Probability Software Assignment

Name -:Saksham
Roll no -: AI22BTECH11024

1 Introduction

The goal of this project was to create a Python program that plays audio files from a specified folder in a random order. The program utilizes the `pygame` library for audio playback and the `numpy` library for shuffling the playlist. The project created is a Python program that allows you to create and play a playlist of songs. It utilizes the `pygame` library for audio playback, `numpy` for generating random numbers, and `threading` for multi-threading support.

2 Implementation

The program is implemented in Python and consists of the following key components:

- File Selection: The user provides the path to the folder containing the audio files.
- Randomization: The program lists all the audio files in the folder and shuffles them randomly.
- Audio Playback: The `pygame` library is used to load and play the audio files.

3 Usage

To use the program, follow these steps:

1) Run the program in a Python environment (Python 3 or above).
2) Provide the path to the folder containing the audio files.
3) The program will play the audio files in a random order each time it is run.

4 Dependencies

The program relies on the following external libraries:

- `pygame`: Used for audio playback and volume control.
- `numpy`: Used for shuffling the playlist.
- `threading`: Used for multi-threading support.

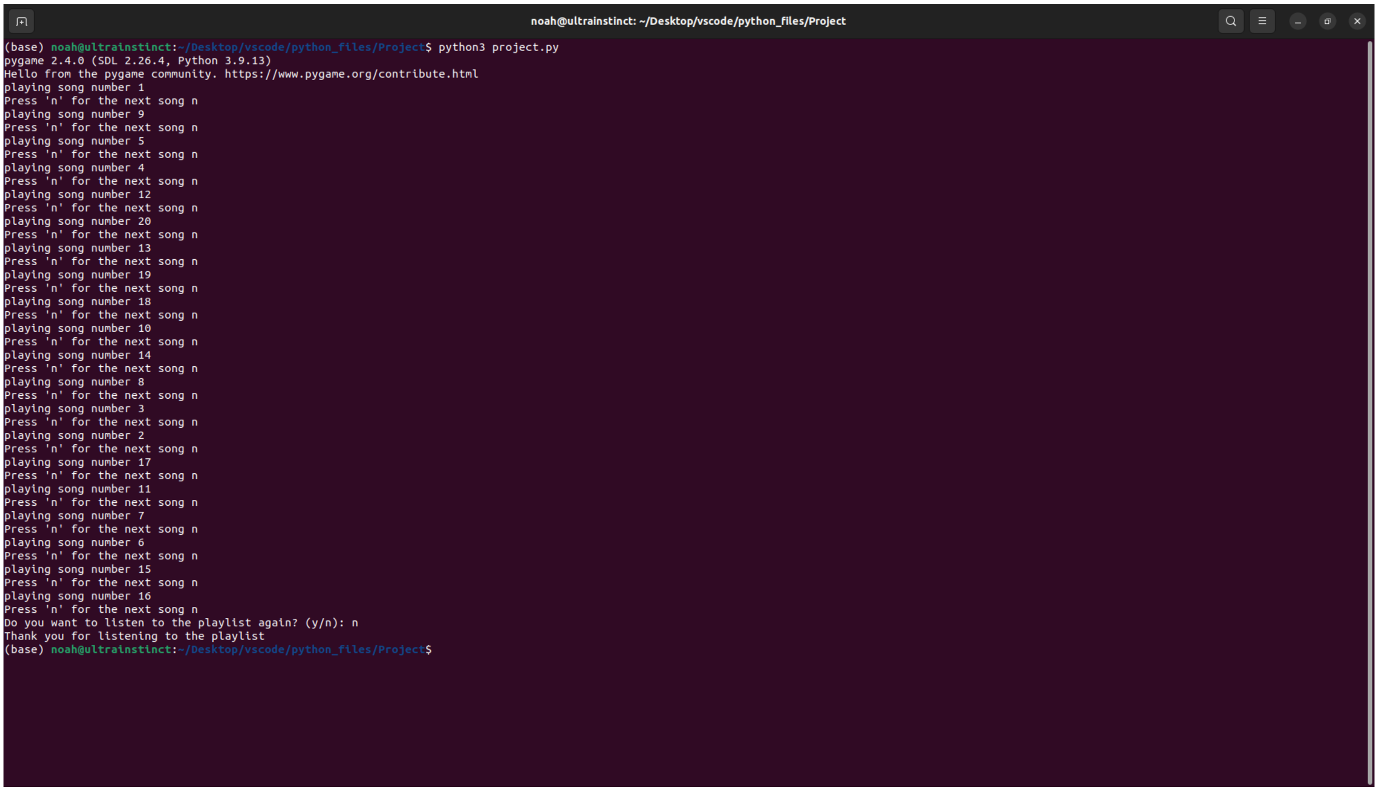
Ensure that these libraries are installed in the Python environment before running the program.

5 Conclusion

In conclusion, the project allows users to create and play a playlist of songs using Python. It utilizes the `pygame` library for audio playback, `numpy` for generating random numbers, and threading for multi-threading support. The project creates a playlist by randomly selecting songs from a folder and plays them one by one. It provides options for the user to skip a song or stop the playlist altogether. The program runs in a loop, allowing the user to listen to the playlist again if desired. By employing multi-threading, each song is played in a separate thread, enabling the next song to start automatically after the previous one finishes. This ensures a seamless listening experience without interruptions. The project demonstrates the usage of key Python modules and concepts, including file handling, threading, user input, and audio playback. It provides a basic framework that can be expanded and customized further to incorporate additional features, such as song metadata, duration display, or user interfaces. Overall, this project serves as a starting point for building a simple and interactive playlist application, allowing users to enjoy their favorite songs in a randomized order.

Output

Random numbers are generated on the display.



```
(base) noah@ultrainstinct:~/Desktop/vscode/python_files/Project$ python3 project.py
pygame 2.4.0 (SDL 2.26.4, Python 3.9.13)
Hello from the pygame community. https://www.pygame.org/contribute.html
playing song number 1
Press 'n' for the next song n
playing song number 9
Press 'n' for the next song n
playing song number 5
Press 'n' for the next song n
playing song number 4
Press 'n' for the next song n
playing song number 12
Press 'n' for the next song n
playing song number 20
Press 'n' for the next song n
playing song number 13
Press 'n' for the next song n
playing song number 19
Press 'n' for the next song n
playing song number 18
Press 'n' for the next song n
playing song number 10
Press 'n' for the next song n
playing song number 14
Press 'n' for the next song n
playing song number 8
Press 'n' for the next song n
playing song number 3
Press 'n' for the next song n
playing song number 2
Press 'n' for the next song n
playing song number 17
Press 'n' for the next song n
playing song number 11
Press 'n' for the next song n
playing song number 7
Press 'n' for the next song n
playing song number 6
Press 'n' for the next song n
playing song number 15
Press 'n' for the next song n
playing song number 16
Press 'n' for the next song n
Do you want to listen to the playlist again? (y/n): n
Thank you for listening to the playlist
(base) noah@ultrainstinct:~/Desktop/vscode/python_files/Project$
```

Fig. 1. output