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: Hariom Shrikrishna Gundale **PRN: 124B1B036**

Department: Computer Engineering **Division: A**

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

Completion Date:

Assignment No. 4

Problem Statement: Design a music playlist system using a linked list where:

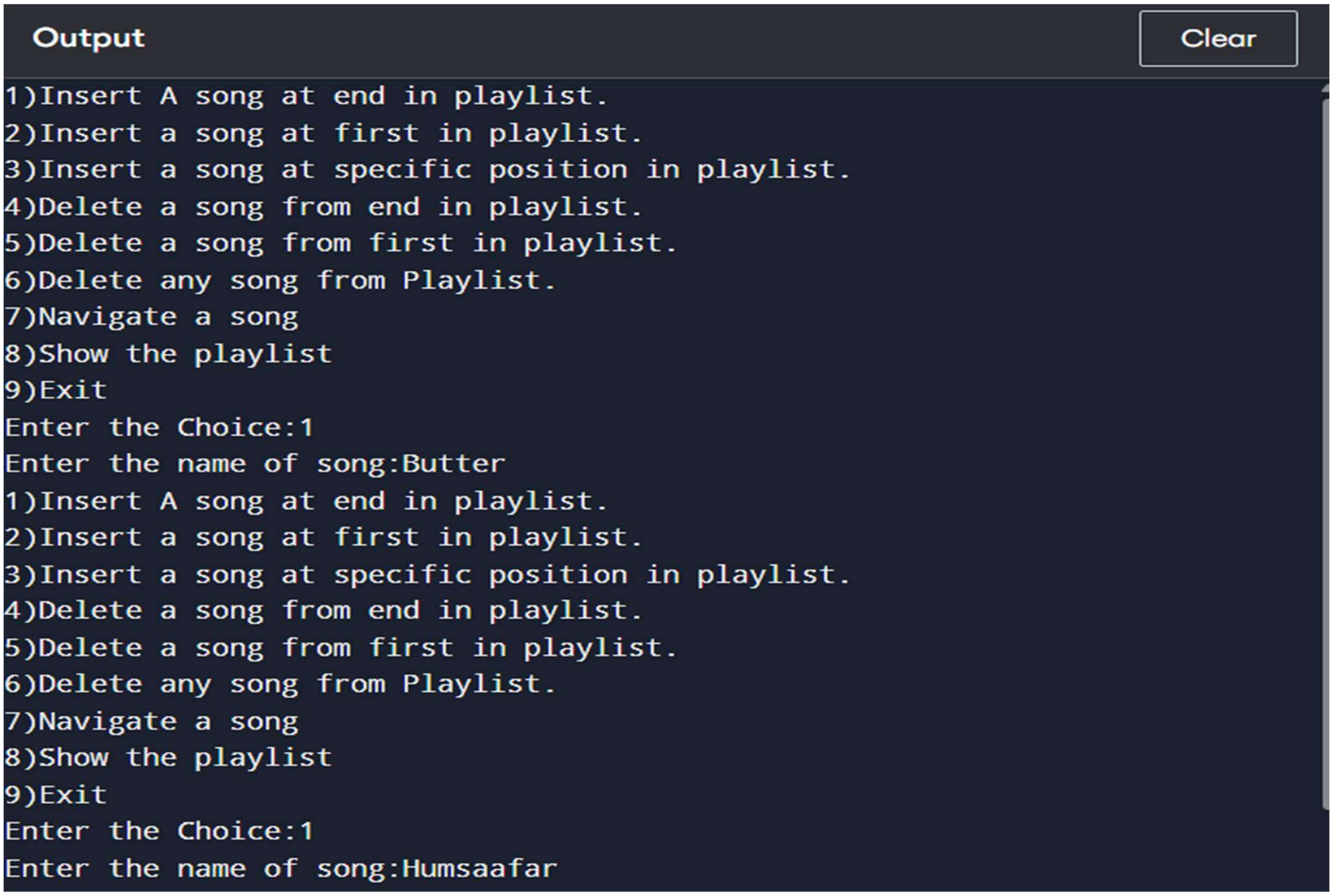
- Songs can be added to the beginning/end
- Songs can be deleted
- Next and previous songs can be navigated

Source Code:

```
#include <iostream>
using namespace std;

class node {
public:
  string data;
  node* next;

  node() {
    data = "0";
    next = NULL;
  }
```

```
};

class LL {
private:
    node* head;

public:
    LL() {
        head = NULL;
    }

    void insertAtEnd(string d) {
        node* nn = new node;
        nn->data = d;
        nn->next = NULL;

        if (head == NULL) {
            head = nn;
        } else {
            node* t = head;
            while (t->next)
                t = t->next;
            t->next = nn;
        }
    }

    void insertAtHead(string d) {
        node* nn = new node;
        nn->data = d;
        nn->next = head;
        head = nn;
    }

    void insertAtPos(int position, string d) {
        node* nn = new node;
        nn->data = d;
        nn->next = NULL;

        if (position == 1) {
            insertAtHead(d);
            return;
        }
        node* t = head;
        for (int i = 1; i < position - 1 && t != NULL; i++) {
            t = t->next;
        }
        if (t) {
            nn->next = t->next;
            t->next = nn;
        }
    }

    void deletefromfront(){
        if(head==NULL){
            cout<<"Linked List is empty.";
        }else{
            node* temp = head;
            head = head->next;
            delete temp;
        }
    }

    void deletefromend(){
        if(head==NULL){
            cout<<"Linked List is empty.";
        }else{
            node* temp=head;
            node* k=head;
            while(temp->next != NULL){
                k=temp;
                temp=temp->next;
            }

            k->next = NULL;
            delete temp;
        }
    }

    void deleteatpos(int d){
        int count=1;
        node* temp=head;
        node* k=head;
        while(count!=d){
            k=temp;
            temp=temp->next;
        }
        k->next=temp->next;
        delete temp;
    }

    void navigate(int d){
        if(head == NULL){
            cout<<"Linked list is empty";
        }else{
            int count=1;
            node* temp=head;
            node* k= head;
            node* t = temp->next;
            while(count != d){
                k=temp;
                temp=temp->next;
                t=t->next;
                count++;
            }
            if(temp == head){
                cout<<"Playlist start from Current"<<endl;
                cout<<"Current:"<<temp->data<<"\t"<<"Next:"<<t->data<<endl;
            }else if(temp->next == NULL){
                cout<<"Current:"<<temp->data<<"\t"<<"Previous:"<<k->data<<"\t"<<endl;
                cout<<"Current Song is end of playlist.";
            }else{
                cout<<"Current:"<<temp->data<<"\t"<<"Previous:"<<k->data<<"\t"<<"Next:"<<t->data<<endl;
            }
        }
    }

    void printList() {
        node* t = head;
        while (t) {
            cout << t->data << endl;
            t = t->next;
        }
    }

};

int main() {
    LL myList;
    int c,d;
    string song;
    while(c!=9){
        cout<<"1)Insert A song at end in playlist.\n2)Insert a song at first in playlist.\n3)Insert a song at specific position in playlist.\n4)Delete a song from end in playlist.\n5)Delete a song from first in playlist.\n6)Delete any song from Playlist.\n7)Navigate a song\n8)Show the playlist\n9)Exit\n";
        cout<<"Enter the Choice:";
        cin>>c;
        switch(c){
            case 1:cout<<"Enter the name of song:";
            cin>>song;
            myList.insertAtEnd(song);
            break;

            case 2:cout<<"Enter the name of song:";
            cin>>song;
            myList.insertAtHead(song);
            break;

            case 3:cout<<"Enter the name of song:";
            cin>>song;
            cout<<"Enter the position to be insert in playlist:";
            cin>>d;
            myList.insertAtPos(d,song);
            break;

            case 4:myList.deletefromend();
            break;

            case 5:myList.deletefromfront();
            break;

            case 6:cout<<"Enter the no of song:";
            cin>>d;
            myList.deleteatpos(d);
            break;

            case 7:cout<<"Enter the no of song:";
            cin>>d;
            myList.navigate(d);
            break;

            case 8:myList.printList();
            break;

            case 9:cout<<"Exiting the Playlist!"<<endl;
            break;

            default:cout<<"Enter the Valid choice";
            break;
        }
    }
    return 0;
}
```

Screen Shot of Output:



```
1)Insert A song at end in playlist.
2)Insert a song at first in playlist.
3)Insert a song at specific position in playlist.
4)Delete a song from end in playlist.
5)Delete a song from first in playlist.
6)Delete any song from Playlist.
7)Navigate a song
8)Show the playlist
9)Exit
Enter the Choice:1
Enter the name of song:Butter
1)Insert A song at end in playlist.
2)Insert a song at first in playlist.
3)Insert a song at specific position in playlist.
4)Delete a song from end in playlist.
5)Delete a song from first in playlist.
6)Delete any song from Playlist.
7)Navigate a song
8)Show the playlist
9)Exit
Enter the Choice:1
Enter the name of song:Humsaafar
```

Conclusion: Thus, we have successfully implemented the C++ program to build a music playlist using the Data Structure Linked List. Using above code we can add a song, delete a existing song and navigate a playlist.