SMART CRICKET ATTENDANCE MARKING SYSTEM

Save the Login ID & Pin to the text file called login.txt

```
{
    FILE *login;              //create a file pointer in the memory

    login = fopen ("login.txt","w");         //fopen

    fprintf(login, "Login Id:%d\n", temp_ID);
    fprintf(login, "Pin:%d\n", temp_pw);

    fclose(login);

}
```

Save temporary attendance to the text file called Today_Attendence.txt

```
    {
        FILE *curent;                                  //create a file pointer in memory

    curent = fopen( "Today_Attendence.txt", "w");         //opening file

    fprintf(curent, "StudentID\t Attendance\n");

    for (int i = 0; i < playerCount; i++) {                //loop for saving Today_Attendence
        fprintf(curent, "%s\t", attendance[i].studentID);

            fprintf(curent, "  %d\t\t", attendance[i].marks[i]);

        fprintf(curent, "\n");
    }

    fclose(curent);                             //close the Today_Attendence.txt
    }
```

FINAL CODE: -

#include <stdio.h>

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

// Define the maximum number of students
#define MAX_STUDENTS  100

// Define the data structure for a student's attendance
typedef struct {
    char studentID[20];
    int marks[MAX_STUDENTS];
} playerAttendance;

//function prototypes
void updateAttendance(playerAttendance *attendance, int
playerCount);
void saveAttendanceToFile(playerAttendance *attendance, int
playerCount);
void viewAttendence();

int main ()
{
    //Login System
    int temp_ID = 12345  ,temp_pw = 98765;

    printf("\t \t ---Welcome TO UOK - CRICKET---\n\n");
    printf("****ONLY USE INTEGER****\n\n");
    printf("Login ID:");            //Login Id: 12345
```

```
    scanf("%d",&temp_ID);
    printf("Pin:");                 //Pin:98765
    scanf("%d",&temp_pw);

{
    FILE *login;              //create a file pointer in the memory

    login = fopen ("login.txt","w");         //fopen

    fprintf(login, "Login Id:%d\n", temp_ID);
    fprintf(login, "Pin:%d\n", temp_pw);

    fclose(login);

}

    if (temp_ID == 12345 && temp_pw == 98765){
        printf("Login Completed...\n\n");

            int playerCount;
            printf("Enter the number of players:");           //input no. of
players
            scanf("%d", &playerCount);
            printf("\n");

            playerAttendance attendance[MAX_STUDENTS];
//declaration of attendance array
```

```
        //input player's students ID
        for (int i = 0; i < playerCount; i++) {
        printf("Enter student ID for player %d: ", i + 1);          //ex:-
PS/2021/001
        scanf("%s", attendance[i].studentID);
        }

        int choice;
        int day = 0;                // Initialize day to 0

        do {
            //Main Menu
            printf("\n\t\t---Menu---\n\n");
            printf("1. Update Attendance\n");
            printf("2. Save Attendance Data to File\n");
            printf("3. View Present Students\n");
            printf("4. Quit\n\n");
            printf("Enter your choice: ");
            scanf("%d", &choice);

          switch (choice) {
            case 1:
              updateAttendance(attendance,  playerCount);
//Marking Attendence
                day++;                                  // Increment the day
after updating attendance
```

```
                break;
            case 2:
                saveAttendanceToFile(attendance, playerCount);
//Save Attendence
                break;
            case 3:
                if (day == 0) {
                    printf("Please update attendance first.\n");
                } else {
                    viewAttendence();                    //View
Attendence Sheet
                    }
                    break;
            case 4:
                printf("Exiting program.\n");          //End The Program.
                break;
            default:
                printf("Invalid input. Please try again.\n");
//invalid inputs
            }
        } while (choice != 4);


        return 0;
}
   else
    {
        printf("Please Try Again");
```

```
    }
}

// Add or update attendance for a player
void updateAttendance(playerAttendance *attendance, int
playerCount) {
    char studentID[20];
    int present;

    printf("Enter student ID:");
    scanf("%s", &studentID);

    int found = 0;
    for (int i = 0; i < playerCount; i++) {
        if (strcmp(attendance[i].studentID, studentID) == 0) {
//compare student IDs
            found = 1;
            printf("Is the player present (1) or absent (0)? ");
//input attendence of player
            scanf("%d", &present);
            attendance[i].marks[i] = present;
            printf("Attendance updated for %s\n", studentID);
            break;
        }
    }
    {
```

```
    FILE *curent;                          //create a file pointer in
memory

   curent = fopen( "Today_Attendence.txt", "w");        //opening file

   fprintf(curent, "StudentID\t Attendance\n");

   for (int i = 0; i < playerCount; i++) {              //loop for saving
Today_Attendence
      fprintf(curent, "%s\t", attendance[i].studentID);

         fprintf(curent, "  %d\t\t", attendance[i].marks[i]);

      fprintf(curent, "\n");
   }

   fclose(curent);                        //close the
Today_Attendence.txt
   }

   if (!found) {
      printf("Student not found.\n");
   }
}

// Save attendance data to a text file
```

```
void saveAttendanceToFile(playerAttendance *attendance, int
playerCount) {
   FILE *fp;                          //create a file pointer in memory

   fp = fopen( "attendance.txt", "a");        //opening file

   if (fp == NULL) {
      printf("Error opening file");
      return;
   }

   fprintf(fp, "StudentID\t Attendance\n");

   for (int i = 0; i < playerCount; i++) {              //loop for saving
attendence
      fprintf(fp, "%s\t", attendance[i].studentID);

         fprintf(fp, "  %d\t\t", attendance[i].marks[i]);

      fprintf(fp, "\n");
   }
fprintf(fp, "\n");

   fclose(fp);                        //close the attendence.txt
   printf("Attendance data saved....\n");
}
```

```
// View attendance sheet
void viewAttendence(){
    char temp[100];
    FILE *fp1;              //create a file pointer in the memory

    fp1 = fopen ("attendance.txt","r");         //fopen

    if(fp1 == NULL){
        printf("Unable to read the specified location\n");
    }

    while(fgets(temp,sizeof(temp),fp1)){
            printf("%s",temp);
    }


  fclose(fp1);
}
```

USER MANUAL: -

- First of all, you should login into the system
    - Login ID & pin were saved to the login.txt file
- Then enter the number of players

- Then enter the student IDs one by one

```
Enter student ID for player 1: PS/2021/001
Enter student ID for player 2: BS/2021/002
Enter student ID for player 3: SS/2021/003
```

- Then the menu will be displayed to you

```
                ---Menu---

1. Update Attendance
2. Save Attendance Data to File
3. View Present Students
4. Quit

Enter your choice:
```

- Enter your choice
  - Choice 1 for update attendance
  - Choice 2 for save attendance data to file
  - Choice 3 for view present students
  - Choice 4 for quit

- If you want to update attendance, enter choice No.1

```
Enter your choice: 1
Enter student ID:PS/2021/001
Is the player present (1) or absent (0)?
```

  - First you need to enter student ID
  - Then mark the attendance for the player

- Temporary attendance sheet was saved to the Today_Attendence.txt file
- If you want to mark attendance for next player, choose choice 1 again

- If you want to save attendance data to permanent file, enter choice No.2
  - Data save permanent for attendance.txt file

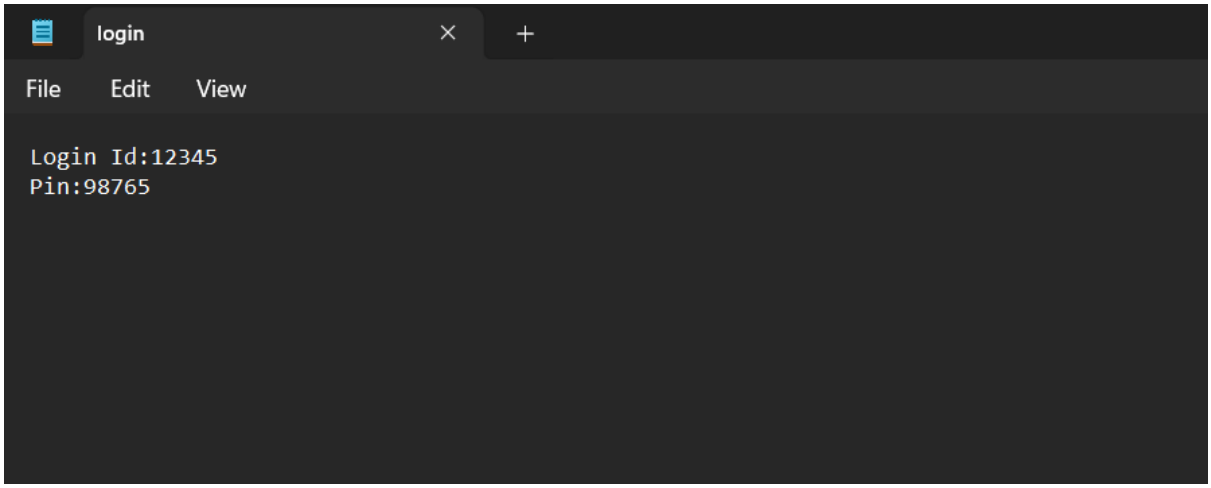
```
Enter your choice: 2
Attendance data saved....
```

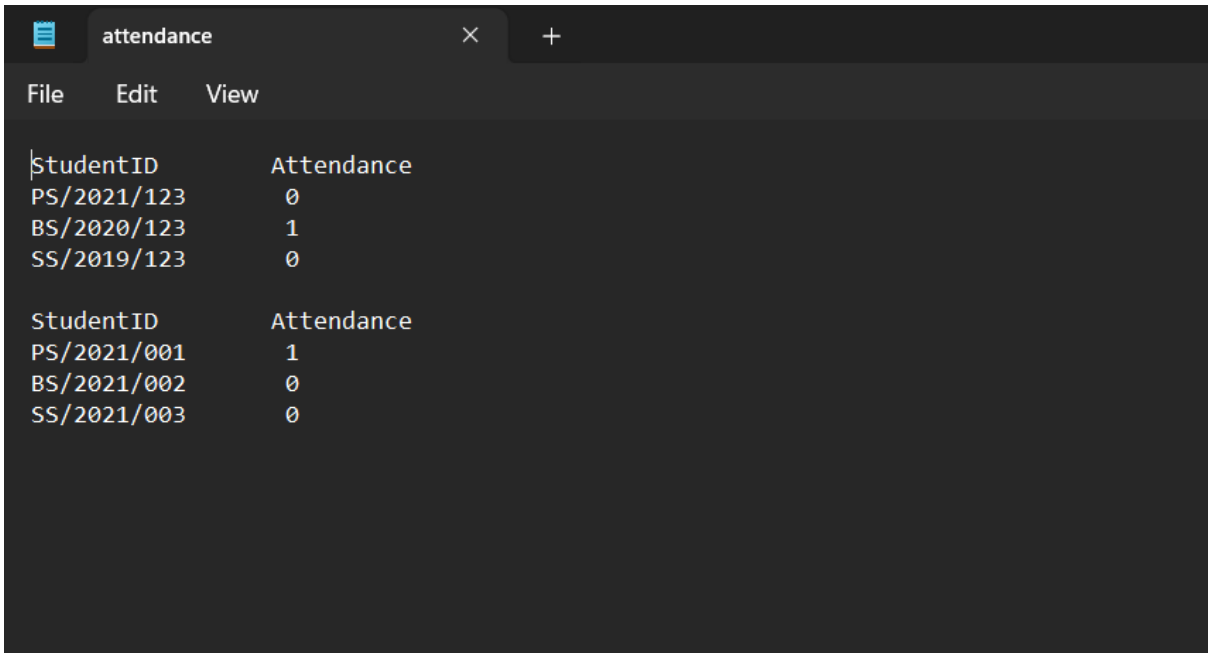
- If you want to view permanent attendance data file, enter choice No.3

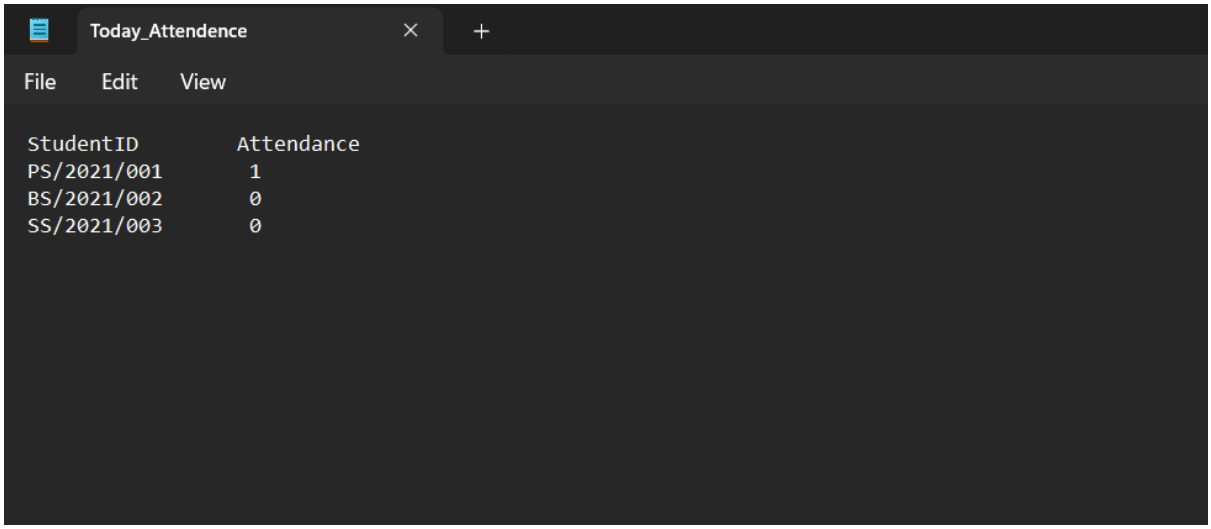
```
Enter your choice: 3
StudentID        Attendance
PS/2021/123       0
BS/2020/123       1
SS/2019/123       0

StudentID        Attendance
PS/2021/001       1
BS/2021/002       0
SS/2021/003       0
```

- If you want to exit from the system, enter choice No.4

```
Enter your choice: 4
Exiting program.
```


login
File
Edit
View
Login Id:12345
Pin:98765


attendance
File
Edit
View
StudentID Attendance
PS/2021/123 0
BS/2020/123 1
SS/2019/123 0
StudentID Attendance
PS/2021/001 1
BS/2021/002 0
SS/2021/003 0


Today_Attendence
File
Edit
View
StudentID Attendance
PS/2021/001 1
BS/2021/002 0
SS/2021/003 0