STUDENT MANAGEMENT SYSTEM

PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THEAWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY

2nd Year(Branch) (CSE)

SUBMITTED BY:-

(GANESH TIWARI- 2203490100032)

SUBMITTED TO:-

Miss. SHRUTI VERMA

Dr. APJ Abdul Kalam Technical University, Uttar Pradesh

LUCKNOW, INDIA

ABSTRACT

A student management system is a software application that helps educational institutions to manage student data and handle various student-related tasks. It can be used to store and organize information such as student personal and contact details, enrollment information, and student records.

In this post I will explain with examples how can you implement a simple student management system project using the C programming language. C is a general-purpose, procedural programming language that is widely used for developing systems software and applications. It is a popular choice for student management systems because it is efficient, simple to learn, and has a rich set of libraries and functions that can be used to implement various features of the system.

Student management systems are designed to streamline and automate various processes related to managing students, such as student registration, course management, attendance tracking, and grades and transcripts. They can be used by schools, colleges, and universities to manage student data and improve the efficiency of various student-related tasks.

INDEX

TOPIC:- **PAGE NO:-**

1.**Introduction**

- The given code is a C program for managing student records in a school. It includes functions to add, search, delete, update, and display student records. The program uses a file "schoolRecord.txt" to store the student information. It also uses a structure "stuInfo" to define the format of the student record.
- The program starts with the main function, which presents a menu to the user and allows them to choose from various options such as adding a student record, searching for a student record, deleting a student record, updating a student record, showing all student records, and exiting the program.
- The "addStudentRecord" function is used to add a new student record to the file. It prompts the user to enter the student's name, class, age, gender, and admission date. It then appends the new record to the file.
- The "searchStudentRecord" function allows the user to search for a student record based on the student's ID number. It reads the file and compares the entered ID number with the ID numbers in the file to find a match.
- The "deleteStudentRecord" function enables the user to delete a student record based on the student's ID number. It reads the file, creates a temporary file, and copies all records except the one to be deleted to the temporary file. Then it replaces the original file with the temporary file.
- The "updateStudentRecord" function provides the user with options to update a student's name, class, or age based on the student's ID number. It reads the file, makes the necessary changes, and updates the file with the modified record.
- The "showAllStudentRecord" function displays all student records in the file. It also provides an option to display records for a specific class.
- The program runs in a loop until the user chooses to exit.

2. Objective

- All the information of the students can be managed as the main objective of a student management system. For example, information about the students' registration, their detail, courses as well as the personal profile of the students can be obtained by creating such a system.

- Creating a system like this can reduce unnecessary data collection using paper. Therefore, such a system is very important in managing time. Similarly, the use of such a system makes it easy to access student information at any time.

- Also, the number of students who come in a year, as well as the admission of those students as well as the data can be taken from the system to provide the necessary facilities for school staff.

- Using such a student management system enables students to maintain their information, as well as easy access and secure information over a long period of time without any changes.

- Also, when retrieving a book from a students' data can be easily entered, and in some cases, such a system helps to obtain information about the students who obtained the books or who the students are. Managing a library using such a system also makes it easier to manage time.

- Using such a system, student result management can be done, and the information related to the records can be found easily when required.

- As well as can maintain the data accuracy.

- Also make it easy for students to enter information, modify entered information, or remove certain information. This can be done easily using a student management system.

3-SYSTEM REQUIREMENTS

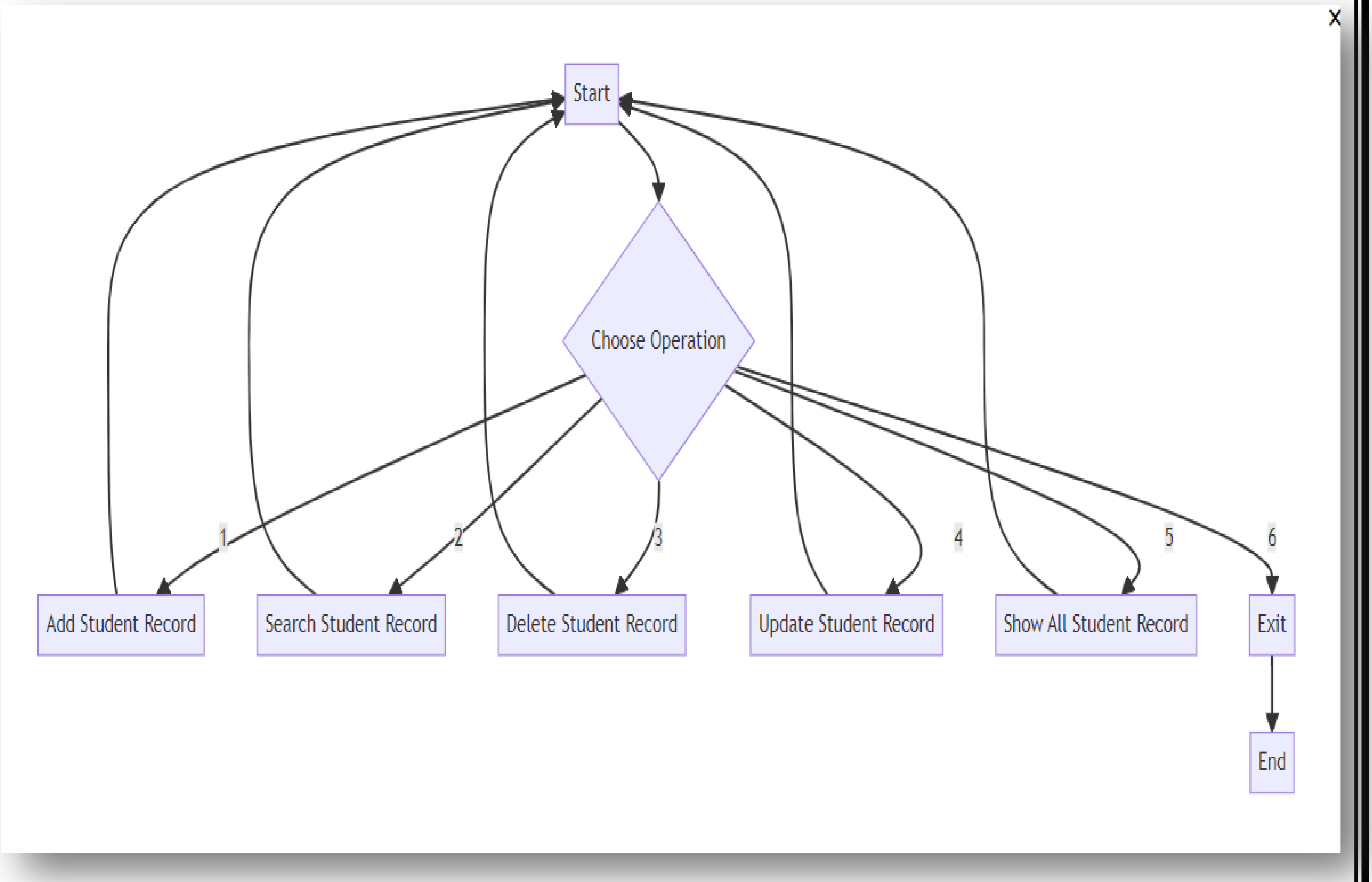
3.1-Hardware Requirements :-

- **Processor:** - Intel Pentium 4 or Above
- **Hard Disk: -** 64GB or more
- **RAM: -** 1GB or more
- **Printer: -** Any.
- **Monitor: -** SVGA Color Monitor (Touch Screen or Simple).
- **Pointing Device: -** Touch Pad or Keys.

3.2-Software Requirements :-

- **Operating System: -** Microsoft Windows XP or above.
- **Software used : -** Dev-C++ [MSC v.192532 or 64 bit (Intel)]on window 32 or 64 bits RAM.

4.FLOWCHART


x
Start
Choose Operation
1
2
3
4
5
6
Add Student Record
Search Student Record
Delete Student Record
Update Student Record
Show All Student Record
Exit
End

5. Project code

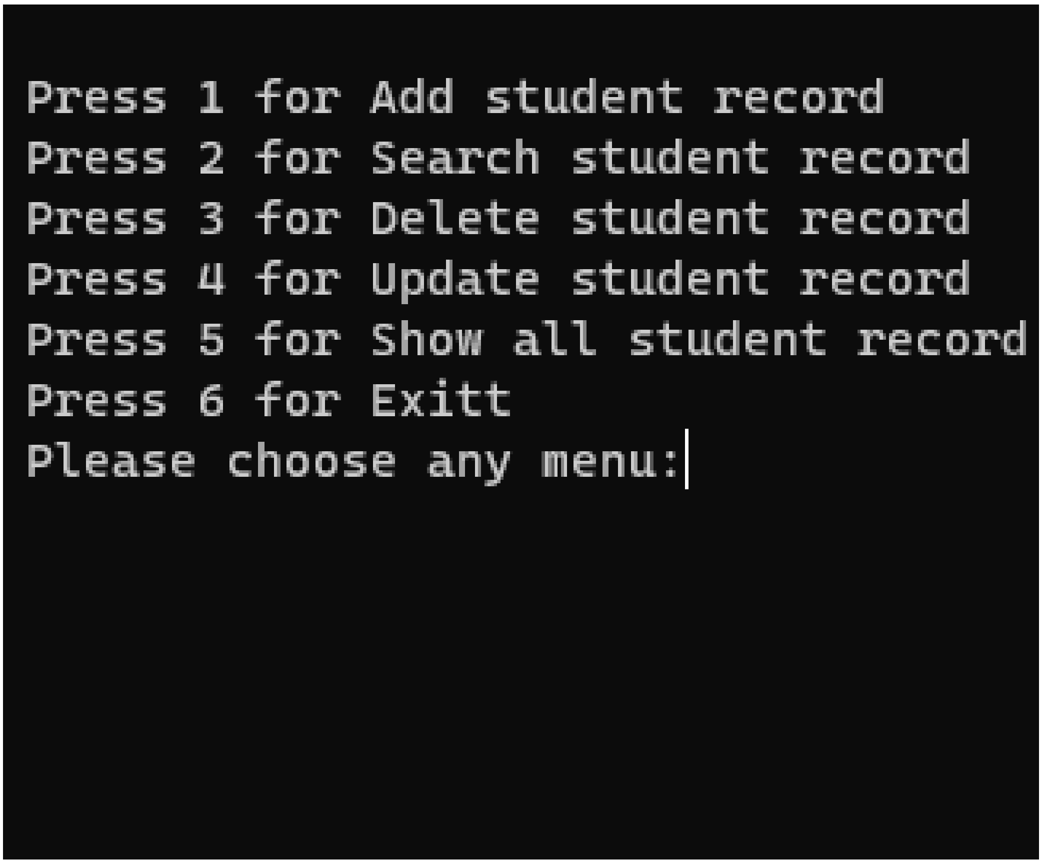
```
#include<stdio.h >
#include<stdlib.h >
#include<string.h>

FILE *ptr = NULL, *temp = NULL;
int countRecord = 0;
struct stuInfo
{
    int idNumber;
    char name[30];
    int class;
    int age;
    char gender[10];
    char admissionDate[12];
}student;

void addStudentRecord()
{
    printf("\nWELCOME TO S.G.M PUBLIC SCHOOL\n\n");
    student.idNumber=100;
    ptr = fopen("schoolRecord.txt", "rb");
    while (fread(&student, sizeof(student), 1, ptr)>0)
    {
        student.idNumber+=1;
        countRecord++;
    }
    fclose(ptr);
    printf("Enter Student Name : ");
    fflush(stdin);
    gets(student.name);
    printf("Enter Student Class:");
    scanf("%d", &student.class);
    printf("Enter Student Age:");
    scanf("%d", &student.age);
    printf("Enter Student Gender:");
    fflush(stdin);
    gets(student.gender);
    printf("Enter Admission Date:");
    fflush(stdin);
    gets(student.admissionDate);
    ptr = fopen("schoolRecord.txt", "ab");
    if (ptr == NULL)
    {
        printf("File is not open\n");
        exit(0);
    }
    fwrite(&student, sizeof(student), 1, ptr);
    countRecord++;
    printf("\nStudent record is successfully added\n");
    fclose(ptr);
}

void searchStudentRecord()
{
    int studentID, check = 0;
    if (countRecord==0)
    {
        printf("\nSchool Record is empty");
        goto goToEnd;
    }
    printf("\nWELCOME TO S.G.M PUBLIC SCHOOL\n\n");
    printf("Enter student ID No. : ");
    scanf("%d", &studentID);
    ptr = fopen("schoolRecord.txt", "rb");
    if (ptr == NULL)
    {
        printf("File is not open.\n");
        exit(0);
    }
    while (fread(&student, sizeof(student), 1, ptr) > 0 && check == 0)
    {
        if (studentID == student.idNumber)
        {
            printf("\nStudent ID No: %d\n", student.idNumber);
            printf("Student Name: %s\n", student.name);
            printf("Student Class: %d\n", student.class);
            printf("Student Age: %d\n", student.age);
            printf("Student Gender: %s\n", student.gender);
            printf("Student Admission Date: %s\n\n", student.admissionDate);
            check = 1;
        }
    }
    if (check == 0)
    {
        printf("\nSorry! Record is not found\n");
    }
    fclose(ptr);
    goToEnd:
    printf("");
}

void deleteStudentRecord()
{
    int stuID, check = 0;
    if (countRecord==0)
    {
        printf("\nSchool Record is empty");
        goto goToEnd;
    }
    printf("Enter student ID number :");
    scanf("%d", &stuID);
    ptr = fopen("schoolRecord.txt", "rb");
    temp = fopen("tempfile.txt", "wb");
    if (ptr == NULL || temp == NULL)
    {
        printf("File is not open.\n");
        exit(0);
    }
    while (fread(&student, sizeof(student), 1, ptr) > 0)
    {
        if (student.idNumber != stuID)
        {
            fwrite(&student, sizeof(student), 1, temp);
            check = 1;
        }
        else
        {
            if (countRecord == 1)
            {
                check = 2;
            }
        }
    }
    fclose(ptr);
    fclose(temp);
    if (check == 1)
    {
        remove("SchoolRecord.txt");
        rename("tempfile.txt", "schoolRecord.txt");
        countRecord--;
        printf("\nThe Student record is successfully deleted\n");
    }
    else if (check == 2)
    {
        remove("schoolRecord.txt");
        rename("tempfile.txt", "schoolRecord.txt");
        countRecord--;
        printf("\nThe Student record is successfully deleted\n");
    }
    else
    {
        printf("\nSorry! Record is not found\n");
    }
goToEnd:
printf("");
}
```

```
void updateStudentRecord()
{
    int stuID, choose, check = 0;
    char name[30];
    int class;
    int age;
    char gender[10];
    if (countRecord == 0)
    {
        printf("\nSchool Record is empty");
        goto goToEnd;
    }
    printf("Enter student ID :");
    scanf("%d", &stuID);
    printf("\nYou can not change your ID number\n");
    printf("What do you want to update\n");
    printf("press 1 for Student Name\n");
    printf("press 2 for Student Class\n");
    printf("press 3 for Student Age\n");
again:
    printf("Choose any menu :");
    scanf("%d", &choose);
    switch (choose)
    {
    case 1:
        printf("Enter new Name :");
        fflush(stdin);
        gets(name);
        ptr = fopen("schoolRecord.txt", "rb");
        temp = fopen("tempRecord.txt", "wb");
        if (ptr == NULL || temp == NULL)
        {
            printf("File is not open\n");
            exit(0);
        }
        while (fread(&student, sizeof(student), 1, ptr) > 0)
        {
            if (student.idNumber == stuID)
            {
                strcpy(student.name, name);
                fwrite(&student, sizeof(student), 1, temp);
                check = 1;
            }
            else
            {
                fwrite(&student, sizeof(student), 1, temp);
            }
        }
        fclose(ptr);
        fclose(temp);
        if (check == 1)
        {
            remove("schoolRecord.txt");
            rename("tempRecord.txt", "schoolRecord.txt");
            printf("\nThe student record is successfully updated\n");
        }
        else
        {
            printf("\nStudent record is not found\n");
        }
        break;
    case 2:
        printf("Enter new Class :");
        scanf("%d", &class);
        ptr = fopen("schoolRecord.txt", "rb");
        temp = fopen("tempRecord.txt", "wb");
        if (ptr == NULL || temp == NULL)
        {
            printf("File is not open\n");
            exit(0);
        }
        while (fread(&student, sizeof(student), 1, ptr) > 0)
        {
            if (student.idNumber == stuID)
            {
                student.class = class;
                fwrite(&student, sizeof(student), 1, temp);
                check = 1;
            }
            else
            {
                fwrite(&student, sizeof(student), 1, temp);
            }
        }
        fclose(ptr);
        fclose(temp);
        if (check == 1)
        {
            remove("schoolRecord.txt");
            rename("tempRecord.txt", "schoolRecord.txt");
            printf("\nThe student record is successfully updated\n");
        }
        else
        {
            printf("\nStudent record is not found\n");
        }
        break;
    case 3:
        printf("Enter new Age :");
```

```
        scanf("%d", &age);
        ptr = fopen("schoolRecord.txt", "rb");
        temp = fopen("tempRecord.txt", "wb");
        if (ptr == NULL || temp == NULL)
        {
            printf("File is not open\n");
            exit(0);
        }
        while (fread(&student, sizeof(student), 1, ptr) > 0)
        {
            if (student.idNumber == stuID)
            {
                student.age = age;
                fwrite(&student, sizeof(student), 1, temp);
                check = 1;
            }
            else
            {
                fwrite(&student, sizeof(student), 1, temp);
            }
        }
        fclose(ptr);
        fclose(temp);
        if (check == 1)
        {
            remove("schoolRecord.txt");
            rename("tempRecord.txt", "schoolRecord.txt");
            printf("\nThe student record is successfully updated\n");
        }
        else
        {
            printf("\nStudent record is not found\n");
        }
        break;
    default:
        printf("Invalid press try again\n");
        goto again;
    }
goToEnd:
printf("");
}

void showAllStudentRecord()
{
    char check;
    int class;
    if (countRecord == 0)
    {
        printf("\nSchool Record is empty");
        goto goToEnd;
    }
    ptr = fopen("schoolRecord.txt", "rb");
    if (ptr == NULL)
    {
        printf("File is not open\n");
        exit(0);
    }
again:
    printf("If you want show all classes student record then press 'Y'\n");
    printf("If you want to show a specific class student record then press 'N'\n");
    fflush(stdin);
    scanf("%c", &check);
    if (check == 'Y' || check == 'y')
    {
        while (fread(&student, sizeof(student), 1, ptr) > 0)
        {
            printf("------------------------------------------------------------------------------------------");
            printf("\n%d\t\t %s\t\t %d\t\t ", student.idNumber, student.name, student.class);
            printf("%d\t\t %s\t\t %s\n", student.age, student.gender, student.admissionDate);
        }
    }
    else if (check == 'N' || check == 'n')
    {
    again1:
        printf("Enter class :");
        scanf("%d", &class);
        if (class >= 1 && class <= 12)
        {
            while (fread(&student, sizeof(student), 1, ptr) > 0)
            {
                if (student.class == class)
                {
                        printf("------------------------------------------------------------------")
                    printf("\n%d\t\t %s\t\t %d\t\t ", student.idNumber, student.name, student.class);
                    printf("%d\t\t %s\t\t %s\n", student.age, student.gender, student.admissionDate);
                }
            }
        }
        else
        {
            printf("Invalid class try again\n");
            goto again1;
        }
    }
    else
    {
        printf("Invalid press try again\n");
        goto again;
    }
    fclose(ptr);
```

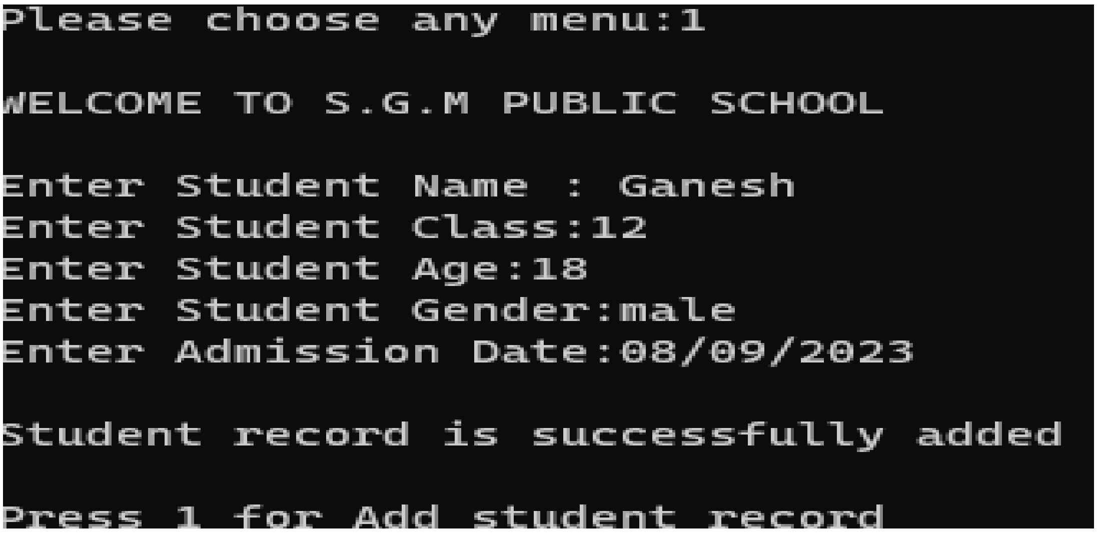
```
goToEnd:
printf("");
}

void main()
{
    int choose;
    while (1)
    {
    again:
        printf("\nPress 1 for Add student record\n");
        printf("Press 2 for Search student record\n");
        printf("Press 3 for Delete student record\n");
        printf("Press 4 for Update student record\n");
        printf("Press 5 for Show all student record\n");
        printf("Press 6 for Exitt\n");
        printf("Please choose any menu:");
        scanf("%d", &choose);
        switch (choose)
        {
        case 1:
            addStudentRecord();
            break;
        case 2:
            searchStudentRecord();
            break;
        case 3:
            deleteStudentRecord();
            break;
        case 4:
            updateStudentRecord();

            break;
        case 5:
            showAllStudentRecord();
            break;
        case 6:
            exit(0);
        default:
            printf("Invalid Press try again\n");
            goto again;
        }
    }
}
```

6.Result/Output



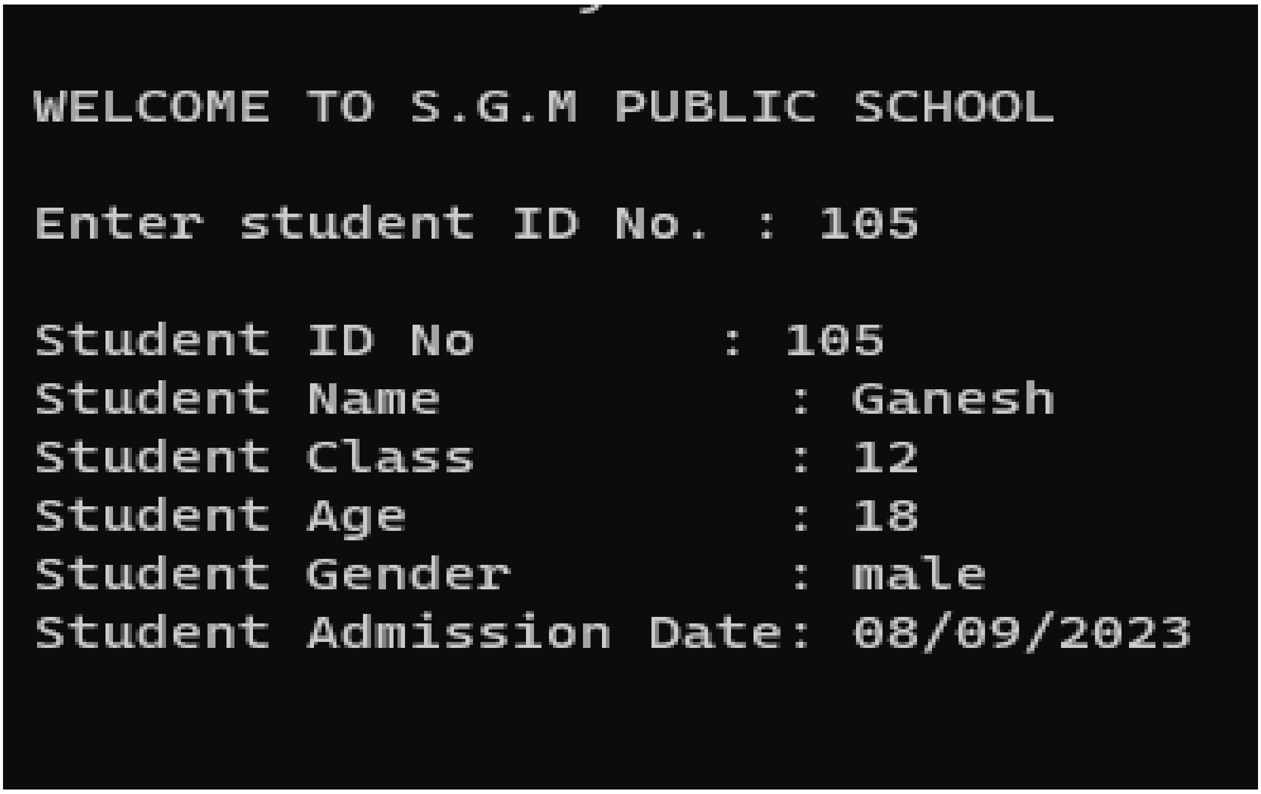
```
Press 1 for Add student record
Press 2 for Search student record
Press 3 for Delete student record
Press 4 for Update student record
Press 5 for Show all student record
Press 6 for Exitt
Please choose any menu:
```

Fig 1:- Menu.



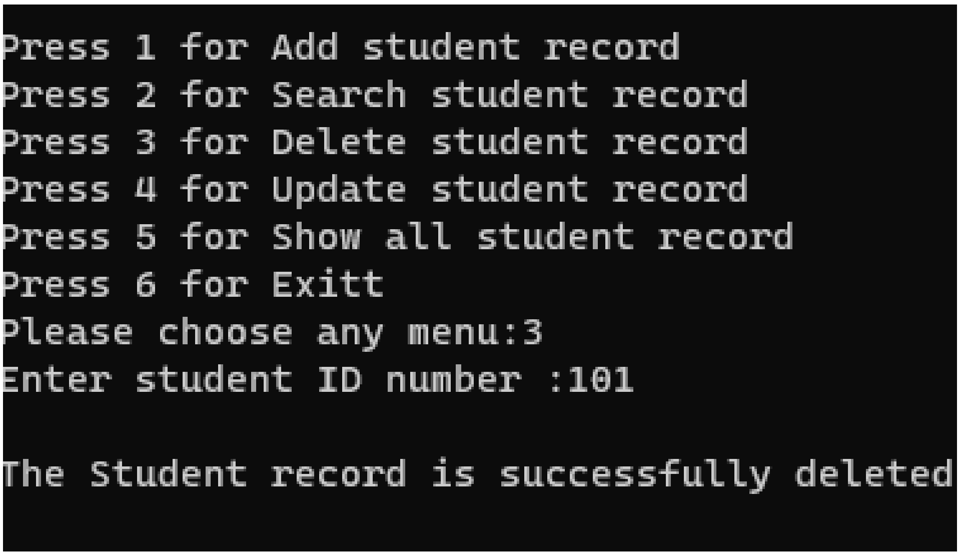
```
Please choose any menu:1

WELCOME TO S.G.M PUBLIC SCHOOL

Enter Student Name : Ganesh
Enter Student Class:12
Enter Student Age:18
Enter Student Gender:male
Enter Admission Date:08/09/2023

Student record is successfully added

Press 1 for Add student record
```

Fig2:- Option 1 -> Add student.



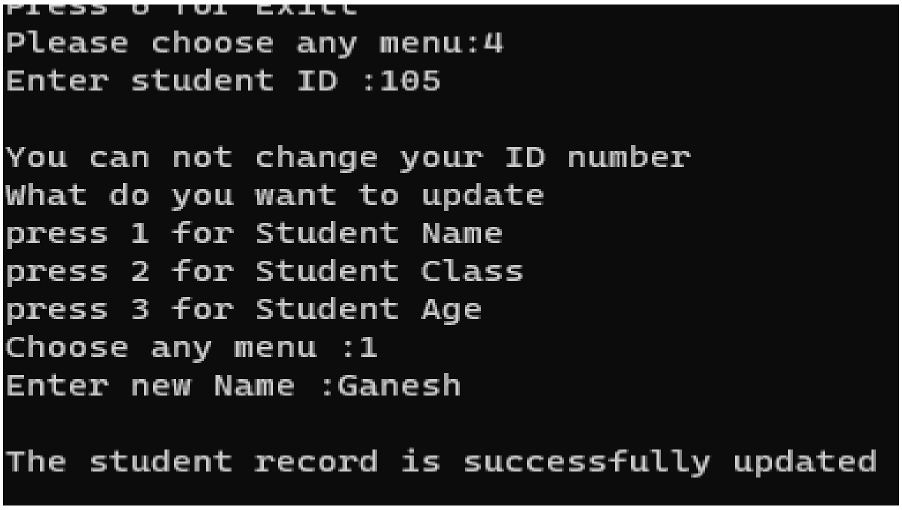
```
WELCOME TO S.G.M PUBLIC SCHOOL

Enter student ID No. : 105

Student ID No       : 105
Student Name          : Ganesh
Student Class         : 12
Student Age           : 18
Student Gender        : male
Student Admission Date: 08/09/2023
```

Fig 3:- Option 2 ->Search .



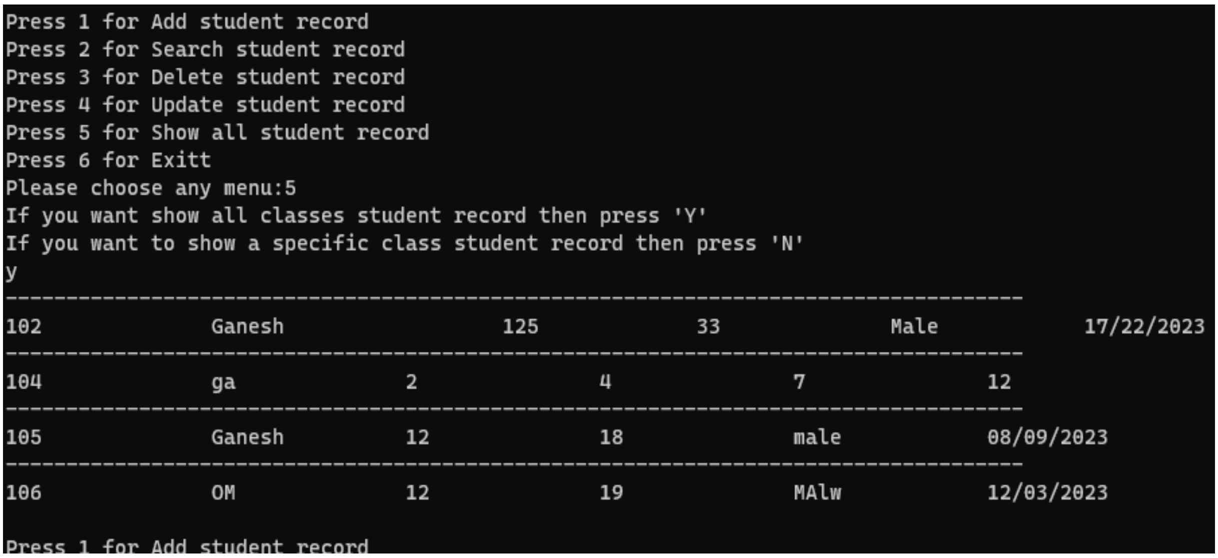
```
Press 1 for Add student record
Press 2 for Search student record
Press 3 for Delete student record
Press 4 for Update student record
Press 5 for Show all student record
Press 6 for Exitt
Please choose any menu:3
Enter student ID number :101

The Student record is successfully deleted
```

Fig 4:- Option 3 ->Delete.



```
Press 6 for Exitt
Please choose any menu:4
Enter student ID :105

You can not change your ID number
What do you want to update
press 1 for Student Name
press 2 for Student Class
press 3 for Student Age
Choose any menu :1
Enter new Name :Ganesh

The student record is successfully updated
```

Fig 5:- Option 4->Update.



```
Press 1 for Add student record
Press 2 for Search student record
Press 3 for Delete student record
Press 4 for Update student record
Press 5 for Show all student record
Press 6 for Exitt
Please choose any menu:5
If you want show all classes student record then press 'Y'
If you want to show a specific class student record then press 'N'
y
------------------------------------------------------------------------------------
102              Ganesh                  125             33              Male            17/22/2023
------------------------------------------------------------------------------------
104              ga              2               4               7               12
------------------------------------------------------------------------------------
105              Ganesh          12              18              male            08/09/2023
------------------------------------------------------------------------------------
106              OM              12              19              MAlw            12/03/2023

Press 1 for Add student record
```

Fig 6:-Option 5->Show all.

7. Technology Used

Origin of C Language

C is a general-purpose programming language created by Dennis Ritchie at the Bell Laboratories in 1972. It is a very popular language, despite being old. The main reason for its popularity is because it is a fundamental language in the field of computer science.

It is said that 'C' is a god's programming language. One can say, C is a base for the programming. If you know 'C,' you can easily grasp the knowledge of the other programming languages that uses the concept of 'C'

It is essential to have a background in computer memory mechanisms because it is an important aspect when dealing with the C programming language.

Why learn C?

- It is one of the most popular programming language in the world
- If you know C, you will have no problem learning other popular programming languages such as Java, Python, C++, C#, etc, as the syntax is similar
- C is very fast, compared to other programming languages, like Java and Python
- C is very versatile; it can be used in both applications and technologies

Applications of C Language

- Development of video games
- Applications using graphical user interfaces
- Databases and computer operating systems
- Browsers on the internet
- Computational and graphical methods
- Banking
- Cloud computing and distributed systems
- Compilers
- Embedded systems are systems that are integrated into a larger system.
- Integrated software libraries for enterprises.

8. SCOPE OF THE PROJECT

At present, everyone is looking for a system that is superior to the facilities provided by higher education institutions such as universities, without being limited to the basic facilities The purpose of this student management system is to create a user-friendly and user-friendly management system. This system makes it easy to manage student administration, university administration, as well as other internal affairs such as student exams, fee payments, etc.

At present such systems are accepted by many educational institutions such as universities and the main objective is to facilitate the work efficiently and modernize the use of such a system based on smart technology. When we create this system, we used better techniques for a get high output. For an example we used Database Tigger function for this system. It is very efficient and useful system to keep all the data accurately and can be accessed just in seconds

9.CONCLUSION

This system is described in detail as a student management system, a system that assists educational institutions such as universities in managing student information and all relevant data properly. The main objective is to manage the relevant systems through a well-designed and computerized system for all aspects such as All students have their own smart ID. Here you can get a lot of information related to them such as their student number, name, gender, phone number and the relevant degree. This system is described in detail as a student management system, a system that assists educational institutions such as universities in managing student information and all relevant data properly. The main objective is to manage the relevant systems through a well-designed and computerized system for all aspects such as student. It is quite clear that such a system would be a great help in carrying out well-managed work in institutions such as an educational institution.

10.Future Implementation

A student information system is simply called a Student Management System. Such a system facilitates communication coordination. Such a system makes it easy for administrative staff as well as parents to find information. The system we have designed above gives the user several easily identifiable interfaces as well as easy access to it. All you'll need is a user ID and a password to get started. We hope to make this interface even more attractive. And this system we have designed can be customized to maximize its security. We have not yet applied such development to this system within the time frame given to us. We also hope to include SMS Alert system to further validate the validity of such a system. This allows the user to see the accuracy of the work he or she has done, as well as easily identify any flaws and fix them again in a very short time. Simply put, the idea here is to prove authenticity. (v. Forrester, 2019) We hope to include some more important icons for these interfaces as well. We think it would be more appropriate to include the Back, Min and Esc icons for the peace interface itself. Here we hope for the convenience of the user. We also intend to include validations for such sections in order to obtain accurate data without error for data entry instances. For example, it is best to use an email address. And as we have used, all the relevant data should be provided to all the users themselves.

We know that all the data entered by the user is contained in a database. We also want to make sure that the user sees and analyzes all of this data. We feel it is appropriate to allow a selection to enter data in some cases. For example, suppose a user enters a degree program. It needs to be typed and it will take some time. But with selection, it can be easily done right. This system works on a number of selected application areas. They are the Registration Department, IT Division, Examination Department, Library Management and Administration Department. The system created above provides a good connection for all these areas. This method is more suitable and practical for all existing educational institutions. The Student Management System can be described as the only and best solution available for the entire college management. By improving this system, we hope effectively manage student data. The above enhancements are the reason for providing a higher level of quality service

11. REFERENCES

11.1 BOOK References :-

[1]. v. Forrester, V. (2019) “SCHOOL MANAGEMENT INFORMATION SYSTEMS: CHALLENGES TO EDUCATIONAL DECISIONMAKING IN THE BIG DATA ERA,”

[**2**]**.** Zhang, H. (2019) Research on the Information Management System of University Student Status Archives.

[3]. Z. Kai, “Design and implementation of college students' entrepreneurship management system based on B/S structure,” *RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao*, vol. 2016, no. 17, pp. 102–113, 2016.

[4]. S. R. Bharamagoudar, R. B. Geeta, and S. G. Totad, “Web based student information management system,”

11.2 ONLINE References:-

1. https://en.wikipedia.org/wiki/student-management/
2. https://www.eeweb.com/tools/tudent-management/
3. https://www.geeksforgeeks.org/