

```

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()
root.set_theme("plastik") # Change the theme to 'plastik'

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

menubar = Menu(root)
root.config(menu=menubar)
subMenu = Menu(menubar, tearoff=0)

playlist = []

def browse_file():
    global filename_path
    filename_path = filedialog.askopenfilename()
    add_to_playlist(filename_path)
    mixer.music.queue(filename_path) 56

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

```

```

index += 1

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 Melody', 'This is a music player build using Python Tkinter by
@attreyabhata')

subMenu = Menu(menubar, tearoff=0)

menubar.add_cascade(label="Help", menu=subMenu)

subMenu.add_command(label="About Us", command=about_us)

mixer.init()

root.title("Melody")

root.iconbitmap(r'images/melody.ico')

leftframe = Frame(root, bg='#ADD8E6') # Light blue background color

leftframe.pack(side=LEFT, padx=30, pady=30) 57

playlistbox = Listbox(leftframe)

playlistbox.pack()

addBtn = ttk.Button(leftframe, text="+ Add", 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", command=del_song)

delBtn.pack(side=LEFT)

rightframe = Frame(root, bg='#ADD8E6') # Light blue background color

rightframe.pack(pady=30)

topframe = Frame(rightframe, bg='#ADD8E6') # Light blue background color

topframe.pack()

```

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

currenttimelabel = ttk.Label(topframe, text='Current Time : --:--', relief=GROOVE)
currenttimelabel.pack() 58
```

```
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()
    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
    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" + ' - ' + 59
```

```
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 music" + ' - ' + os.path.basename(play_it)
```

```
            show_details(play_it)
```

```
        except:
```

```
            tkinter.messagebox.showerror('File not found', 'Melody could not find the file. Please check again.')
```

```
def stop_music():
```

```
    mixer.music.stop()
```

```
    statusbar['text'] = "Music Stopped" 60
```

```
    paused = FALSE
```

```
def pause_music():
```

```
    global paused
```

```

paused = TRUE

mixer.music.pause()

statusbar['text'] = "Music Paused"

def rewind_music():
    play_music()

statusbar['text'] = "Music Rewinded"

def set_vol(val):
    volume = float(val) / 100
    mixer.music.set_volume(volume)

muted = FALSE

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

middleframe = Frame(rightframe, bg='#ADD8E6') # Light blue background color
middleframe.pack(pady=30, padx=30)

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

stopPhoto = PhotoImage(file='images/stop.png')
stopBtn = ttk.Button(middleframe, image=stopPhoto, command=stop_music)
stopBtn.grid(row=0, column=1, padx=10)

```

```
pausePhoto = PhotoImage(file='images/pause.png')
pauseBtn = ttk.Button(middleframe, image=pausePhoto, command=pause_music)
pauseBtn.grid(row=0, column=2, padx=10)

bottomframe = Frame(rightframe, bg='#ADD8E6') # Light blue background color
bottomframe.pack() 62

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)

scale = ttk.Scale(bottomframe, from_=0, to=100, orient=HORIZONTAL, command=set_vol)
scale.set(70)

mixer.music.set_volume(0.7)
scale.grid(row=0, column=2, pady=15, padx=30)

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

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