Task 3: Creating GUI to Extract Lyrics from Songs using Python

Problem Statement: Your task is to design a graphical user interface (GUI) application that can extract lyrics from songs using python. You will integrate APIs or web scraping techniques to fetch lyrics and display them on the interface.

- **Features**

User friendly Interface: Simple GUI for entering the artist's name and song title.
Lyrics Display: Displays the fetched lyrics in a scrollable text area.
Clear Fields: Option to clear the input fields and lyrics display area.

- **Prerequisites**

Before running the application, ensure that the following Python libraries are installed:
`lyricsgenius`: For interacting with the Genius API.

```
pip install lyricsgenius
```

`tkinter`: For creating the GUI. (Preinstalled with Python)

Code Explanation

- **Importing Required Libraries**

```
import tkinter as tk
from tkinter import messagebox
import lyricsgenius
```

tkinter: Used to create the GUI.
messagebox: For displaying message boxes (warnings, errors, etc.).
lyricsgenius: A Python wrapper for the Genius API to fetch song lyrics.

- **Functions**

```
get_lyrics():
   Fetches lyrics based on the artist's name and song title.
   Handles cases where no input is provided or lyrics are not found.
   Displays the lyrics in the text area if found.
  def get_lyrics():
     artist = artist_entry.get()
     song = song_entry.get()

     if not artist or not song:
        messagebox.showwarning("Input Error", "Please enter both artist and song name.")
        return

     try:
        song_lyrics = genius.search_song(song, artist)
        if song_lyrics:
           lyrics_text.delete(1.0, tk.END)
           lyrics_text.insert(tk.END, song_lyrics.lyrics)
        else:
           messagebox.showinfo("No Lyrics Found", "Could not find lyrics for the song.")
     except Exception as e:
        messagebox.showerror("Error", f"An error occurred: {e}")
```

clear_fields():
Clears the artist and song input fields, as well as the lyrics display area.

```
def clear_fields():
    artist_entry.delete(0, tk.END)
    song_entry.delete(0, tk.END)
    lyrics_text.delete(1.0, tk.END)
```

- **GUI Setup**

The main window is set up with labels, entry fields, buttons, and a text box for displaying lyrics.

```
# Initialize the main window
root = tk.Tk()
root.title("Lyrics Extractor")
root.geometry("600x400")

 # Artist label and entry
artist_label = tk.Label(root, text="Artist:")
artist_label.pack(pady=10)
artist_entry = tk.Entry(root, width=50)
artist_entry.pack(pady=5)

# Song label and entry
song_label = tk.Label(root, text="Song:")
song_label.pack(pady=10)
song_entry = tk.Entry(root, width=50)
song_entry.pack(pady=5)

# Fetch Lyrics Button
fetch_button = tk.Button(root, text="Get Lyrics", command=get_lyrics)
fetch_button.pack(pady=20)

# Clear Fields Button
clear_button = tk.Button(root, text="Clear", command=clear_fields)
clear_button.pack(pady=10)

# Text box to display the lyrics
lyrics_text = tk.Text(root, wrap=tk.WORD, height=10, width=60)
lyrics_text.pack(pady=10)

# Start the application
root.mainloop()
```

root = tk.Tk(): Initializes the main application window.
Entry widgets: Used to take user input for the artist's name and song title.
Buttons:
"Get Lyrics": Triggers the `get_lyrics()` function.
"Clear": Triggers the `clear_fields()` function.
Text widget: Displays the fetched lyrics.

- **Running the Application**

Run the script to start the application.
Enter the artist's name and song title in the provided fields.
Click "Get Lyrics" to fetch and display the song lyrics.
Use the "Clear" button to reset the input fields and text area.

Error Handling

The application includes basic error handling for:
Missing artist or song input.
Lyrics not found for the given song.
APIrelated issues.

Future Enhancements

Save Lyrics: Add a feature to save the fetched lyrics to a file.
Improved UI: Enhance the user interface with more styling and layout adjustments.
Additional API Integrations: Support other lyrics APIs or web scraping methods to increase coverage.

Conclusion

This application is a simple yet effective way to extract and display song lyrics using Python. It serves as a foundation for more advanced features and can be easily customized to fit specific needs.