# Probability Software Assignment

Name -: Gunnam Sri Satya Koushik
Roll no -: CS22BTECH11026

*Abstract*—**In this assignment we made a Music player in python which uses numpy module of python to shuffle the songs.**

## Introduction:

The provided program is a Music Player application developed using the Tkinter library for the GUI and the Pygame library for handling music playback. It offers basic functionality such as playing, pausing, resuming, skipping to the next track, and exiting the program. The proopgram also incorporates a shuffle feature that plays the music tracks in a randomized order.

## Program Structure

1) **Importing Libraries:**
   The program starts by importing the required libraries: tkinter, os, numpy, and pygame.mixer.
2) **TInitializing Pygame:**
   Pygame mixer is initialized to handle music playback.
3) **Creating the GUI Window:**
   a) A Tkinter window is created with dimensions 600x400 pixels.
   b) The window is titled ”Music Player” and has a light blue background.
4) **Setting up the Music Directory:**
   a) The path variable is set to the directory containing the music files.
   b) The dirlist variable is populated with the list of files in the music directory.
   c) The checklist variable is also initialized with the same list of files.
5) **Checking Music Playback Status:**
   a) The checkend() function is defined to check if the music has finished playing or paused.
   b) The function checks if the music is not currently playing and not paused.
   c) If these conditions are met, it calls the playnext() function to play the next track.
   d) The checkend() function is called periodically using the window.after() method.
6) **Playing the First Track and Shuffle Functionality:**
   a) A random track is selected from the dirlist using numpy.random.choice().
   b) The selected track is removed from the checklist to avoid repetition.
   c) The track is loaded and played using mixer.music.load() and mixer.music.play().
7) **Defining Control Functions:**
   a) pause(): Pauses the music playback and sets the paused flag to True.
   b) resume(): Resumes the music playback if it was paused and sets the paused flag to False.
   c) exit(): Exits the program.
8) **Creating GUI Elements:**
   a) A label is created to display the current track's name.
   b) A frame is created to hold the control buttons (Pause, Resume, Next, Exit).
   c) The control buttons are created and associated with their respective functions.
9) **Playing the Next Track:**
   a) playnext() function is defined to play the next random track.
   b) It checks if there are any tracks remaining in the checklist.
   c) If the list is empty, the program exits. Otherwise:
   a) Another random track is selected and removed from the checklist.
   b) The label is updated with the name of the current track.
   c) The new track is loaded and played using mixer.music.load() and mixer.music.play().
10) **Running the Application:**
    The main event loop is started using window.mainloop(), which keeps the GUI responsive.

## Conclusion

The Music Player program provides a basic graphical interface for playing music files from a specified directory. It utilizes the Tkinter library to create the GUI window and Pygame to handle music playback. The program allows users to pause, resume, skip to the next track, and exit the application. With the addition of shuffle functionality, the program plays the music tracks in a randomized order, enhancing the listening experience and preventing monotony. The Music Player program offers a simple yet enjoyable way to listen to music on a computer.

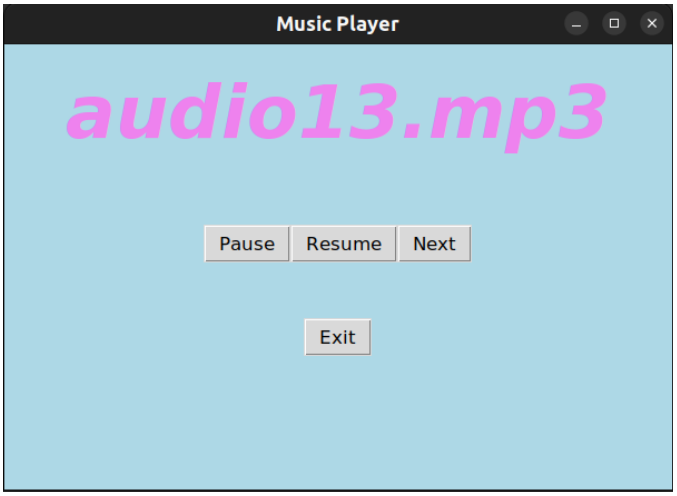

Fig. 1. Music Player