# FINAL SUBMISSION REPORT

**Introduction:**

This code is a simple IPL management system, developed using the C programming language. The program uses file handling to store and manage data about IPL teams and their schedules.

The main features of this program are:

1. Display Existing Teams
2. Create Team
3. Add Team
4. View all Created Teams
5. Make a Schedule
6. Update Schedule
7. Delete Team

The program uses a menu-driven approach to allow the user to select the desired module. The menu is displayed repeatedly until the user chooses to exit the program.

## MODULES

1. Display existing teams: This module simply reads the names of existing IPL teams from a file and displays them on the console. The program opens the file in read mode using fopen() function and then reads the data line by line using fgets() function. The team names are then printed on the console using printf() function.
2. Create new teams: This module allows the user to create new IPL teams by entering their names. The user is prompted to enter the number of teams they want to create, and then the program reads the team names entered by the user and writes them to a file using fprintf() function.

3. Add team: This module allows the user to add a new team to the existing list of IPL teams. The user is prompted to enter the name of the team they want to add, and then the program opens the relevant file in append mode using fopen() function and writes the team name to the end of the file using fwrite() function.
4. View all teams: This module is similar to module 1 but displays all teams including newly added ones. It reads the names of all IPL teams from a file and then prints them to the console using printf() function.
5. Make a schedule: This module generates a schedule of matches between the IPL teams based on the start date entered by the user. The program prompts the user to enter the start date, number of teams, and the names of the teams. Then it uses nested loops to generate all possible match combinations and prints them to the console using printf() function and writes them to a file using fprintf() function.
6. View schedule: This module reads the schedule of matches generated by module 5 from a file and displays it on the console using printf() function.
7. Update the schedule: This module updates the schedule of matches based on the start date entered by the user. It uses the same logic as module 5 to generate the updated schedule and overwrites the previous schedule file with the new one using fopen() function and fprintf() function.
8. Delete team: This module allows the user to delete a team from the existing list of IPL teams. The user is prompted to enter the name of the team they want to delete, then it reads the names of all IPL teams from a file, deletes the specified team, and saves the updated team list to a new file using fopen() function, fclose() function, fgets() function,

strcmp() function, fprintf() function, remove() function, and rename() function.

All modules use file handling to store and manage data about IPL teams and their schedules. The program uses separate files for storing team names and schedules, and it reads from or writes to these files as required by each module. The use of file handling allows the program to store data even after it is closed and reopened, making it a useful tool for managing IPL teams and scheduling their matches.

# Conclusion:

In conclusion, the program developed for managing IPL teams has been successfully completed and tested. The program is designed to add, delete, and display the teams along with creating and updating schedules. It also utilizes file handling in C programming language to store and retrieve information about the teams and schedules.

Through the use of loops, conditional statements, and input validation techniques, the program provides efficient and error-free execution. However, there are still some enhancements that can be implemented to further improve the program's performance and user experience.

Overall, this program has the potential to streamline IPL team management, thereby saving time and effort. It could serve as a useful tool for coaches, players, or anyone who wants to stay updated on IPL team matches and schedules.

# Source Code:

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
struct team
{
    char T_name[60];
    int Num_players;
    char player_names[11][60];
} T;
char d[50] = "No Data Found";
void add_team()
{
    printf("Enter The team Name: ");
    scanf("%s", T.T_name);
    FILE *fp;
    fp = fopen("E:\\Mine Code\\C Files\\IPL Teams.txt", "a");
    fwrite(&T, sizeof(T), 1, fp);
    fclose(fp);
}
void delete_team()
{

    char team[50];
    printf("Enter the team You want To Delete: ");
    scanf("%s", team);
    FILE *f;
    FILE *fp;
    f = fopen("E:\\Mine Code\\C Files\\IPL Teams.txt", "r");
    fp = fopen("E:\\Mine Code\\C Files\\tempTeams.txt", "w");
    char buffer[1000];
    while (fgets(buffer, sizeof(buffer), f))
    {

        int len = strlen(buffer);
        if (len > 0 && buffer[len - 1] == '\n')
        {
            buffer[len - 1] = '\0';
        }

        // Check if line contains team to be removed
        if (strcmp(buffer, team) != 0)
        {
            fprintf(fp, "%s\n", buffer);
```

```
        }
    }

    fclose(f);
    fclose(fp);
    remove("E:\\Mine Code\\C Files\\IPL Teams.txt");
    rename("E:\\Mine Code\\C Files\\tempTeams.txt", "E:\\Mine Code\\C Files\\IPL
Teams.txt");
}
void making_schedule()

{
    int d, m, y, N;
    printf("Enter the Start Date(dd mm yyyy): ");
    scanf("%d %d %d", &d, &m, &y);
    int n, n2 = 0;
    FILE *fp;
    fp = fopen("E:\\Mine Code\\C Files\\schedule.txt", "w");
    printf("Enter The number of Teams: ");
    scanf("%d", &n);
    char Team_names[n][50];
    char Schedule[n2];
    for (int i = n - 1; i > 0; i--)
    {
        n2 += i;
    }
    printf("Number of Matches: %d\n", n2);

    for (int i = 0; i < n; i++)
    {
        printf("Enter Team %d: ", i + 1);
        scanf("%s", &Team_names[i]);
    }
    for (int i = 0; i < n; i++)
    {
        for (int j = i + 1; j < n; j++)
        {
            if (d == 31)
            {
                d = 0;
                m = m + 1;
            }
            if (d == 30)
            {
                if (m == 4 || m == 6 || m == 9 || m == 11)
```

```
                {
                    d = 0;
                    m = m + 1;
                }
            }
            if (d == 28 && m == 2)
            {
                d = 0;
                m = m + 1;
            }
            d = d + 1;
            printf("%s vs %s \t%d/%d/%d\n", Team_names[i], Team_names[j], d, m, y);
            fprintf(fp, "%s vs %s \t%d/%d/%d\n", Team_names[i], Team_names[j], d, m,
y);
        }
    }
    fclose(fp);
}
void create_team()
{
    FILE *fp;
    fp = fopen("E:\\Mine Code\\C Files\\IPL Teams.txt", "w");
    printf("Enter the Team Names:\n");
    int n;
    printf("Enter the Number of Teams:");
    scanf("%d", &n);
    printf("\nEnter the Team Names:\n");
    char names[n][40];
    for (int i = 0; i < n; i++)
    {
        printf("Enter The %d Team: ", i + 1);
        scanf("%s", names[i]);
        fprintf(fp, "%s\n", names[i]);
    }
    fclose(fp);
    display();
}
void display()
{
    char ch;
    FILE *fp;
    fp = fopen("E:\\Mine Code\\C Files\\IPL Teams.txt", "r");
    printf("\nCurrent Added Teams:\n");
    while (!feof(fp))
    {
```

```
        ch = fgetc(fp);
        printf("%c", ch);
    }
    fclose(fp);
}
void update_schedule()
{
    printf("Update The Schedule:\n");
    int d, m, y, N;
    printf("Enter the Start Date(dd mm yyyy): ");
    scanf("%d %d %d", &d, &m, &y);
    int n, n2 = 0;
    FILE *fp;
    fp = fopen("E:\\Mine Code\\C Files\\schedule.txt", "w");
    printf("Enter The number of Teams: ");
    scanf("%d", &n);
    char Team_names[n][50];
    for (int i = n - 1; i > 0; i--)
    {
        n2 += i;
    }
    printf("Number of Matches: %d\n", n2);
    char Schedule[n2];
    for (int i = 0; i < n; i++)
    {
        printf("Enter Team %d: ", i + 1);
        scanf("%s", &Team_names[i]);
    }
    for (int i = 0; i < n; i++)
    {
        for (int j = i + 1; j < n; j++)
        {
            if (d == 31)
            {
                d = 0;
                m = m + 1;
            }
            if (d == 30)
            {
                if (m == 4 || m == 6 || m == 9 || m == 11)
                {
                    d = 0;
                    m = m + 1;
                }
            }
```

```
            if (d == 28 && m == 2)
            {
                d = 0;
                m = m + 1;
            }
            d = d + 1;
            printf("%s vs %s \t%d/%d/%d\n", Team_names[i], Team_names[j], d, m, y);
            fprintf(fp, "%s vs %s \t%d/%d/%d\n", Team_names[i], Team_names[j], d, m,
y);
        }
    }
    fclose(fp);
    printf("Schedule Upadated Succesfully");
}
void view_schedule()
{
    char ch;
    FILE *fp;
    printf("IPL SCHEDULE\n______\n");
    fp = fopen("E:\\Mine Code\\C Files\\schedule.txt", "r");
    while (!feof(fp))
    {
        ch = fgetc(fp);
        printf("%c", ch);
    }
}
void displays()
{
    char ch;
    FILE *fp;
    printf("Team Names:\n");
    fp = fopen("E:\\Mine Code\\C Files\\IPL.txt", "r");
    while (!feof(fp))
    {
        ch = fgetc(fp);
        printf("%c", ch);
    }
    fclose(fp);
}
void main()
{
    int n;
    do
    {
        printf("\n__________\n");
```

```
        printf("1.Display Teams which Already Exist:\n");
        printf("2.Create a New Teams\n");
        printf("3.Add Team\n");
        printf("4.View All Teams\n");
        printf("5.Make a Schedule\n");
        printf("6.View schedule:\n");
        printf("7.Update The Schedule\n");
        printf("8.Delete team");
        printf("9.Exit\n");
        printf("Enter Your Choice: ");
        scanf("%d", &n);
        switch (n)
        {
        case 1:
            displays();
            break;
        case 2:
            create_team();
            break;
        case 3:
            add_team();
            break;
        case 4:
            display();
            break;
        case 5:
            making_schedule();
            break;
        case 6:
            view_schedule();
            break;
        case 7:
            update_schedule();
            break;
        case 8:
            delete_team();
            break;

        case 9:
            printf("Thank You!");
            break;
        }
    } while (n != 7);
}
```