



Arab Academy for Science, Technology and Maritime Transport.
College of computing and information technology.

Structure of Programming (CS445)

Music Player

Team Members:

- Ali Salamouny 19107921
- Habiba Amr 19107385
- Nada Mohamed 19107229
- Zeiad Ahmed 19107161

Lecturer: Dr. Sherif Morsy

T.A: Eng. Ali Zeen ElDin

Description:

- Everybody loves listening to music wouldn't it be cool to have our very own mp3 music player. So, in this python project, we are going to create an mp3 player with the help of python and its libraries. We will create an mp3 music player in which we can play the song, pause it, resume it, and navigate from the current song to the next song as well as previous songs.

Libraries:

- **Time:** Python has a module named time to handle time-related tasks
- **Pygame:** Cross-platform set of Python modules that are used to create video games. It consists of computer graphics and sound libraries designed to be used with the Python programming language.
- **Tkinter:** Creating a GUI application
- **filedialog:** You can open a single file, a directory, save as file and much more
- **mutagen.mp3:** Mutagen is a Python module to handle audio metadata. It supports all audio files
- **tkinter.ttk:** provides access to the Tk themed widget set

Functions:

- **Play_Time():** get the time of the sound and when you move the slider the time changes also.

```
def play_time():
    if stopped:
        return

    current_time = pygame.mixer.music.get_pos() / 1000
    converted_current_time = time.strftime('%M:%S', time.gmtime(current_time))
    song = song_box.get(ACTIVE)
    song = f'C:/Users/Habiba/Desktop/Music Player/Songs/{song}.mp3'
    song_mut = MP3(song)

    global song_length
    song_length = song_mut.info.length
    converted_song_length = time.strftime('%M:%S', time.gmtime(song_length))

    current_time += 1
    if int(my_slider.get()) == int(current_time):
        status_bar.config(text=f'Time Elapsed: {converted_song_length}')
    elif paused:
        pass
    elif int(my_slider.get()) == int(current_time):
        slider_position = int(song_length)
        my_slider.config(to=slider_position, value=int(current_time))
    else:
        slider_position = int(song_length)
        my_slider.config(to=slider_position, value=int(my_slider.get()))
        converted_current_time = time.strftime('%M:%S', time.gmtime(my_slider.get()))
        status_bar.config(text=f'Time Elapsed: {converted_current_time} of {converted_song_length}')
        next_time = int(my_slider.get()) + 1
        my_slider.config(value=next_time)

    status_bar.after(1000, play_time)
```

- **Add_Song():** add song in the list.

```
def add_song():
    song = filedialog.askopenfilename(initialdir='Songs/', title="Choose a Song", filetypes=(("mp3 Files", "*.mp3"), ))
    song = song.replace("C:/Users/Habiba/Desktop/Music Player/Songs/", "")
    song = song.replace(".mp3", "")
    song_box.insert(END, song)
```

- **Add_Many_Songs():** add many songs in the list.

```
def add_many_song():
    songs = filedialog.askopenfilenames(initialdir='Songs/', title="Choose a Song", filetypes=(("mp3 Files", "*.mp3"), ))
    for song in songs:
        song = song.replace("C:/Users/Habiba/Desktop/Music Player/Songs/", "")
        song = song.replace(".mp3", "")
        song_box.insert(END, song)
```

- **Delete_Song():** delete one song from the list.

```
def delete_song():
    stop()
    song_box.delete(ANCHOR)
    pygame.mixer.music.stop()
```

- **Delete_All_Songs():** delete all the songs from the list.

```
def delete_all_songs():
    stop()
    song_box.delete(0, END)
    pygame.mixer.music.stop()
```

- **Play():** play the song.

```
def play():
    stopped = False
    song = song_box.get(ACTIVE)
    song = f'C:/Users/Habiba/Desktop/Music Player/Songs/{song}.mp3'
    pygame.mixer.music.load(song)
    pygame.mixer.music.play(loops=0)
    play_time()
```

- **Back():** go to the song before.

```
def back():
    back_one = song_box.curselection()
    back_one = back_one[0]-1
    song = song_box.get(back_one)
    song = f'C:/Users/Habiba/Desktop/Music Player/Songs/{song}.mp3'
    pygame.mixer.music.load(song)
    pygame.mixer.music.play(loops=0)
    song_box.selection_clear(0, END)
    song_box.activate(back_one)
    song_box.selection_set(back_one, last=None)
```

- **Next():** go to the song after.

```
def next():
    next_one = song_box.curselection()
    next_one = next_one[0]+1
    song = song_box.get(next_one)
    song = f'C:/Users/Habiba/Desktop/Music Player/Songs/{song}.mp3'
    pygame.mixer.music.load(song)
    pygame.mixer.music.play(loops=0)
    song_box.selection_clear(0, END)
    song_box.activate(next_one)
    song_box.selection_set(next_one, last=None)
```

- **Pause():** pause the song.

```
def pause(is_paused):
    global paused
    paused = is_paused

    if paused:
        pygame.mixer.music.unpause()
        paused = False
    else:
        pygame.mixer.music.pause()
        paused = True
```

- **Stop():** stop the song.

```
def stop():
    status_bar.config(text='')
    my_slider.config(value=0)
    pygame.mixer.music.stop()
    song_box.selection_clear(ACTIVE)
    status_bar.config(text='')
    global stopped
    stopped = True
```

- **Slider():** slider to move the song to specific minute.

```
def slider(X):
    song = song_box.get(ACTIVE)
    song = f'C:/Users/Habiba/Desktop/Music Player/Songs/{song}.mp3'
    pygame.mixer.music.load(song)
    pygame.mixer.music.play(loops=0, start=int(my_slider.get()))
```

- **Volume():** slider to volume up and volume down sound.

```
def volume(X):
    pygame.mixer.music.set_volume(volume_slider.get())
    current_volume = pygame.mixer.music.get_volume()
    current_volume = int(current_volume * 100)
    volume_label.config(text=current_volume)
```

GUI:

