Play !t

Mini Project Report

Submitted in partial fulfillment of the requirements

For the degree of

Bachelor of Engineering (Computer Engineering)

by:

Sahil Rane. [C-21]	Student ID -TU3F2122153
Harsh Minde. [C-32]	Student ID –TU3F2122164
Rushikesh Jadhav.[C-35]	Student ID –TU3F2122167

Under the Guidance of

PROF. KIRTI SURYAWANSHI

Department of Computer Engineering

TERNA ENGINEERING COLLEGE

Nerul (W), Navi Mumbai 400706

(University of Mumbai)

(2022-2023)

Internal Approval Sheet

TERNA ENGINEERING COLLEGE, NERUL

Department of Computer Engineering

Academic Year 2022-2023

CERTIFICATE

This is to certify that project entitled “Play !t” is a bonafide work

of:

Sahil Rane. [C-21]	Student ID -TU3F2122153
Harsh Minde. [C-32]	Student ID –TU3F2122164
Rushikesh Jadhav.[C-35]	Student ID –TU3F2122167

Submitted to the University of Mumbai in partial fulfilment of the requirement for the award of the Bachelor of Engineering (Computer Engineering).

Guide **Head of Department** **Principal**

Table of Contents

Chapter 1

Introduction

The "Music Player: Play !t" is developed. The project is a web application that serves as a music player, allowing users to enjoy their favourite songs without any distractions such as notifications, ads, internet requirement, or noise. The project is built using Python programming language, Visual Studio, Pygame, and Tkinter for GUI development. The main purpose of this project is to provide a practical learning experience in software development and gain proficiency in using Python and GUI libraries.

The "Music Player: Play !t" project aims to provide a simple and convenient solution for music enthusiasts who seek a hassle-free listening experience. The application allows users to create a playlist of their favourite songs, play, pause, stop, and manage the playlist. The GUI interface created using Tkinter library provides a visually appealing and user-friendly interface for seamless navigation and control of the music player functionalities. The project is developed as a standalone web application, eliminating the need for an internet connection or external music player applications.

The report will cover various aspects of the "Music Player: Play !t" project, including the software and hardware requirements, flowchart depicting the program's process, functions used in the program, output primitives and transformations utilized for GUI development, the actual program code, and output snapshots. The report will also include a conclusion summarizing the project's achievements and a reference section citing the sources used for the project. The "Music Player: Play !t" project is expected to provide a comprehensive and practical understanding of software development using Python and GUI libraries, and serve as a valuable learning experience for the student.

User Interface:

The user interface of the "Music Player: Play !t" project is designed to provide a seamless and intuitive experience for the users. The graphical user interface (GUI) is developed using Tkinter, a popular Python library for creating desktop applications with interactive user interfaces. The GUI interface of the music player application features a clean and modern design, with easy-to-navigate menus, buttons, and playlists.

The main window of the music player displays the current song being played, along with controls for play, pause, stop, and volume adjustment. Users can create and manage playlists, add songs to the playlist, and switch between different songs in the playlist. The interface also provides feedback on the status of the music player, such as indicating whether a song is currently playing, paused, or stopped.

The GUI of the "Music Player: Play !t" project is designed to be visually appealing, responsive, and user-friendly, allowing users to easily interact with the music player functionalities without any confusion. The intuitive layout and controls of the GUI make it easy for users to navigate through the application and enjoy their favorite songs hassle-free.

The user interface is a crucial aspect of the "Music Player: Play !t" project, as it plays a significant role in enhancing the overall user experience. The GUI is designed to be responsive and efficient, providing a smooth and enjoyable interaction for users while listening to their favourite music. The report will provide further details on the development of the user interface and its functionalities in the "Music Player: Play !t" project.

Chapter 2

Software And Hardware Requirements

The "Music Player: Play !t" project requires the following software and hardware components:

Software Requirements:

1. OS -Windows XP, Vista, 7, 8 or 10.
2. Python programming Interpreter
3. Visual Studio Code or any other Python IDE
4. Pygame library for audio playback
5. Tkinter library for GUI development

Hardware Requirements:

1. Computer with minimum system requirements to run Python & Pygame
2. Memory -4GB RAM for 64 bit.

Chapter 3

Flowchart

Process of the program in detail:

The flowchart of the "Music Player: Play !t" project outlines the process of the program in detail, from the start to the end. It provides a visual representation of how the different components of the project interact with each other and the steps involved in the functioning of the music player application.

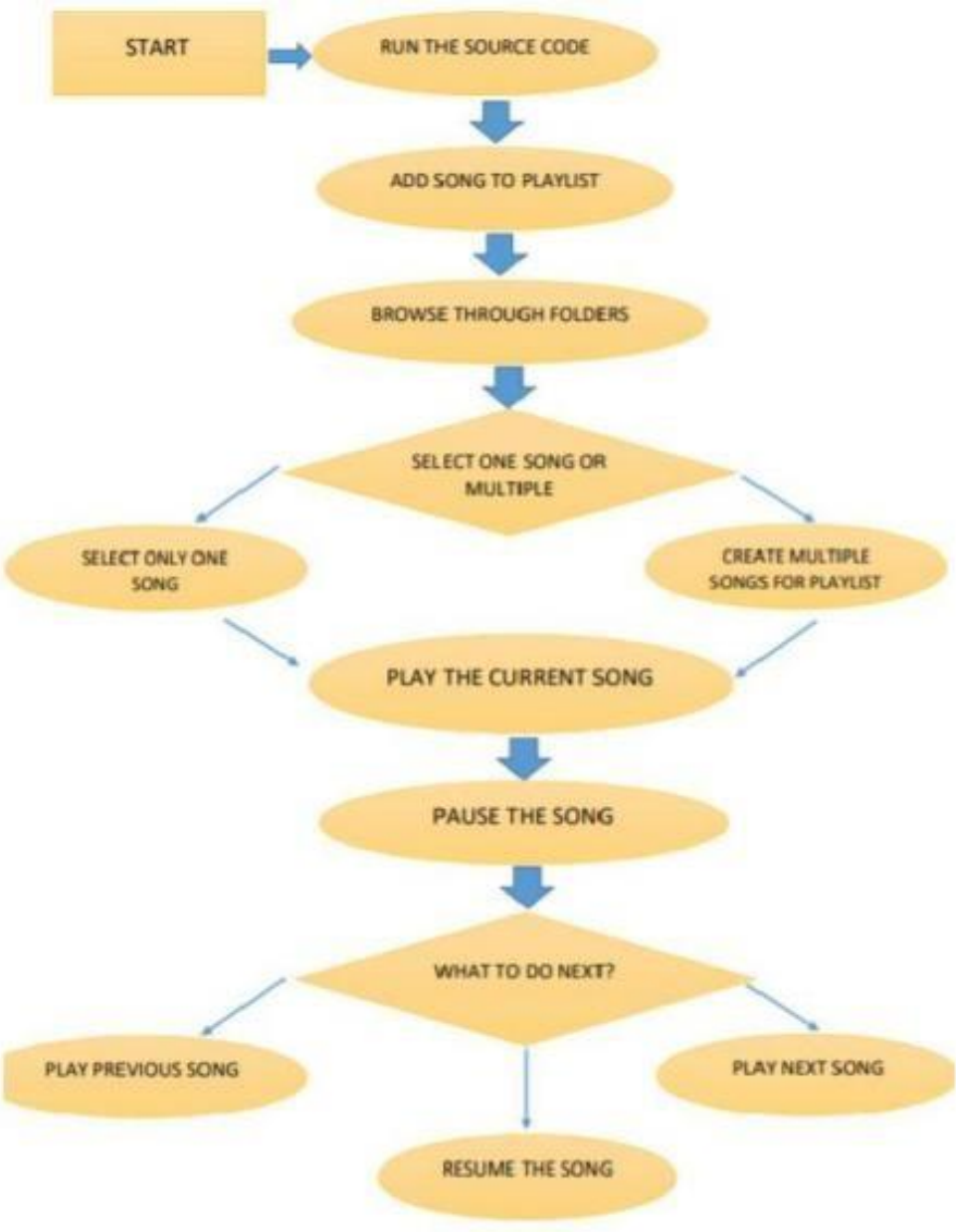

Figure 3.1

Chapter 4

Functions Used

The "Music Player: Play !t" project utilizes various functions to implement different functionalities of the music player application. Some of the key functions used in the project include:

Function	Description
Play()	This function is used to play the selected song from the playlist.
Pause()	This function is used to pause the currently playing song.
Stop()	This function is used to stop the currently playing song.
Next()	This function is used to play the next song in the playlist.
Previous()	This function is used to play the previous song in the playlist.
Add_to_playlist()	1. This function is used to add songs to the playlist.
Remove_from_playlist()	This function is used to remove songs from the playlist.

Tanle 4.1

Chapter 5

Output Primitives And Transformations

The "Music Player: Play !t" project utilizes output primitives and transformations to create a visually appealing and user-friendly graphical user interface (GUI) for the music player application.

Output Primitives:

Output primitives are basic graphical elements that are used to create various graphical elements in the GUI for the music player application. In this project, filled area primitives are used to create buttons, labels, and other graphical elements. The flood-fill approach is used to give color to different components of the GUI, such as buttons, labels, and backgrounds, to enhance the visual aesthetics of the application.

3D Transformations:

3D transformations are used to create a three-dimensional effect on the GUI elements, adding depth and realism to the graphics. In the "Music Player: Play !t" project, 3D transformations are used to create a sense of depth and dimension to the graphical elements of the application. For example, translation is used to move the graphical elements on the screen, giving a sense of motion and interactivity to the GUI.

The combination of output primitives and 3D transformations allows the "Music Player: Play !t" project to create an engaging and visually appealing GUI for the music player application, providing users with an enjoyable and interactive experience while using the application.

The MP3 player is a device for playing and listening to digital audio files, which can be MP3 files or other audio files. The player was created in Python language. A GUI implementation of the application has been developed that is simple and easy to use. The application gives the user five options: add a song to a playlist, play the song, pause or resume the song, play the previous song, and play the next song. The player can also add multiple tracks to the playlist at the same time. It has a large display area in which the playlist is visible. Once a track has been selected and played we can listen to it and view details, the song is at the top of the screen. This information includes details about the song, such as: B. the name of the song, the name of the singer, the length of the song, the file size, etc.

Chapter 6

Program

The "Music Player: Play !t" project includes a Python program that implements the functionalities of the music player application. The program takes user input for selecting songs, playing, pausing, stopping, and managing the playlist. It also incorporates the GUI elements using the Tkinter library for creating a user-friendly interface.

Input Code:

```
import os
import time
from tkinter import *
from tkinter import filedialog
from pygame import mixer

root = Tk()
root.title("Simpli Music Player")
root.geometry("485x700+290+10")
root.configure(background='#333333')
root.resizable(False, False)
mixer.init()

# Create a function to open a file
def AddMusic():
    path = filedialog.askdirectory()
    if path:
      os.chdir(path)
      songs = os.listdir(path)

      for song in songs:
            if song.endswith(".mp3"):
                  Playlist.insert(END, song)


def PlayMusic():
    Music_Name = Playlist.get(ACTIVE)
    print(Music_Name[0:-4])
    mixer.music.load(Playlist.get(ACTIVE))
    mixer.music.play()


# icon
lower_frame = Frame(root , bg = "#FFFFFF", width = 485 , height = 180 )
lower_frame.place ( x = 0 , y = 400)
```

```
image_icon = PhotoImage(file="logo png.png")
root.iconphoto(False, image_icon)


frameCnt = 30
frames = [PhotoImage(file='aa1.gif',format = 'gif -index %i' %(i)) for i in range(frameCnt)]

def update(ind):

    frame = frames[ind]
    ind += 1
    if ind == frameCnt:
        ind = 0
    label.configure(image=frame)
    root.after(40, update, ind)
label = Label(root)
label.place(x=0, y=0)
root.after(0, update, 0)




# Button
ButtonPlay = PhotoImage(file="play1.png")
Button(root, image=ButtonPlay, bg="#FFFFFF", bd=0, height = 60, width =60,
       command=PlayMusic).place(x=215, y=487)

ButtonStop = PhotoImage(file="stop1.png")
Button(root, image=ButtonStop, bg="#FFFFFF", bd=0, height = 60, width =60,
       command=mixer.music.stop).place(x=130, y=487)

Buttonvolume = PhotoImage(file="volume.png")
Button(root, image=Buttonvolume, bg="#FFFFFF", bd=0, height = 60, width =60,
       command=mixer.music.unpause).place(x=20, y=487)

ButtonPause = PhotoImage(file="pause1.png")
Button(root, image=ButtonPause, bg="#FFFFFF", bd=0, height = 60, width =60,
       command=mixer.music.pause).place(x=300, y=487)

# Label
Menu = PhotoImage(file="menu.png")
Label(root, image=Menu).place(x=0, y=580, width=485, height=120)

Frame_Music = Frame(root, bd=2, relief=RIDGE)
Frame_Music.place(x=0, y=585, width=485, height=100)



Button(root, text="Browse Music", width=59, height=1, font=("calibri",
      12, "bold"), fg="Black", bg="#FFFFFF", command=AddMusic).place(x=0, y=550)

Scroll = Scrollbar(Frame_Music)
```

```
Playlist = Listbox(Frame_Music, width=100, font=("Times new roman", 10), bg="#333333", fg="grey",
selectbackground="lightblue", cursor="hand2", bd=0, yscrollcommand=Scroll.set)
Scroll.config(command=Playlist.yview)
Scroll.pack(side=RIGHT, fill=Y)
Playlist.pack(side=RIGHT, fill=BOTH)

# Execute Tkinter

root.mainloop()
```

Chapter 7

: Output Snapshot

Figure 7.1

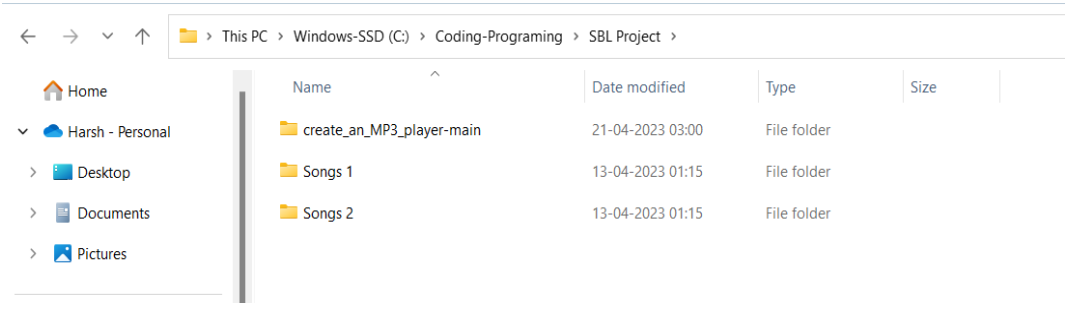

Figure 7.2

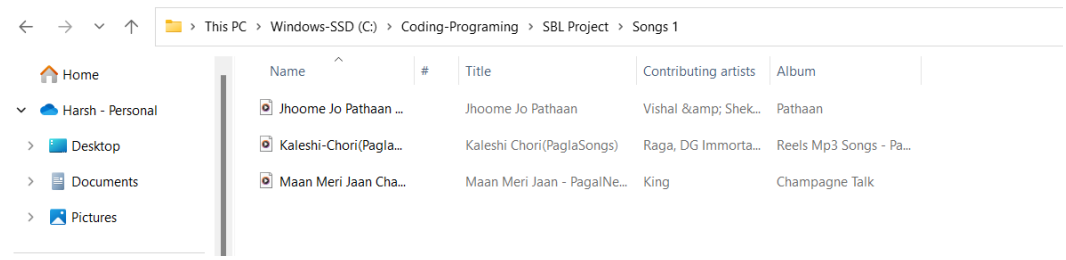

Figure 7.3

Chapter 8

Conclusion

The "Music Player: Play !t" project has been successfully developed as a web application for playing MP3 songs, utilizing Python programming language, Visual Studio, Pygame, and Tkinter for GUI development. The project has been developed with the aim of providing a practical learning experience in software development, specifically using Python and GUI libraries. Throughout the development process, various aspects of software engineering, such as requirements analysis, design, implementation, and testing, have been applied, allowing the student to gain valuable hands-on experience in software development.

The project has successfully achieved its objective of creating a music player that provides a seamless and hassle-free listening experience for users. The user-friendly GUI, with its clean and modern design, allows users to easily navigate through the application, create playlists, and manage their favorite songs. The functionalities of the music player, including play, pause, stop, and volume adjustment, have been implemented and tested for smooth and efficient operation.

The "Music Player: Play !t" project has also provided the student with an opportunity to gain proficiency in Python programming language, GUI development using Tkinter, and other relevant technologies. The student has learned how to design and implement a user-friendly interface, handle audio files, and manage playlists. The project has also fostered skills in problem-solving, code optimization, and project management, which are valuable in the field of software development.

In conclusion, the "Music Player: Play !t" project has been successfully developed as a web application for playing MP3 songs, providing a practical learning experience in software development. The project has enabled the student to gain proficiency in Python programming language, GUI development using Tkinter, and other relevant technologies, while also enhancing skills in problem-solving, code optimization, and project management. The report presents a detailed overview of the project's development process, functionalities, and outcomes. Overall, the "Music Player: Play !t" project has been a valuable learning experience and has successfully met its objectives.

Chapter 9

Reference

- [1]. Matthew E. P. Davies, Philippe Hamel, Kazuyoshi Yoshii, and Masataka, “Automatic Creation of Multi-Song Music Mashups” ,
- IEEE/ACM transactions on audio, speech, and language processing, vol. 22, no. 12, December 2014

- [2]. Sushmita G. Kamble ; A. H. Kulkarni, “IEEE/ACM transactions on audio, speech, and language processing”, vol. 22, no. 12, December 2014 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)

- [3]. Nirmal R Bhalani ; Jaikaran Singh ; Mukesh Tiwari, “Karaoke Machine implementation and validation using Out of Phase Stereo method” in 2012 International Conference on Communication, Information & Computing Technology (ICCICT)

- [4]. Karthik Subramanian Nathan ; Manasi Arun ; Megala S Kannan 2017 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)

- Linkedin

- Github

- Youtube