REPORT FOR MUSIC PLAYER

As a project work for Course

PYTHON PROGRAMMING (INT 213)

Name : Pranjal Dwivedi
Registration Number : 12014734
Name : Akshat Bhartia
Registration Number : 12011925
Program : B.Tech CSE
Semester : Third
School : School of Computer Science and Engineering
Name of the University : Lovely Professional University

Date of submission :15th NOVEMBER 2021

Lovely Professional University
Jalandhar, Punjab, India.

LOVELY PROFESSIONAL UNIVERSITY

Transforming Education Transforming India

MUSIC PLAYER

15 NOVEMBER 2021

ABSTRACT:-

Music is **a basic need of human survival**.. Music is one of the ways we make sense of our lives, one of the ways in which we express feelings when we have no words, a way for us to understand things with our hearts when we can't with our minds.

.

We know that music can stimulate us both physically and psychologically; it can match and manipulate our moods and emotions. Studies tell us that this ability to change our state is one of the main reasons why people choose to listen to music. That's why having a good Music Player is important in todays time

ACKNOWLEDGEMENT:-

I would like to thank my mentor - Sukhvir Kaur Ma'am for her advice and inputs on this project. Many thanks to my friends as well, who spent countless hours to listen and provide feedbacks.

Table of Contents

INTRODUCTION:-

1. Context

This project has been done as part of my course for the CSE at Lovely Professional University. Supervised by Sukhvir Kaur Ma'am, two months to fulfill the requirements in order to succeed the module.

2. Motivations

Being extremely interested in music and keen to learn more about music, we both were motivated in creating something related to music and what better than creating an app that allows users to play their favourite music

TEAM MEMBERS:-

TEAM LEADER:-

Pranjal Dwivedi:-

Contributions:-

1. Coding(joined)
2. Logic Building(joined)
3. GUI
4. Interface Design

Akshat Bhartia:-

Contributions:-

1. Coding(joined)
2. Report
3. Logic Building(joined)

LIBRARIES AND MODULES:-

Tkinter:-

Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's *de facto* standard GUI. Tkinter is included with standard GNU/Linux, Microsoft Windows and macOS installs of Python.
The name *Tkinter* comes from *Tk interface*. Tkinter was written by Fredrik Lundh. Tkinter is free software released under a Python license
.

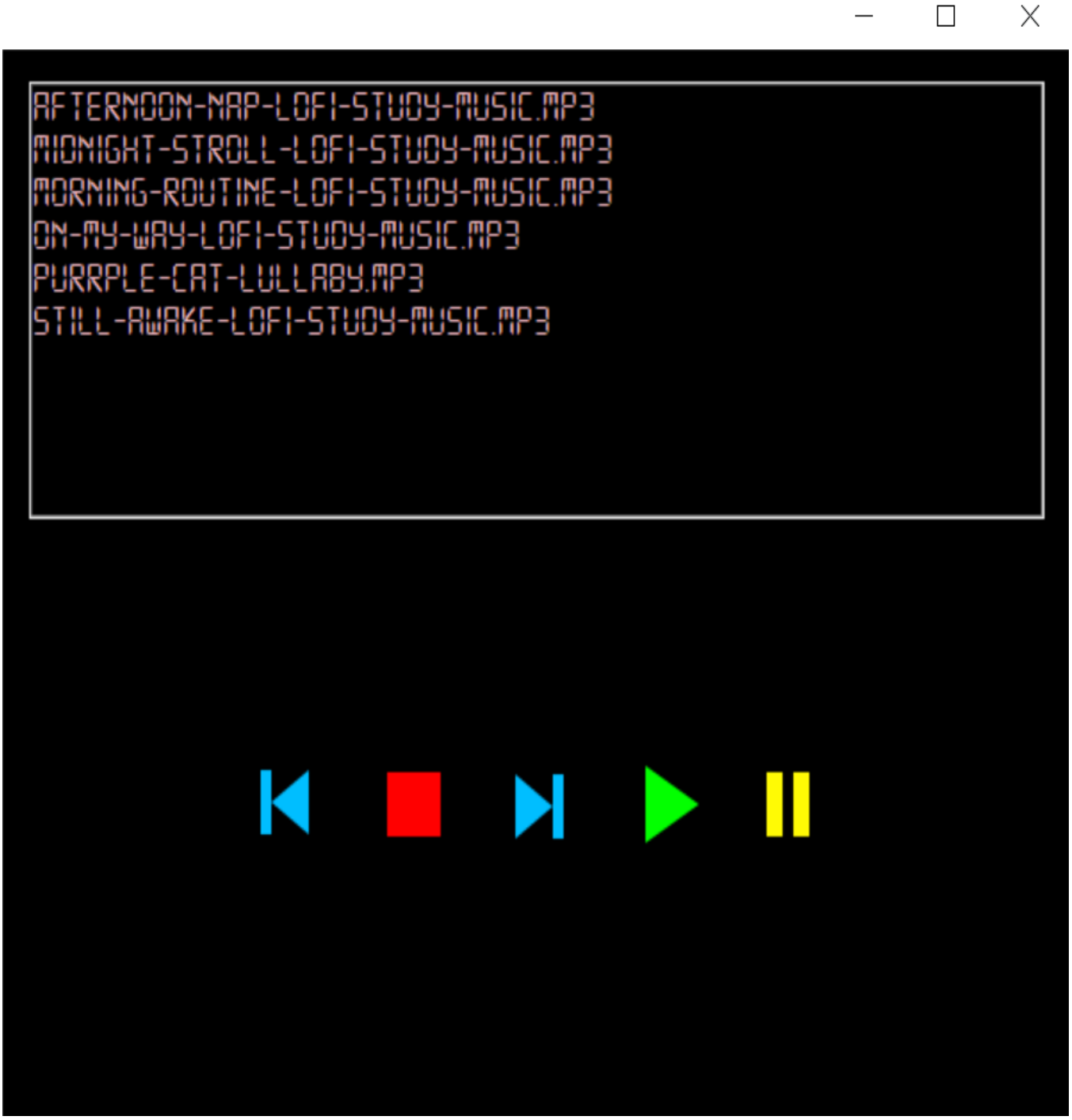
Fnmatch:-

This module is used for matching Unix shell-style wildcards. fnmatch() compares a single file name against a pattern and returns TRUE if they match else returns FALSE.
The comparison is case-sensitive when the operating system uses a case-sensitive file system.

Pygame:-

Pygame is a cross-platform set of Python modules which is used to create video games. It consists of computer graphics and sound libraries designed to be used with the Python programming language. Pygame was officially written by Pete Shinners to replace PySDL.. Pygame is suitable to create client-side applications that can be potentially wrapped in a standalone executable.

SCREENSHOTS:-



Code Snippets For Importing All The LIBRARIES And MODULES

```
import tkinter as tk
import os
import fnmatch
import pygame
from pygame import mixer
```

Code Snippets For Initializing Tkinter And Geometry Of The Canvas

```
canvas = tk.Tk()
canvas.title("Music Player:
canvas.geometry("600x500")
canvas.config(bg='black')
```

Code Snippets For Initializing Mixer And Creating Root Path For Music Files

```
#Initialized Mixer
mixer.init()

#Created Root Path For The MP3 Files
rootpath = "C:\\Users\hp\Desktop\Pyhton Project\Songs"
pattern = "*.mp3"
```

Code Snippets For Assigning Icons In Canvas

```
#Assigned Icons
prev_img = tk.PhotoImage(file = "prev_img.png")
next_img = tk.PhotoImage(file = "next_img.png")
pause_img = tk.PhotoImage(file = "pause_img.png")
stop_img = tk.PhotoImage(file = "stop_img.png")
play_img = tk.PhotoImage(file = "play_img.png")
```

Code Snippets For Different Functionalities For Icons

```
#Method For Selecting A Song And Playing It
def selectAndPlaySong():
    label.config(text = listBox.get("anchor"))
    mixer.music.load(rootpath + "\\" + listBox.get("anchor"))
    mixer.music.play()

#Method For Stopping A Song
def stopSong():
    mixer.music.stop()
    listBox.select_clear('active')

#Method For Playing The Next Song
def playNextSong():
    next_song = listBox.curselection()
    next_song = next_song[0] + 1
    next_song_name = listBox.get(next_song)
    label.config(text = next_song_name)

    mixer.music.load(rootpath + "\\" + next_song_name)
    mixer.music.play()

    listBox.select_clear(0, 'end')
    listBox.activate(next_song)
    listBox.select_set(next_song)
```

Code Snippets For Remaining Functions

```
#Method For Playing The Previous Song
def playPrevSong():
    prev_song = listBox.curselection()
    prev_song = prev_song[0] - 1
    prev_song_name = listBox.get(prev_song)
    label.config(text = prev_song_name)

    mixer.music.load(rootpath + "\\" + prev_song_name)
    mixer.music.play()

    listBox.select_clear(0, 'end')
    listBox.activate(prev_song)
    listBox.select_set(prev_song)

#Method For Pausing The Current Song
def pauseSong():
    if pauseButton["text"] == "Pause":
        mixer.music.pause()
        pauseButton["text"] = "Play"
    else:
        mixer.music.unpause()
        pauseButton["text"] = "Pause"
```

Code Snippets For Creating A ListBox And Space For Displaying The Current Song Name

```
#Initialized The ListBox For Displaying The Songs List
listBox = tk.Listbox(canvas, fg="pink", bg="black", width=100, font=('ds-digital', 15))
listBox.pack(padx=15, pady=15)

#Initialized The Space For Displaying The Current Song Name
label = tk.Label(canvas, text='', bg='black', fg='yellow', font=('ds-digital', 18))
label.pack(pady=15)

#Initialized The Body Of The Canvas
top = tk.Frame(canvas , bg='black')
top.pack(padx=10 , pady= 5, anchor= 'center')
```

Code Snippets For Initializing Different Icon Functionality

```
#Initialized The Prev Button
prevButton = tk.Button(canvas, text= "Prev", image = prev_img , bg = 'black', borderwidth = 0, command = playPrevSong)
prevButton.pack(pady=15 , in_= top , side="left")

#Initialized The Stop Button
stopButton = tk.Button(canvas, text= "Stop", image = stop_img , bg = 'black', borderwidth = 0, command = stopSong)
stopButton.pack(pady=15, in_= top , side="left")

#Initialized The Next Button
nextButton = tk.Button(canvas, text= "Next", image = next_img , bg = 'black', borderwidth = 0, command = playNextSong)
nextButton.pack(pady=15, in_= top , side="left")

#Initialized The Play Button
playButton = tk.Button(canvas, text= "Play", image = play_img , bg = 'black', borderwidth = 0, command = selectAndPlaySong)
playButton.pack(pady=15, in_= top , side="left")

#Initialized The Pause Button
pauseButton = tk.Button(canvas, text= "Pause", image = pause_img , bg = 'black', borderwidth = 0, command = pauseSong)
pauseButton.pack(pady=15, in_= top , side="left")
```

Code Snippet For Fetching The Files From The Directory By Using The Rootpath

```
#Fetching The Files From The Directory By Using The Rootpath
for root, dirs, files in os.walk(rootpath):
    for filename in fnmatch.filter(files, pattern):
        listBox.insert('end', filename)

canvas.mainloop()
```

REFRENCES:-

To conduct this project the following tools have been used :

- Jupyter notebook and spyder
- Pygame (Library) : https://www.pygame.org/news

1. Wikipedia:-

We have used this side for our basis knowledge gain of the methods that will be used in the project

2. GeeksForGeeks:-

We have used this site for our datasets gathering and asample case project.