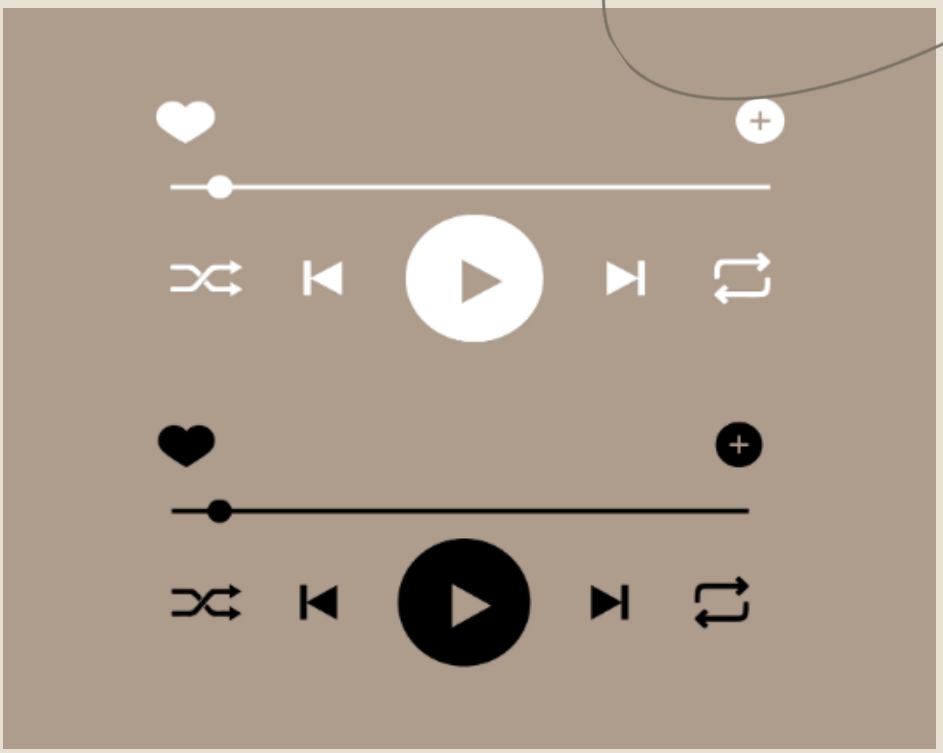
PROJECT REPORT



MUSIC PLAYER

Submitted By-
Mahek Trivedi
12-A

INDEX

NEED FOR PROJECT

Amusement, joy, eroticism, beauty, relaxation, sadness, dreaminess, triumph, anxiety, scariness, annoyance, defiance, and feeling pumped up, whatever feeling one goes through music lets you float. In order for us to listen to music freely without any kind of disturbance from the forced advertisements, frozen screens and loading pages we get from the apps available, utilizing our skills can be essential. This project aims to create a Music Playlist whose display is a sight to sore eyes, has features that are easy to use, and this time, is our creation and of course, add free.

USER'S MANUAL

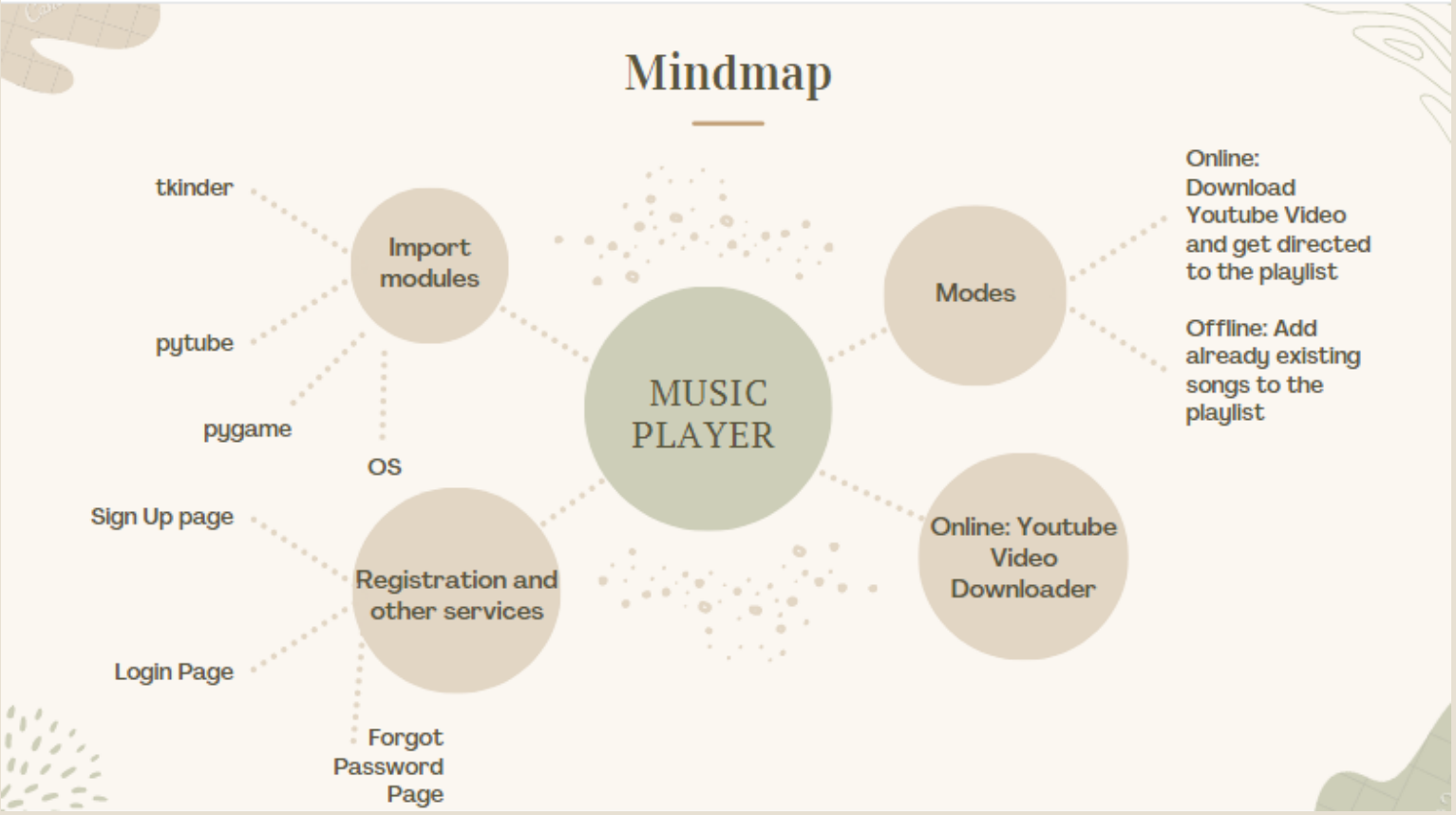
HARDWARE USED :

- Modern Operating System:
  - Windows 7 or 10
  - Mac OS X 10.11 or higher, 64-bit
  - Linux: RHEL 6/7, 64-bit (almost all libraries also work in Ubuntu)
- x86 64-bit CPU (Intel / AMD architecture)
- 4 GB RAM
- 5 GB free disk space

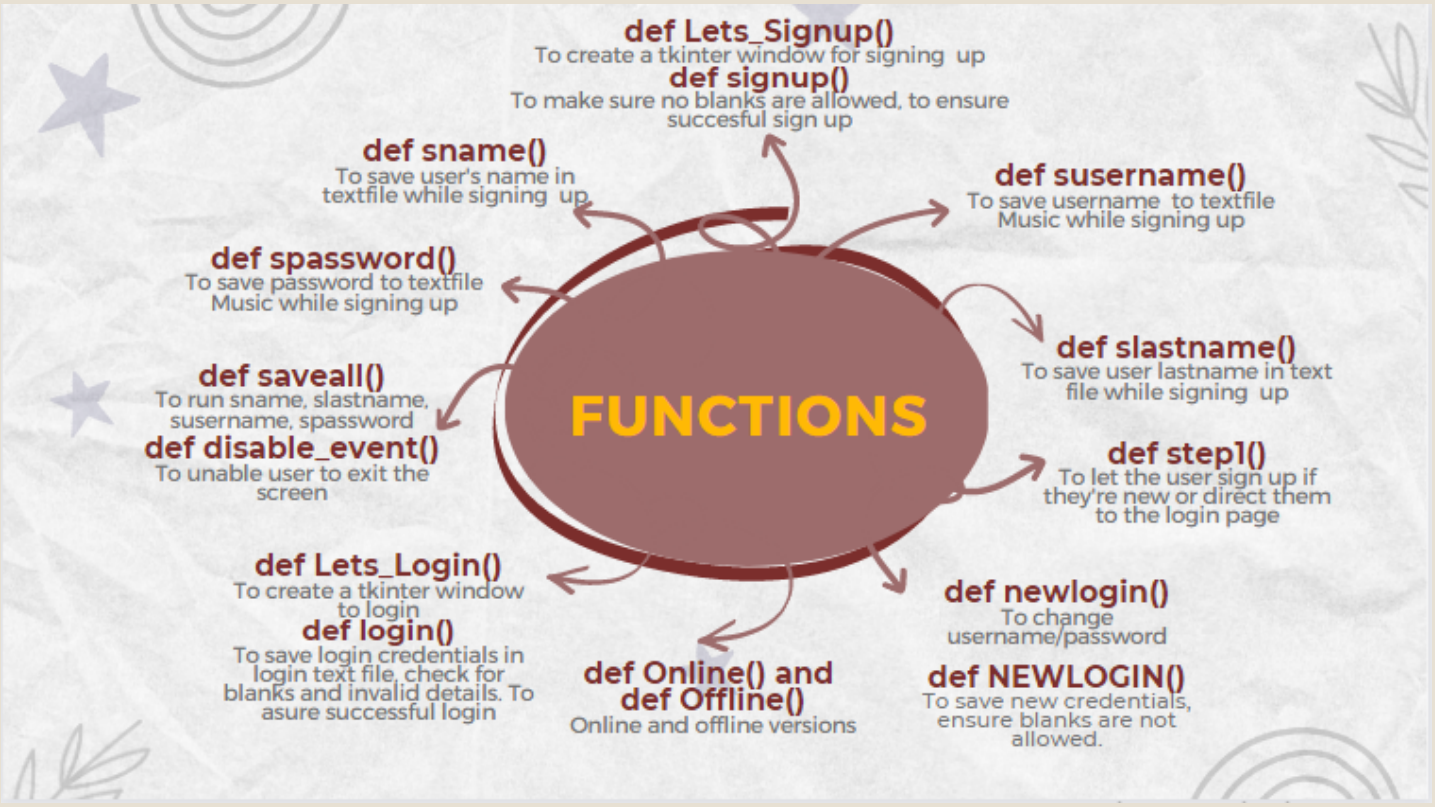
SOFTWARE REQUIRED :

1:Python
2:Tkinter
3:Pytube
4:Pygame

MENU TREE:


Mindmap
MUSIC PLAYER
Import modules
tkinder
pytube
pygame
OS
Registration and other services
Sign Up page
Login Page
Forgot Password Page
Modes
Online: Download Youtube Video and get directed to the playlist
Offline: Add already existing songs to the playlist
Online: Youtube Video Downloader

FUNCTIONS


def Lets_Signup()
To create a tkinter window for signing up
def signup()
To make sure no blanks are allowed, to ensure succesful sign up
def sname()
To save user's name in textfile while signing up
def susername()
To save username to textfile Music while signing up
def spassword()
To save password to textfile Music while signing up
def slastname()
To save user lastname in text file while signing up
FUNCTIONS
def saveall()
To run sname, slastname, susername, spassword
def disable_event()
To unable user to exit the screen
def step1()
To let the user sign up if they're new or direct them to the login page
def Lets_Login()
To create a tkinter window to login
def login()
To save login credentials in login text file, check for blanks and invalid details. To asure successful login
def newlogin()
To change username/password
def Online() and def Offline()
Online and offline versions
def NEWLOGIN()
To save new credentials, ensure blanks are not allowed.

WIDGETS

- Tk() is a top level widget that is used to create the main application window in which we will be building our python project.
- The Toplevel() widget is used to create and display the toplevel windows which are directly managed by the window manager.
- The title() method is used to give a name to a python mp3 player application which is displayed at the top.
- The geometry() method is used to decide the size of tkinter window
- The mixer.init() program is used to initialize the mixer module so that we can use its various functions
- The protocol() method, here, is used to disabling the user from deleting the window
- Tkinter Photoimage is one of the built-in methods which has been used to add the user-defined images in the application
- The Label is used to specify the container box where we can place the text or images.
- Place() allows you to precisely position widgets within its container using the (x, y) coordinate system
- The Entry widget is used to provide the single line text-box to the user to accept a value from the user
- The get() method is used to return the data entered in an Entry widget
- To open a text file we use open()
- To write in a text file we use write()

- The messagebox() widget is used to to show a particular message to users
- The withdraw() method is used to hide the tkinter/Toplevel window without deleting it
- The Button widget is a standard Tkinter widget, which is used for various kinds of buttons. A button is a widget which is designed for the user to interact with
- The readlines() method reads all lines of a text file as a member of a list
- The close() method is used to close a file
- To find the length of a list, we use len()
- The mainloop() is an infinite loop used to run the application, wait for an event to occur and process the event as long as the window is not closed
- The pytube module provides us with YouTube(), download() and also helps us stream the url
- The tkinter. filedialog module provides classes and factory functions for creating file/directory selection windows
- The chdir() method in Python used to change the current working directory to specified path
- Python method listdir() returns a list containing the names of the entries in the directory given by path
- The python string endswith() method checks if the input string ends with the specified suffix
- To insert text in the widget, you need to use insert()
- The pygame module provides us with play(), pause(), unpause() and stop() methods
- The iconphoto() method is used to set the titlebar icon of any tkinter/toplevel window

CODE:

```
#USE TKINTER(STANDARD GUI APPLICATION TO IMPORT WIDGETS SUCH AS MESSAGEBOX, TTK AND FILEDIALOG
from tkinter import*
from tkinter import messagebox
from tkinter import ttk,filedialog

#IMPORTING WIDGETS FROM PYTUBE AND PYGAME MODULES
from pytube import YouTube
from pygame import mixer

#IMPORTING OPERATING SYSTEM
import os

# USE PYTHON IMAGING LIBRARY TO IMPORT IMAGETK AND IMAGE
from PIL import ImageTk, Image

#FUNCTION TO SIGN UP WITH THE APPLICATION
def Lets_Signup():
    #Creating tkinter window
    root=Tk()
    root.title("Sign Up")
    root.geometry("730x550")
```

```
    #TO UNABLE THE USER TO SIGN IN WITHOUT FILLING THE
FORM
    def disable_event():
        pass
    root.protocol("WM_DELETE_WINDOW", disable_event)

    #Adding background image and other required labels and
entry widgets
    img = PhotoImage(file="FINAL_SIGNUP.png")
    label = Label(root,image=img)
    label.place(x=0, y=0)

    Label(root,text="Name:",font=("Comic Sans
MS",10),bg="white").place(x=450,y=70)
    Label(root,text="Last Name:",font=("Comic Sans
MS",10),bg="white").place(x=450,y=120)
    Label(root,text="Username:",font=("Comic Sans
MS",10),bg="white").place(x=450,y=170)
    Label(root,text="Password:",font=("Comic Sans
MS",10),bg="white").place(x=450,y=220)

    #ASKING USER TO ENTER NAME AND STORING THE DATA IN
A TEXT FILE FOR FURTHER REFERENCE
    name = Entry(root,bd=5)
    name.place(x=535,y=70)

    def sname():

        NAME = name.get()
        f=open("Music.txt","w")
        f.write(NAME)
```

```
#ASKING USER TO ENTER LASTNAME AND STORING THE
DATA IN A TEXT FILE FOR FURTHER REFERENCE
    lastname=Entry(root,bd=5)
    lastname.place(x=535,y=120)

    def slastname():

        LASTNAME = lastname.get()
        f=open("Music.txt","a")
        f.write("\n")
        f.write(LASTNAME)

    username=Entry(root,bd=5)
    username.place(x=535,y=170)

    #ASKING USER TO ENTER USERNAME AND STORING THE
DATA IN A TEXT FILE FOR FURTHER REFERENCE
    def susername():

        USERNAME = username.get()
        f=open("Music.txt","a")
        f.write("\n")
        f.write(USERNAME)

    password=Entry(root,bd=5)
    password.place(x=535,y=220)
```

```
    #ASKING USER TO ENTER PASSWORD AND STORING THE
DATA IN A TEXT FILE FOR FURTHER REFERENCE
    def spassword():

        PASSWORD = password.get()
        f=open("Music.txt","a")
        f.write("\n")
        f.write(PASSWORD)
        f.write("\n")

    #ADDING RESTRICTIONS SO THE USER CAN'T LEAVE
BLANKS EMPTY
    def signup():
        f=open("Music.txt","r")
        d=f.readlines()
        if '\n' in d:
            messagebox.showinfo("","Blank Not Allowed")

        else:
            messagebox.showinfo("","You have successfully signed
in")
            root.withdraw()

    def save_all():
        sname()
        slastname()
        susername()
        spassword()
        for i in range(1):
            signup()
```

```
    Button(root,text="Sign up",font=("Comic Sans
MS",15),command=save_all,height=2,width=10,bd=6,bg='light
pink',fg='black').place(x=487,y=390)
    root.mainloop()


 #ASKING THE USER TO CHOOSE A MODE, OFFLINE OR ONLINE
def mode():
    MODE=Toplevel()
    MODE.title("Select mode of running app")
    MODE.geometry("280x220")
    MODE['background']="white"
    img = PhotoImage(file="CHOOSEMODE.png",master=MODE)
    label = Label(MODE,image=img)
    label.place(x=0, y=0)

    #Offline
    Button(MODE,text="Offline,I have videos",font=("Comic
Sans
MS",10),command=Offline,height=1,width=17,bd=1,bg='white',f
g='purple',relief='solid').place(x=125,y=99)

    #Online
    Button(MODE,text=""Online:Download
the videos"",font=("Comic Sans
MS",10),command=Online,height=1,width=16,bd=1,bg='white',f
g='purple',relief='solid').place(x=125,y=9)
    MODE.mainloop()

#ONLINE VERSION
def Online():
```

```
    #CREATING TOPLEVEL WINDOW AND ADDING
BACKGROUND IMAGE
    ONL=Toplevel()
    ONL.title("Youtube Video Downloader")
    ONL.geometry("650x400")
    ONL['background']="white"
    img = PhotoImage(file="ONLINEMODE.png",master=ONL)
    label = Label(ONL,image=img)
    label.place(x=0, y=0)

    #CREATING REQUIRED LABELS AND ENTRIES
    label_1=Label(ONL,text="'Youtube Video
    Downloader"',font=("Comic Sans
MS",25),bg="white",fg="maroon").place(x=410,y=30)
    link=StringVar()
    label_2=Label(ONL,text="Paste link here:",font=("Comic Sans
MS",15),bg="light pink",fg="maroon").place(x=450, y=160)

    #Asking the user to enter the link of the video they wish to
download

Entry(ONL,width=20,bd=2,relief="solid",textvariable=link).pla
ce(x=460,y=210)
    label_3=Label(ONL,text="'Your videos will be
    saved in 'My Songs' folder"',font=("Comic Sans
MS",11),bg="white",fg="maroon").place(x=420,y=260)


    #YOUTUBE VIDEO DOWNLOADER CODE
    def downloader():
        url=YouTube(str(link.get()))
        video=url.streams.first()
```

```
        video.download('New songs') #DOWNLOAD PATH MAY BE
NEEDED
        label_4=Label(ONL,text="Downloaded",font=("Comic Sans
MS",15)).place(x=470,y=5)



    btn=Button(ONL,text="Download",font=("Comic Sans
MS",15),bg='light
pink',padx=2,command=downloader).place(x=415,y=330)
    btn=Button(ONL,text="Done!",font=("Comic Sans
MS",15),bg='light
pink',padx=2,command=mode).place(x=550,y=330)

    ONL.mainloop()

#OFFLINE VERSION-USES ALREADY EXISTING VIDEOS .OR.
CAN BE USED AFTER DOWNLOADING THE VIDEOS
def Offline():
    OFL= Toplevel()
    OFL.title("music player")
    OFL.geometry("920x670+290+85")
    OFL.configure(bg="#0f1a2b")
    OFL.resizable(False,False)

    mixer.init()

    def open_folder():
        path=filedialog.askdirectory()
        if path:
            os.chdir(path)
            songs=os.listdir(path)
            #print(songs)
```

```
        for song in songs:
            if song.endswith(".mp3"):
                playlist.insert(END,song)
    def play_song():
        music_name=playlist.get(ACTIVE)
        mixer.music.load(playlist.get(ACTIVE))
        mixer.music.play()
        music.config(text=music_name[0:-4])


    #Icon
    image_icon=PhotoImage(file="logo.png")
    OFL.iconphoto(False,image_icon)

    Top=PhotoImage(file="top.png")
    Label(OFL,image=Top,bg="#0f1a2b").pack()

    #Logo
    Logo=PhotoImage(file="logo.png")
    Label(OFL,image=Logo,bg="#0f1a2b").place(x=65,y=115)

    #Button
    play_button=PhotoImage(file="play.png")

Button(OFL,image=play_button,bg="#0f1a2b",bd=0,command
=play_song).place(x=100,y=400)

    stop_button=PhotoImage(file="stop.png")

Button(OFL,image=stop_button,bg="#0f1a2b",bd=0,command
=mixer.music.stop).place(x=30,y=500)
```

```
    resume_button=PhotoImage(file="resume.png")

Button(OFL,image=resume_button,bg="#0f1a2b",bd=0,comm
and=mixer.music.unpause).place(x=115,y=500)

    pause_button=PhotoImage(file="pause.png")

Button(OFL,image=pause_button,bg="#0f1a2b",bd=0,comman
d=mixer.music.pause).place(x=200,y=500)

    #label

music=Label(OFL,text="",font=('arial',15),fg='white',bg="#0f1a2
b")
    music.place(x=150,y=340,anchor="center")

    #Music
    Menu=PhotoImage(file="menu.png")

Label(OFL,image=Menu,bg="#0f1a2b").pack(padx=10,pady=50,
side=RIGHT)

    music_frame=Frame(OFL,bd=2,relief=RIDGE)
    music_frame.place(x=330,y=350)

    Button(OFL,text="Open
Folder",width=100,height=2,font=('arial',10,'bold'),fg='white',bg
="#21b3de",command=open_folder).place(x=330,y=300)
    scroll=Scrollbar(music_frame)

playlist=Listbox(music_frame,width=100,font=("arial",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=LEFT, fill=BOTH)
    OFL.mainloop()

#FUNCTION TO LOGIN AFTER SIGNING UP
def Lets_Login():

    #CREATING TKINTER WINDOW
    log=Tk()
    log.title("Login")
    log.geometry("630x490")
    img = PhotoImage(file="FINAL_LOGIN.png")
    Label(log,image=img).place(x=0, y=0)

    #TO UNABLE THE USER TO SIGN IN WITHOUT FILLING THE
FORM
    def disable_event():
        pass
    log.protocol("WM_DELETE_WINDOW", disable_event)

    #CREATING REQUIRED LABELS AND ENTRIES
    Label(log,text=" Username: ",font=("Comic Sans
MS",10),bg="white",fg="blue",borderwidth=1,
relief="solid").place(x=375,y=150)
    Label(log,text=" Password: ",font=("Comic Sans
MS",10),bg="white",fg="blue",borderwidth=1,
relief="solid").place(x=375,y=185)

    username=Entry(log,bd=3, width=10)
    username.place(x=465,y=150)
```

```
    password=Entry(log,bd=3,width=10)
    password.place(x=465,y=185)

    #FUNCION TO MAKE SURE THE GIVEN CREDENTIALS ARE
VALID
    def login():
        L=[]
        fl=open("Music.txt","r")
        d=fl.readlines()
        l=[]
        for i in d:
            l.append(i.replace("\n",""))

        l1=open("login.txt","w")
        User = username.get()
        l1.write(User)
        l1.close()
        l2=open("login.txt","a")
        Pass = password.get()
        l2.write("\n")
        l2.write(Pass)
        l2.close()
        l3=open("login.txt","r")
        dd=l3.readlines()

        ll=[]
        for j in dd:
            ll.append(j.replace("\n",""))

        if len(ll)!=2:
            messagebox.showinfo("","Blank not allowed")
        if len(ll)==2:
```

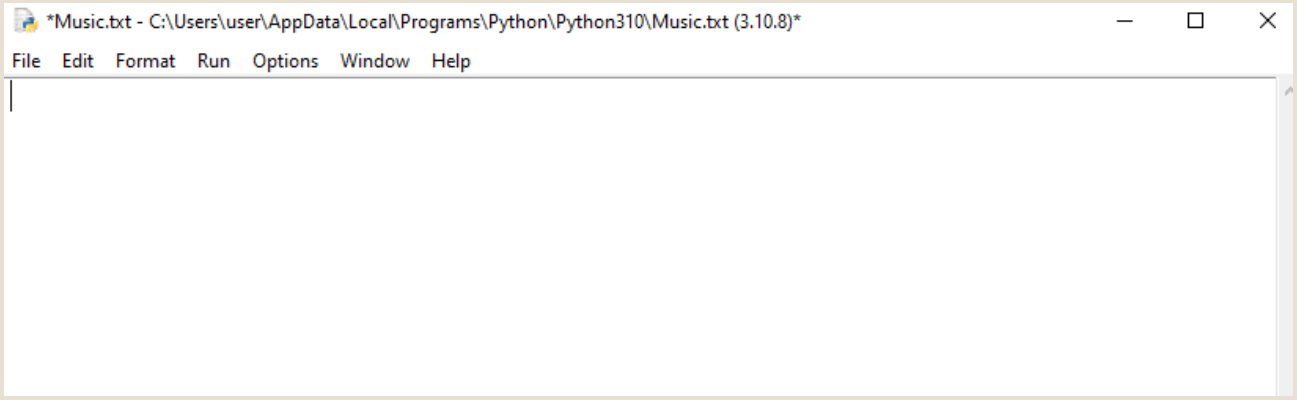
```
        if (ll[0]==l[2] and ll[1]==l[3]):
            messagebox.showinfo("","Login Successful")
            log.withdraw()
            mode()
        if (ll[0]==l[4] and ll[1]==l[5]):
            messagebox.showinfo("","Login Successful")
            log.withdraw()
            mode()
    if len(l)==7:
        if (ll[0]==l[6] and ll[1]==l[7]):
            messagebox.showinfo("","Login Successful")
            log.withdraw()
            mode()
        for k in ll:
            if k not in l[2:]:
                messagebox.showinfo("","Invalid Credentials")
                break

    Button(log,text="Login",font=("Comic Sans
MS",12),command=login,height=2,width=10,bd=6,bg='light
blue',fg='black').place(x=395,y=220)
    Button(log,text="'Forgot Password?

Change it for max.2 times"',font=("Comic Sans
MS",12),command=newlogin,height=2,width=23,bd=6,bg='ligh
t blue',fg='black').place(x=325,y=330)

    log.mainloop()


#FUNCTIONS TO CHANGE USERNAME/PASSWORD
def newlogin():
```

```
    #CREATING TOPLEVEL WINDOW AND ADDING
BACKGROUND IMAGE
    log=Toplevel()
    log.title("New Username and Password")
    log.geometry("266x300")
    img = PhotoImage(file="FORGOT.png",master=log)
    label = Label(log,image=img)
    label.place(x=0, y=0)

    #CREATING REQUIRED LABELS AND ENTRIES
    Label(log,text=" New Username: ",font=("Comic Sans
MS",11),bg="white",fg="maroon",borderwidth=1,
relief="solid").place(x=20,y=215)
    Label(log,text=" New Password: ",font=("Comic Sans
MS",11),bg="white",fg="maroon",borderwidth=1,
relief="solid").place(x=20,y=251)

    nusername=Entry(log,bd=3, width=10)
    nusername.place(x=170,y=216)

    npassword=Entry(log,bd=3,width=10)

    npassword.place(x=170,y=252)

    #SAVING THE NEW LOGIN DETAILS FOR FUTURE REFERENCE
    def NEWLOGIN():
        n1=open("Music.txt","a")
        User = nusername.get()
        n1.write(User)
        n1.close()
        n2=open("Music.txt","a")
        Pass = npassword.get()
```

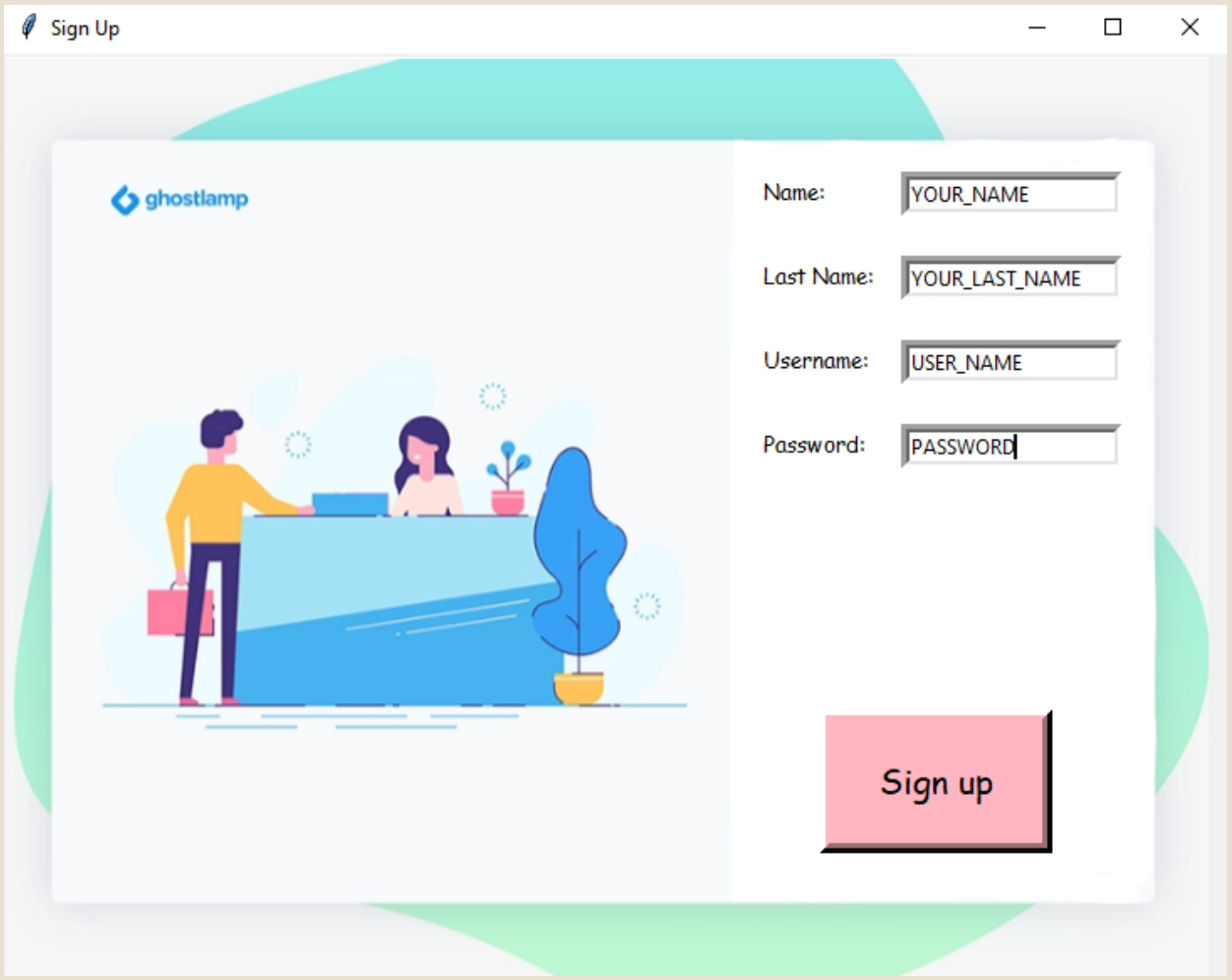
```
        n2.write("\n")
        n2.write(Pass)
        n2.close()
        n3=open("Music.txt","r")
        dd=n3.readlines()
        ll=[]
        for j in dd:
            ll.append(j.replace("\n",""))

        if len(ll)!=6:
            messagebox.showinfo("","Blank not allowed")
        else:
            messagebox.showinfo("","Your new login credentials
have been saved")
            log.withdraw()
    Button(log,text="Login",font=("Comic Sans
MS",12),command=NEWLOGIN,height=1,width=4,bd=4,bg='ligh
t blue',fg='black').place(x=100,y=110)
    log.mainloop()

#FUNCTION TO CHECK WHETHER THE USER HAS SIGNED IN
PREVIOUSLY OR IS NEW TO THE APPLICATION
def step1():
    f2=open("Music.txt","r")
    d=f2.read()
    if d!="":
        Lets_Login()
    else:
        Lets_Signup()
step1()
```

OUTPUT:

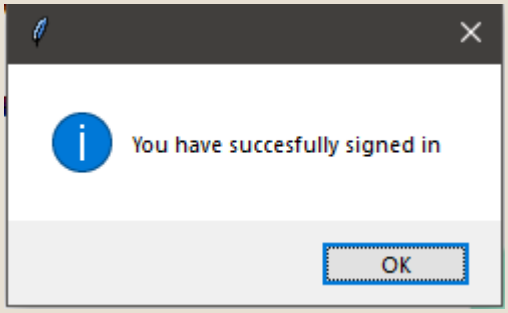
Empty text file before signing up:



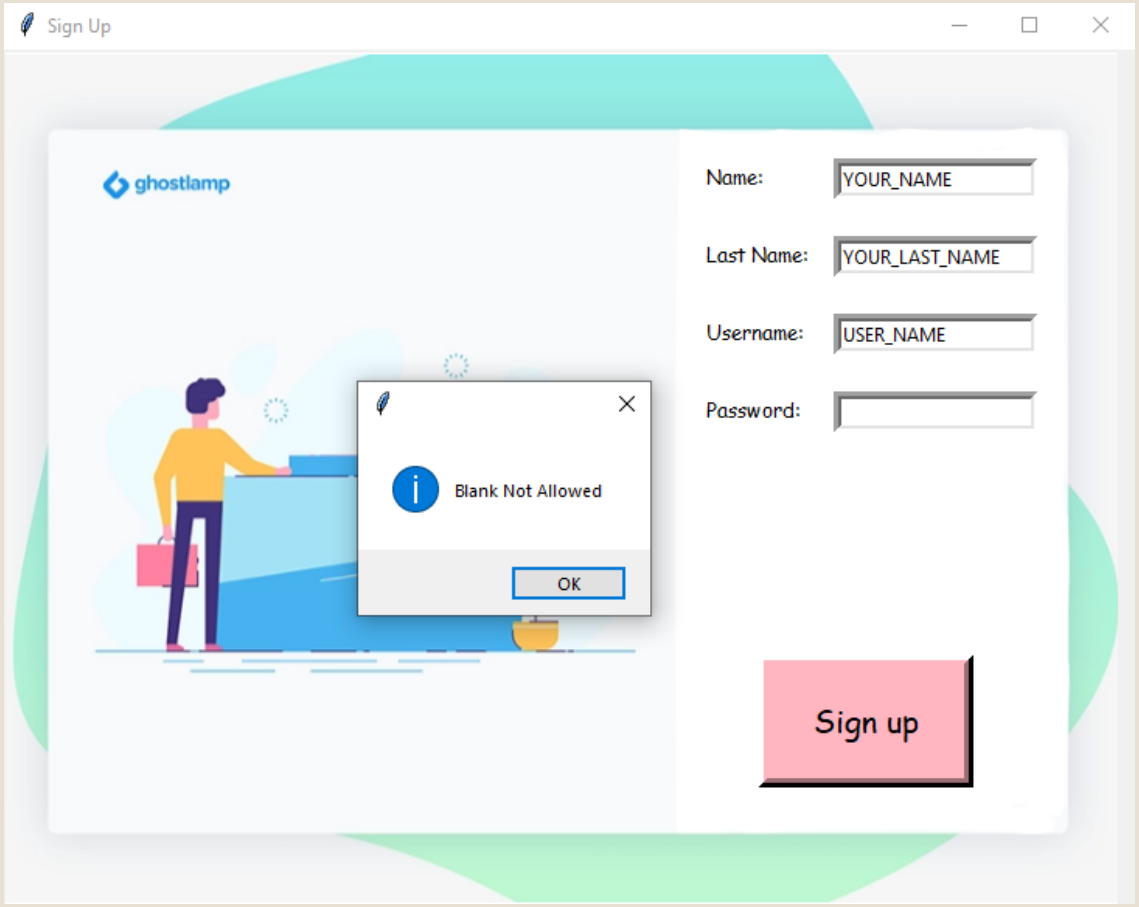
Sign Up page:



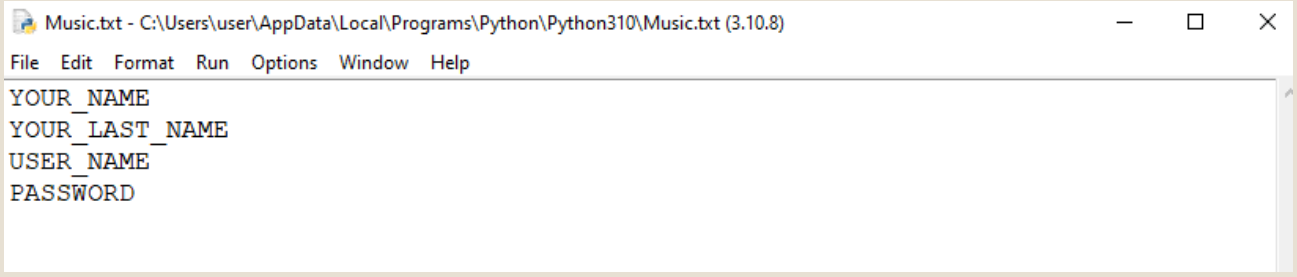
Successful sign up:



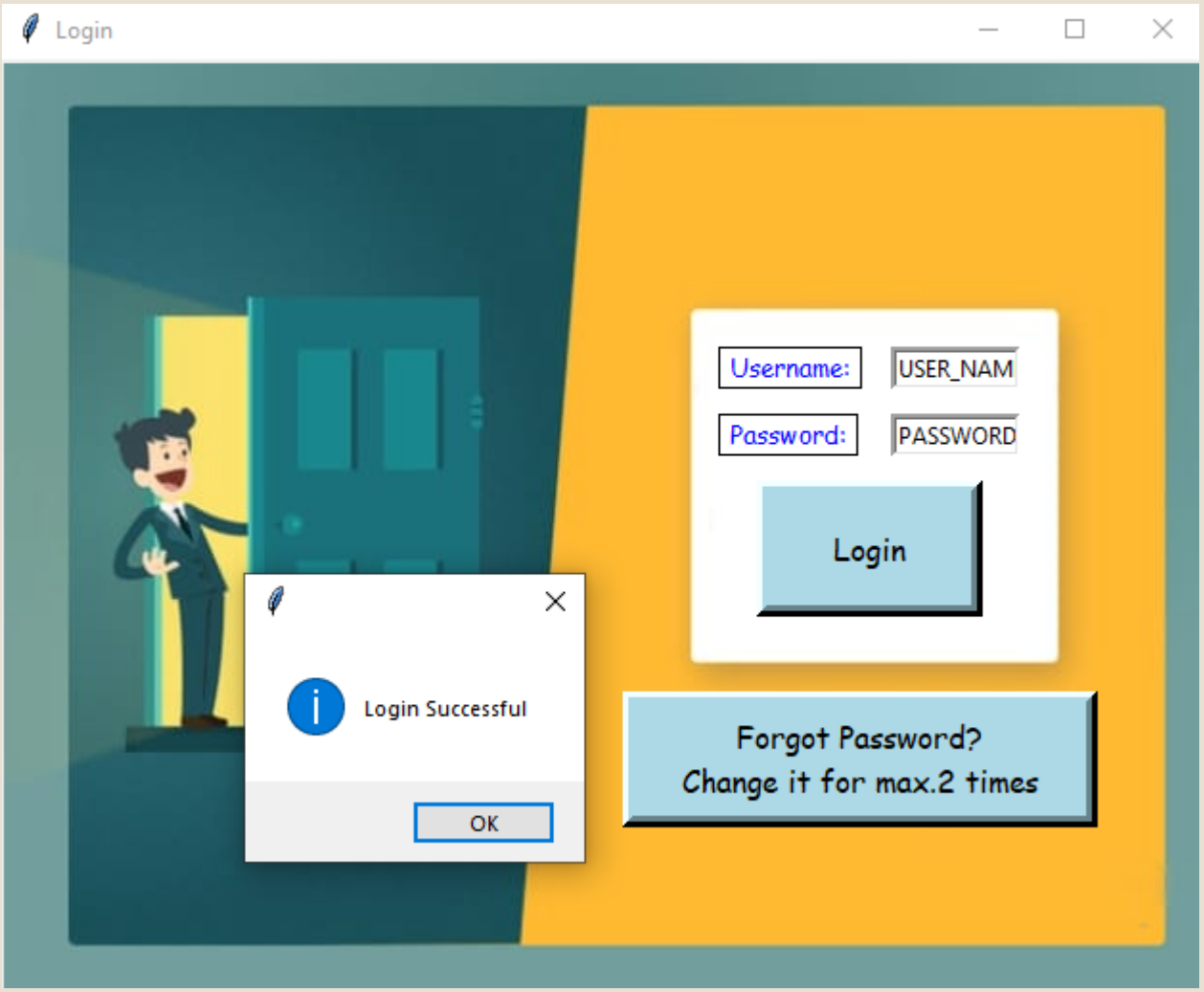
When the user keeps one input empty:



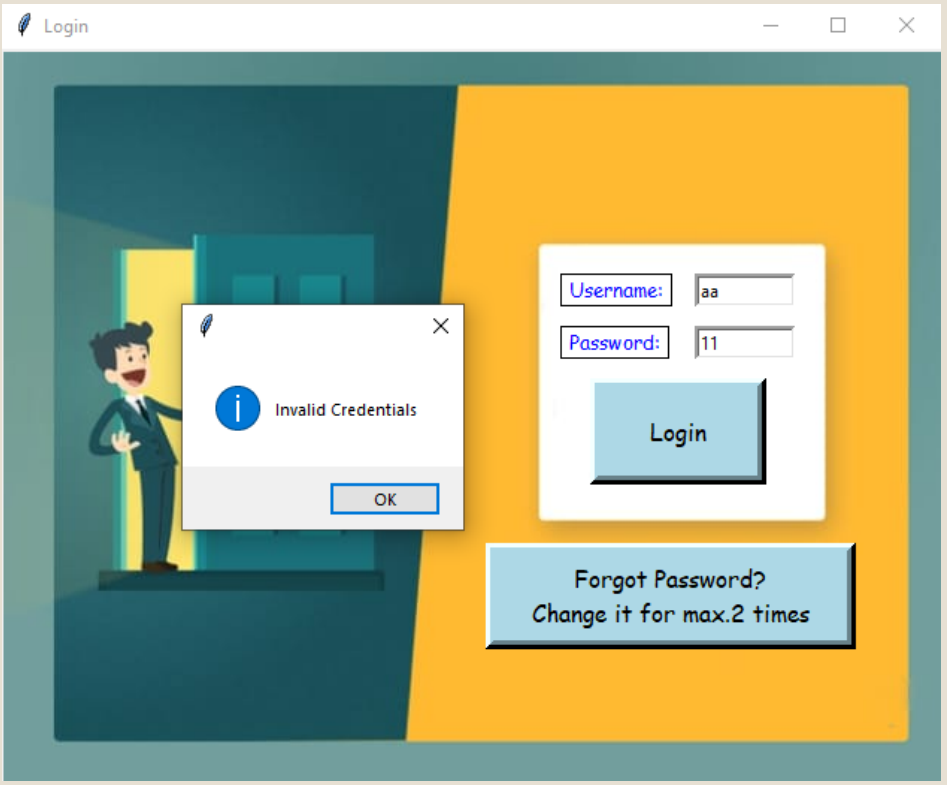
Text file after signing up:



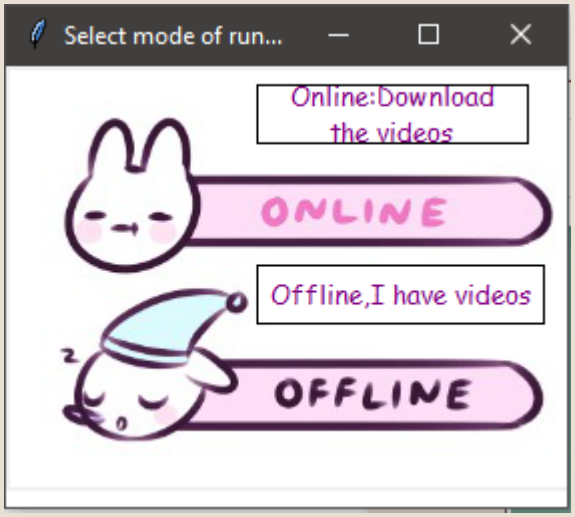
Login page with successful login:



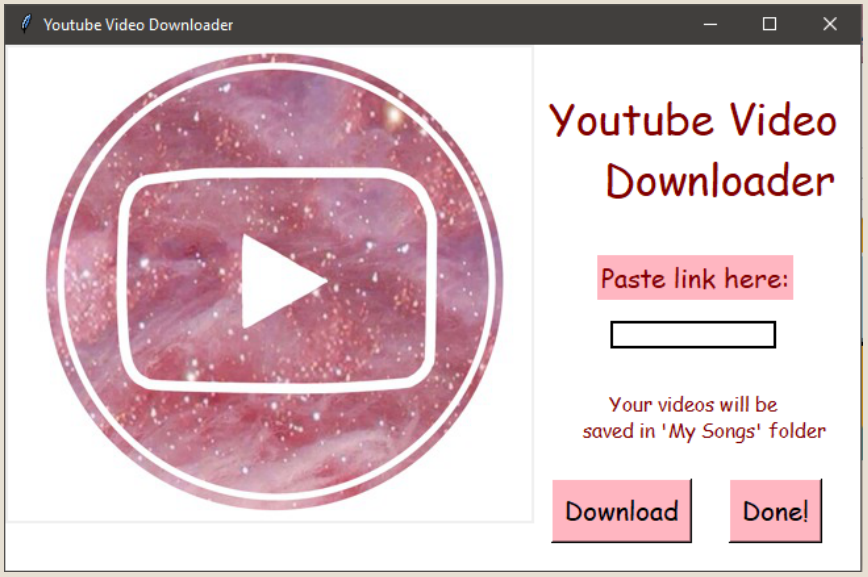
If the user inputs invalid credentials:

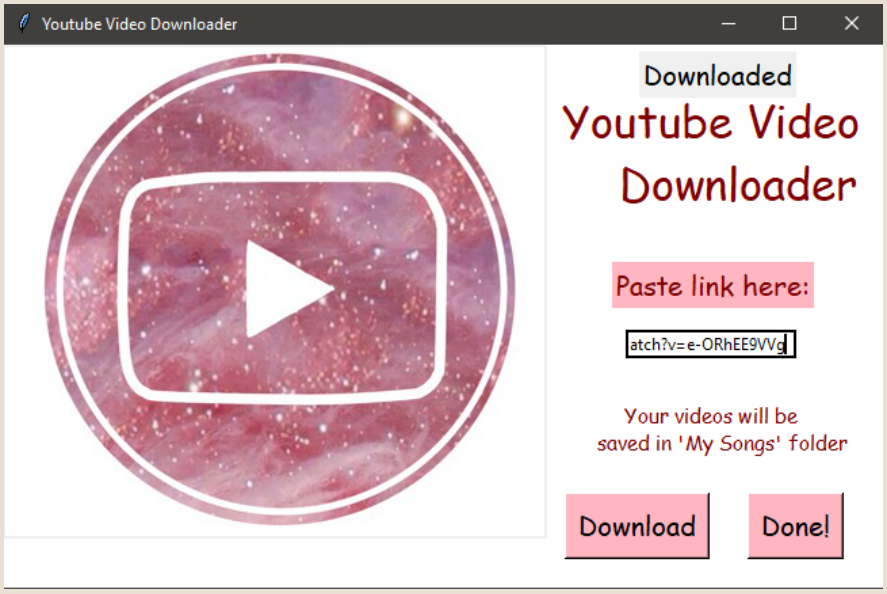


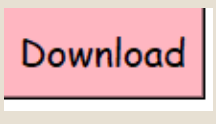
Selecting mode of running:

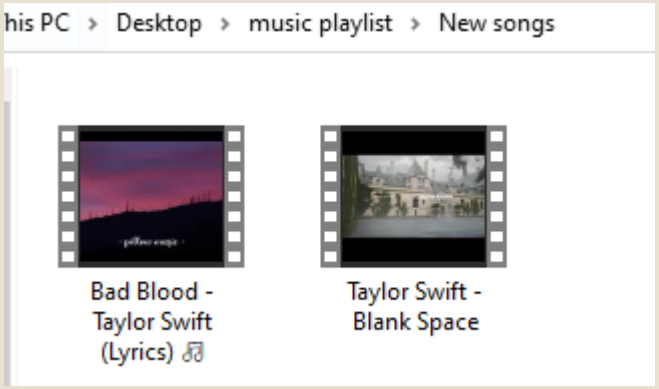


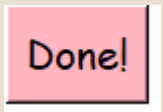
Online version: Download videos and then get directed to playlist

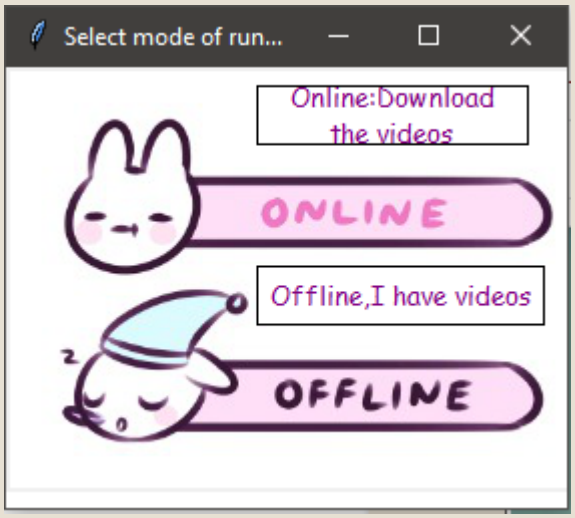




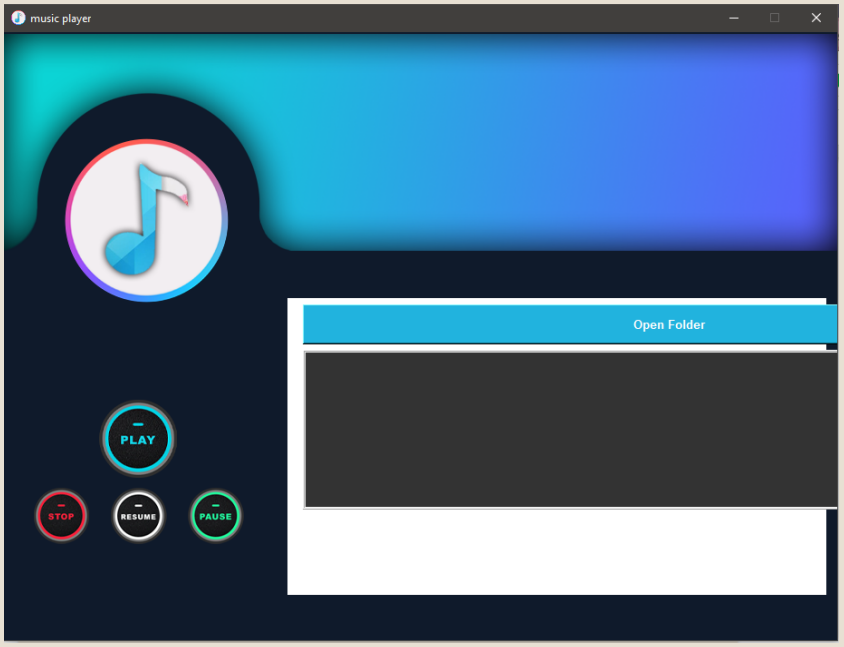




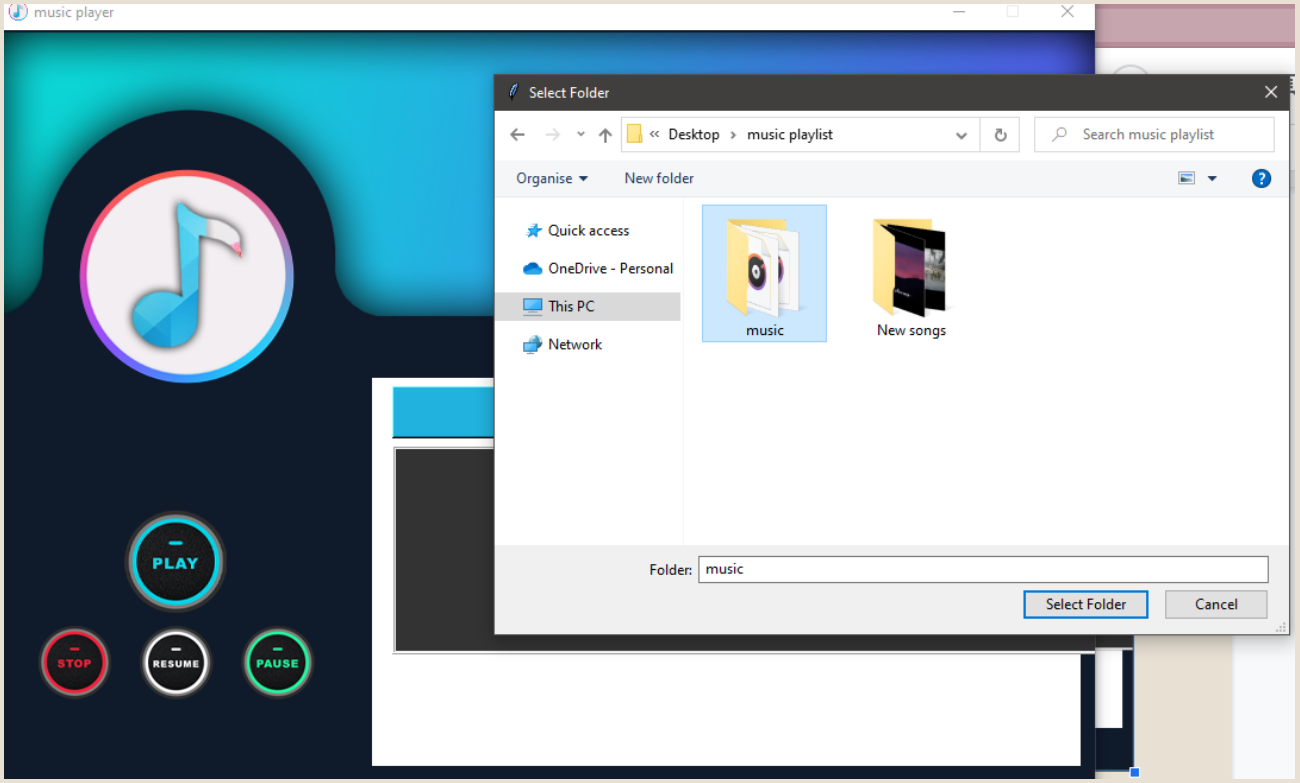




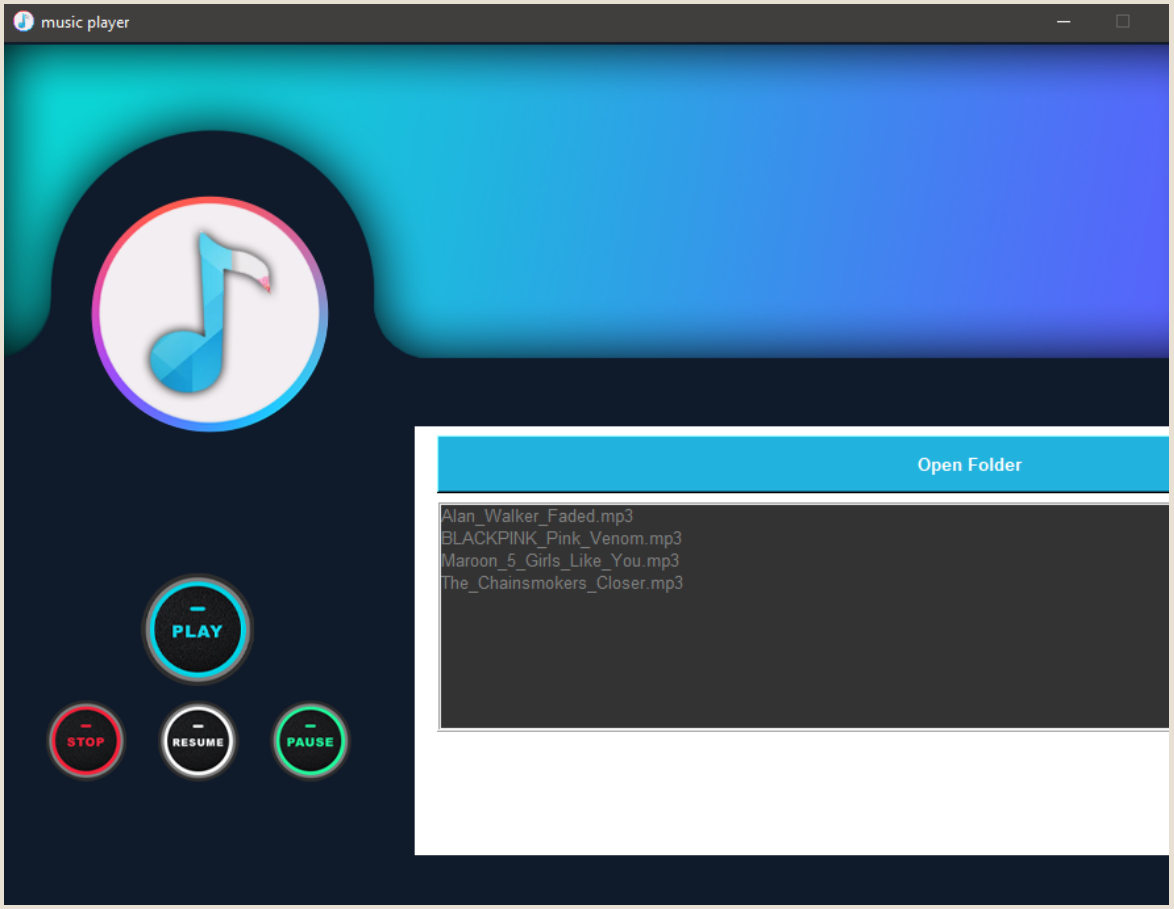
Offline mode:



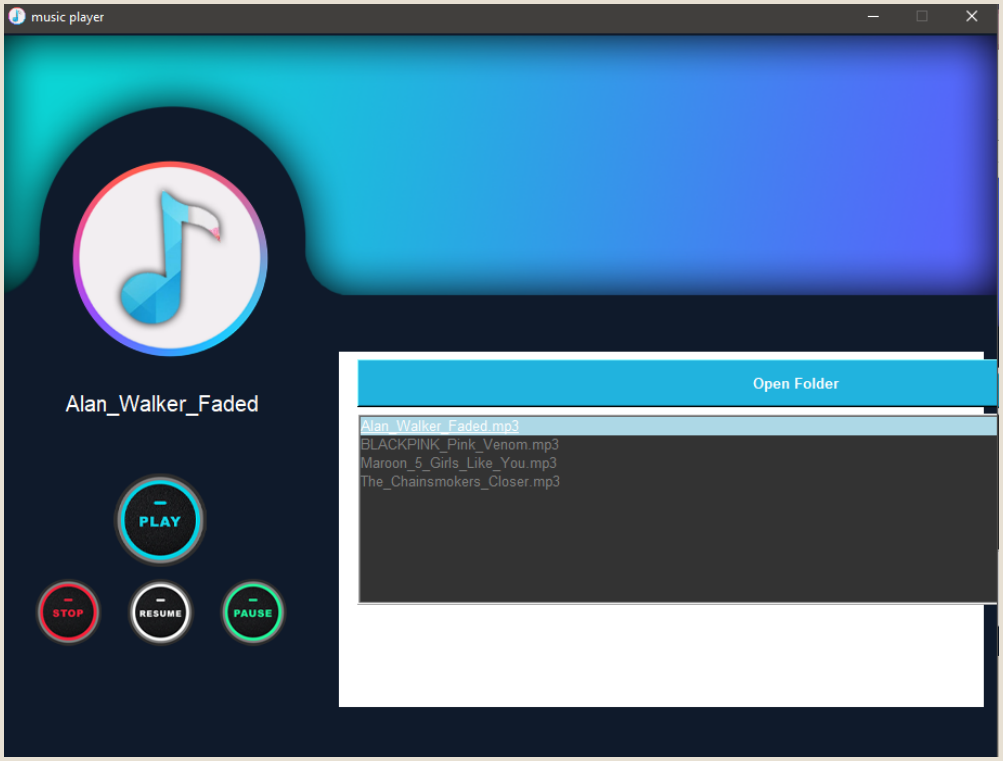
Add songs from your folder:



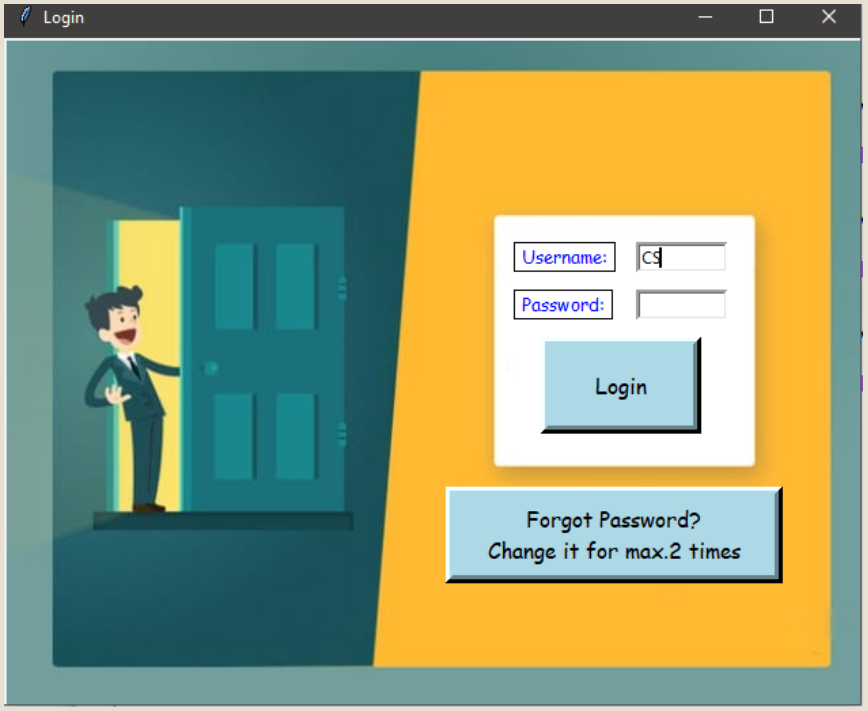
Songs added to the playlist:

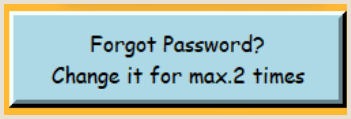


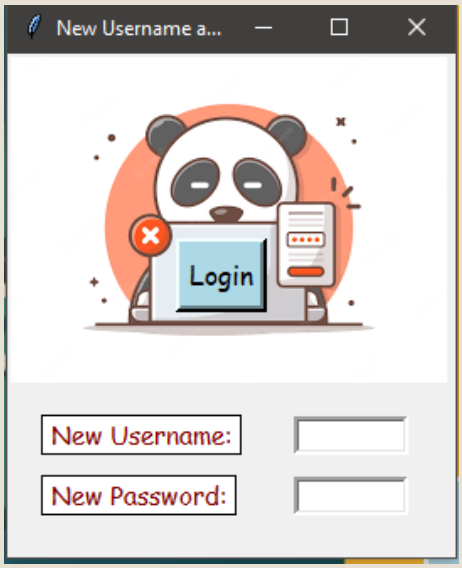
Play/Pause/Resume/Stop:

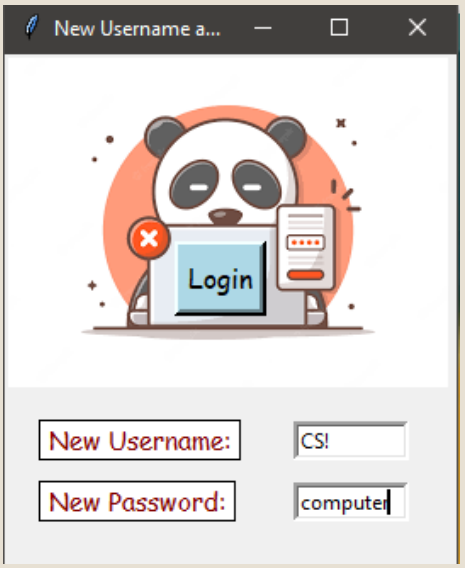


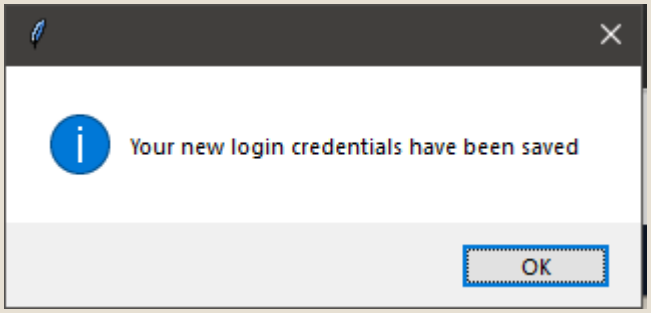
Forgot password Option:

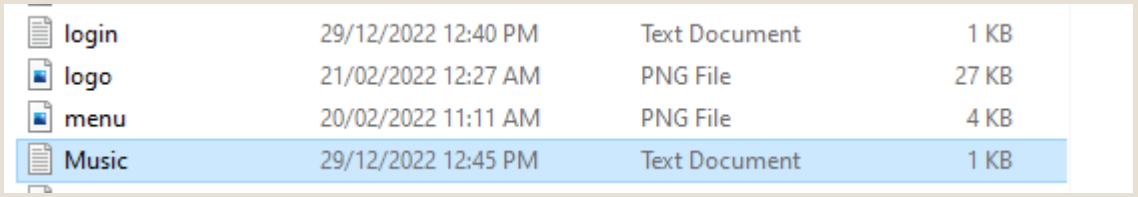




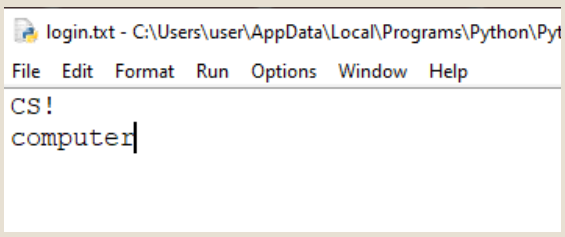


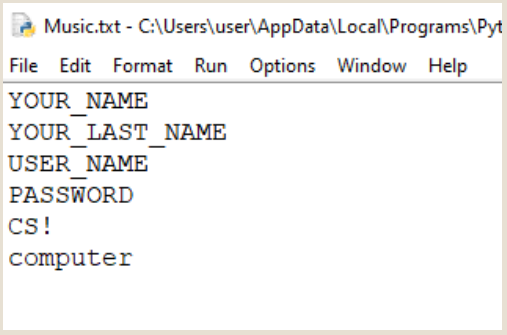






Files after changing password:





Advantages:

- No ads.
- Enhanced audio quality.
- No need to download another application, works well with python.
- Available offline
- Favorite songs at one place.
- Easy to download whichever song needed.
- No need to use unlicensed websites for downloading songs.
- Secured with login credentials.

Disadvantages:

- Doesn't sort music by genre.

Bibliography:

- https://stackoverflow.com/questions/tagged/tkinter
- https://realpython.com/python-gui-tkinter/


thank
you
LYNDSEY & MIKE