```python
##
# Hello Welcome to Harmony Source Code, based on pygame, tkinter and various other modules.
# Please note the minimum required modules : mutagen, pygame, ttkthemes
# To install you must use: "pip install *" as * followed by the right module syntax.
# Coded to work on Python 3.8
# Last test on Python 3.8
# Might need some changed for older versions.
# Written by Ayushmaan Karmokar.
##

# Code Quality - A++
# Last Edited - 2 Feb, 2020
# Check my Github on - ***

# modules
import os
import threading
import time
import tkinter.messagebox
from tkinter import *
from tkinter import filedialog

from tkinter import ttk
from ttkthemes import themed_tk as tk

from mutagen.mp3 import MP3
from pygame import mixer

root = tk.ThemedTk()
root.get_themes()                 # Returns a list of all themes that can be set
root.set_theme("black")         # Sets an available theme

# Fonts - Arial (corresponds to Helvetica), Courier New (Courier), Comic Sans MS, Fixedsys,
# MS Sans Serif, MS Serif, Symbol, System, Times New Roman (Times), and Verdana
#
# Styles - normal, bold, roman, italic, underline, and overstrike.

statusbar = ttk.Label(root, text="Welcome to Harmony (nImEHuntetD)", relief=SUNKEN, anchor=W, font='Times 10
italic')
statusbar.pack(side=BOTTOM, fill=X)

# Create the menubar
menubar = Menu(root)
root.config(menu=menubar)

# Create the submenu

subMenu = Menu(menubar, tearoff=0)

playlist = []


# playlist - contains the full path + filename
# playlistbox - contains just the filename
# Fullpath + filename is required to play the music inside play_music load function

def browse_file():
    global filename_path
    filename_path = filedialog.askopenfilename()
    add_to_playlist(filename_path)
    mixer.music.queue(filename_path)


def add_to_playlist(filename):
    filename = os.path.basename(filename)
    index = 0
    playlistbox.insert(index, filename)
    playlist.insert(index, filename_path)
    index += 1

def basic_cleanup():
    global default_music_add
    global filename_test
    default_music_add = "copyright_free_music_for_your_ears/Lost Sky - Dreams.mp3"
    filename_test = default_music_add
    filename_test = os.path.basename(filename_test)
    index = 0
    playlistbox.insert(index, filename_test)
    playlist.insert(index, default_music_add)
    index += 1
    mixer.music.queue(default_music_add)

```

```python


menubar.add_cascade(label="File", menu=subMenu)
subMenu.add_command(label="Open", command=browse_file)
subMenu.add_command(label="Exit", command=root.destroy)


def about_us():
    tkinter.messagebox.showinfo('About Harmony', 'Open Source Python Based Music Player by @ayushmaan_karmokar')

def how_to_use():
    tkinter.messagebox.showinfo('How to Use!', 'Simple, Click on File>Open and choose the song(s) you want to add. Alternatively you can also drag and drop. \nSelect the music on the left, before using the control buttons. \n CHEERS!')


subMenu = Menu(menubar, tearoff=0)
menubar.add_cascade(label="Help", menu=subMenu)
subMenu.add_command(label="About Us", command=about_us)
subMenu.add_command(label="How to Use?", command=how_to_use)

mixer.init()  # initializing the mixer

root.title("Harmony")
root.iconbitmap(r'images/melody.ico')

# Root Window - StatusBar, LeftFrame, RightFrame
# LeftFrame - The listbox (playlist)
# RightFrame - TopFrame,MiddleFrame and the BottomFrame

leftframe = Frame(root)
leftframe.pack(side=LEFT, padx=30, pady=30)

playlistbox = Listbox(leftframe)
playlistbox.pack()

addBtn = ttk.Button(leftframe, text="Add Song(s)", command=browse_file)
addBtn.pack(side=LEFT)


def del_song():
    selected_song = playlistbox.curselection()
    selected_song = int(selected_song[0])
    playlistbox.delete(selected_song)
    playlist.pop(selected_song)


delBtn = ttk.Button(leftframe, text="Del Song", command=del_song)
delBtn.pack(side=LEFT)

rightframe = Frame(root)
rightframe.pack(pady=30)

#logoframe = PhotoImage(file='images/logo_big.jpg')            # test do not change yet !
#logoframe.pack()

topframe = Frame(rightframe)
topframe.pack()

'''logoframe = PhotoImage(file='images/logo_big.png')
logoframe=Frame(rightframe)
logoframe.pack()'''

lengthlabel = ttk.Label(topframe, text='Total Length : --:--')
lengthlabel.pack(pady=5)

currenttimelabel = ttk.Label(topframe, text='Current Time : --:--', relief=GROOVE)
currenttimelabel.pack()


def show_details(play_song):
    file_data = os.path.splitext(play_song)

    if file_data[1] == '.mp3':
        audio = MP3(play_song)
        total_length = audio.info.length
    else:
        a = mixer.Sound(play_song)
        total_length = a.get_length()

    # div - total_length/60, mod - total_length % 60
    mins, secs = divmod(total_length, 60)
    mins = round(mins)
    secs = round(secs)
    timeformat = '{:02d}:{:02d}'.format(mins, secs)
    lengthlabel['text'] = "Total Length" + ' - ' + timeformat

    t1 = threading.Thread(target=start_count, args=(total_length,))
    t1.start()


def start_count(t):
    global paused
    # mixer.music.get_busy(): - Returns FALSE when we press the stop button (music stop playing)
    # Continue - Ignores all of the statements below it. We check if music is paused or not.
    current_time = 0
    while current_time <= t and mixer.music.get_busy():
        if paused:
            continue
        else:
            mins, secs = divmod(current_time, 60)
            mins = round(mins)
            secs = round(secs)
            timeformat = '{:02d}:{:02d}'.format(mins, secs)
            currenttimelabel['text'] = "Current Time" + ' - ' + timeformat
            time.sleep(1)
            current_time += 1


def play_music():
    global paused

    if paused:
        mixer.music.unpause()
        statusbar['text'] = "Music Resumed"
        paused = FALSE
    else:
        try:
            stop_music()
            time.sleep(1)
            selected_song = playlistbox.curselection()
            selected_song = int(selected_song[0])
            play_it = playlist[selected_song]
            mixer.music.load(play_it)
            mixer.music.play()
            statusbar['text'] = "Playing..>" + ' - ' + os.path.basename(play_it)
            show_details(play_it)
        except:
            tkinter.messagebox.showerror('File not found', 'Harmony could not find the file. Please check again.')


def stop_music():
    mixer.music.stop()
    statusbar['text'] = "STOP!!"


paused = FALSE


def pause_music():
    global paused
    paused = TRUE
    mixer.music.pause()
    statusbar['text'] = "Pause..||"


def rewind_music():
    play_music()
    statusbar['text'] = "Rewind..<<"


def set_vol(val):
    volume = float(val) / 100
    mixer.music.set_volume(volume)
    # set_volume of mixer takes value only from 0 to 1. Example - 0, 0.1,0.55,0.54.0.99,1


muted = FALSE


def mute_music():
    global muted
    if muted:  # Unmute the music
        mixer.music.set_volume(0.7)
        volumeBtn.configure(image=volumePhoto)
        scale.set(70)
        muted = FALSE
    else:  # mute the music
        mixer.music.set_volume(0)
        volumeBtn.configure(image=mutePhoto)
        scale.set(0)
        muted = TRUE


middleframe = Frame(rightframe)
middleframe.pack(pady=30, padx=30)

logoframe = PhotoImage(file='images/logo_big.png')
logoBtn = ttk.Button(middleframe, image=logoframe)   # command=play_music
logoBtn.grid(row=0, column=0, padx=10)

playPhoto = PhotoImage(file='images/play.png')
playBtn = ttk.Button(middleframe, image=playPhoto, command=play_music)
playBtn.grid(row=0, column=1, padx=10)

stopPhoto = PhotoImage(file='images/stop.png')
stopBtn = ttk.Button(middleframe, image=stopPhoto, command=stop_music)
stopBtn.grid(row=0, column=2, padx=10)

pausePhoto = PhotoImage(file='images/pause.png')
pauseBtn = ttk.Button(middleframe, image=pausePhoto, command=pause_music)
pauseBtn.grid(row=0, column=3, padx=10)

# Bottom Frame for volume, rewind, mute etc.

bottomframe = Frame(rightframe)
bottomframe.pack()

rewindPhoto = PhotoImage(file='images/rewind.png')
rewindBtn = ttk.Button(bottomframe, image=rewindPhoto, command=rewind_music)
rewindBtn.grid(row=0, column=0)

mutePhoto = PhotoImage(file='images/mute.png')
volumePhoto = PhotoImage(file='images/volume.png')
volumeBtn = ttk.Button(bottomframe, image=volumePhoto, command=mute_music)
volumeBtn.grid(row=0, column=1)
```

```python

scale = ttk.Scale(bottomframe, from_=0, to=100, orient=HORIZONTAL, command=set_vol)
scale.set(70)  # implement the default value of scale when music player starts
mixer.music.set_volume(0.7)
scale.grid(row=0, column=2, pady=15, padx=30)


def on_closing():
    stop_music()
    root.destroy()


# test
basic_cleanup()

root.protocol("WM_DELETE_WINDOW", on_closing)
root.mainloop()

```