Aim:

The aim of this Python project is to create a simple music player using a graphical user interface (GUI). The music player allows users to select songs from a list, play, pause, stop, and navigate between songs. Additionally, the project includes a feature to display song lyrics in a separate window with the help of database integration.

Theory:

GUI Interface:

The project utilizes the Tkinter library for creating a graphical user interface. The interface is divided into 2 windows(Main window, Lyrics window ) and 5 frames(left frame, right frame, down frame, lyrics frame, Text frame), each serving a specific purpose, such as displaying the list of songs, controlling playback, and showing song lyrics.

Music Playback:

The Pygame library is used for music playback functionality. The project includes buttons for playing, pausing, stopping, and navigating between songs. The selected song is loaded and played using Pygame's mixer module.

Song List:

The project reads the list of available songs from a specified directory using the os module. The song list is displayed in a Tkinter Listbox within the GUI.

Lyrics Integration:

A button labeled "Lyrics" triggers the display of song lyrics in a separate window. Lyrics are loaded from text files associated with each song and are displayed using Tkinter Text widgets.

Database Integration:

The project incorporates SQLite3 for storing information about played songs. The played\_songs table in the example.db database stores the song name and a serial number each time a song is played.

File Handling:

The code includes file handling operations to read song lyrics from text files. Different text files contain lyrics for different songs, and the appropriate file is selected based on the currently playing song.

Conclusion:

From this project, several key concepts were learned and applied, including GUI development using Tkinter, audio playback using Pygame, integration with SQLite3 for basic database operations, and file handling to read lyrics from text files. Additionally, the project demonstrates how to structure a simple music player with a clean interface and basic functionality. It provides a foundation for further enhancement and customization, such as adding more features, improving the user interface, or integrating additional functionalities.

Source Code:-

from tkinter import \*

from PIL import ImageTk,Image

import os

from pygame import mixer

co1 = "#ffffff" #overall window color

co2 = "#FFFF00" #additional color

co3 = "#800080" #used for listbox mostly

co4 = "#333333" #used around buttons

window = Tk()

window.title("Music player using GUI")

window.geometry('352x325')

window.configure(background=co1)

window.resizable(width=FALSE,height=FALSE)

left\_frame = Frame(window, width=150,height=150, bg =co1)

left\_frame.grid(row=0, column=0, padx=1, pady=1)

right\_frame = Frame(window, width=250,height=150, bg =co3)

right\_frame.grid(row=0, column=1, padx=0)

down\_frame = Frame(window, width=400,height=100, bg =co4)

down\_frame.grid(row=1, column=0,columnspan = 3, padx=0,pady=1)

lyrics\_frame = Frame(window, width=400,height=100, bg =co4)

lyrics\_frame.grid(row=2, column=0,columnspan = 3, padx=0,pady=1)

#right frame

listbox = Listbox(right\_frame, selectmode =SINGLE,font=("Arial 10 bold"), width = 22, bg =co3 , fg = co1)

listbox.grid(row = 0, column = 0)

w = Scrollbar(right\_frame)

w.grid(row = 0, column = 1)

listbox.config(yscrollcommand = w.set)

w.config(command = listbox.yview)

os.chdir(r'C:\\Users\\chaud\\OneDrive\\Desktop\\10 pointer\\sem 3 DJ\\python mini\\music player\\mp3')

songs = os.listdir()

def show():

for i in songs:

listbox.insert(END,i)

show()

p\_song = ""

line = Label(left\_frame, width=200, height=1,padx=10,bg=co3)

line.place(x=0, y=1)

line = Label(left\_frame, width=200, height=1,padx=10,bg=co1)

line.place(x=0, y=3)

running\_song = Label(down\_frame,text ='Chose a song',font=("Ivy 10"), width=200, height=1, padx=10, bg=co1)

running\_song.place(x=0,y=1)

#events

def play\_music():

selected\_index = listbox.curselection()

if selected\_index: # Check if anything is selected

selected\_song = listbox.get(selected\_index[0]) # Get the selected song

running\_song['text'] = selected\_song

mixer.music.load(selected\_song)

mixer.music.play()

def pause\_music():

mixer.music.pause()

def continue\_music():

mixer.music.unpause()

def stop\_music():

mixer.music.stop()

def next\_music():

playing = running\_song['text']

index = songs.index(playing)

new\_index = index+1

playing = songs[new\_index]

mixer.music.load(playing)

mixer.music.play()

listbox.delete(0,END)

show()

listbox.select\_set(new\_index)

running\_song['text'] = playing

def previous\_music():

playing = running\_song['text']

if playing in songs:

index = songs.index(playing)

new\_index = index - 1 if index > 0 else len(songs) - 1

playing = songs[new\_index]

p\_song = playing

mixer.music.load(playing)

mixer.music.play()

listbox.delete(0, END)

show()

listbox.select\_set(new\_index)

running\_song['text'] = playing

else:

print("Not in the list")

def reset\_directory():

    os.chdir(r'C:\\Users\\chaud\\OneDrive\\Desktop\\10 pointer\\sem 3 DJ\\python mini\\music player\\mp3')

def create\_new\_window():

import sqlite3 as s

import os

os.chdir(r'C:\Users\\chaud\\OneDrive\\Desktop\\10 pointer\\sem 3 DJ\\python mini\\music player\\music files')

# Connect to a database (creates it if it doesn't exist)

conn = s.connect('example23.db')

# Create a cursor object

cur = conn.cursor()

# Execute SQL queries

# For example, create a table

cur.execute('''CREATE TABLE IF NOT EXISTS lyrics (song\_name TEXT PRIMARY KEY, lyrics TEXT)''')

# Insert some data

playing\_song = p\_song+running\_song["text"]

if playing\_song == 'aruni kirani.mp3':

file1 = open('C:\\Users\\chaud\\OneDrive\\Desktop\\10 pointer\\sem 3 DJ\\python mini\\music player\\music files\\lyrics\\l1.txt', 'r')

lyric\_lines = file1.read()

elif playing\_song == 'Bass rani Nucleya.mp3':

file2 = open('C:\\Users\\chaud\\OneDrive\\Desktop\\10 pointer\\sem 3 DJ\\python mini\\music player\\music files\\lyrics\\l2.txt', 'r')

lyric\_lines = file2.read()

elif playing\_song == 'He\'s a pirate Cello Version.mp3':

file3 = open('C:\\Users\\chaud\\OneDrive\\Desktop\\10 pointer\\sem 3 DJ\\python mini\\music player\\music files\\lyrics\\l3.txt', 'r')

lyric\_lines = file3.read()

elif playing\_song == 'hip hop snoop.mp3':

file4 = open('C:\\Users\\chaud\\OneDrive\\Desktop\\10 pointer\\sem 3 DJ\\python mini\\music player\\music files\\lyrics\\l4.txt', 'r')

lyric\_lines = file4.read()

elif playing\_song == 'lil nas X.mp3':

file5 = open('C:\\Users\\chaud\\OneDrive\\Desktop\\10 pointer\\sem 3 DJ\\python mini\\music player\\music files\\lyrics\\l5.txt', 'r')

lyric\_lines = file5.read()

elif playing\_song == 'Poker face.mp3':

file6 = open('C:\\Users\\chaud\\OneDrive\\Desktop\\10 pointer\\sem 3 DJ\\python mini\\music player\\music files\\lyrics\\l6.txt', 'r')

lyric\_lines = file6.read()

elif playing\_song == 'rain on me drums.mp3':

file7 = open('C:\\Users\\chaud\\OneDrive\\Desktop\\10 pointer\\sem 3 DJ\\python mini\\music player\\music files\\lyrics\\l7.txt', 'r')

lyric\_lines = file7.read()

elif playing\_song == 'Shree Ram Janki(PaglaSongs).mp3':

file8 = open('C:\\Users\\chaud\\OneDrive\\Desktop\\10 pointer\\sem 3 DJ\\python mini\\music player\\music files\\lyrics\\l8.txt', 'r')

lyric\_lines = file8.read()

elif playing\_song == 'Zindagi-Zindagi-Sachin-Pilgaonkar-Mahesh-Manjrekar-Sumeet-Raghavan-Sunil-Barve-Prasad-Oak-Kedar-Shinde.mp3':

file9 = open('C:\\Users\\chaud\\OneDrive\\Desktop\\10 pointer\\sem 3 DJ\\python mini\\music player\\music files\\lyrics\\l9.txt', 'r')

lyric\_lines = file9.read()

cur.execute("delete from lyrics")

cur.execute(''' Insert or ignore into lyrics (song\_name,lyrics) values (?,?) ''',(playing\_song,lyric\_lines))

# Commit changes

conn.commit()

# Fetch data

cur.execute("SELECT \* FROM lyrics")

rows = cur.fetchall()

for row in rows:

print(row)

# Close the connection

conn.close()

new\_window = Toplevel(window)

new\_window.title("~Lyrics~")

label = Label(new\_window, text="~Lyrics~")

new\_window.geometry('300x250')

new\_window.resizable(height=True , width = True)

Text\_frame = Frame(new\_window, width=250, height=220, bg='#ffffff')

Text\_frame.grid(row=0, column=2, columnspan=10, padx=10, pady=10)

output\_text = lyric\_lines

output\_area = Text(Text\_frame, wrap="word", bg="white", fg="black", font=("Arial", 10))

output\_area.pack(expand=True, fill="both", padx=5, pady=5)

output\_text = lyric\_lines+""

# Clear any previous content in the text widget

output\_area.delete('1.0', END)

#switching of windows optimisation

output\_area.insert(END, output\_text)

def close\_lyrics\_window():

reset\_directory() # Reset directory path to 'mp3' when the lyrics window is closed

new\_window.destroy() # Close the lyrics window

new\_window.protocol("WM\_DELETE\_WINDOW", close\_lyrics\_window)

# image

img\_1 = Image.open('C:\\Users\\chaud\\OneDrive\\Desktop\\10 pointer\\sem 3 DJ\\python mini\\music player\\music-435.png')

img\_1 = img\_1.resize((130,130))

img\_1 = ImageTk.PhotoImage(img\_1)

app\_image = Label(left\_frame, height=130,image=img\_1,padx=10 , bg= co1)

app\_image.place(x=10, y=15)

img\_2 = Image.open('C:\\Users\\chaud\\OneDrive\\Desktop\\10 pointer\\sem 3 DJ\\python mini\\music player\\prevbutton.png')

img\_2 = img\_2.resize((30,30))

img\_2 = ImageTk.PhotoImage(img\_2)

prev\_button = Button(down\_frame, width = 40 ,height=40,image=img\_2,padx=10 , bg= co1 , font = ("Ivy 10"),command =previous\_music)

prev\_button.place(x=84, y=35)

img\_3 = Image.open('C:\\Users\\chaud\\OneDrive\\Desktop\\10 pointer\\sem 3 DJ\\python mini\\music player\\play-button\_482059.png')

img\_3 = img\_3.resize((30,30))

img\_3 = ImageTk.PhotoImage(img\_3)

play\_button = Button(down\_frame, width = 40 ,height=40,image=img\_3,padx=10 , bg= co1 , font = ("Ivy 10"),command = play\_music)

play\_button.place(x=38, y=35)

img\_4 = Image.open('C:\\Users\\chaud\\OneDrive\\Desktop\\10 pointer\\sem 3 DJ\\python mini\\music player\\right-arrow.png')

img\_4 = img\_4.resize((30,30))

img\_4 = ImageTk.PhotoImage(img\_4)

next\_button = Button(down\_frame, width = 40 ,height=40,image=img\_4,padx=10 , bg= co1 , font = ("Ivy 10"),command = next\_music)

next\_button.place(x=130, y=35)

img\_5 = Image.open('C:\\Users\\chaud\\OneDrive\\Desktop\\10 pointer\\sem 3 DJ\\python mini\\music player\\pause\_254433.png')

img\_5 = img\_5.resize((30,30))

img\_5 = ImageTk.PhotoImage(img\_5)

pause\_button = Button(down\_frame, width = 40 ,height=40,image=img\_5,padx=10 , bg= co1 , font = ("Ivy 10"),command = pause\_music)

pause\_button.place(x=176, y=35)

img\_6 = Image.open('C:\\Users\\chaud\\OneDrive\\Desktop\\10 pointer\\sem 3 DJ\\python mini\\music player\\fastfo.png')

img\_6 = img\_6.resize((30,30))

img\_6 = ImageTk.PhotoImage(img\_6)

continue\_button = Button(down\_frame, width = 40 ,height=40,image=img\_6,padx=10 , bg= co1 , font = ("Ivy 10"),command = continue\_music)

continue\_button.place(x=222, y=35)

img\_7 = Image.open('C:\\Users\\chaud\\OneDrive\\Desktop\\10 pointer\\sem 3 DJ\\python mini\\music player\\stop.png')

img\_7 = img\_7.resize((30,30))

img\_7 = ImageTk.PhotoImage(img\_7)

stop\_button = Button(down\_frame, width = 40 ,height=40,image=img\_7,padx=10 , bg= co1 , font = ("Ivy 10"),command = stop\_music)

stop\_button.place(x=268, y=35)

img\_8 = Image.open('C:\\Users\\chaud\\OneDrive\\Desktop\\10 pointer\\sem 3 DJ\\python mini\\music player\\audio-playlist-icon.png')

img\_8 = img\_8.resize((30,30))

img\_8 = ImageTk.PhotoImage(img\_8)

new\_window\_button = Button(lyrics\_frame,width = 40 , height = 40, image=img\_8, padx=20 , bg= co1,font = ("Ivy 10"), command=create\_new\_window)

new\_window\_button.place(x=285, y=35)

new\_window\_button.pack()

mixer.init()

music\_state = StringVar()

music\_state.set("Choose one!")

window.mainloop()

Output :