A

Project Report

On

STUDENT DATABASE MANAGEMENT SYSTEM

Submitted in partial fulfillment of the requirement for the III semester

Bachelor of Technology

By

Kuldeep Shahi (2061849)

Rohit Kandpal (2161003)

Sauarbh Singh Gaira (2061928)

Under the Guidance of

Mr Ravindra Koranga

Associate Professor

Department of CSE

DEPARTMENT OF COMPUTER SCIENCE &ENGINEERING

GRAPHIC ERA HILL UNIVERSITY, BHIMTAL CAMPUS

SATTAL ROAD, P.O. BHOWALI

DISTRICT- NAINITAL-263132

2021 – 2022

DECLARATION

We **Kuldeep Shahi**(**2061849**), **Rohit Kandpal (2161003), Sauarbh Singh Gaira(2061928)** hereby declare the work, which is being presented in the project, entitled "**Student Database Management System**" in partial fulfillment of the requirement for the award of the degree **Bachelor of Technology** in the session **2021-2022**, is an authentic record of my own work carried out under the supervision of **Mr Ravinder Koranga** The matter embodied in this project has not been submitted by me for the award of any other degree.

Date: 23-01-2022

CERTIFICATE

The project report entitled “Student Database Management System” being submitted by Kuldeep Shahi (2061849), Rohit Kandpal (2161003), Sauarbh Singh Gaira (2061928) to Graphic Era Hill University Bhimtal Campus for the award of bonafide work carried out by them. they has worked under my guidance and supervision and fulfilled the requirement for the submission of report.

MR Ravinder Koranga

Project Guide

Department of CS & E

GEHU Bhimtal

Dr. Ankur Bisht

(HOD, CSE Dept.)

ACKNOWLEDGEMENT

we take immense pleasure in thanking Honorable **"Mr Ravinder Koranga"** to permit me and carry out this project work with his excellent and optimistic supervision. This has all been possible due to his novel inspiration, able guidance and useful suggestions that helped me to develop as a creative researcher and complete the research work, in time. Words are inadequate in offering my thanks to GOD for providing me everything that we need

Many thanks to Professor **"Dr. Ankur Bisht"** (HOD-CS&A, GEHU) and other faculties for their insightful comments, constructive suggestions, valuable advice, and time in reviewing this thesis.

Finally, yet importantly, we would like to express my heartiest thanks to my beloved parents for their moral support, affection and blessings. We would also like to pay my sincere thanks to all my friends and well-wishers for their help and wishes for the successful completion of this research.

Kuldeep Shahi (2061849)

Rohit Kandpal (2161003)

Sauarbh Singh Gaira(2061928)

TABLE OF CONTENTS

1. INTRODUCTION

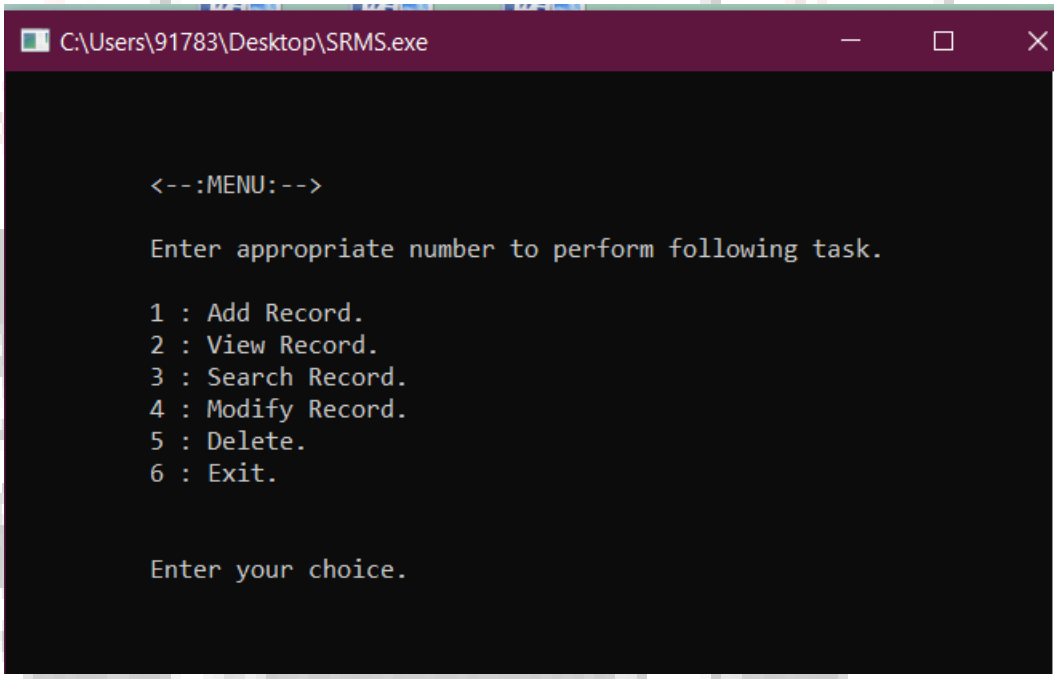
This project is made to make a student database management system using C language to perform basic tasks like add a new record, view record, search record, modify record , delete record and exit. The program achieves this by printing a list on the screen listing all the operations the user can perform in this program. The user will be able to choose ant one of them using using specific keys made for them. The primary aim of this “Student Database Management System” is to provide an improved design methodology, which envisages the future expansion, and modification, which is necessary for a core sector like database management. This necessitates the design to be expandable and modifiable and so a modular approach is used in developing the application software. Anybody who is a student in the corresponding organization has a data stored in the system can access his/her data using name . He has to first enter his/her data in the management system. In today as we have too much data in different organizations this application can be used to handle the data very efficiently so that it do not get mugged up .

2.OBJECTIVE AND SCOPE

The objective of this project is to make a Working database management system that will perform all the basic operations required by a user while being light on the system and fast to execute while having a low complexity and low chances of failures. The code in itself will be very easy to understand and will reduce the problems faced during debugging or solving errors which were not sorted out in the time of development of the code. The User Interface will be clean and easy to understand even for a basic level user and provide all options on the fingertips of the user. our main objective is the customer's satisfaction considering today's faster in the world. When the user enters the program, he will be greeted with a Clean screen(MENU) asking him for different options , after entering the correct option he will be directed to a new page according to which he has choose Since the program is based in an infinite while loop while(1) we are going to need the exit button. When the user selects add record, he will be presented with a prompt asking name roll no etc. Which will the add up in the memory. When the user selects view record, he will be presented with a screen in which the details will be shown. When the user selects search, he will be presented with a screen that will ask him about the user id and will show him record of corresponding user with the same id. when the user select delete user can delete a record with help of specific id and alt last when user select exit he will be exited from the MENU

3 . Methodology/Process Description

The program contains 8 functions a structure to hold all the information of a student which includes Name, mobile number, roll number, course, branch. The program begins with the main function printing out "<--:Student Record Management System:-->" and "Press any key to continue." , the key press is read using the getch() function and then the control of the program is passed on to the menu function which holds a switch case statement containing alll the relevent choices presented to the user by the program,



The code for the switch statement is as follows:-

```
int choice;

system("cls");

gotoxy(10,3);
```

```
printf("<--:MENU:-->");

gotoxy(10,5);

printf("Enter appropriate number to perform following task.");

gotoxy(10,7);

printf("1 : Add Record.");

gotoxy(10,8);

printf("2 : View Record.");

gotoxy(10,9);

printf("3 : Search Record.");

gotoxy(10,10);

printf("4 : Modify Record.");

gotoxy(10,11);

printf("5 : Delete.");

gotoxy(10,12);

printf("6 : Exit.");

gotoxy(10,15);
```

```
printf("Enter your choice.");

scanf("%d",&choice);

switch(choice)

{

case 1:

add();

break;

case 2:

view();

break;

case 3:

search();

break;

case 4:

modify();

break;
```

```
case 5:

deleterec();

break;

case 6:

exit(1);

break;

default:

gotoxy(10,17);

printf("Invalid Choice.");

if(printf("Invalid Choice.")

{

gotoxy(10,18);

printf("Press any key to continue.");

getch();

menu();

}
```

break;

}

Here the list of actions presented to the user is as follows :-

1. Add record

2. View record

3. Search record

4. Modify record

5. Delete record

6. And exit

After the list switch case function is used to move the control flow of the program to the corresponding functions for performing a specific task.

At the end of Switch there is a special case called default which prompts the user if there is any other key pressed other than the ones in the menu, this adds to the robustness of the program. This default case allows the program to deal with inputs which are not supposed to be given by the user.

This robustness is further carried on in the whole program.

The first function is add which is used to add new data to the record. In the function the file containing the record is opened and again to add to the robustness of the program there is a check to make sure that the file exists, if the is not present it will give an error and exit out of the program, if the file exists the user is asked to enter the name, mobile number, roll number, course and branch. After it a prompt asks the user for if he wants to enter any extra data and then according to the input he is asked to enter information or the control of the program jumps back to the switch statement.

The second function is called view. This is a function to view all the records present in the file after being called it prints out the records of all the students present in the file this again has a check for the existence of the file.

The third function is Search this function is used to search a particular entry in the file containing all the data of the students. This function begins with a similar check to check the legibility and existence of the file and then asks the user to enter the name of the student to be searched in the file if a similar record is found the program prints out all the information associated to that entry.

The fourth function is modify this function is used to modify existing data in the file this after the normal check for the legibility and existence of the file containing all the data of the students. This function begins with asking the name of the student whose data we have to alter. Then after entering the new data the program overwrites this new data over the old data and saves it instead.

The fifth function is delete this function is used to completely remove a particular record from the file, after the normal check for legibility and existence of the file the code asks the user the

name of the student whose data the program has to remnove from the file after this the control of the program jumps back to the switch statement.

4.FUNCTIONS USED

The Switch Case

The switch cased statements is one of the main parts of the program since it connects all the functions to the menu() function and call them according to the need of the user.

```
    printf("5 : Delete.");
    gotoxy(10,12);
    printf("6 : Exit.");
    gotoxy(10,15);
    printf("Enter your choice.");
    scanf("%d",&choice);
    switch(choice)
    {
    case 1:
        add();
        break;

    case 2:
        view();
        break;

    case 3:
        search();
        break;

    case 4:
        modify();
        break;

    case 5:
        deleterec();
        break;

    case 6:
        exit(1);
        break;

    default:
        gotoxy(10,17);
        printf("Invalid Choice.");
    }
}
```

The switch case statement itself is written inside a menu function and this function call in other functions, when the operations or tasks performed by user of adding , modifying records of students or any given tasks then it return back to the menu screen. And the if the user type any other input which is not defined in cases so it print “Invalid Choice”.

The case 1 is about add() function , which add record data of new students.

The case 2 is about view() function, which show all records of students stored in files.

The case 3 is about search() function, which help us to search any particular student record by searching its name.

The case 4 is about modify() function, which help us to modify any particular student record, by changing name, address or course and can correct any incorrect data of student.

The case 5 is about deleterec() function, which help us to delete record of the student.

The case 6 is about exit() function, by which we can exit this program.

The first function is main():-

When the user run the program first it shows “Student Record Management System” heading and ‘press any key to continue’ , this instruction is stored in main() function. And then we call the menu() function from the main function.

```
int main()
{
    gotoxy(15,8);
    printf("<--:Student Record Management System:-->");
    gotoxy(19,15);
    printf("Press any key to continue.");
    getch();
    menu();
    return 0;
}
```

The second function is menu():-

When the user press any key to continue , then we go to the menu() which shows all choices to perform tasks and the all other functions call from menu function by switch statements. This function does not return any value because it is declared void function. The user will be asked for choice, which task user have to perform. After the choice entered , according to choice they call that function, ask for input and perform the task.

```
void menu()
{
    int choice;
    system("cls");
    gotoxy(10,3);
    printf("<--:MENU:-->");
    gotoxy(10,5);
    printf("Enter appropriate number to perform following task.");
    gotoxy(10,7);
    printf("1 : Add Record.");
    gotoxy(10,8);
    printf("2 : View Record.");
    gotoxy(10,9);
    printf("3 : Search Record.");
    gotoxy(10,10);
    printf("4 : Modify Record.");
    gotoxy(10,11);
    printf("5 : Delete.");
    gotoxy(10,12);
    printf("6 : Exit.");
    gotoxy(10,15);
    printf("Enter your choice.");
    scanf("%d",&choice);
    switch(choice)
    {
    case 1:
        add();
        break;

    case 2:
        view();
        break;

    case 3:
        search();
        break;

    case 3:
        search();
        break;

    case 4:
        modify();
        break;

    case 5:
        deleterec();
        break;

    case 6:
        exit(1);
        break;

    default:
        gotoxy(10,17);
        printf("Invalid Choice.");
    }
}
```

The third function is add():-

This function is used to add the record of new student. Data which include name of student, mobile number, roll number, course and branch. This function is in void type, since it returns no value. If the user choose to add record of the student, then program ask the name of the student, mobile number, roll number, student course and branch of the course. Afterwards, when the user entered complete and correct information, then the program ask whether you want to add another student record[Y/N]. If you type yes[Y], it again ask the data of another student, student name, mobile number,etc or if you type

no[N], it will move forward and bring back to menu() function to perform another task according to user.

The forth function is view():-

This function will be called when the user pressed 2 in the menu screen. After this the view() function is called from the menu function, and this function shows all record stored in files. If no record file will be there so it print the “Error opening file” and it will terminate the program. And if that file is present it will shows all stored records and bring back the user to the menu screen.

```
void view()
{
    FILE *fp;
    int i=1,j;
    struct student std;
    system("cls");
    gotoxy(10,3);
    printf("<--:VIEW RECORD:-->");
    gotoxy(10,5);
    printf("S.No   Name of Student       Mobile No   Roll No  Course      Branch");
    gotoxy(10,6);
    printf("-----------------------------------------------------------------");
    fp = fopen("record.txt","rb+");
    if(fp == NULL){
        gotoxy(10,8);
        printf("Error opening file.");
        exit(1);
    }
    j=8;
    while(fread(&std,sizeof(std),1,fp) == 1){
        gotoxy(10,j);
        printf("%-7d%-22s%-12s%-9d%-12s%-12s",i,std.name,std.mobile,std.rollno,std.course,std.branch);
        i++;
        j++;
    }
    fclose(fp);
    gotoxy(10,j+3);
    printf("Press any key to continue.");
    getch();
    menu();
}
```

The fifth function is search():-

This function will be called when the user pressed 3 in the menu screen. After this the search() function is called from the menu function, and this function asked the name of student whose data user have to search. This function using strcmp for comparing string data the input name of the student and the record students name and the input name matches to the record name and then it shows all record of that student. After then the program bring back the user to menu screen.

```
void search()
{
    FILE *fp;
    struct student std;
    char stname[20];
    system("cls");
    gotoxy(10,3);
    printf("<--:SEARCH RECORD:-->");
    gotoxy(10,5);
    printf("Enter name of student : ");
    fflush(stdin);
    gets(stname);
    fp = fopen("record.txt","rb+");
    if(fp == NULL){
        gotoxy(10,6);
        printf("Error opening file");
        exit(1);
    }
    while(fread(&std,sizeof(std),1,fp ) == 1){
        if(strcmp(stname,std.name) == 0){
            gotoxy(10,8);
            printf("Name : %s",std.name);
            gotoxy(10,9);
            printf("Mobile Number : %s",std.mobile);
            gotoxy(10,10);
            printf("Roll No : %d",std.rollno);
            gotoxy(10,11);
            printf("Course : %s",std.course);
            gotoxy(10,12);
            printf("Branch : %s",std.branch);
        }
    }
    fclose(fp);
    gotoxy(10,16);
    printf("Press any key to continue.");
    getch();
    menu();
```

The sixth function is modify():-

This function will be called when the user pressed 4 in the menu screen. After this the modify() function is called from the menu function, the modify() function asked the name of student whose data user want to modify after then it follow same process as search()

function follows, it compare the strings and search the name from record. After then program ask the modified data i.e., the new name, mobile number, rollno. , course and branch and press enter your data will be modified. And then again program will send you back to menu screen, and if user want to check your modified record press 2 and go to view() function to view the modified record. This function first read the data from user, then it search and write the modified record in the file.

```
void modify()
{
    char stname[20];
    FILE *fp;
    struct student std;
    system("cls");
    gotoxy(10,3);
    printf("<--:MODIFY RECORD:-->");
    gotoxy(10,5);
    printf("Enter name of student to modify: ");
    fflush(stdin);
    gets(stname);
    fp = fopen("record.txt","rb+");
    if(fp == NULL){
        gotoxy(10,6);
        printf("Error opening file");
        exit(1);
    }
    rewind(fp);
    fflush(stdin);
    while(fread(&std,sizeof(std),1,fp) == 1)
    {
        if(strcmp(stname,std.name) == 0){
            gotoxy(10,7);
            printf("Enter name: ");
            gets(std.name);
            gotoxy(10,8);
            printf("Enter mobile number : ");
            gets(std.mobile);
            gotoxy(10,9);
            printf("Enter roll no : ");
            scanf("%d",&std.rollno);
            gotoxy(10,10);
            printf("Enter Course : ");
            fflush(stdin);
            printf("Enter Course : ");
            fflush(stdin);
            gets(std.course);
            gotoxy(10,11);
            printf("Enter Branch : ");
            fflush(stdin);
            gets(std.branch);
            fseek(fp ,-sizeof(std),SEEK_CUR);
            fwrite(&std,sizeof(std),1,fp);
            break;
        }
    }
    fclose(fp);
    gotoxy(10,16);
    printf("Press any key to continue.");
    getch();
    menu();
}
```

The seventh function is deleterec():-

This function will be called when the user pressed 5 in the menu screen. After this the deleterec() function is called from the menu function, this function is used to completely delete a particular student record from the file, after the normal check for legibility and existence of the file the code asks the user the name of the student whose data the program has to remove from the file after this the control of the program jumps back to the switch statement.

```
void deleterec()
{
    char stname[20];
    FILE *fp,*ft;
    struct student std;
    system("cls");
    gotoxy(10,3);
    printf("<--:DELETE RECORD:-->");
    gotoxy(10,5);
    printf("Enter name of student to delete record : ");
    fflush(stdin);
    gets(stname);
    fp = fopen("record.txt","rb+");
    if(fp == NULL){
        gotoxy(10,6);
        printf("Error opening file");
        exit(1);
    }
    ft = fopen("temp.txt","wb+");
    if(ft == NULL){
        gotoxy(10,6);
        printf("Error opening file");
        exit(1);
```

```
    }
    ft = fopen("temp.txt","wb+");
    if(ft == NULL){
        gotoxy(10,6);
        printf("Error opening file");
        exit(1);
    }
    while(fread(&std,sizeof(std),1,fp) == 1){
        if(strcmp(stname,std.name)!=0)
            fwrite(&std,sizeof(std),1,ft);
    }
    fclose(fp);
    fclose(ft);
    remove("record.txt");
    rename("temp.txt","record.txt");
    gotoxy(10,10);
    printf("Press any key to continue.");
    getch();
    menu();
}
```

4. Resources and Limitations :

Hardware :

- Keyboard
- Monitor
- Cabinet (CPU/ALU/MU)

Software :

- Windows Operating System(64 bit)
- Dev cpp portable

Limitations :

- Only compatible for windows operating system.
- Only keyboard interface is implemented.
- It doesn't include various other student information fields like attendance, marks, and fees record.

5. Program Code :

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<windows.h>
#include<string.h>
void gotoxy(int ,int );
void menu();
void add();
void view();
void search();
void modify();
void deleterec();
struct student
{
    char name[20];
    char mobile[10];
    int rollno;
    char course[20];
    char branch[20];
};
int main()
{
    gotoxy(15,8);
    printf("<--:Student Record Management System:-->");
    gotoxy(19,15);
    printf("Press any key to continue.");
    getch();
    menu();
    return 0;
```

```
}
void menu()
{
    int choice;
    system("cls");
    gotoxy(10,3);
    printf("<--:MENU:-->");
    gotoxy(10,5);
    printf("Enter appropriate number to perform following
task.");
    gotoxy(10,7);
    printf("1 : Add Record.");
    gotoxy(10,8);
    printf("2 : View Record.");
    gotoxy(10,9);
    printf("3 : Search Record.");
    gotoxy(10,10);
    printf("4 : Modify Record.");
    gotoxy(10,11);
    printf("5 : Delete.");
    gotoxy(10,12);
    printf("6 : Exit.");
    gotoxy(10,15);
    printf("Enter your choice.");
    scanf("%d",&choice);
    switch(choice)
    {
    case 1:
        add();
        break;
```

```
    case 2:
        view();
        break;

    case 3:
        search();
        break;

    case 4:
        modify();
        break;

    case 5:
        deleterec();
        break;

    case 6:
        exit(1);
        break;

    default:
        gotoxy(10,17);
        printf("Invalid Choice.");
    }
}
void add()
{
    FILE *fp;
    struct student std;
    char another ='y';
    system("cls");
```

```
    fp = fopen("record.txt","ab+");
    if(fp == NULL){
        gotoxy(10,5);
        printf("Error opening file");
        exit(1);
    }
    fflush(stdin);
    while(another == 'y')
    {
        gotoxy(10,3);
        printf("<--:ADD RECORD:-->");
        gotoxy(10,5);
        printf("Enter details of student.");
        gotoxy(10,7);
        printf("Enter Name : ");
//        gets(std.name);///???
        gets(std.name);
        gotoxy(10,8);
        printf("Enter Mobile Number : ");
        gets(std.mobile);
        gotoxy(10,9);
        printf("Enter Roll No : ");
        scanf("%d",&std.rollno);
        fflush(stdin);
        gotoxy(10,10);
        printf("Enter Course : ");
//        gets(std.course);///???
        gets(std.course);
        gotoxy(10,11);
        printf("Enter Branch : ");
```

```
        gets(std.branch);
//        gotoxy(10,12);
//        printf("Enter Father's Name : ");
//        gets(std.fathername);
        fwrite(&std,sizeof(std),1,fp);
        gotoxy(10,15);
        printf("Want to add of another record? Then press 'y' else
'n'.");
        fflush(stdin);
//        another = getch();///???
        another = getch();
        system("cls");
        fflush(stdin);
    }
    fclose(fp);
    gotoxy(10,18);
    printf("Press any key to continue.");
    getch();
    menu();
}
void view()
{
    FILE *fp;
    int i=1,j;
    struct student std;
    system("cls");
    gotoxy(10,3);
    printf("<--:VIEW RECORD:-->");
    gotoxy(10,5);
    printf("S.No   Name of Student       Mobile No   Roll No
Course      Branch");
```

```
    gotoxy(10,6);
    printf("-----------------------------------------------------------
--");
    fp = fopen("record.txt","rb+");
    if(fp == NULL){
        gotoxy(10,8);
        printf("Error opening file.");
        exit(1);
    }
    j=8;
    while(fread(&std,sizeof(std),1,fp) == 1){
        gotoxy(10,j);
        printf("%-7d%-22s%-12s%-9d%-12s%-
12s",i,std.name,std.mobile,std.rollno,std.course,std.branch);
        i++;
        j++;
    }
    fclose(fp);
    gotoxy(10,j+3);
    printf("Press any key to continue.");
    getch();
    menu();
}
void search()
{
    FILE *fp;
    struct student std;
    char stname[20];
    system("cls");
    gotoxy(10,3);
    printf("<--:SEARCH RECORD:-->");
```

```
    gotoxy(10,5);
    printf("Enter name of student : ");
    fflush(stdin);
    gets(stname);
    fp = fopen("record.txt","rb+");
    if(fp == NULL){
        gotoxy(10,6);
        printf("Error opening file");
        exit(1);
    }
    while(fread(&std,sizeof(std),1,fp ) == 1){
        if(strcmp(stname,std.name) == 0){
            gotoxy(10,8);
            printf("Name : %s",std.name);
            gotoxy(10,9);
            printf("Mobile Number : %s",std.mobile);
            gotoxy(10,10);
            printf("Roll No : %d",std.rollno);
            gotoxy(10,11);
            printf("Course : %s",std.course);
            gotoxy(10,12);
            printf("Branch : %s",std.branch);
        }
    }
    fclose(fp);
    gotoxy(10,16);
    printf("Press any key to continue.");
    getch();
    menu();
}
void modify()
```

```
{
    char stname[20];
    FILE *fp;
    struct student std;
    system("cls");
    gotoxy(10,3);
    printf("<--:MODIFY RECORD:-->");
    gotoxy(10,5);
    printf("Enter name of student to modify: ");
    fflush(stdin);
    gets(stname);
    fp = fopen("record.txt","rb+");
    if(fp == NULL){
       gotoxy(10,6);
       printf("Error opening file");
       exit(1);
    }
    rewind(fp);
    fflush(stdin);
    while(fread(&std,sizeof(std),1,fp) == 1)
    {
       if(strcmp(stname,std.name) == 0){
          gotoxy(10,7);
          printf("Enter name: ");
          gets(std.name);
          gotoxy(10,8);
          printf("Enter mobile number : ");
          gets(std.mobile);
          gotoxy(10,9);
          printf("Enter roll no : ");
          scanf("%d",&std.rollno);
```

```
            gotoxy(10,10);
            printf("Enter Course : ");
            fflush(stdin);
            gets(std.course);
            gotoxy(10,11);
            printf("Enter Branch : ");
            fflush(stdin);
            gets(std.branch);
            fseek(fp ,-sizeof(std),SEEK_CUR);
            fwrite(&std,sizeof(std),1,fp);
            break;
        }
    }
    fclose(fp);
    gotoxy(10,16);
    printf("Press any key to continue.");
    getch();
    menu();
}
void deleterec()
{
    char stname[20];
    FILE *fp,*ft;
    struct student std;
    system("cls");
    gotoxy(10,3);
    printf("<--:DELETE RECORD:-->");
    gotoxy(10,5);
    printf("Enter name of student to delete record : ");
    fflush(stdin);
    gets(stname);
```

```
    fp = fopen("record.txt","rb+");
    if(fp == NULL){
        gotoxy(10,6);
        printf("Error opening file");
        exit(1);
    }
    ft = fopen("temp.txt","wb+");
    if(ft == NULL){
        gotoxy(10,6);
        printf("Error opening file");
        exit(1);
    }
    while(fread(&std,sizeof(std),1,fp) == 1){
        if(strcmp(stname,std.name)!=0)
            fwrite(&std,sizeof(std),1,ft);
    }
    fclose(fp);
    fclose(ft);
    remove("record.txt");
    rename("temp.txt","record.txt");
    gotoxy(10,10);
    printf("Press any key to continue.");
    getch();
    menu();
}
void gotoxy(int x,int y)
{
        COORD c;
        c.X=x;
        c.Y=y;
```

```
SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE),c);
}
```

6. Conclusion :

Benefits of Student Record Management System:

The first key **advantage** of a **student record management system** is that an educational institution can keep proper track of **students**' data.

This includes areas such as fee **records**, examination **records** of the **students**, hostel **records**, transport **records**, offered by the institutes.

It is not easy to maintain registers or hard copy of all the students data .

A hard copy of record can be destroyed, or misplaced and it requires lot of manual work for updation and has to be kept secure .

On the other hand student record management system can be stored in and viewed from multiple devices, so there are less chances of data loss.

It helps institution to modify and search for a student's credentials and follow the student's progress.

Student record management is even better when online, as it aids students from the luxury of their homes.

Online student record management system allows you to watch your attendance on a daily basis if you want to.

It also keeps you updated about your fees and finances, and your performance in previous semesters and examinations.

This was the report file for the mini project for B.Tech 3rd semester and consisted of a student record management system software which was capable of doing all the basic operations without being too heavy on the system or without requiring too much resources.

It lets you add a new record, view existing records, search an existing record, modify a record and delete a record.

7. Bibliography and references

- Geeksforgeeks.org
- youtube.com
- wikipedia.org