**TRƯỜNG ĐẠI HỌC AN GIANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO ĐỒ ÁN**

**MÔN: Chuyên Đề Python (COS525)**

**ỨNG DỤNG PHÁT/TẢI**

**NHẠC, VIDEO, LỜI BÀI HÁT**

**SINH VIÊN THỰC HIỆN: DTH225757 - Lê Hoàng Tuấn**

**NHÓM BÁO CÁO ĐỒ ÁN: Nhóm cá nhân**

**AN GIANG, 12-2024**

**Mục Lục**

[I. Lời nói đầu 2](#_Toc11543)

[II. Mục tiêu 3](#_Toc3486)

[III. Các thư viện cần thiết 3](#_Toc24857)

[IV. Cấu trúc chương trình 3](#_Toc2920)

[1. Các chức năng chính: 3](#_Toc21100)

[ Quản lý thư viện nhạc và video 3](#_Toc20217)

[2. Mã Nguồn chương trình: 4](#_Toc14902)

[a) Khai báo thư viện 4](#_Toc5317)

[b) Khởi tạo chương trình 4](#_Toc13171)

[c) Tạo các widget 5](#_Toc711)

[d) Cài đặt cửa sổ music 6](#_Toc12126)

[e) Cài đặt cửa sổ video 7](#_Toc17364)

[f) Cài đặt cửa sổ lyric 7](#_Toc14833)

[g) Các nút tương tác của Music 8](#_Toc30348)

[h) Các nút tương tác của Video 8](#_Toc30616)

[i) Chuyển đổi cửa sổ 9](#_Toc21710)

[j) Thêm thư mục nhạc 9](#_Toc19790)

[k) Thêm tệp nhạc 10](#_Toc24540)

[l) Thêm thư mục video 10](#_Toc28339)

[m) Thêm tệp video 10](#_Toc32406)

[n) Tải nhạc 10](#_Toc10730)

[o) Tải video 11](#_Toc9830)

[p) Xóa nhạc, video 11](#_Toc18231)

[q) Trích xuất lời bài hát 12](#_Toc11492)

[r) Chuyển đổi mp4 sang mp3 12](#_Toc17076)

[s) Hiển thị nhạc 12](#_Toc3890)

[t) Hiển thị video 13](#_Toc6299)

[u) Hiển thị lyric 13](#_Toc9634)

[v) Chọn nổi bật 13](#_Toc24824)

[w) Cuộn chuột 13](#_Toc11559)

[x) Thêm placeholder 14](#_Toc7608)

[y) Thực thi chương trình 14](#_Toc7574)

[V. Các thay đổi và cải tiến so với bài mẫu 14](#_Toc30328)

[VI. Hướng dẫn sử dụng 15](#_Toc15056)

[VII. Tiềm năng phát triển 15](#_Toc14206)

[VIII. Tài liệu tham khảo 15](#_Toc66)

# I. Lời nói đầu

Với sự phát triển mạnh mẽ của công nghệ, các ứng dụng phát nhạc, xem video và hỗ trợ người dùng trở nên phổ biến và thiết yếu. Đồ án này phát triển phần mềm mavPlayer, một trình phát nhạc, video, tải xuống nội dung từ YouTube và hỗ trợ trích xuất lời bài hát.

# II. Mục tiêu

Mục tiêu của đồ án là xây dựng một ứng dụng đa năng với các chức năng chính sau:  
- Phát nhạc và video từ thư viện cục bộ.  
- Tải nội dung từ YouTube và chuyển đổi định dạng.  
- Trích xuất lời bài hát từ tên bài hát và nghệ sĩ.  
- Giao diện người dùng thân thiện và dễ sử dụng.

# III. Các thư viện cần thiết

Ứng dụng sử dụng các thư viện Python như:  
- tkinter: Tạo giao diện người dùng.  
- pytubefix: Tải nội dung từ YouTube.  
- pydub: Xử lý âm thanh và chuyển đổi định dạng.  
- lyricsgenius: Lấy lời bài hát thông qua API Genius.  
- vlc: Phát nhạc và video.

# IV. Cấu trúc chương trình

## 1. Các chức năng chính:

* Quản lý thư viện nhạc và video: Thêm, xóa, và sắp xếp các file nhạc và video từ thư viện cục bộ.
* Phát nhạc và video: Hỗ trợ phát lại nội dung, tua nhanh/chậm, và dừng nội dung.
* Tải nội dung từ YouTube: Hỗ trợ tải video hoặc âm thanh, chuyển đổi MP4 sang MP3.
* Trích xuất lời bài hát: Sử dụng lyricsgenius để lấy lời bài hát từ API.

## Mã Nguồn chương trình:

### Khai báo thư viện

from tkinter import \*

from tkinter import filedialog, simpledialog, messagebox, ttk

from pytubefix import YouTube

from pydub import AudioSegment

import lyricsgenius

import vlc

import os, shutil

### Khởi tạo chương trình

class MusicPlayer:

    def \_\_init\_\_(self, root):

        self.root = root

        self.root.title("mavPlayer")

        screen\_width = root.winfo\_screenwidth()

        screen\_height = root.winfo\_screenheight()

        center\_x = int(screen\_width / 2 - 900 / 2)

        center\_y = int(screen\_height / 2 - 600/ 2)

        root.geometry(f'900x600+{center\_x}+{center\_y}')

        self.root.resizable(False, False)

        self.genius = lyricsgenius.Genius("I1Skzzd5zK9eEuzOojzk-t\_z\_cFb9FhkAykiHQKFTRhpwpj-gcOYRgX23Dz2pWOT7jF9H9Nd-p3gKAnMEMUSOg")

        os.makedirs("music", exist\_ok=True)

        os.makedirs("video", exist\_ok=True)

        self.music\_directory = "music"

        self.video\_directory = "video"

        self.videos = []

        self.songs = []

        self.instance = vlc.Instance()

        self.mediaplayer = self.instance.media\_player\_new()

        self.songplayer = self.instance.media\_player\_new()

        self.song\_title = StringVar()

        self.artist\_name = StringVar()

        self.highlighted\_label = None

        self.create\_widgets()

### Tạo các widget

    def create\_widgets(self):

        self.navbar = Frame(self.root, bg="#FCF7D2", width=200)

        self.navbar.pack(fill=Y, side=LEFT)

        Button(self.navbar, text="♫ Music ♫", width=20, padx=10, pady=10, background='lightblue', command=lambda: self.change\_tab("music"), bd=2, relief=RAISED, fg='black', font=('Consolas', 10, 'bold')).pack(padx=10, pady=10)

        Button(self.navbar, text="▶ Video ▶", width=20, padx=10, pady=10, background='lightblue', command=lambda: self.change\_tab("video"), bd=2, relief=RAISED, fg='black', font=('Consolas', 10, 'bold')).pack(padx=10, pady=0)

        Button(self.navbar, text="𝄙 Lyric 𝄙", width=20, padx=10, pady=10, background='lightblue', command=lambda: self.change\_tab("lyric"), bd=2, relief=RAISED, fg='black', font=('Consolas', 10, 'bold')).pack(padx=10, pady=10)

        self.notebook = ttk.Notebook(self.root, style="TNotebook")

        self.notebook.pack(fill=BOTH, expand=True)

        self.music\_tab = Frame(self.notebook, bg="gray")

        self.video\_tab = Frame(self.notebook, bg="gray")

        self.lyrics\_tab = Frame(self.notebook, bg="gray")

        self.notebook.add(self.music\_tab, text="Music")

        self.notebook.add(self.video\_tab, text="Video")

        self.notebook.add(self.lyrics\_tab, text="Lyric")

        self.notebook.place(relwidth=0.8, relheight=1.1, x=185, y=-23)

        self.setup\_music\_tab()

        self.setup\_lyrics\_tab()

        self.setup\_video\_tab()

        self.notebook.select(self.music\_tab)

### Cài đặt cửa sổ music

    def setup\_music\_tab(self):

        self.config\_frame = Frame(self.music\_tab, bg="gray")

        self.config\_frame.pack(fill=X, pady=5)

        Button(self.config\_frame, text="🗁 Thêm thư mục", command=self.add\_song\_folder, bd=0, width=20, background='white').pack(side=LEFT, padx=10, pady=10)

        Button(self.config\_frame, text="♪ Thêm tệp mp3", command=self.add\_mp3\_file, bd=0, width=20, background='white').pack(side=LEFT, padx=25, pady=10)

        Button(self.config\_frame, text="⬇ Tải từ youtube", command=lambda: self.download\_mp3(), bd=0, width=20, background='white').pack(side=LEFT, padx=(13, 25), pady=10)

        Button(self.config\_frame, text="✘ Xóa bài hát", command=lambda: self.delete\_selected\_file("mp3"), bd=0, width=20, background='white').pack(side=RIGHT, padx=10, pady=10)

        self.song\_frame = Frame(self.music\_tab, bg="white")

        self.song\_frame.pack(fill=BOTH, expand=True, padx=10, pady=0)

        self.song\_canvas = Canvas(self.song\_frame, bg="gray", highlightthickness=0)

        self.song\_canvas.pack(side=LEFT, fill=BOTH, expand=True, padx=2, pady=2)

        self.scrollbar = Scrollbar(self.song\_frame, orient=VERTICAL, command=self.song\_canvas.yview)

        self.song\_canvas.config(yscrollcommand=self.scrollbar.set)

        self.song\_container = Frame(self.song\_canvas, bg="gray")

        self.song\_canvas.create\_window((0, 0), window=self.song\_container, anchor="nw")

        self.song\_container.bind("<Configure>", lambda e: self.song\_canvas.configure(scrollregion=self.song\_canvas.bbox("all")))

        self.song\_canvas.bind\_all("<MouseWheel>", self.mouse\_wheel)

        self.control\_frame = Frame(self.music\_tab, bg="gray")

        self.control\_frame.pack(fill=X, pady=(0, 35))

        Button(self.control\_frame, text="Play", width=20, background='white', bd=0, command=self.play\_song).pack(side=LEFT, padx=10, pady=10)

        Button(self.control\_frame, text="⏮", width=6, background='white', bd=2, command=self.rewind\_song).pack(side=LEFT, padx=(100, 10), pady=10)

        Button(self.control\_frame, text="[▷]", width=6, background='white', bd=2, command=self.pause\_resume\_song).pack(side=LEFT, padx=10, pady=10)

        Button(self.control\_frame, text="⏭", width=6, background='white', bd=2, command=self.fast\_forward\_song).pack(side=LEFT, padx=(10, 100), pady=10)

        Button(self.control\_frame, text="Stop", width=20, background='white', bd=0, command=self.stop\_song).pack(side=RIGHT, padx=10, pady=10)

        self.display\_songs()

### Cài đặt cửa sổ video

    def setup\_video\_tab(self):

        self.config\_frame = Frame(self.video\_tab, bg="gray")

        self.config\_frame.pack(fill=X, pady=5)

        Button(self.config\_frame, text="🗁 Thêm thư mục", command=self.add\_video\_folder, bd=0, width=20, background='white').pack(side=LEFT, padx=10, pady=10)

        Button(self.config\_frame, text="▷Thêm tệp mp4", command=self.add\_mp4\_file, bd=0, width=20, background='white').pack(side=LEFT, padx=25, pady=10)

        Button(self.config\_frame, text="⬇ Tải từ youtube", command=lambda: self.download\_mp4(), bd=0, width=20, background='white').pack(side=LEFT, padx=(13, 25), pady=10)

        Button(self.config\_frame, text="✘ Xóa video", command=lambda: self.delete\_selected\_file("mp4"), bd=0, width=20, background='white').pack(side=RIGHT, padx=10, pady=10)

        self.video\_frame = Frame(self.video\_tab, bg="white")

        self.video\_frame.pack(fill=BOTH, expand=True, padx=10, pady=0)

        self.video\_canvas = Canvas(self.video\_frame, bg="gray", highlightthickness=0)

        self.video\_canvas.pack(side=LEFT, fill=BOTH, expand=True, padx=2, pady=2)

        self.video\_container = Frame(self.video\_canvas, bg="gray")

        self.video\_canvas.create\_window((0, 0), window=self.video\_container, anchor="nw")

        self.control\_frame = Frame(self.video\_tab, bg="gray")

        self.control\_frame.pack(fill=X, pady=(0, 35))

        Button(self.control\_frame, text="Play", width=20, background='white', bd=0, command=self.play\_video).pack(side=LEFT, padx=10, pady=10)

        Button(self.control\_frame, text="⏮", width=6, background='white', bd=2, command=self.rewind\_video).pack(side=LEFT, padx=(100, 10), pady=10)

        Button(self.control\_frame, text="[▷]", width=6, background='white', bd=2, command=self.pause\_resume\_video).pack(side=LEFT, padx=10, pady=10)

        Button(self.control\_frame, text="⏭", width=6, background='white', bd=2, command=self.fast\_forward\_video).pack(side=LEFT, padx=(10, 100), pady=10)

        Button(self.control\_frame, text="Stop", width=20, background='white', bd=0, command=self.stop\_video).pack(side=RIGHT, padx=10, pady=10)

        self.video\_container.bind("<Configure>", lambda e: self.video\_canvas.configure(scrollregion=self.video\_canvas.bbox("all")))

        self.video\_canvas.bind\_all("<MouseWheel>", self.mouse\_wheel)

        self.display\_videos()

### Cài đặt cửa sổ lyric

    def setup\_lyrics\_tab(self):

        self.config\_frame = Frame(self.lyrics\_tab, bg="gray")

        self.config\_frame.pack(fill=X, pady=5)

        Button(self.config\_frame, text="📜 Trích xuất lời bài hát", command=self.fetch\_lyric, bd=0, width=20, background='white').pack(side=LEFT, padx=10, pady=10)

        song\_title\_input = Entry(self.config\_frame, textvariable=self.song\_title, width=25 , background='white')

        song\_title\_input.pack(side=LEFT, padx=10, pady=10)

        self.add\_placeholder(song\_title\_input, "Nhập tên bài hát")

        artist\_name\_input = Entry(self.config\_frame, textvariable=self.artist\_name, width=25, background='white')

        artist\_name\_input.pack(side=LEFT, padx=10, pady=10)

        self.add\_placeholder(artist\_name\_input, "Nhập tên nghệ sĩ")

        Label(self.config\_frame, text="💡Để trống nếu không rõ nghệ sĩ", bg="gray", fg="darkred").pack(side=LEFT, padx=0, pady=10)

        self.lyrics\_frame = Frame(self.lyrics\_tab, bg="white")

        self.lyrics\_frame.pack(fill=BOTH, expand=True, padx=10, pady=(0, 45))

        self.lyrics\_canvas = Canvas(self.lyrics\_frame, bg="gray", highlightthickness=0)

        self.lyrics\_canvas.pack(side=LEFT, fill=BOTH, expand=True, padx=2, pady=2)

        self.lyrics\_container = Frame(self.lyrics\_canvas, bg="gray")

        self.lyrics\_canvas.create\_window((0, 0), window=self.lyrics\_container, anchor="nw")

        self.lyrics\_canvas.bind("<Configure>", lambda e: self.lyrics\_canvas.configure(scrollregion=self.lyrics\_canvas.bbox("all")))

        self.lyrics\_canvas.bind\_all("<MouseWheel>", self.mouse\_wheel)

### Các nút tương tác của Music

    def play\_song(self):

        if self.highlighted\_label:

            song\_name = self.highlighted\_label.cget("text")

            if not song\_name.endswith(".mp3"):

                return

            song\_path = os.path.join(self.music\_directory, song\_name)

            self.media = self.instance.media\_new(song\_path)

            self.songplayer.set\_media(self.media)

            self.songplayer.play()

    def stop\_song(self):

        self.songplayer.stop()

        self.time = 0

    def pause\_resume\_song(self):

        if self.songplayer.is\_playing():

            self.songplayer.pause()

        else:

            self.songplayer.play()

    def rewind\_song(self):

        self.songplayer.set\_time(self.songplayer.get\_time() - 10000)

    def fast\_forward\_song(self):

        self.songplayer.set\_time(self.songplayer.get\_time() + 10000)

### Các nút tương tác của Video

    def play\_video(self):

        if self.highlighted\_label:

            video\_name = self.highlighted\_label.cget("text")

            if not video\_name.endswith(".mp4"):

                return

            video\_path = os.path.join(self.video\_directory, video\_name)

            self.media = self.instance.media\_new(video\_path)

            self.mediaplayer.set\_media(self.media)

            for widget in self.video\_container.winfo\_children():

                widget.destroy()

            self.video\_container.config(background="black")

            self.video\_canvas.config(background="black")

            video\_frame\_id = self.video\_frame.winfo\_id()

            self.mediaplayer.set\_hwnd(video\_frame\_id)

            self.mediaplayer.play()

    def stop\_video(self):

        self.mediaplayer.stop()

        self.video\_container.config(background="gray")

        self.video\_canvas.config(background="gray")

        self.display\_videos()

    def pause\_resume\_video(self):

        if self.mediaplayer.is\_playing():

            self.mediaplayer.pause()

        else:

            self.mediaplayer.play()

    def rewind\_video(self):

        self.mediaplayer.set\_time(self.mediaplayer.get\_time() - 10000)

    def fast\_forward\_video(self):

        self.mediaplayer.set\_time(self.mediaplayer.get\_time() + 10000)

### Chuyển đổi cửa sổ

    def change\_tab(self, tab):

        if tab == "music":

            self.notebook.select(self.music\_tab)

        elif tab == "video":

            self.notebook.select(self.video\_tab)

        elif tab == "lyric":

            self.notebook.select(self.lyrics\_tab)

### Thêm thư mục nhạc

    def add\_song\_folder(self):

        directory = filedialog.askdirectory()

        if directory:

            for file in os.listdir(directory):

                if file.endswith(".mp3"):

                    source\_path = os.path.join(directory, file)

                    destination\_path = os.path.join(self.music\_directory, file)

                    if not os.path.exists(destination\_path):

                        shutil.copy(source\_path, destination\_path)

            self.display\_songs()

### Thêm tệp nhạc

    def add\_mp3\_file(self):

        files = filedialog.askopenfilenames(filetypes=[("MP3 files", "\*.mp3")])

        if files:

            for file in files:

                file\_name = os.path.basename(file)

                destination\_path = os.path.join(self.music\_directory, file\_name)

                if not os.path.exists(destination\_path):

                    shutil.copy(file, destination\_path)

            self.display\_songs()

### Thêm thư mục video

    def add\_video\_folder(self):

        directory = filedialog.askdirectory()

        if directory:

            for file in os.listdir(directory):

                if file.endswith(".mp4"):

                    source\_path = os.path.join(directory, file)

                    destination\_path = os.path.join(self.video\_directory, file)

                    if not os.path.exists(destination\_path):

                        shutil.copy(source\_path, destination\_path)

            self.display\_videos()

### Thêm tệp video

    def add\_mp4\_file(self):

        files = filedialog.askopenfilenames(filetypes=[("MP4 files", "\*.mp4")])

        if files:

            for file in files:

                file\_name = os.path.basename(file)

                destination\_path = os.path.join(self.video\_directory, file\_name)

                if not os.path.exists(destination\_path):

                    shutil.copy(file, destination\_path)

            self.display\_videos()

### Tải nhạc

    def download\_mp3(self):

        link = simpledialog.askstring("Link", "Nhập link của bài hát bạn muốn tải:")

        if link:

            try:

                video = YouTube(link)

                stream = video.streams.filter(file\_extension='mp4', progressive=True).order\_by('resolution').desc().first()

                downloaded\_video\_path = stream.download(self.video\_directory)

                mp3\_file\_path = os.path.join(self.music\_directory, os.path.splitext(os.path.basename(downloaded\_video\_path))[0] + ".mp3")

                AudioSegment.from\_file(downloaded\_video\_path).export(mp3\_file\_path, format="mp3")

                messagebox.showinfo("MP3", "Tải thành công.")

                self.display\_songs()

                self.display\_videos()

            except Exception as e:

                messagebox.showerror("Lỗi", e)

### Tải video

    def download\_mp4(self):

        link = simpledialog.askstring("Link", "Nhập link của video bạn muốn tải:")

        if link:

            try:

                video = YouTube(link)

                stream = video.streams.filter(file\_extension='mp4', progressive=True).order\_by('resolution').desc().first()

                downloaded\_video\_path = stream.download(self.video\_directory)

                messagebox.showinfo("MP4", "Tải thành công.")

                self.display\_videos()

            except Exception as e:

                messagebox.showerror("Lỗi", e)

### Xóa nhạc, video

    def delete\_selected\_file(self, file):

        try:

            if file == "mp3":

                if self.highlighted\_label:

                    song\_name = self.highlighted\_label.cget("text")

                    song\_path = os.path.join(self.music\_directory, song\_name)

                    os.remove(song\_path)

                    self.songs.remove(song\_name)

                    self.highlighted\_label.destroy()

                else:

                    raise Exception("Hãy chọn item cần xóa.")

            elif file == "mp4":

                if self.highlighted\_label:

                    video\_name = self.highlighted\_label.cget("text")

                    video\_path = os.path.join(self.video\_directory, video\_name)

                    os.remove(video\_path)

                    self.videos.remove(video\_name)

                    self.highlighted\_label.destroy()

                else:

                    raise Exception("Hãy chọn item cần xóa.")

        except Exception as e:

            messagebox.showerror("Lỗi", "Hãy chọn item cần xóa.")

### Trích xuất lời bài hát

    def fetch\_lyric(self):

        song\_title = self.song\_title.get()

        artist\_name = self.artist\_name.get()

        if artist\_name == "Nhập tên nghệ sĩ":

            artist\_name = ""

        if not song\_title or song\_title == "Nhập tên bài hát":

            messagebox.showerror("Lỗi", "Hãy nhập tên bài hát.")

            return

        try:

            song = self.genius.search\_song(song\_title, artist\_name)

        except Exception as e:

            messagebox.showerror("Lỗi", e)

            return

        if song:

            lyrics = song.lyrics

            self.display\_lyrics(lyrics)

        else:

            messagebox.showerror("Lỗi", "Không tim thấy lời bài hát.")

### Chuyển đổi mp4 sang mp3

    def convert\_mp4\_to\_mp3(self, mp4\_path):

        audio = AudioSegment.from\_file(mp4\_path, format="mp4")

        mp3\_path = mp4\_path.replace(".mp4", ".mp3")

        audio.export(mp3\_path, format="mp3")

        return mp3\_path

### Hiển thị nhạc

    def display\_songs(self):

        for widget in self.song\_container.winfo\_children():

            widget.destroy()

        for file in os.listdir(self.music\_directory):

            if file.endswith(".mp3") and file not in self.songs:

                self.songs.append(file)

        for song\_name in self.songs:

            song\_label = Label(self.song\_container, text=song\_name, bg="#E0A469", width=94, anchor="w", padx=10, pady=5)

            song\_label.pack(fill=X, padx=5, pady=5)

            song\_label.bind("<Button-1>", lambda e, lbl=song\_label: self.highlight\_label(lbl))

        self.song\_canvas.update\_idletasks()

        self.song\_canvas.config(scrollregion=self.song\_canvas.bbox("all"))

### Hiển thị video

    def display\_videos(self):

        for widget in self.video\_container.winfo\_children():

            widget.destroy()

        for file in os.listdir(self.video\_directory):

            if file.endswith(".mp4") and file not in self.videos:

                self.videos.append(file)

        for video\_name in self.videos:

            video\_label = Label(self.video\_container, text=video\_name, bg="#E0A469", width=94, anchor="w", padx=10, pady=5)

            video\_label.pack(fill=X, padx=5, pady=5)

            video\_label.bind("<Button-1>", lambda e, lbl=video\_label: self.highlight\_label(lbl))

        self.video\_canvas.update\_idletasks()

        self.video\_canvas.config(scrollregion=self.video\_canvas.bbox("all"))

### Hiển thị lyric

    def display\_lyrics(self, lyrics):

        for widget in self.lyrics\_container.winfo\_children():

            widget.destroy()

        lines = lyrics.split("\n")

        for line in lines:

            lyric\_label = Label(self.lyrics\_container, text=line, bg="gray", width=94, anchor="w", padx=10, pady=0)

            lyric\_label.pack(fill=X, padx=5, pady=5)

        self.lyrics\_canvas.update\_idletasks()

        self.lyrics\_canvas.config(scrollregion=self.lyrics\_canvas.bbox("all"))

### Chọn nổi bật

    def highlight\_label(self, label):

        if self.highlighted\_label is not None and self.highlighted\_label.winfo\_exists():

            self.highlighted\_label.config(bg="#E0A469")

        label.config(bg="#AAE54A")

        self.highlighted\_label = label

### Cuộn chuột

    def mouse\_wheel(self, event):

        self.song\_canvas.yview\_scroll(-1 \* (event.delta // 120), "units")

        self.video\_canvas.yview\_scroll(-1 \* (event.delta // 120), "units")

        self.lyrics\_canvas.yview\_scroll(-1 \* (event.delta // 120), "units")

        if self.song\_canvas.yview()[0] <= 0:

            self.song\_canvas.yview\_moveto(0)

        if self.video\_canvas.yview()[0] <= 0:

            self.video\_canvas.yview\_moveto(0)

### Thêm placeholder

    def add\_placeholder(self, entry, placeholder):

        entry.insert(0, placeholder)

        entry.config(fg='grey')

        def on\_focus\_in(event):

            if entry.get() == placeholder:

                entry.delete(0, END)

                entry.config(fg='black')

        def on\_focus\_out(event):

            if not entry.get():

                entry.insert(0, placeholder)

                entry.config(fg='grey')

        entry.bind("<FocusIn>", on\_focus\_in)

        entry.bind("<FocusOut>", on\_focus\_out)

### Thực thi chương trình

if \_\_name\_\_ == "\_\_main\_\_":

    root = Tk()

    app = MusicPlayer(root)

    root.mainloop()

# V. Các thay đổi và cải tiến so với bài mẫu

Chương trình cải tiến dựa trên các bài mẫu từ GeeksforGeeks, với các điểm mới như:  
- Tích hợp nhiều tính năng trong cùng một giao diện.

- Tính năng phát nhạc và video.  
- Quản lý thư viện nhạc và video cục bộ.  
- Tối ưu hóa giao diện người dùng và xử lý lỗi.

# VI. Hướng dẫn sử dụng

1. Cài đặt các thư viện yêu cầu bằng lệnh pip:  
 pip install tkinter pytubefix pydub lyricsgenius python-vlc  
2. Chạy file chính để mở giao diện.  
3. Sử dụng các nút chức năng để thêm, phát, tải hoặc trích xuất nội dung.

**Mã nguồn:** <https://github.com/tuan-cre/LHT-PLAYER>

# VII. Tiềm năng phát triển

Phần mềm có thể mở rộng với các tính năng như:  
- Các chức năng quản lý như tìm kiếm, sắp xếp.

- Tính năng tạo playlist.  
- Hỗ trợ thêm nhiều định dạng file.

- Phát lyric song song với bài hát.  
- Kết nối với các ứng dụng bên thứ ba như Spotify.

# VIII. Tài liệu tham khảo

- [Create First GUI Application Using Python Tkinter](https://www.geeksforgeeks.org/create-first-gui-application-using-python-tkinter/)

- [Create a GUI to Extract Lyrics from Song Using Python](https://www.geeksforgeeks.org/create-a-gui-to-extract-lyrics-from-song-using-python/)

- [Create GUI for Downloading YouTube Video Using Python](https://www.geeksforgeeks.org/create-gui-for-downloading-youtube-video-using-python/)