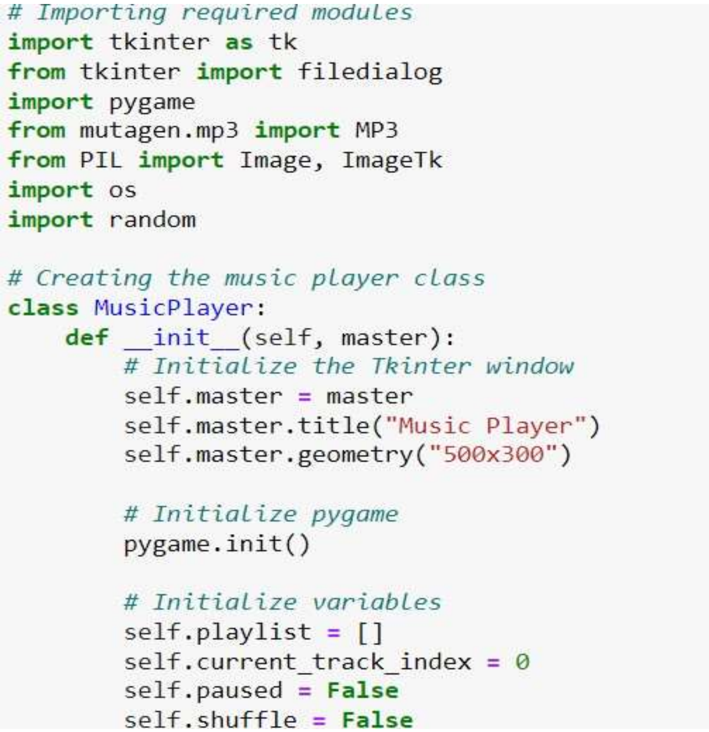
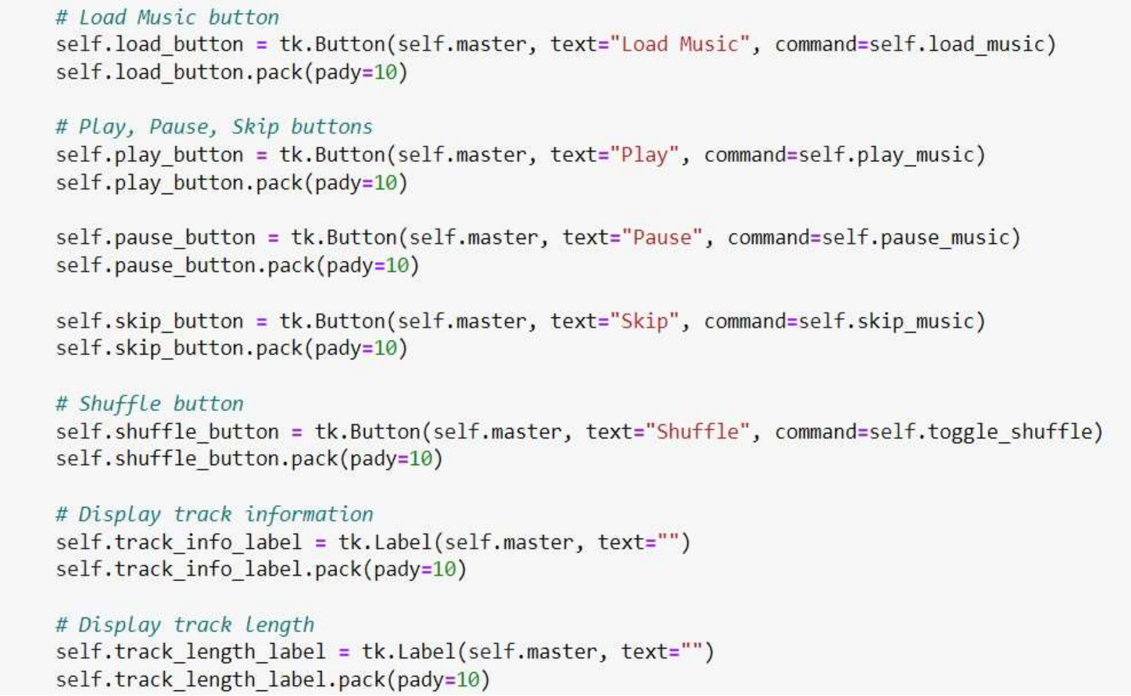
Tkinter Project

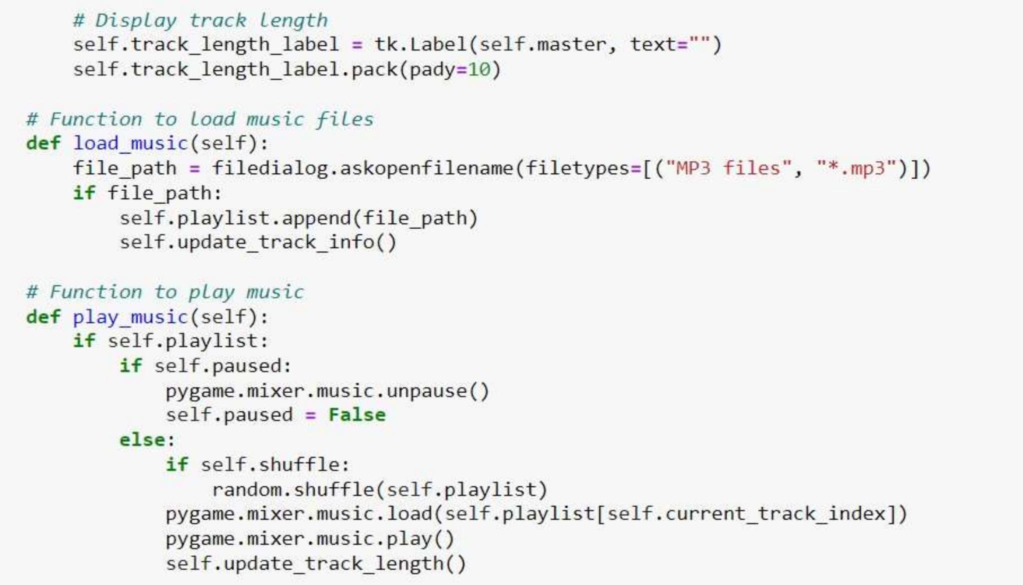
Created a MUSIC PLAYER using Python



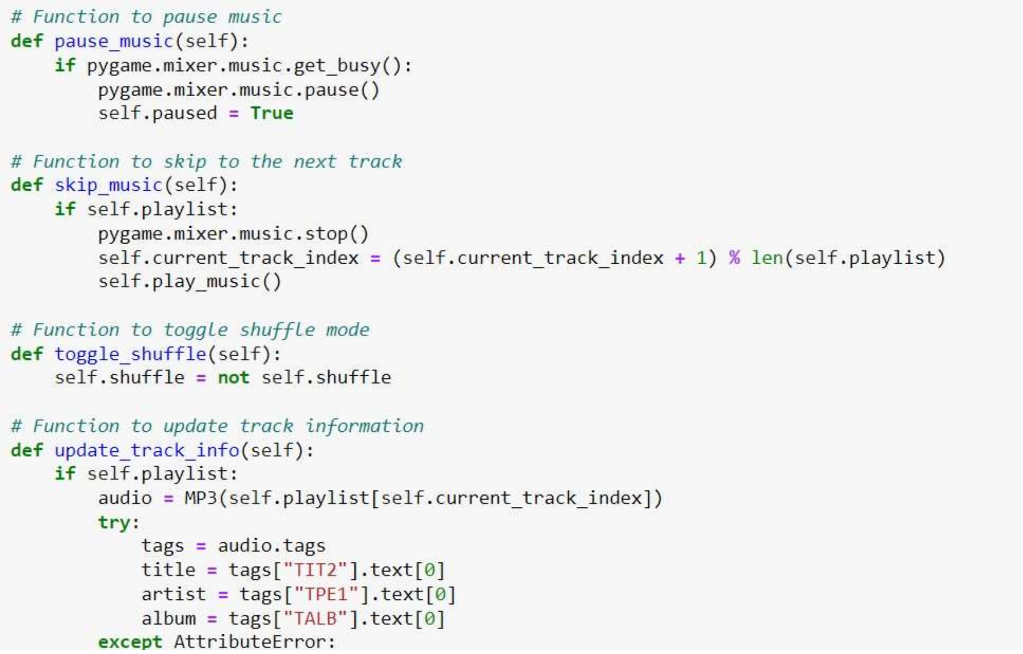
```
# Importing required modules
import tkinter as tk
from tkinter import filedialog
import pygame
from mutagen.mp3 import MP3
from PIL import Image, ImageTk
import os
import random

# Creating the music player class
class MusicPlayer:
    def __init__(self, master):
        # Initialize the Tkinter window
        self.master = master
        self.master.title("Music Player")
        self.master.geometry("500x300")

        # Initialize pygame
        pygame.init()

        # Initialize variables
        self.playlist = []
        self.current_track_index = 0
        self.paused = False
        self.shuffle = False
```



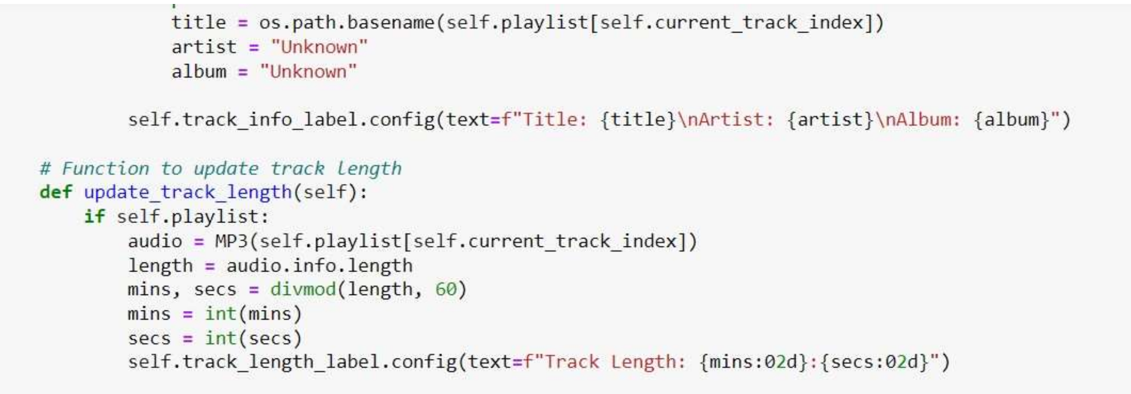
```
    # Load Music button
    self.load_button = tk.Button(self.master, text="Load Music", command=self.load_music)
    self.load_button.pack(pady=10)

    # Play, Pause, Skip buttons
    self.play_button = tk.Button(self.master, text="Play", command=self.play_music)
    self.play_button.pack(pady=10)

    self.pause_button = tk.Button(self.master, text="Pause", command=self.pause_music)
    self.pause_button.pack(pady=10)

    self.skip_button = tk.Button(self.master, text="Skip", command=self.skip_music)
    self.skip_button.pack(pady=10)

    # Shuffle button
    self.shuffle_button = tk.Button(self.master, text="Shuffle", command=self.toggle_shuffle)
    self.shuffle_button.pack(pady=10)

    # Display track information
    self.track_info_label = tk.Label(self.master, text="")
    self.track_info_label.pack(pady=10)

    # Display track length
    self.track_length_label = tk.Label(self.master, text="")
    self.track_length_label.pack(pady=10)
```



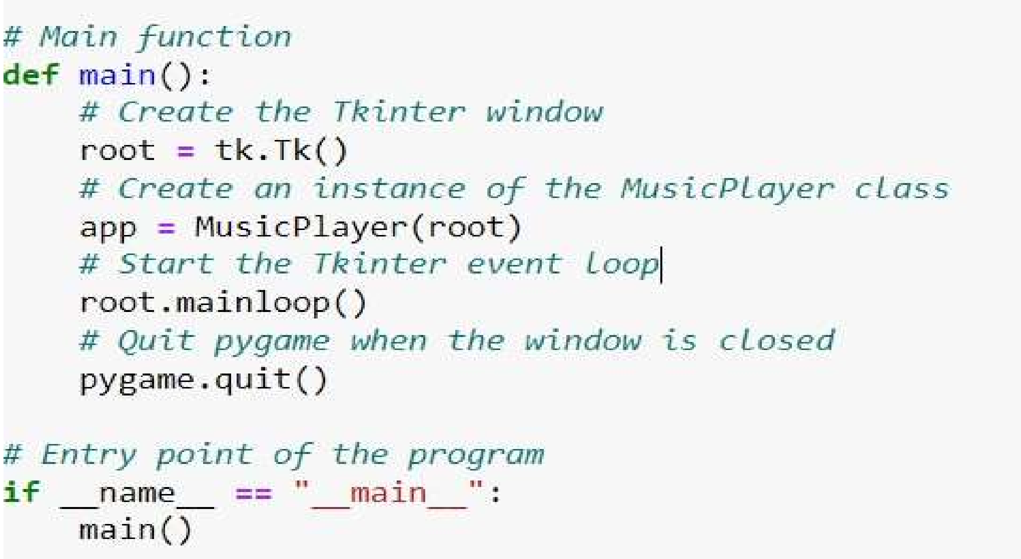
```
        # Display track length
        self.track_length_label = tk.Label(self.master, text="")
        self.track_length_label.pack(pady=10)

    # Function to load music files
    def load_music(self):
        file_path = filedialog.askopenfilename(filetypes=[("MP3 files", "*.mp3")])
        if file_path:
            self.playlist.append(file_path)
            self.update_track_info()

    # Function to play music
    def play_music(self):
        if self.playlist:
            if self.paused:
                pygame.mixer.music.unpause()
                self.paused = False
            else:
                if self.shuffle:
                    random.shuffle(self.playlist)
                pygame.mixer.music.load(self.playlist[self.current_track_index])
                pygame.mixer.music.play()
                self.update_track_length()
```



```
# Function to pause music
def pause_music(self):
    if pygame.mixer.music.get_busy():
        pygame.mixer.music.pause()
        self.paused = True

# Function to skip to the next track
def skip_music(self):
    if self.playlist:
        pygame.mixer.music.stop()
        self.current_track_index = (self.current_track_index + 1) % len(self.playlist)
        self.play_music()

# Function to toggle shuffle mode
def toggle_shuffle(self):
    self.shuffle = not self.shuffle

# Function to update track information
def update_track_info(self):
    if self.playlist:
        audio = MP3(self.playlist[self.current_track_index])
        try:
            tags = audio.tags
            title = tags["TIT2"].text[0]
            artist = tags["TPE1"].text[0]
            album = tags["TALB"].text[0]
        except AttributeError:
```

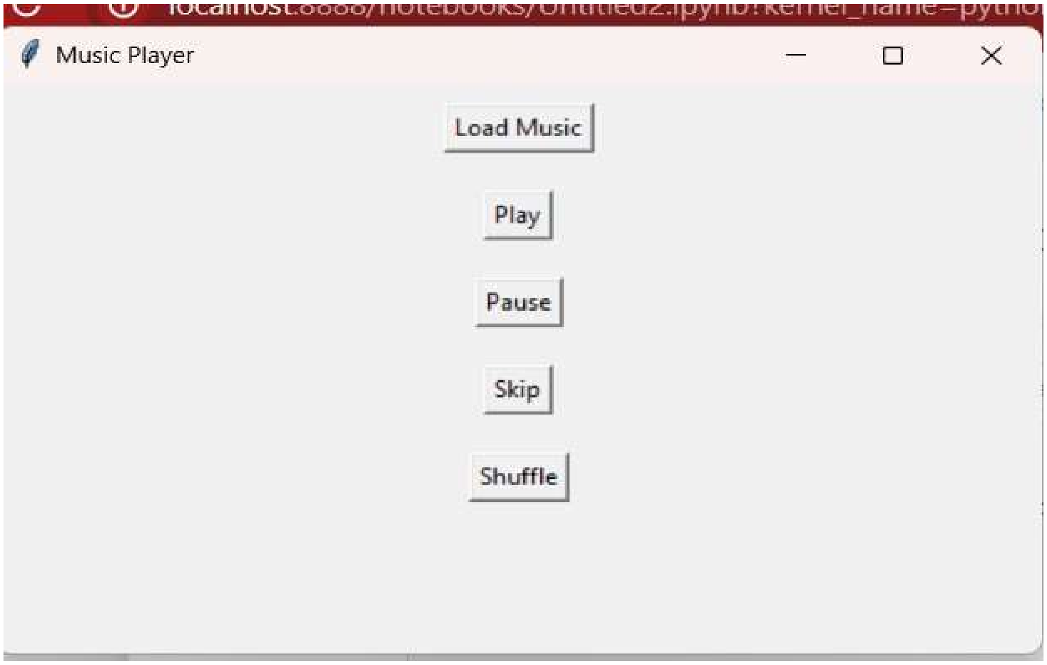


```
                title = os.path.basename(self.playlist[self.current_track_index])
                artist = "Unknown"
                album = "Unknown"

            self.track_info_label.config(text=f"Title: {title}\nArtist: {artist}\nAlbum: {album}")

    # Function to update track length
    def update_track_length(self):
        if self.playlist:
            audio = MP3(self.playlist[self.current_track_index])
            length = audio.info.length
            mins, secs = divmod(length, 60)
            mins = int(mins)
            secs = int(secs)
            self.track_length_label.config(text=f"Track Length: {mins:02d}:{secs:02d}")
```



```python
# Main function
def main():
    # Create the Tkinter window
    root = tk.Tk()
    # Create an instance of the MusicPlayer class
    app = MusicPlayer(root)
    # Start the Tkinter event loop
    root.mainloop()
    # Quit pygame when the window is closed
    pygame.quit()

# Entry point of the program
if __name__ == "__main__":
    main()
```

OUTPUT:


localhost:8888/notebooks/Untitled2.ipynb?kernel_name=pytho
Music Player
Load Music
Play
Pause
Skip
Shuffle