Student Management System

By

Nahid Hasan	ID: 22235103143
Sayeb Senam	ID: 22235103116

Bachelor of Science in

Computer Science and Engineering

Department of Computer Science and Engineering

Bangladesh University of Business and Technology

December 2023

Contents

Declaration

We hereby declare that the Project on Typing Game submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering of Bangladesh University of Business and Technology (BUBT) is our own work and that it contains no material which has been accepted for the award to the can- didate(s) of any other degree or diploma, except where due reference is made in the text of the project. To the best of our knowledge, it contains no materials previously published or written by any other person except where due reference is made in the project.

Signature of Authors

1st member name 2nd member name

Approval

This project work entitled ”Student Management System” is submitted by Nahid Hasan(ID:22235103143), Sayeb Senam(ID:22235103116) under the Department of Computer Science and Engineering of Bangladesh University of Business and Technology (BUBT) is accepted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering..

Supervisor:

Ms Humayra Ahmed
Assistant Professor
Department of Computer Science & Engineering
Bangladesh University of Business and Technology
Dhaka, Bangladesh

Chairman

Md Saifur Rahman
Assistant Professor & Chairman(Acting)
Department of Computer Science & Engineering
Bangladesh University of Business and Technology
Dhaka, Bangladesh

Dedication

We would like to dedicate this research to our loving parents . . .

Acknowledgement

We are deeply thankful to Bangladesh University of Business and Technology (BUBT) for providing us such a wonderful environment to peruse our project. We would like to express our sincere gratitude to Humayra Ahmed Assistance Professor-BUBT. We have completed our project with his help. We found the project area, topic, and problem with his suggestions. He guided us with our study, and supplied us many articles and academic resources in this area. He is patient and responsible. When we had questions and needed his help, he would always find time to meet and discuss with us no matter how busy he was. We also want to give thanks to our CSE department. Our department provide us logistic supports to complete our project with smoothly. We would also like to acknowledge our team members for supporting each other and be grateful to our university for providing this opportunity for us.

Chapter 1

1 Introduction

1.1 Introduction

A Student Management System is a software application that centralizes and automates various administrative tasks in educational institutions.Like this, to use our SMS project you can efficiently handle student's CGPA, add or delete student's information,add marks,search a student by id,display all students information and improving overall efficiency. By providing real-time access to information, reducing paperwork, and enhancing communication, the SMS plays a crucial role in optimizing educational management for schools, colleges, and universities.

1.2 Problem Statement

Suppose, you are a teacher and your class student is 50, now you need to add/store your all student information like student name,student id,student marks, and you want to calculate student CGPA or you need to delete a student information,also you want to show all student information. So, it is difficult and time consuming process for you. To solve this problem you can use this type student management system application to make store smartly large number of student information easily. It is very easy to use and it will reduce your time consumption. For instance, BUBT annex. I think you know about this, it's play a very important role for our university.

1.3 Research Objectives

Feature of your SMS project:

- Add Student
- Display Students
- Search Student by ID

- Add Students marks
- Delete Student
- Exit

1.4 Motivations

If you use this software, it will save your precious time and increase work efficiency and reduce the trouble of handwriting.

1.5 Project Organization

Chapter 2 : Project Overview ,Project Analysis
Chapter 3 : Introduction, Feasibility analysis, Requirement Analysis, The Project Methodology, Feature Selection, Design and Implementation, Introduction, Flow of Work, Evaluation, Result Evaluation, Result and Discussion, Summary
Chapter 4 : Introduction, Conclusion, Future Works and Direction

1.6 Summary

Chapter 1 introduces the Student Management System project, highlighting its relevance in addressing the growing need for an effective tool to store and manage student's data . The chapter provides a brief overview of the project's objectives and outlines the purpose of the report. It sets the context for the subsequent chapters. The focus is on the significance of faster data implemantation in the digital era and the project's aim to fill a gap in accessible and user-friendly management system.

Chapter 2

2 Project Review

2.1 Project Overview

This project must aim towards managing student's data so that the valueable time of user should not be wasted. After using this software the user will be able to add student data more efficiently. This software has a fast and efficient way of calculating results of students.

2.2 Project Analysis

The student management system is a software application where you can add,delete,search students information.

Key Features:

- The program starts with a welcome message.
- The main menu provides options for different functionalities, include add student, display students, search student, add marks, delete student and exiting the program.
- Add student here you can add student with id and section
- You can store maximum 50 student information
- You can view all students information such as name, id, section, CGPA everything.
- If you want to delete a you can search by student id and delete this student information.
- Colorful text- we use system color 0B for change text color.

Chapter 3

3 Proposed Model

3.1 Introduction

The development of a Student Management System project involves a multifaceted exploration, covering various critical aspects essential for its successful execution. This chapter embarks on a comprehensive journey through the project's Feasibility Analysis, Requirement Analysis, The Project Methodology, Data Set details, Data Collection and Preprocessing, Feature Selection, Algorithmic intricacies, and the Design and Implementation considerations.

3.2 Requirement Analysis

Student Management System project involves a thorough examination of functional and nonfunctional specifications. Functionally, the system must accurately store data and measure CGPA. User authentication and result storage are essential. Nonfunctional requirements include a user-friendly interface, cross-platform compatibility, and efficient processing to ensure real-time feedback. Accessibility features for differently-abled users should also be considered. Additionally, scalability to accommodate future enhancements and security measures to protect user data are integral aspects of the project's requirement analysis. The analysis forms the foundation for the successful development and implementation of the Student Management System.

3.3 The Project Methodology

The project methodology for the student management system:

1. Project Planning:Define project scope, objectives, and variables. Develop a detailed project plan outlining tasks, timelines, and resource allocation.

2. Design:Create a detailed system design, including the user interface, back-end working logic.

3. Implementation: Code the application using C programming language.

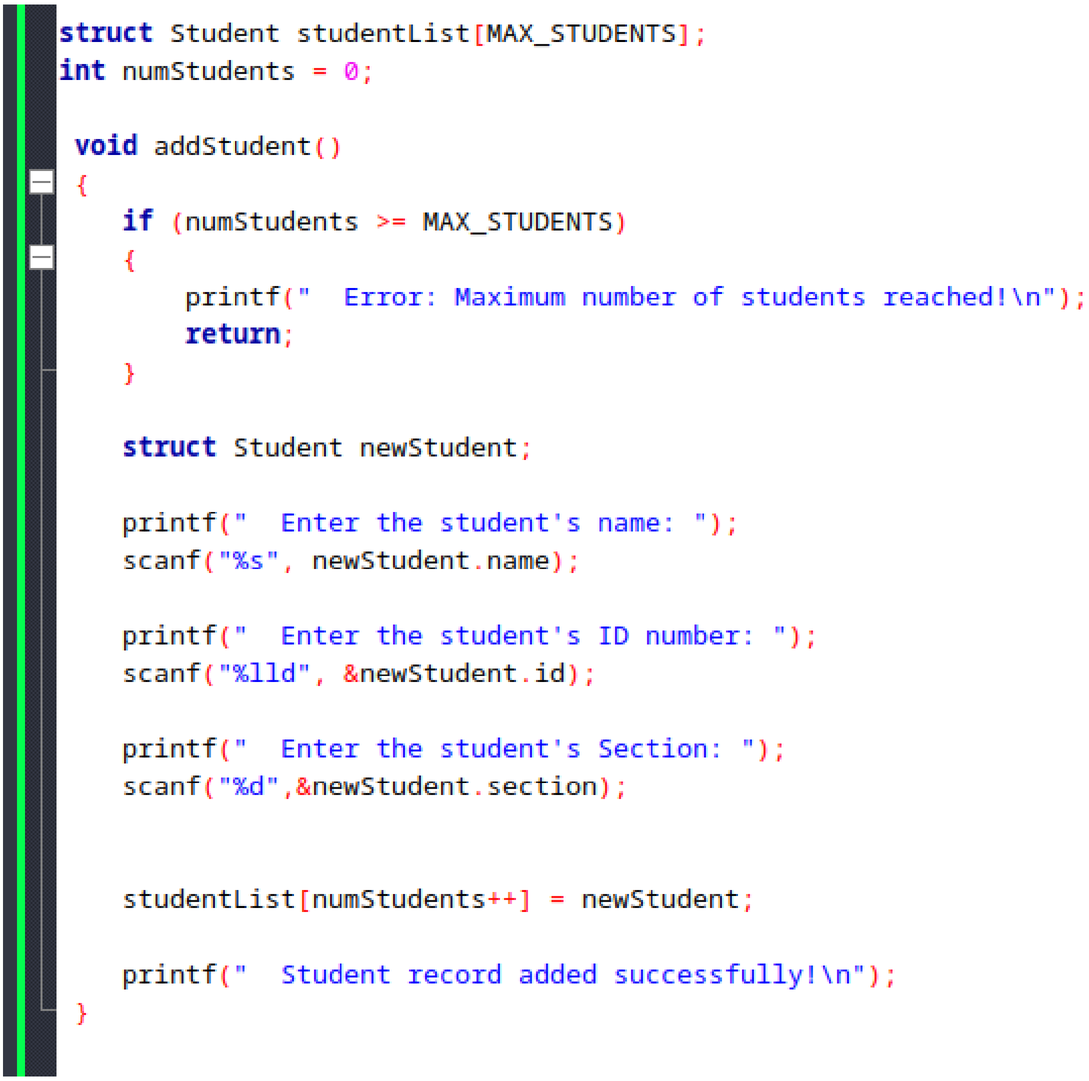
4. Testing: Test the program from user input and check every function that is working perfectly or have any bug then fix it

5. Function: In the program here we use come custom function to add information, calculate cgpa, delete specific student and search s student by id to use every custom function.

3.4 Feature and Flow of Work

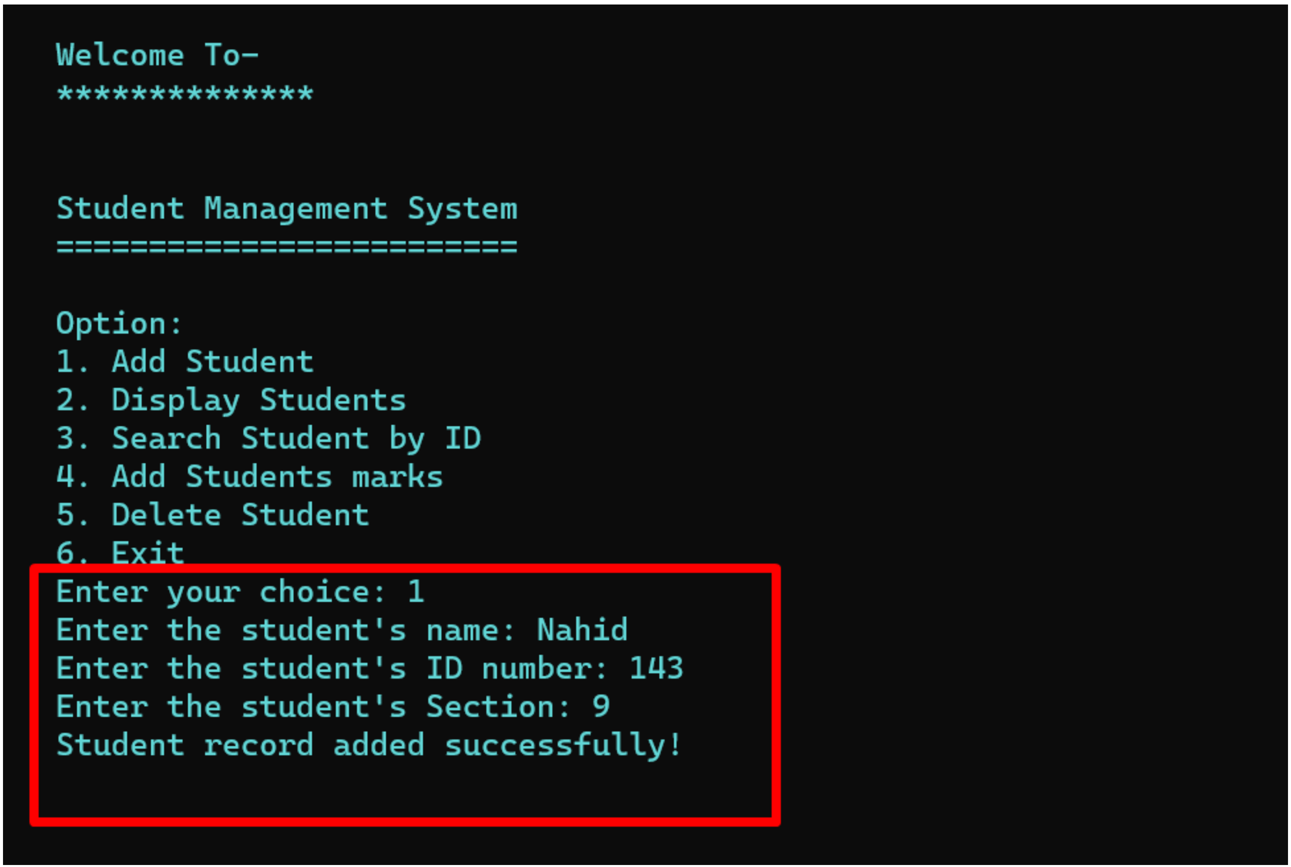
Used Functions:-

addStudent:

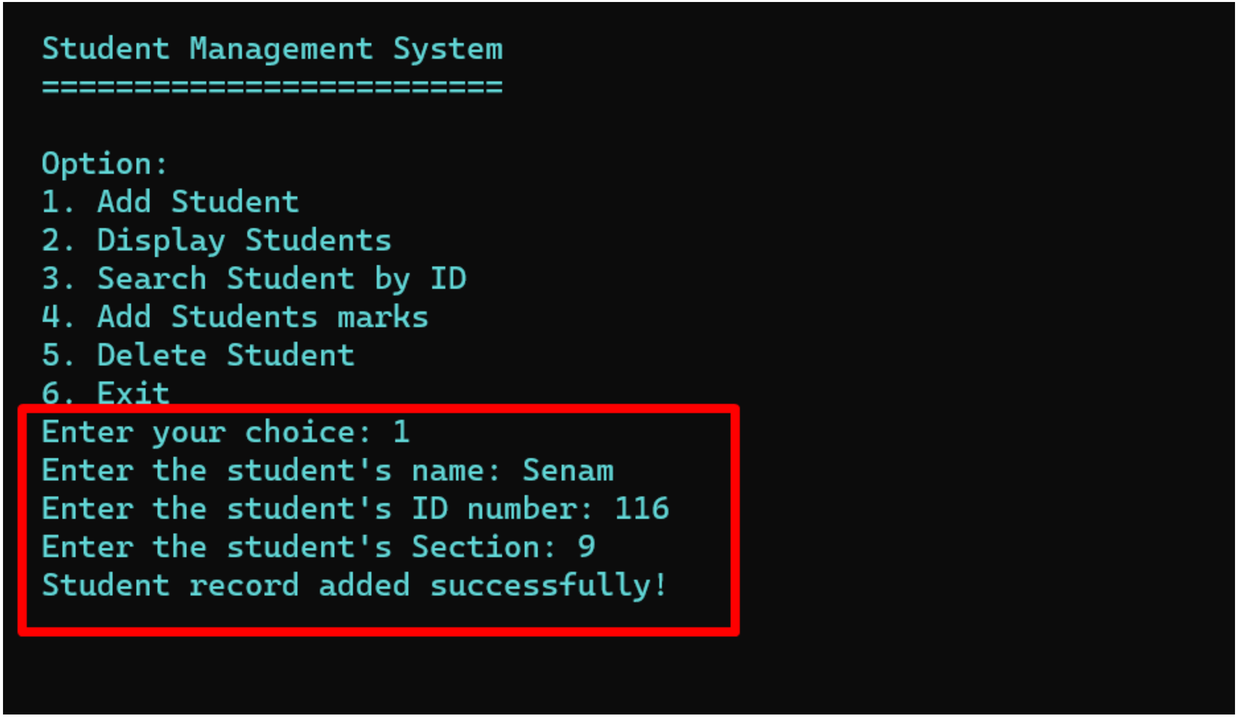


```
struct Student studentList[MAX_STUDENTS];
int numStudents = 0;

 void addStudent()
 {
    if (numStudents >= MAX_STUDENTS)
    {
        printf("  Error: Maximum number of students reached!\n");
        return;
    }

    struct Student newStudent;

    printf("  Enter the student's name: ");
    scanf("%s", newStudent.name);

    printf("  Enter the student's ID number: ");
    scanf("%lld", &newStudent.id);

    printf("  Enter the student's Section: ");
    scanf("%d",&newStudent.section);


    studentList[numStudents++] = newStudent;

    printf("  Student record added successfully!\n");
 }
```

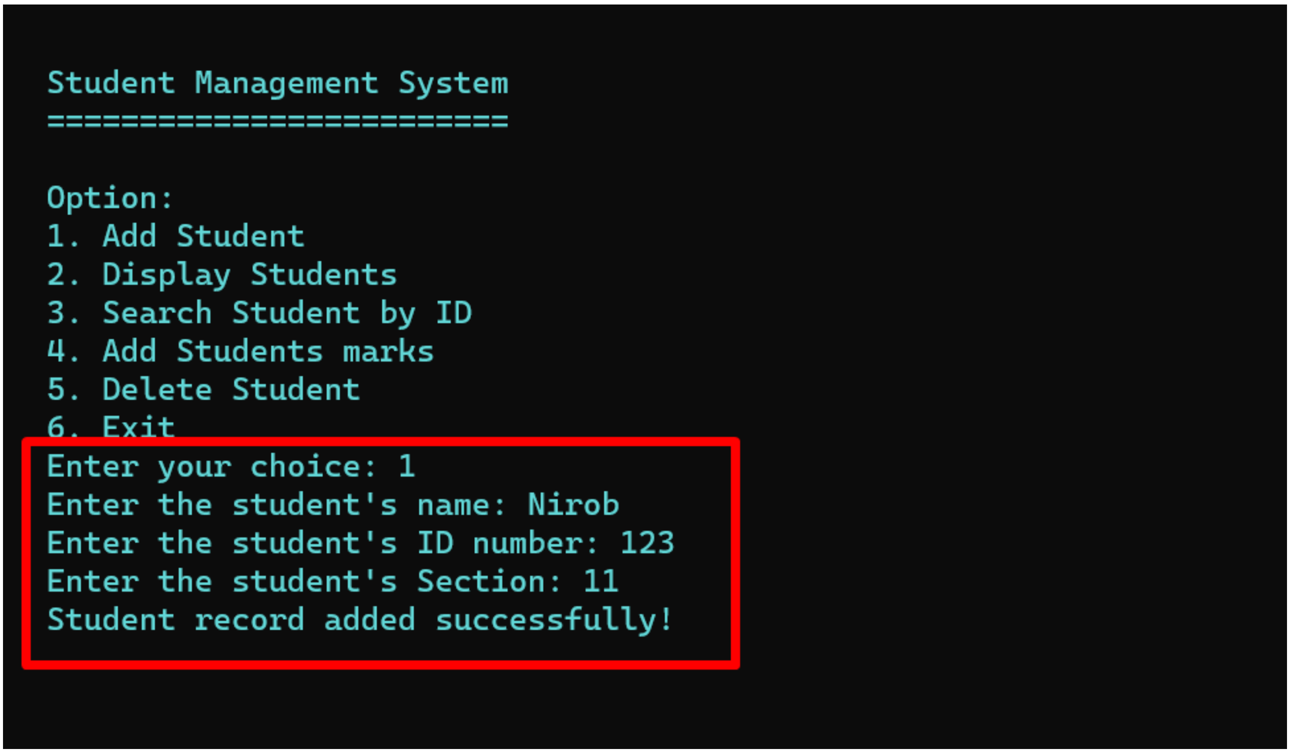
The usage of addStudent function is to add student's Name,ID,Section as user's input.



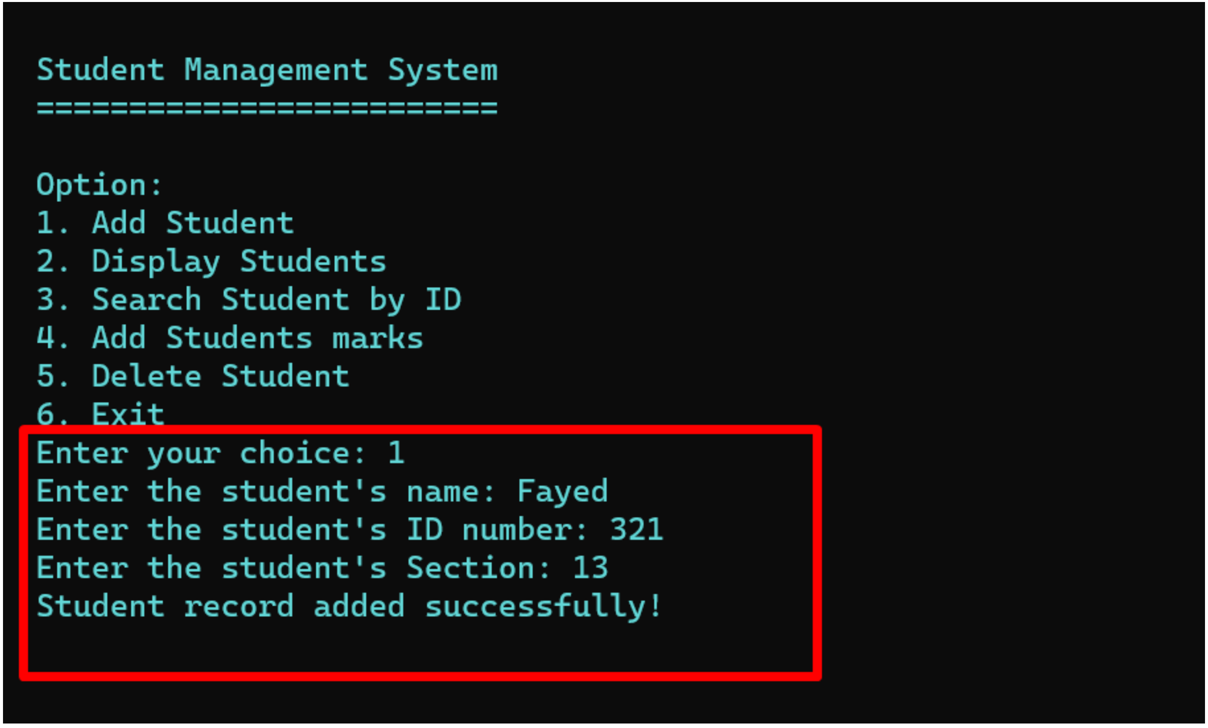
```
Welcome To-
**************


Student Management System
=========================

Option:
1. Add Student
2. Display Students
3. Search Student by ID
4. Add Students marks
5. Delete Student
6. Exit
Enter your choice: 1
Enter the student's name: Nahid
Enter the student's ID number: 143
Enter the student's Section: 9
Student record added successfully!
```



```
Student Management System
=========================

Option:
1. Add Student
2. Display Students
3. Search Student by ID
4. Add Students marks
5. Delete Student
6. Exit
Enter your choice: 1
Enter the student's name: Senam
Enter the student's ID number: 116
Enter the student's Section: 9
Student record added successfully!
```

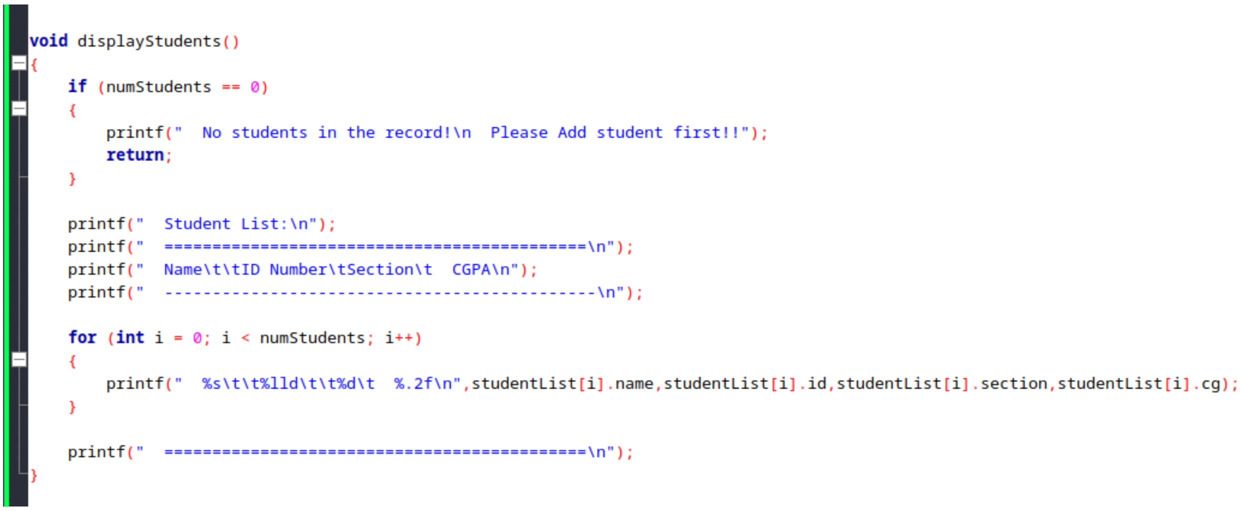


```
Student Management System
=========================

Option:
1. Add Student
2. Display Students
3. Search Student by ID
4. Add Students marks
5. Delete Student
6. Exit
Enter your choice: 1
Enter the student's name: Nirob
Enter the student's ID number: 123
Enter the student's Section: 11
Student record added successfully!
```



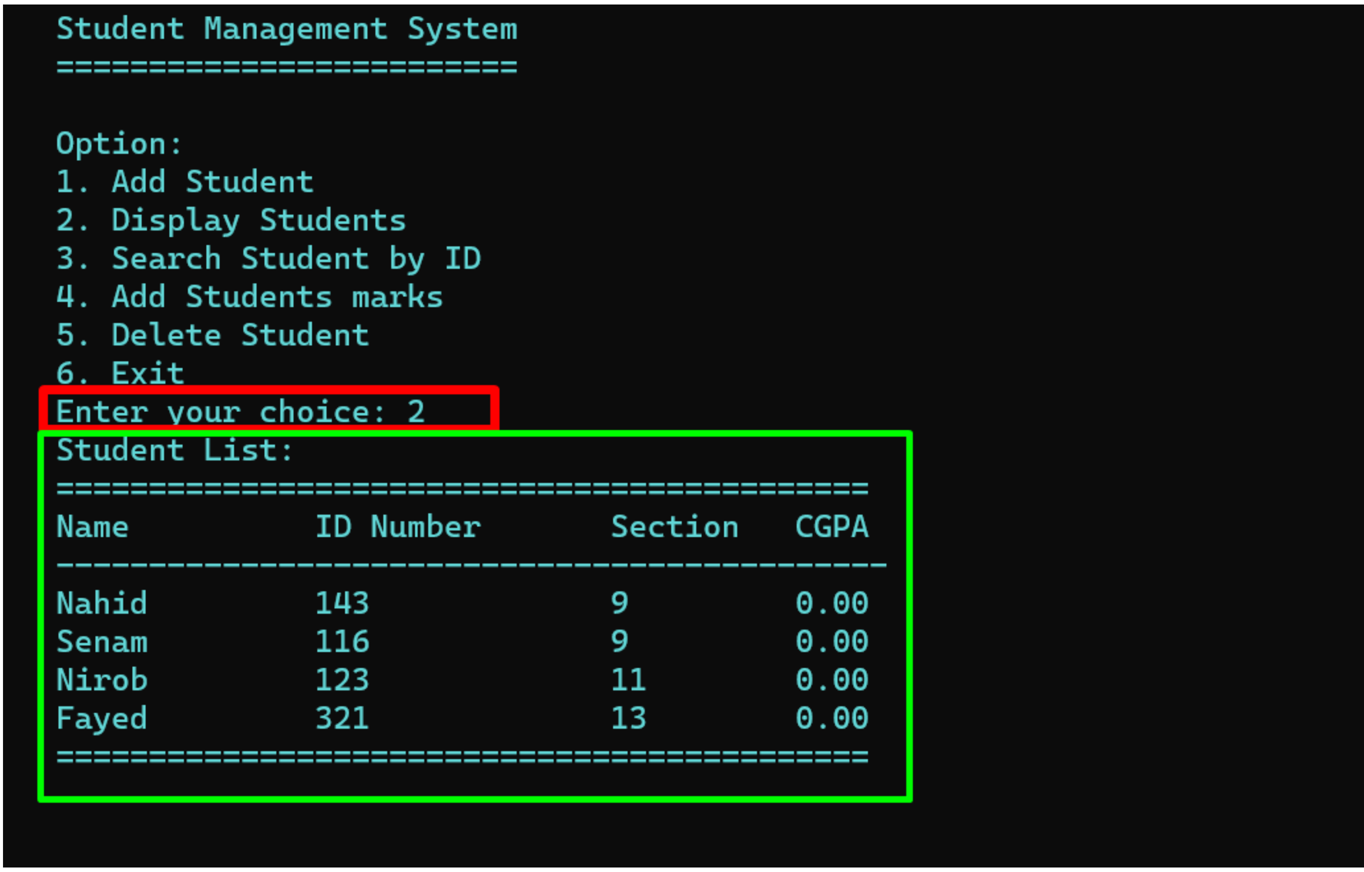
```
Student Management System
=========================

Option:
1. Add Student
2. Display Students
3. Search Student by ID
4. Add Students marks
5. Delete Student
6. Exit
Enter your choice: 1
Enter the student's name: Fayed
Enter the student's ID number: 321
Enter the student's Section: 13
Student record added successfully!
```

displayStudent:



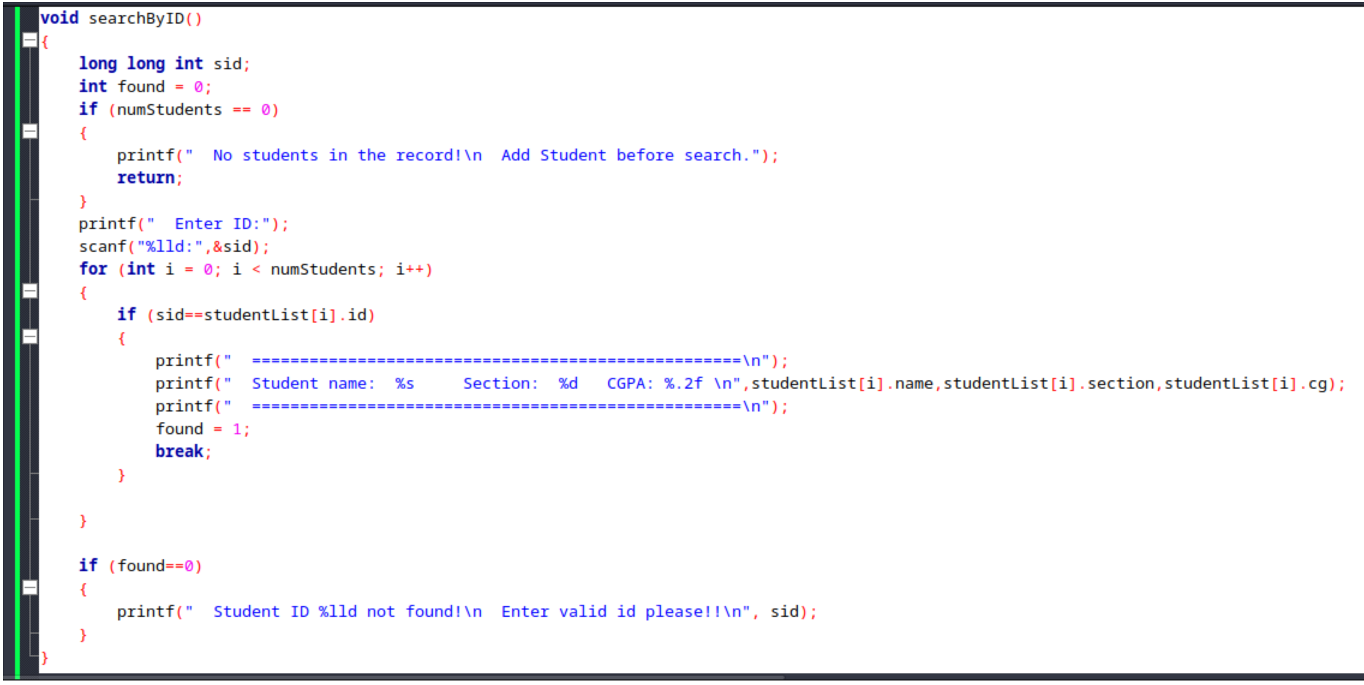
```
void displayStudents()
{
    if (numStudents == 0)
    {
        printf("  No students in the record!\n  Please Add student first!!");
        return;
    }

    printf("  Student List:\n");
    printf("  ============================================\n");
    printf("  Name\t\tID Number\tSection\t  CGPA\n");
    printf("  --------------------------------------------\n");

    for (int i = 0; i < numStudents; i++)
    {
        printf("  %s\t\t%lld\t\t%d\t  %.2f\n",studentList[i].name,studentList[i].id,studentList[i].section,studentList[i].cg);
    }

    printf("  ============================================\n");
}
```

The usage of displayStudent function is to display the gathered data of students.



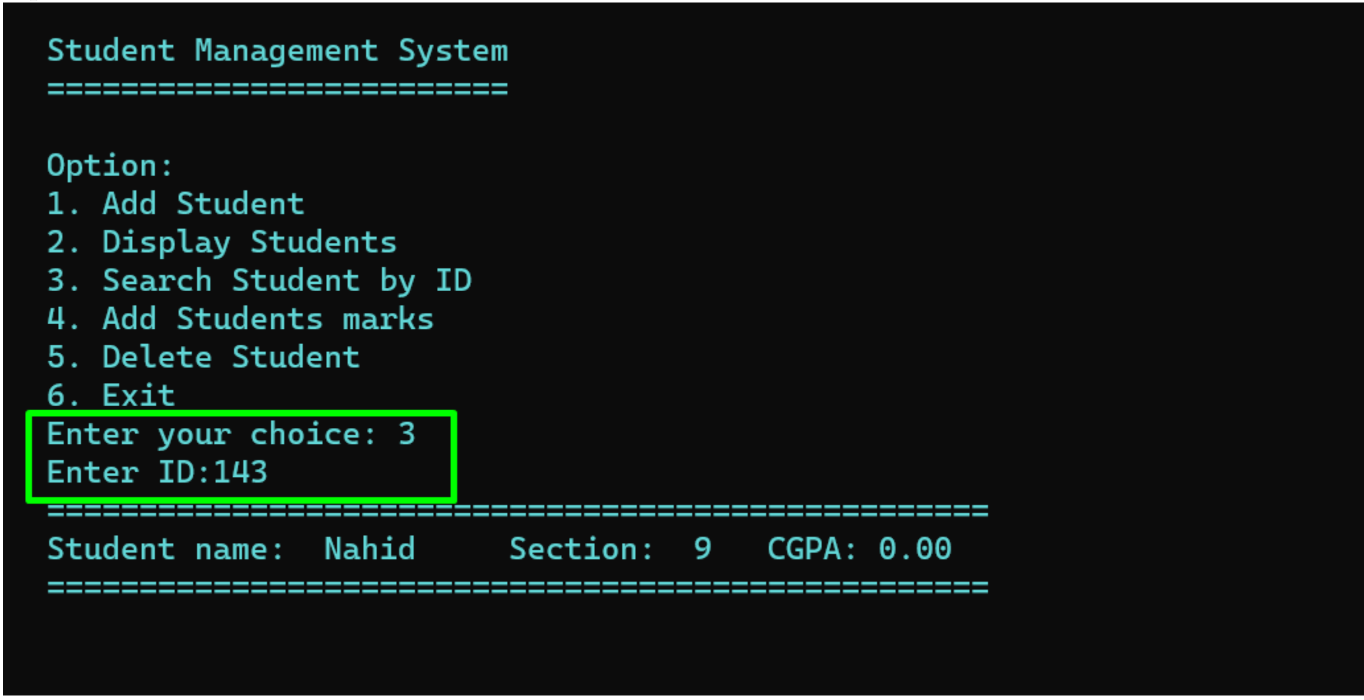
```
  Student Management System
  =========================

  Option:
  1. Add Student
  2. Display Students
  3. Search Student by ID
  4. Add Students marks
  5. Delete Student
  6. Exit
  Enter your choice: 2
  Student List:
  ============================================
  Name          ID Number       Section   CGPA
  ---------------------------------------------
  Nahid         143             9         0.00
  Senam         116             9         0.00
  Nirob         123             11        0.00
  Fayed         321             13        0.00
  ============================================
```

searchByID:



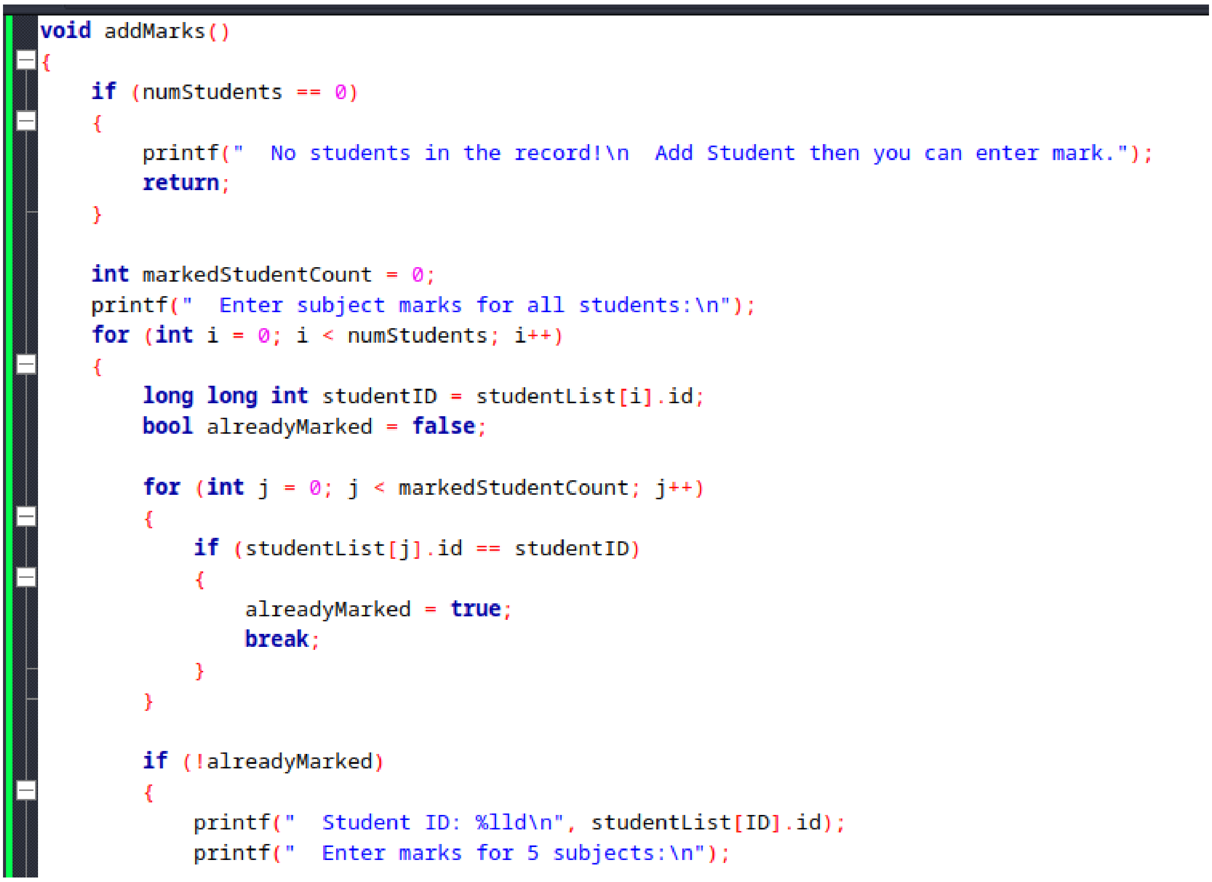
```c
void searchByID()
{
    long long int sid;
    int found = 0;
    if (numStudents == 0)
    {
        printf("  No students in the record!\n  Add Student before search.");
        return;
    }
    printf("  Enter ID:");
    scanf("%lld:",&sid);
    for (int i = 0; i < numStudents; i++)
    {
        if (sid==studentList[i].id)
        {
            printf("  ===================================================\n");
            printf("  Student name:  %s     Section:  %d   CGPA: %.2f \n",studentList[i].name,studentList[i].section,studentList[i].cg);
            printf("  ===================================================\n");
            found = 1;
            break;
        }

    }

    if (found==0)
    {
        printf("  Student ID %lld not found!\n  Enter valid id please!!\n", sid);
    }
}
```

The usage of searchByID function is to search any student by putting it's ID as user input.

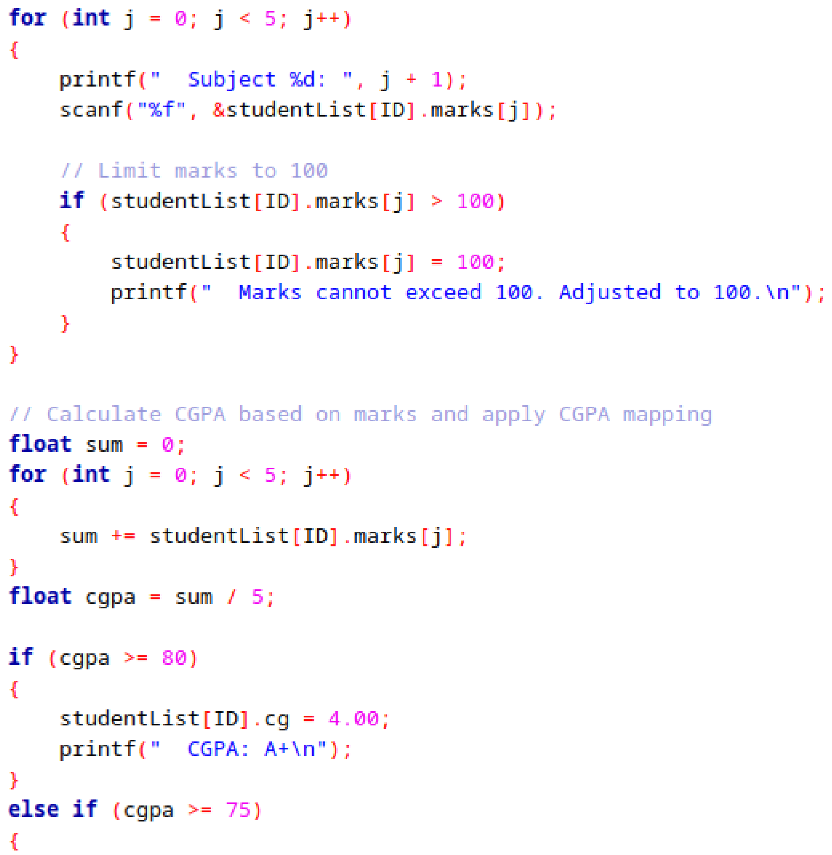


```
Student Management System
=========================

Option:
1. Add Student
2. Display Students
3. Search Student by ID
4. Add Students marks
5. Delete Student
6. Exit
Enter your choice: 3
Enter ID:143
===================================================
Student name:  Nahid     Section:  9   CGPA: 0.00
===================================================
```

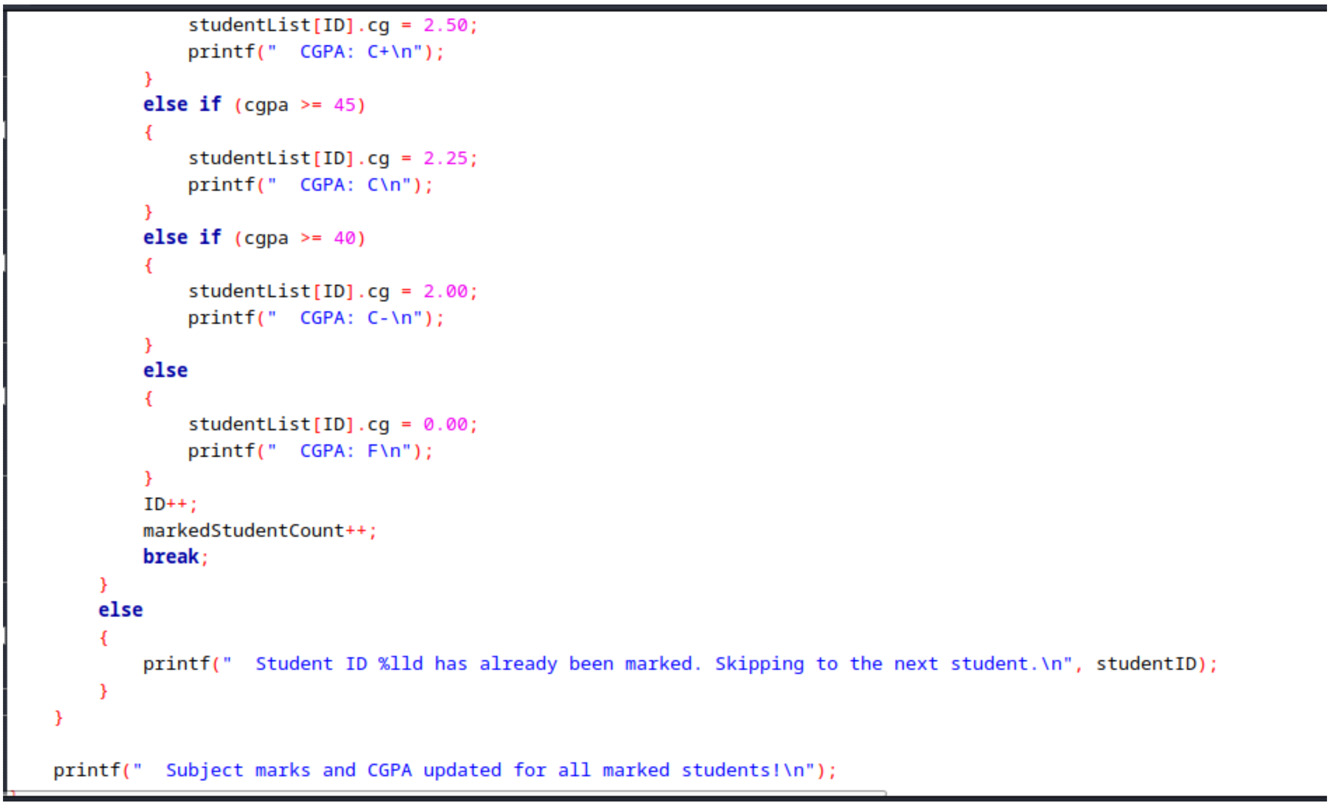
addMarks:



```c
void addMarks()
{
    if (numStudents == 0)
    {
        printf("  No students in the record!\n  Add Student then you can enter mark.");
        return;
    }

    int markedStudentCount = 0;
    printf("  Enter subject marks for all students:\n");
    for (int i = 0; i < numStudents; i++)
    {
        long long int studentID = studentList[i].id;
        bool alreadyMarked = false;

        for (int j = 0; j < markedStudentCount; j++)
        {
            if (studentList[j].id == studentID)
            {
                alreadyMarked = true;
                break;
            }
        }

        if (!alreadyMarked)
        {
            printf("  Student ID: %lld\n", studentList[ID].id);
            printf("  Enter marks for 5 subjects:\n");
```

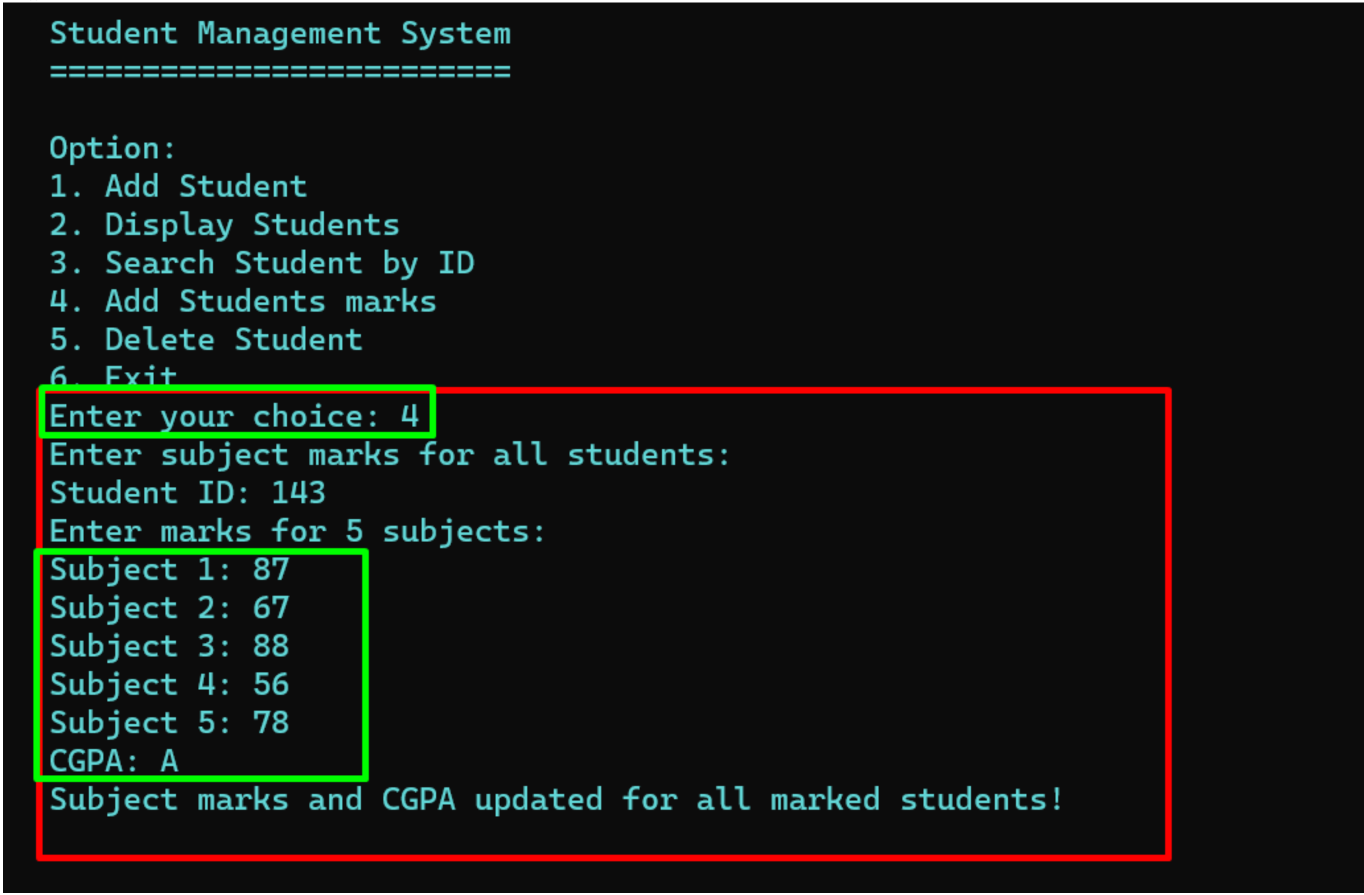


```
        for (int j = 0; j < 5; j++)
        {
            printf("  Subject %d: ", j + 1);
            scanf("%f", &studentList[ID].marks[j]);

            // Limit marks to 100
            if (studentList[ID].marks[j] > 100)
            {
                studentList[ID].marks[j] = 100;
                printf("  Marks cannot exceed 100. Adjusted to 100.\n");
            }
        }

        // Calculate CGPA based on marks and apply CGPA mapping
        float sum = 0;
        for (int j = 0; j < 5; j++)
        {
            sum += studentList[ID].marks[j];
        }
        float cgpa = sum / 5;

        if (cgpa >= 80)
        {
            studentList[ID].cg = 4.00;
            printf("  CGPA: A+\n");
        }
        else if (cgpa >= 75)
        {
```

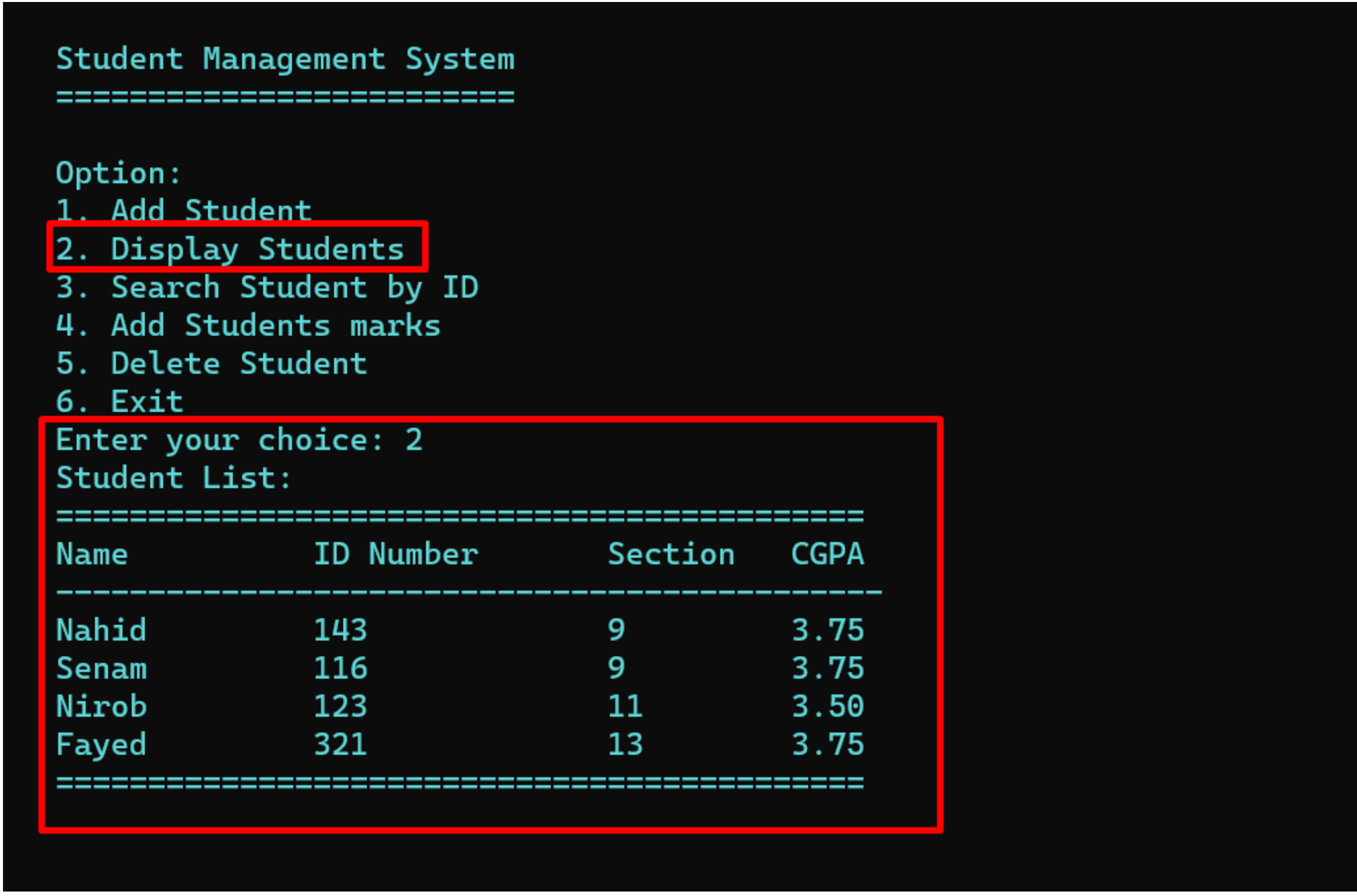


```
            studentList[ID].cg = 2.50;
            printf("  CGPA: C+\n");
        }
        else if (cgpa >= 45)
        {
            studentList[ID].cg = 2.25;
            printf("  CGPA: C\n");
        }
        else if (cgpa >= 40)
        {
            studentList[ID].cg = 2.00;
            printf("  CGPA: C-\n");
        }
        else
        {
            studentList[ID].cg = 0.00;
            printf("  CGPA: F\n");
        }
        ID++;
        markedStudentCount++;
        break;
    }
    else
    {
        printf("  Student ID %lld has already been marked. Skipping to the next student.\n", studentID);
    }
}

printf("  Subject marks and CGPA updated for all marked students!\n");
```

The usage of addMarks function is to add marks and to calculate CGPA of a student through ID and CGPA is directly calculated by putting the marks through user input.

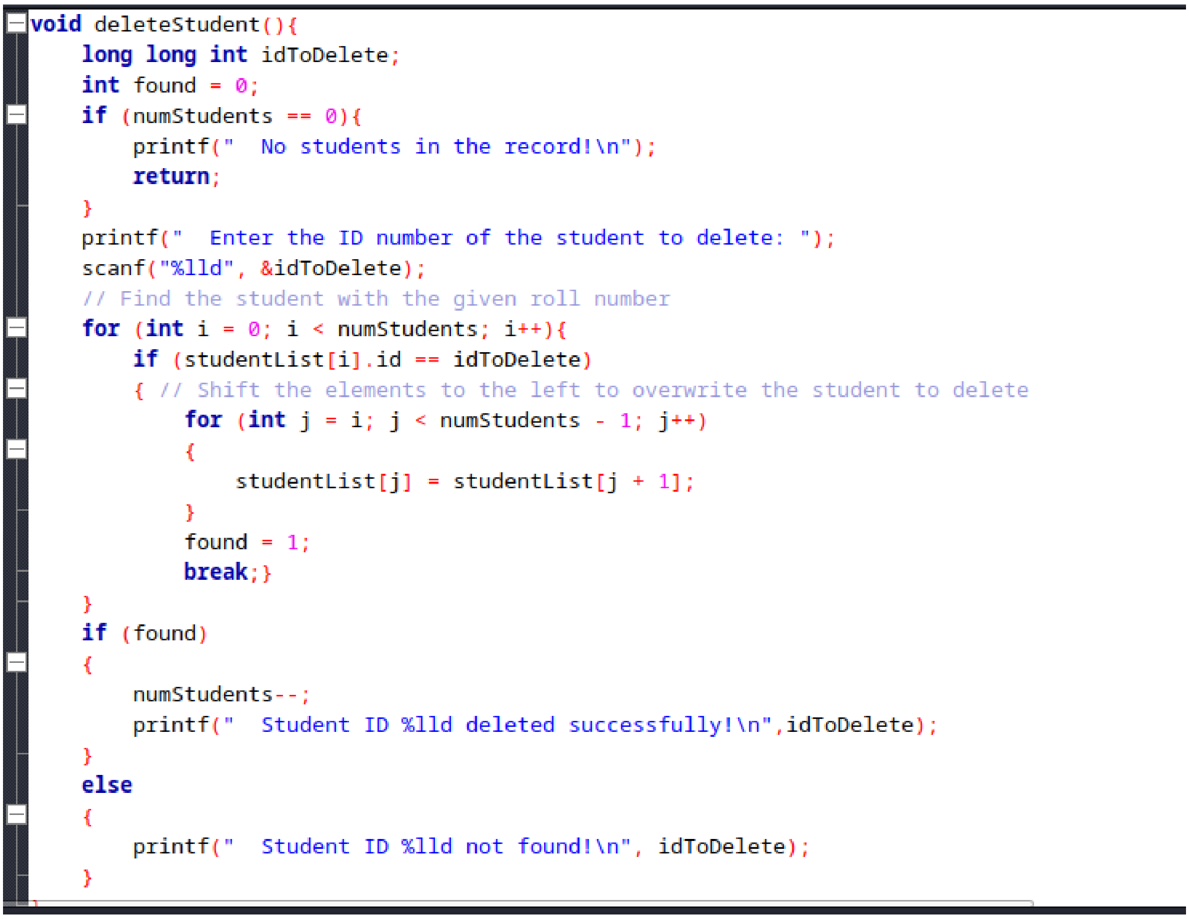


```
Student Management System
=========================

Option:
1. Add Student
2. Display Students
3. Search Student by ID
4. Add Students marks
5. Delete Student
6. Exit
Enter your choice: 4
Enter subject marks for all students:
Student ID: 143
Enter marks for 5 subjects:
Subject 1: 87
Subject 2: 67
Subject 3: 88
Subject 4: 56
Subject 5: 78
CGPA: A
Subject marks and CGPA updated for all marked students!
```



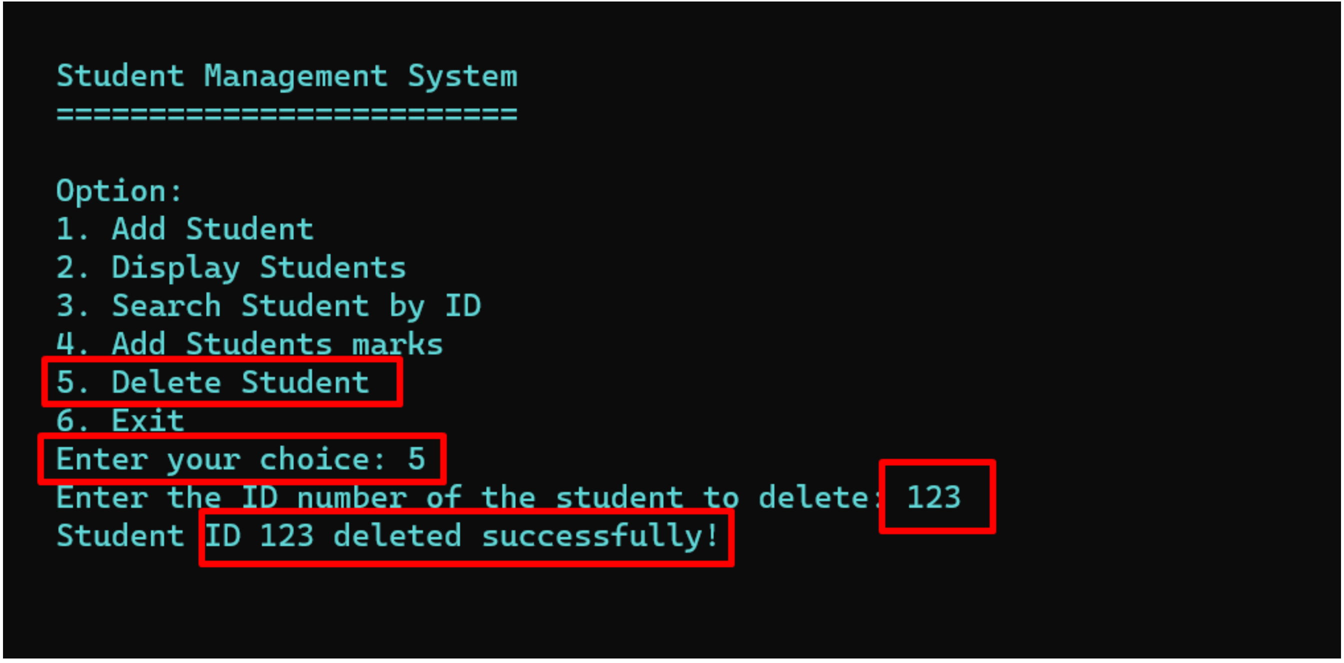
```
Student Management System
=========================

Option:
1. Add Student
2. Display Students
3. Search Student by ID
4. Add Students marks
5. Delete Student
6. Exit
Enter your choice: 2
Student List:
============================================
Name          ID Number        Section    CGPA
---------------------------------------------
Nahid         143              9          3.75
Senam         116              9          3.75
Nirob         123              11         3.50
Fayed         321              13         3.75
============================================
```

deleteStudent:

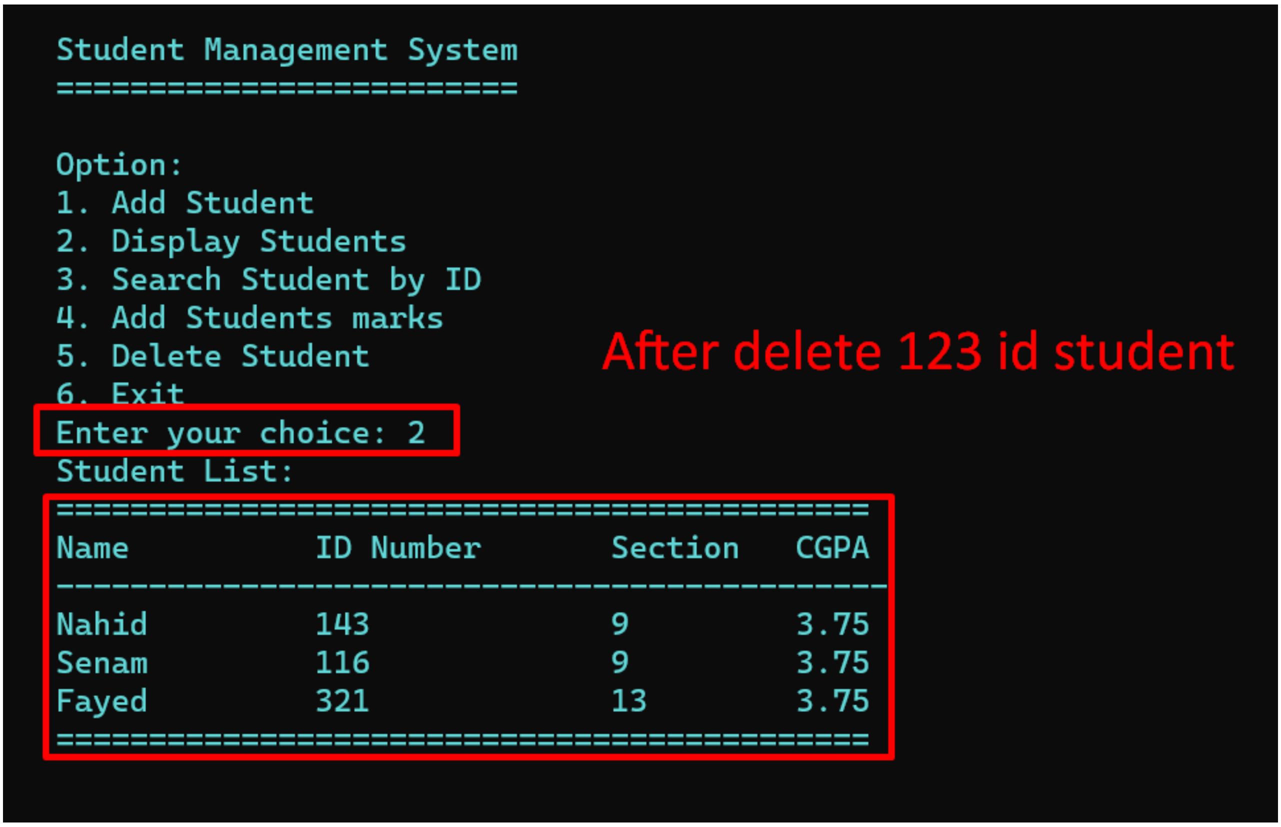


```
void deleteStudent(){
    long long int idToDelete;
    int found = 0;
    if (numStudents == 0){
        printf("  No students in the record!\n");
        return;
    }
    printf("  Enter the ID number of the student to delete: ");
    scanf("%lld", &idToDelete);
    // Find the student with the given roll number
    for (int i = 0; i < numStudents; i++){
        if (studentList[i].id == idToDelete)
        { // Shift the elements to the left to overwrite the student to delete
            for (int j = i; j < numStudents - 1; j++)
            {
                studentList[j] = studentList[j + 1];
            }
            found = 1;
            break;}
    }
    if (found)
    {
        numStudents--;
        printf("  Student ID %lld deleted successfully!\n",idToDelete);
    }
    else
    {
        printf("  Student ID %lld not found!\n", idToDelete);
    }
```

The usage of deleteStudent function is to delete the data of stored students by inputing the ID of a student.



```
Student Management System
=========================

Option:
1. Add Student
2. Display Students
3. Search Student by ID
4. Add Students marks
5. Delete Student
6. Exit
Enter your choice: 5
Enter the ID number of the student to delete: 123
Student ID 123 deleted successfully!
```

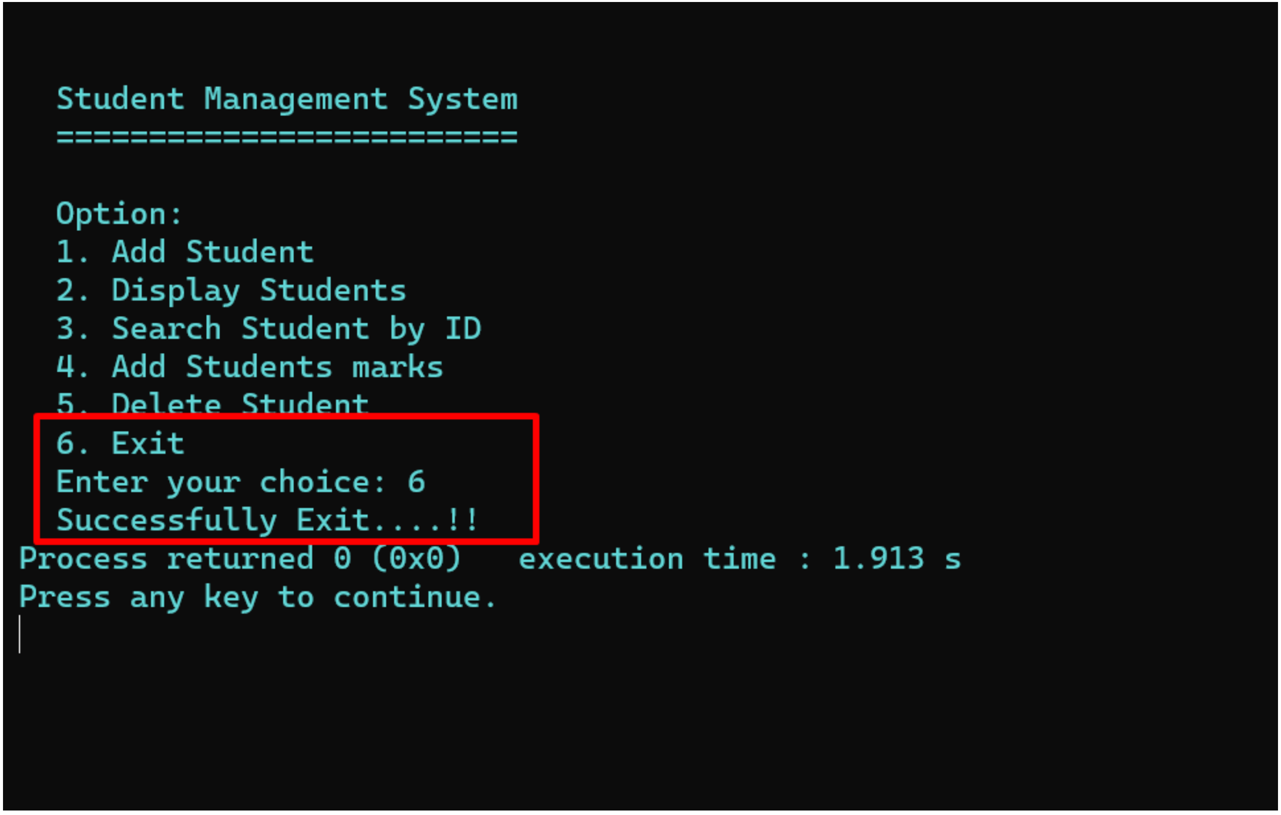


```
Student Management System
=========================

Option:
1. Add Student
2. Display Students
3. Search Student by ID
4. Add Students marks
5. Delete Student
6. Exit
Enter your choice: 2
Student List:
============================================
Name          ID Number       Section   CGPA
---------------------------------------------
Nahid         143             9         3.75
Senam         116             9         3.75
Fayed         321             13        3.75
============================================
```

After delete 123 id student

Exit:

Choosing the Exit option exits the program with a message displaying ”Successfully Exit!!!”



```
  Student Management System
  =========================

  Option:
  1. Add Student
  2. Display Students
  3. Search Student by ID
  4. Add Students marks
  5. Delete Student
  6. Exit
  Enter your choice: 6
  Successfully Exit....!!
Process returned 0 (0x0)   execution time : 1.913 s
Press any key to continue.
```

3.5 Design and Implementation

The Student Management System project reflect a structured and user-friendly approach. The code is organized into modular functions for distinct functionalities. The project follows a console-based interface. The implementation employs features like file handling for user records and time functions for speed calculations. The code structure is well-commented, promoting readability and easy maintenance. The design incorporates a welcoming interface, and the code adheres to good programming practices, enhancing its effectiveness as a data management assessment tool. Overall, the project seamlessly integrates functionality.

3.6 Evaluation

The Student Management System project undergoes a rigorous evaluation encompassing performance metrics, user experience, dataset analysis, result consistency, security measures, and user feedback integration. Its scoring algorithms, measuring Words Per Minute and accuracy, ensure reliable data storing. The user-friendly interface, inclusive features, and real-time feedback enhance the overall user experience. Dataset diversity and effective preprocessing contribute to accurate evaluations. Security measures, including data encryption, fortify user privacy. Continuous feedback integration and alignment with project goals demonstrate a commitment to improvement. This holistic evaluation ensures the Student Management System project's effectiveness, reliability, and ongoing enhancement.

3.7 Result Evaluation

The Student Management System project undergoes a rigorous evaluation encompassing performance metrics, user experience, dataset analysis, result consistency, security measures, and user feedback integration. Its calculating algorithms, putting data and accuracy of storing, ensure reliable management assesment. The user-friendly interface, inclusive features, and real-time feedback enhance the overall user experience. Dataset diversity and effective preprocessing contribute to accurate evaluations. Security measures, including data encryption, fortify user privacy. Continuous feedback integration and alignment with project goals demonstrate a commitment to improvement. This holistic evaluation ensures the Student Management System project's effectiveness, reliability, and ongoing enhancement.

3.8 Summary

This chapter of the project report outlines the structured approach followed in the Student Management System project. The dataset section provides details about the data used, while the flow of work explains the project's methodology. The evaluation section encompasses various aspects, including performance metrics, user experience, dataset analysis, result consistency, security measures, and user feedback integration. The result evaluation highlights the project's effectiveness and continuous improvement. The results and discussion section showcases the success of the project in accurate data management assessments.Overall, this chapter ensures clarity and coherence throughout

Chapter 4

4 Conclusion

4.1 Conclusion

The Student Management System represents a comprehensive solution to enhance the operational capabilities of storing and managing students data. The design and implementation process prioritized user needs, scalability, and data integrity. Ongoing support and maintenance will ensure the continued success of the system in meeting the evolving requirements of the retail industry.

4.2 Future Works and Direction

The Student Management System significantly transformed traditional student data entry, enhancing efficiency on putting and calculating result and faster display of stored data. Future enhancements and scalability considerations promise continued growth and adaptability to evolving needs. This report outlines the detailed aspects of the Student Management System's development, from objectives and scope to implementation, evaluation, and future recommendations. Adjust and add details as per your project's specifics.

References

Project Github link: ***github.com/developernahid/sdp-project/blob/nahid/Final.c***

Here are some of the sites from we have taken help from:

Website-1: **www.geeksforgeeks.org**

Website-2:**www.w3schools.com**

Website-3:**www.google.com**

Website-4:**www.youtube.com**

Website-5:**www.wikipedia.org**