------------------------------------------------------------------------------------------------------------------------------------------  
  
**STUDENTS ATTENDANCE PORTAL**

**DEPT: BS-ARTIFICIAL INTELLIGENCE**

**ROLL NO: BS-AI-58**

**NAME: MUHAMMAD IBAD**

**CODE FOR THE PROGRAM**

There are two files, one **std.c** main c file which is linked with different header file including one custom header file for its functions named as **student.h.**

Main C file:

#include <stdio.h>

#include <string.h>

#include "./student.h"

#define Max\_Students 117

#define Max\_Attendance 117

struct student students[Max\_Students];

struct Attendance attendance[Max\_Attendance];

int studentCount = 1;

int attendanceCount = 1;

int main()

{

    int choice;

    int rollNo, day, month, year, status;

    char name[50];

    int startDay, startMonth, startyear;

    while (1)

    {

        printf("\n\*\*\* Student Attendendance Portal \*\*\*\n");

        printf("[1] Add Student\n");

        printf("[2] Add Attendance\n");

        printf("[3] Check Attendance by Date\n");

        printf("[4] Check Weekly Attendance\n");

        printf("[5] Added Student List\n");

        printf("[6] Update Attendance\n");

        printf("[7] Exit Program\n");

        printf("Enter your choice: ");

        scanf("%d", &choice);

        switch (choice)

        {

        case 1:

            printf("Enter Roll Number: ");

            scanf("%d", &rollNo);

            printf("Enter Name: ");

            scanf(" %[^\n]", name);

            addStudent(rollNo, name);

            break;

        case 2:

            printf("Enter Date (dd mm yyyy): ");

            scanf("%d %d %d", &day, &month, &year);

            int dayCount = 1;

          while (dayCount <8)

          {

              for (int i = 1; i < studentCount; i++)

            {

                printf("Enter 1 for Present & 0 for Absent: ");

                status = getIntegerInput();

                if (status != 0 && status != 1)

                {

                    printf("\n\t\t\tInvalid status. Please enter 1 for Present or 0 for Absent.\n");

                    break;

                }

                addAttendance(students[i].rollNo, day, month, year, status);

            }

            day++;

            dayCount++;

          }

            break;

        case 3:

            printf("Enter Roll Number: ");

            rollNo = getIntegerInput();

            if (rollNo == -1)

            {

                printf("\n\t\t\t\*\*\* Invalid Input For Roll Number \*\*\*\n");

            }

            printf("Enter Date (dd mm yyyy): ");

            scanf("%d %d %d", &day, &month, &year);

            checkAttendanceByDate(rollNo, day, month, year);

            break;

        case 4:

            printf("Enter Roll Number: ");

            rollNo = getIntegerInput();

            if (rollNo == -1)

            {

                printf("\n\t\t\t\*\*\* Invalid Input For Roll Number \*\*\*\n");

                break;

            }

            printf("Enter Date (dd mm yyyy): ");

            scanf("%d %d %d", &startDay, &startMonth, &startyear);

            checkWeeklyAttendance(rollNo, startDay, startMonth, startyear);

            break;

        case 5:

            addStudentList(studentCount);

            break;

        case 6:

            printf("Enter Roll Number: ");

            rollNo = getIntegerInput();

            if (rollNo == -1)

            {

                printf("\n\t\t\t\*\*\* Invalid Input For Roll Number \*\*\*\n");

            }

            printf("Enter Date (dd mm yyyy): ");

            scanf("%d %d %d", &day, &month, &year);

            printf("Enter 1 for present 0 for Absent: ");

            status = getIntegerInput();

            if (status == -1)

            {

                printf("\n\t\t\t \*\*\* Invalid Input for Attendance \*\*\*\n");

                break;

            }

            updateAttendance(rollNo, day, month, year, status);

            break;

        case 7:

            return 0;

            break;

        default:

            printf("Selected Option is Invalid!");

            break;

        }

    }

}

**Custom Header file (containing required Functions):**

#include<stdio.h>

#include<string.h>

#define Max\_Students 117

#define Max\_Attendance 117

struct student

{

    int rollNo;

    char name[50];

};

struct Attendance

{

    int rollNo;

    int day;

    int month;

    int year;

    int status;

};

extern struct student students[Max\_Students];

extern struct Attendance attendance[Max\_Attendance];

extern int studentCount;

extern int attendanceCount;

void addStudent(int rollNo, const char\* name) {

    if (studentCount >= Max\_Students)

    {

        printf("\n\t\t\tMaximum Student limit reached.\n");

        return;

    }

    for (int i = 1; i < studentCount; i++)

    {

        if (rollNo == students[i].rollNo)

        {

            printf("\n\t\t\t\*\*\*RollNO. Already Exist!\*\*\*\n");

            return;

        }

    }

    students[studentCount].rollNo = rollNo;

    strcpy(students[studentCount].name, name);

    studentCount++;

    printf("\n\t\t\tStudent added Successfully.\n");

};

void addAttendance (int rollNo, int day,int month,int year,int status) {

    if (attendanceCount >= Max\_Attendance)

    {

        printf("\n\t\t\tMaximum attendance limit reached.\n");

        return;

    }

    attendance[attendanceCount].rollNo = rollNo;

    attendance[attendanceCount].day = day;

    attendance[attendanceCount].month = month;

    attendance[attendanceCount].year = year;

    attendance[attendanceCount].status = status;

    attendanceCount++;

    printf("\n\t\t\tAttendance added Successfully for %d\n",rollNo);

*// Save attendance to file*

    char fileName[50];

    sprintf(fileName, "%d-%02d-%02d.txt",year,month, day);

    FILE\* file = fopen(fileName, "a");

    if (file == NULL) {

        printf("\n\t\t\tError creating/opening file.\n");

        return;

    }

*//checking if the file is empty*

    fseek(file, 0, SEEK\_END);

    int fileIsEmpty = (ftell(file)==0);

    fseek(file, 0, SEEK\_SET);

*//write header if the file is empty else left it*

if (fileIsEmpty)

{

    fprintf(file, "\*\*\* Attendance for %02d-%02d-%d \*\*\*\n\n", day, month, year);

    fprintf(file, "Roll No\tName\t\tAttendance\n");

}

    for (int i = 1; i < studentCount; i++) {

        if (students[i].rollNo == rollNo) {

            fprintf(file, "%d\t%s\t\t%s\n", students[i].rollNo, students[i].name, status ? "Present" : "Absent");

        }

    }

    fclose(file);

};

void checkAttendanceByDate(int rollNo, int day, int month, int year) {

    for (int i = 1; i < attendanceCount; i++)

    {

        if (attendance[i].rollNo == rollNo && attendance[i].day == day &&attendance[i].month == month && attendance[i].year == year)

        {

            printf("\n\t\t\tAttendence Status: %s\n",attendance[i].status ? "Present" : "Absent");

            return;

        }

    }

    printf("\n\t\t\tNo attendance record found for the specified date.\n");

};

void checkWeeklyAttendance(int rollNo, int startday, int startMonth, int starttyear) {

    printf("Weekly Attendance for Roll Number %d: \n",rollNo);

    for (int i = 1; i < attendanceCount; i++)

    {

        if (attendance[i].rollNo == rollNo && attendance[i].year == starttyear && attendance[i].month == startMonth && attendance[i].day >= startday && attendance[i].day < startday+7)

        {

            printf("\n\t\t\t%d-%d-%d: %s\n",attendance[i].day,attendance[i].month,attendance[i].year,attendance[i].status ? "Present" :"Absent");

        }

    }

};

void addStudentList(int studentCount) {

   if (studentCount > 1)

   {

     printf("\n\t\t\t\*\*\* The List for Added Student \*\*\*\n");

    for (int i = 1; i < studentCount; i++)

    {

        printf("\n\t\t\t%d. Roll No: %d, Name: %s\n",i,students[i].rollNo, students[i].name);

    }

   }else {

   printf("\n\t\t\tNO Students have been Added yet.");

}

}

void updateAttendance(int rollNo, int day, int month, int year, int status) {

  for (int i = 1; i < attendanceCount; i++)

  {

    if (attendance[i].rollNo == rollNo && attendance[i].day == day &&attendance[i].month == month && attendance[i].year == year)

    {

        attendance[i].status = status;

        printf("\n\t\t\tStatus updated to %s for RollNo: %d",attendance[i].status ? "Present" : "Absent", attendance[i].rollNo);

        return;

    }

  }

  printf("\n\t\t\tAttendance record not found for Roll No: %d\n", rollNo);

}

*// for taking input values from users so that it ensure that the program doesnt get stuck in infinite loop or crashes*

int getIntegerInput()

{

    int value;

    int result = scanf("%d", &value);

    if (result != 1)

    {

*// Clear input buffer*

        int c;

        while ((c = getchar()) != '\n' && c != EOF)

        {

        }

        return -1; *// Indicate error*

    }

    return value;

}

**Main menu after execution:**

![A computer screen shot of a program

Description automatically generated]()