MAJOR PROJECT REPORT

on

PYTHON MUSIC PLAYER

(CSE VIII Semester Major project)

2022-2023

Submitted by:

Shrishti Namdeo (0203CS191044)

Nishant Pranav (0203CS191029)

Aman Singh (0203CS191004)

Arunesh Singh (0203CS191007)

Submitted to:

Prof. Nitin Jharbade

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ACKNOWLEDGEMENT

I would like to express my gratitude and my humblest of the humble respect towards the Almighty, the Most Beneficient, and the most Merciful for my successful completion of the previously mentioned project and its components.

I would like to thank my parents for their continuous support and encouragement. I also wish and thank them for providing me an opportunity to reach this far in my studies.

I would like to forward my thanks and the warmest of regards to our HOD Prof. Nitin Jharbade sir for his guidance and camaraderie during and after the course towards not just me, but all of my classmates as well.

I am hugely indebted by all the people who helped me during, before, and after the course completion. I would like to forward my regards and thanks to all of the mentioned above.

TABLE OF CONTENTS

1. INTRODUCATION **4**

2. ABSTRACT

2.1 About the Project 5

2.2 Working Of Project 5

3. REQUIREMENT AND INSTALLATIONS **6**

4. OBJECTIVE **7**

5. PROBLEM DESCRIPTION **8**

6. PROCESS FLOW **9**

7. DESIGN **10**

8. SCREENSHOTS **11**

9. WORKING **13**

10. CONCLUSION AND FUTURE SCOPE **16**

11. REFERNCES **17**

1. INTRODUCTION

We need an application that will allow us to play or listen to digital audio files. The music player is the device for playing MP3 and other digital audio files. The music GUI program application attempts to emulate the physical music player. This program allows you to play songs, music and all music files on your desktop or laptop. The main goal of this project is to enable users to play music and digital audio files. In order to be attractive to users, the application must have a simple but beautiful user interface. Music Player has options to play, pause and stop. We can have an interface to list the music files available. You can also allow users to list other digital audio files that are not music. Users are also waiting for the music player. have an interface that displays information about the file being played. Python has libraries that can play audio files, such as: B. Pygame, which allows you to work with media files in just a few lines of code.

2. ABSTRACT

2.1 About the Project

Audio is an important source of communication and is just as important today as text. We know that audio files are digital files. So you need a tool to run the digital files, or in other words to play the files. Without this tool or player, we can never listen to music, movies or the contents of an audio file. So we need music players. This device is used to play music and other digital audio files. You can do it yourself, without having to download and install premium music players. The music player GUI project idea tries to emulate the physical music player.

This program allows you to play songs, music and all music files on your desktop or laptop. Using Python is a basic programming application built using the Python programming language. It is a GUI program created using the Python Tkinter Pygame libraries. The music player application must be able to play a song, create and display a playlist, pause and resume a long song, and change the song. play the previous or next song.

2.2 Working Of Project

First, we Import the libraries. And then we Create an object of the Tkinter and Pygame libraries. After that we Create a window using Tkinter object. Further we add buttons that provide different functionalities like Playing the song, Pausing the song, Resuming the song and to Stop the song. The Player automatically add songs as we already given the path location. The song's information will be displayed in the window. Display screen will display the details of the entire playlist. Close button will automatically clear the song list and will stop playing the song.

3. REQUIREMENT AND INSTALLATIONS

• Installing Python package.

• Pygame - It is a Python library used to create video games. To create a music player with Python, we will be using the Pygame sound component. Pygame is also a very library that gives us the power of playing with different multimedia formats like audio, video, etc. We will be using Pygame's 'mixer.music' module for providing different functionality to are music player application, related to manipulation with the song tracks.

• Tkinter- This is the most popular and very easy to use library that comes with many widgets which helps in creating of seamless and nice-looking GUI Applications.

• Os- To access the song folder.

4 OBJECTIVE

The objective of this project is to create a GUI based python music player from scratch using python. For this project, you will need intermediate knowledge of the Tkinter widgets, basic knowledge about tkinter.filedialog, pygame.mixer, and os libraries.

To create an MP3 player using the Python programming language to play and listen to songs, MP3 files, and other digital audio files. The player must have a simple and easy-to-use interface with options for various functions and a screen around the entire playlist and buttons to turn off the player. The player should be able to play any song. It must be able to play MP3 files or other digital audio files. It should give the user the option to pause or resume the song. the user should be given basic details about the song being played.

5 PROBLEM DESCRIPTION

• To build an MP3 player using Python programming language to be able to play and listen to songs, MP3 files and other digital audio files.

• Determine the functionalities of the MP3 player.

• The player should be have a simple and easy to use GUI with options for various functions, display screen and buttons to shut down the player.

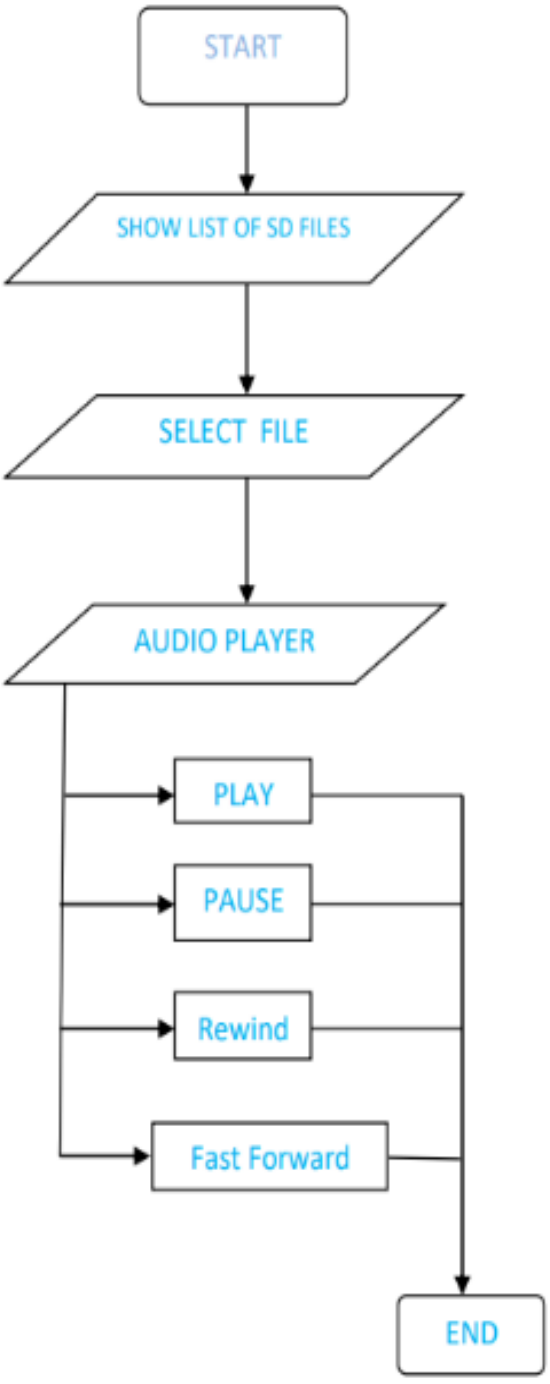
• The player should be able to play any song. It should be capable of playing MP3 files or any other digital audio files.

• The player should allow the user to browse through the contents of the computer drive to choose songs to be played or queued.

• It should provide the user with option to pause or resume the song.

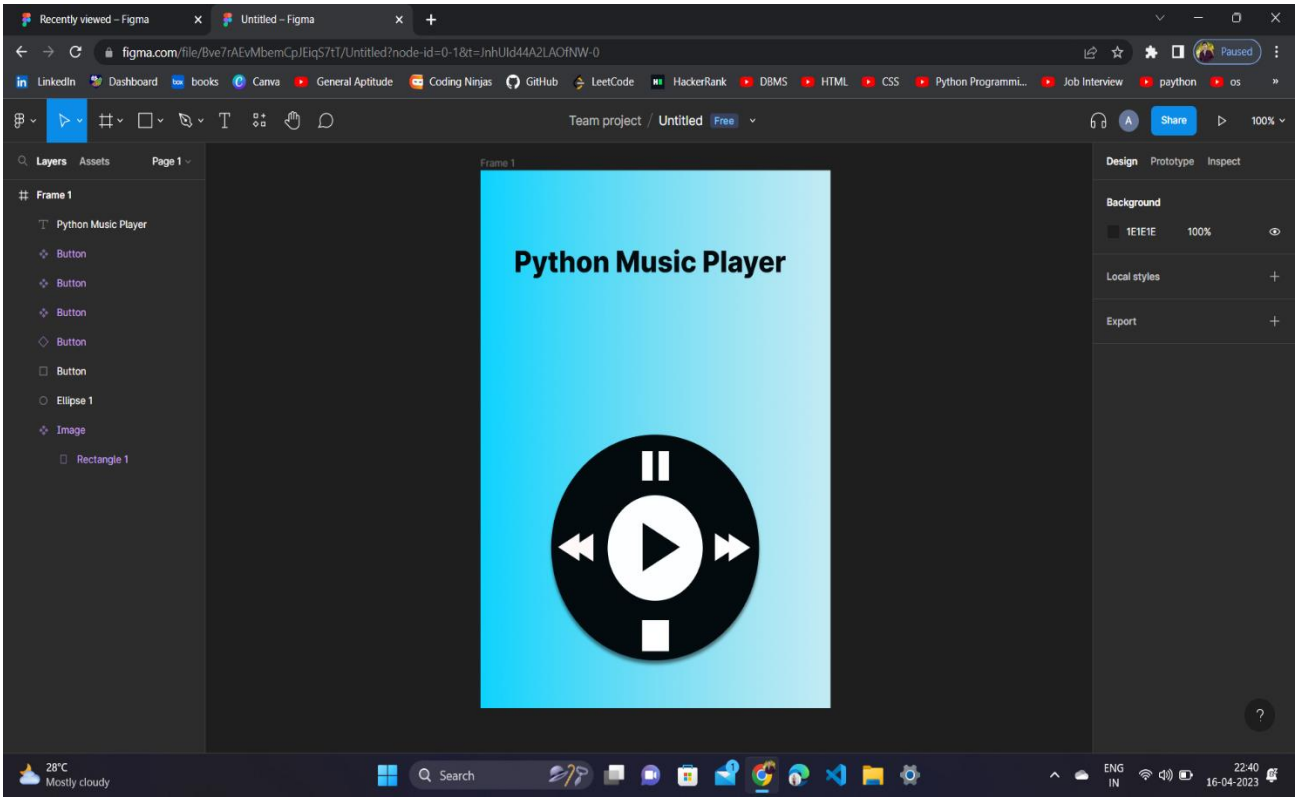
• Lastly, the user should get basic details about the current playing song.

6. PROCESS FLOW

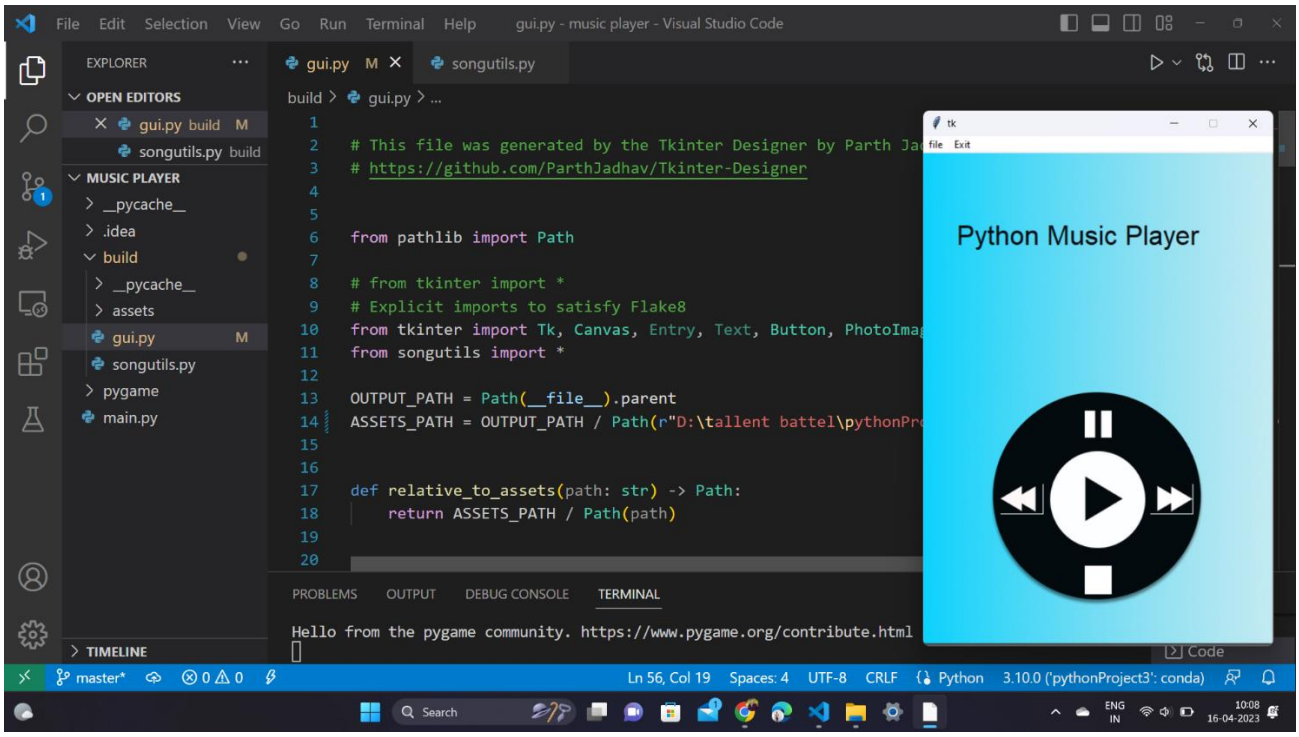


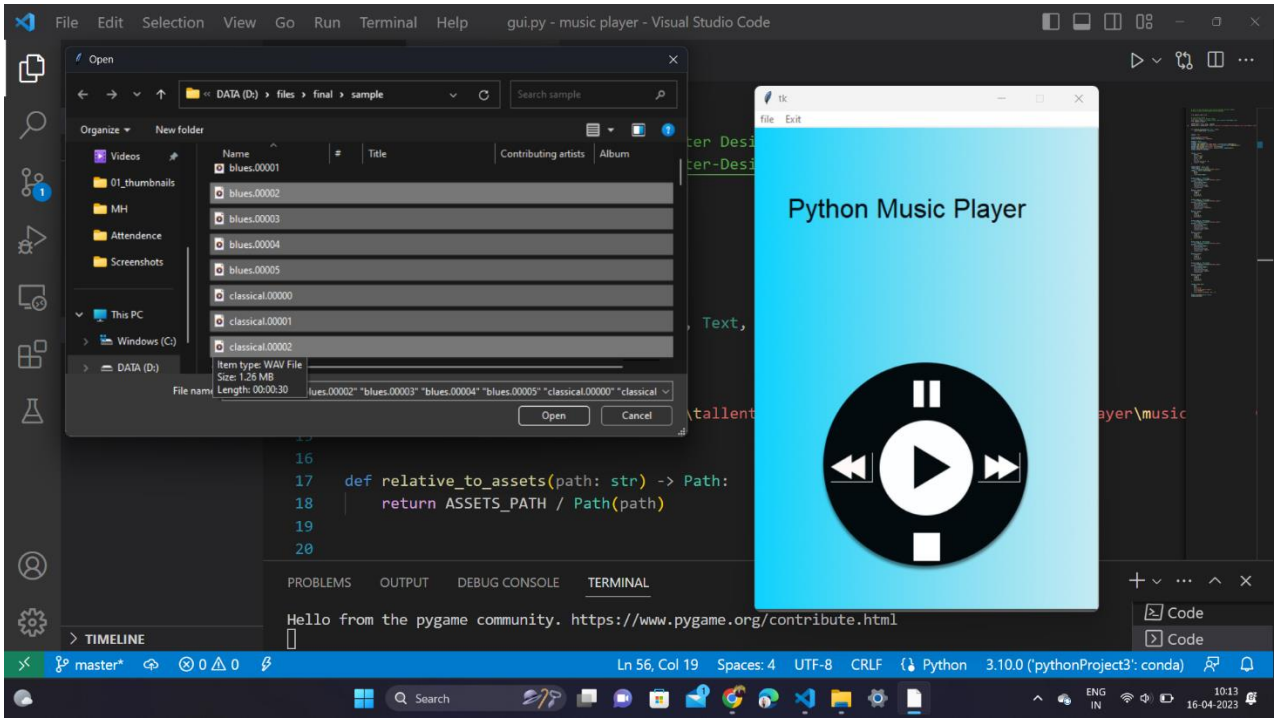
7. DESIGN

Front end design was created using Figma and with the help of TKINTER we create a GUI of that design.



8. SCREENSHOTS




File Edit Selection View Go Run Terminal Help gui.py - music player - Visual Studio Code
Open
DATA (D:) › files › final › sample
Search sample
Organize
New folder
Videos
01_thumbnails
MH
Attendence
Screenshots
This PC
Windows (C:)
DATA (D:)
Name
Title
Contributing artists
Album
blues.00001
blues.00002
blues.00003
blues.00004
blues.00005
classical.00000
classical.00001
classical.00002
Item type: WAV File
Size: 1.26 MB
Length: 00:00:30
Open
Cancel
tk
file Exit
Python Music Player
def relative_to_assets(path: str) -> Path:
return ASSETS_PATH / Path(path)
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
Hello from the pygame community. https://www.pygame.org/contribute.html
TIMELINE
Code
Code
master*
Ln 56, Col 19
Spaces: 4
UTF-8
CRLF
Python
3.10.0 ('pythonProject3': conda)
Search
ENG
IN
10:13
16-04-2023

9. WORKING

1.Project File Structure:

Here are the steps you will need to perform to build python music player project:

1. Importing all the libraries
2. Initializing the root window and pygame.mixer
3. Defining the play, stop, pause, resume and load functions
4. Creating the Label Frames and String Var variables
5. Placing all the objects in all the three Label Frames
6. Creating the final Label that will display the status of the song

```
5
6   from pathlib import Path
7   from tkinter import Tk, Canvas, Entry, Text, Button, PhotoImage, Menu
8   from songutils import *
9
```

2.Initializing the root window and pygame.mixer:

To make python music player project, we will use some elements in the music file of the mixer module. Those elements are:

1. **.load(filename) –** This method is used to load a file so that other actions can be performed on that file. The argument it takes is a file of a supported audio format [.wav, .mp3, .ogg].
2. **.play() –** This method is used to play the music file that was loaded by the .load() method.
3. **.stop()** – This method can stop the loaded file such that it cannot be resumed again.
4. **.pause() –** This method is used to pause a loaded file, at least with this option, it can be played again before needing to be loaded again.
5. **.unpause() –** This method is used to unpause a loaded audio file, also known as the resume option.

```
25
26    menubar = Menu()
27    filemenu = Menu(menubar)
28    filemenu.add_command(label="Open Song", command=lambda:loadsingle())
29    filemenu.add_command(label="Open Playlist", command=lambda:loadplaylist())
30    menubar.add_cascade(label="file", menu=filemenu)
31    menubar.add_command(label="Exit", command=lambda :window.quit())
32    window.config(menu=menubar)
33
```

3. Defining the play, stop, pause, resume and load functions:

```
17
18  def play():
19      if mixer.music.get_pos()>0:
20          mixer.music.unpause()
21      else:
22          mixer.music.play()
23
24  def pause():
25      mixer.music.pause()
26
27  def stop():
28      mixer.music.stop()
29
30  def loadplaylist():
31      global songindex
32      songlist.clear()
33      songindex=0
34      filepaths = filedialog.askopenfilenames()
35      for filename in filepaths:
36          songlist.append(filename)
37      mixer.init()
38      mixer.music.load(songlist[songindex])
39      # mixer.music.play()
40      print(songlist)
41
42  def next():
43      global songindex
44      songindex = songindex+1
45      if songindex>=len(songlist):
46          songindex = 0
47      mixer.music.load(songlist[songindex])
48      mixer.music.play()
49
50  def previous():
51      global songindex
52      songindex = songindex-1
53      if songindex<0:
54          songindex = len(songlist)-1
55      mixer.music.load(songlist[songindex])
56      mixer.music.play()
57
```

4. Code Explanation:

- We will have to initialize the mixer module of pygame by using the .init() method on it.
- Next, we will have to initialize our GUI window for python mp3 music player.
    - Tk() assignment will be used to initialize the window.
    - title() and .geometry() will be used to specify the title and initial geometry of the GUI window of music player.
    - resizable() method is used to permit/forbid the user from being able to resize the window. This method takes the arguments in the form of (height, width) ; the default for both of these are True but you can change it 0 or False to forbid the user from resizing the window.
    - update() and mainloop() methods are used to put the window on loop and stop it from closing the moment it opens.
    - Note: The line of code where the .mainloop() is written will be the last line that will be executed that updates the main window.

10. CONCLUSION AND FUTURE SCOPE

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.

11. REFERENCE

1. Various open source materials from Internet.

2. Training notes.

3. Discussion among the group and with guide.

4. Some requirements are gathered through various books from library.