A

MINI PROJECT REPORT ON

## SUR MUSIC PLAYER

Submitted in partial fulfillment of the requirements of the degree of

### Bachelor of Engineering In

**Computer Engineering**

By

38. Shubham Kothari

40. Aniket Kurkute

23. Atul Gadekar

### Under the guidance of Prof. Dhanashri Kanade

**Department of Computer Engineering**

K.C. College of Engineering and Management Studies and Research, Thane (E)

Mumbai University 2019-2020

# ACKNOWLEDGEMENT

No project is ever complete without the guidance of those expert who have already traded this past before and hence become master of it and as a result, our leader. So we would like to take this opportunity to take all those individuals who have helped us in visualizing this project.

We express our deep gratitude to our mini project guide Prof. Dhanashri Kanade for providing timely assistant to our query and guidance that she gave owing to her experience in this field for past many years. She has indeed been a lighthouse for us in this journey.

We extend our sincerity appreciation to all our Professors from K.C. College of Engineering & Management Studies & Research for their valuable suggestions during the designing of the project. Their contributions have been valuable in so many ways that we find it difficult to acknowledge of them individual.

We also express our gratitude to our HOD Prof. Mandar Ganjapurkar for extending his help directly and indirectly through various channels in our project work.

Thanking You.

# CERTIFICATE

This is to certify that the report on mini project entitled,

## SUR MUSIC PLAYER

### Project Members

38. Shubham Kothari

40. Aniket Kurkute

23. Atul Gadekar

As a partial fulfillment for Bachelor of Computer Engineering

Degree course of Mumbai University for Academic year 2018-2019

### INTERNAL GUIDE

Prof. Dhanashri Kanade

INTERNAL EXAMINER EXTERNAL EXAMINER

HOD PRINCIPAL

(Prof. Mandar Ganjapurkar) (Dr. Vilas Nitnaware)

### TABLE OF CONTENTS

|  |  |
| --- | --- |
| **Ch.No. Chapter Name** | **Page No** |
| 1. ABSTRACT | 01 |
| 2. INTRODUCTION | 02 |
| 3. PROGRAM CODE | 03 |
| 4. OUTPUT SCREENSHOTS | 04 |
| 5. APPLICATIONS | 05 |
| CONCLUSION | 06 |
| REFERENCES | 07 |

**ABSTRACT**

**My group loves to hear music and had a dream to make a music player with various features which are essential for extreme music lovers. Various developers have developed music player with various functions required for music all of them have made music players on various platforms such as vb.net, java etc. we found none players using python as major language**

**My group members have recently learnt Basic python language and wanted to explore python in deep to get somewhat deep knowledge of python we decided to make such application which use python**

**INTRODUCTION**

**To make SUR Mplayer we have used various libraries available online for python language some of used libraries are pygame, mutagen, ttkinter . SUR Mplayer Comes with basic functionality such as**

* **play**
* **pause**
* **stop**
* **rewind**
* **next**
* **volume scale**

**with the basic functionality various other functionality such as**

* **viewing lyrics**
* **Adding Lyrics**
* **Viewing Metadata of song**
* **Adding metadata for a song**
* **Playlist management**

**Are added to make SUR a complete Music player**

**Additional feature (coming soon):**

* **Equalizer**
* **Viewing all song list**

**Are been researched by Our group and will be added to the player as soon as possible**

**PROGRAM CODE**

# all imports

import os

import threading

import time

import tkinter.messagebox

from tkinter import \*

from tkinter import filedialog

from tkinter import ttk

from ttkthemes import themed\_tk as tk

from mutagen.mp3 import MP3

from pygame import mixer

from mutagen.easyid3 import EasyID3

from re import sub

# Global variables

queue = []

muted = FALSE

paused = FALSE

filename\_path=""

play\_song=""

current\_time=0

t1=''

close=False

total\_length=10

selected=0

i=0

abm=""

nextmusic=0

file='playlist/all.txt'

#initializing window

root = tk.ThemedTk()

root.get\_themes()# Returns a list of all themes that can be set

root.set\_theme("breeze") # Sets an available theme

root.geometry('585x400+200+200') # window geometry

root.title("SUR MPlayer")

root.iconbitmap(r'images/sur.ico')

mixer.init() # initializing the mixer

# functions

def on\_closing():

global paused

global close

paused = True

close=True

stop\_music()

root.destroy()

def pause\_music():

global paused

paused = TRUE

mixer.music.pause()

statusbar['text'] = "Music Paused"

def rewind\_music():

global current\_time

global selected

if current\_time<=7:

if selected==0:

play\_music()

else:

playlistbox.selection\_clear(0, tkinter.END)

playlistbox.select\_set(selected-1)

play\_music()

else:

play\_music()

statusbar['text'] = "Music Rewinded"

def set\_vol(val):

volume = round(float(val),0)

lowvol['text']=volume

volume=volume / 100

mixer.music.set\_volume(volume)

def seek\_play(val):

global current\_time

global play\_song

val=int(round(float(val)))

if val>current\_time+2 or val<current\_time-2:

try:

stop\_music()

mixer.music.play(loops=0, start=val)

current\_time=val

statusbar['text'] = "Playing music" + ' - ' + os.path.basename(play\_song)

except:

tkinter.messagebox.showerror('NO SONG SELECTED','PLZ Select a Song and try again')

play\_scale.set(0)

def mute\_music():

global muted

if muted: # Unmute the music

mixer.music.set\_volume(0.7)

volumeBtn.configure(image=volumePhoto)

volscale.set(70)

muted = FALSE

else: # mute the music

mixer.music.set\_volume(0)

volumeBtn.configure(image=mutePhoto)

volscale.set(0)

muted = TRUE

def show\_details(play\_song):

global total\_length

file\_data = os.path.splitext(play\_song)

if file\_data[1] == '.mp3':

audio = MP3(play\_song)

total\_length = audio.info.length

else:

a = mixer.Sound(play\_song)

total\_length = a.get\_length()

# div - total\_length/60, mod - total\_length % 60

mins, secs = divmod(total\_length, 60)

mins = round(mins)

secs = round(secs)

timeformat = '{:02d}:{:02d}'.format(mins, secs)

lengthlabel['text'] =timeformat

play\_scale['to']=total\_length

t1 = threading.Thread(target=start\_count, args=(total\_length,))

t1.start()

def start\_count(t):

global paused

global current\_time

# mixer.music.get\_busy(): - Returns FALSE when we press the stop button (music stop playing)

# Continue - Ignores all of the statements below it. We check if music is paused or not.

current\_time = 0

while current\_time <= t and mixer.music.get\_busy():

if paused:

continue

else:

mins, secs = divmod(current\_time, 60)

mins = round(mins)

secs = round(secs)

timeformat = '{:02d}:{:02d}'.format(mins, secs)

currenttimelabel['text'] =timeformat

play\_scale.set(current\_time)

time.sleep(1)

current\_time += 1

def play\_music(x=0):

global paused

global queue

global play\_song

global selected

if paused:

mixer.music.unpause()

statusbar['text'] = "Music Resumed"

paused = FALSE

else:

try:

stop\_music()

time.sleep(1)

selected\_song = playlistbox.curselection()

selected\_song = int(selected\_song[0])

selected\_song = selected\_song+x

selected = selected\_song

play\_it = queue[selected\_song]

play\_song=play\_it

abtmusic = EasyID3(play\_it)

titletxt['text']=abtmusic['title'][0]

albumtxt['text']=abtmusic['album'][0]

artisttxt['text']=abtmusic['artist'][0]

mixer.music.load(play\_it)

mixer.music.play()

statusbar['text'] = "Playing music" + ' - ' + os.path.basename(play\_it)

show\_details(play\_it)

except:

tkinter.messagebox.showerror('NO SONG SELECTED', 'PLZ Select a Song and try again')

titletxt['text']='No title'

albumtxt['text']='No album'

artisttxt['text']='No artist'

def stop\_music():

mixer.music.stop()

statusbar['text'] = "Music Stopped"

def next\_music():

global current\_time

global total\_length

global nextmusic

if nextmusic>2:

current\_time=total\_length+10

nextmusic=0

else:

mixer.music.stop()

current\_time+=10

mixer.music.play(loops=0, start=current\_time)

nextmusic+=1

print(nextmusic-1)

def del\_song():

global filename\_path

global queue

global file

try:

selected\_song = playlistbox.curselection()

selected\_song = int(selected\_song[0])

f=open(file,'r')

c=open('playlist/all1.txt','a')

var=f.read()

for lines in var.split('\n'):

if lines!=queue[selected\_song]:

if lines!="":

c.write(lines+"\n")

f.close()

c.close()

os.remove(file)

os.rename('playlist/all1.txt',file)

playlistbox.delete(selected\_song)

queue.pop(selected\_song)

tkinter.messagebox.showinfo('Info','Deletion sucessfull')

except:

tkinter.messagebox.showerror('Error','Please Select A song from list to remove song')

def moreabt\_song():

global abm

global abtm

global play\_it

selected\_song = playlistbox.curselection()

if not selected\_song:

tkinter.messagebox.showerror('Error','Please Select A song from list to display about song')

else:

selected\_song = int(selected\_song[0])

play\_it = queue[selected\_song]

try:

abtmusic = EasyID3(play\_it)

filename = abtmusic['title'][0]

abtm=tk.ThemedTk()

abtm.geometry('300x280+200+200')

abtm.title("About ~~ "+filename)

abtm.iconbitmap(r'images/sur.ico')

abm=Listbox(abtm,width=50,height=15)

abm.place(x=0,y=0)

for key in sorted(abtmusic.keys()):

val = abtmusic[key]

abm.insert(END,key+" : "+val[0])

editBtn = ttk.Button(abtm,text="Edit Data" ,command=ed\_gui)

editBtn.place(x=10,y=250)

except:

tkinter.messagebox.showerror('File not found', 'play the song to get info')

def ed\_gui():

global abm

global abtm

global play\_it

global selected\_song

global entry\_edval

def edit\_save():

abtmusic = EasyID3(play\_it)

try:

abtmusic[selected\_song]==str(entry\_edval.get())

abtmusic.save()

except:

tkinter.messagebox.showerror('Error in Saving','Saving cannot be completed as\n 1)Your windows is trying to interfere the program as an security threats\n2) Your Anitvirus is trying to interfere the program as an security threats\n3)Sorry nothing can be done for that ')

eabtm.destroy()

selected\_song=abm.curselection()

if not selected\_song:

tkinter.messagebox.showerror('Error','Please Select A song from list to display about song')

selected\_song = int(selected\_song[0])

abtmusic = EasyID3(play\_it)

l1=list(abtmusic.keys())

l1=sorted(l1)

selected\_song=l1[selected\_song]

eabtm=tk.ThemedTk()

eabtm.geometry('300x100+200+200')

eabtm.title("Edit Value")

title = ttk.Label(eabtm, text=selected\_song)

title.pack()

entry\_edval=ttk.Entry(eabtm,)

entry\_edval.pack()

saveBtn = ttk.Button(eabtm,text="SAVE" ,command=edit\_save)

saveBtn.pack()

abtm.destroy()

def about\_us():

tkinter.messagebox.showinfo('About SUR','This is an open Source Music player build using Python Tkinter by \nSHUBHAM KOTHARI\nANIKET KURKUTE\nATUL GADEKAR')

def browse\_file():

global filename\_path

global queue

global file

def add\_to\_q():

global queue

add\_to\_playlist(filename\_path)

mixer.music.queue(filename\_path)

if filename\_path=="":

filename\_path = filedialog.askopenfilename()

if filename\_path!="":

f=open(file,'a')

f.write(filename\_path+"\n")

f.close()

add\_to\_q()

else:

add\_to\_q()

filename\_path=""

def add\_to\_playlist(filename):

global queue

file\_data = os.path.splitext(filename)

if file\_data[1] == '.mp3':

abtmusic = EasyID3(filename\_path)

filename = abtmusic['title'][0]

else:

filename = os.path.basename(filename)

playlistbox.insert(END, filename)

queue.append(filename\_path)

playlistbox.config(yscrollcommand=scrollbar.set)

scrollbar.config(command=playlistbox.yview)

def open\_lyrics():

play\_it=''

def edit\_lyrics():

def save\_lyrics():

val=display\_label.get(1.0,END)

play\_it = queue[selected\_song]

abtmusic = EasyID3(play\_it)

var=abtmusic['title'][0]

f=open("lyrics/"+var+".txt",'a')

f.write(val)

f.close()

eabtm.destroy()

open\_lyrics()

global display\_label

eabtm=tk.ThemedTk()

eabtm.geometry('450x400+200+200')

eabtm.title("ADD LYRICS")

title = ttk.Label(eabtm, text="To Add Lyrics Copy Lyrics From Internet And Paste In Below text area")

title.pack()

display\_label = Text(eabtm, width=50,height=20, font="ariel", fg="black", bg="white")

display\_label.pack()

saveBtn = ttk.Button(eabtm,text="Save",command=save\_lyrics)

saveBtn.pack()

lyrm.destroy()

selected\_song = playlistbox.curselection()

if not selected\_song:

tkinter.messagebox.showerror('Error','Please Select A song from list to display about song')

else:

selected\_song = int(selected\_song[0])

play\_it = queue[selected\_song]

abtmusic = EasyID3(play\_it)

var=abtmusic['title'][0]

lyrm=tk.ThemedTk()

lyrm.title('LYRICS ~~'+var)

lyrm.geometry('590x280+200+200')

lyrm.iconbitmap(r'images/sur.ico')

abm=Listbox(lyrm,width=100,height=15)

abm.place(x=0,y=0)

LYscrollbar = Scrollbar(lyrm)

LYscrollbar.pack(side=RIGHT,fill=Y)

try:

f=open("lyrics/"+var+".txt",'r')

for x in f:

abm.insert(END,x.split())

f.close()

abm.config(yscrollcommand=LYscrollbar.set)

LYscrollbar.config(command=abm.yview)

except:

abm.insert(END,"no lyrics")

editBtn = ttk.Button(lyrm,text="Edit Data",command=edit\_lyrics)

editBtn.place(x=10,y=250)

def avplaylist():

path = 'playlist/.'

allplaylist=[]

def add\_playlist():

def create\_playlist():

name=entry\_edval.get()

if name=="":

tkinter.messagebox.showerror('Error','Name Is Not Valid')

else:

f=open('playlist/'+name+'.txt','a')

tkinter.messagebox.showinfo('Info','Playlist created sucessfully')

f.close()

edwin.destroy()

edwin=tk.ThemedTk()

edwin.geometry('300x100+200+200')

edwin.title("Edit Value")

title = ttk.Label(edwin, text='Enter name for playlist')

title.pack()

entry\_edval=ttk.Entry(edwin,)

entry\_edval.pack()

saveBtn = ttk.Button(edwin,text="SAVE",command=create\_playlist)

saveBtn.pack()

avplaylistgui.destroy()

def play\_playlist():

global file

mixer.music.stop()

playlistbox.delete(0,tkinter.END)

queue.clear()

selected\_playlist = avplaylist\_listbox.curselection()

selected\_playlist = int(selected\_playlist[0])

play\_it = allplaylist[selected\_playlist]

file=play\_it

on\_start(play\_it)

playlistbox.select\_set(0)

play\_music()

avplaylistgui=tk.ThemedTk()

avplaylistgui.title('ALL PLAYLISTS')

avplaylistgui.geometry('280x280+200+200')

avplaylistgui.iconbitmap(r'images/sur.ico')

avplaylist\_listbox=Listbox(avplaylistgui,width=100,height=15)

avplaylist\_listbox.place(x=0,y=0)

addBtn = ttk.Button(avplaylistgui,text="Add Playlist",command=add\_playlist)

addBtn.place(x=10,y=250)

playplaylistBtn = ttk.Button(avplaylistgui,text="Play Playlist",command=play\_playlist)

playplaylistBtn.place(x=90,y=250)

files = os.listdir(path)

for name in files:

allplaylist.append("playlist/"+name)

name=name.split('.')

avplaylist\_listbox.insert(END,name[0])

def on\_start(file='playlist/all.txt'):

global filename\_path

f=open(file,'r')

for x in f:

filename\_path=x.strip()

browse\_file()

filename\_path=""

f.close()

def main\_thread():

global queue

global close

global current\_time

global total\_length

global selected

while close==False:

i=0

if current\_time+3>total\_length:

if selected==len(queue)-1:

i=0-selected

playlistbox.selection\_clear(0, tkinter.END)

playlistbox.select\_set(0)

else:

i=1+selected

playlistbox.selection\_clear(0, tkinter.END)

playlistbox.select\_set(i)

stop\_music()

play\_music()

else:

continue

print("break")

# menu bar

menubar = Menu(root)

root.config(menu=menubar)

subMenu = Menu(menubar, tearoff=0)

menubar.add\_cascade(label="File", menu=subMenu)

subMenu.add\_command(label="Open", command=browse\_file)

subMenu.add\_command(label="Exit", command=root.destroy)

subMenu = Menu(menubar, tearoff=0)

menubar.add\_cascade(label="Help", menu=subMenu)

subMenu.add\_command(label="About Us", command=about\_us)

# widgets

playlistbox = Listbox(root,yscrollcommand='yes',width=78)

playlistbox.place(x=10,y=10)

scrollbar = Scrollbar(root)

scrollbar.pack(side=RIGHT,fill=Y)

moreBtn = ttk.Button(root, text="More About Song", command=moreabt\_song)

moreBtn.place(x=25,y=190)

lyricsBtn = ttk.Button(root, text="LYRICS", command=open\_lyrics)

lyricsBtn.place(x=155,y=190)

addBtn = ttk.Button(root, text="+ Add", command=browse\_file)

addBtn.place(x=255,y=190)

delBtn = ttk.Button(root, text="- Remove", command=del\_song)

delBtn.place(x=355,y=190)

playlistBtn = ttk.Button(root, text="Playlist", command=avplaylist)

playlistBtn.place(x=455,y=190)

titlelab= ttk.Label(root, text="Title : ",font='Times 12 bold')

titlelab.place(x=10,y=225)

titletxt= ttk.Label(root, text="No Title",font='Times 12 bold')

titletxt.place(x=55,y=225)

albumlab= ttk.Label(root, text="Album : ",font='Times 11')

albumlab.place(x=10,y=245)

albumtxt= ttk.Label(root, text="No Album",font='Times 11')

albumtxt.place(x=60,y=245)

artistlab= ttk.Label(root, text="Artist : ",font='Times 11')

artistlab.place(x=10,y=265)

artisttxt= ttk.Label(root, text="No Artist",font='Times 11')

artisttxt.place(x=55,y=265)

currenttimelabel = ttk.Label(root, text='--:--')

currenttimelabel.place(x=2,y=285)

play\_scale = ttk.Scale(root, from\_=0, to=100, orient=HORIZONTAL,length=500,command=seek\_play)

play\_scale.place(x=35,y=285)

lengthlabel = ttk.Label(root, text='--:--')

lengthlabel.place(x=538,y=285)

prevPhoto = PhotoImage(file='images/previous.png').subsample(3,3)

prevBtn = ttk.Button(root, image=prevPhoto, command=rewind\_music)

prevBtn.place(x=10,y=315)

playPhoto = PhotoImage(file='images/play.png').subsample(2,2)

playBtn = ttk.Button(root, image=playPhoto, command=play\_music)

playBtn.place(x=60,y=310)

pausePhoto = PhotoImage(file='images/pause.png').subsample(2,2)

pauseBtn = ttk.Button(root, image=pausePhoto, command=pause\_music)

pauseBtn.place(x=120,y=310)

nextPhoto = PhotoImage(file='images/next.png').subsample(3,3)

nextBtn = ttk.Button(root, image=nextPhoto, command=next\_music)

nextBtn.place(x=180,y=315)

stopPhoto = PhotoImage(file='images/stop.png').subsample(3,3)

stopBtn = ttk.Button(root, image=stopPhoto, command=stop\_music)

stopBtn.place(x=230,y=315)

mutePhoto = PhotoImage(file='images/mute.png').subsample(2,2)

volumePhoto = PhotoImage(file='images/volume.png').subsample(2,2)

volumeBtn = ttk.Button(root, image=volumePhoto, command=mute\_music)

volumeBtn.place(x=380,y=320)

lowvol = ttk.Label(root, text="0")

lowvol.place(x=420,y=325)

volscale = ttk.Scale(root, from\_=0, to=100, orient=HORIZONTAL, command=set\_vol)

volscale.set(70)

set\_vol('70') # implement the default value of scale when music player starts

mixer.music.set\_volume(0.7)

volscale.place(x=445,y=325)

highvol = ttk.Label(root, text="100")

highvol.place(x=545,y=325)

# status bar

statusbar = ttk.Label(root, text="Welcome to SUR MPlayer", relief=SUNKEN, anchor=W, font='Times 10 italic')

statusbar.pack(side=BOTTOM, fill=X)

# end

root.protocol("WM\_DELETE\_WINDOW", on\_closing)

on\_start()

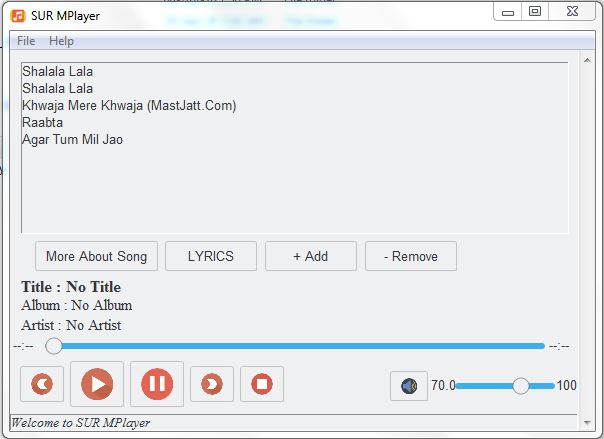
tm = threading.Thread(target=main\_thread,)

tm.start()

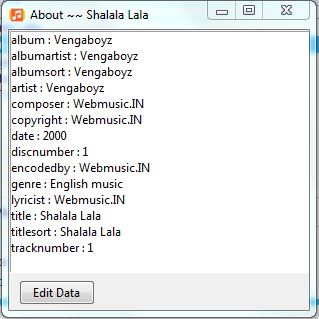
root.mainloop()

**OUTPUT SCREENSHOTS**

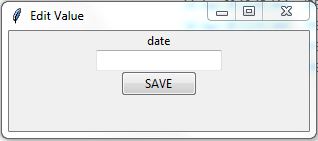
MAIN SCREEN



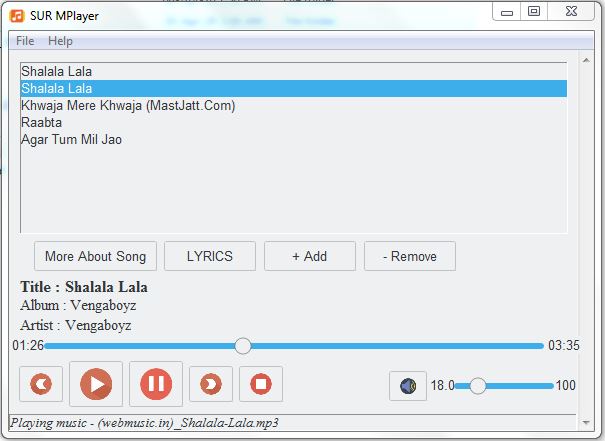
More about song screen



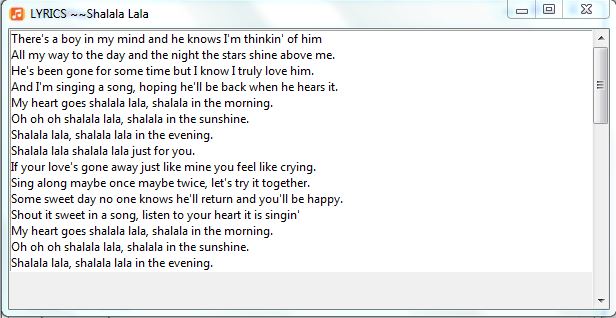
Edit metadata screen



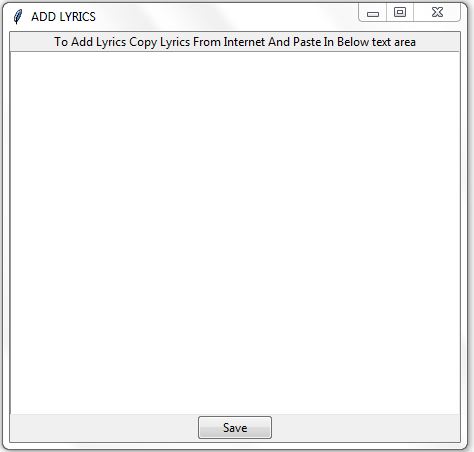
Screen while playing song



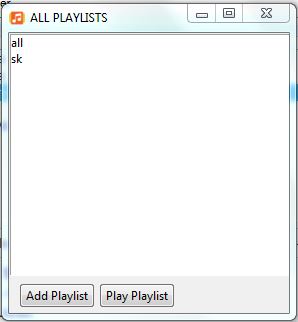
Lyrics screen



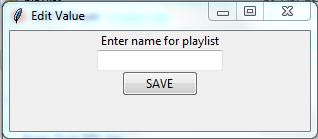
Add lyrics to song screen



All Playlist screen



Enter value for new playlist screen



**APPLICATION**

**SUR MPLAYER is an open source music player and can be use for personal work (for playing music at home ,shops, restaurants etc**

**The player can be used in party’s**

**The metadata editor can be used to edit wrongly entered data**

**CONCLUSION**

**By making Sur Mplayer I and my team have learned various concepts related to**

* **Threading in python**
* **GUI in python**
* **Music in python**
* **Writing an optimized code in python**

**This mini project has encouraged us to make similar projects and increase our development skills**

**REFERENCE**

* [**https://github.com/attreyabhatt/Python-Music-Player**](https://github.com/attreyabhatt/Python-Music-Player)
* [**https://www.javatpoint.com/python-tkinter-listbox**](https://www.javatpoint.com/python-tkinter-listbox)
* [**https://mutagen.readthedocs.io/en/latest/user/gettingstarted.html**](https://mutagen.readthedocs.io/en/latest/user/gettingstarted.html)