LYRICS EXTRACTOR

**Overview**

This task involves creating a GUI application using Tkinter that allows users to enter a song title and artist name, then fetches and displays the lyrics.

**Prerequisites**

* **Python** installed on your system.
* **Tkinter** for GUI development (comes with most Python installations).
* **lyricsgenius** library to interact with the Genius API.
* A Genius API token (sign up at Genius to get one).

**Installation**

Install the lyricsgenius library using pip:

pip install lyricsgenius

**Implementation Steps**

1. **Import Required Libraries:**

import tkinter as tk

from tkinter import messagebox

import lyricsgenius

1. **Initialize the Genius API:** Replace "YOUR\_GENIUS\_API\_TOKEN" with your actual Genius API token.

genius = lyricsgenius.Genius("YOUR\_GENIUS\_API\_TOKEN")

1. **Define the Function to Fetch Lyrics:** This function retrieves the song lyrics based on user input.

def fetch\_lyrics():

song\_title = song\_entry.get()

artist\_name = artist\_entry.get()

try:

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

if song:

lyrics\_text.delete(1.0, tk.END)

lyrics\_text.insert(tk.END, song.lyrics)

else:

messagebox.showinfo("Not Found", "Lyrics not found for the given song.")

except Exception as e:

messagebox.showerror("Error", str(e))

1. **Set Up the GUI:** Create the main application window and add labels, entry fields, and buttons.

root = tk.Tk()

root.title("Lyrics Extractor")

# Song Title

tk.Label(root, text="Song Title").grid(row=0, column=0, padx=10, pady=10)

song\_entry = tk.Entry(root, width=40)

song\_entry.grid(row=0, column=1, padx=10, pady=10)

# Artist Name

tk.Label(root, text="Artist Name").grid(row=1, column=0, padx=10, pady=10)

artist\_entry = tk.Entry(root, width=40)

artist\_entry.grid(row=1, column=1, padx=10, pady=10)

# Fetch Lyrics Button

fetch\_button = tk.Button(root, text="Fetch Lyrics", command=fetch\_lyrics)

fetch\_button.grid(row=2, column=0, columnspan=2, pady=10)

# Text Box to Display Lyrics

lyrics\_text = tk.Text(root, wrap='word', width=60, height=20)

lyrics\_text.grid(row=3, column=0, columnspan=2, padx=10, pady=10)

1. **Run the Main Event Loop:** This starts the GUI application.

root.mainloop()

**Complete Code**

import tkinter as tk

from tkinter import messagebox

import lyricsgenius

# Initialize Genius API with your token

genius = lyricsgenius.Genius("YOUR\_GENIUS\_API\_TOKEN")

def fetch\_lyrics():

song\_title = song\_entry.get()

artist\_name = artist\_entry.get()

try:

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

if song:

lyrics\_text.delete(1.0, tk.END)

lyrics\_text.insert(tk.END, song.lyrics)

else:

messagebox.showinfo("Not Found", "Lyrics not found for the given song.")

except Exception as e:

messagebox.showerror("Error", str(e))

# Setting up the GUI

root = tk.Tk()

root.title("Lyrics Extractor")

# Song Title

tk.Label(root, text="Song Title").grid(row=0, column=0, padx=10, pady=10)

song\_entry = tk.Entry(root, width=40)

song\_entry.grid(row=0, column=1, padx=10, pady=10)

# Artist Name

tk.Label(root, text="Artist Name").grid(row=1, column=0, padx=10, pady=10)

artist\_entry = tk.Entry(root, width=40)

artist\_entry.grid(row=1, column=1, padx=10, pady=10)

# Fetch Lyrics Button

fetch\_button = tk.Button(root, text="Fetch Lyrics", command=fetch\_lyrics)

fetch\_button.grid(row=2, column=0, columnspan=2, pady=10)

# Text Box to Display Lyrics

lyrics\_text = tk.Text(root, wrap='word', width=60, height=20)

lyrics\_text.grid(row=3, column=0, columnspan=2, padx=10, pady=10)

# Start the GUI event loop

root.mainloop()

**Explanation**

* **Labels and Entry Fields:** Allow the user to input the song title and artist name.
* **Fetch Lyrics Button:** Triggers the fetch\_lyrics function to retrieve lyrics.
* **Text Box:** Displays the lyrics of the song.
* **Error Handling:** Provides user-friendly error messages if the song or artist is not found, or if there is another issue.

**Conclusion**

This task provides a hands-on experience in GUI development with Python using Tkinter and the Genius API. You can further enhance this application by adding features like saving lyrics, selecting different languages, or improving error handling.