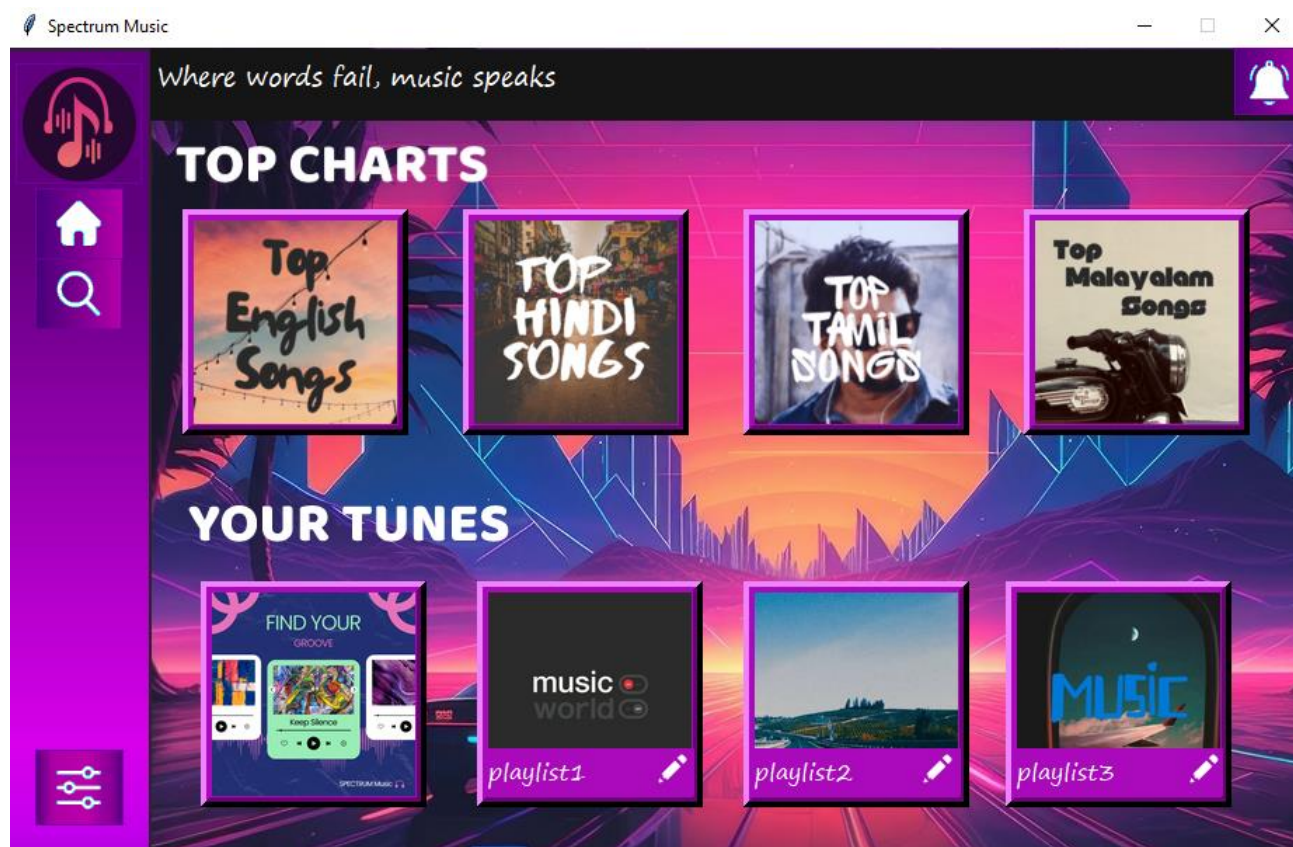
Login page:

This is the login page, enter the correct details of your account to sign in.

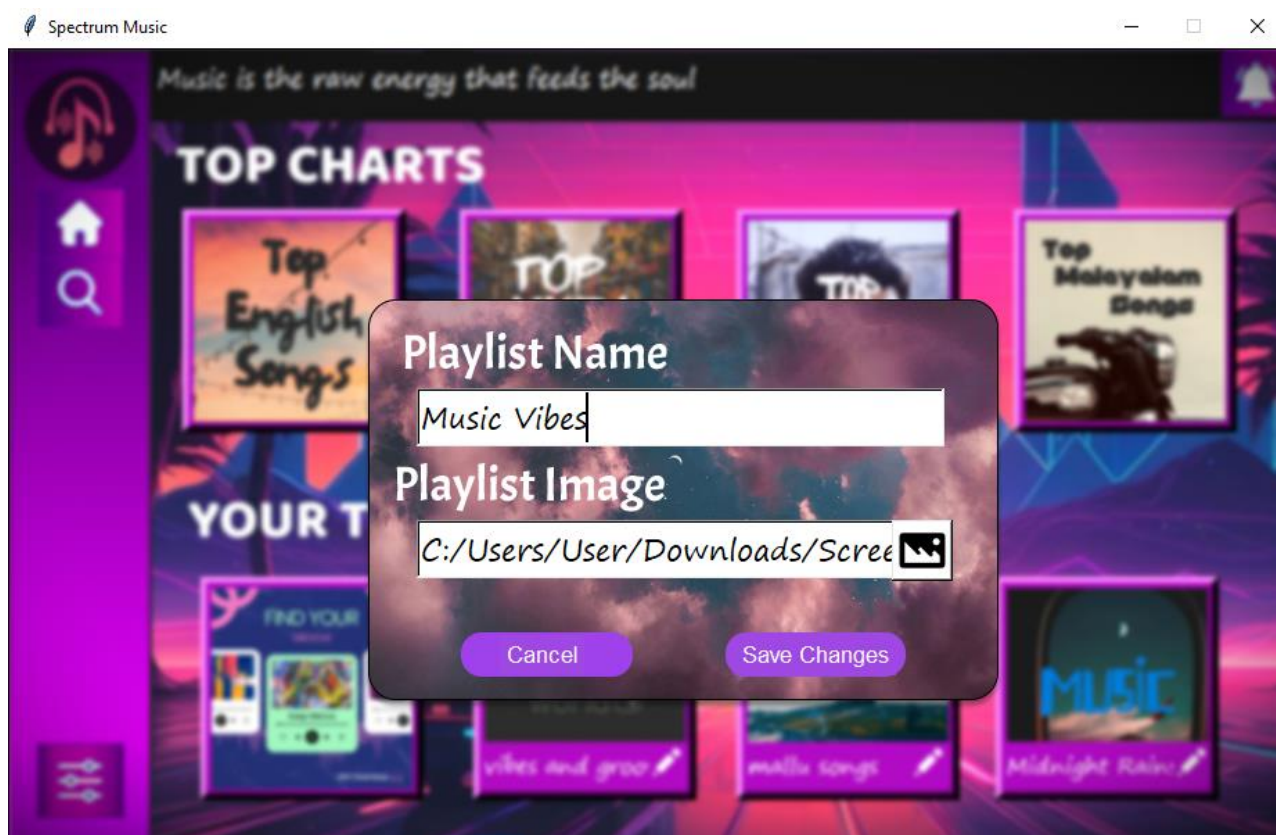


Home page:

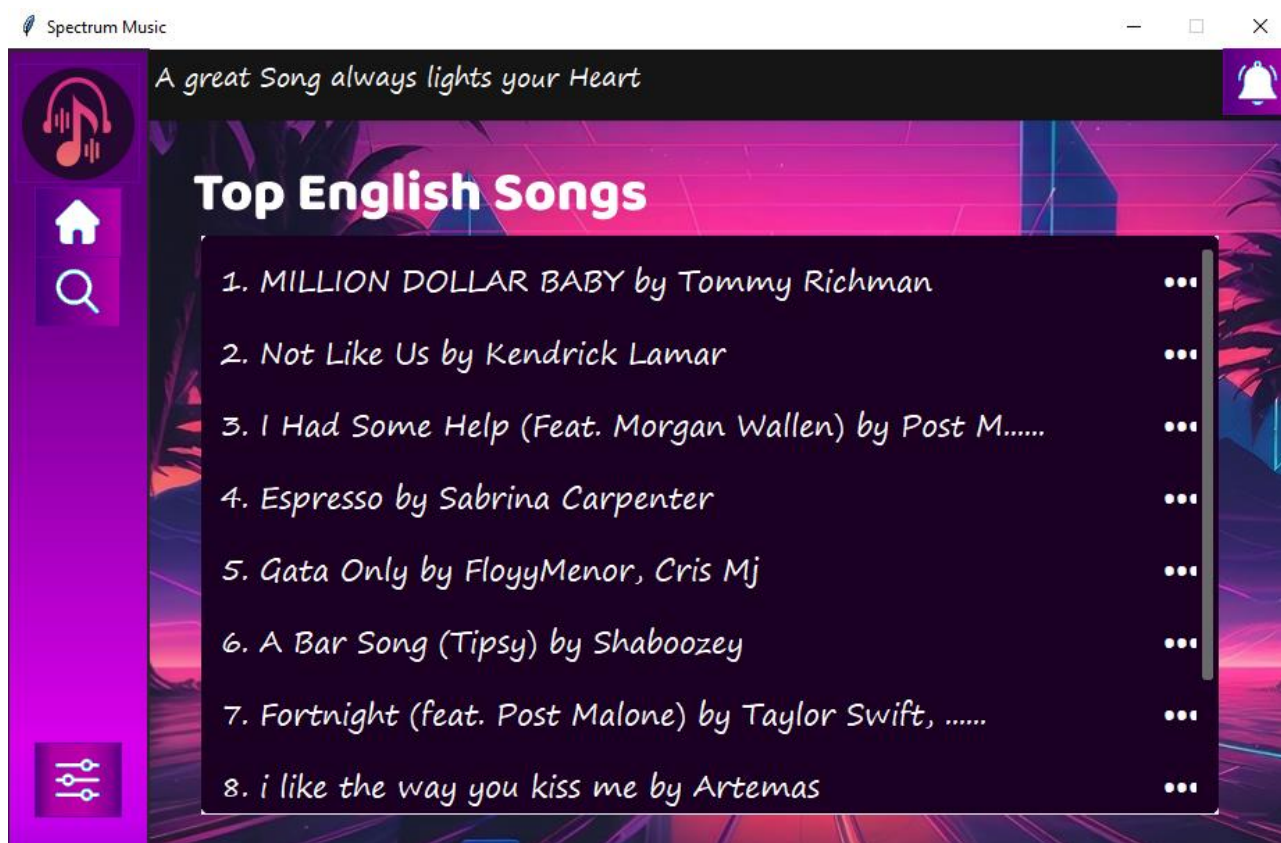
This is the Home page, where you can listen to songs suggested for you and get to know about the top trending English, Hindi, Tamil, and Malayalam songs. You can even listen to songs added to ur playlists.



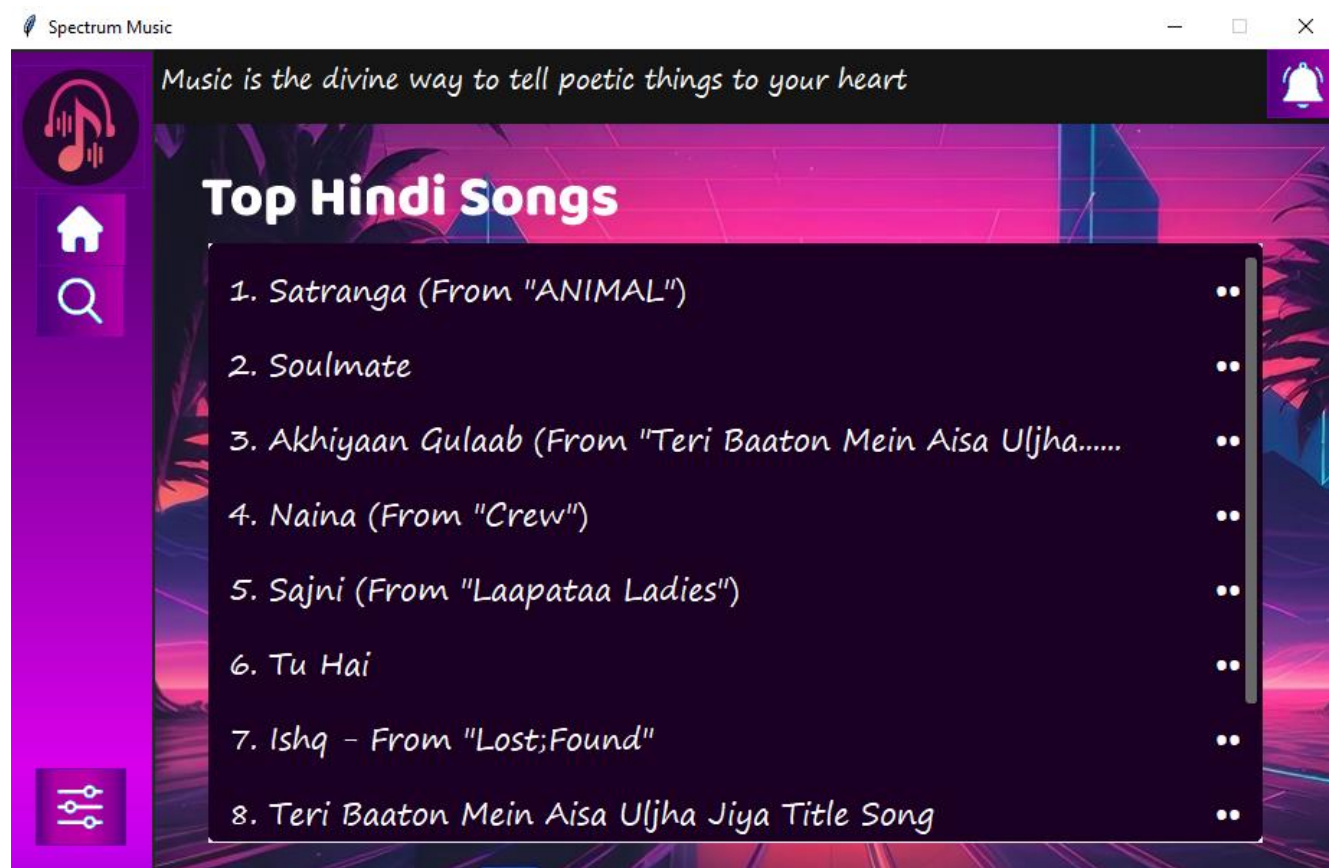
You can edit your playlist name and as well as the playlist profile image by clicking on the edit button



Top English Songs:



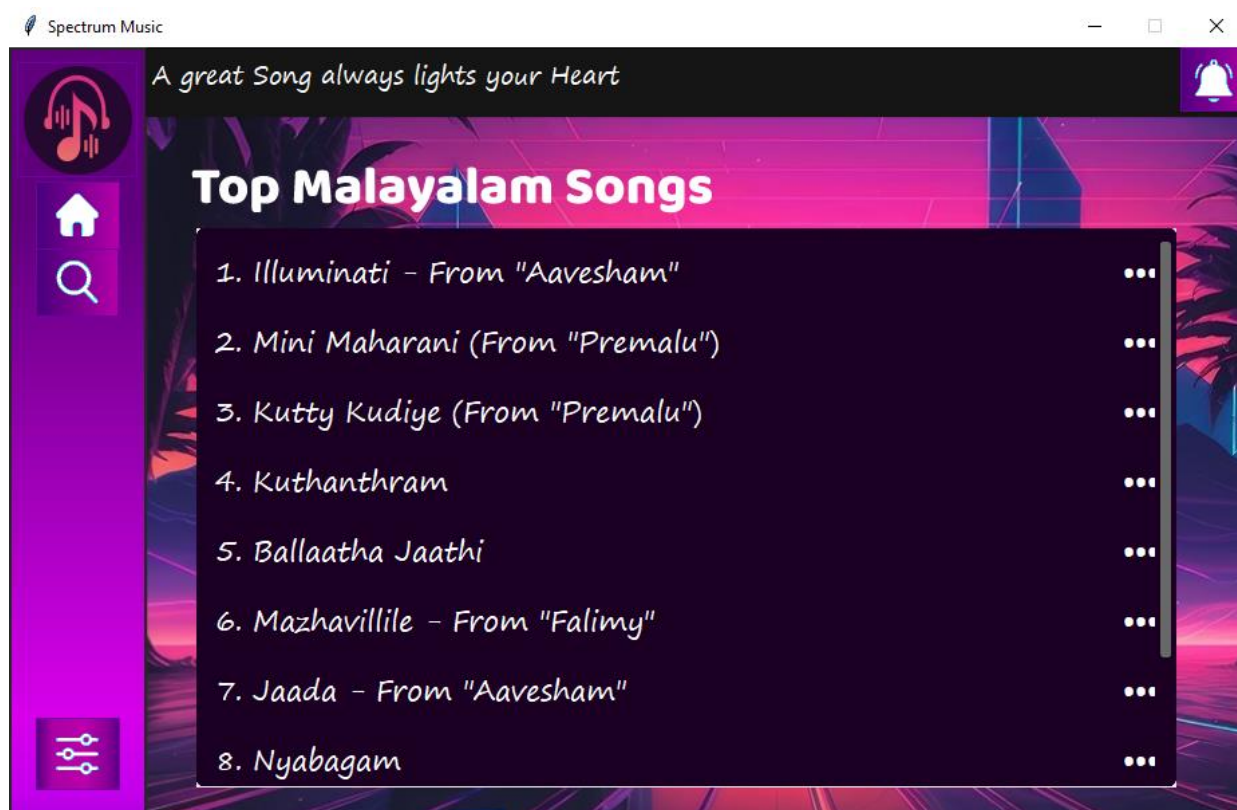
Top Hindi Songs:



Top Tamil Songs:



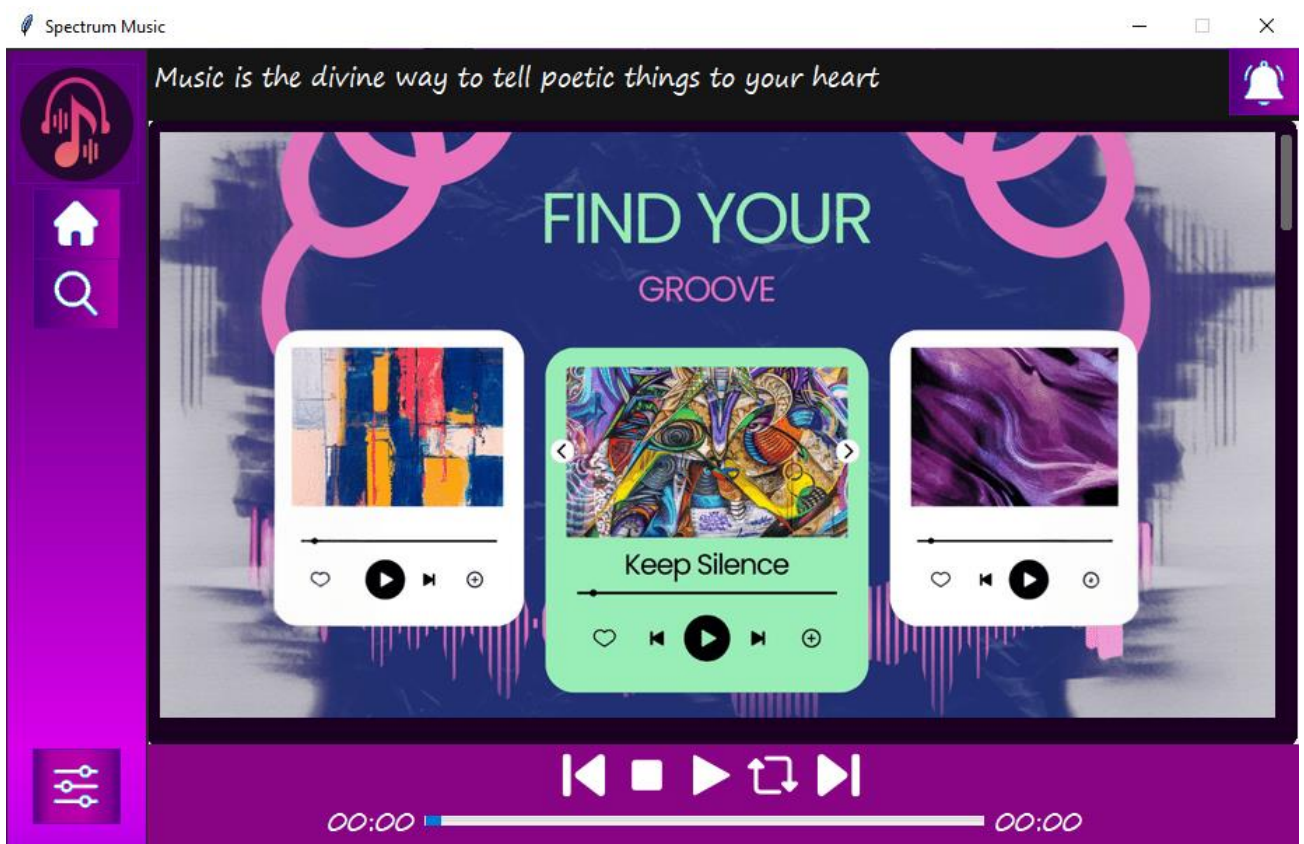
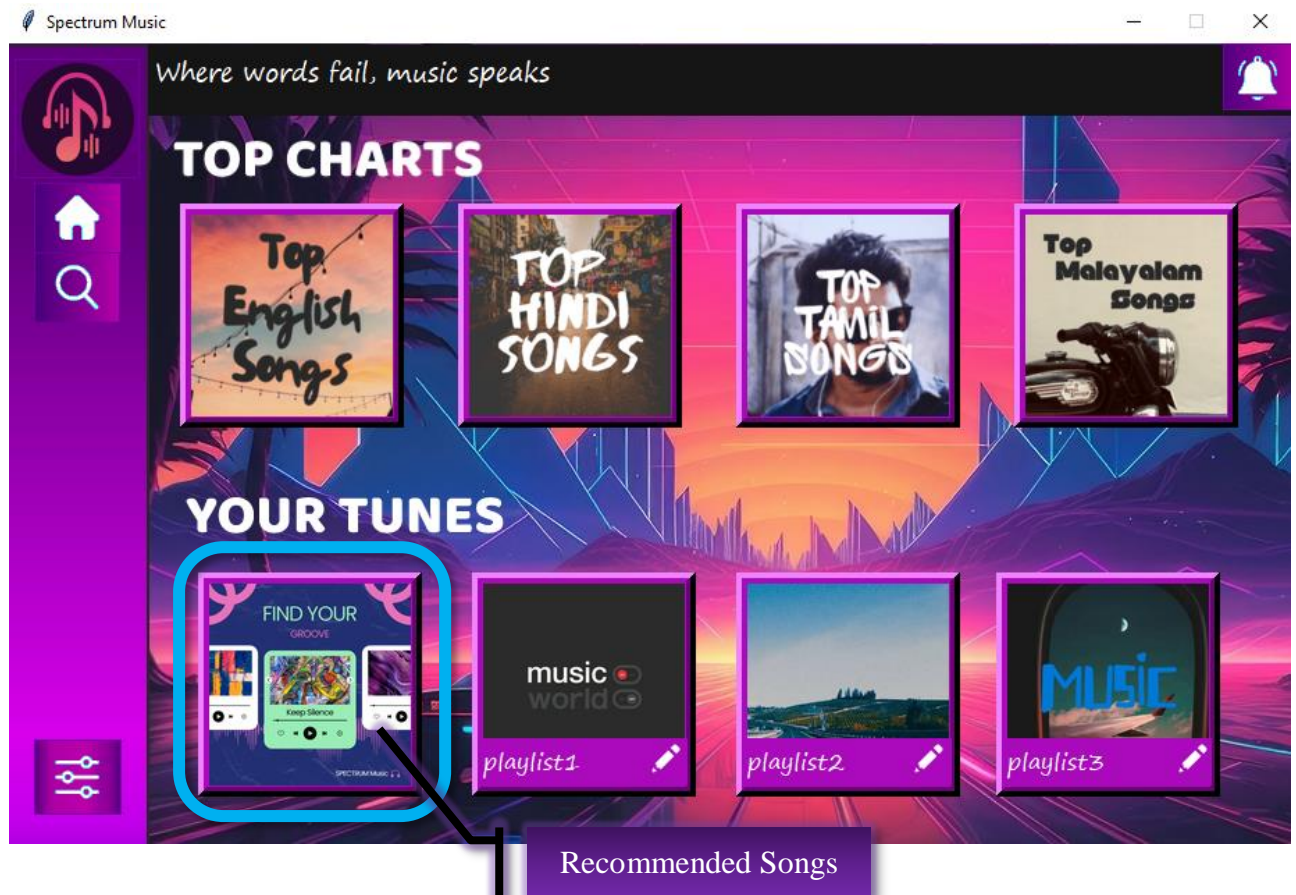
Top Malayalam Songs:

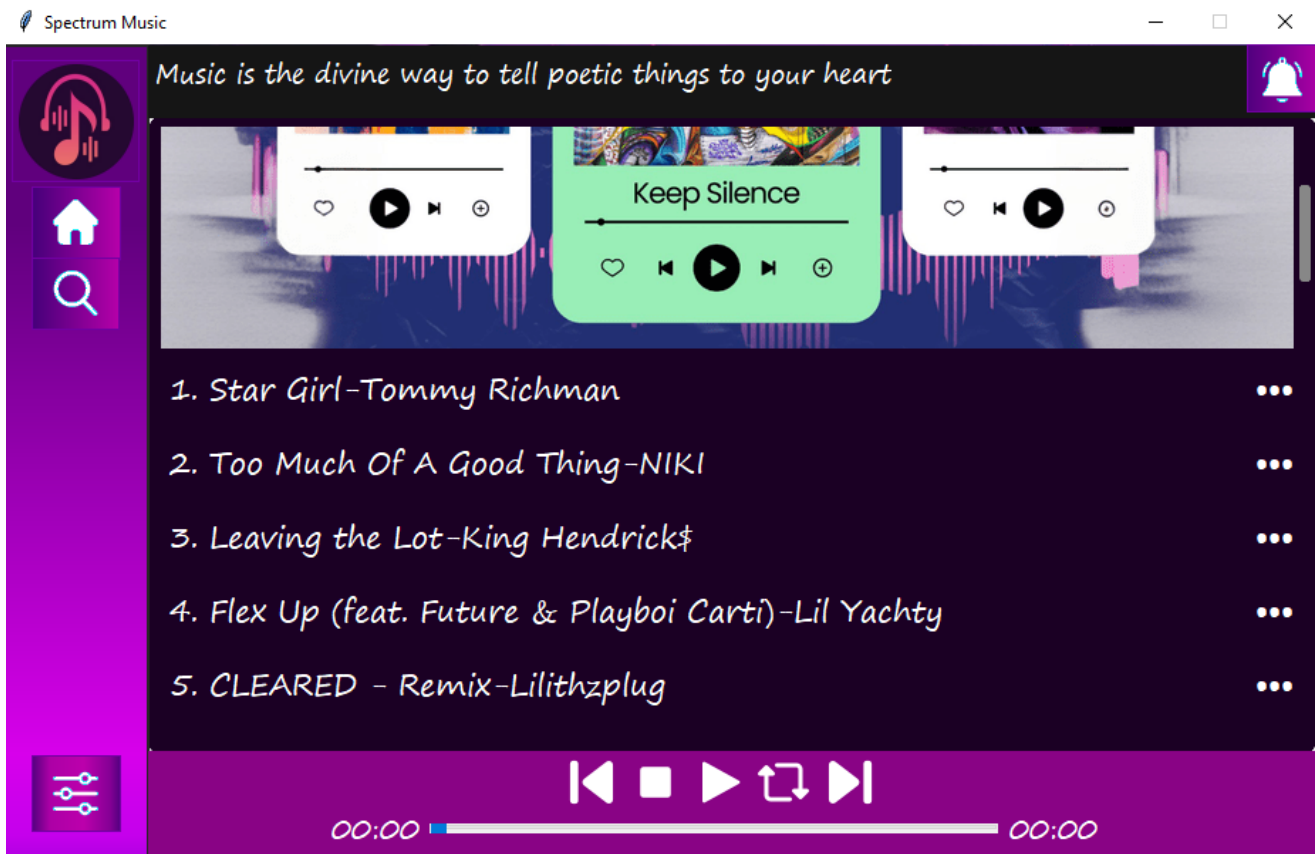


On clicking on any of the songs displayed in the top songs page, you can visit the song page and get to more about the song as well as the artist. You can also add the songs to your playlists or favourites folder by clicking on the menu button and choosing the respective option.

Song Recommendations:

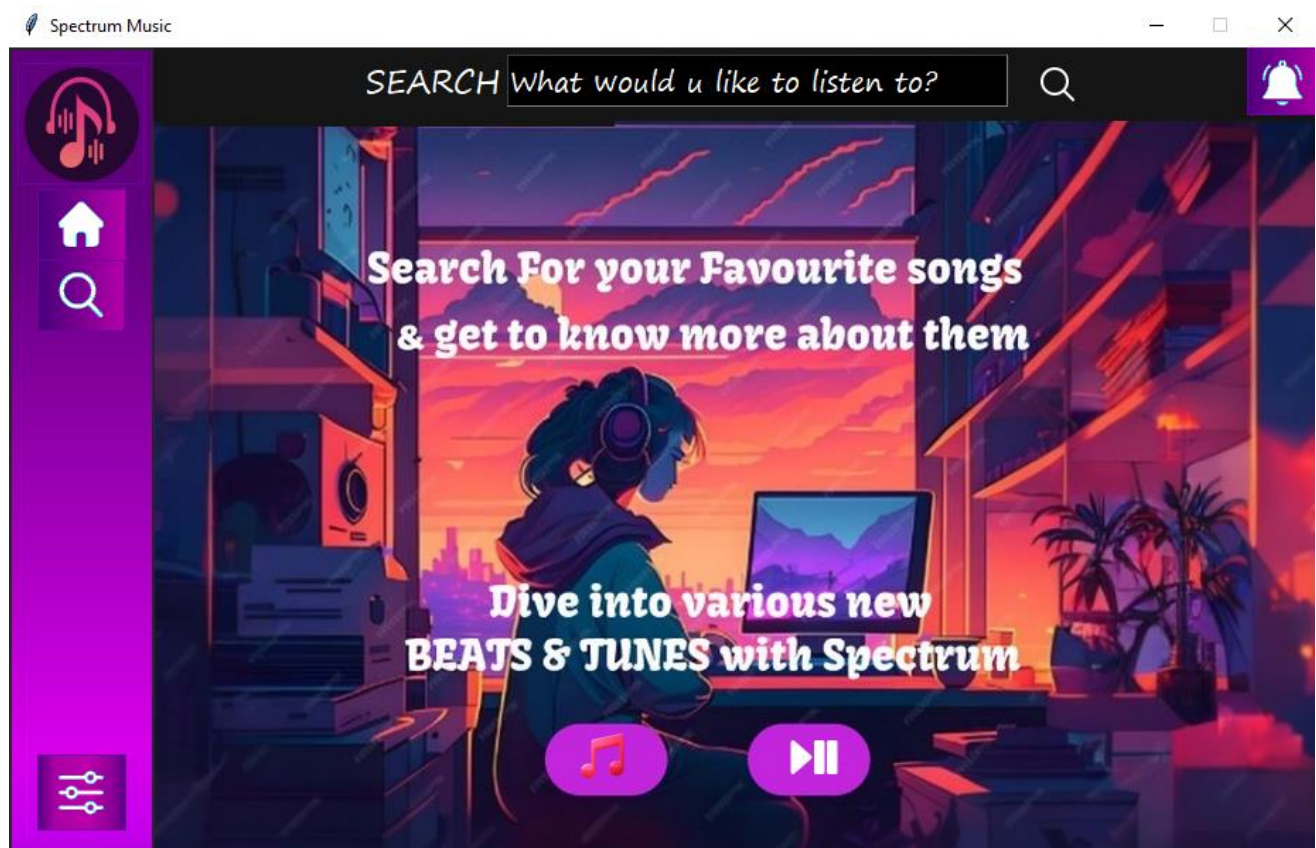
On clicking on the recommendation button, Spectrum suggests and introduces you to a wide range of songs based on your music taste and top trending songs.



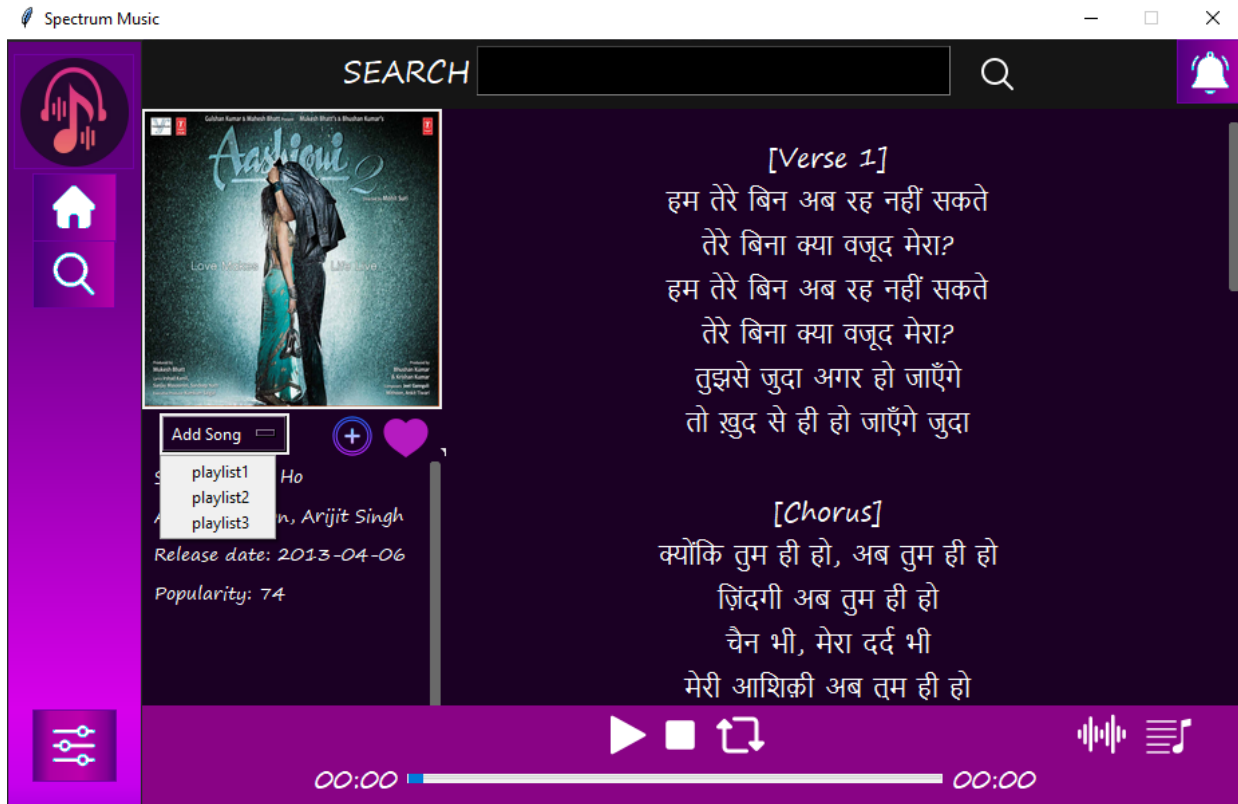


Search Page:

The Search window opens up using which you can search and listen to your favorite songs or dive into a new varieties of Tunes and Beats with spectrum.....

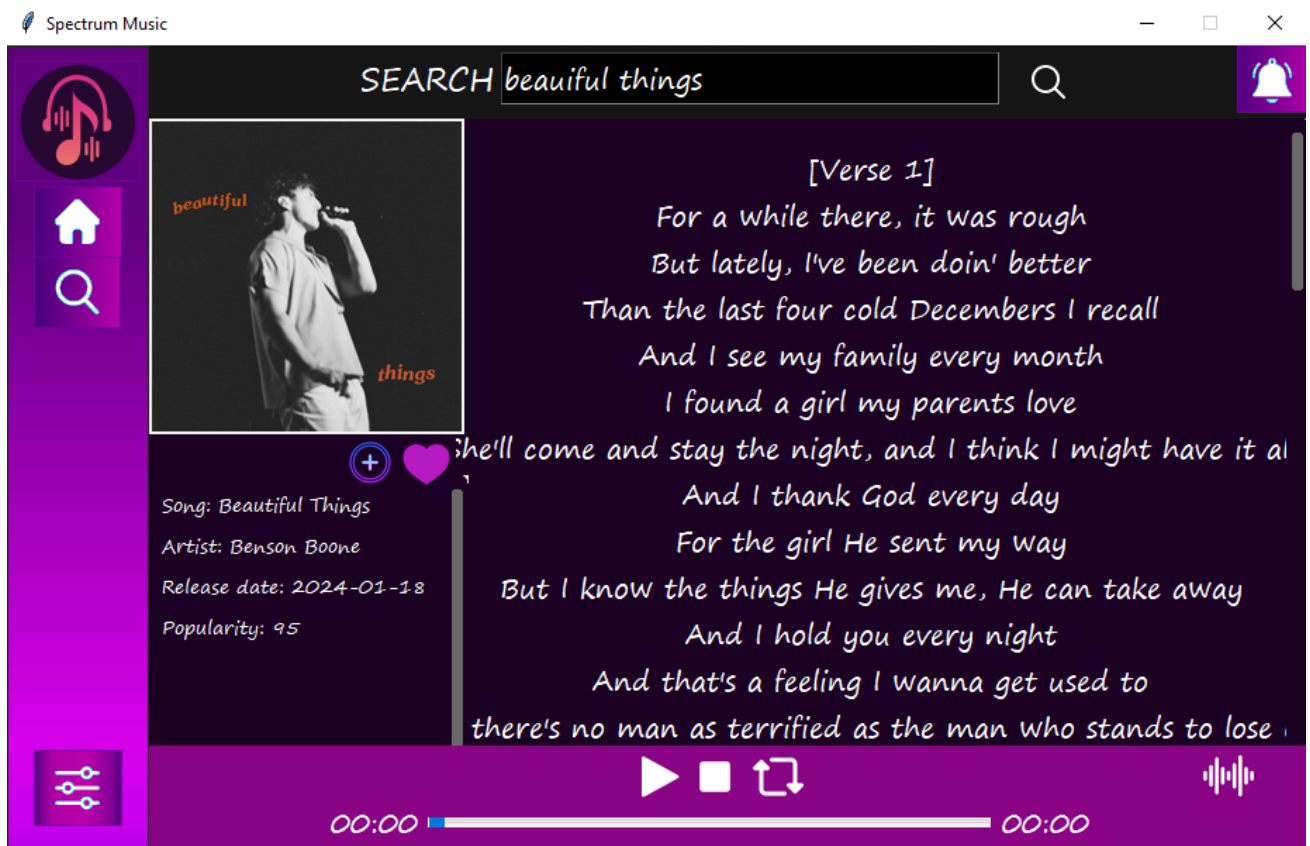


Searching For “Tum Hi Ho”:



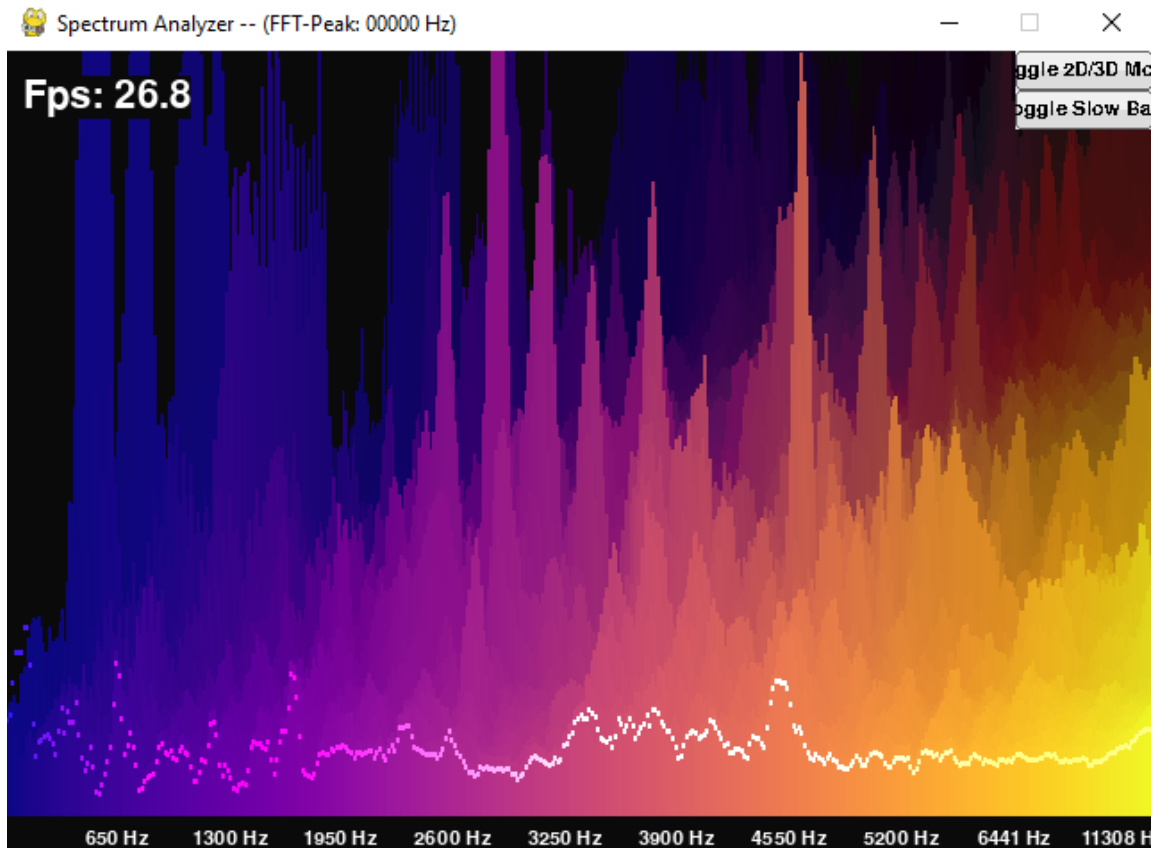
You can click on the heart button to add the searched song to your favourites album. On clicking the ADD button, you can add the searched song to your playlists.

Searching for “Beautiful Things”:

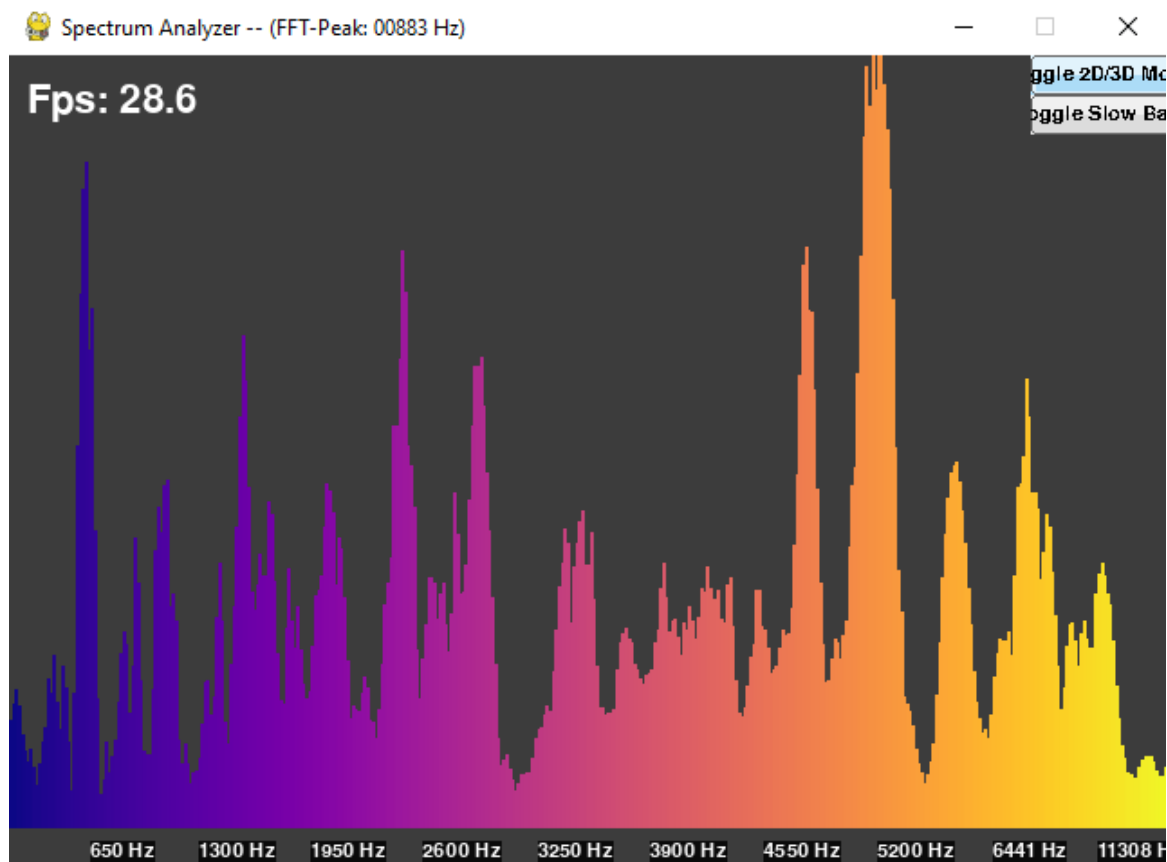


Music Visualizer:

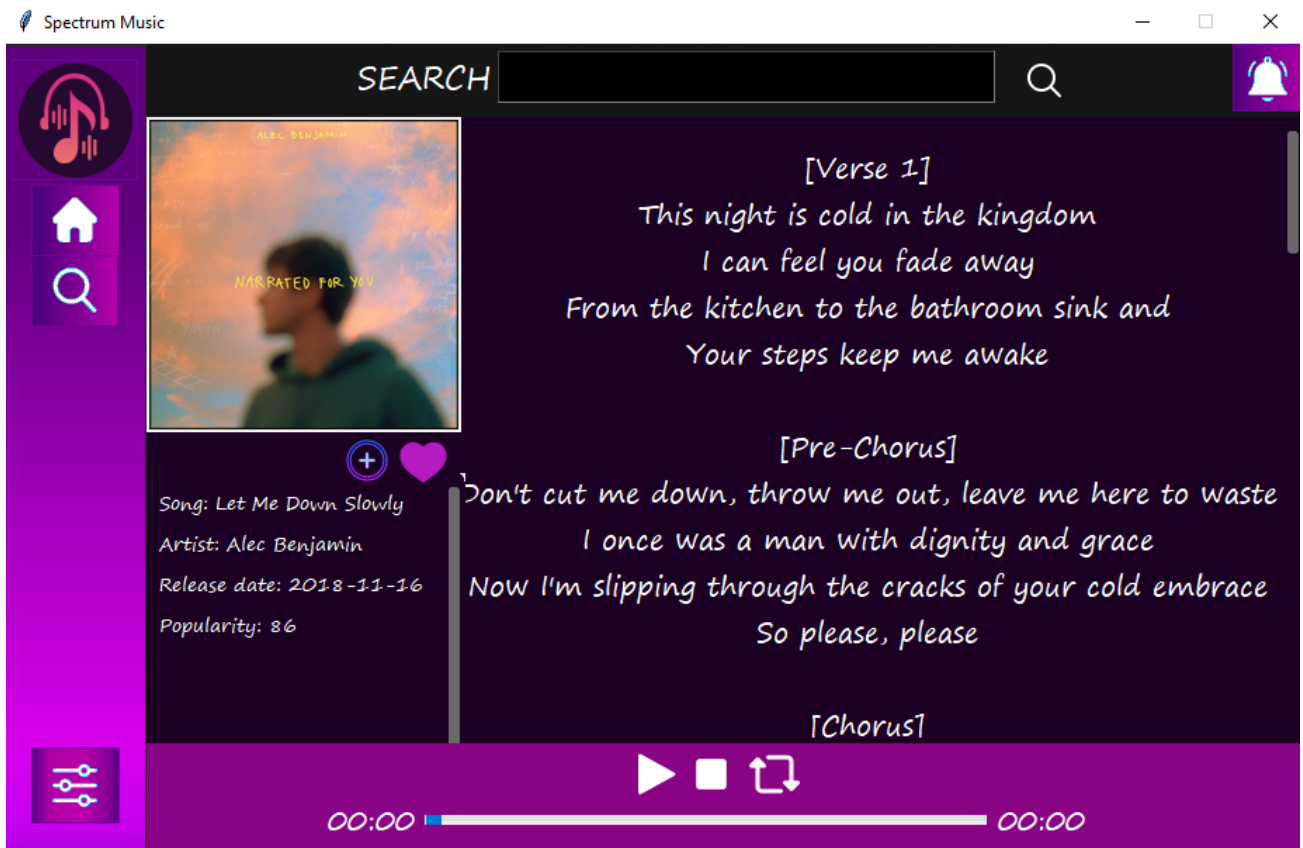
On clicking the Music visualizer button, it opens up a separate music visualizer window:



You can toggle between the 2D and 3D modes of the visualizer:



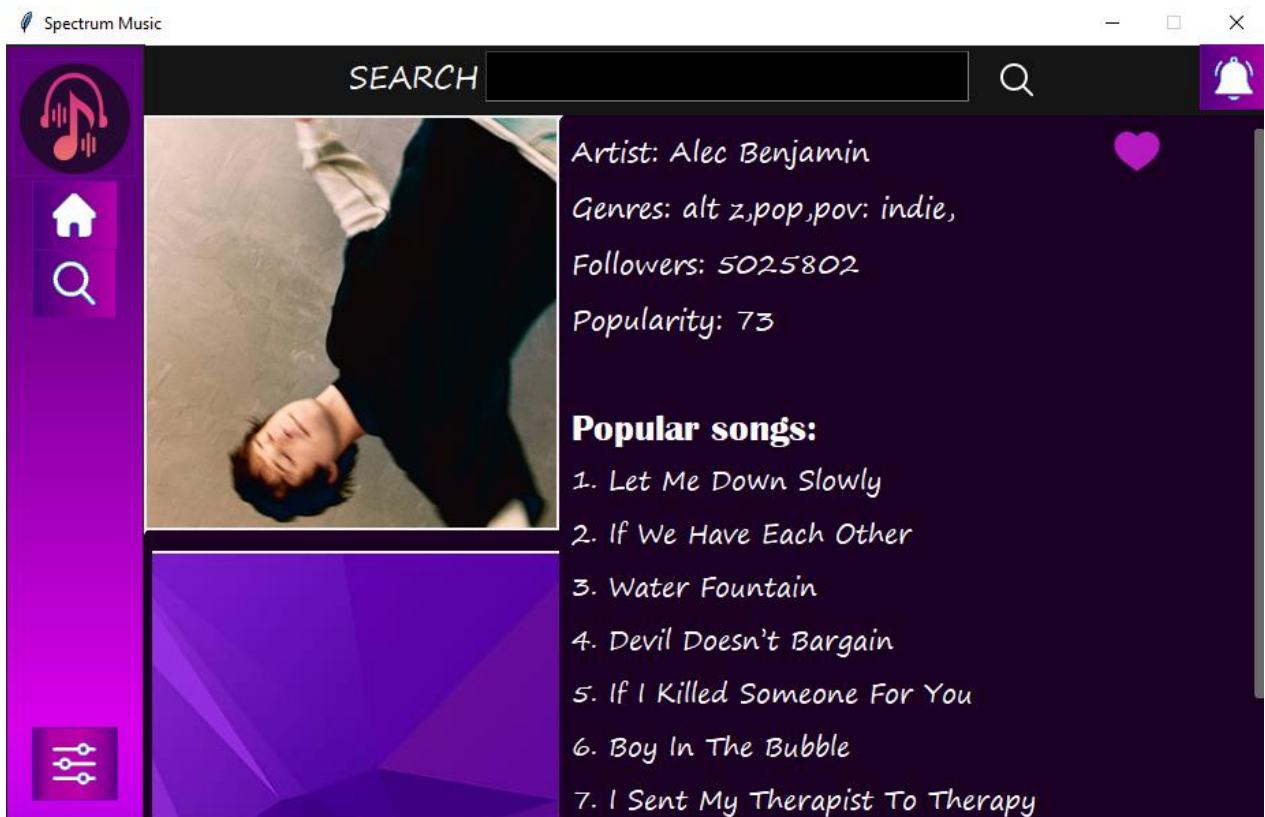
Searching for “Let Me Down Slowly”:



Artist Info: Alec Benjamin

You can get to know more about the artist by clicking on the artist name.


You can follow an artist by clicking on the heart button.



Searching for “Jaada”:

Spectrum Music

SEARCH jaada



Song: Jaada
Artist: Sushin Shyam, Sreenat
Release date: 2024-04-19
Popularity: 67


കുലീനരേ
ഉധാത്തരേ
ഉറ്റ തോഴരേ
ശുദ്ധ മർദ്ദ്യരേ
താഴെ വീണ കണ്ട് പല്ലിളിച്ച കുട്ടരേ
ഏറ്റ തോൽവി കണ്ട് നോക്കി നിന്ന മുഖരേ
പെട്ട മാനഹാനി ആസ്വദിച്ച നീചരേ
തീർന്നു പോകുമെന്ന് മുൻവിധിച്ച മുറരേ
ശക്തി ഉള്ളവന്റെ കൂട പിടിക്കും അൽപ്പരെ
കണ്ണുനീരിൻ ഉപ്പ് കറിയിലിട്ട സ്വാർത്ഥരെ
മങ്ങി മാഞ്ഞു ഭൂതകാലം
ഇന്നിവന്റെ ഉഴഴം
കൺ തുറന്ന് കൺ നിറച്ച് കാണുക

00:00 00:00

Artist Info: Sushin Shyam

Spectrum Music

A great Song always lights your Heart



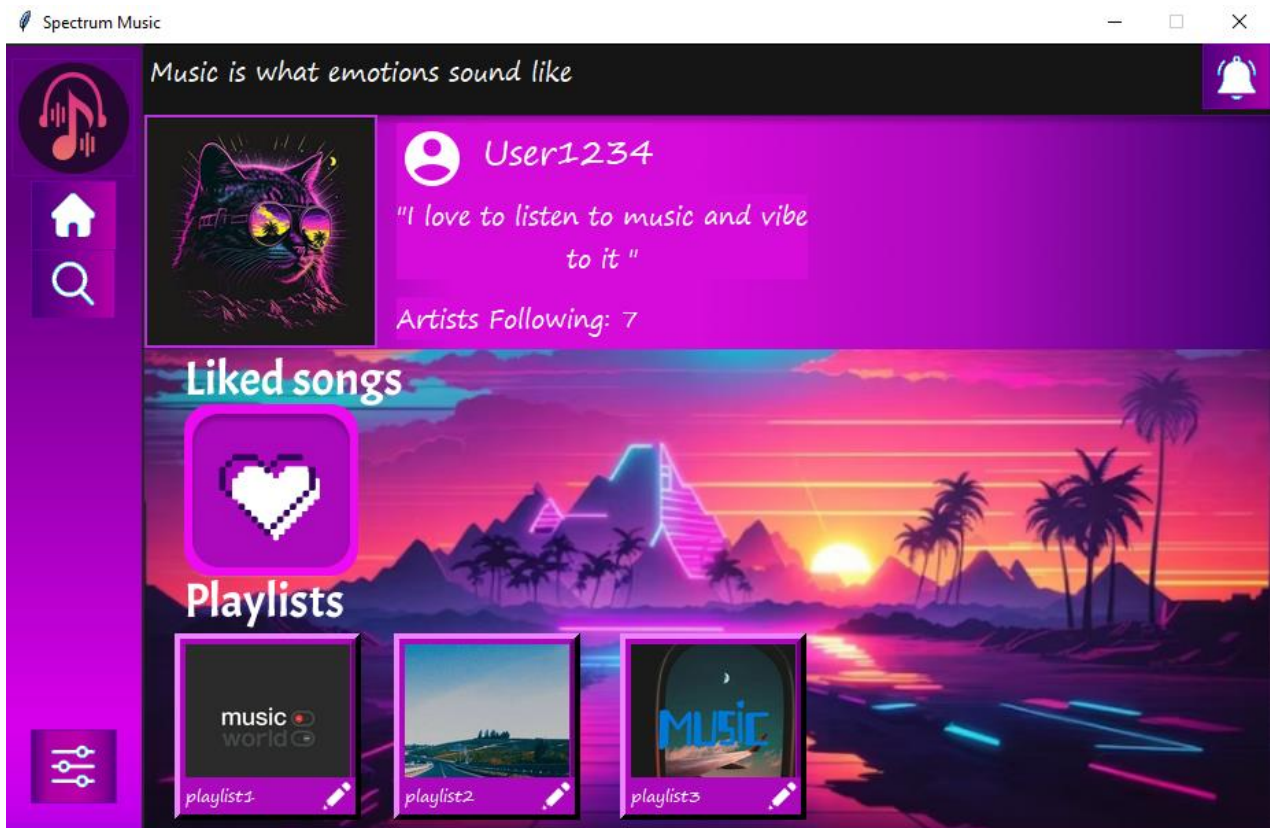
Artist: Sushin Shyam
Genres: mollywood,
Followers: 1161073
Popularity: 71

Popular songs:

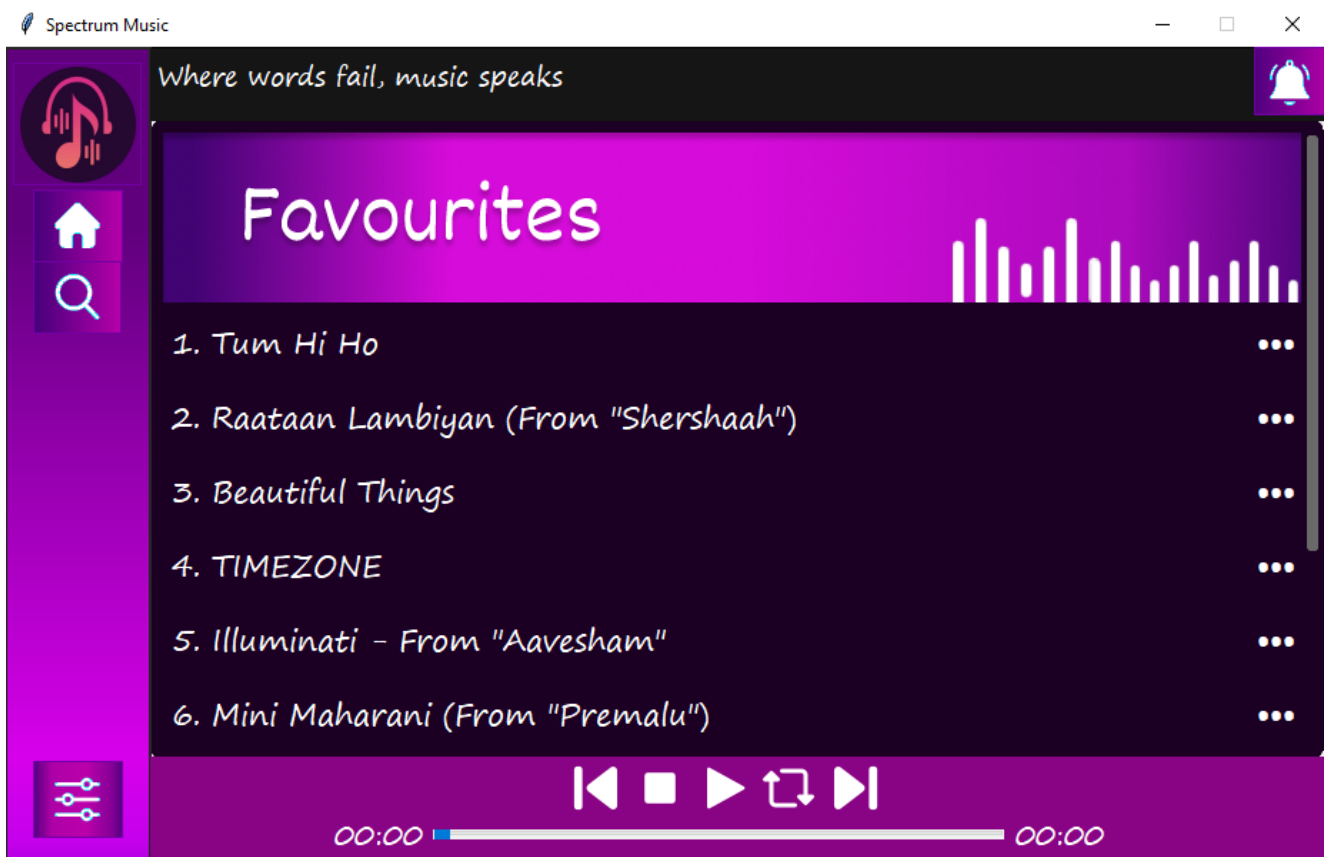
1. Illuminati
2. Jaada
3. Galatta
4. Armadham
5. Kuthanthram
6. Mathapithakkale
7. Odimag

Profile Page:

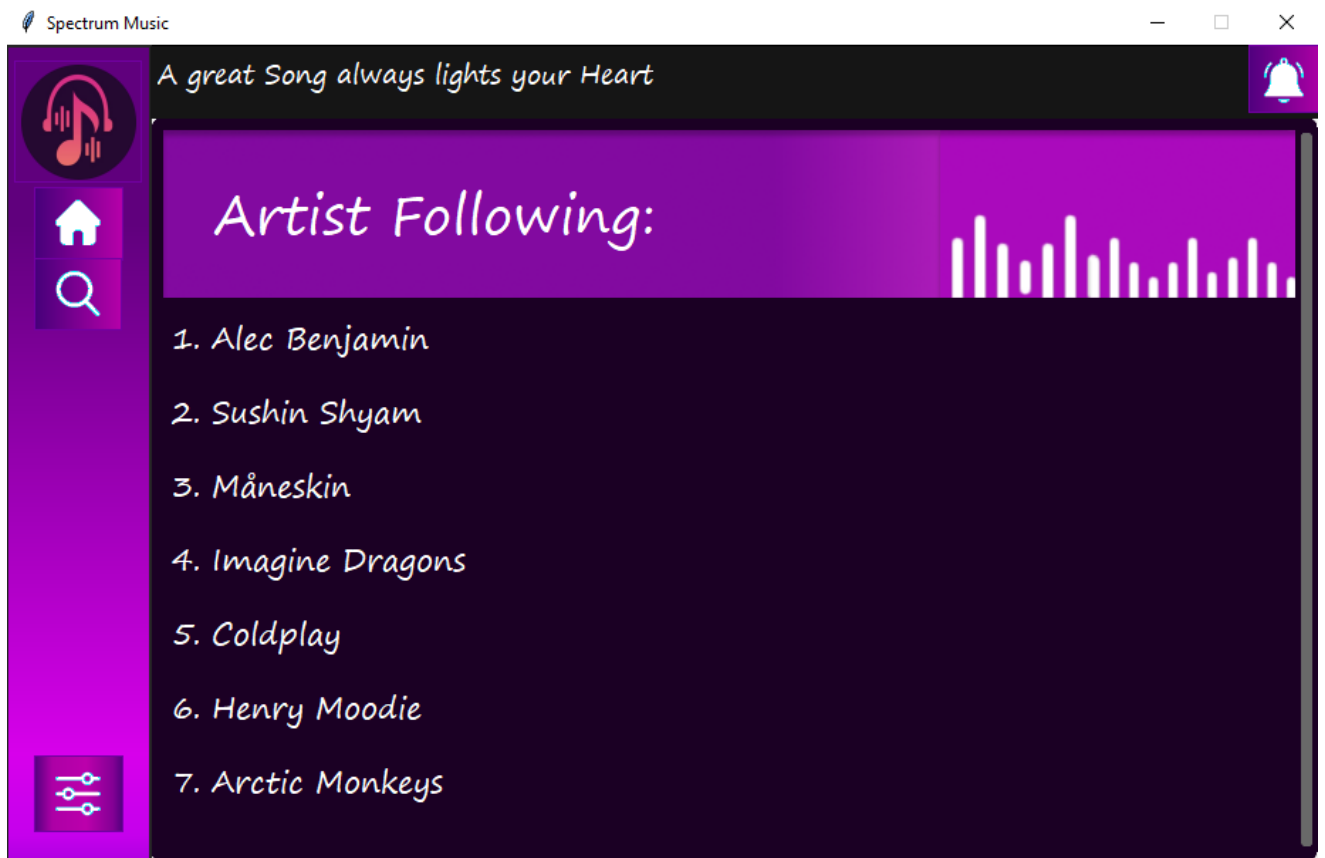
From the user profile page, you can view your playlists, rewind back to your Liked songs and catch up with the latest songs of artists you follow.



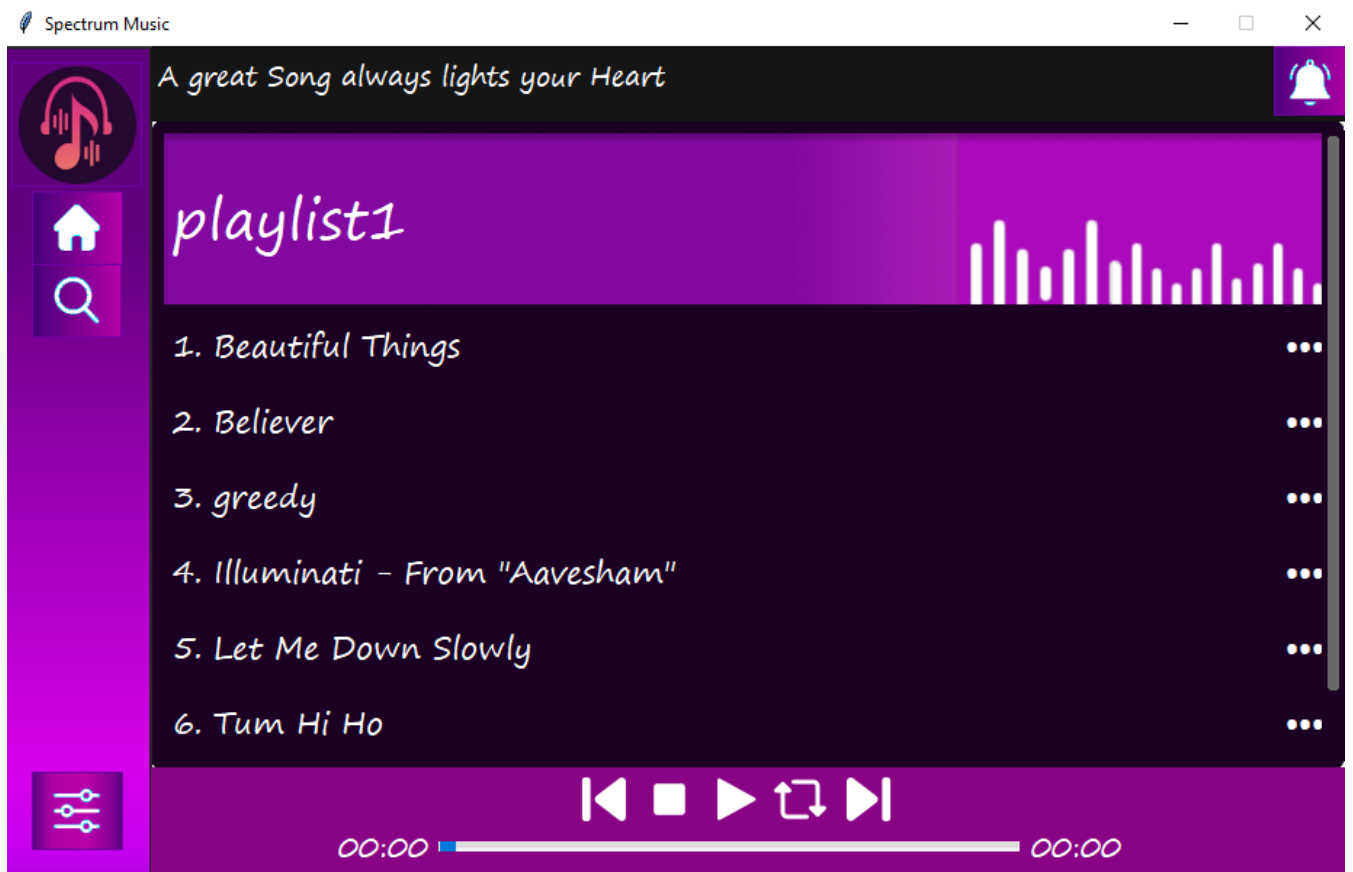
Liked songs:



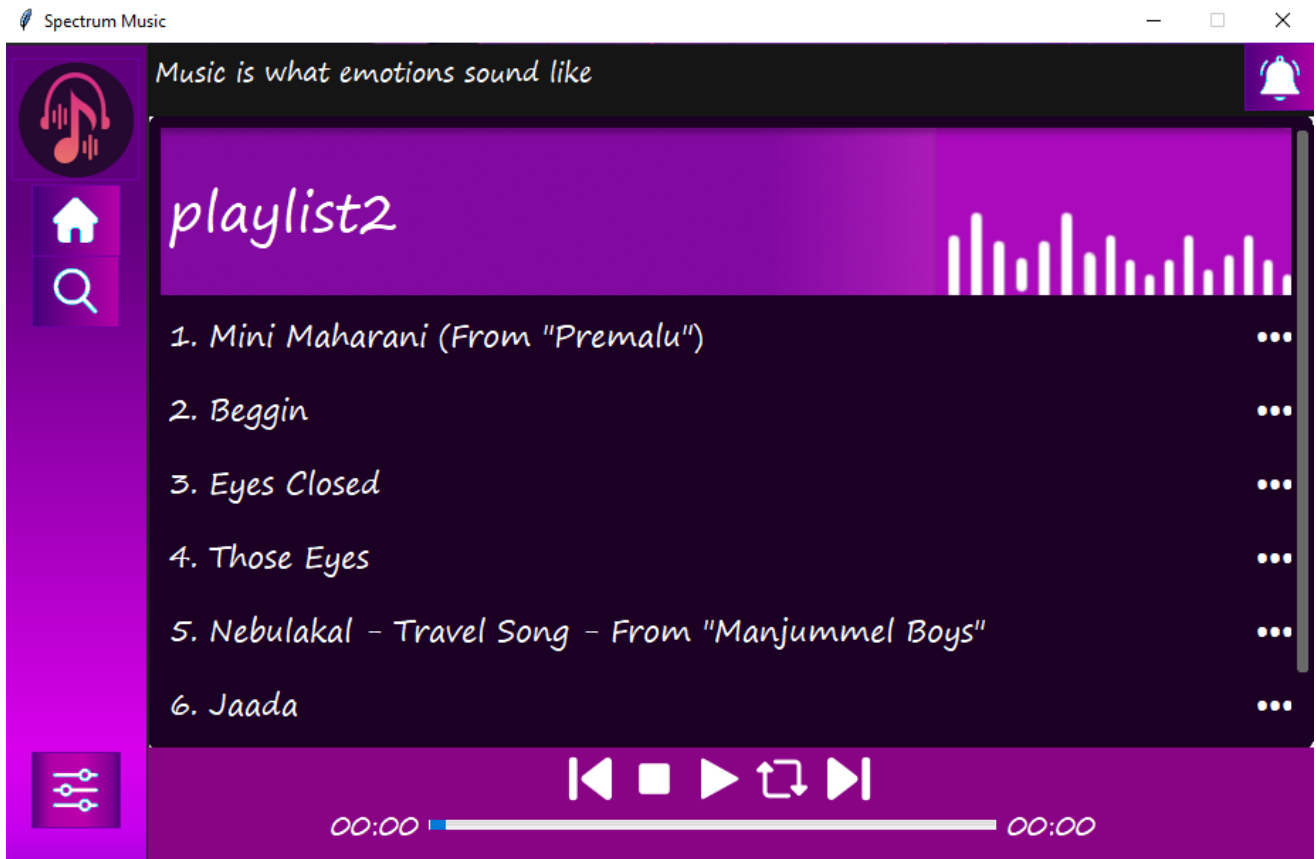
Following Artist page:



Playlist 1:

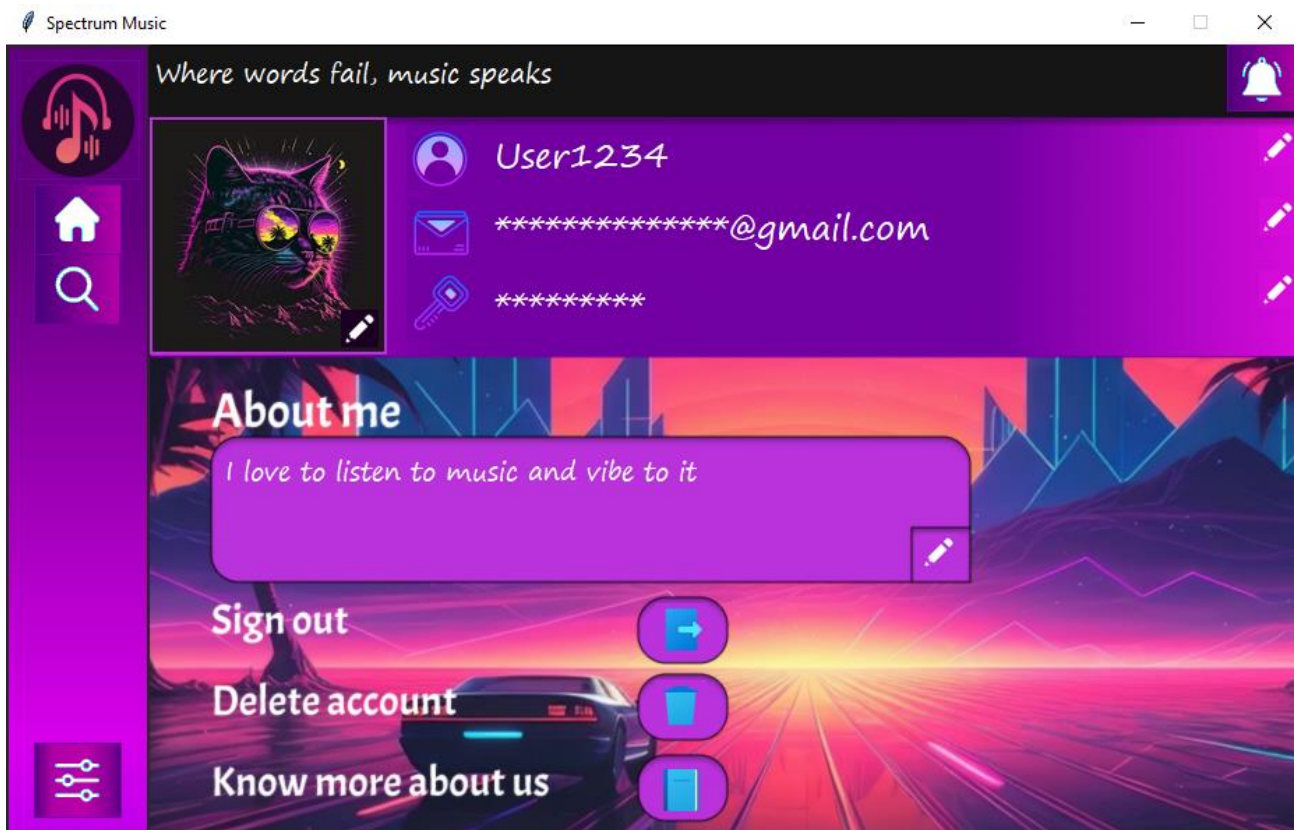


Playlist 2:

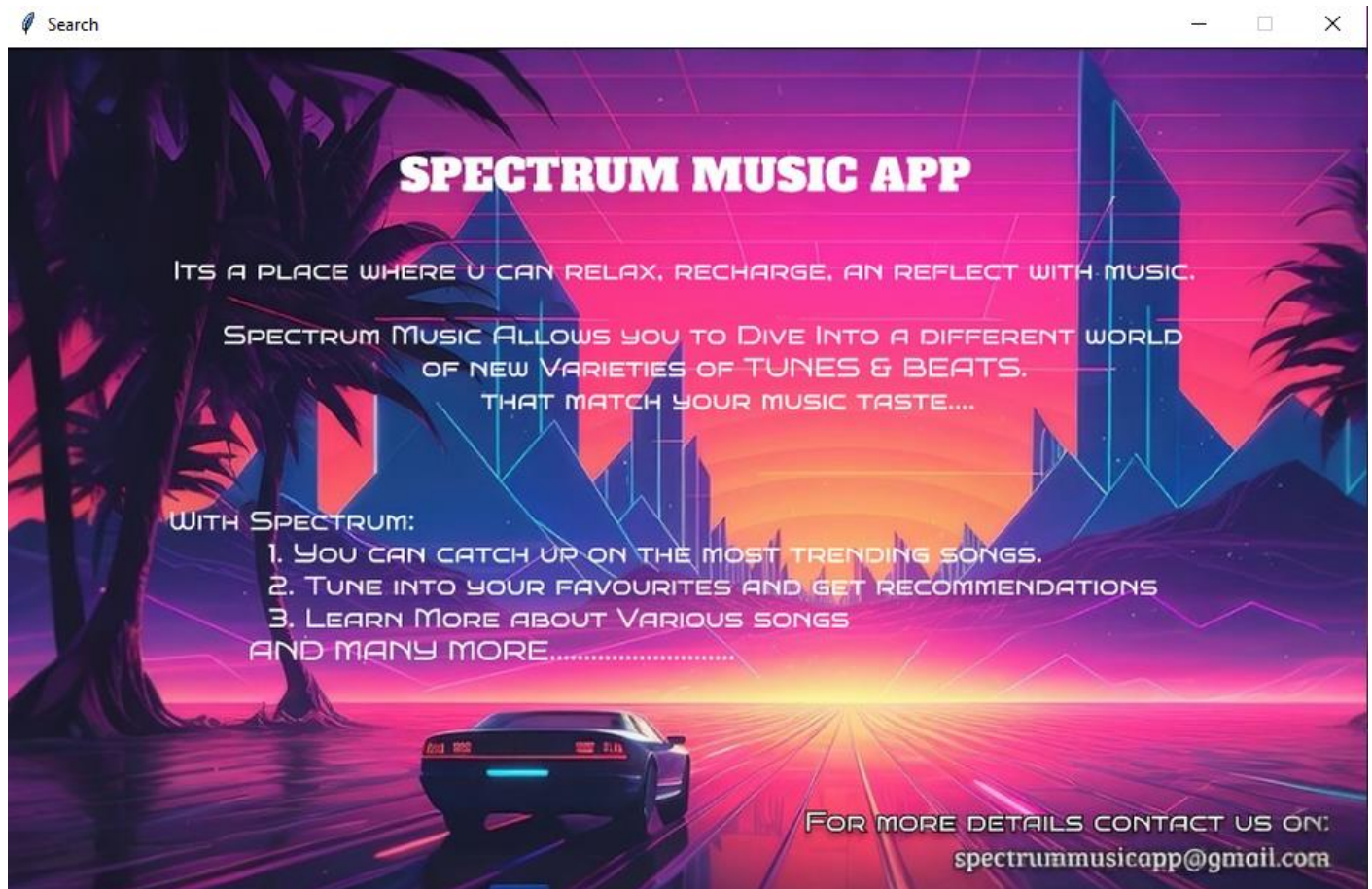


Settings Page:

You can edit your username, password, and email id as per your requirement and as well as add a Profile image.



You can click on the 'About us' button to get to know more about Spectrum and its features.

A promotional banner for the Spectrum Music App. The background is a vibrant, stylized illustration of a tropical beach at sunset or sunrise, with palm trees on the left and a futuristic, geometric cityscape on the right. A sleek, dark car is driving away from the viewer on a wet, reflective surface. The text is overlaid on this background in a clean, sans-serif font. At the top left, there is a search icon and the word 'Search'. At the top right, there are standard window control icons (minimize, maximize, close).

Search

SPECTRUM MUSIC APP

ITS A PLACE WHERE U CAN RELAX, RECHARGE, AN REFLECT WITH MUSIC.

SPECTRUM MUSIC ALLOWS YOU TO DIVE INTO A DIFFERENT WORLD
OF NEW VARIETIES OF TUNES & BEATS.
THAT MATCH YOUR MUSIC TASTE....

WITH SPECTRUM:

1. YOU CAN CATCH UP ON THE MOST TRENDING SONGS.
2. TUNE INTO YOUR FAVOURITES AND GET RECOMMENDATIONS
3. LEARN MORE ABOUT VARIOUS SONGS
AND MANY MORE.....

FOR MORE DETAILS CONTACT US ON:
spectrummusicapp@gmail.com

CONCLUSION

Spectrum strives to push the limits of music programming that can be achieved with Python, providing a music streaming service that goes beyond the conventional – a statistically-inclined, and more importantly, free service.

That being said, there is still scope for improvement. The first and foremost priority would be a cloud-based system to store and retrieve songs for efficiency and data optimisation. Enhancing the current recommendation algorithm to provide more accurate and curated suggestions, and improving visuals are other key targets that are expected to be achieved in the near future.

Overall, this project not only delivers beyond general expectations and in spite of the constraints of Python, but also possesses the potential for future enhancements and success.

BIBLIOGRAPHY

—→ Computer Science with Python (Textbook for Class XII) *by Sumita Arora*

—→ <https://github.com>

—→ <https://codemy.com>

—→ <https://www.geeksforgeeks.org>

—→ <https://stackoverflow.com>

—→ <https://www.w3schools.com>