A purple text on a black background

AI-generated content may be incorrect.

**Project- Quran Player**

Name: Omar Afef Mohammed

Roll number: SU92-BSSEM-S24-086

Section: 3A

Teacher: Mr. Rasikh Ali

Subject: DSA LAB

Date: May 4, 2025

**CODE**

#include <iostream>

#include <windows.h>

#include <mmsystem.h>

#include <conio.h> // for \_kbhit() and \_getch()

#pragma comment(lib, "Winmm.lib")

using namespace std;

class Node {

public:

char filname[42];

Node\* next;

Node\* prev;

Node() {

next = NULL;

prev = NULL;

}

Node(Node\* prv) {

strcpy(filname, prv->filname);

prev = prv;

next = NULL;

}

};

class Quranlist {

public:

Node\* head;

Quranlist() {

head = new Node;

strcpy(head->filname, "C:\\Users\\OMAR\\Music\\Quran wav\\001.wav");

Node\* temp = head;

for (int i = 2; i <= 4; i++) {

Node\* newNode = new Node(temp);

temp->next = newNode;

sprintf(temp->next->filname, "C:\\Users\\OMAR\\Music\\Quran wav\\%03d.wav", i);

temp = temp->next;

}

temp->next = head;

head->prev = temp;

}

Node\* search() {

int num;

cout << "Enter the number of the surah: ";

cin >> num;

if (num < 1 || num > 4) {

cout << "Wrong number\n\n";

return NULL;

}

Node\* temp = head;

for (int i = 1; i < num; i++) {

temp = temp->next;

}

return temp;

}

void playsound() {

char returnString[200] = { 0 };

Node\* current = search();

if (!current) return;

string command = string("open \"") + current->filname + "\" type mpegvideo alias sound";

DWORD error = mciSendString(command.c\_str(), returnString, sizeof(returnString), NULL);

if (error != 0) {

char errorText[200];

mciGetErrorString(error, errorText, sizeof(errorText));

cout << "Failed to open file: " << errorText << endl;

return;

}

cout << "Surah opened successfully. Playing now..." << endl;

mciSendString("play sound", NULL, 0, NULL);

bool isPaused = false;

cout << "\nControls:\n";

cout << " Right Arrow : Next Surah\n";

cout << " Left Arrow : Previous Surah\n";

cout << " SPACE : Pause/Resume\n";

cout << " ESC : Exit\n\n";

while (true) {

if (\_kbhit()) {

int key = \_getch();

if (key == 32) { // Space bar

if (!isPaused) {

mciSendString("pause sound", NULL, 0, NULL);

cout << "Paused.\n";

isPaused = true;

} else {

mciSendString("resume sound", NULL, 0, NULL);

cout << "Resumed.\n";

isPaused = false;

}

} else if (key == 0 || key == 224) {

int arrow = \_getch();

if (arrow == 77) { // Right arrow key

current = current->next;

} else if (arrow == 75) { // Left arrow key

current = current->prev;

} else {

continue;

}

mciSendString("close sound", NULL, 0, NULL);

command = string("open \"") + current->filname + "\" type mpegvideo alias sound";

DWORD error = mciSendString(command.c\_str(), returnString, sizeof(returnString), NULL);

if (error != 0) {

char errorText[200];

mciGetErrorString(error, errorText, sizeof(errorText));

cout << "Failed to open file: " << errorText << endl;

return;

}

cout << "Surah changed. Playing now..." << endl;

mciSendString("play sound", NULL, 0, NULL);

isPaused = false;

} else if (key == 27) { // ESC key

mciSendString("close sound", NULL, 0, NULL);

cout << "Closed sound and exiting.\n";

break;

}

}

Sleep(100); // Reduce CPU usage

}

}

};

int main() {

Quranlist list;

list.playsound();

return 0;

}

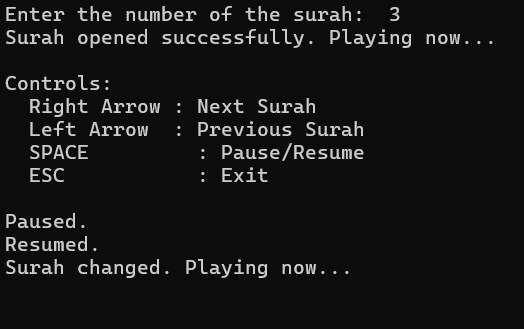
**How:**

* It builds a doubly circular linked list of 4 .wav Quran audio files.
* It allows user to select a Surah by number and plays it using Windows Media Control Interface (mciSendString).
* During playback, user can control audio using arrow keys (next/previous Surah), spacebar (pause/resume), and ESC (exit).

**Why:**

* Doubly linked list enables smooth navigation in both directions (next/previous Surahs).
* Circular structure ensures wrap-around navigation (e.g., after last Surah, it goes to first).
* mciSendString provides a simple way to control audio playback in Windows without external libraries.

**OUTPUT**

****