PIMPRI CHINCHWAD EDUCATION TRUST's.

**PIMPRI CHINCHWAD COLLEGE OF ENGINEERING**

(An Autonomous Institute)



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Class : SY BTech Acad. Yr. 2025-26 Semester : I**

**Name of the student: Varad Amol Pisale PRN : 124B1B043**

**Department: Computer Engineering Division : A**

**Course Name :** **Data Structures Laboratory Code:BCE23PC02**

**Completion Date : 11/08/2025**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Assignment No.**

Problem Statement:

Design a music playlist system using a linked list where:

● Songs can be added to the beginning/end

● Songs can be deleted

● Next and previous songs can be navigated

Source Code :

#include <iostream>

#include <string.h>

#include<cstdlib>

using namespace std;

class node

{

public:

    string data = "";

    node \*next = NULL;

    node() {}

    node(string dat);

};

node::node(string dat)

{

    data = dat;

}

class LL

{

private:

    node \*head = NULL;

    node \*tail = head;

public:

    void insert(string val, int op = 0)

    {

        // op=0 for add last and 1 for add head

        node \*temp = new node(val);

        if (head == NULL)

        {

            head = temp;

            tail = head;

        }

        else if (op == 1)

        {

            temp->next = head;

            head = temp;

        }

        else

        {

            tail->next = temp;

            tail = temp;

        }

    }

    void print()

    {

        node \*it = head;

        while (it != NULL)

        {

            cout << it->data << endl;

            it = it->next;

        }

    }

    void deleteany(string val)

    {

        node \*it = new node();

        node \*itminus1 = new node();

        it = head;

        itminus1 = head;

        while (it != NULL)

        {

            if (it->data == val)

            {

                if (it == head)

                {

                    head = it->next;

                    delete it;

                }

                else if (it == tail)

                {

                    tail = itminus1;

                    tail->next = NULL;

                    delete it;

                }

                else

                {

                    itminus1->next = it->next;

                    delete it;

                }

                break;

            }

            else

            {

                if (itminus1 == it)

                {

                    it = it->next;

                }

                else

                {

                    it = it->next;

                    itminus1 = itminus1->next;

                }

            }

        }

    }

    void navi(string val, int op)

    {

        node \*it = head;

        node \*itminus1 = head;

        // 0 for prev and 1 for next

        while (it != NULL)

        {

            if (it->data == val)

            {

                if (it == head)

                {

                    if (op == 0)

                    {

                        cout << "\nPrevious does not exist!!" << endl;

                    }

                    else

                    {

                        cout << it->next->data << endl;

                    }

                }

                else if (it == tail)

                {

                    if (op == 1)

                    {

                        cout << "\nNext does not exist!!" << endl;

                    }

                    else

                    {

                        cout << itminus1->data << endl;

                    }

                }

                else

                {

                    if (op == 1)

                    {

                        cout << it->next->data << endl;

                    }

                    else

                    {

                        cout << itminus1->data << endl;

                    }

                }

                break;

            }

            else

            {

                if (itminus1 == it)

                {

                    it = it->next;

                }

                else

                {

                    it = it->next;

                    itminus1 = itminus1->next;

                }

            }

        }

    }

};

int main()

{

    LL obj;

    int op;

    while (true)

    {

        cout << "Welcome to MP3 PLayer !!" << endl;

        cout << "1. To Insert Song" << endl;

        cout << "2. To Delete Song" << endl;

        cout << "3. To Navigate Next and Previous of given song" << endl;

        cout << "4. To EXIT" << endl;

        cout << "Your Choice :";

        cin >> op;

        if(op==4){

            cout<<"\nBYE BYE";

            break;

        }

        switch (op)

        {

        case 1:

        {

            string name;

            int n;

            cout << "Enter song name to insert: ";

            getchar();

            getline(cin,name);

            cout << "\nEnter 0 to add at end and 1 for beginning: ";

            cin >> n;

            obj.insert(name,n);

            obj.print();

            break;

        }

        case 2:

        {

            string name;

            int n;

            cout << "\nEnter song name to delete: ";

            getchar();

            getline(cin,name);

            obj.deleteany(name);

            obj.print();

            break;

        }

        case 3:

        {

            string name;

            int n;

            cout << "Enter song name to navigate: ";

            getchar();

            getline(cin,name);

            cout<<"\nEnter 0 to play prev and 1 for next: ";

            cin>>n;

            obj.navi(name,n);

            break;

        }

        default:

        cout<<"\nEnter valid option";

            break;

        }

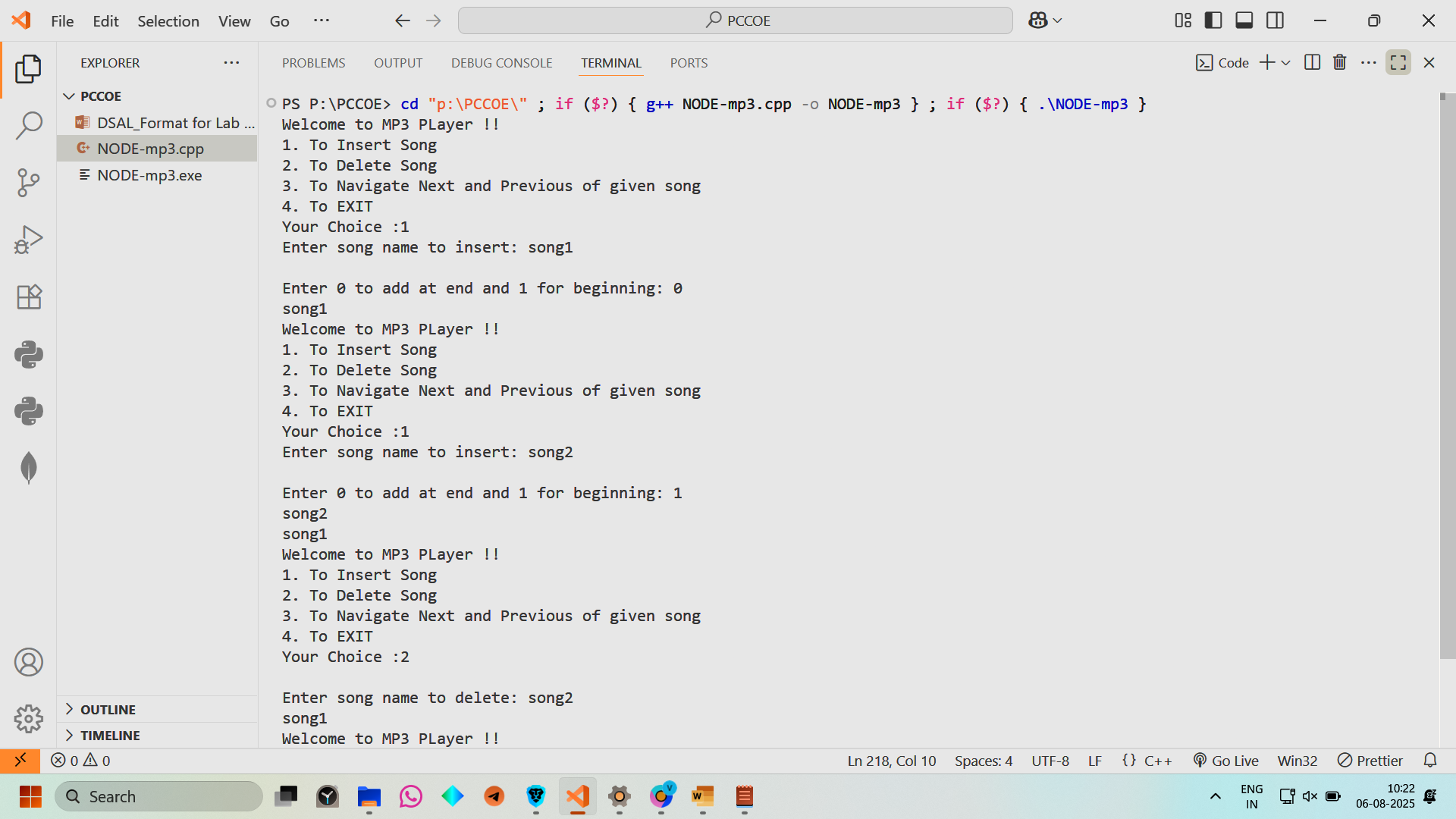
    }

    obj.print();

    return 0;

}

Screen Shot of Output :



Conclusion:

Thus, we have successfully implemented a program using Linked List that can insert,delete, play next/previous songs.