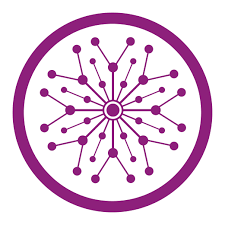
**

**The Superior University**

|  |  |  |
| --- | --- | --- |
| Name:MOHID NAGHMAN | Roll No:049 | Course: DSA Lab |
| Semester:3 | Section: 3A | Department: |
| Submitted To: | Total Marks: 10 | Date: |

**Lab Task 04**

**Circular Linked List**

**Implement any dynamic functionality in the music\_player.py code and do it with circular linked list.**

**Code:**

import pygame

import time

class Song:

    def \_\_init\_\_(self, title, path):

        self.title = title

        self.path = path

        self.next = None

class MusicPlayer:

    def \_\_init\_\_(self):

        pygame.mixer.init()

        self.head = None

        self.current = None

        self.n = 0

        self.is\_playing = False

    def add\_song(self, title, path):

        new\_song = Song(title, path)

        if self.head is None:

            # Empty list

            self.head = new\_song

            new\_song.next = self.head

        else:

            # Non-empty list, insert at the end

            curr = self.head

            while curr.next != self.head:

                curr = curr.next

            curr.next = new\_song

            new\_song.next = self.head

        if self.current is None:

            self.current = new\_song

        self.n += 1

        print(f"Song '{title}' added to the playlist.")

    def play(self):

        if self.current is None:

            print("No songs in the playlist.")

        else:

            print(f"Now playing: {self.current.title}")

            pygame.mixer.music.load(self.current.path)

            pygame.mixer.music.play()

            self.is\_playing = True

    def next\_song(self):

        if self.current is None:

            print("No songs in the playlist.")

        else:

            self.current = self.current.next

            self.play()

    def previous\_song(self):

        if self.current is None:

            print("No songs in the playlist.")

        else:

            prev = self.head

            while prev.next != self.current:

                prev = prev.next

            self.current = prev

            self.play()

    def stop(self):

        if self.is\_playing:

            pygame.mixer.music.stop()

            print(f"Stopped: {self.current.title}")

            self.is\_playing = False

        else:

            print("No song is currently playing.")

    def pause(self):

        if self.is\_playing:

            pygame.mixer.music.pause()

            print(f"Paused: {self.current.title}")

            self.is\_playing = False

        else:

            print("No song is currently playing.")

    def unpause(self):

        if not self.is\_playing:

            pygame.mixer.music.unpause()

            print(f"Resumed: {self.current.title}")

            self.is\_playing = True

    def display\_playlist(self):

        if self.head is None:

            print("Playlist is empty.")

            return

        print("Playlist:")

        curr = self.head

        while True:

            print(f"- {curr.title}")

            curr = curr.next

            if curr == self.head:

                break

    def remove\_song(self, title):

        if self.head is None:

            print("No songs in the playlist.")

            return

        if self.head.title == title and self.head.next == self.head:

            self.head = None

            self.current = None

            self.n = 0

            print(f"Song '{title}' removed. Playlist is empty now.")

            return

        curr = self.head

        prev = None

        while True:

            if curr.title == title:

                if prev is None:

                    tail = self.head

                    while tail.next != self.head:

                        tail = tail.next

                    self.head = curr.next

                    tail.next = self.head

                    self.current = self.head

                else:

                    prev.next = curr.next

                    if curr == self.current:

                        self.current = prev.next

                self.n -= 1

                print(f"Song '{title}' removed from the playlist.")

                return

            prev = curr

            curr = curr.next

            if curr == self.head:

                break

        print(f"Song '{title}' not found in the playlist.")

player = MusicPlayer()

# Add songs to the playlist with file paths

player.add\_song("Song A", "music\_files/triumphant-long-6673.mp3")

player.add\_song("Song B", "music\_files/pure-motivation-158726.mp3")

player.add\_song("Song C", "music\_files/triumphant-long-6673.mp3")

# Display playlist

player.display\_playlist()

# Play the current song

player.play()

time.sleep(60)

# # Next song

player.next\_song()

time.sleep(60)

# # Pause the song

player.pause()

time.sleep(5)

# # Unpause the song

player.unpause()

time.sleep(5)

# # Stop the song

player.stop()

