

**Tribhuvan university**

**Institute of engineering**

**Pulchwok Campus**

**Student Record System**

**A course project submitted to the department of Electronics and Computer Engineering in partial fulfillment of the requirements for the practical course on computer programming[CT 401]**

***Submitted by:***

Ashish Sharma (076/BCE/030)

Ashish Upadhaya Sharma (076/BCE/031)

Anuj Chalise (076/BCE/022)

Amrit Bhetuwal (076/BCE/017)

***Submitted to:***

Department of Electronics and Computer Engineering

Pulchwok Campus

Institute of Engineering, Tribhuvan University

Lalitpur,Nepal

Falgun, 2076

# Abstract

The main objective of this program is to enable the school administration or any other organization to create and handle records of their students in computer databases and avoiding the use of bulky paperworks which is more time consuming, costly and tedious.. The program is developed in c programming language. The program is password protected for safety and privacy of information. This program enables the administrator or user to add records of students( including their and their parents name, address, class, contact number and unique student id). It enables to search record for information of particular student from their id and also allows to modify and delete records. There is included option for changing of password too.

Keywords: add record, search record, modify rcord, delete record, change password, database.

# ACKNOWLEDGEMENTS

First and foremost of all, our special gratitude goes to **Mr. SG** for his guidelines and ideas for this project. We thank him for organizing this course and acknowledge his effort that encouraged us to take this challenging project and relate the programming more or less to our own field of study in upcoming years i.e. aerospace engineering or simply airlines.

We also thank everyone who have helped us directly or indirectly in coming up with the idea of this project, friends, who provided us with their valuable comments and suggestions regarding the project all the books’ authors whose books we referred to and helped us gain the knowledge and information required for the accomplishment of this project.

Finally, we would also like to offer our gratitude to all our teachers whose lectures and ideas were the basis for our project research and appreciate the support rendered by DOECE, IOE. Without all your support and guidance the completion of this project would have been impossible.

# Table of Contents

[1. Abstract 2](#_Toc34110862)

[2. ACKNOWLEDGEMENTS 3](#_Toc34110863)

[3. Table of Contents 4](#_Toc34110864)

[4. Introduction 6](#_Toc34110865)

[4.1. Background and problem statement 6](#_Toc34110866)

[4.2. Objectives 7](#_Toc34110867)

[4.3. Limitations 7](#_Toc34110868)

[4.4. Problem analysis 7](#_Toc34110869)

[Understanding the problem 7](#_Toc34110870)

[Input requirements 8](#_Toc34110871)

[Output requirements 8](#_Toc34110872)

[Processing requirement 8](#_Toc34110873)

[Technical feasilbility 8](#_Toc34110874)

[4.5. Review of related literature 8](#_Toc34110875)

[5. Chapter 2: Algorithm and flowchart 14](#_Toc34110876)

[5.1. Algorithm 14](#_Toc34110877)

[5.2. Flowchart 19](#_Toc34110878)

[6. Implementation and coding 30](#_Toc34110879)

[6.1. Implementation of code 30](#_Toc34110880)

[6.2. Coding of the project: 31](#_Toc34110881)

[7. Result and discussion 32](#_Toc34110882)

[8. Conclusion 35](#_Toc34110883)

[9. Appendix 35](#_Toc34110884)

**List of Figures**

[Figure 5.1 19](file:///C:\Users\htf\Desktop\c_project_bce_030_022_031_017%20(Repaired).docx#_Toc34110885)

[Figure 5.2 20](file:///C:\Users\htf\Desktop\c_project_bce_030_022_031_017%20(Repaired).docx#_Toc34110886)

[Figure 5.3 21](file:///C:\Users\htf\Desktop\c_project_bce_030_022_031_017%20(Repaired).docx#_Toc34110887)

[Figure 5.4 22](file:///C:\Users\htf\Desktop\c_project_bce_030_022_031_017%20(Repaired).docx#_Toc34110888)

[Figure 5.5 23](file:///C:\Users\htf\Desktop\c_project_bce_030_022_031_017%20(Repaired).docx#_Toc34110889)

[Figure 5.6 24](file:///C:\Users\htf\Desktop\c_project_bce_030_022_031_017%20(Repaired).docx#_Toc34110890)

[Figure 5.7 25](file:///C:\Users\htf\Desktop\c_project_bce_030_022_031_017%20(Repaired).docx#_Toc34110891)

[Figure 5.8 26](file:///C:\Users\htf\Desktop\c_project_bce_030_022_031_017%20(Repaired).docx#_Toc34110892)

[Figure 5.9 27](file:///C:\Users\htf\Desktop\c_project_bce_030_022_031_017%20(Repaired).docx#_Toc34110893)

[Figure 5.10 28](file:///C:\Users\htf\Desktop\c_project_bce_030_022_031_017%20(Repaired).docx#_Toc34110894)

[Figure 5.11 29](file:///C:\Users\htf\Desktop\c_project_bce_030_022_031_017%20(Repaired).docx#_Toc34110895)

[Figure 6.1 30](#_Toc34110896)

[Figure 6.2 30](#_Toc34110897)

[Figure 6.3 31](#_Toc34110898)

[Figure 7.1 32](#_Toc34110899)

[Figure 7.2 33](#_Toc34110900)

[Figure 7.3 33](#_Toc34110901)

[Figure 7.4 34](#_Toc34110902)

[Figure 7.5 34](#_Toc34110903)

[Figure 7.6 35](#_Toc34110904)

# Introduction

This mini project in c is aimed towards efficient, easy and affordable handling of student records as suggested by the name. Specially record management in bulky paperwork is time consu,ming, tedious and costly job. This program can eleminater all these drawbacks by reducing the time required, manpower required and resources required for this task. Apart from that computerized system are the lifeforce and necessity of modern day organizations. So, this project has useful practical applications.

File handling, control statements, loops and structure and strings form the core of this project. It is a simple application program written in in C programming language, and compiled using GNU GCC compiler on Code::Blocks IDE. The source code consists of over 310 lines of code which are made user friendly as possible by the optimum use of functions to make the function discrete and divide the project coding among the group members. The project source code effectively utilizes the user defined functions and the concept of file handling.

## Background and problem statement

Engineering being a technical and applied field of study, the most important factor for students is to measure their skills, ability and understandings in practical approach rather than theoretical knowledge. Considering this fact a project is given to students of Civil Engineering-I(1st year 1st part) to prepare a program in c language to reinforce our insights and understandings gained during the lab and theory classes of computer programming[CT401] as per requirement.

This project is a part of the requirement of practical course on computer programming for civil 1st year 1st part students.

This project that we have worked on is based on file handling. We decided to do project on this topic as it not only utilizes the concepts of file handling but covers use of almost all topiucs that we have studied under this course. Also this program has practical applkications.

To uild this project it took us nearly a week managing time between our day to day classes. It required knowledge of use of looping, conditions, structure, strings, functions, data files and basic concepts of C programming language. The various theory and lab classes conducted by our teacher **SG sir** throughout the semester proved very helpful. Apart from that we consulted various books and internet in gaining concepts of computer programming.

After gaining the concepts and imagining a layout and design of our program, we divided the workload among us bby use of user defined functions and started writing the codes. After successfully writing , compiling, linking and executing the program we then debugged some errors that were occurred during programming. This helped a lot to get insight at debugging too. Then we divided the workload for writing report and at last combined it too. It was also very helpful in developing our skill in team working.

## Objectives

The main objective of our program are as follows:

1. To explore the features of Cprogramming
2. To be familiar with resource reusability by making user defined functions.
3. To learn about file handling through c programming.
4. To make the program to occupy less memory as far as possible.
5. To divide the program into smaller manageable parts using functions and make it more eligible and understanble .
6. To gain insight on debugging.
7. To get basic idea of software development process.
8. To learn to work as a team and share responsibilities.

## Limitations

The limitations of this project are:

1. The user must enter the options and data in correct format.
2. There is lack of batch operations.

## Problem analysis

### Understanding the problem

A meeting of the group members was held and discussion regarding the project was made. First we decided to make project in student record system them discussions and decisions for the: *input, layout, options, logic and ouput* of the project were made.

### Input requirements

The program was made command prompt type. The code must be clear so as to direct the user to input desired data in desired format. The program starts by opening login page where the user should enter password. Then main menu opens where numbers as options should be entered as choice to proceed as choosed.

### Output requirements

Before programming the it should be decided how the output should be displayed. In the program the previous window is cleared and next window is opened based on the choice of user.

### Processing requirement

The processing requirement should be clearly defined to convert the given data to the required output. In this project four people collaborated together for the completion of project. The coding is done in 64-bit OS using GNU GCC compiler on Code::Blocks IDE.

### Technical feasilbility

In the context of feasibility the coding didn’t require that much hard work and manpower due to simplicity of program. Though the program was short, it was user interacvtive and it yielded the required output.

## Review of related literature

**C programming language:**

C is structured programming based computer programming language was developed by Dennis Ritchie at Bell laboratories in 1972. C is a general-purpose, imperative computer programming language, supporting structured programming, lexical variable scope and recursion, while a static type system prevents many unintended operations. By design, C provides constructs that map efficiently to typical machine instructions, and therefore it has found lasting use in applications that had formerly been coded in assembly language, including operating systems, as well as various application software for computers ranging from supercomputers to embedded systems.

**Control Statement:**

Logical operation is carried out by several symmetrical or logical statements. There are two types of control statement based on their function.

**Selective structure:**

Selective structures are used when we have a number of situations where we need to change the order of execution of statements based on certain condition. The selective statements make a decision to take the right path before changing the order of execution. C provides the following statements for selective structure:

i.If statements

ii.switch statements

***if statements:***

The if statement is a powerful decision making statement and it is used to control the flow of execution of statements. It is a two way statement and is used in conjunction with an expression.

If statement allows the computer to evaluate the expression first and then on depending whether the value of the expression is true or false it transfer the control to the particular statement. At this point of the program has two paths to follow: one for true condition and other for false condition. The types of if statements are explained below:

***Simple if statement:***

The simple if statement is used to conditionally excite a block of code based on whether a test condition is true or false. If the condition is true the block of code is executed, otherwise it is skipped. The syntax of if statement is given below:

if(test expression)

{

statement-block;

}

statement-x;

***if else statement:***

Another use of else is when there are multiple conditional statements that may all evaluate to true, yet you want only one if statement's body to execute. You can use an "else if" statement following an if statement and its body; that way, if the first statement is true, the "else if" will be ignored, but if the if statement is false, it will then check the condition for the else if statement. If the if statement was true the else statement will not be checked. It is possible to use numerous else if statements to ensure that only one block of code is executed. The syntax of if else statement is given below:

if(test expression)

{ true block statement;

} else {

false block statement;

}

**The switch statement:**

C has built in multi way decision statement known as switch. It successively test the value of an expression against a list of case values (integer or character consonants).when a match is found the statement associated with that case is executed. The syntax of switch expression is given below:

switch(expression)

{

case constant-1:

block-1;

break;

case constant-2:

block-2;

break;

…………. ………….

case constant-n:

block-n;

break;

default:

default statement:

}

**Goto:**

Goto is a jumping statement in c language, which transfer the program’s control from one statement to another statement (where label is defined). Goto can transfer the program’s within the same block and there must a label, where you want to transfer program’s control.

***Looping:***

You may encounter situations, when a block of code needs to be executed several number of times. In general, statements are executed sequentially. The first statement in a function is executed first, followed by the second, and so on. Programming languages provide various control structures that allow for more complicated execution paths. A loop statement allows us to execute a statement or group of statements multiple times. Different types of loops are discussed below with their major characteristics and syntax used in C:

**while loop:**

The “while loop” specifies that a section of code should be executed while a certain condition holds true. The syntax of while loop is given below:

while(test expression)

{

body of loop

( statements block) ;

}

**do while statement:**

The do while statement is very similar to while statement. It also specifies that a section of code should be executed while a certain condition holds true. the difference between while and do while loop is that while loop test its condition at the top of its loop but do while loop tests its condition at the bottom of loop. In while loop, if the test condition is false, the block of code is skipped. Since condition is tested at the bottom of loop in do while loop, its block of code is always executed at least once. The syntax of do while loop is given below:

do {

body of loop

}while (test expression);

**For loop:**

The for loop is used to execute a block of code for a fixed number of repetitions.

Initialization is generally an assignment statement used to set loop control variable. Test expression is a relational expression that determines when loop exits. Update expression defines how the loop variable changes each time when the loop is repeated. The syntax of for loop is given below:

for (initialization expression ; test expression ; update expression)

{

body of loop;

}

**break statement:**

The break statement in C programming has the following two usages −

* When a break statement is encountered inside a loop, the loop is immediately terminated and the program control resumes at the next statement following the loop.
* It can be used to terminate a case in the switch statement (covered in the next chapter).

In case of nested loops, the break statement will stop the execution of the innermost loop and start executing the next line of code after the block.

**Function:**

A function is a self-contained program segment that carries out some specific well defined task. Every c program consists of one or functions. Execution of program always begins by carrying out instruction in main. Function makes program significantly easier to understand and maintain. A well written function may be reused in multiple programs. Program that are easier to design, debug and maintain.

**Return statement:**

A function may or may not send back any value to the calling function. If it does, it is through return statement. The called function can only return only one value per call at most. The syntax of return statement is given below:

Return variable;

Or return ;

**File:**

Many applications require that information be written to or read from an auxiliary memory device. Such information is stored on the memory device in the form of a data file. The data files allow us to store information permanently and to access and alter that information whenever necessary.

**File operations**

1. **Opening a file - for creation and edit**

Opening a file is performed using the fopen() function defined in the stdio.h header file.

The syntax for opening a file in standard I/O is:

ptr = fopen("fileopen","mode");

where modes are ***r*** for reading, ***w*** for writing, a for appending, ***rb*** for reading binary files, ***wb*** for writing binary files and ***ab*** for appending binary files.

1. **Closing a File**

The file (both text and binary) should be closed after reading/writing.

Closing a file is performed using the fclose() function.

fclose(fptr);

Here, fptr is a file pointer associated with the file to be closed.

1. **Reading and writing to a file**

Reading and writing to afile can be done with the help of functions like fgetc(), fputc(), fgets(), fputs(), fread(), fwrite().

# Chapter 2: Algorithm and flowchart

Algorithm

A. main()  
1. Start  
2. Ask user to enter password and read user entered value as pass  
3. If pass does not match 'password', display 'invalid password ',goto 6  
4. Else display all choices and ask user to enter the choice.  
5. If   
 choice=1 , call add function   
 choice=2 , call search function  
 choice =3 ,call modify function  
 choice=4 ,call delete function  
 choice=5 ,call change password  
 choice=6 , exit  
6. Stop

B add\_data()  
1. Start  
2. Call drawrectangle function   
3. Open record.txt file in append mode  
4. If file=null ,display error message, goto 8  
5. Else ask user to enter ID, name, mother's name, father's name, address, phone number, and class  
6. Write above data in file record.txt  
7. Close the file  
8. Stop

C search\_data()  
1. Start  
2. Call drawrectangle function   
3. Ask user to enter student's ID name and read it as S\_ID  
4. Open read.txt file in read mode  
5. If file=null, display error message,goto 9.  
6. Else check if ID matches S \_ID  
7. If the condition is true, display the records  
8. If the condition is false, display no records found.  
9. Stop  
D. modify\_data()  
1. Start  
2. Call drawrectangle function   
3. Ask user to enter ID to modify the record and read it as S\_ID  
4. Open record.txt file in read mode  
5. If file=null, display error message, goto 10  
6. Else check existence of the file  
7. If the condition is false, display no records found, goto 10  
8. If the condition is true ask the user to enter modified records.  
9. Write the modified data in record.txt and close the file  
10. Stop

E delete\_data()

1. Start

2. Call drawrectangle function

3. Ask the user to enter the ID to delete

4. Read S\_ID

5. Open record.txt in read mode and temp.txt in write mode

6. If file=null, display error message goto 9

7. Else if S\_ID=ID, continue to next record

8. Else write the data in file temp.txt and rename the file as record.txt

9. Stop

F change\_password()

1. Start

2. Call drawrectangle function

3. Enter the new password

4. Read the new password

5. Open password.txt in write mode

6. If file=null, display error message, goto 9.

7. else write the new password in the file password.txt

8. close the file password.txt

9. Stop

G main\_window()

1. Start

2. Display the choices

Add Student

Search student

modify student

delete student

change password

exit

3. Stop

H draw\_rectangle()

1. Start

2. Set the cursor to (0,0)

3. print corner character for rectangle

4.i=1

5. if i<78

6. set the cursor to (i,0)

7.print hrozontal double lines

8. i=i+1,goto 5

9. else set the cursor to (78,0)

10.print corner character for rectangle

11.i=1

12.if i<24

13.set the cursor to (78,i)

14.print vertical double line

15. i=i+1,goto 12

16.else set the cursor to (78,24)

17.print corner for rectangle

18.i=77

19.is i>0, if yes

20 set the cursor to (i,24)

21.print vertical double lines

22.i=i-1,goto 19

23.if no set the cursor to (0,24)

24.print corner character

25.i=23

26.is i>0 if yes

27.set the cursor to (0,i)

28.print horizontal double lines

29. i=i-1, goto 26

30.else stop

Flowchart

Figure .

Enter the password

Read pass

Is pass=password

Display all the choice and ask to enter the choice

Read choice

Is choice =1,2,3,4,5,6?

call add function

callsearch function

call modify function

call delete function

call change password function

Exit

Invalid password The database is password protected.

* + 1. Main()
    2. add\_data()

call drawrectangle function

Open record.txt file in append mode

is file=null

Display error message

write the above data in file record.txt

close the file

Ask the user to enter the ID NAME,FATHER'SNAME, MOTHER'S NAME,ADDRESS PHONENUMBER,CLASS

Figure .

* + 1. search\_data()

Figure .

Open record.txt file in read mode

is file=null

is S\_ID=ID

Display the record

No record found

Display error message

call drawrectangle function

Enter ID to search

Read S\_ID

* + 1. Modify\_data()

Figure .

call drawrectangle function

Enter ID to modify

Read S\_ID

Open record.txt file in read mode

is file=null

Display error message

is S\_ID=ID

write the above data in file record.txt and close the file

Ask the user to enter the modified records.

No records found

* + 1. delete\_data()

Figure .

is file=null

call drawrectangle function

Enter ID to delete

Read S\_ID

Open record.txt file in read mode and temp.txt in write mode

Display error message

is S\_ID=ID

continue to another record

write the data in file temp.txt and rename the file as record.txt

Figure .

close the file password.txt

call drawrectangle function

Enter the new password

Read new password

Open password.txt file in write mode

is file=null

Display error message

write new password in file password.txt

* + 1. change\_password()
    2. main\_window()

Figure .

1. Add student

2.Search student

3. Modify student record

4.Delete student record

5. Change password

6. Exit

* + 1. draw\_rectangle()

1st step

Set the cursor to (0,0)

print ("%c",201)

i=1

is i<78?

print ("%c",205)

set cursor to (i,0),i=i+1

i=i+1

A

Figure .

2nd step

A

Set the cursor to (78,0)

print ("%c",187)

i=1

is i<24?

print ("%c",186)

set cursor to (78,i)

i=i+1

B

Figure .

3rd step

B

Set the cursor to (78,24)

print ("%c",188)

i=77

is i>0?

print ("%c",205)

set cursor to (i,24)

i=i-1

C

Figure .

4th step

Set the cursor to (0,24)

print ("%c",200)

i=23

is i>0?

print ("%c",186)

set cursor to (0,i)

i=i-1

C

Figure .

# Implementation and coding

Implementation of code

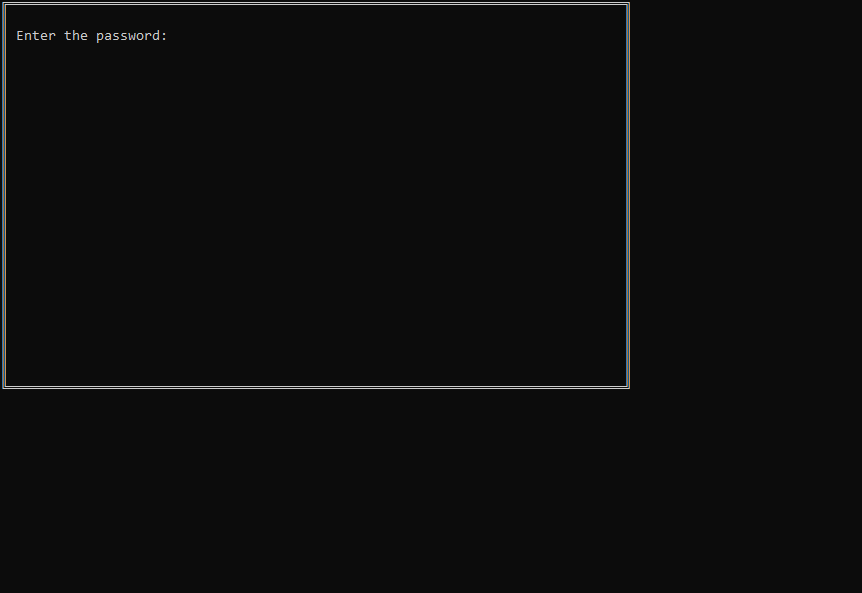
The objective of building a student record system has already been solved and planned as a problem in the above chapters. But the implementation of the plan and the idea of how the final program somehow looks like can be obtained from the below snapshots of the program. 

Figure .

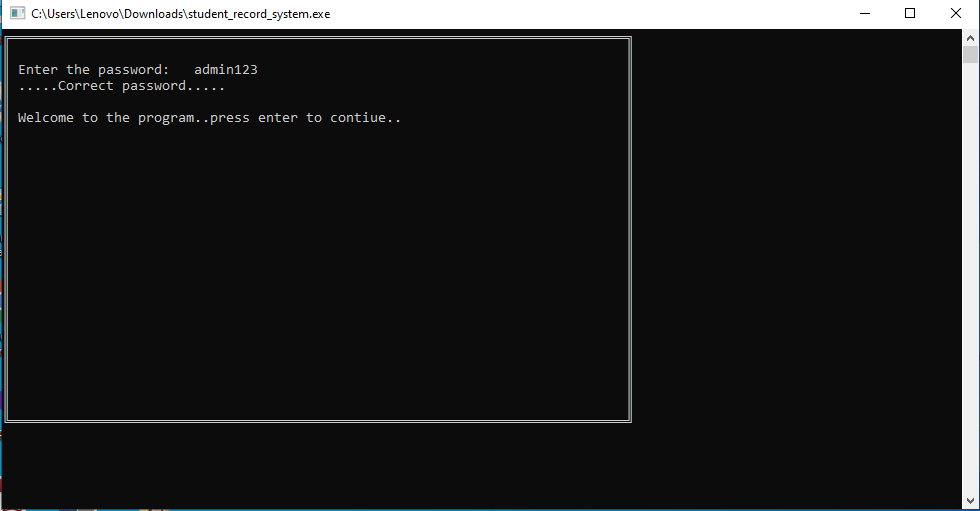
Figure 6.1 shows homepage of the program. Here the program asks the user to enter the password. The password has been set to “admin123”. If the password entered by the user matches then the program proceeds. 

Figure .

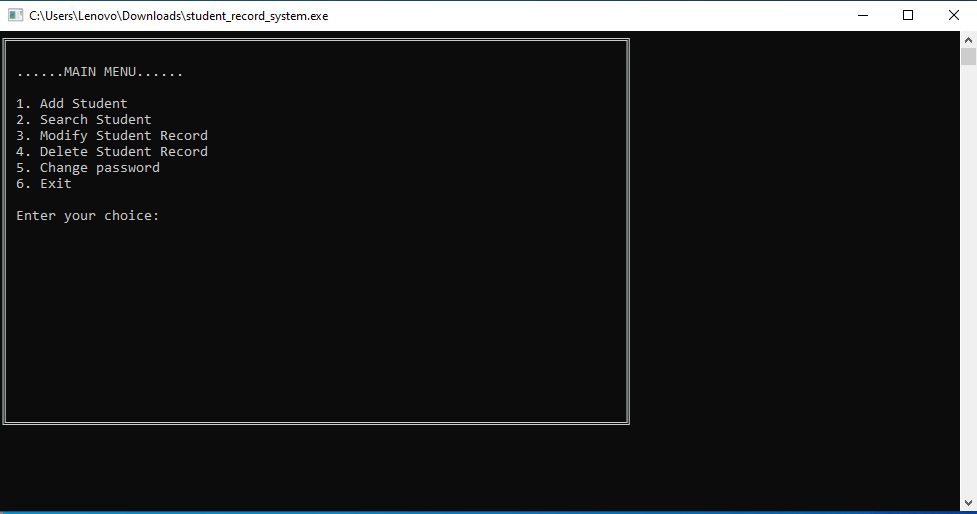
Figure 6.2 shows screen displayed after entering password. 

Figure .

Figure shows the menu of the program. Here the user can choose an option from the menu by entering the corresponding ‘number of the option’.

Coding of the project:

In this project, C programming language is used and since C is a procedure-based programming language, the procedures involved in meeting the objective of this game are there in the source code.

The functions used are

1)void gotoxy(int x, int y)

gotoxy function places cursor at a desired location on screen i.e., we can change cursor position using gotoxy function.

2) void drawRectangle()

rectangle function is used to draw a rectangle. Coordinates of left top and right bottom corner are required to draw the rectangle. left specifies the X-coordinate of top left corner, top specifies the Y-coordinate of top left corner, right specifies the X-coordinate of right bottom corner, bottom specifies the Y-coordinate of right bottom corner. Here loop is used to print characters so as to create continuous lines for rectangle.

3) void add\_data():This function is basically used for adding details of a student. In this function, a text file “record.txt” is created and used for storing the details. This function is of no argument and no return type.

4) void search\_data():This function is used to view the stored details of a student by entering the id no. For this, “record.txt” file is opened in read mode. The contents are read and displayed if matched.

5) void modify\_data():This function is used to modify the stored details of a student by entering the id no. For this, “record.txt” file is opened in read mode. The new contents are to be typed.

6) void delete\_data():This function is used for deleting the stored information.User needs to enter the id no of the student to delete the details. This is also a user defined function of no argument and no return type.

7) Void main\_window(): this function is used for displaying main window. It displays the options among which the user has to select one for performing necessary task.

8) void change\_password(): this function is used to change password . a screen is displayed asking the user to type new password.

# Result and discussion

The programs flowchart, algorithms are all written and according to the flowchart the program code was written. The implementation and coding of program was done.

Now the result of the program was observed.

To get the idea about the results the following snapshots are used:

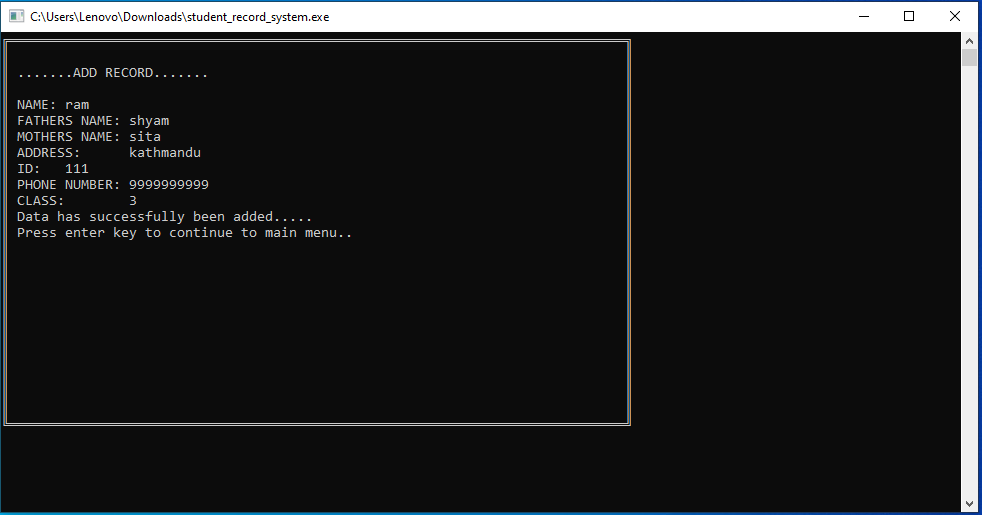


Figure .

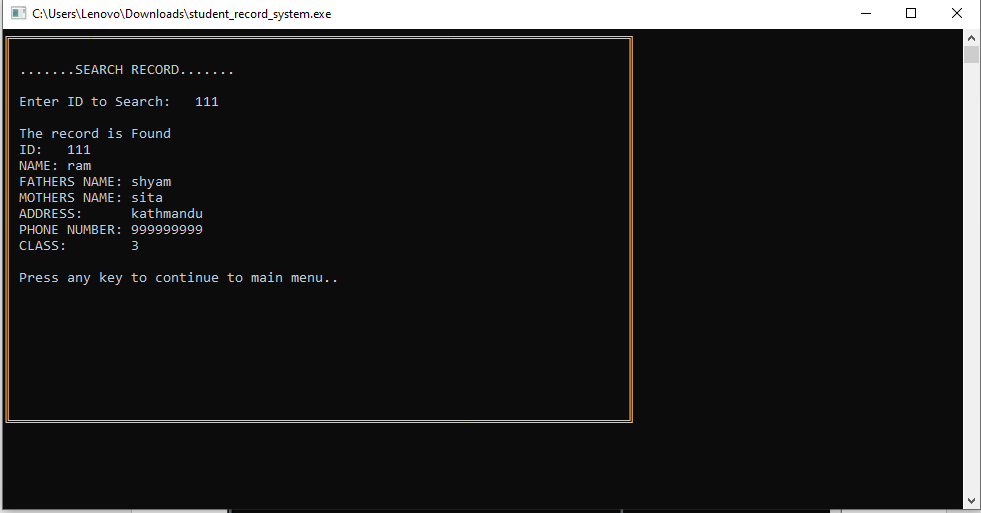
After selecting option 1 in main menu, add\_data() function is implemented and user needs to add necessary data. The screen displayed after entering the data is shown abovein fig 7.1: 

Figure .

After selecting option 2 in main menu search\_data() is implemented and after entering id no, user can search and look at the record as shown in fig7.2

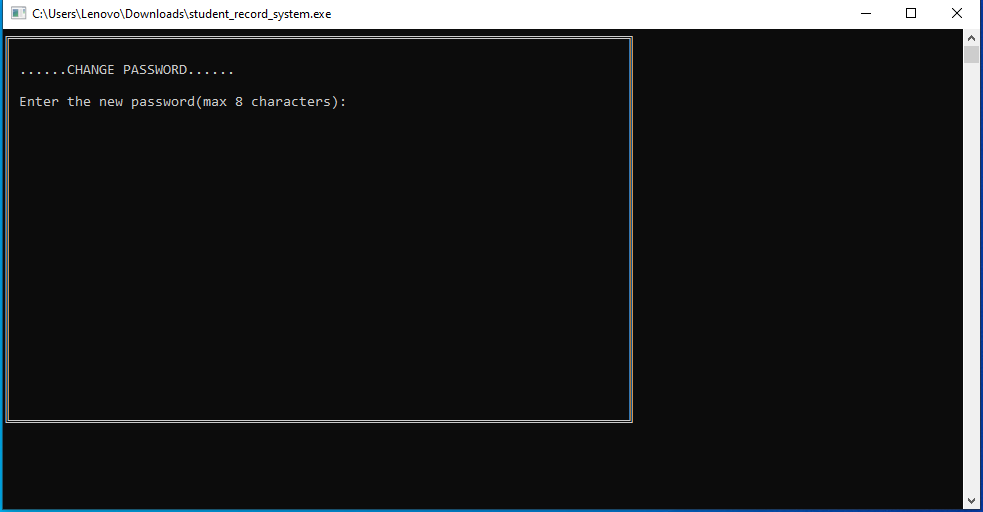


Figure .

After option 5 is selected, change\_password() is implemented and user can enter new password. The above Figure shows screen displayed to enter new password.

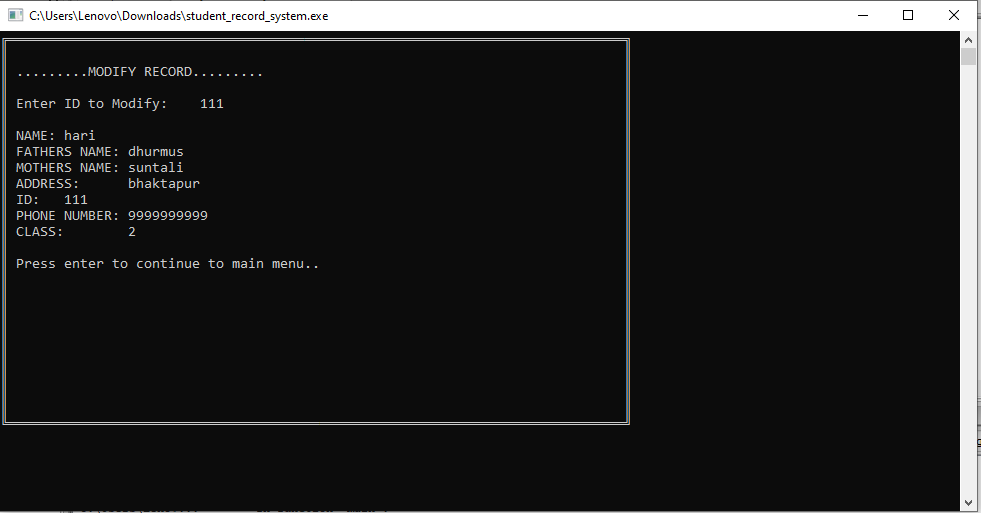


Figure .

After option 3 is selected modify\_data() is implemented and user can modify the data. The above Figure shows screen displayed after user entered option to modify record and has modified it .

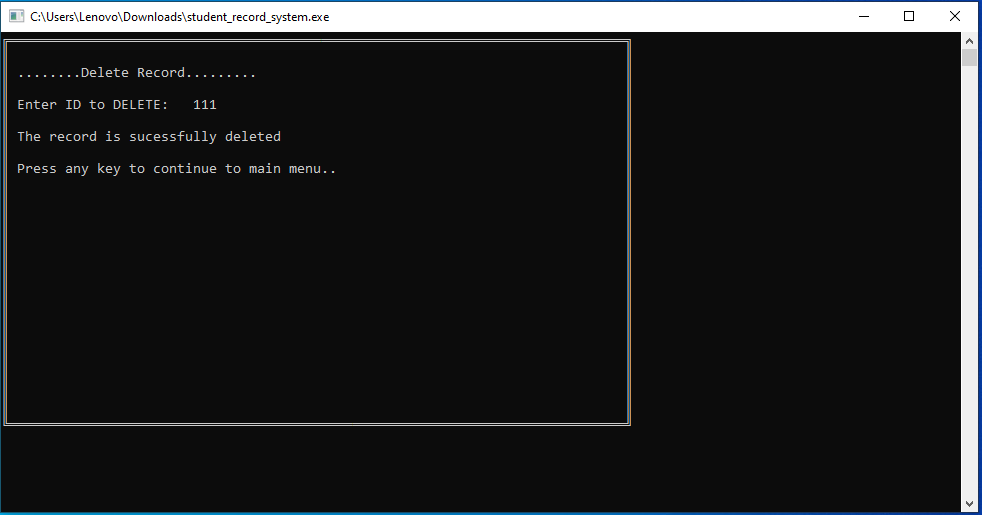


Figure .

After option 4 is selected in main menu delete\_data() is implemented and user can delete specific record. The above Figure shows screen displayed after selecting delete option in main menu and entering id no to delete record.

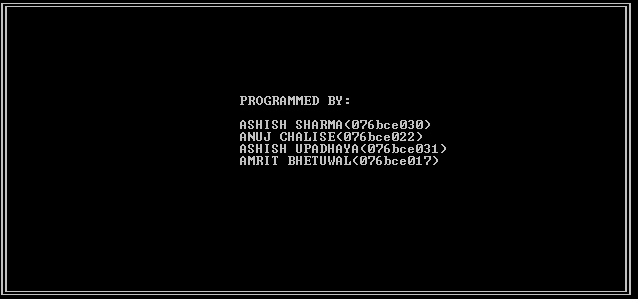


Figure .

The above figure shows the screen when exit option is selected.

# Conclusion

After the testing and running of this program following conclusions can be made:  
  
1. This program enables to store information like name ,address ,phone number and class number of large number of students.  
2. This program also allows to modify and add relevant data according to the needs of user.  
3. This program demonstrates how structures and file handling can be used to store large amount of data in an organized manner in C.   
4. This program also shows how to use various control structures and string handling functions to create a program in C.   
5. This program also shows how various techniques can be used to make the program visually appealing.

# Appendix

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<windows.h>

FILE \*fp;

struct student

{

char name[30],address[30],f\_name[30],m\_name[30],id[9];

long int phn\_n;

int class\_no;

}stu;

COORD coord = {0,0}; ///set the cordinate to 0, 0 (top-left corner of window);

void gotoxy(int x, int y){

coord.X = x; coord.Y = y; /// X and Y coordinates

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), coord);

}

void drawRectangle(){

int i, j;

gotoxy(0,0);

printf("%c",201);

for(i = 1; i < 78; i++){

gotoxy(i, 0);

printf("%c",205);

}

gotoxy(78,0);

printf("%c",187);

for(i = 1; i < 24; i++){

gotoxy(78, i);

printf("%c",186);

}

gotoxy(78, 24);

printf("%c",188);

for(i = 77; i > 0; i--){

gotoxy(i,24);

printf("%c",205);

}

gotoxy(0,24);

printf("%c",200);

for(i = 23; i > 0; i--){

gotoxy(0,i);

printf("%c",186);

}

}

void add\_data()

{

system("cls");

drawRectangle();

gotoxy(2,2);printf(".......ADD RECORD.......");

getchar();

fp= fopen("record.txt","ab+");

if(fp==NULL)

{

printf("unable to open file....");

exit(1);

}

else

{

gotoxy(2,4); printf("NAME:\t"); gets(stu.name);

gotoxy(2,5); printf("FATHERS NAME:\t"); gets(stu.f\_name);

gotoxy(2,6);printf("MOTHERS NAME:\t"); gets(stu.m\_name);

gotoxy(2,7);printf("ADDRESS:\t"); gets(stu.address);

gotoxy(2,8);printf("ID:\t"); gets(stu.id);

gotoxy(2,9); printf("PHONE NUMBER:\t"); scanf("%ld",&stu.phn\_n);

gotoxy(2,10);printf("CLASS:\t"); scanf("%d",&stu.class\_no);

fwrite(&stu, sizeof(stu), 1, fp);

gotoxy(2,11);printf("Data has successfully been added.....");

gotoxy(2,12);printf("Press enter key to continue to main menu..");

getchar();

getchar();

fclose(fp);

return;

}

}

void search\_data()

{

system ("cls");

drawRectangle();

gotoxy(2,2);printf(".......SEARCH RECORD.......");

getchar();

char s\_id[15];

int isFound = 0;

gotoxy(2,4);printf("Enter ID to Search:\t");

gets(s\_id);

fp = fopen("record.txt","rb");

if(fp==NULL)

{

printf("unable to open file....");

exit(1);

}

while(fread(&stu,sizeof(stu),1,fp)==1)

{

if(strcmp(s\_id,stu.id) == 0)

{

isFound = 1;

break;

}

}

if(isFound == 1)

{

gotoxy(2,6);printf("The record is Found");

gotoxy(2,7);printf("ID:\t%s",stu.id);

gotoxy(2,8);printf("NAME:\t%s",stu.name);

gotoxy(2,9);printf("FATHERS NAME:\t%s",stu.f\_name);

gotoxy(2,10);printf("MOTHERS NAME:\t%s",stu.m\_name);

gotoxy(2,11);printf("ADDRESS:\t%s",stu.address);

gotoxy(2,12);printf("PHONE NUMBER:\t%ld",stu.phn\_n);

gotoxy(2,13);printf("CLASS:\t%d",stu.class\_no);

gotoxy(2,15);printf("Press any key to continue to main menu..");

getchar();

fclose(fp);

return;

}

else

{

gotoxy(2,4); printf("Sory, No record found in the database");

gotoxy(2,6);printf("Press any key to continue to main menu..");

getchar();

}

fclose(fp);

return;

}

void modify\_data()

{

system ("cls");

drawRectangle();

gotoxy(2,2);printf(".........MODIFY RECORD.........");

getchar();

char s\_id[15];

int isFound = 0;

gotoxy(2,4); printf("Enter ID to Modify:\t ");

gets(s\_id);

fp = fopen("record.txt","rb+");

if(fp==NULL)

{

printf("unable to open file....");

exit(1);

}

while(fread(&stu,sizeof(stu),1,fp)==1)

{

if(strcmp(s\_id, stu.id) == 0)

{

gotoxy(2,6);printf("NAME:\t"); gets(stu.name);

gotoxy(2,7);printf("FATHERS NAME:\t"); gets(stu.f\_name);

gotoxy(2,8);printf("MOTHERS NAME:\t"); gets(stu.m\_name);

gotoxy(2,9);printf("ADDRESS:\t"); gets(stu.address);

gotoxy(2,10);printf("ID:\t"); gets(stu.id);

gotoxy(2,11);printf("PHONE NUMBER:\t"); scanf("%ld",&stu.phn\_n);

gotoxy(2,12);printf("CLASS:\t"); scanf("%d",&stu.class\_no);

fseek(fp,-sizeof(stu), SEEK\_CUR);

fwrite(&stu,sizeof(stu), 1, fp);

getchar();

gotoxy(2,14);printf("Press enter to continue to main menu..");

getchar();

isFound = 1;

break;

}

}

if(!isFound)

{

gotoxy(2,6); printf("No Record Found");

gotoxy(2,8);printf("Press enter to continue to main menu..");

getchar();

}

fclose(fp);

return;

}

void delete\_data()

{

system("cls");

drawRectangle();

gotoxy(2,2);printf("........Delete Record.........");

getchar();

char s\_id[15];

gotoxy(2,4);printf("Enter ID to DELETE:\t");

gets(s\_id);

FILE \*temp;

fp = fopen("record.txt","rb");

temp = fopen("temp.txt", "wb");

if(fp==NULL||temp==NULL)

{

printf("unable to open file....");

exit(1);

}

else

{

while(fread(&stu, sizeof(stu),1,fp)==1)

{

if(strcmp(s\_id, stu.id) == 0)

{

continue;

}

else

fwrite(&stu,sizeof(stu),1,temp);

}

fclose(fp);

fclose(temp);

remove("record.txt");

rename("temp.txt","record.txt");

gotoxy(2,6); printf("The record is sucessfully deleted");

gotoxy(2,8);printf("Press any key to continue to main menu..");

getchar();

}

return;

}

void main\_window()

{

gotoxy(2,2); printf("......MAIN MENU......");

gotoxy(2,4); printf("1. Add Student");

gotoxy(2,5); printf("2. Search Student");

gotoxy(2,6); printf("3. Modify Student Record");

gotoxy(2,7); printf("4. Delete Student Record");

gotoxy(2,8); printf("5. Change password");

gotoxy(2,9);printf("6. Exit");

return;

}

void change\_password()

{ system("cls");

drawRectangle();

char new\_pass[8];

gotoxy(2,2); printf("......CHANGE PASSWORD......");

getchar();

gotoxy(2,4);printf("Enter the new password(max 8 characters):\t");

gets(new\_pass);

fp=fopen("password.txt","wb");

fwrite(&new\_pass,sizeof(new\_pass),1,fp);

fclose(fp);

gotoxy(2,6);printf("The password has been successfully changed.");

gotoxy(2,8);printf("Press enter to continue to main menu..");

getchar();

return;

}

int main()

{

drawRectangle();

gotoxy(2,2);

int choice, check;

char pass[9],password[9]="Wrong\_value";

fp=fopen("password.txt","rb");

fgets(password,sizeof(password),fp);

fclose(fp);

printf("Enter the password:\t");

scanf("%s",&pass);

if (strcmp(pass,password)==0)

{

gotoxy(2,3) ;printf(".....Correct password.....\n");

gotoxy(2,5);printf("Welcome to the program..press enter to contiue..");

getchar();

getchar();

system("cls");

program\_loop:

main\_window();

drawRectangle();

gotoxy(2,11);printf("Enter your choice: ");

scanf("%d",&choice);

switch(choice)

{

case 1:

add\_data();

system("cls");

break;

case 2:

search\_data();

system("cls");

break;

case 3:

modify\_data();

system("cls");

break;

case 4:

delete\_data();

system("cls");

break;

case 5:

change\_password();

system("cls");

break;

case 6:

system("cls");

drawRectangle();

getchar();

gotoxy(30,8); printf("PROGRAMMED BY:");

gotoxy(30,10);printf("ASHISH SHARMA(076bce030)");

gotoxy(30,11); printf("ANUJ CHALISE(076bce022)");

gotoxy(30,12);printf("ASHISH UPADHAYA(076bce031)");

gotoxy(30,13);printf("AMRIT BHETUWAL(076bce017)");

getchar();

Beep(750,300);

gotoxy(2,19);

exit (0);

break;

default:

break;

}

goto program\_loop;

}

else

{

getchar();

gotoxy(2,3); printf(".........Invalid Password.........");

gotoxy(2,5);printf("The database is password protected.");

gotoxy(2,7);printf("Press enter to terminate the program.");

getchar();

gotoxy(3,9);

exit(1);

}

return 0;

system("pause");

}