UNIVERSITY PARTNER
UNIVERSITY OF WOLVERHAMPTON
iAcademy
Quality Education Improving Lives

HERALD COLLEGE KATHMANDU

Academic Year	Module	Assessment Number	Assessment Type
A19	Introduction to Programming I(DipIT04)	A2	Group Report

Student Record Management System

Student Id : NP03A190031
Student Name : Ankit Tamrakar
Student Id : NP03A190059
Student Name : Ashmin Gurung
Section : DC8
Module Leader : [Module Leader Name]
Lecturer : Subiran Shrestha
Group Name : Group 23

Submitted on : 15-02-2020

MARKING SCHEME

S. N	Title	Full marks	Obtained Marks	Comments		
1	Format of Report [Report title, Module title,Student names,Group Name]	2				
2	Abstract	2				
3	Acknowledgment	2				
4	Table of contents and list of figures	2				
5	Project description and applications	3				
6	Project algorithm, pseudocode and flowchart	10				
7	C Source Code	3				
8	Testing with screenshots	3				
9	Conclusion and References	3				
10	**Continuous Assessments (5 %)**			**Total Marks Obtained**		
	Ankit Tamrakar NP03A190031	15		**0**		
	Ashmin Gurung NP03A190059	15		**0**		
Full Marks		35				

ABSTRACT

We were given a task on creating a project that performs CRUD (create, read, update and delete) operations. The main objective of our project is to build a student database system that will store and handle student records. We aim to reduce time taken to perform administrative tasks. The system is designed to accurately create, read, update and delete information about students.

The system is designed with C which means it will run in any computers with minimal specifications. We've tried our best to make the program interface as friendly as possible. Our goal is to offer user efficient and less time-consuming application.

ACKNOWLEDGEMENT

We would like to thank Herald College for providing us with opportunity to learn C programming language and develop a program based on it. We would also like to thank our module leader, Mr. Raj Prasad Shrestha for providing quality lecture slides and study materials. Huge thanks to Mr. Subiran Shrestha and Mr. Rizzu Rohit Bhandari for providing us with their professional guidance and feedback on our project.

Sincerely,
Ankit Tamrakar
Ashmin Gurung

CONTENTS

TABLE OF FIGURES

1. INTRODUCTION

1.1 General Introduction

Student Record Management Sysytem is a simple console application without graphics, developed with the help of C programming language. It is compiled in Dev C++.This project utilizes various aspects of C language such as data structure, file handling, functions, pointers and arrays. With this application, users can create, read, update and delete information of students.

Features of our project: -

a) Lightweight and easily accessible.
b) Less Human Error
c) Easy to handle
d) Easy data updating
e) Easy record keeping

Our project fully utilizes void functions and structures to break down the whole program into further smaller parts. File pointer and handling is used to create, read, update and edit data stored in txt file.

Program is structured into small chunks in such way that other developers can easily understand, improve and debug the program.

1.2 Requirements:

Hardware requirements: -

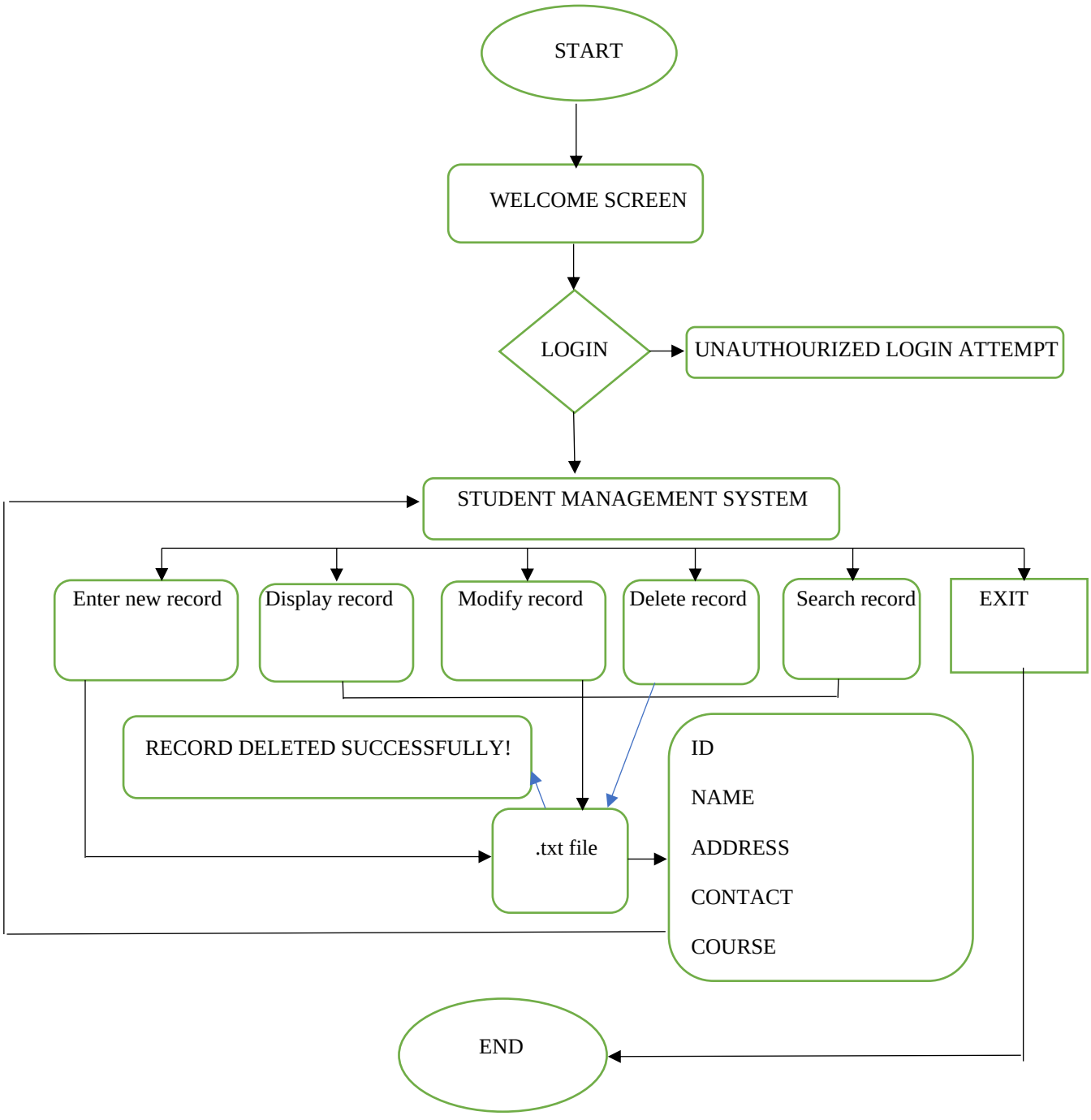
- 512 MB RAM
- Basic GPU – Any vendor
- 1024x768 or higher-resolution monitor
- Mouse or any other pointing device
- Keyboard

Software Requirements: -

- Windows XP/Vista/7/8/10
- DirectX 9.0 class or better (Pixel Shader Level 2)

2. PROJECT DESIGN

Flowchart: -



3. C SOURCE CODE

```
#include<stdio.h>

#include <stdlib.h>


void returnfunc(void);

void mainmenu(void);

void addstudent(void);

void deletestudent(void);

void editstudent(void);

void searchstudent(void);

void viewstudent(void);

void closeapplication(void);

void Password(void);
//void functions-breaking programs into smaller chunks.


FILE *fp,*ft,*fs;
//file-handling


int s;
char findstudent;
//globalvariables


struct student
{

int id;
char name[20];
```

```
char address[20];
char course[10];
int contact;

};
//defining primary structure of how data stores inside our txt file


struct student a;

int main()
{
Password();
getch();
return 0;
}
//login authentication


void mainmenu()
//mainemenu for the program
{

system("cls");

int i;
printf("\t\t\tStudent Management System \n");
printf("<1> Enter or Add Record \n");
printf("<2> Delete Records\n");
printf("<3> Search Records\n");
printf("<4> View Record's list\n");
printf("<5> Modify Record\n");
printf("<6> Close Application\n\n");
printf("Enter your choice:\n");

switch(getch())
//linking parts together into a single menu
{

case '1':
addstudent();
break;

case '2':
```

```
deletestudent();
break;

case '3':
searchstudent();
break;

case '4':
viewstudent();
break;

case '5':
editstudent();
break;

case '6':
{
exit(0);
}
default:
{

printf("\aInvalid Entry!!Please enter correct option");

if(getch())
mainmenu();
//use of recursion
}

}

}



void addstudent(void)

{
        FILE *fptr;
        fptr=fopen("stf.txt","a");//append mode
        printf("\nEnter ID of student:");
        scanf("%d",&a.id);
        printf("Enter name of student:");
        scanf("%s",a.name);
```

```
        printf("\nEnter address of student:");
        scanf("%s",a.address);
        printf("\nEnter contact info of student:");
        scanf("%d",&a.contact);
        printf("\nEnter course of student:");
        scanf("%s",a.course);
        fprintf(fptr,"%d %s %s %d %s\n",a.id,a.name,a.address,a.contact,a.course);
        fclose(fptr);

printf("The record is sucessfully saved\n");
printf("Save any more?(Y / N):");
if(getch()=='n')
mainmenu();
else
system("cls");
addstudent();//recursion

}



void deletestudent()

{
                FILE *fptr;
        fptr=fopen("stf.txt","r");
        struct student s[100];
        int count=0,i=0,choice;
        while(!feof(fptr))//until end of file
        {
                fscanf(fptr,"%d %s %s %d
%s\n",&s[i].id,s[i].name,s[i].address,&s[i].contact,s[i].course);

                printf("%d %s %s %d
%s\n",s[i].id,s[i].name,s[i].address,s[i].contact,s[i].course);
                count++;//to count total no. of 'students' present in the data file.
                i++;

        }
        fclose(fptr);
        printf("Enter the ID of student you want to delete\n");
        scanf("%d",&choice);
        fptr=fopen("stf.txt","w");
        for(i=0;i<count;i++)
```

```
	{
		if(s[i].id==choice)
		{
			continue;//w mode deletes(overwrites) data with matched id
		}
		else
		{

		fprintf(fptr,"%d %s %s %d
%s\n",s[i].id,s[i].name,s[i].address,s[i].contact,s[i].course);
		}
	}
		fclose(fptr);

printf("Record Successfully Deleted\n");
printf("\n\tDelete another record?(Y/N)");
if(getch()=='n')
mainmenu();
else
system("cls");
deletestudent();//recursion
}



void searchstudent()

{

system("cls");

FILE *fptr;
fptr=fopen("stf.txt","r");
struct student s[100];
int count=0,i=0,choice,loop;
char t;

system("cls");

printf("Enter the ID:");
scanf("%d",&choice);
while(!feof(fptr))
{
```

```
fscanf(fptr,"%d %s %s %d
%s",&s[i].id,s[i].name,s[i].address,&s[i].contact,s[i].course);
                                if(s[i].id==choice)
                                {
                                        printf("The record is available");
                        printf("\n ID \t\t NAME \t\tADDRESS \t\t CONTACT \t\t
COURSE\n");
                        printf("%d\t\t %s \t\t%s\t\t\t %d\t\t
%s\n",s[i].id,s[i].name,s[i].address,s[i].contact,s[i].course);
                        loop='t';
                i++;
}
        }
fclose(fptr);


if (loop!='t')
{
printf("Record Not Found\n");
}
printf("Try another search?(Y/N)");
if(getch()=='y')
searchstudent();
else
mainmenu();


}









void viewstudent(void)

{

system("cls");

printf("\n ID \t\t NAME \t\tADDRESS \t\t CONTACT \t COURSE\n");
```

```
	FILE *fptr;
	fptr=fopen("stf.txt","r");
	struct student s[100],update;
	int count=0,i=0,choice;
	while(!feof(fptr))
	{
		fscanf(fptr,"%d %s %s %d
%s\n",&s[i].id,s[i].name,s[i].address,&s[i].contact,s[i].course);

		printf("%d\t\t %s \t\t%s\t\t\t %d\t\t
%s\n",s[i].id,s[i].name,s[i].address,s[i].contact,s[i].course);
		count++;
		i++;

	}

returnfunc();

}



void editstudent(void)

{

system("cls");
printf("\nEdit Records\n");


{
	FILE *fptr;
	fptr=fopen("stf.txt","r");
	struct student s[100],update;
	int count=0,i=0,choice;
	while(!feof(fptr))
	{
		fscanf(fptr,"%d %s %s %d
%s\n",&s[i].id,s[i].name,s[i].address,&s[i].contact,s[i].course);

		printf("%d %s %s %d
%s\n",s[i].id,s[i].name,s[i].address,s[i].contact,s[i].course);
		count++;
		i++;
```

```
	}
	fclose(fptr);
	printf("Enter the ID of student you want to update\n");
	scanf("%d",&choice);
	printf("Enter the details of student again");
	printf("\nEnter ID of student:");
	scanf("%d",&update.id);
	printf("Enter name of student:");
	scanf("%s",update.name);
	printf("\nEnter address of student:");
	scanf("%s",update.address);
	printf("\nEnter contact info of student:");
	scanf("%d",&update.contact);
	printf("\nEnter course of student:");
	scanf("%s",update.course);
	fptr=fopen("stf.txt","r+");
	for(i=0;i<count;i++)
	{
		if(s[i].id==choice)
		{
			fprintf(fptr,"%d %s %s %d
%s\n",update.id,update.name,update.address,update.contact,update.course);
			printf("Successfully Updated\n");//strucure 'update' to temporarily
hold data
		}
		else
		{
			fprintf(fptr,"%d %s %s %d
%s\n",s[i].id,s[i].name,s[i].address,s[i].contact,s[i].course);
		}
	}
		fclose(fptr);



printf("Edit any more?(Y / N):");

if(getch()=='n')

mainmenu();

else
```

```
system("cls");

editstudent();

}



returnfunc();

}

void returnfunc(void)

{

{

printf("Press ENTER to return to main menu");

}

a:

if(getch()==13)

mainmenu();

else

goto a;

}


void Password(void)

{

char password[10]={"pass"};
system("cls");
```

```
char ch,pass[10];
int i=0,j;

        printf("\t\t\t\tWelcome\n\t\t\t\t  To \n\t\t\t  College Management System \n");
        printf("\t \n\n\n Enter Password:");

while(ch!=13)
{
ch=getch();

if(ch!=13 && ch!=8){

putch('*');

pass[i] = ch;

i++;

}

}

pass[i] = '\0';

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

{
printf("\n\n\n\t\tPassword matched!!");
printf("\n\n\tPress any key to continue.....");
getch();
mainmenu();
}

else

{
printf("\n\n\n\t\t\aWarning!! \n\t   Incorrect Password");
getch();
Password();
}

}
```

4. IMPLEMENTATION

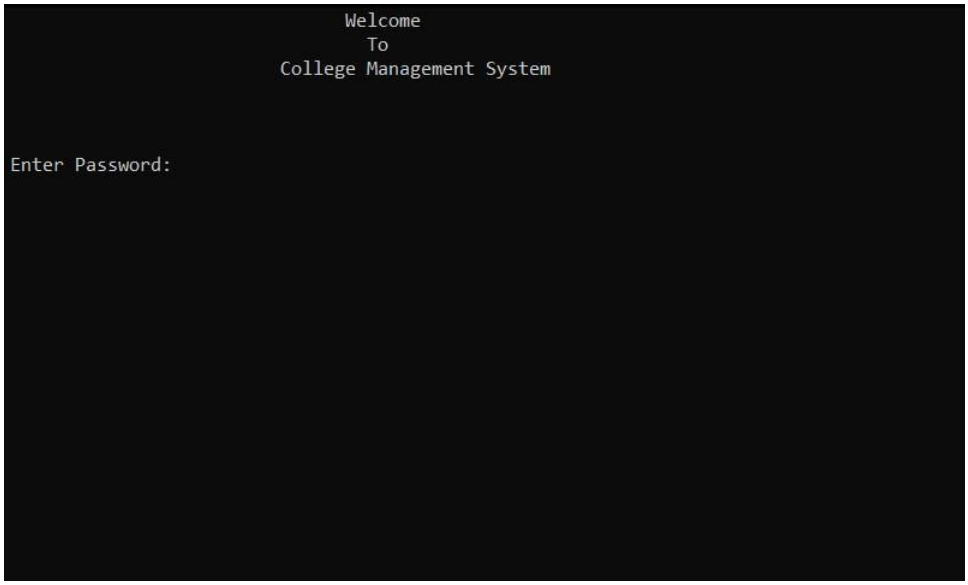


Figure 1: Login Page

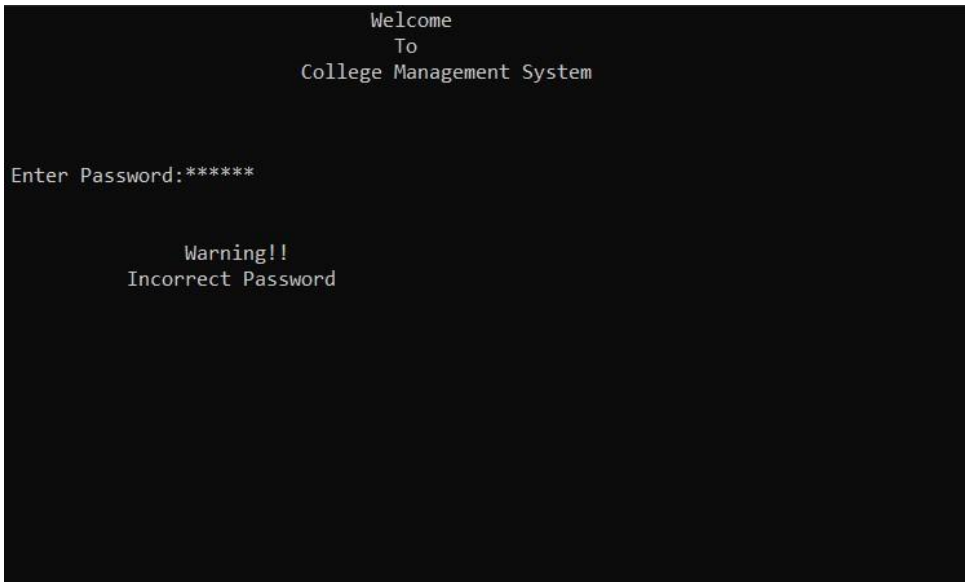


Figure 2: Wrong Password

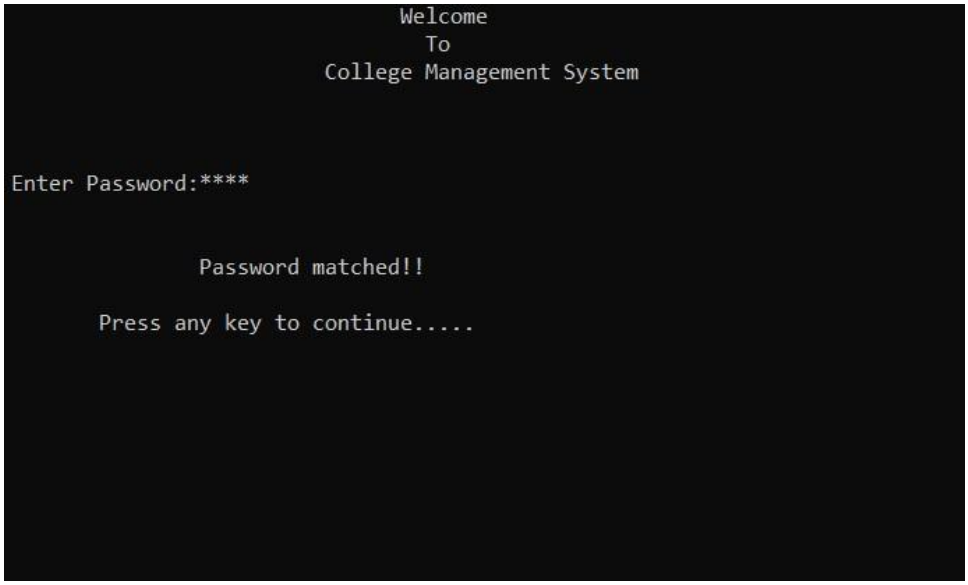


Figure 3: Password Matched!

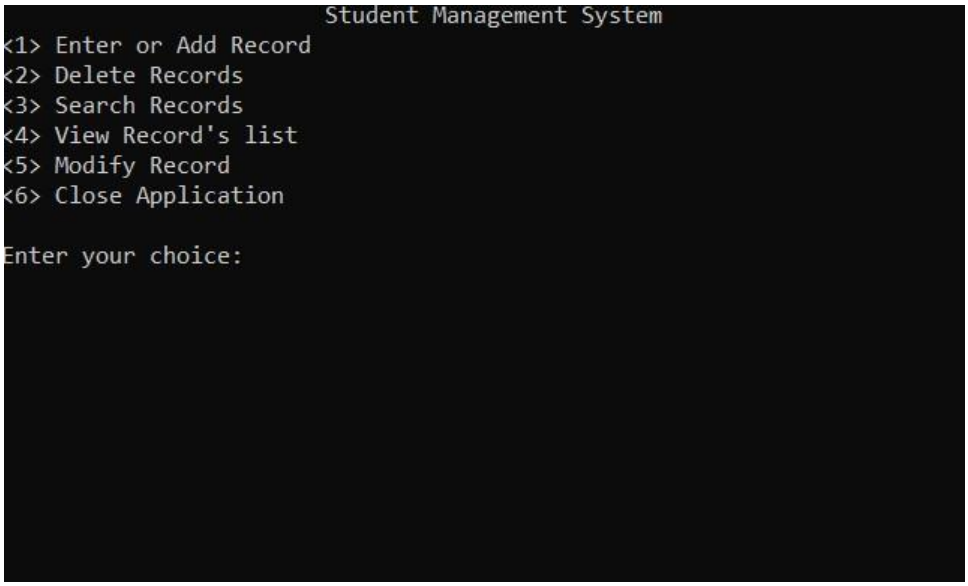


Figure 4: Main menu

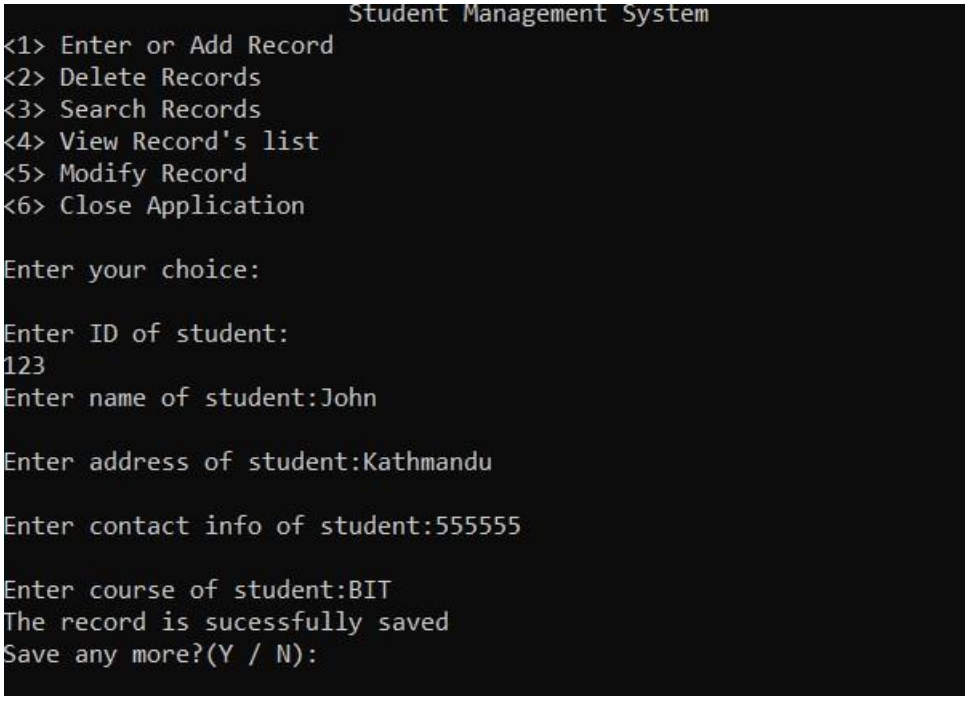


Figure 5: Add New Record

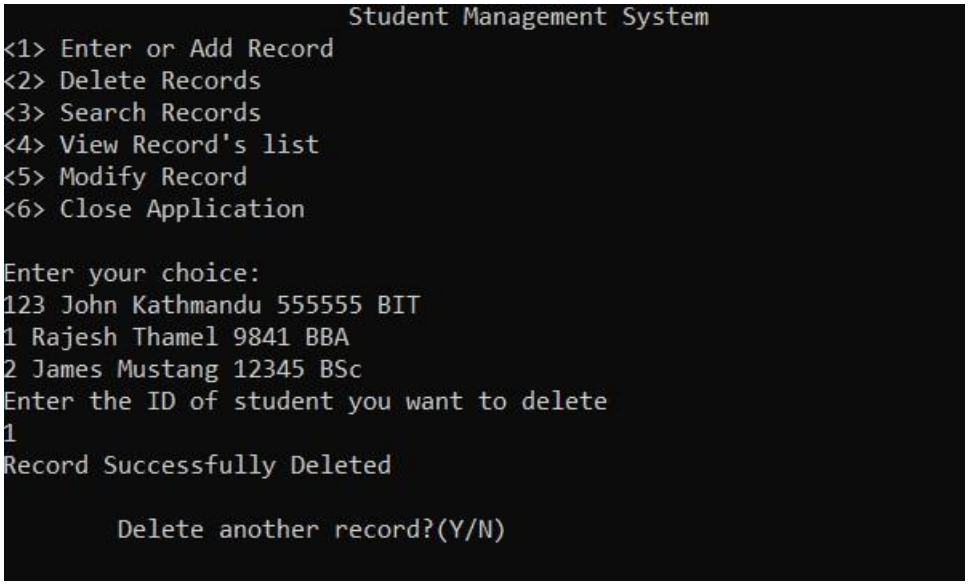


Figure 6: Delete Existing Record

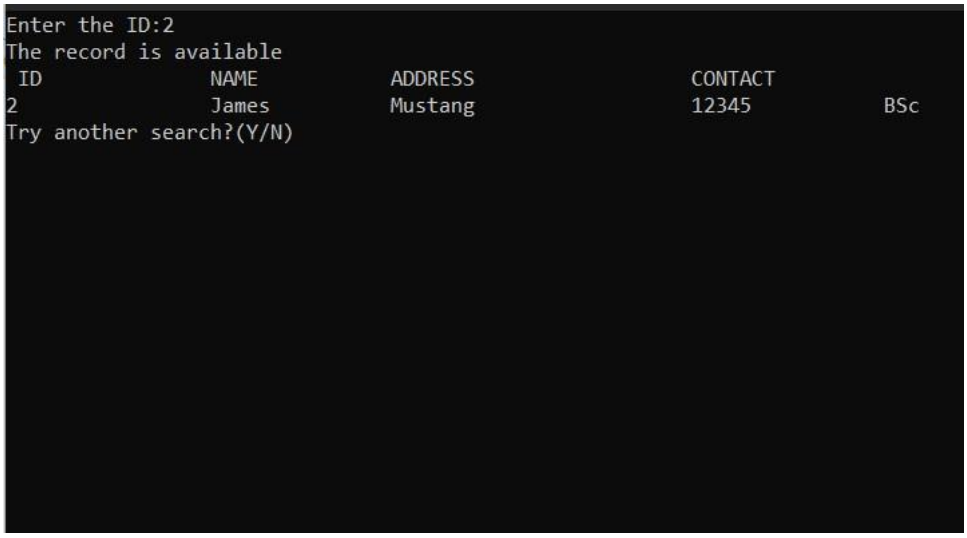


Figure 7: Search Existing Record

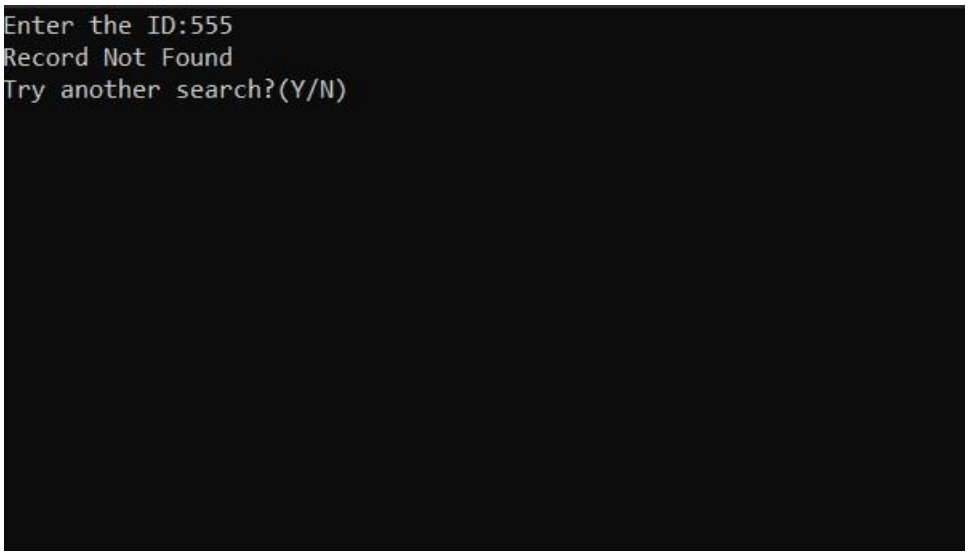


Figure 8: Record Not Found

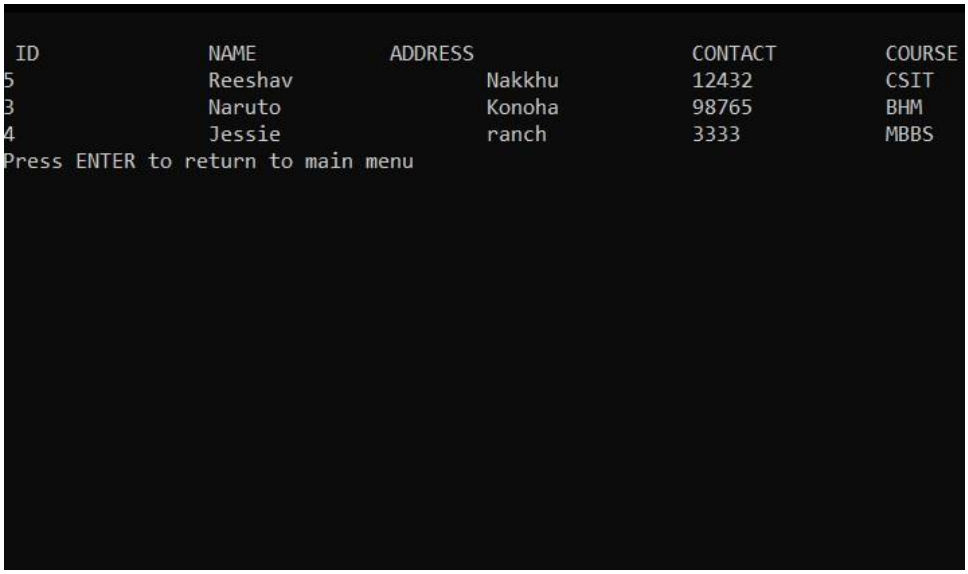


Figure 9: View Existing Records

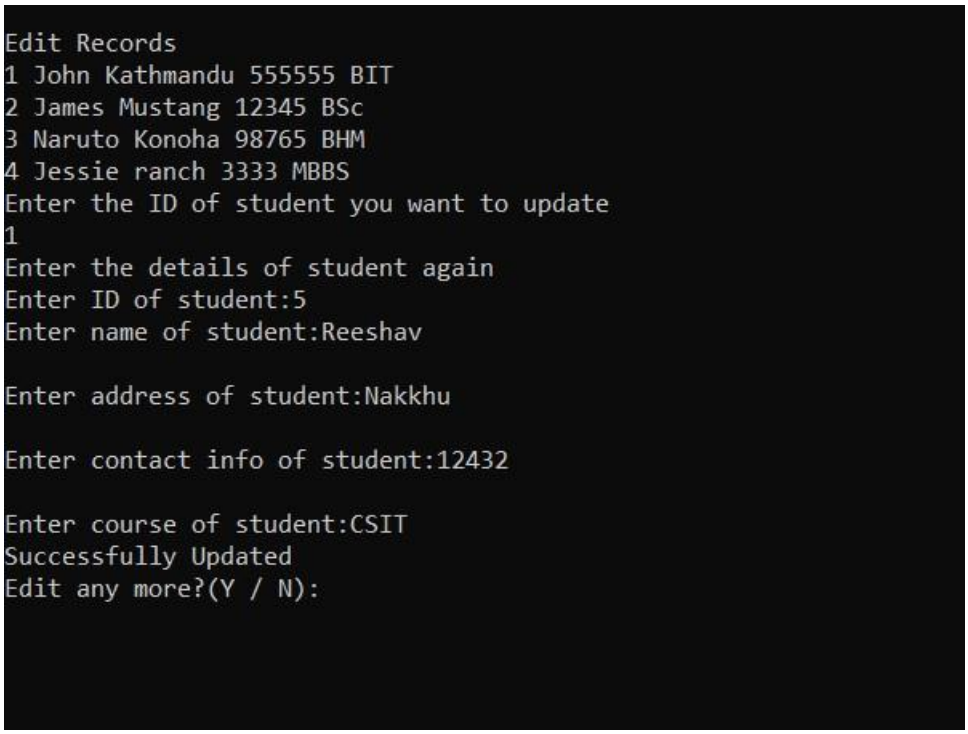


Figure 10: Edit New Record

5. CONCLUSION

The project turned out pretty good than what we had expected. There is always a room for further improvement. It can be made better by adding some GUI components and turning the program visually appealing. We believe it can be modified according to the need and can be used to manage any record or database system.

6. References

Anon., n.d. *Learn C (Introduction and Tutorials to C Programming).* [Online]
Available at: https://www.programiz.com/c-programming
[Accessed 10 02 2020].
Kerninghan, B. & Ritchie, D., 1988. *The C Programming Language.* s.l.:Prentice-Hall.
Kochan, S. G., 2005. *Programming in C, Third Edition.* s.l.:Sams Publishing.