Software Project Report

Amitabha Ghosh
EE22BTECH11203

AIM :- To make an audioplayer in which songs are played in random order.

Scripting Language :- Python (3.10.6)

Library Used :-

1) Numpy Library,
2) Playsound Library.

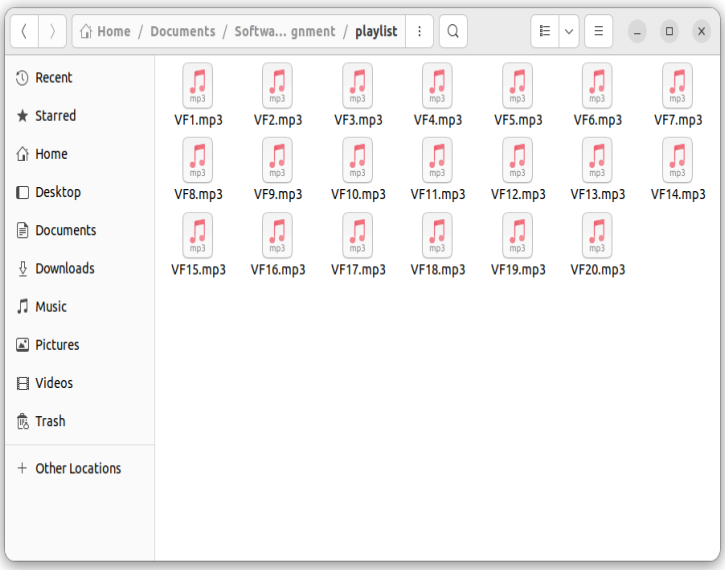

Image of the playlist folder

Mechanism :-

1) Import numpy, pygame, random and tkinter library.
2) Give the directory of the songs so that songs can be accessed.
3) Making a list of audio files.
4) Shuffling the audio randomly.
5) Play the songs according to their position is the playlist.
6) If all the songs are played once in the loop, then a new random list of the songs is generated.
7) This process continues till the program is closed.

```
import pygame
import random
import os
import numpy as np
import soundfile as sf
from tkinter import Tk, filedialog
import tkinter as tk

os.environ['SDL_VIDEODRIVER'] = 'x11'

def choose_music_folder():
    Tk().withdraw()
    folder_path = filedialog.askdirectory()
    return folder_path

def get_music_files(folder_path):
    music_files = []
    for file in os.listdir(folder_path):
        if file.endswith(".mp3"):
            music_files.append(os.path.join(folder_path, file))
    return music_files

def play_next_song():
    global current_song_index
    current_song_index = (current_song_index + 1) % len(playlist)
    pygame.mixer.music.load(playlist[current_song_index])
    pygame.mixer.music.play()

def update_current_song_label():
    current_song_label.config(text=f"Current Song: {os.path.basename(playlist[current_song_index])}")

def play_pause():
    global paused
    if paused:
        pygame.mixer.music.unpause()
        paused = False
        play_pause_button.config(text="Pause Current Song")
    else:
        pygame.mixer.music.pause()
        paused = True
        play_pause_button.config(text="Play Song")

def next_song():
    play_next_song()
    update_current_song_label()
```

Conclusion :- In this way we have created a program which plays songs randomly, selected from the created playlist.

Output Images :-

```
def next_song():
    play_next_song()
    update_current_song_label()

def quit_music_player():
    pygame.mixer.music.stop()
    window.quit()

pygame.init()
music_folder = choose_music_folder()
playlist = get_music_files(music_folder)
current_song_index = 0
window = tk.Tk()
window.title("Song Playlist")
window.geometry("400x200")
current_song_label = tk.Label(window, text="Current Song: ")
current_song_label.pack()

paused = False
play_pause_button = tk.Button(window, text="Pause Current Song", command=play_pause)
play_pause_button.pack()

next_song_button = tk.Button(window, text="Play Next Song", command=next_song)
next_song_button.pack()


quit_button = tk.Button(window, text="Quit", command=quit_music_player)
quit_button.pack()


update_current_song_label()

pygame.mixer.init()
pygame.mixer.music.load(playlist[current_song_index])
pygame.mixer.music.play()

window.mainloop()

pygame.quit()
```

Images of the code

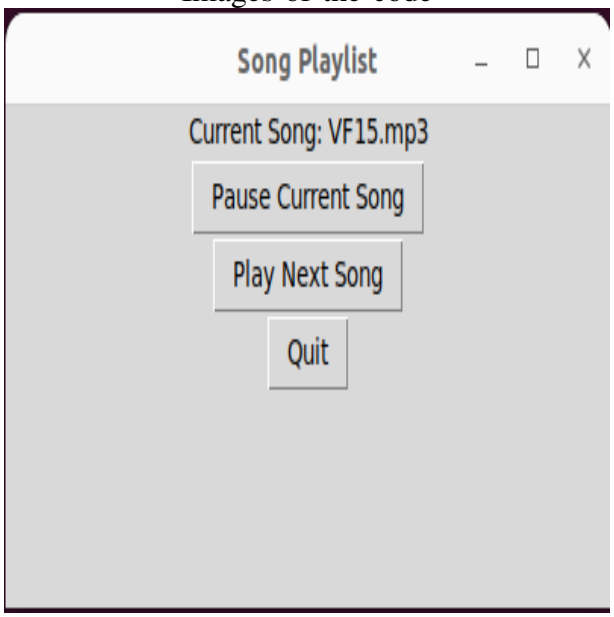

Image of the output of the code