Section:-K20RG **Project No:-CA1**

Final Report

MUSIC PLAYER.

Internet Programming Laboratory (CSE 326)

By

Sr. No.	Registration No	Name of Student	Roll No	Total Marks	Marks Obtained	Signature
1	12007529	RITIK PANDEY	RK20RGA21	30		

Transforming Education Transforming India

Submitted To

Ankita Wadhawan

School of Computer Science and Engineering

Lovely Professional University

Jalandhar, Punjab, India.

Table of Contents

ABSTRACT

Music players are media software that are specifically designed to play audio files. These tools support a wide range of music formats, including MP3, WAV, and WMA. Such applications help you to organize your song library with ease. In this project I have created complex but simple gui music player system,In which I have used the tkinter, pygame, os, pickle. In which I have created many function like play button, stop button, next button,select music button and volume range button which is from 0 to 100 and also I created playlist box where you can store your favourite song.

ACKNOWLEDGEMENT

I would like to thanks my teacher Ankita Wadhawan For her advice and inputs on this project and also my mentors.Many Thanks to my friends and seniors as well, who spent countless hours to listen and provide feedbacks.

INTRODUCTION:

Python:

It is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding on Python programming language. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

Tkinter:

Tkinter tutorial provides basic and advanced concepts of Python Tkinter. Our Tkinter tutorial is designed for beginners and professionals.Python provides the standard library Tkinter for creating the graphical user interface for desktop based applications.Developing desktop based applications with python Tkinter is not a complex task.

tkinter also offers access to the geometric configuration of the widgets which can organize the widgets in the parent windows. There are mainly three geometry manager classes class.

1. **pack() method:It organizes the widgets in blocks before placing in the parent widget.**
2. **grid() method:It organizes the widgets in grid (table-like structure) before placing in the parent widget.**
3. **place() method:It organizes the widgets by placing them on specific positions directed by the programmer.**

PYGAME:

- Pygame has an inbuilt method called *mixer ()* which provides us intuitive syntax on dealing with sounds files in python. It consists of computer graphics and sound libraries designed to be used with the Python programming language.

- loading and playing music
- pausing and unpausing music
- stoping the music file

Music Player:

This project has been done as part of my course for the CSE213 at Lovely Professional University . Supervised by Ankita Wadhawan, I have two months to fulfill the requirements in order to succeed the module.

It is scientifically proven that listening music daily can increase our brain power and constration level and also I have special inclination toward music,so I opt to choose creating a music player in python.

Music is one of the best ways to relieve pressure in stressful modern society life. The purpose of this project is to develop a player which can play the mainstream file format. The graphic user interface of the application is very simple and attractive and easy to understand.

To browse and query the storage space as well as operation of playing can be realised. Meanwhile, this software can play, pause and select songs with back button and next button and also you can set the volume level according to sets requirement as well as set up songs. Even You can add any songs of your choice from your computer. There is an option which is the 'Add' button next to the song title, it represents to add the song to the playlist created by users. In "Music Player" users can create their own music world by adding favorite songs into the playlist. All playlists are free to create and unlimited in number. It performs extremely well in terms of functionality and interaction between users and application.

OBJECTIVES

The purpose of this project is to develop a player which can play the mainstream file format. To browse and query the storage space as well as operation of playing can be realised. Meanwhile, this software can play, pause and select songs with back button and next button and also you can set the volume level according to sets requirement as well as set up songs. Even You can add any songs of your choice from your computer. There is an option which is the 'Add' button next to the song title, it represents to add the song to the playlist created by users.

TO DO LIST:-

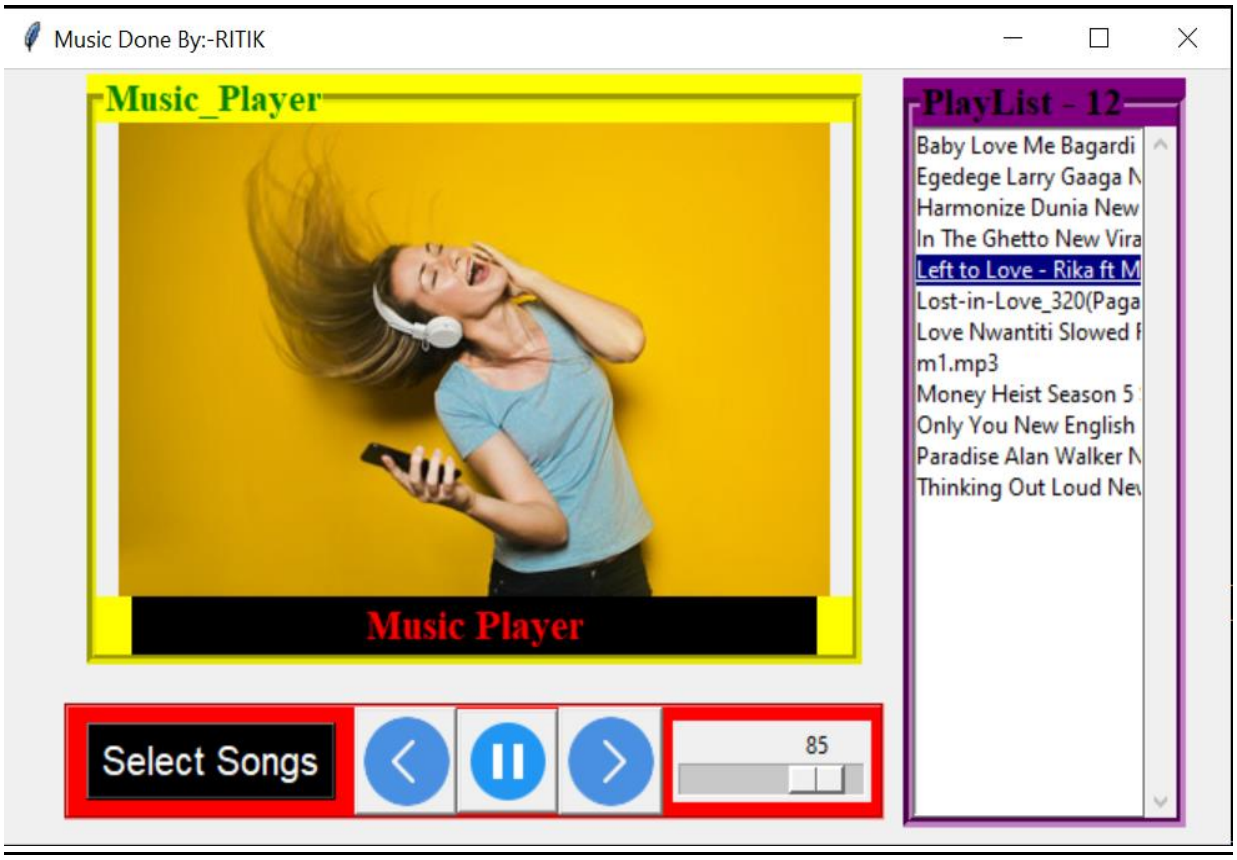
- Making GUI Interface
- Making play button
- Making back button
- Making pause button
- Making volume button
- Making playlist
- Making select song button

CODE

```python
import os
import pickle
import tkinter as tk
from tkinter import filedialog
from tkinter import PhotoImage
from pygame import mixer

class Player(tk.Frame):
    def __init__(self, master=None):
        super().__init__(master)
        self.master = master
        self.pack()
        mixer.init()

        if os.path.exists('songs.pickle'):
            with open('songs.pickle', 'rb') as f:
                self.playlist = pickle.load(f)
        else:
            self.playlist=[]

        self.current = 0
        self.paused = True
        self.played = False

        self.create_frames()
        self.track_widgets()
        self.control_widgets()
        self.tracklist_widgets()

    def create_frames(self):
        self.track = tk.LabelFrame(self, text='Music_Player',
                    font=("times new roman",15,"bold"),
                    bg="yellow",fg="green",bd=5,relief=tk.GROOVE)
        self.track.config(width=410,height=300)
        self.track.grid(row=0, column=0, padx=10)

        self.tracklist = tk.LabelFrame(self, text=f'PlayList -
{str(len(self.playlist))}',
                            font=("times new roman",15,"bold"),
                            bg="purple",fg="black",bd=5,relief=tk.GROOVE)
        self.tracklist.config(width=190,height=400)
        self.tracklist.grid(row=0, column=1, rowspan=3, pady=5)

        self.controls = tk.LabelFrame(self,
                            font=("times new roman",15,"bold"),
                            bg="red",fg="red",bd=2,relief=tk.GROOVE)
        self.controls.config(width=410,height=80)
        self.controls.grid(row=2, column=0, pady=5, padx=10)

    def track_widgets(self):
        self.canvas = tk.Label(self.track, image=img)
        self.canvas.configure(width=400, height=240)
        self.canvas.grid(row=0,column=0)

        self.songtrack = tk.Label(self.track, font=("times new
roman",16,"bold"),
                        bg="black",fg="red")
        self.songtrack['text'] = 'Music Player'
        self.songtrack.config(width=30, height=1)
        self.songtrack.grid(row=1,column=0,padx=10)

    def control_widgets(self):
        self.loadSongs = tk.Button(self.controls, bg='black', fg='white',
font=10)
        self.loadSongs['text'] = 'Select Songs'
        self.loadSongs['command'] = self.retrieve_songs
        self.loadSongs.grid(row=0, column=0, padx=10)

        self.prev = tk.Button(self.controls, image=prev)
        self.prev['command'] = self.prev_song
        self.prev.grid(row=0, column=1)

        self.pause = tk.Button(self.controls, image=pause)
        self.pause['command'] = self.pause_song
        self.pause.grid(row=0, column=2)

        self.next = tk.Button(self.controls, image=next_)
        self.next['command'] = self.next_song
        self.next.grid(row=0, column=3)

        self.volume = tk.DoubleVar(self)
        self.slider = tk.Scale(self.controls, from_ = 0, to = 100, orient =
tk.HORIZONTAL)
        self.slider['variable'] = self.volume
        self.slider.set(8)
        mixer.music.set_volume(0.8)
        self.slider['command'] = self.change_volume
        self.slider.grid(row=0, column=4, padx=5)

    def tracklist_widgets(self):
        self.scrollbar = tk.Scrollbar(self.tracklist, orient=tk.VERTICAL)
        self.scrollbar.grid(row=0,column=1, rowspan=5, sticky='ns')
        self.list = tk.Listbox(self.tracklist, selectmode=tk.SINGLE,
                     yscrollcommand=self.scrollbar.set,
selectbackground='navy')
        self.enumerate_songs()
        self.list.config(height=22)
        self.list.bind('<Double-1>', self.play_song)

        self.scrollbar.config(command=self.list.yview)
        self.list.grid(row=0, column=0, rowspan=5)

    def retrieve_songs(self):
        self.songlist = []
        directory = filedialog.askdirectory()
        for root_, dirs, files in os.walk(directory):
                for file in files:
                    if os.path.splitext(file)[1] == '.mp3':
                        path = (root_ + '/' + file).replace('\\','/')
                        self.songlist.append(path)

        with open('songs.pickle', 'wb') as f:
            pickle.dump(self.songlist, f)
        self.playlist = self.songlist
        self.tracklist['text'] = f'PlayList - {str(len(self.playlist))}'
        self.list.delete(0, tk.END)
        self.enumerate_songs()

    def enumerate_songs(self):
        for index, song in enumerate(self.playlist):
            self.list.insert(index, os.path.basename(song))


    def play_song(self, event=None):
        if event is not None:
            self.current = self.list.curselection()[0]
            for i in range(len(self.playlist)):
                self.list.itemconfigure(i, bg="white")

        print(self.playlist[self.current])
        mixer.music.load(self.playlist[self.current])
        self.songtrack['anchor'] = 'w'
        self.songtrack['text'] = os.path.basename(self.playlist[self.current])

        self.pause['image'] = play
        self.paused = False
        self.played = True
        self.list.activate(self.current)
        self.list.itemconfigure(self.current, bg='navy')
        mixer.music.play()

    def pause_song(self):
        if not self.paused:
            self.paused = True
            mixer.music.pause()
            self.pause['image'] = pause
        else:
            if self.played == False:
                self.play_song()
            self.paused = False
            mixer.music.unpause()
            self.pause['image'] = play

    def prev_song(self):
        if self.current > 0:
            self.current -= 1
        else:
            self.current = 0
        self.list.itemconfigure(self.current + 1, bg='white')
        self.play_song()

    def next_song(self):
        if self.current < len(self.playlist) - 1:
            self.current += 1
        else:
            self.current = 0
        self.list.itemconfigure(self.current - 1, bg='white')
        self.play_song()

    def change_volume(self, event=None):
        self.v = self.volume.get()
        mixer.music.set_volume(self.v / 10)

# ----------------------------- Main ------------------------------------------
--

root = tk.Tk()
root.geometry('600x400')
root.wm_title('Music Done By:-RITIK')

img = PhotoImage(file='5.png')
next_ = PhotoImage(file ='i2@.png')
prev = PhotoImage(file='i3.png')
play = PhotoImage(file='play.gif')
pause = PhotoImage(file='pause.png')

app = Player(master=root)
```

```
app.mainloop()
```

INTERFACE OF MUSIC PLAYER:-



Baby Love Me Bagardi
Egedege Larry Gaaga N
Harmonize Dunia New
In The Ghetto New Vira
Left to Love - Rika ft M
Lost-in-Love_320(Paga
Love Nwantiti Slowed R
m1.mp3
Money Heist Season 5
Only You New English
Paradise Alan Walker N
Thinking Out Loud New

CONCULATION:-

I have created Music Player from python.It has graphic user interface. The graphic user interface of the application is very simple and attractive and easy to understand and learn.In this application you can play, pause and select songs with back button, stop button and next button and also you can set the volume level according to your required range from 0 to 100.You and easily add unlimited your favourite song from your computer by pressing select songs and store in playlist. And I have learn a lot things from this project..

REFERENCES:-

- WIKIPEDIA
- YOUTUBE
- SPOTIFY