1. INTRODUCTION

This database for students is designed keeping in mind the needs of managing a student's database. Both at the user level and product level it provides high quality security. The design of this system includes two kinds of users that is students and administration.

A typical flow of events can be seen in this database. Its design is created as per the point of view of the students. As this system is based around the central that is students therefore the records in the database also revolve around the activities of the students. There are many other independent categories of this data system that includes Fest, cultural activities, alumni association, library facility and sports.

These information's helps to know about the progression that is achieved by the institution. Special attention can be given therefore to all the students. This database will also help to know the performance of the students. Plus is the organization can out it on the system of LAN then it can add more efficiency and flexibility to the staffs of the organization.

Student profile includes address, details of the study, details of the dependent and admission details as well. Attendance details include the number classes that are attended by the students in each subject and the total class numbers. It also handles the information of examinations and internal assessments. It takes the informations of the exams and 3 internal assessment details.

- Manipulation of the database includes:
- Modifying results of examinations and attendance of students
- Deleting profiles of students
- Adding details of the department and its corresponding subjects
- Adding profiles of students
- Generation of reports includes
- Students list in each of the department
- Attendance list of students according to the different departments
- Status of examination as per the departments

Taking the considerations that are stated above we have designed a system for keeping the records of the number of students, their examination details and attendance details.

2. SYSTEM ANALYSIS & DESIGN

2.1 Requirement Specification :-

2.1.1 Hardware Requirements :

1. AMD processors 4000+ series.
2. 256MB RAM.
3. 1 Gb hard free drive space.

2.1.2 Software Requirements :

1. Operating system: Windows 10 or any other version.

2. Compiler: Dev C++.

2.2 Pseudo Code

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
#include <stdlib.h>
#include <windows.h>

struct student{
   char ID[15];
   char name[20];
   char add[20];
   char parname[20];
   int Class;
   long unsigned int phone_no;
};

struct student stu;

///This will set the forground color for printing in a console window.
void SetColor(int ForgC)
{
    WORD wColor;
    ///We will need this handle to get the current background attribute
    HANDLE hStdOut = GetStdHandle(STD_OUTPUT_HANDLE);
    CONSOLE_SCREEN_BUFFER_INFO csbi;

    ///We use csbi for the wAttributes word.
    if(GetConsoleScreenBufferInfo(hStdOut, &csbi))
    {
      ///Mask out all but the background attribute, and add in the forgournd color
```

```
            wColor = (csbi.wAttributes & 0xF0) + (ForgC & 0x0F);
            SetConsoleTextAttribute(hStdOut, wColor);
        }
        return;
    }

    void ClearConsoleToColors(int ForgC, int BackC)
    {
        WORD wColor = ((BackC & 0x0F) << 4) + (ForgC & 0x0F);
        ///Get the handle to the current output buffer...
        HANDLE hStdOut = GetStdHandle(STD_OUTPUT_HANDLE);
        ///This is used to reset the carat/cursor to the top left.
        COORD coord = {0, 0};
        ///A return value... indicating how many chars were written
        ///   not used but we need to capture this since it will be
        ///   written anyway (passing NULL causes an access violation).
        DWORD count;
        ///This is a structure containing all of the console info
        /// it is used here to find the size of the console.
        CONSOLE_SCREEN_BUFFER_INFO csbi;
        ///Here we will set the current color
        SetConsoleTextAttribute(hStdOut, wColor);
        if(GetConsoleScreenBufferInfo(hStdOut, &csbi))
        {
            ///This fills the buffer with a given character (in this case 32=space).
            FillConsoleOutputCharacter(hStdOut, (TCHAR) 32, csbi.dwSize.X *
csbi.dwSize.Y, coord, &count);
            FillConsoleOutputAttribute(hStdOut, csbi.wAttributes, csbi.dwSize.X *
csbi.dwSize.Y, coord, &count );
            ///This will set our cursor position for the next print statement.
            SetConsoleCursorPosition(hStdOut, coord);
        }
```

```
    return;
}

void SetColorAndBackground(int ForgC, int BackC)
{
    WORD wColor = ((BackC & 0x0F) << 4) + (ForgC & 0x0F);;
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), wColor);
    return;
}

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 < 25; i++){
      gotoxy(78, i);
      if(i == 6){
         printf("%c",185);
      }else{
         printf("%c",186);
```

```
    }
}
gotoxy(78, 25);
printf("%c",188);
for(i = 77; i > 0; i--){
    gotoxy(i,25);
    if(i == 35){
        printf("%c",202);
    }else{
        printf("%c",205);
    }
}
gotoxy(0,25);
printf("%c",200);
for(i = 24; i > 0; i--){
    gotoxy(0,i);
    if(i == 6){
        printf("%c",204);
    }else{
        printf("%c",186);
    }
}

for(i = 1; i < 78; i++){
    gotoxy(i,6);
    if(i == 35){
        printf("%c",203);
    }else{
        printf("%c",205);
    }
}
```

```
    for(i = 7; i < 25; i++){
        gotoxy(35,i);
        printf("%c",186);
    }

}

void clearWindow(){
    int i,j;
    for(i = 37; i < 78; i++){
        for(j = 7; j < 25; j++){
            gotoxy(i,j);printf(" ");
        }
    }
    return;
}

void window(){
    drawRectangle();
    gotoxy(28,2);
    SetColor(35);
    printf("STUDENT RECORD SYSTEM");
    gotoxy(20,3);
    printf("Bansal Institute of Engineering & Technology,Lucknow");
    gotoxy(31,4);
    printf("2019");
    gotoxy(25,24);
    SetColor(17);

}

void get_password(char* pass)
```

```
{
   char temp_passP[25];
   int i=0;
    while(1)
   {
         temp_passP[i]=getch();
         if(temp_passP[i]==13){break;}
         else if(temp_passP[i]==8)
         {
            if(i!=0) {
            printf("\b \b");
            i--;
            } else {printf("\a");}
         }
         else
         {
            printf("*");
            *(pass+i) = temp_passP[i];
            i++;
         }
          *(pass+i)='\0';
    }
}

void use_pass_field(){
   int x = 15, y = 16;
   int use;
   char pass[10];
   SetColor(10);
   gotoxy(15,12);printf("The database is password protected.");
   gotoxy(15,13);printf(" Enter Valid username and password");
   SetColor(17);
```

```
    gotoxy(20,x);printf("USERNAME:- ");
    gotoxy(20,y);printf("PASSWORD:- ");
    gotoxy(34,x);scanf("%d",use);
    gotoxy(34,y);get_password(pass);
}

void print_heading(const char st[]){
    SetColorAndBackground(31,28);
    gotoxy(45,8);printf("SRS : %s",st);
    SetColorAndBackground(17,15);
}

int conf_record(char id[]){
  // left for you
  //it checks whether the entered id for
  //new record is already in the database.
}

void add_student(){
    clearWindow();
    print_heading("Add Record");
    int print = 37;
    FILE *fp;
    fp = fopen("record.txt","ab+");
    SetColor(45);
    if(fp == NULL){
        MessageBox(0,"Error in Opening file\nMake sure your file is not write
protected","Warning",0);

    }else{
        fflush(stdin);
        gotoxy(print,10);printf("ID: ");gets(stu.ID);
```

```
        //here you can confirms the ID
        gotoxy(print,12);printf("Name: ");gets(stu.name);
        gotoxy(print,14);printf("Address: ");gets(stu.add);
        gotoxy(print,16);printf("Parent's name: ");gets(stu.parname);
        gotoxy(print,18);printf("Class: ");scanf("%d",&stu.Class);
        gotoxy(print,20);printf("Phone Number: ");scanf("%ld",&stu.phone_no);
        fwrite(&stu, sizeof(stu), 1, fp);
        gotoxy(40,22); printf("The record is sucessfully added");
    }
    SetColor(28);
    fclose(fp);
    return;
}

void search_student(){
    clearWindow();
    print_heading("Search Record");
    SetColor(45);
    char s_id[15];
    int isFound = 0;
    gotoxy(37,10);printf("Enter ID to Search: ");fflush(stdin);
    gets(s_id);
    FILE *fp;
    fp = fopen("record.txt","rb");
    while(fread(&stu,sizeof(stu),1,fp) == 1){
        if(strcmp(s_id,stu.ID) == 0){
            isFound = 1;
            break;
        }
    }
    if(isFound == 1){
        gotoxy(37,12);printf("The record is Found");
```

```
        gotoxy(37,14);printf("ID: %s",stu.ID);
        gotoxy(37,15);printf("Name: %s",stu.name);
        gotoxy(37,16);printf("Address: %s",stu.add);
        gotoxy(37,17);printf("Parent's Name: %s",stu.parname);
        gotoxy(37,18);printf("Class: %d",stu.Class);
        gotoxy(37,19);printf("Phone No: %ld",stu.phone_no);
    }else{
        gotoxy(37,12);printf("Sory, No record found in the database");
    }
    SetColor(28);
    fclose(fp);
    return;
}

void mod_student(){
    clearWindow();
    print_heading("Modify Record");
    SetColor(45);
    char s_id[15];
    int isFound = 0, print = 37;
    gotoxy(37,10);printf("Enter ID to Modify: ");fflush(stdin);
    gets(s_id);
    FILE *fp;
    fp = fopen("record.txt","rb+");
    while(fread(&stu, sizeof(stu),1,fp) == 1){
        if(strcmp(s_id, stu.ID) == 0){
            fflush(stdin);
            gotoxy(print,12);printf("ID: ");gets(stu.ID);
            gotoxy(print,13);printf("Name: ");gets(stu.name);
            gotoxy(print,14);printf("Address: ");gets(stu.add);
            gotoxy(print,15);printf("Parent's name: ");gets(stu.parname);
            gotoxy(print,16);printf("Class: ");scanf("%d",&stu.Class);
```

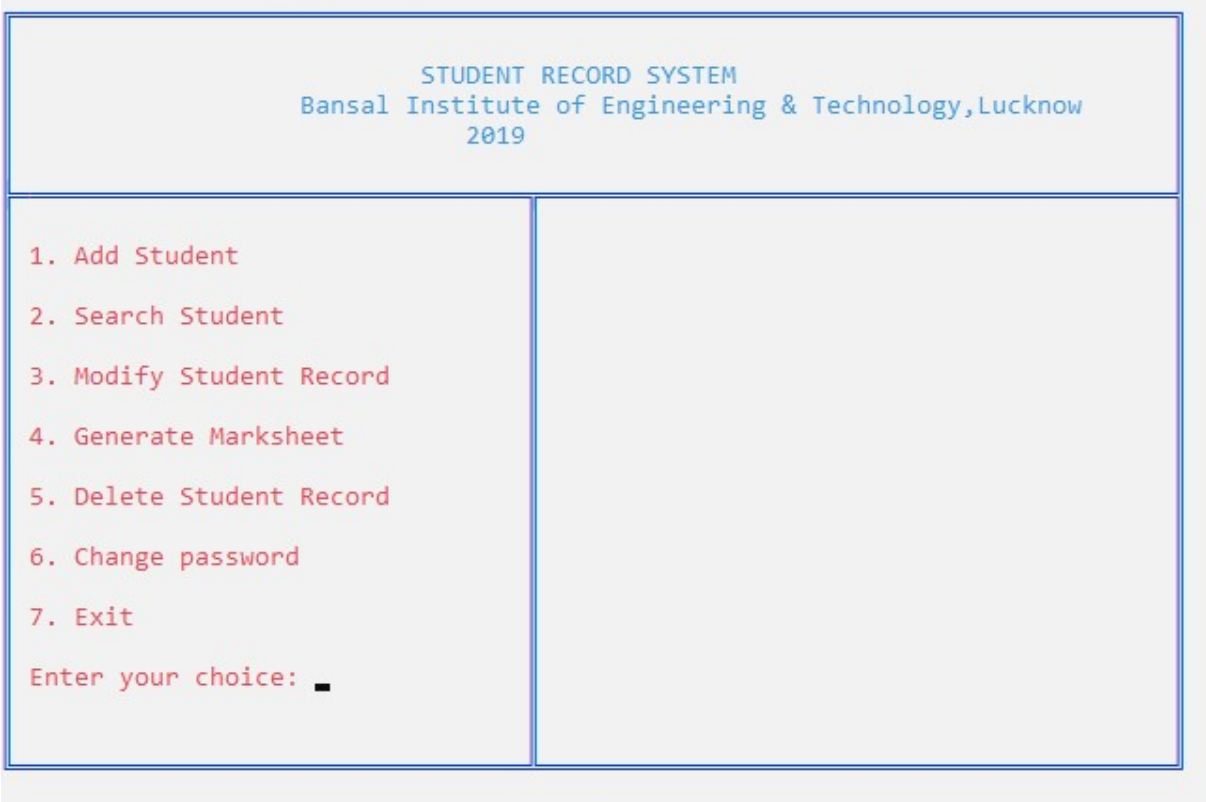
```
            gotoxy(print,17);printf("Phone Number: ");scanf("%ld",&stu.phone_no);
            fseek(fp,-sizeof(stu), SEEK_CUR);
            fwrite(&stu,sizeof(stu), 1, fp);
            isFound = 1;
            break;
        }
    }
    if(!isFound){
        gotoxy(print, 12);printf("No Record Found");
    }
    fclose(fp);
    SetColor(28);
    return;
}

void gen_marksheet(){
    //left for further enhancement
}

void delete_student(){
    clearWindow();
    print_heading("Delete Record");
    SetColor(45);
    char s_id[15];
    int isFound = 0, print = 37;
    gotoxy(37,10);printf("Enter ID to Modify: ");fflush(stdin);
    gets(s_id);
    FILE *fp, *temp;
    fp = fopen("record.txt","rb");
    temp = fopen("temp.txt", "wb");
    while(fread(&stu, sizeof(stu),1,fp) == 1){
        if(strcmp(s_id, stu.ID) == 0){
```

```
            fwrite(&stu,sizeof(stu),1,temp);
        }
    }
    fclose(fp);
    fclose(temp);
    remove("record.txt");
    rename("temp.txt","record.txt");
    gotoxy(37,12);printf("The record is sucessfully deleted");
    SetColor(28);
    return;
}

void main_window(){
    int choice;
    SetColor(28);
    int x = 2;
    while(1){
        gotoxy(x,8);printf("1. Add Student");
        gotoxy(x,10);printf("2. Search Student");
        gotoxy(x,12);printf("3. Modify Student Record");
        gotoxy(x,14);printf("4. Generate Marksheet");
        gotoxy(x,16);printf("5. Delete Student Record");
        gotoxy(x,18);printf("6. Change password");
        gotoxy(x,20);printf("7. Exit");
        gotoxy(x,22);printf("Enter your choice: ");
        scanf("%d",&choice);
        switch(choice){
            case 1:
                add_student();
                break;
            case 2:
                search_student();
```

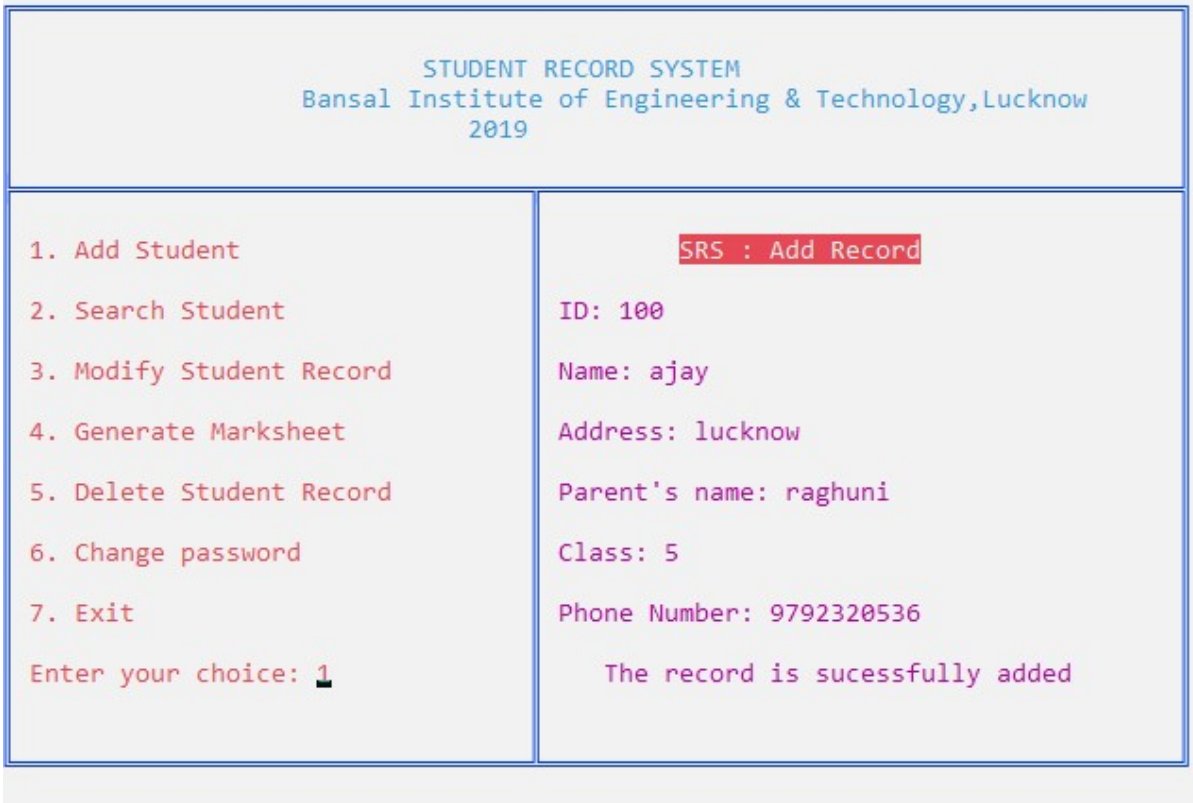
```
                break;
            case 3:
                mod_student();
                break;
            case 4:
                break;
            case 5:
                delete_student();
                break;
            case 6:
                break;
            case 7:
                exit(0);
                break;
            default:
                break;
          }

       }

    }

    int main(){
       ClearConsoleToColors(17,15);
       SetConsoleTitle("Programming-technique.blogspot.com - Student Record
System");
       window();
       //use_pass_field();
       main_window();
       return 0;
    }
```

3.1 RESULTS / OUTPUTS

Student Database Management System



Student Database Management System



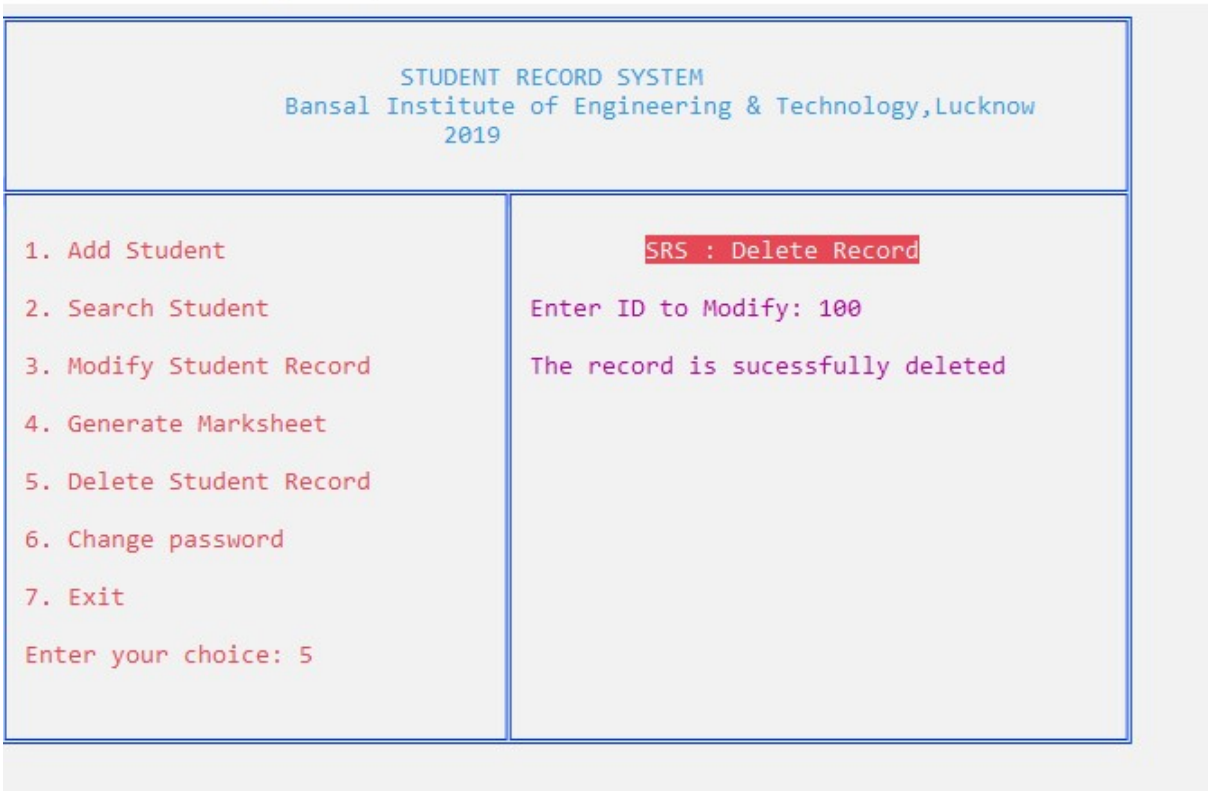
Student Database Management System

```
                    STUDENT RECORD SYSTEM
              Bansal Institute of Engineering & Technology,Lucknow
                              2019

1. Add Student                      SRS : Search Record

2. Search Student                   Enter ID to Search: 100

3. Modify Student Record            The record is Found

4. Generate Marksheet               ID: 100
                                    Name: ajay
5. Delete Student Record            Address: lucknow
                                    Parent's Name: raghuni
6. Change password                  Class: 5
                                    Phone No: 1202385944
7. Exit

Enter your choice: 2
```

Student Database Management System

```
                    STUDENT RECORD SYSTEM
              Bansal Institute of Engineering & Technology,Lucknow
                              2019

1. Add Student                      SRS : Modify Record

2. Search Student                   Enter ID to Modify: 100

3. Modify Student Record            ID: 100
                                    Name: ajay
4. Generate Marksheet               Address: lucknow
                                    Parent's name: raghuni
5. Delete Student Record            Class: 6
                                    Phone Number: 8674637563
6. Change password

7. Exit

Enter your choice: 3
```

Student Database Management System



4. CONCLUSIONS / RECOMMENDATIONS

We have tried to develop a system that can be a great help for the college to manage student records. Despite all our efforts there are some bugs in the system, which are still to be removed. This is possible by testing being done in the system. We have left all the option open so that if there is any other future requirement in the system by the user for the enhancement of the system then it is possible to implement them.

In the last we would like to thanks involved in the development of the system directly or indirectly. We are also thankful to Mr. Sandeep Mishra for so much taken by them in helping to develop the system.

In the end We wish to say that computers should be put to such use where not only their capabilities are exploited but, What is more important, serve the society by raising the standard of living of people, thereby making the world better place to live and work in.

At the end it is concluded that we have made effort on following points:

1. We define the problem on which we are working in the project.
2. Made statement of the aims and objectives of the project.
3. The description of the purpose, scope and applicability.
4. A description of background and context of the project and its relation to work already done in the area.
5. We describe the requirement specifications of the system and the actions that can be done on these things.
6. We understand the problem domain and produce a model of the system, which describes operations that can be performed on the system.
7. We included features and operations in detail, including screen layouts.

5. REFERENCES

Book referred :

1. Let Us C 16th edition, Author: Yashavant kanetkar.
2. 1.Programming in ANSI C, Author: E Balagurusamy.

Website referred:

1. https://www.geeksforgeeks.org/
2. https://www.hackerrank.com/
3. https://www.codechef.com/
4. https://www.google.com/
5. https://www.youtube.com/