A Project report on

MUSIC PLAYLIST MANAGER

1. Introduction

This Project describes a Simple Music Playlist Manager with functionalities like adding tracks, removing tracks, and displaying the playlist based on mood or location.
We Design and implement an efficient music playlist manager in C that allows users to organize, manipulate, and interact with their music collection. The program should employ a doubly linked list data structure to represent the playlist, with each node containing information about a specific music track, including details such as title, artist, mood, and location.

2. Process

This C code implements a Simple Music Playlist Manager with functionalities like adding tracks, removing tracks, and displaying the playlist based on mood or location. Here's a brief overview of the code:

- The struct Track defines a structure for a music track with fields for title, artist, mood, location, and pointers to the next and previous tracks in the playlist.
- The global variable playlistHead is a pointer to the head of the playlist.
- Functions are defined for adding a track (addTrack), displaying the entire playlist (displayPlaylist), displaying the playlist based on mood (displayPlaylistByMood), displaying the playlist based on location (displayPlaylistByLocation), removing a track (removeTrack), adding predefined songs (addPredefinedSongs), and cleaning up memory (cleanUp).
- The main function serves as the program's entry point and provides a simple text-based menu for the user to interact with the playlist manager.
- The program initializes with some predefined songs using the addPredefinedSongs function.

- The main menu allows the user to perform actions such as adding a track, removing a track, displaying the playlist, displaying the playlist based on mood or location, and quitting the program.
- The program uses a loop to repeatedly prompt the user for input until the user chooses to quit.

Overall, this program is a basic implementation of a music playlist manager in C. It uses a doubly linked list to represent the playlist, and the user interacts with the program through a text-based menu in the console.

3. Code

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct Track {
    char title[100];
    char artist[100];
    char mood[20];
    char location[20];
    struct Track* next;
    struct Track* prev;
};

struct Track* playlistHead = NULL;

void addTrack(char title[], char artist[], char mood[], char location[]) {
    struct Track* newTrack = (struct Track*)malloc(sizeof(struct Track));
    if (newTrack == NULL) {
        printf("Memory allocation failed.\n");
        return;
    }
```

```
    strcpy(newTrack->title, title); strcpy(newTrack-
    >artist, artist);
    strcpy(newTrack->mood, mood);
    strcpy(newTrack->location, location);
    newTrack->next = NULL; newTrack-
    >prev = NULL;

    if (playlistHead == NULL) {
        playlistHead = newTrack;
    } else { newTrack->next =
        playlistHead; playlistHead-
        >prev = newTrack; playlistHead
        = newTrack;
    }

    printf("Added track: %s by %s to %s playlist\n", title, artist, mood);
}

void displayPlaylist() { struct Track*
    current = playlistHead; if (current
    == NULL) {
        printf("Playlist is empty.\n");
        return;
    }

    printf("Current Playlist:\n");
    while (current != NULL) {
        printf("%s by %s - Mood: %s - Location: %s\n", current->title, current->artist,
```

```
current->mood, current->location);
        current = current->next;
    }
}

void displayPlaylistByMood(char mood[]) {
    struct Track* current = playlistHead;
    int found = 0;

    if (current == NULL) {
        printf("Playlist is empty.\n");
        return;
    }

    printf("Playlist for mood %s:\n", mood);
    while (current != NULL) { if
    (strcmp(current->mood, mood) == 0) {
            printf("%s by %s - Mood: %s - Location: %s\n", current->title, current->artist,
current->mood, current->location);
            found = 1;
        }
        current = current->next;
    }

    if (!found) {
        printf("No tracks found for mood %s\n", mood);
    }
}
```

```
void displayPlaylistByLocation(char location[])
   { struct Track* current = playlistHead; int
   found = 0;

   if (current == NULL) {
      printf("Playlist is empty.\n");
      return;
   }

   printf("Playlist for location %s:\n", location);
   while (current != NULL) {
      if (strcmp(current->location, location) == 0) {
         printf("%s by %s - Mood: %s - Location: %s\n", current->title, current->artist,
current->mood, current->location);
         found = 1;
      }
      current = current->next;
   }

   if (!found) { printf("No tracks found for location %s\n",
      location);
   }
}

void removeTrack(char title[]) { struct
   Track* current = playlistHead;
   struct Track* temp = NULL;
   int found = 0;
```

```
if (current == NULL) {
    printf("Playlist is empty.\n");
    return;
}

while (current != NULL) {
    if (strcmp(current->title, title) == 0) {
        if (current->prev == NULL) {
            playlistHead = current->next;
            if (playlistHead != NULL) {
            playlistHead->prev = NULL;
            }
        } else { current->prev->next = current-
            >next; if (current->next != NULL) {
            current->next->prev = current->prev;
            }
        }
        temp = current; current
        = current->next;
        free(temp);
        found = 1;
        printf("Track removed: %s\n", title);
    } else { current = current-
        >next;
    }
}
```

```
    if (!found) { printf("Track not found:
        %s\n", title);
    }
}

void addPredefinedSongs() { addTrack("Hanuman chalisa","Shankar
    Mahadevan","Devotional","Temple"); addTrack("Raayini matram", "Hari
    Haran", "Devotional", "Temple"); addTrack("Sada Siva", "Karunya",
    "Devotional", "Temple"); addTrack("Swagatham Krishna", "Ramanan",
    "Devotional", "Temple");


    addTrack("Gallo Telinattunde", "DSP", "happy vibes", "Travel");
    addTrack("Hoyna Hoyna", "Anirudh Ravichander","happy vibes","Travel");
    addTrack("Arere Manasa", "Sid Sriram", "happy vibes", "Travel");
    addTrack("Neeve","Anuraag Kulkarni", "happy vibes", "Travel");


    addTrack("Meghaalu Lekunna", "DSP", "romantic", "Home");
    addTrack("Adiye", "Artist7", "romantic", "Home");
    addTrack("Undiporaadhey", "Artist7", "romantic", "Home");
    addTrack("Inthandam", "Artist7", "romantic", "Home");


    addTrack("Telisene", "LV Revanth", "Broken", "Home");
    addTrack("Kannula Basalu", "Karthick", "Broken", "Home");
    addTrack("Agar tum saath ho", "Arjith singh", "Broken", "Home");
```

```
    addTrack("Vinave Vinave", "GV Prakash", "Broken", "Home");
    addTrack("Adiga Adiga", "Sid Sriram", "Broken", "Home");


    addTrack("Jennifer Lopez", "Benny Dayal", "Mass Beats", "Party");
    addTrack("Dum Masala", "Sanjith Hegde", "Mass Beats", "Party");
    addTrack("Ringa Ringaa", "DSP", "Mass Beats", "Party");
    addTrack("Seeti Maar", "DSP", "Mass Beats", "Party");

}
void cleanUp() { struct Track* current
    = playlistHead; struct Track* next;

    while (current != NULL) {
        next = current->next;
        free(current);
        current = next;
    }

    playlistHead = NULL; printf("Memory
    freed. Exiting program.\n");
}

int main() {
    char option;
    char title[100];
    char artist[100];
    char mood[20];
    char location[20];
```

```
addPredefinedSongs();

do {
    printf("\nMenu:\n"); printf("A: Add a track to the
    playlist\n"); printf("R: Remove a track from the playlist\n");
    printf("D: Display the entire playlist\n"); printf("M: Display
    the playlist based on a specific mood\n"); printf("L: Display
    the playlist based on a specific location\n"); printf("Q:
    Quit\n"); printf("Enter your choice: "); scanf(" %c",
    &option);

    switch (option) {
        case 'A':
            printf("Enter track title: ");
            scanf(" %[^\n]", title);
            printf("Enter artist: ");
            scanf(" %[^\n]", artist);
            printf("Enter mood: ");
            scanf(" %[^\n]", mood);
            printf("Enter location: "); scanf("
        %[^\n]", location); addTrack(title,
        artist, mood, location); break; case 'R':
            printf("Enter track title to remove:
        "); scanf(" %[^\n]", title);
        removeTrack(title); break; case 'D':
        displayPlaylist(); break; case 'M':
            printf("Enter mood to display playlist: ");
            scanf(" %[^\n]", mood);
            displayPlaylistByMood(mood);
```

```
            break;
        case 'L':
            printf("Enter location to display playlist:
        "); scanf(" %[^\n]", location);
        displayPlaylistByLocation(location); break;
        case 'Q': cleanUp(); break; default:
            printf("Invalid option. Please try again.\n");
      }
   } while (option != 'Q');

   return 0;
}
```

4. Results

Output Clear

```
/tmp/JCPx0OuFQt.o
Added track: Hanuman chalisa by Shankar Mahadevan to Devotional playlist
Added track: Raayini matram by Hari Haran to Devotional playlist
Added track: Sada Siva by Karunya to Devotional playlist
Added track: Swagatham Krishna by Ramanan to Devotional playlist
Added track: Gallo telinattunde by DSP to happy vibes playlist
Added track: Hoyna Hoyna by Anirudh Ravichander to happy vibes playlist
Added track: Arere Manasa by Sid Sriram to happy vibes playlist
Added track: Neeve by Anuraag Kulkarni to happy vibes playlist
Added track: Meghaalu Lekunna by DSP to romantic playlist
Added track: Adiye by Artist7 to romantic playlist
Added track: Undiporaadhey by Artist7 to romantic playlist
Added track: Inthandam by Artist7 to romantic playlist
Added track: Teliseney by LV Revanth to Broken playlist
Added track: Kannula Basalu by Karthick to Broken playlist
Added track: Agar tum saath ho by Arjith singh to Broken playlist
Added track: Vinave Vinave by GV Prakash to Broken playlist
Added track: Adiga Adiga by Sid Sriram to Broken playlist
Added track: Jennifer Lopez by Benni Dayal to Mass Beats playlist
Added track: Dum Masala by Sanjith Hegde to Mass Beats playlist
Added track: Ringa Ringaa by DSP to Mass Beats playlist
Added track: Seeti Maar by DSP to Mass Beats playlist
```

```
Menu:
A: Add a track to the playlist
R: Remove a track from the playlist
D: Display the entire playlist
M: Display the playlist based on a specific mood
L: Display the playlist based on a specific location
Q: Quit
Enter your choice: M
Enter mood to display playlist: Broken
Playlist for mood Broken:
Adiga Adiga by Sid Sriram - Mood: Broken - Location: Home
Vinave Vinave by GV Prakash - Mood: Broken - Location: Home
Agar tum saath ho by Arjith singh - Mood: Broken - Location: Home
Kannula Basalu by Karthick - Mood: Broken - Location: Home
Teliseney by LV Revanth - Mood: Broken - Location: Home

Menu:
A: Add a track to the playlist
R: Remove a track from the playlist
D: Display the entire playlist
M: Display the playlist based on a specific mood
L: Display the playlist based on a specific location
Q: Quit
Enter your choice: Q
Memory freed. Exiting program.
```

5. Conclusion

In conclusion, the provided code offers a foundation for a music playlist manager in C. Further development could include refining existing features, adding new functionalities, and enhancing the user interface for a more comprehensive and user-friendly experience.

- This project allows users to customise the playlists according to their choices based on their Mood and Locations.

THANKYOU