# FINAL REPORT

## Student Attendance Calculator

Group No. - 5

GROUP MEMBERS :-

Ishaant Kumar Singh - 12203987
Shivam Shekhar - 12204084
Gaurav  Pathak - 12204085
Steve S Joy - 12205035

# Introduction

Student attendance calculator is an application that will help the students to calculate their attendance in each course and the aggregate attendance based on the number of based on number of lectures delivered till that week and no of lectures student have missed in a particular course. This application will help the student to have a check on his attendance for each course in order to avoid the detention from the final exams

# Module Explanation

1. Login/Authentication Module:

This module will allow the user to log in and authenticate themselves to access the student attendance calculator application. The user will be required to enter their login credentials, such as a username and password, to gain access to the application. The authentication module will ensure that only authorized users can access and modify the student attendance data.

2. Insert Details of Student Module:

This module will allow the user to add new student details to the attendance calculator system. The user will be required to enter the student's name, course name, lecture attendance details, and other necessary information. The module will then store this data in the system's database for future use.

3. Display All Details Module:

This module will display all the attendance details of each student in the system. The user can view this information by selecting the display all option from the menu. The module will retrieve the data from the database and present it to the user in a readable format.

4. Search Specific Information Module:

This module will allow the user to search for a specific student's attendance information by entering their name or course name. The module will retrieve the data from the database and display it to the user in a readable format.

5. Delete/Modify Any Student Detail Module:

This module will allow the user to delete or modify the attendance details of any student in the system. The user can select the delete or modify option from the menu and enter the student's name or course name. The module will then retrieve the data from the database and allow the user to delete or modify it as required. This module will help the user to keep the attendance data up-to-date and accurate.

6. Save Data Module:

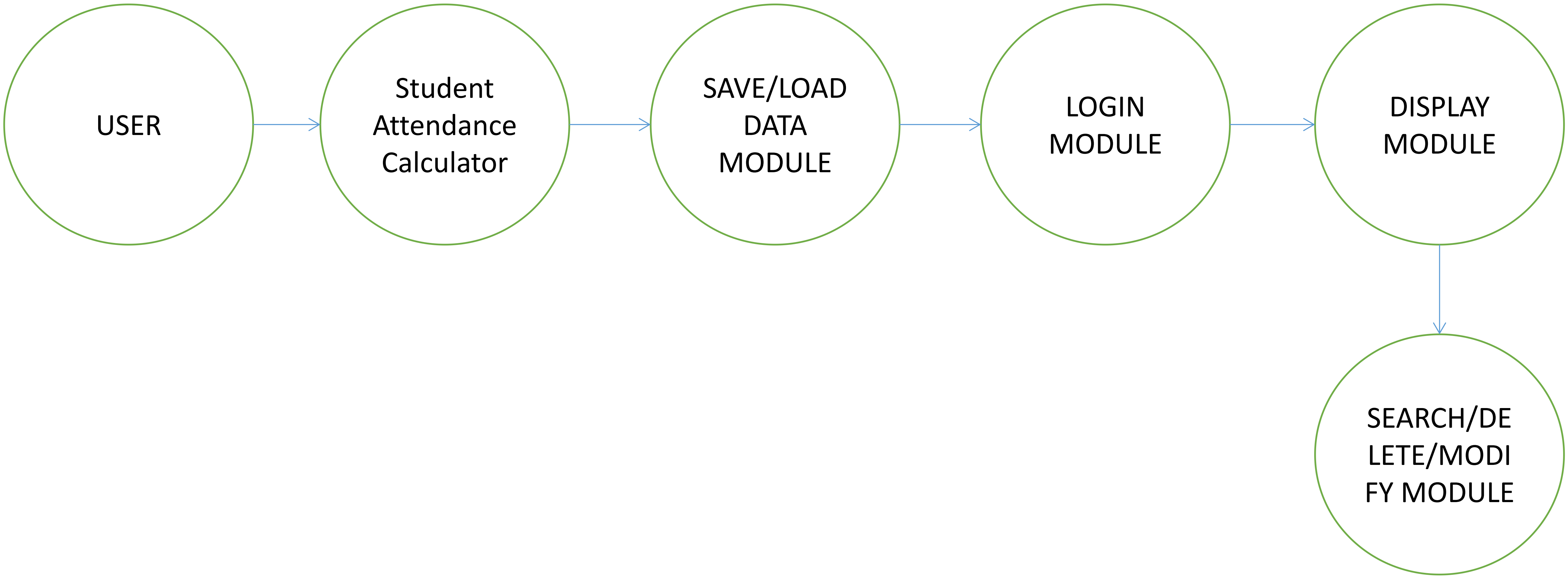
This module will allow the user to save the student attendance data to a file or database. The module will retrieve the data from the database and save it to a file in a readable format. This module will ensure that the attendance data is backed up and can be retrieved in case of any system failure or data loss.

7. Load Data Module:

This module will allow the user to load the student attendance data from a saved file or database. The user can select the load option from the menu, and the module will retrieve the data from the file and load it into the system's database. This module will ensure that the user can access and modify the attendance data even after a system restart or shutdown.

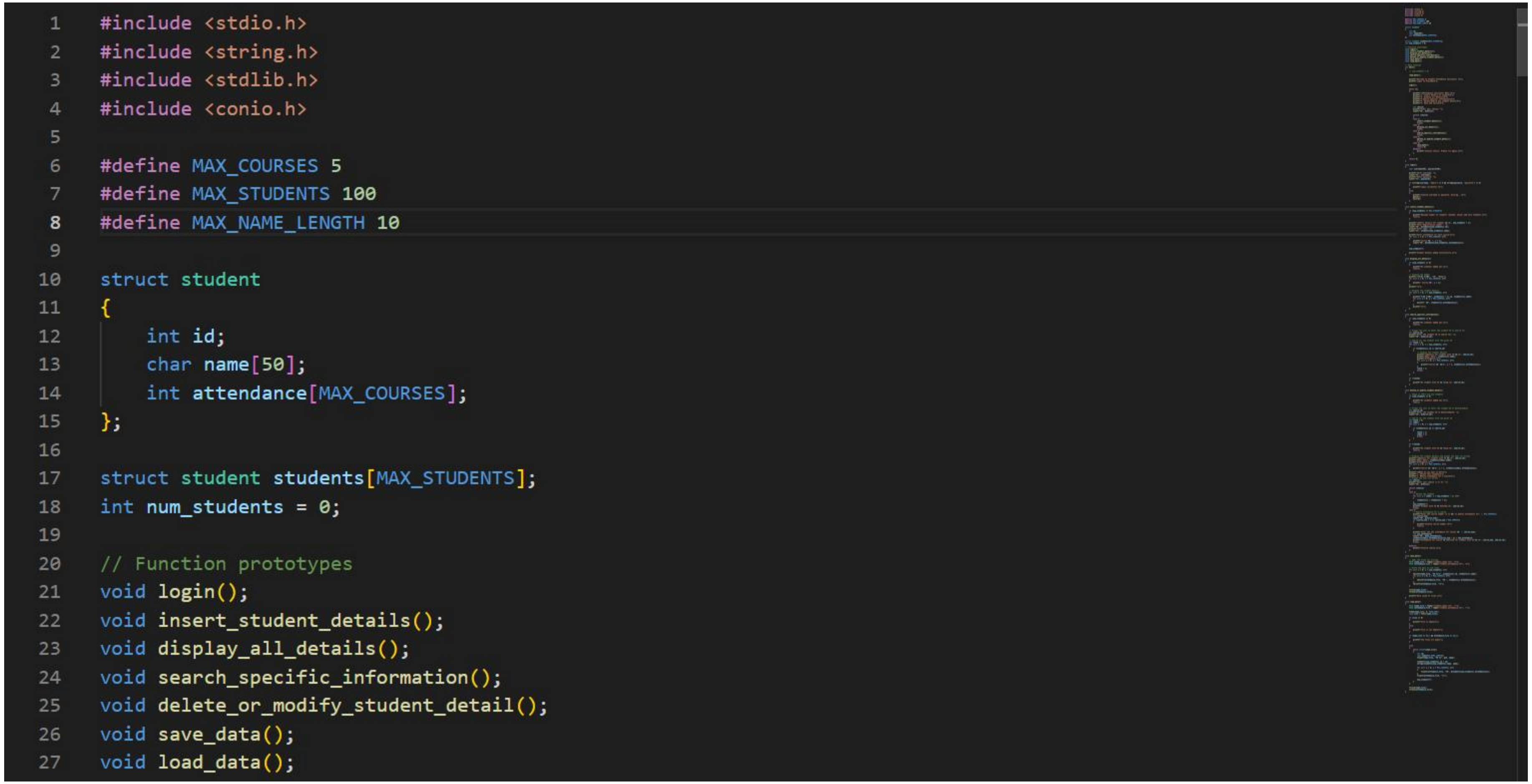
# Data Flow Diagram(DFD)

## Level Zero



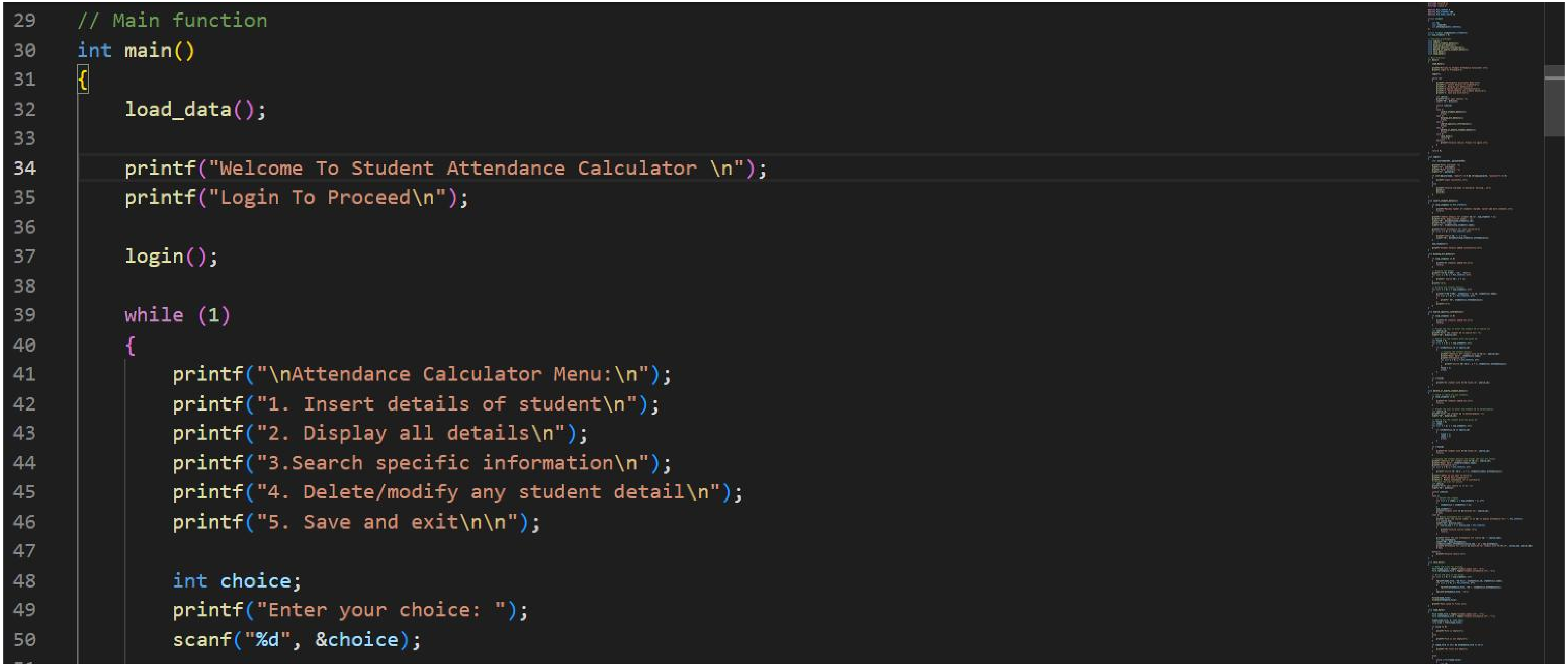
# Code Snapshots

## Header Files, Main Structure and Function Initializations

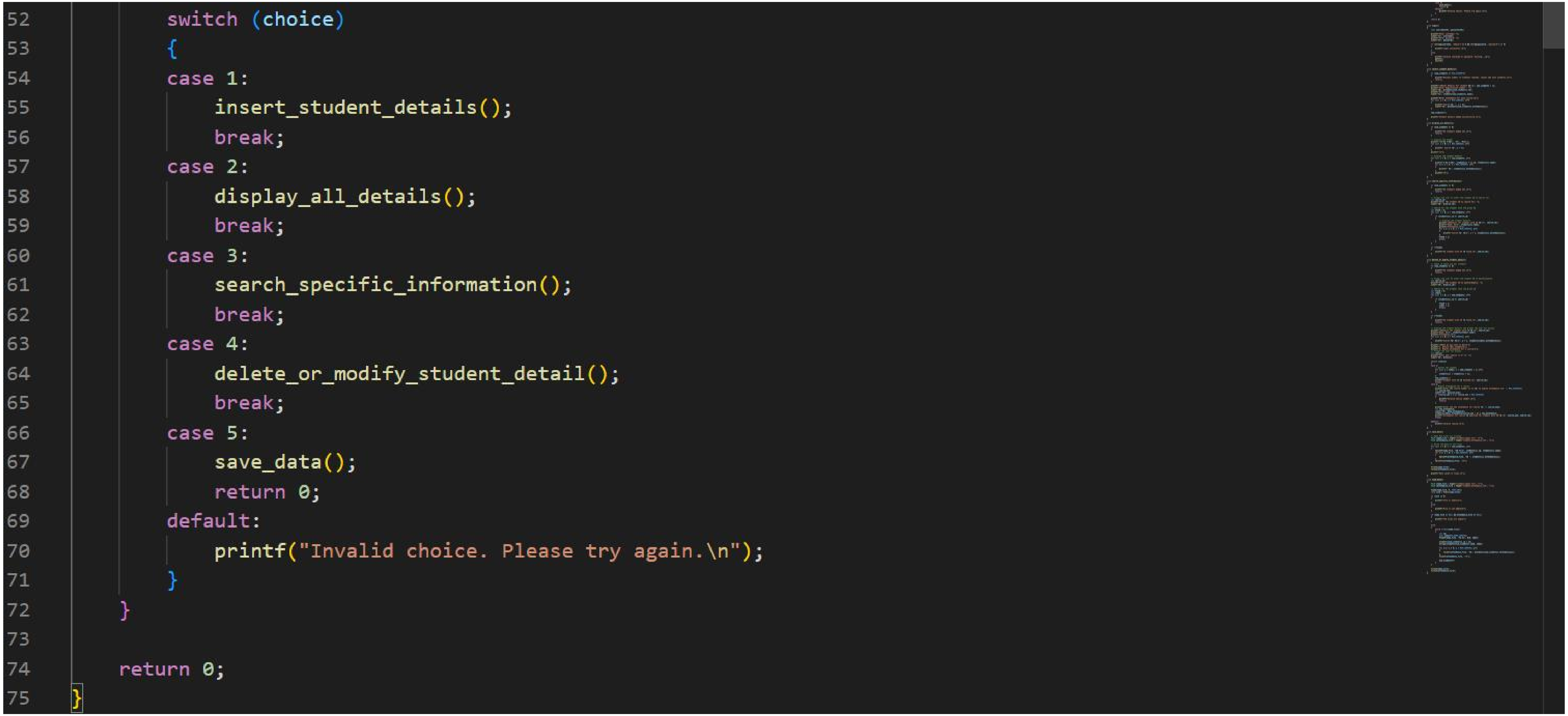


```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <conio.h>

#define MAX_COURSES 5
#define MAX_STUDENTS 100
#define MAX_NAME_LENGTH 10

struct student
{
    int id;
    char name[50];
    int attendance[MAX_COURSES];
};

struct student students[MAX_STUDENTS];
int num_students = 0;

// Function prototypes
void login();
void insert_student_details();
void display_all_details();
void search_specific_information();
void delete_or_modify_student_detail();
void save_data();
void load_data();
```

# Main Function

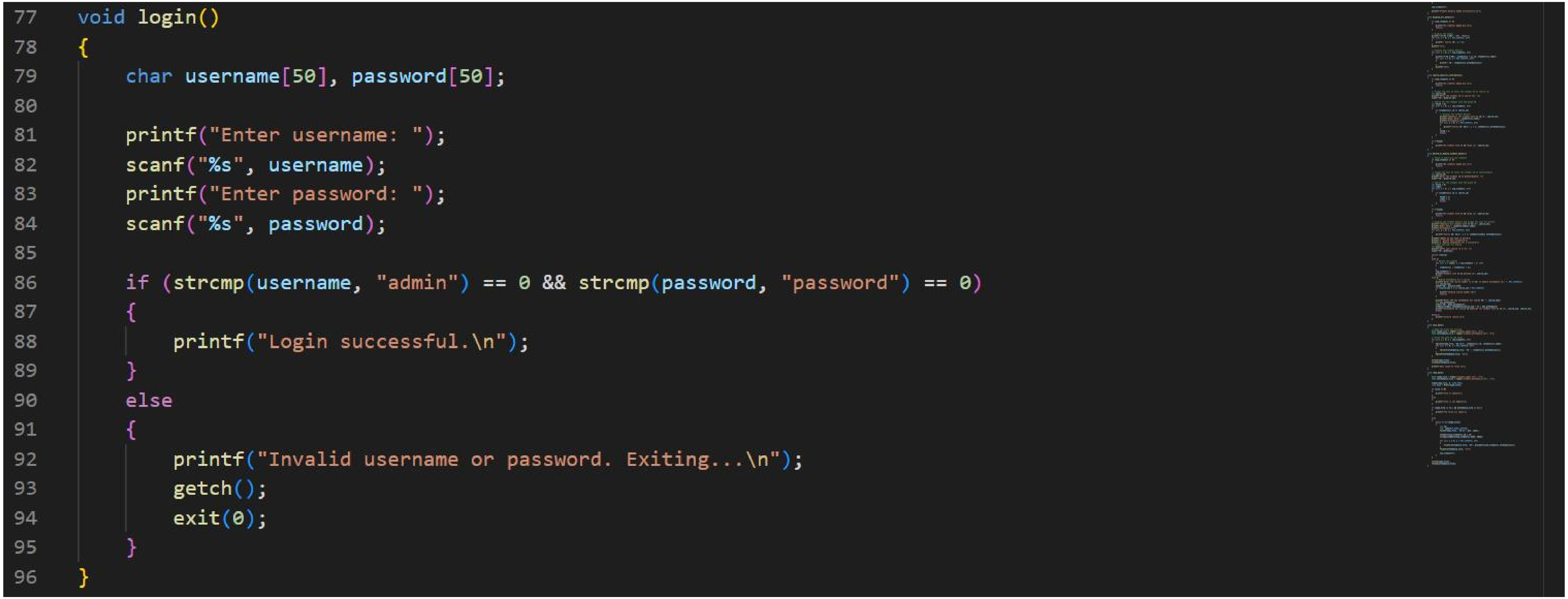


```c
// Main function
int main()
{
    load_data();

    printf("Welcome To Student Attendance Calculator \n");
    printf("Login To Proceed\n");

    login();

    while (1)
    {
        printf("\nAttendance Calculator Menu:\n");
        printf("1. Insert details of student\n");
        printf("2. Display all details\n");
        printf("3.Search specific information\n");
        printf("4. Delete/modify any student detail\n");
        printf("5. Save and exit\n\n");

        int choice;
        printf("Enter your choice: ");
        scanf("%d", &choice);
```



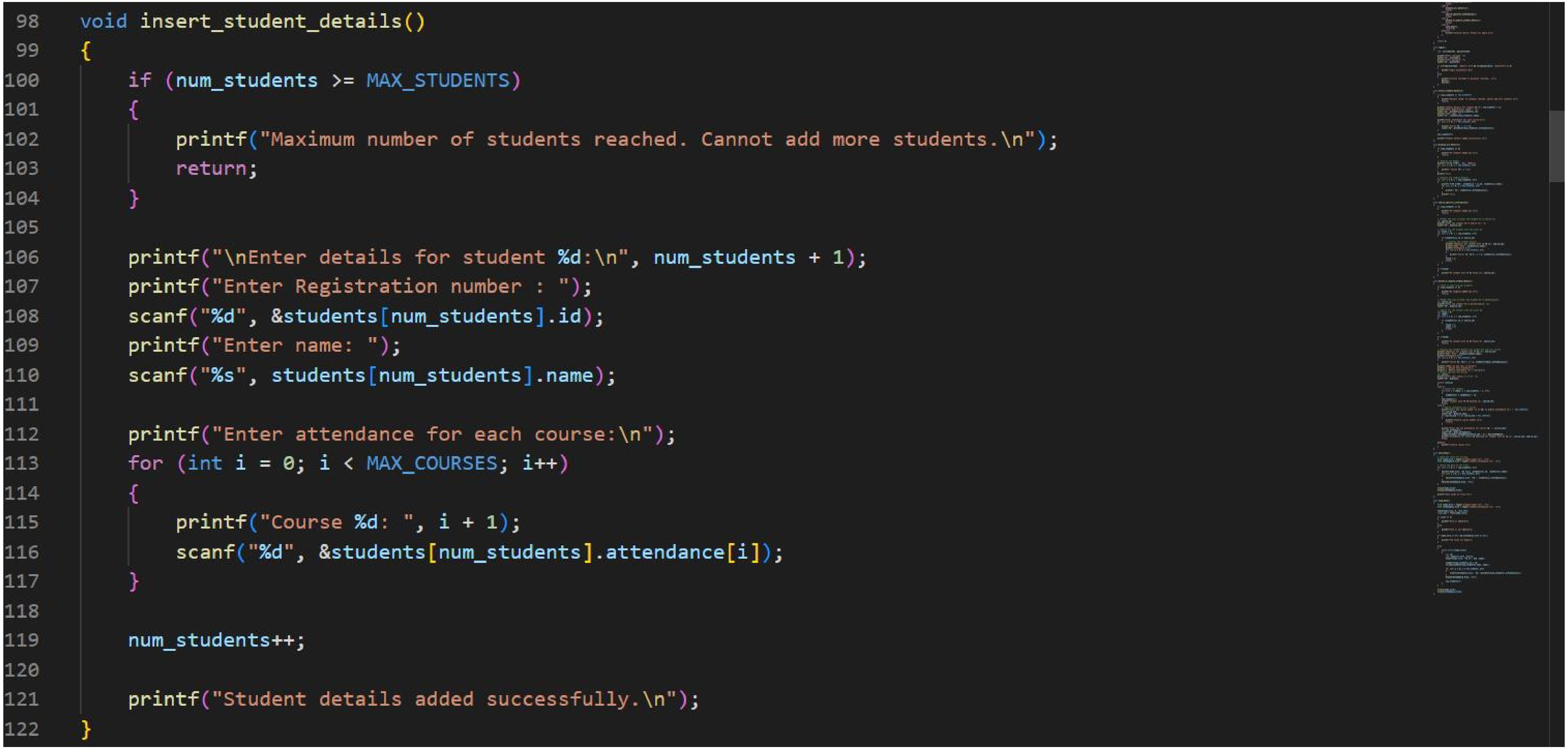
```c
        switch (choice)
        {
        case 1:
            insert_student_details();
            break;
        case 2:
            display_all_details();
            break;
        case 3:
            search_specific_information();
            break;
        case 4:
            delete_or_modify_student_detail();
            break;
        case 5:
            save_data();
            return 0;
        default:
            printf("Invalid choice. Please try again.\n");
        }
    }

    return 0;
}
```

# Login Function



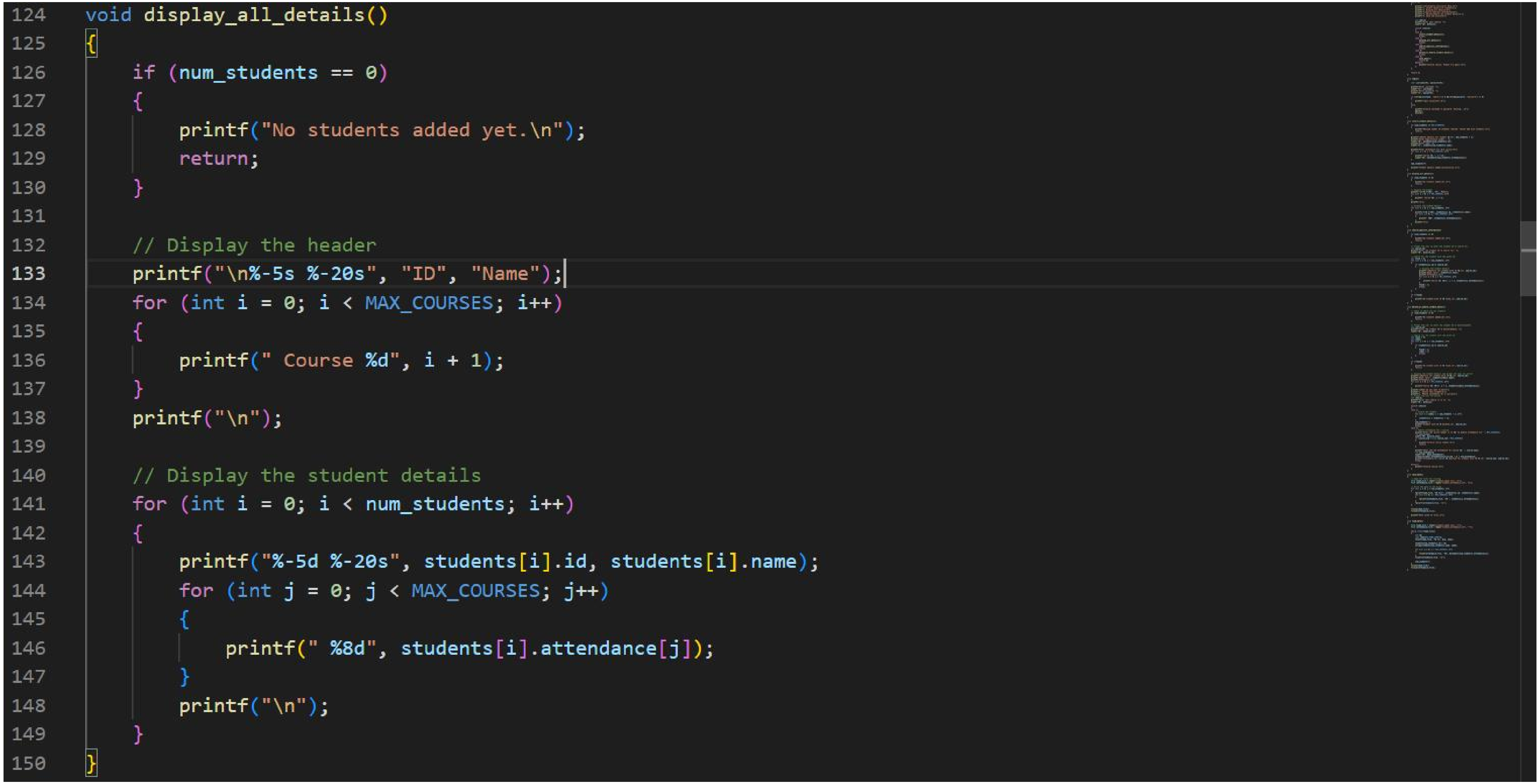
```c
void login()
{
    char username[50], password[50];

    printf("Enter username: ");
    scanf("%s", username);
    printf("Enter password: ");
    scanf("%s", password);

    if (strcmp(username, "admin") == 0 && strcmp(password, "password") == 0)
    {
        printf("Login successful.\n");
    }
    else
    {
        printf("Invalid username or password. Exiting...\n");
        getch();
        exit(0);
    }
}
```

# Insert Student Detail Function



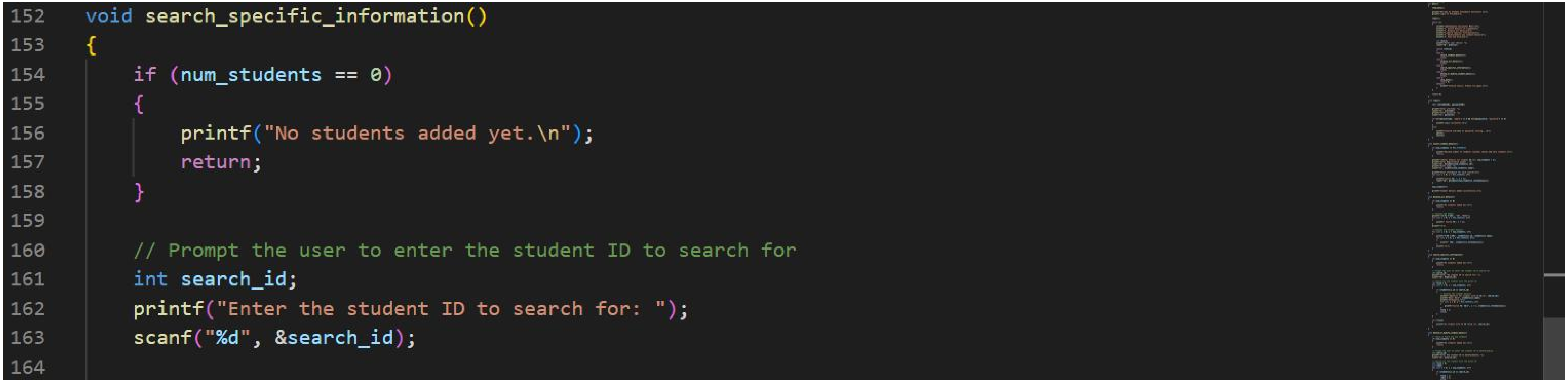
```c
void insert_student_details()
{
    if (num_students >= MAX_STUDENTS)
    {
        printf("Maximum number of students reached. Cannot add more students.\n");
        return;
    }

    printf("\nEnter details for student %d:\n", num_students + 1);
    printf("Enter Registration number : ");
    scanf("%d", &students[num_students].id);
    printf("Enter name: ");
    scanf("%s", students[num_students].name);

    printf("Enter attendance for each course:\n");
    for (int i = 0; i < MAX_COURSES; i++)
    {
        printf("Course %d: ", i + 1);
        scanf("%d", &students[num_students].attendance[i]);
    }

    num_students++;

    printf("Student details added successfully.\n");
}
```

# Display All Details Function

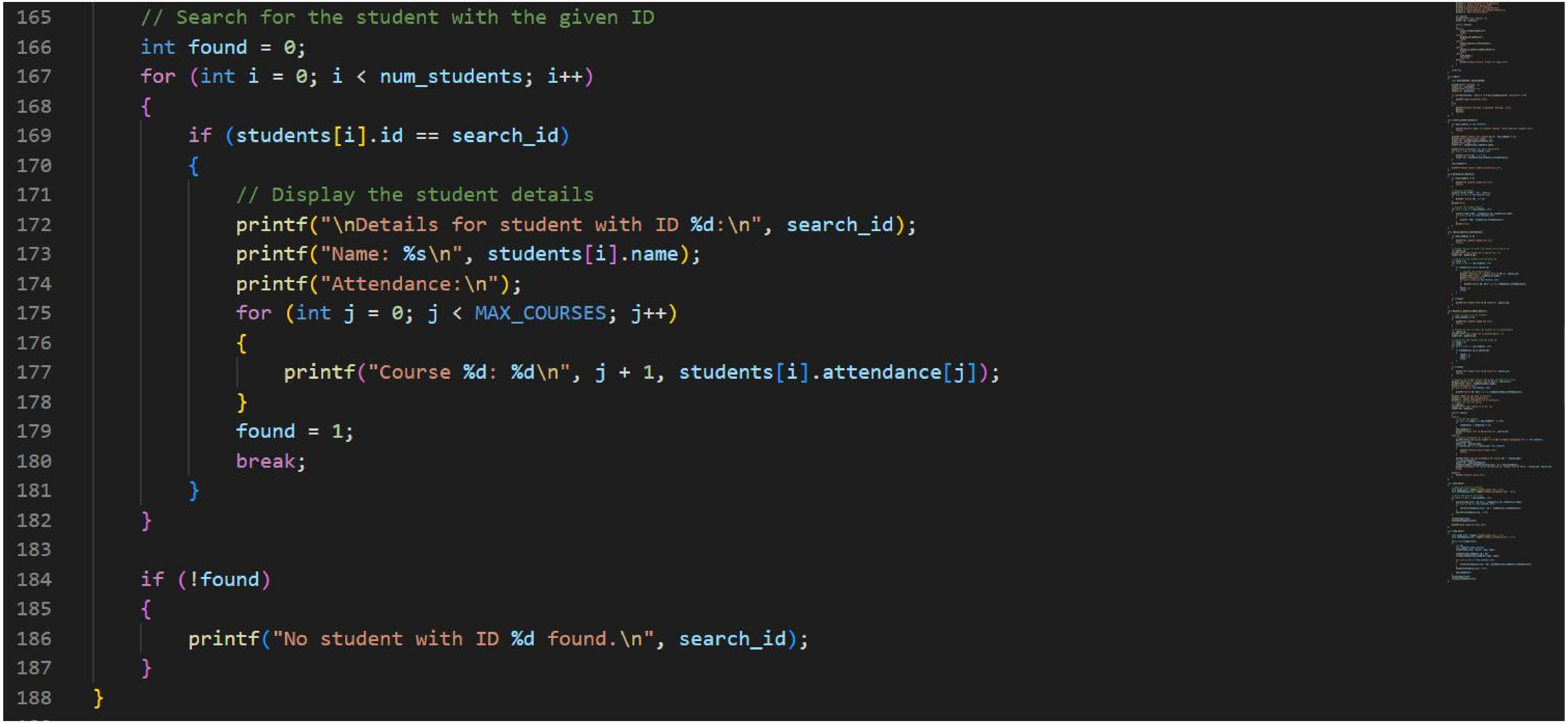


```c
void display_all_details()
{
    if (num_students == 0)
    {
        printf("No students added yet.\n");
        return;
    }

    // Display the header
    printf("\n%-5s %-20s", "ID", "Name");
    for (int i = 0; i < MAX_COURSES; i++)
    {
        printf(" Course %d", i + 1);
    }
    printf("\n");

    // Display the student details
    for (int i = 0; i < num_students; i++)
    {
        printf("%-5d %-20s", students[i].id, students[i].name);
        for (int j = 0; j < MAX_COURSES; j++)
        {
            printf(" %8d", students[i].attendance[j]);
        }
        printf("\n");
    }
}
```

# Search Specific Information Function

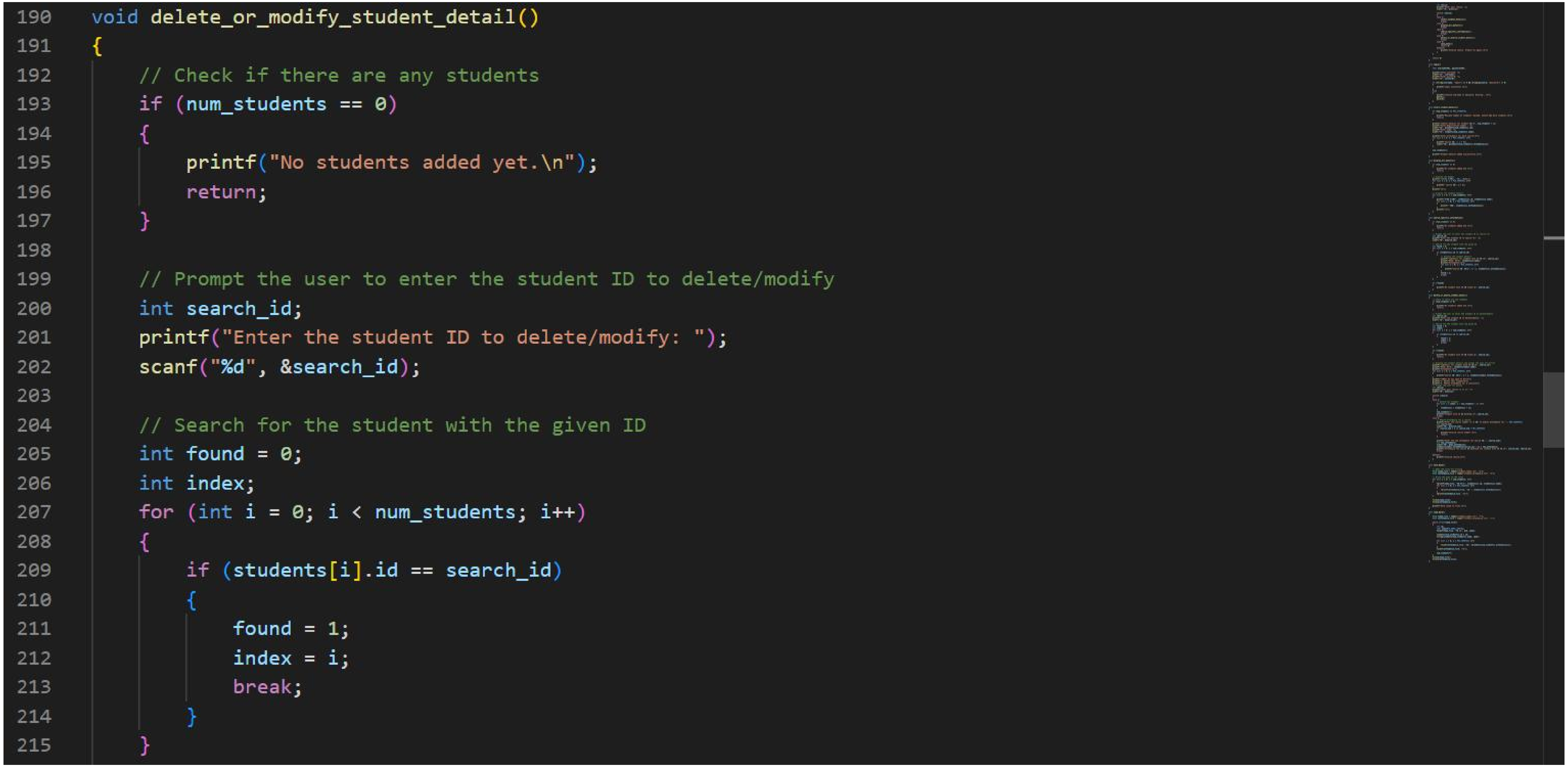


```c
void search_specific_information()
{
    if (num_students == 0)
    {
        printf("No students added yet.\n");
        return;
    }

    // Prompt the user to enter the student ID to search for
    int search_id;
    printf("Enter the student ID to search for: ");
    scanf("%d", &search_id);

```

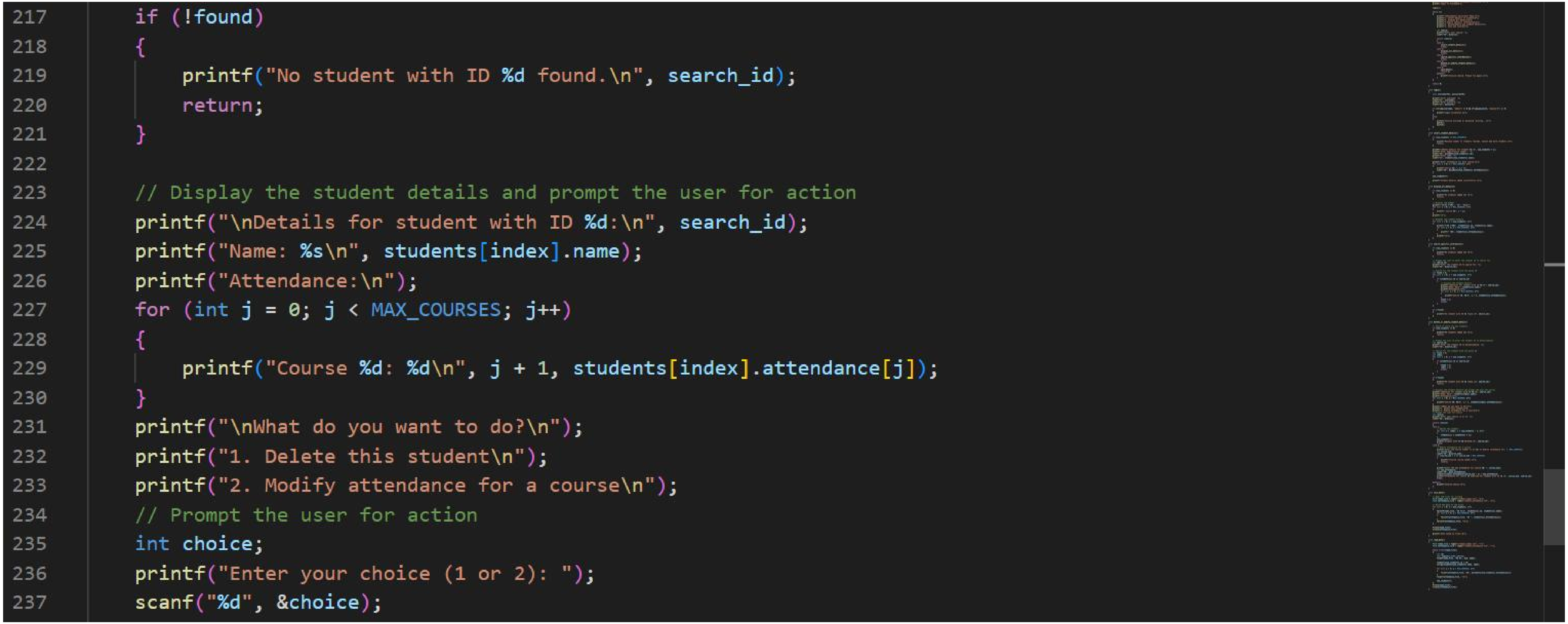


```c
    // Search for the student with the given ID
    int found = 0;
    for (int i = 0; i < num_students; i++)
    {
        if (students[i].id == search_id)
        {
            // Display the student details
            printf("\nDetails for student with ID %d:\n", search_id);
            printf("Name: %s\n", students[i].name);
            printf("Attendance:\n");
            for (int j = 0; j < MAX_COURSES; j++)
            {
                printf("Course %d: %d\n", j + 1, students[i].attendance[j]);
            }
            found = 1;
            break;
        }
    }

    if (!found)
    {
        printf("No student with ID %d found.\n", search_id);
    }
}
```

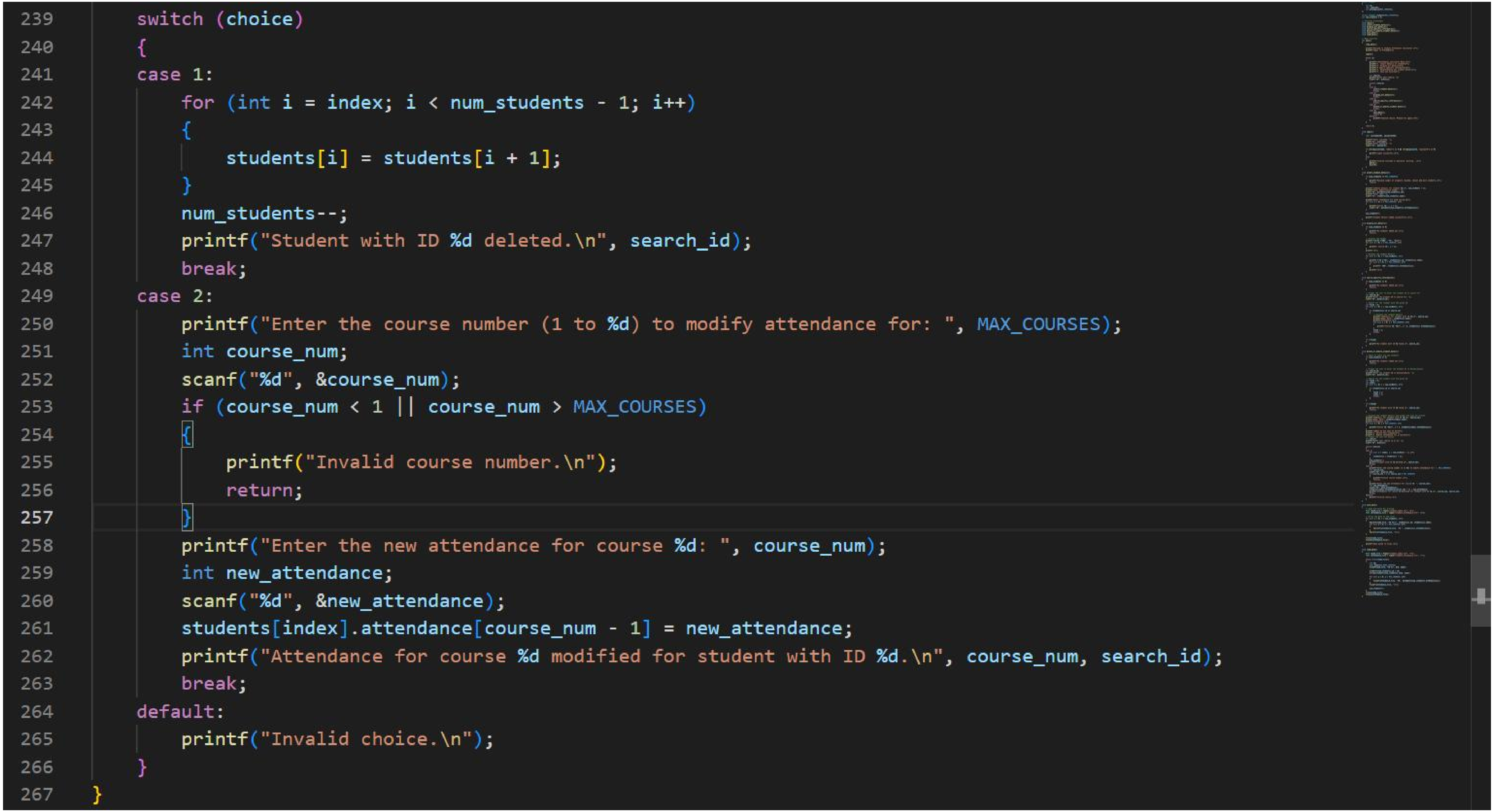
# Delete Or Modify Data Function



```c
void delete_or_modify_student_detail()
{
    // Check if there are any students
    if (num_students == 0)
    {
        printf("No students added yet.\n");
        return;
    }

    // Prompt the user to enter the student ID to delete/modify
    int search_id;
    printf("Enter the student ID to delete/modify: ");
    scanf("%d", &search_id);

    // Search for the student with the given ID
    int found = 0;
    int index;
    for (int i = 0; i < num_students; i++)
    {
        if (students[i].id == search_id)
        {
            found = 1;
            index = i;
            break;
        }
    }
```

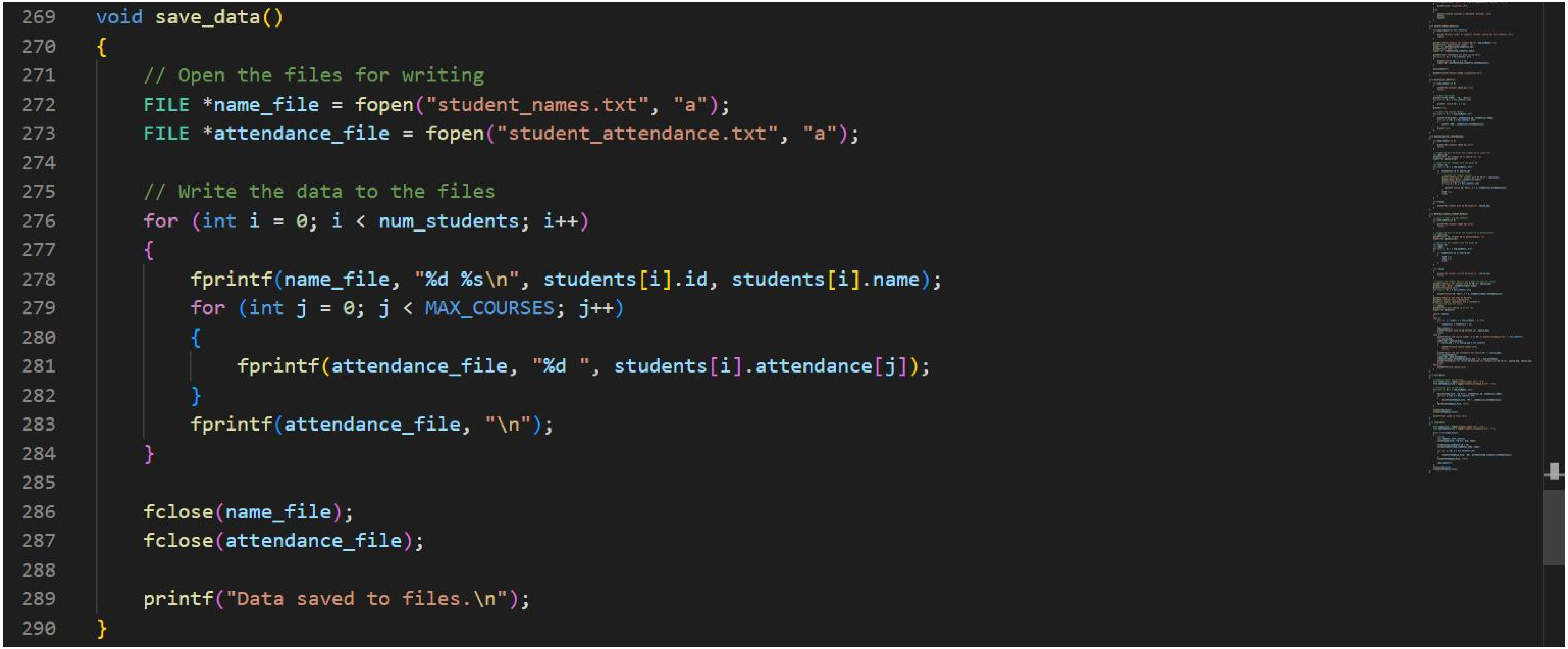


```c
    if (!found)
    {
        printf("No student with ID %d found.\n", search_id);
        return;
    }

    // Display the student details and prompt the user for action
    printf("\nDetails for student with ID %d:\n", search_id);
    printf("Name: %s\n", students[index].name);
    printf("Attendance:\n");
    for (int j = 0; j < MAX_COURSES; j++)
    {
        printf("Course %d: %d\n", j + 1, students[index].attendance[j]);
    }
    printf("\nWhat do you want to do?\n");
    printf("1. Delete this student\n");
    printf("2. Modify attendance for a course\n");
    // Prompt the user for action
    int choice;
    printf("Enter your choice (1 or 2): ");
    scanf("%d", &choice);
```



```c
    switch (choice)
    {
    case 1:
        for (int i = index; i < num_students - 1; i++)
        {
            students[i] = students[i + 1];
        }
        num_students--;
        printf("Student with ID %d deleted.\n", search_id);
        break;
    case 2:
        printf("Enter the course number (1 to %d) to modify attendance for: ", MAX_COURSES);
        int course_num;
        scanf("%d", &course_num);
        if (course_num < 1 || course_num > MAX_COURSES)
        {
            printf("Invalid course number.\n");
            return;
        }
        printf("Enter the new attendance for course %d: ", course_num);
        int new_attendance;
        scanf("%d", &new_attendance);
        students[index].attendance[course_num - 1] = new_attendance;
        printf("Attendance for course %d modified for student with ID %d.\n", course_num, search_id);
        break;
    default:
        printf("Invalid choice.\n");
    }
}
```

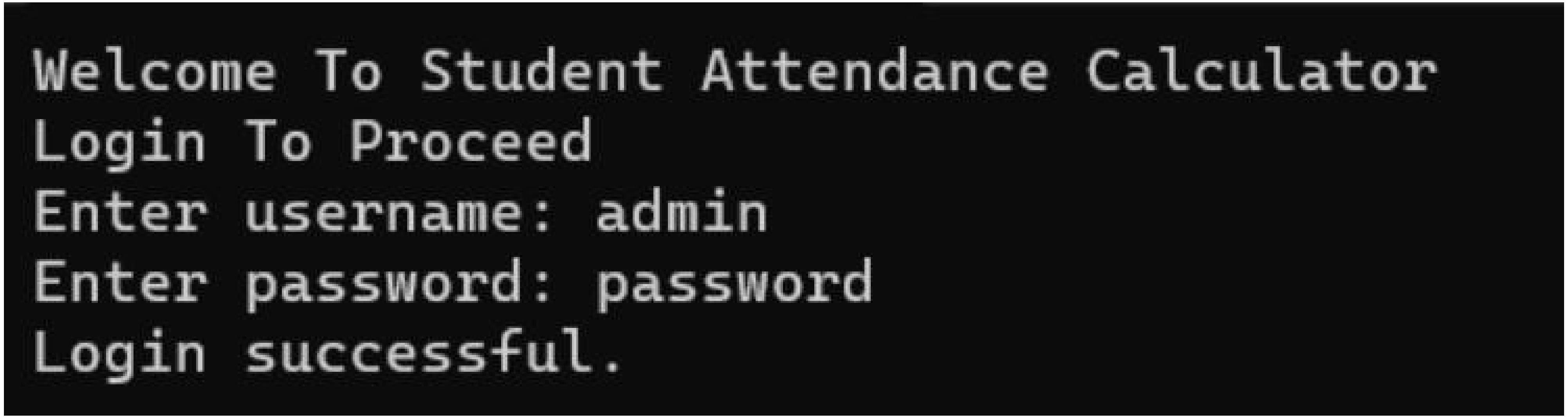
# Save Data Function



```c
void save_data()
{
    // Open the files for writing
    FILE *name_file = fopen("student_names.txt", "a");
    FILE *attendance_file = fopen("student_attendance.txt", "a");

    // Write the data to the files
    for (int i = 0; i < num_students; i++)
    {
        fprintf(name_file, "%d %s\n", students[i].id, students[i].name);
        for (int j = 0; j < MAX_COURSES; j++)
        {
            fprintf(attendance_file, "%d ", students[i].attendance[j]);
        }
        fprintf(attendance_file, "\n");
    }

    fclose(name_file);
    fclose(attendance_file);

    printf("Data saved to files.\n");
}
```

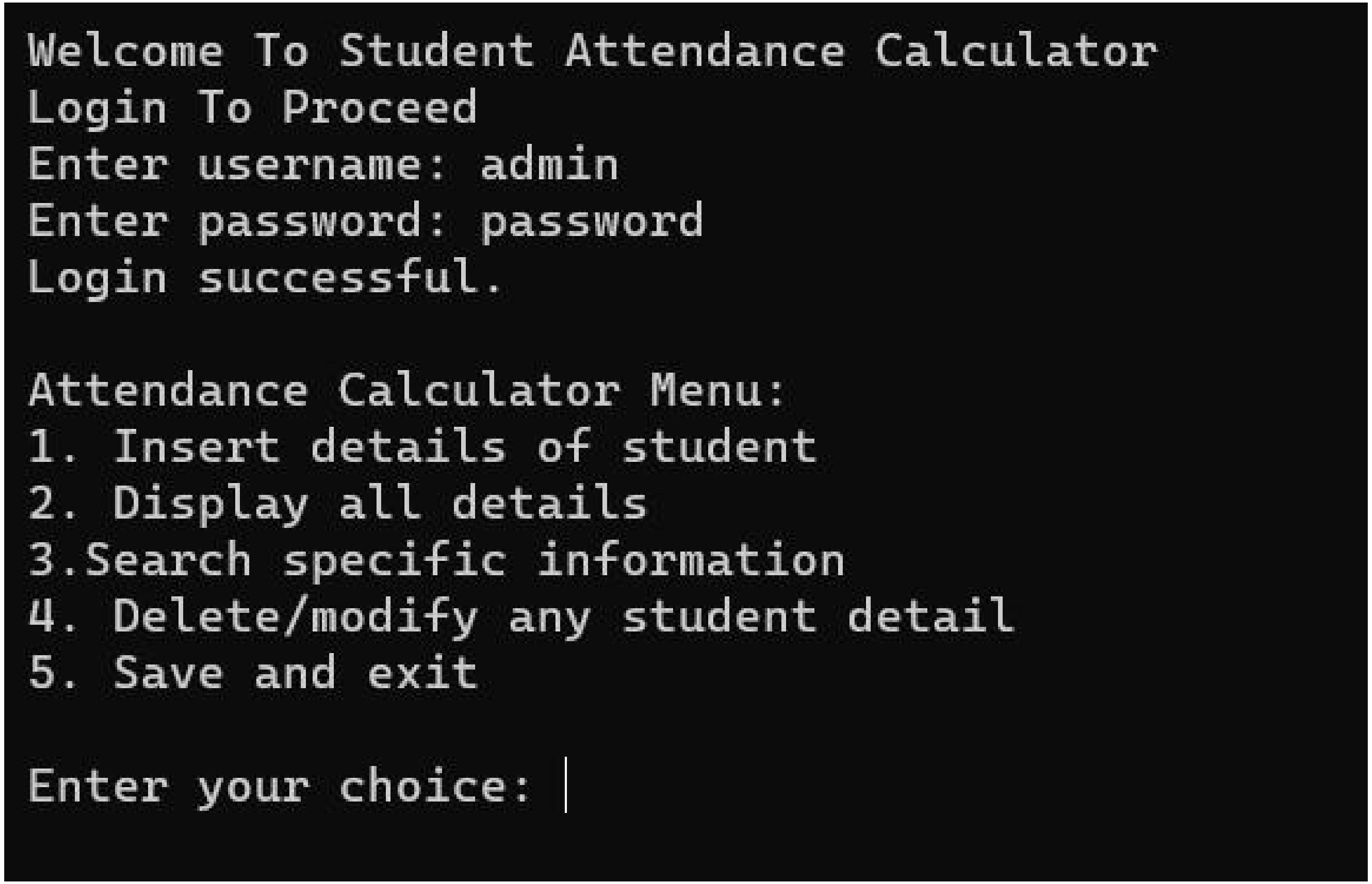
# Output Snapshots

## Login Function



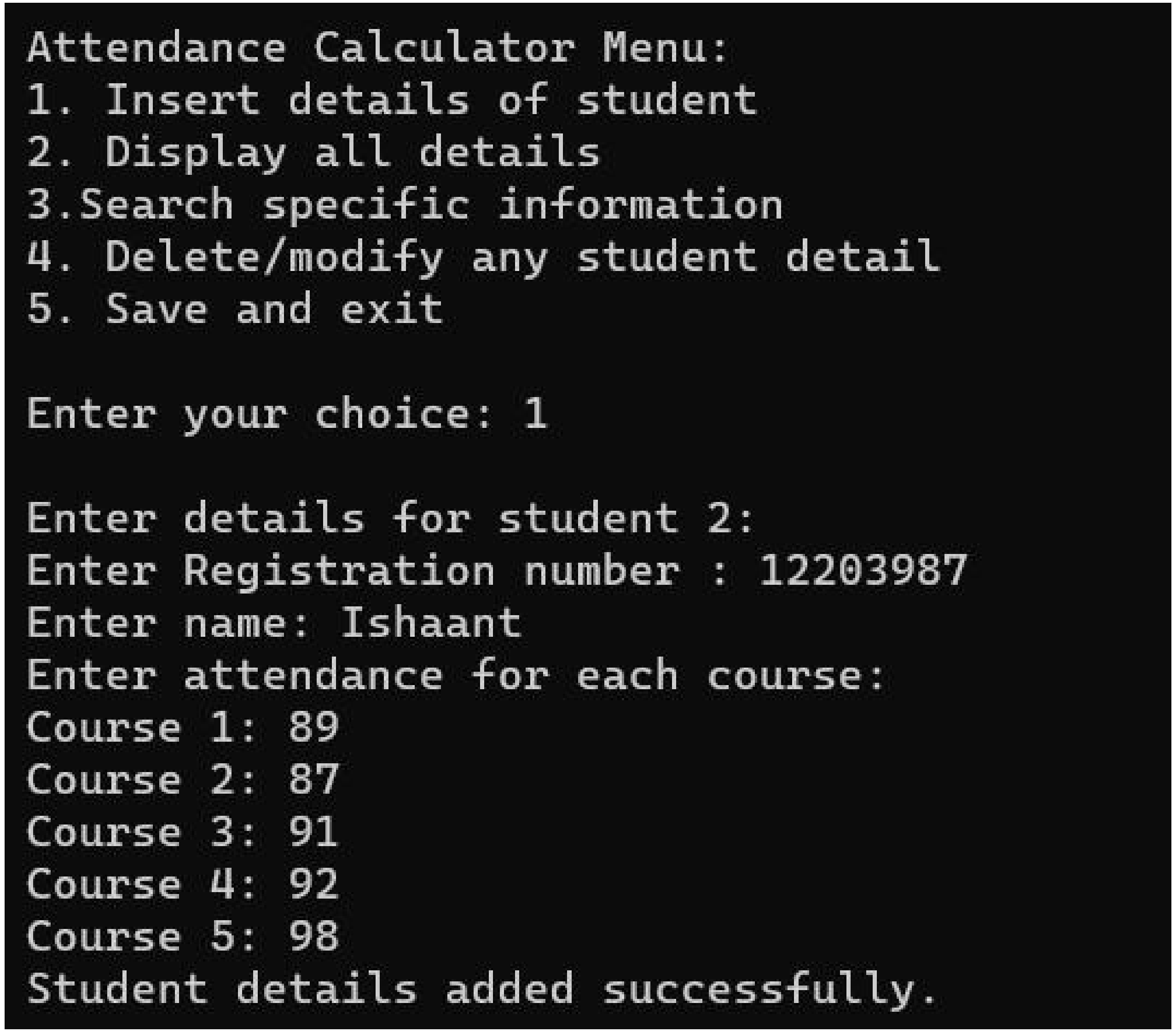
```
Welcome To Student Attendance Calculator
Login To Proceed
Enter username: admin
Enter password: password
Login successful.
```

# After Login



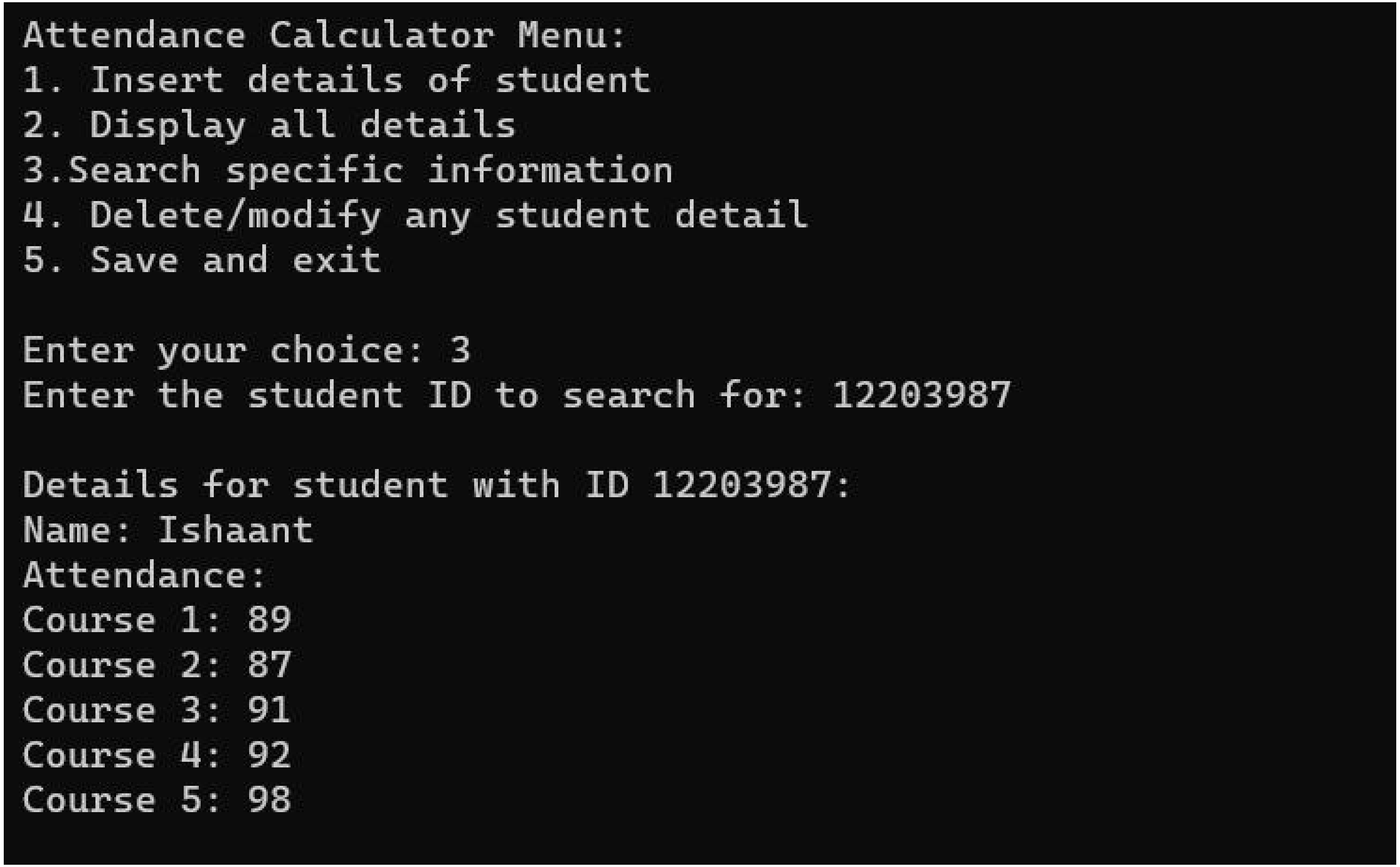
```
Welcome To Student Attendance Calculator
Login To Proceed
Enter username: admin
Enter password: password
Login successful.

Attendance Calculator Menu:
1. Insert details of student
2. Display all details
3.Search specific information
4. Delete/modify any student detail
5. Save and exit

Enter your choice: 
```

# Insert Details Of Student



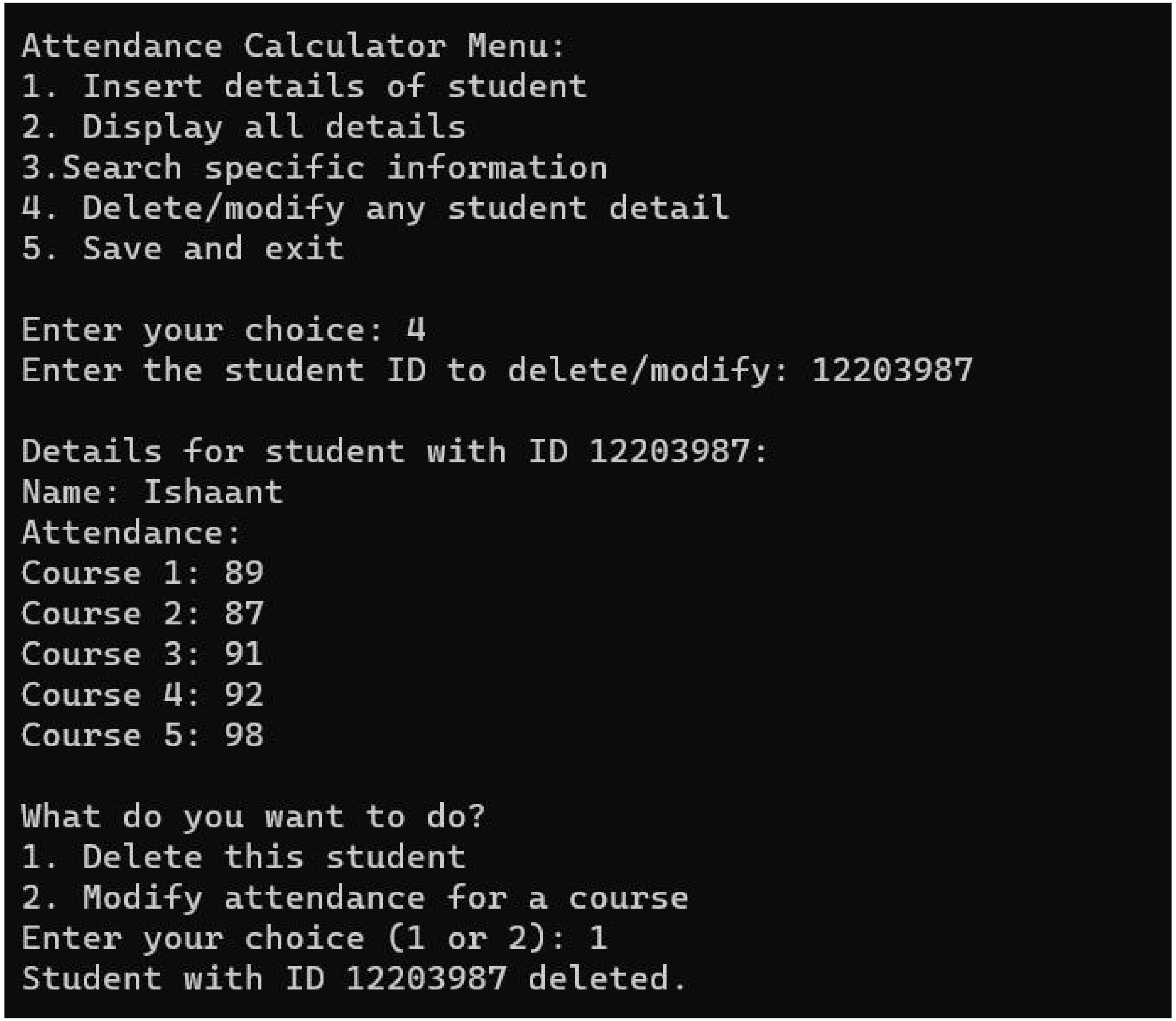
```
Attendance Calculator Menu:
1. Insert details of student
2. Display all details
3.Search specific information
4. Delete/modify any student detail
5. Save and exit

Enter your choice: 1

Enter details for student 2:
Enter Registration number : 12203987
Enter name: Ishaant
Enter attendance for each course:
Course 1: 89
Course 2: 87
Course 3: 91
Course 4: 92
Course 5: 98
Student details added successfully.
```

# Search Specific Information



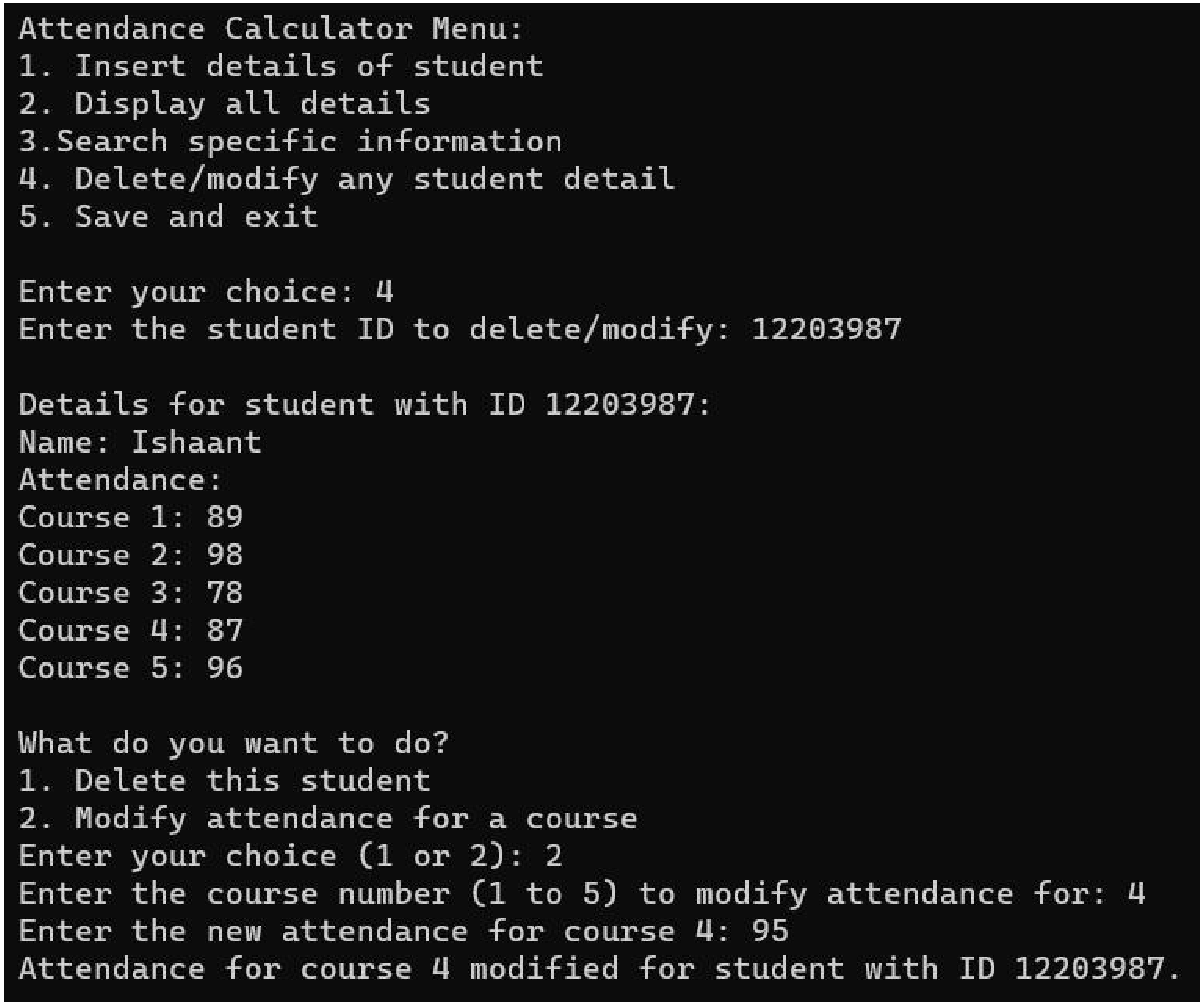
```
Attendance Calculator Menu:
1. Insert details of student
2. Display all details
3.Search specific information
4. Delete/modify any student detail
5. Save and exit

Enter your choice: 3
Enter the student ID to search for: 12203987

Details for student with ID 12203987:
Name: Ishaant
Attendance:
Course 1: 89
Course 2: 87
Course 3: 91
Course 4: 92
Course 5: 98
```

# Delete Any Student Detail



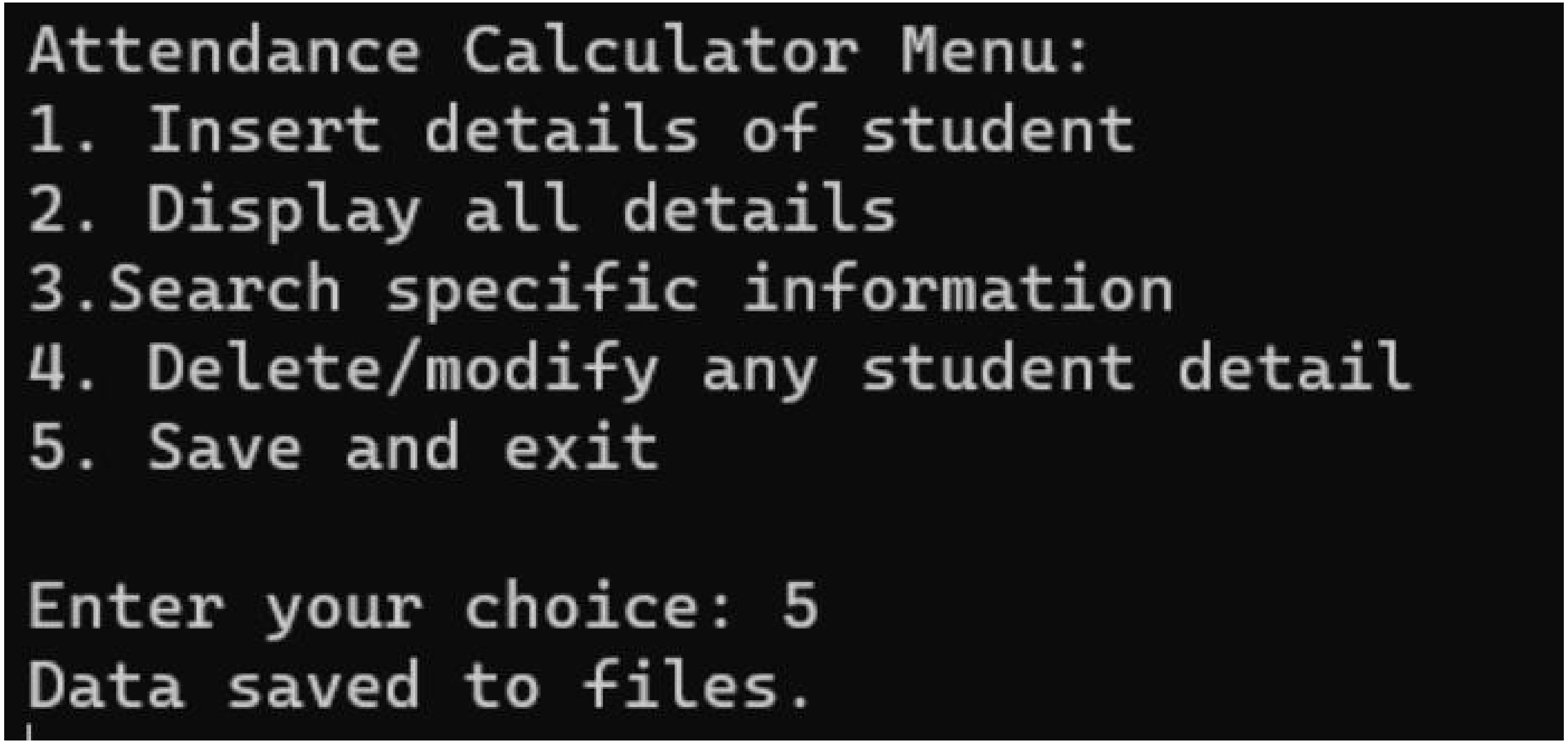
```
Attendance Calculator Menu:
1. Insert details of student
2. Display all details
3.Search specific information
4. Delete/modify any student detail
5. Save and exit

Enter your choice: 4
Enter the student ID to delete/modify: 12203987

Details for student with ID 12203987:
Name: Ishaant
Attendance:
Course 1: 89
Course 2: 87
Course 3: 91
Course 4: 92
Course 5: 98

What do you want to do?
1. Delete this student
2. Modify attendance for a course
Enter your choice (1 or 2): 1
Student with ID 12203987 deleted.
```

# Modify Any Student Detail



```
Attendance Calculator Menu:
1. Insert details of student
2. Display all details
3.Search specific information
4. Delete/modify any student detail
5. Save and exit

Enter your choice: 4
Enter the student ID to delete/modify: 12203987

Details for student with ID 12203987:
Name: Ishaant
Attendance:
Course 1: 89
Course 2: 98
Course 3: 78
Course 4: 87
Course 5: 96

What do you want to do?
1. Delete this student
2. Modify attendance for a course
Enter your choice (1 or 2): 2
Enter the course number (1 to 5) to modify attendance for: 4
Enter the new attendance for course 4: 95
Attendance for course 4 modified for student with ID 12203987.
```

# Save Data



```
Attendance Calculator Menu:
1. Insert details of student
2. Display all details
3.Search specific information
4. Delete/modify any student detail
5. Save and exit

Enter your choice: 5
Data saved to files.
```