Data Structure & Algorithm Lab (PCC-IT391)

Lab Sessional Report Submitted to

Maulana Abul Kalam Azad University of Technology, West Bengal

for

B. Tech

in

Department of Information Technology

Submitted by

Suman Tewary
Somnath Das
G Sharan Sai
Probhakar Thapa

Course Faculty
Mrs. Sayantani Saha

Department of Information Technology

Maulana Abul Kalam Azad University of Technology, West Bengal
Simhat, Haringhata, Nadia
Pin-741249

Self Certificate

This is to certify that the dissertation/project proposal entitled "Student Management System" is done by Suman Tewary, Probhakar Thapa, G Sharan Sai , Somnath Das is an authentic work carried out for the partial fulfilment of the requirements for the award of Mrs.Sayantani Saha. The matter embodied in this project work has not been submitted earlier for award of any degree to the best of my knowledge and belief.

Name of the Students:

Name	RegNo	Rollno
Suman Tewary	221000120125	10000222046
Somnath Das	221000120122	10000222043
G Sharan Sai	221000120124	10000222045
Probhakar Thapa	221000120137	10000222054

❖ Introduction:

The student database management system project is a program developed using the C programming language that allows users to manage a database of student records. The program allows users to perform various operations on the student records, such as adding new students, editing existing students, deleting students, and searching for students.

The program uses a struct to store the student information, which includes fields for the student's name, ID, and grades. An array of structs is used to store all of the student records. The program uses a command-line interface for user interaction, where users can enter commands to perform different operations on the student records.

The program has several functions that perform specific tasks. For example, the addStudent function allows users to add a new student record to the database, while the editStudent function allows users to make changes to an existing student record. The delStudent function allows users to delete a student record from the database, and the searchStudent function allows users to search for a specific student by name or ID. The view function allows users to view all student records in the database.

The program uses a text file to store the student records. The file is created and populated with data when the program is first run. The file is updated with any changes made to the student records, such as adding, editing, or deleting students.

It is important to note that this is a simple demonstration project and should not be used to store sensitive information in a real-world setting. Additionally, the program does not include any security measures to protect the student records. For real-world use, it is essential to implement a login system and backup the student records.

Overall, the student database management system project is a good example of how to use C programming to create a program that can manage student records. It provides a solid foundation for developing more advanced and secure student database management systems.

❖ All About The Functions:

The student database management system project includes several functions that perform specific tasks on the student records. These functions are:

addStudent: This function allows users to add a new student record to the database. It takes in the student's name, age, mobile number, and grades as input and adds it to the array of student records.

editStudent: This function allows users to edit an existing student record. It takes in the student's ID as input, searches for the corresponding student in the database, and then allows the user to update the student's name, ID, and grades.

deleteStudent: This function allows users to delete a student record from the database. It takes in the student's ID as input, searches for the corresponding student in the database, and then deletes the student record from the array.

searchStudent: This function allows users to search for a specific student by name or ID. It takes in the search criteria as input and searches the array of student records for a match. If a match is found, the student's information is displayed, otherwise, it shows that no record is found.

view: This function allows users to view all student records in the database. It loops through the array of student records and displays the information for each student.

All the above functions are designed to interact with the struct and file handling to perform the respective operations. These functions are essential components of the student database management system project and make it easy for users to perform various operations on the student records.

❖ Datastructure used in the project:

The student database management system project uses a struct to store the student information. A struct is a user-defined data type that can be used to group together different types of data into a single entity. In this project, the struct is used to store the student's name, ID, and grades.

The struct used in this project might look something like this:

```
struct ad
{
    char name[30];
    char department[10];
    int dsaLab, itLab, deLab, phone, age;
}
```

In this example, the struct 'student' has 7 fields:

name: a character array of length 30 that stores the student's name
department: character array of length 10 that stores the student's depentment
dsaLab,itLab,deLab: are integer that stores the student's grades

An array of structs is used to store all student records. This array is used to store all the student records that are entered into the system. The program uses this array to perform various operations on the student records, such as adding, editing, and deleting students.

The struct and array of structs are used to store and manipulate the student records in this project. They provide a way to group related data together and make it easy to perform various operations on the student records.

❖ System Requirements:

The system requirements for the student database management system project are:

C compiler: The program is written in C programming language and requires a C compiler to be installed on the computer in order to be compiled and executed. Popular C compilers include GCC (GNU Compiler Collection) and Microsoft Visual C++.

Operating System: The program is developed to run on Windows or Linux operating systems, so it is necessary to have one of this OS to run the program.

Memory and Storage: The program does not require a significant amount of memory or storage. However, it does require enough memory to store the student records and the text file used to store the data.

Text editor: The program should be written in a text editor like notepad, notepad++ or sublime.

The program should be executed in the command line or terminal, so it would be necessary to have one.

By meeting these system requirements, the student database management system project should be able to run smoothly on the computer. However, it's important to keep in mind that this is a simple demonstration project and may not have the same system requirements as a more advanced and feature-rich program.

❖ Problems in the project:

As with any software development project, there may be some problems or limitations in the student database management system project. Some of the potential problems include:

Security: The project does not include any security measures to protect the student records. This means that anyone with access to the program can view, edit, and delete student records. In a real-world setting, it would be important to implement a login system and encrypt the student records to protect the sensitive information.

Scalability: The program is designed to store and manage a limited number of student records. If the program needs to handle a large number of student records, the program may not be efficient and may need to be optimized for performance.

File handling: The program uses a text file to store the student records. This method of storing data is not suitable for large number of data and may cause some problems when the program is used in a real-world setting. The program should use a database system such as MySQL or PostgreSQL to store the data in production.

User interface: The program uses a command-line interface for user interaction. While this is suitable for a simple program such as this, it may not be suitable for more complex programs or for users who are not comfortable using the command line. A graphical user interface would be more user-friendly and intuitive.

Error handling: The program includes error handling for invalid user input and file operations. However, it may not handle all possible errors and may not provide enough information for the user to understand what went wrong.

These are some of the potential problems that may be encountered in the student database management system project. These problems can be resolved by implementing proper security measures, optimizing performance, using a more robust data storage solution, improving the user interface, and providing more detailed error messages.

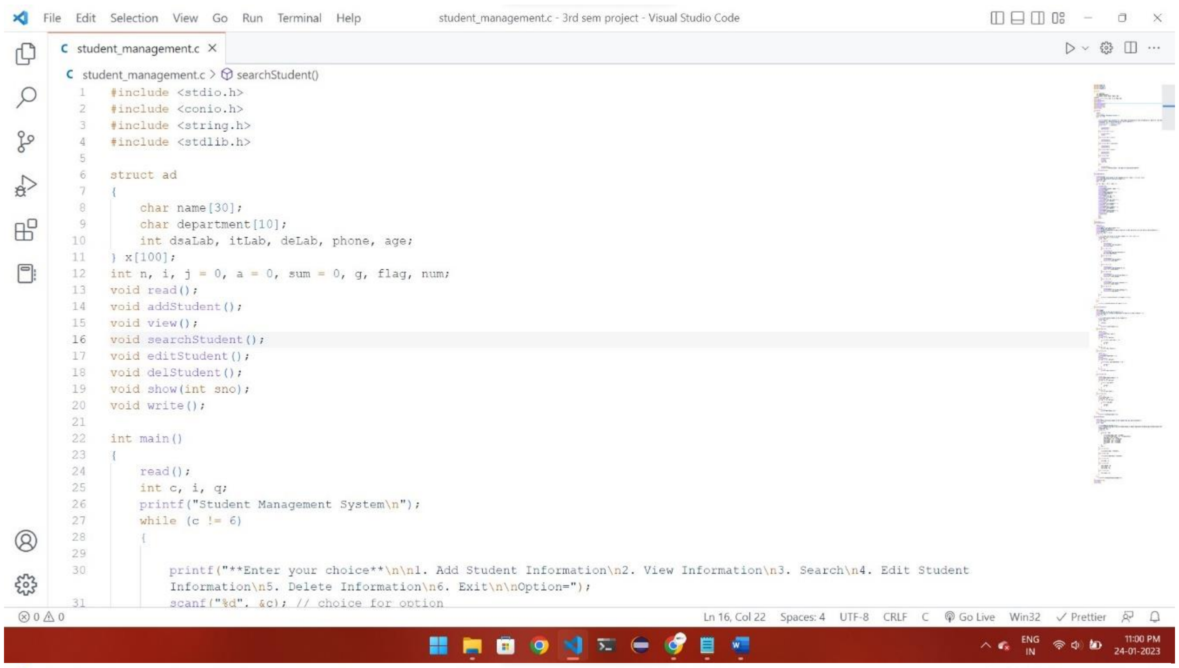
❖ Future enhancements:

- Implement a login system to restrict access to the database
- Add the ability to export student records to a CSV file
- Allow for sorting of student records by different fields (name, ID, grade)
- Add a graphical user interface for a more user-friendly experience
- Implement a way to backup the student records for safety purpose.

❖ Conclusion:

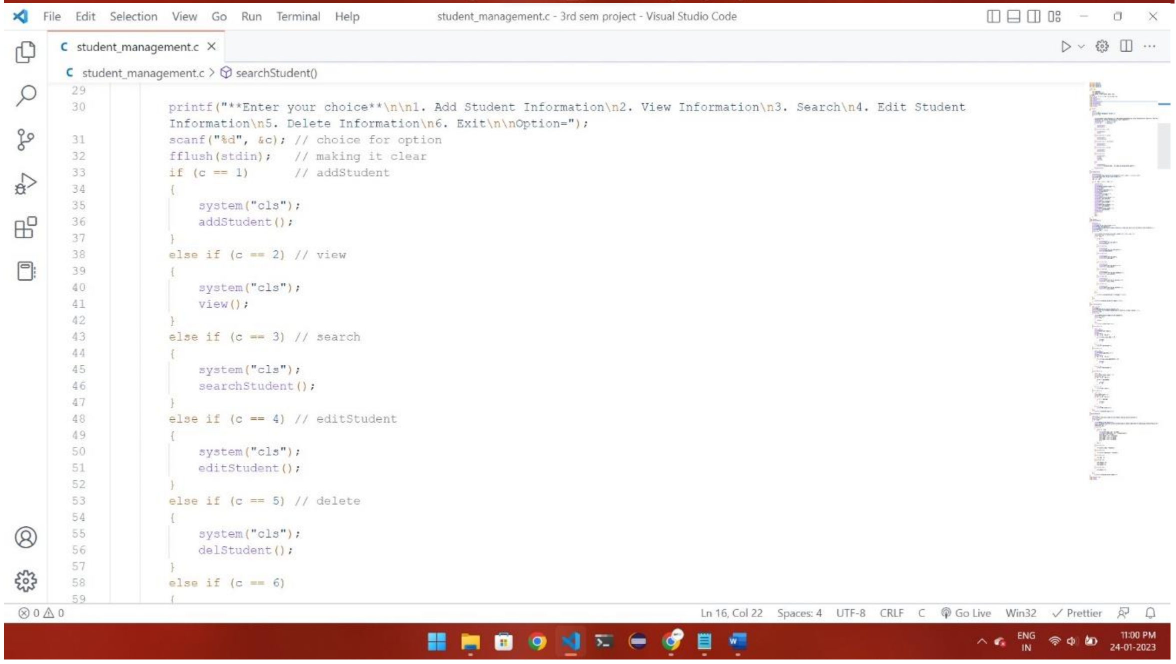
This project provides a basic student database management system developed using C programming language. It demonstrates the ability to create a program that can add, edit, delete, search and display student records using struct and file handling. However, it should be noted that the program is not a secure or production-ready system and should not be used in any real-world setting without proper security measures.

❖ Snapshots:



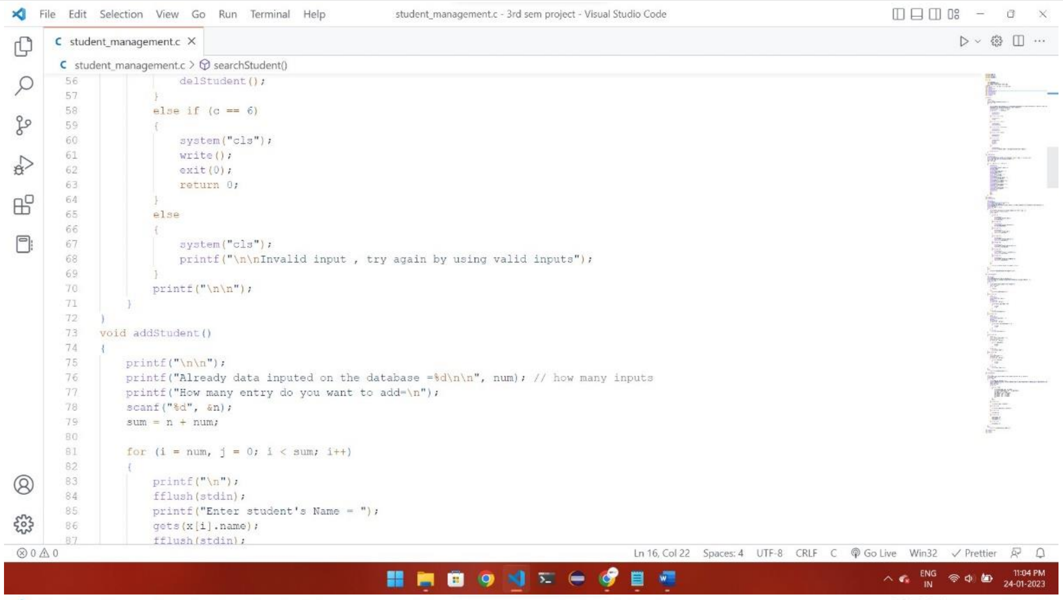
student_management.c - 3rd sem project - Visual Studio Code

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

struct ad
{
    char name[30];
    char department[10];
    int dsaLab, itLab, deLab, phone, age;
} x[100];
int n, i, j = 0, a = 0, sum = 0, g, flag, num;
void read();
void addStudent();
void view();
void searchStudent();
void editStudent();
void delStudent();
void show(int sno);
void write();

int main()
{
    read();
    int c, i, q;
    printf("Student Management System\n");
    while (c != 6)
    {

        printf("**Enter your choice**\n\n1. Add Student Information\n2. View Information\n3. Search\n4. Edit Student
        Information\n5. Delete Information\n6. Exit\n\nOption=");
        scanf("%d", &c); // choice for option
```

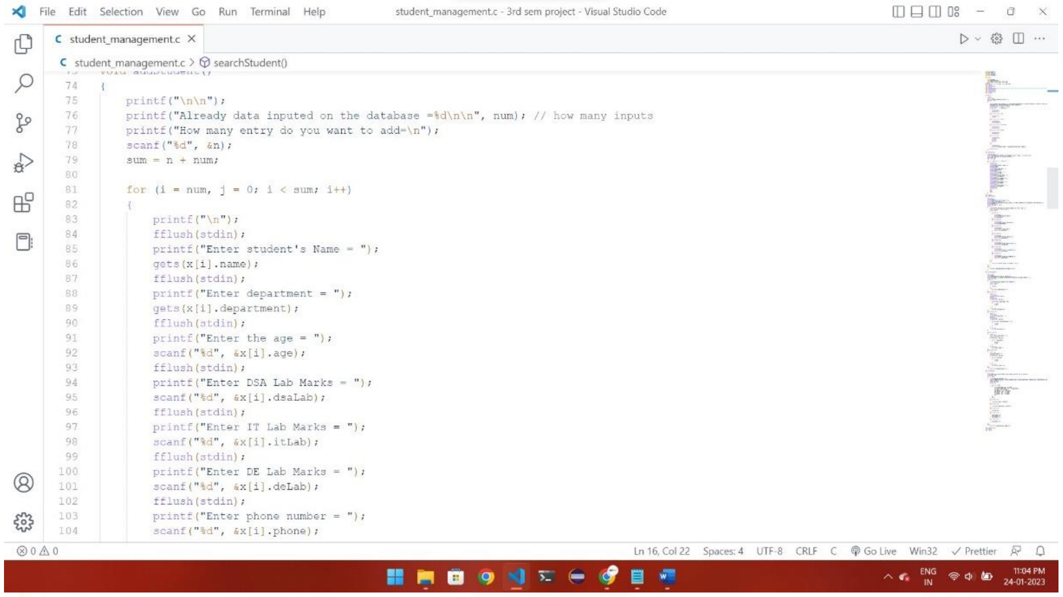


student_management.c - 3rd sem project - Visual Studio Code

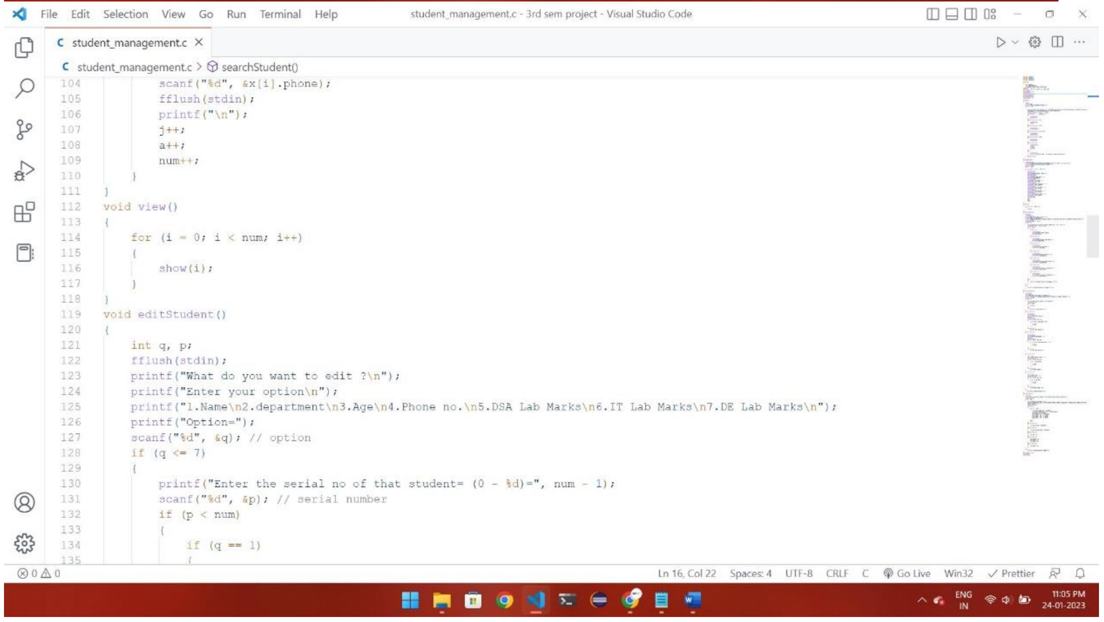
```c

        printf("**Enter your choice**\n\n1. Add Student Information\n2. View Information\n3. Search\n4. Edit Student
        Information\n5. Delete Information\n6. Exit\n\nOption=");
        scanf("%d", &c); // choice for option
        fflush(stdin);   // making it clear
        if (c == 1)      // addStudent
        {
            system("cls");
            addStudent();
        }
        else if (c == 2) // view
        {
            system("cls");
            view();
        }
        else if (c == 3) // search
        {
            system("cls");
            searchStudent();
        }
        else if (c == 4) // editStudent
        {
            system("cls");
            editStudent();
        }
        else if (c == 5) // delete
        {
            system("cls");
            delStudent();
        }
        else if (c == 6)
        {
```


File Edit Selection View Go Run Terminal Help
student_management.c - 3rd sem project - Visual Studio Code
student_management.c
student_management.c > searchStudent()
Ln 16, Col 22 Spaces: 4 UTF-8 CRLF C Go Live Win32 ✓ Prettier
11:04 PM 24-01-2023

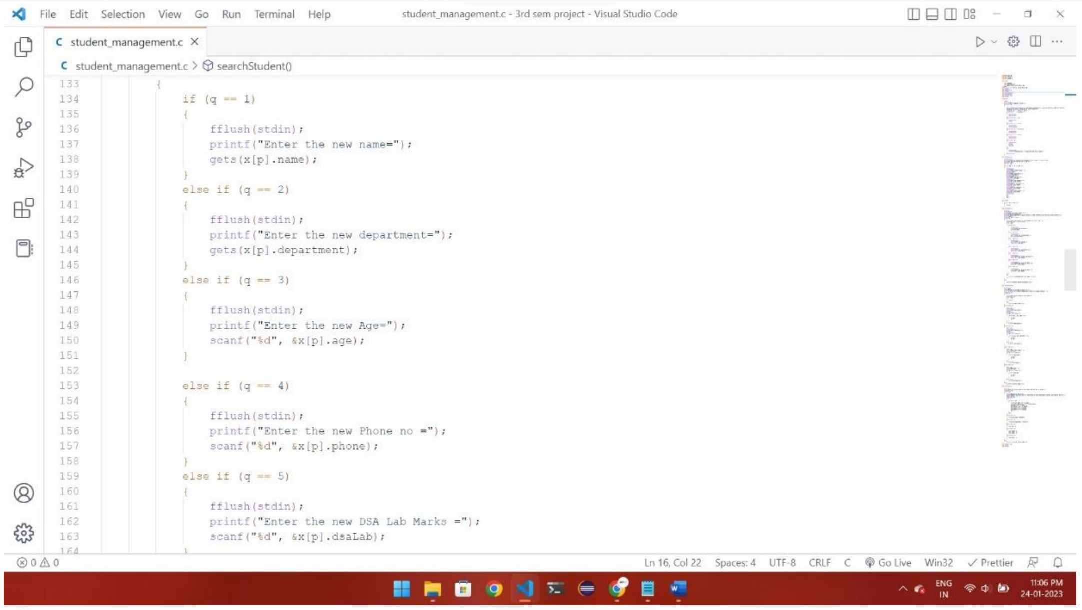
```c
            delStudent();
        }
        else if (c == 6)
        {
            system("cls");
            write();
            exit(0);
            return 0;
        }
        else
        {
            system("cls");
            printf("\n\nInvalid input , try again by using valid inputs");
        }
        printf("\n\n");
    }
}
void addStudent()
{
    printf("\n\n");
    printf("Already data inputed on the database =%d\n\n", num); // how many inputs
    printf("How many entry do you want to add=\n");
    scanf("%d", &n);
    sum = n + num;

    for (i = num, j = 0; i < sum; i++)
    {
        printf("\n");
        fflush(stdin);
        printf("Enter student's Name = ");
        gets(x[i].name);
        fflush(stdin);
```


File Edit Selection View Go Run Terminal Help
student_management.c - 3rd sem project - Visual Studio Code
student_management.c
student_management.c > searchStudent()
Ln 16, Col 22 Spaces: 4 UTF-8 CRLF C Go Live Win32 ✓ Prettier
11:04 PM 24-01-2023

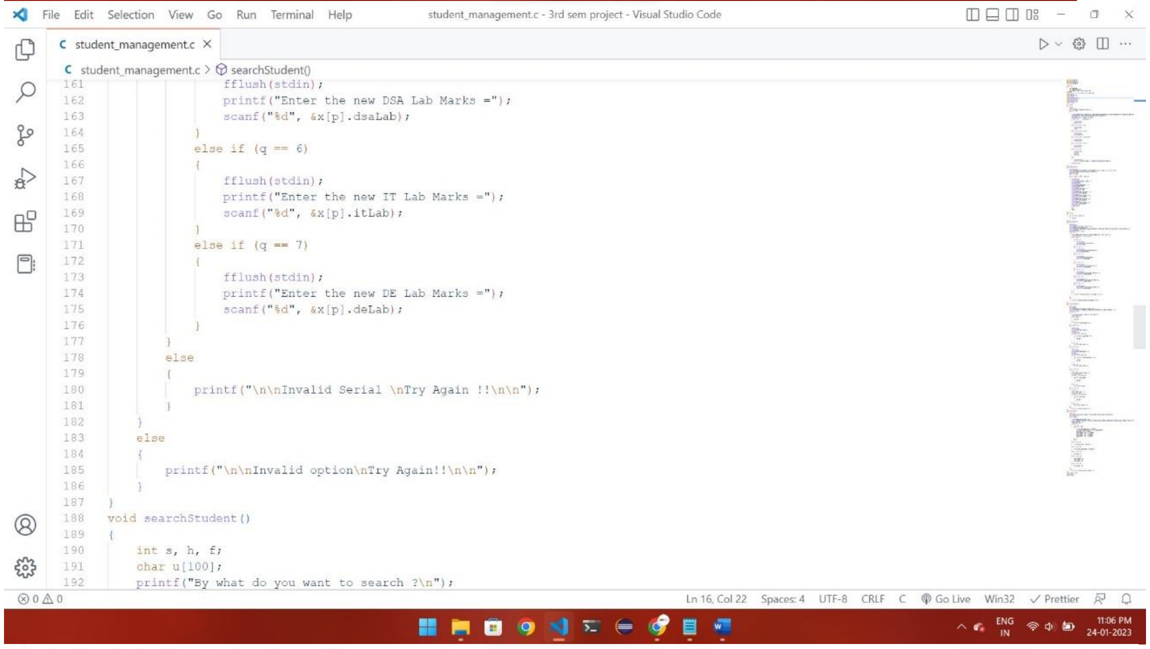
```c
{
    printf("\n\n");
    printf("Already data inputed on the database =%d\n\n", num); // how many inputs
    printf("How many entry do you want to add=\n");
    scanf("%d", &n);
    sum = n + num;

    for (i = num, j = 0; i < sum; i++)
    {
        printf("\n");
        fflush(stdin);
        printf("Enter student's Name = ");
        gets(x[i].name);
        fflush(stdin);
        printf("Enter department = ");
        gets(x[i].department);
        fflush(stdin);
        printf("Enter the age = ");
        scanf("%d", &x[i].age);
        fflush(stdin);
        printf("Enter DSA Lab Marks = ");
        scanf("%d", &x[i].dsaLab);
        fflush(stdin);
        printf("Enter IT Lab Marks = ");
        scanf("%d", &x[i].itLab);
        fflush(stdin);
        printf("Enter DE Lab Marks = ");
        scanf("%d", &x[i].deLab);
        fflush(stdin);
        printf("Enter phone number = ");
        scanf("%d", &x[i].phone);
```


File Edit Selection View Go Run Terminal Help
student_management.c - 3rd sem project - Visual Studio Code
student_management.c
student_management.c > searchStudent()
Ln 16, Col 22 Spaces: 4 UTF-8 CRLF C Go Live Win32 ✓ Prettier
11:05 PM 24-01-2023

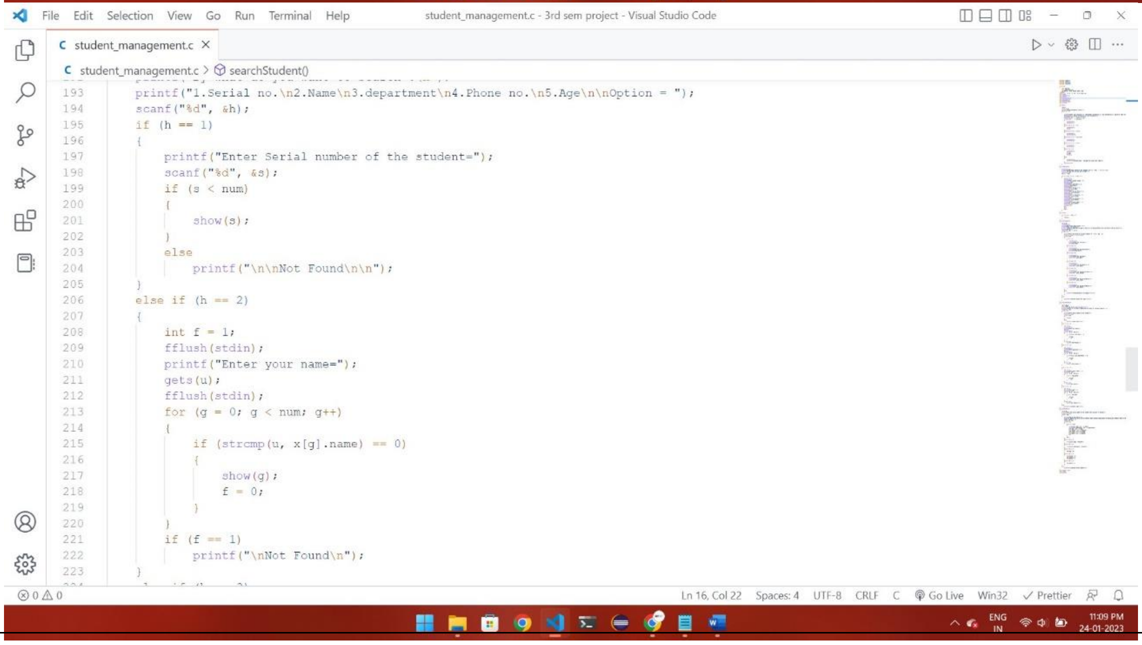
```c
        scanf("%d", &x[i].phone);
        fflush(stdin);
        printf("\n");
        j++;
        a++;
        num++;
    }
}
void view()
{
    for (i = 0; i < num; i++)
    {
        show(i);
    }
}
void editStudent()
{
    int q, p;
    fflush(stdin);
    printf("What do you want to edit ?\n");
    printf("Enter your option\n");
    printf("1.Name\n2.department\n3.Age\n4.Phone no.\n5.DSA Lab Marks\n6.IT Lab Marks\n7.DE Lab Marks\n");
    printf("Option=");
    scanf("%d", &q); // option
    if (q <= 7)
    {
        printf("Enter the serial no of that student= (0 - %d)=", num - 1);
        scanf("%d", &p); // serial number
        if (p < num)
        {
            if (q == 1)
```



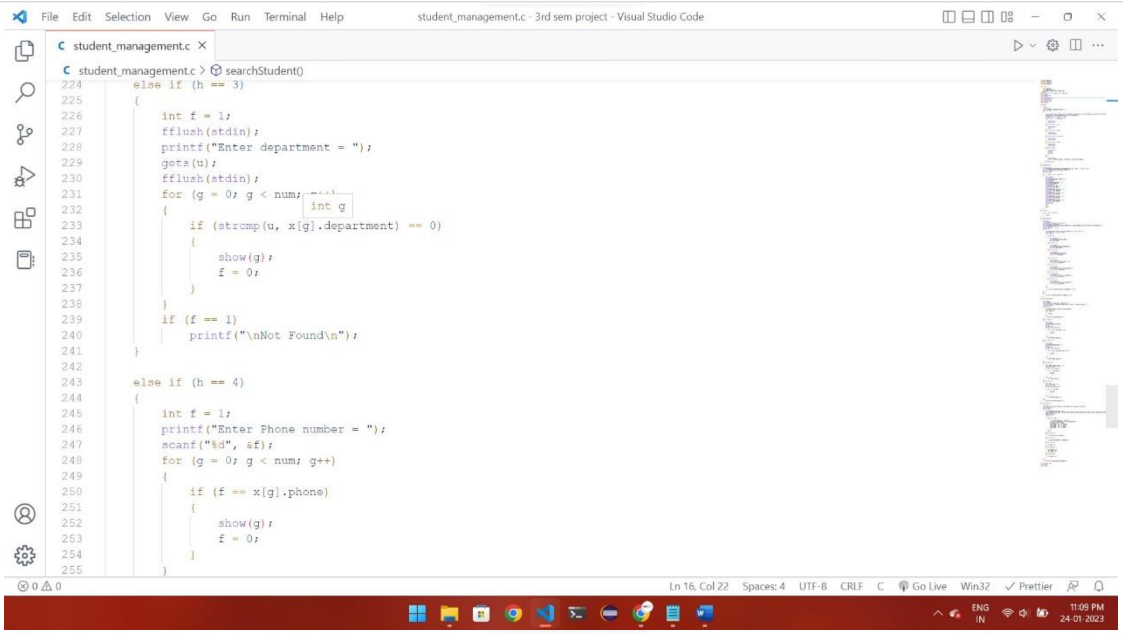
```c
        {
            if (q == 1)
            {
                fflush(stdin);
                printf("Enter the new name=");
                gets(x[p].name);
            }
            else if (q == 2)
            {
                fflush(stdin);
                printf("Enter the new department=");
                gets(x[p].department);
            }
            else if (q == 3)
            {
                fflush(stdin);
                printf("Enter the new Age=");
                scanf("%d", &x[p].age);
            }

            else if (q == 4)
            {
                fflush(stdin);
                printf("Enter the new Phone no =");
                scanf("%d", &x[p].phone);
            }
            else if (q == 5)
            {
                fflush(stdin);
                printf("Enter the new DSA Lab Marks =");
                scanf("%d", &x[p].dsaLab);
```



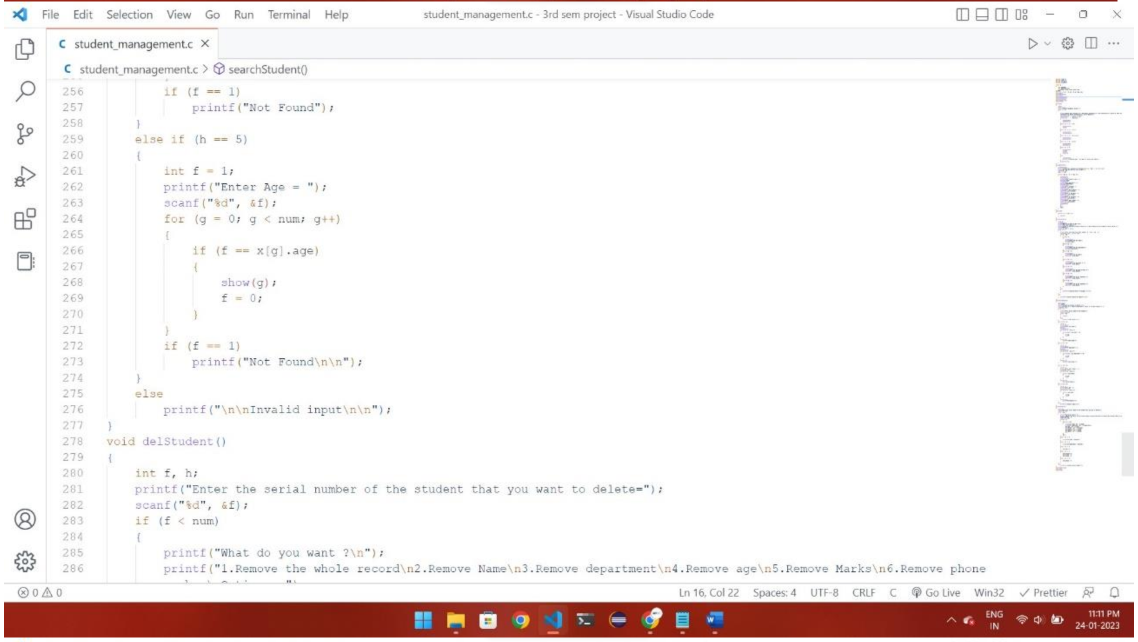
```c
                fflush(stdin);
                printf("Enter the new DSA Lab Marks =");
                scanf("%d", &x[p].dsaLab);
            }
            else if (q == 6)
            {
                fflush(stdin);
                printf("Enter the new IT Lab Marks =");
                scanf("%d", &x[p].itLab);
            }
            else if (q == 7)
            {
                fflush(stdin);
                printf("Enter the new DE Lab Marks =");
                scanf("%d", &x[p].deLab);
            }
        }
        else
        {
            printf("\n\nInvalid Serial \nTry Again !!\n\n");
        }
    }
    else
    {
        printf("\n\nInvalid option\nTry Again!!\n\n");
    }
}
void searchStudent()
{
    int s, h, f;
    char u[100];
    printf("By what do you want to search ?\n");
```



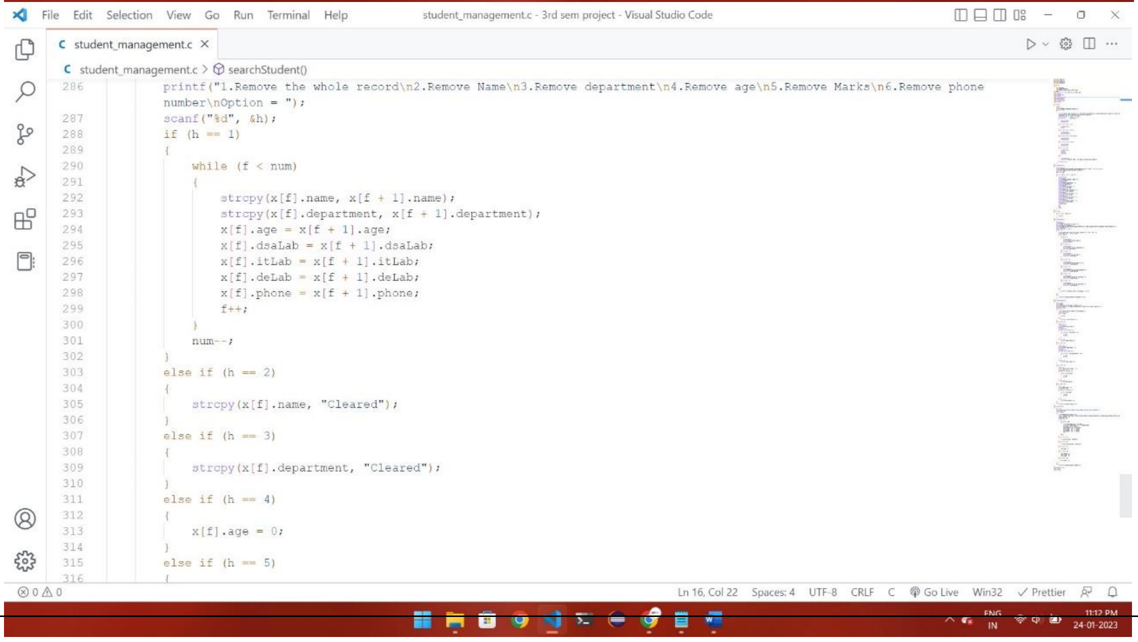
```c
    printf("1.Serial no.\n2.Name\n3.department\n4.Phone no.\n5.Age\n\nOption = ");
    scanf("%d", &h);
    if (h == 1)
    {
        printf("Enter Serial number of the student=");
        scanf("%d", &s);
        if (s < num)
        {
            show(s);
        }
        else
            printf("\n\nNot Found\n\n");
    }
    else if (h == 2)
    {
        int f = 1;
        fflush(stdin);
        printf("Enter your name=");
        gets(u);
        fflush(stdin);
        for (g = 0; g < num; g++)
        {
            if (strcmp(u, x[g].name) == 0)
            {
                show(g);
                f = 0;
            }
        }
        if (f == 1)
            printf("\nNot Found\n");
    }
```


File Edit Selection View Go Run Terminal Help
student_management.c - 3rd sem project - Visual Studio Code
student_management.c
student_management.c > searchStudent()
int g
Ln 16, Col 22 Spaces: 4 UTF-8 CRLF C Go Live Win32 Prettier
ENG IN
11:09 PM
24-01-2023

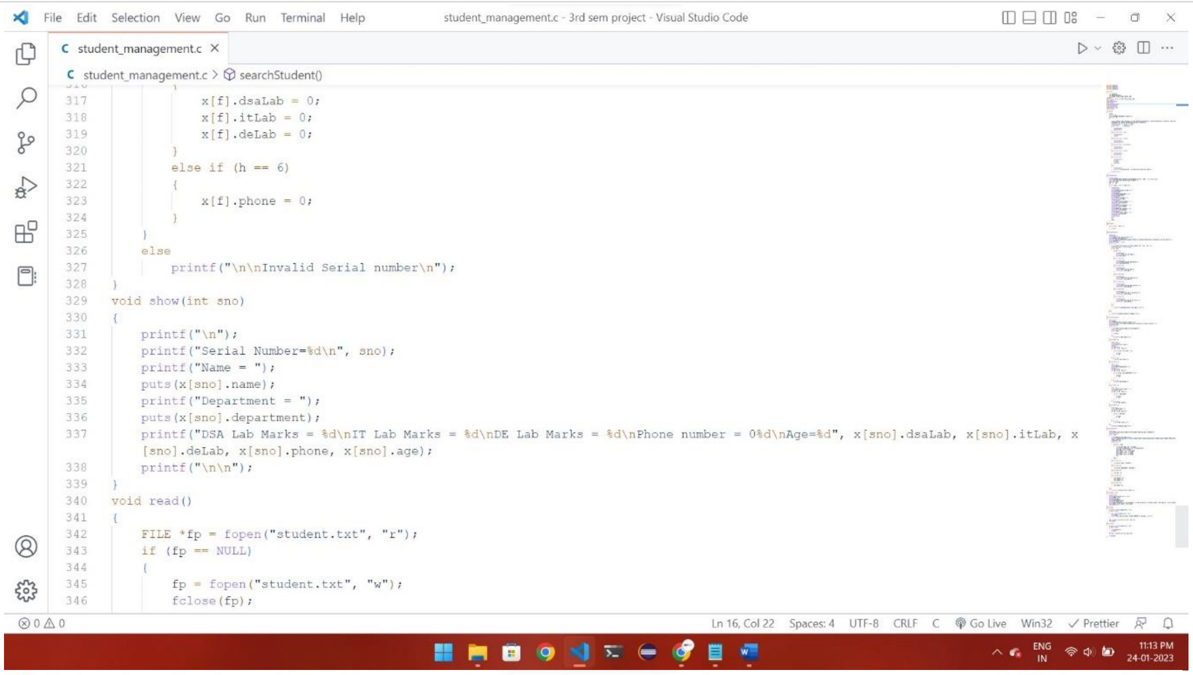
```
    else if (h == 3)
    {
        int f = 1;
        fflush(stdin);
        printf("Enter department = ");
        gets(u);
        fflush(stdin);
        for (g = 0; g < num;
        {
            if (strcmp(u, x[g].department) == 0)
            {
                show(g);
                f = 0;
            }
        }
        if (f == 1)
            printf("\nNot Found\n");
    }

    else if (h == 4)
    {
        int f = 1;
        printf("Enter Phone number = ");
        scanf("%d", &f);
        for (g = 0; g < num; g++)
        {
            if (f == x[g].phone)
            {
                show(g);
                f = 0;
            }
        }
```


File Edit Selection View Go Run Terminal Help
student_management.c - 3rd sem project - Visual Studio Code
student_management.c
student_management.c > searchStudent()
Ln 16, Col 22 Spaces: 4 UTF-8 CRLF C Go Live Win32 Prettier
ENG IN
11:11 PM
24-01-2023

```
        if (f == 1)
            printf("Not Found");
    }
    else if (h == 5)
    {
        int f = 1;
        printf("Enter Age = ");
        scanf("%d", &f);
        for (g = 0; g < num; g++)
        {
            if (f == x[g].age)
            {
                show(g);
                f = 0;
            }
        }
        if (f == 1)
            printf("Not Found\n\n");
    }
    else
        printf("\n\nInvalid input\n\n");
}
void delStudent()
{
    int f, h;
    printf("Enter the serial number of the student that you want to delete=");
    scanf("%d", &f);
    if (f < num)
    {
        printf("What do you want ?\n");
        printf("1.Remove the whole record\n2.Remove Name\n3.Remove department\n4.Remove age\n5.Remove Marks\n6.Remove phone
```

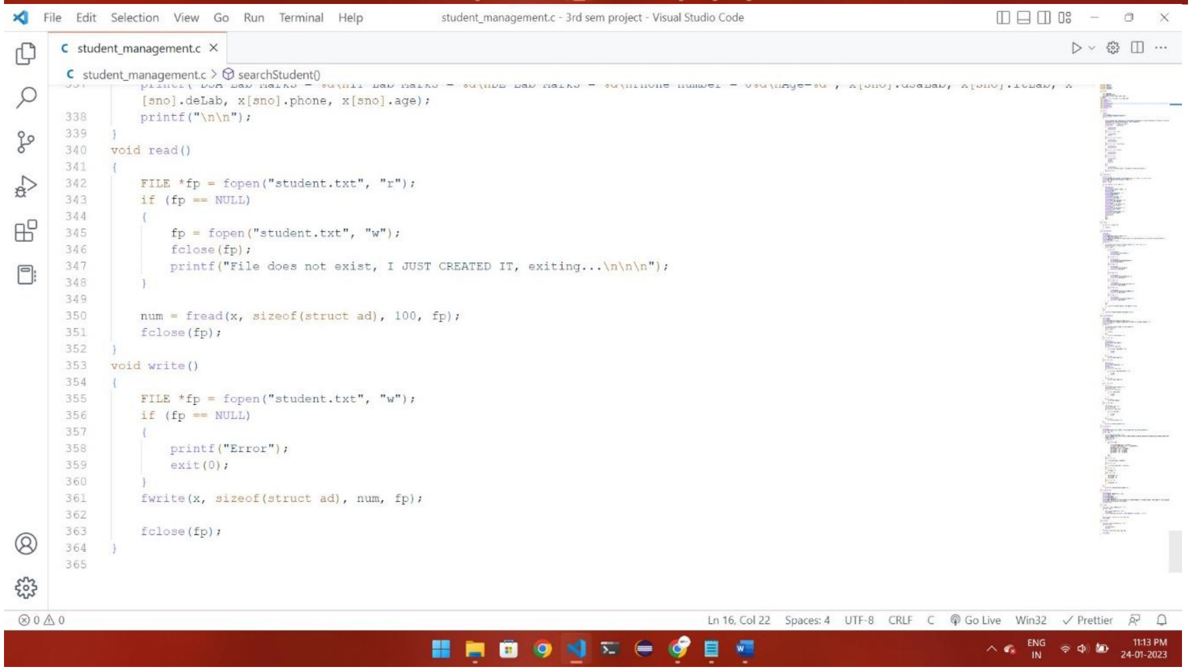

File Edit Selection View Go Run Terminal Help
student_management.c - 3rd sem project - Visual Studio Code
student_management.c
student_management.c > searchStudent()
Ln 16, Col 22 Spaces: 4 UTF-8 CRLF C Go Live Win32 Prettier
ENG IN
11:12 PM
24-01-2023

```
        printf("1.Remove the whole record\n2.Remove Name\n3.Remove department\n4.Remove age\n5.Remove Marks\n6.Remove phone
        number\nOption = ");
        scanf("%d", &h);
        if (h == 1)
        {
            while (f < num)
            {
                strcpy(x[f].name, x[f + 1].name);
                strcpy(x[f].department, x[f + 1].department);
                x[f].age = x[f + 1].age;
                x[f].dsaLab = x[f + 1].dsaLab;
                x[f].itLab = x[f + 1].itLab;
                x[f].deLab = x[f + 1].deLab;
                x[f].phone = x[f + 1].phone;
                f++;
            }
            num--;
        }
        else if (h == 2)
        {
            strcpy(x[f].name, "Cleared");
        }
        else if (h == 3)
        {
            strcpy(x[f].department, "Cleared");
        }
        else if (h == 4)
        {
            x[f].age = 0;
        }
        else if (h == 5)
        {
```



File Edit Selection View Go Run Terminal Help
student_management.c - 3rd sem project - Visual Studio Code
student_management.c
student_management.c > searchStudent()
Ln 16, Col 22 Spaces: 4 UTF-8 CRLF C Go Live Win32 ✓ Prettier
ENG IN
11:13 PM
24-01-2023

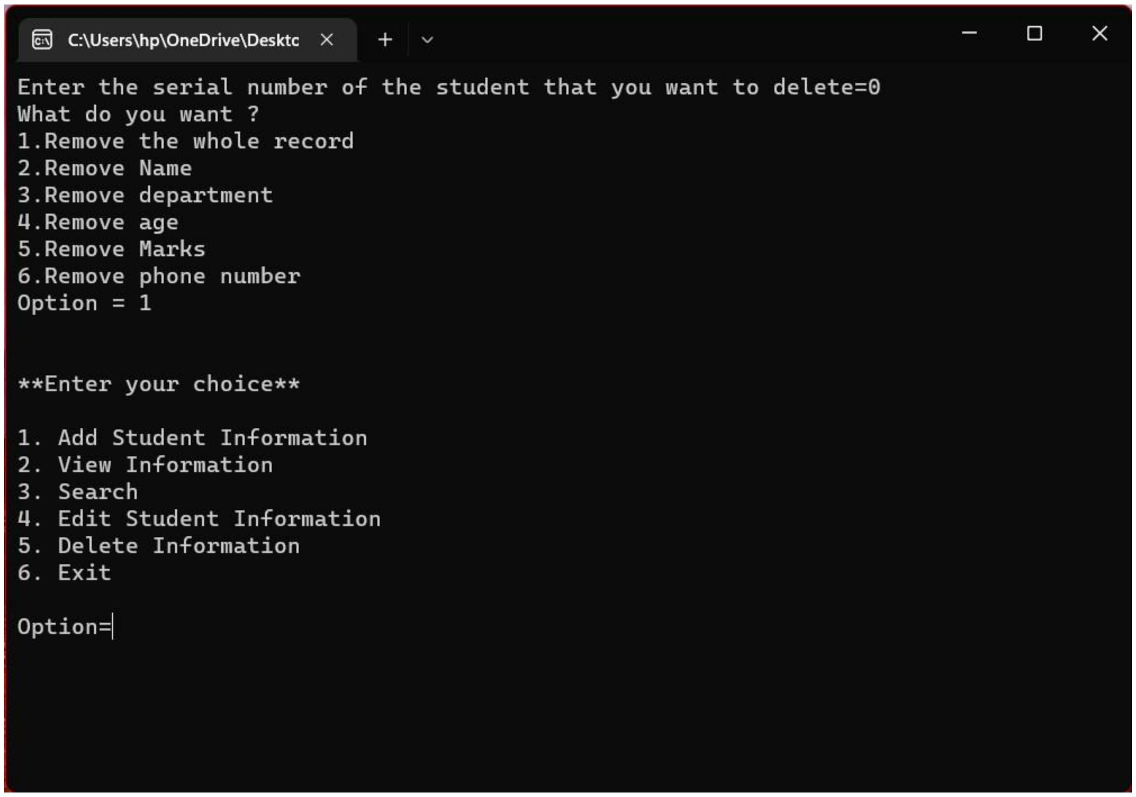
```c
            x[f].dsaLab = 0;
            x[f].itLab = 0;
            x[f].deLab = 0;
        }
        else if (h == 6)
        {
            x[f].phone = 0;
        }
    }
    else
        printf("\n\nInvalid Serial number\n");
}
void show(int sno)
{
    printf("\n");
    printf("Serial Number=%d\n", sno);
    printf("Name = ");
    puts(x[sno].name);
    printf("Department = ");
    puts(x[sno].department);
    printf("DSA Lab Marks = %d\nIT Lab Marks = %d\nDE Lab Marks = %d\nPhone number = 0%d\nAge=%d", x[sno].dsaLab, x[sno].itLab, x
    [sno].deLab, x[sno].phone, x[sno].age);
    printf("\n\n");
}
void read()
{
    FILE *fp = fopen("student.txt", "r");
    if (fp == NULL)
    {
        fp = fopen("student.txt", "w");
        fclose(fp);
```



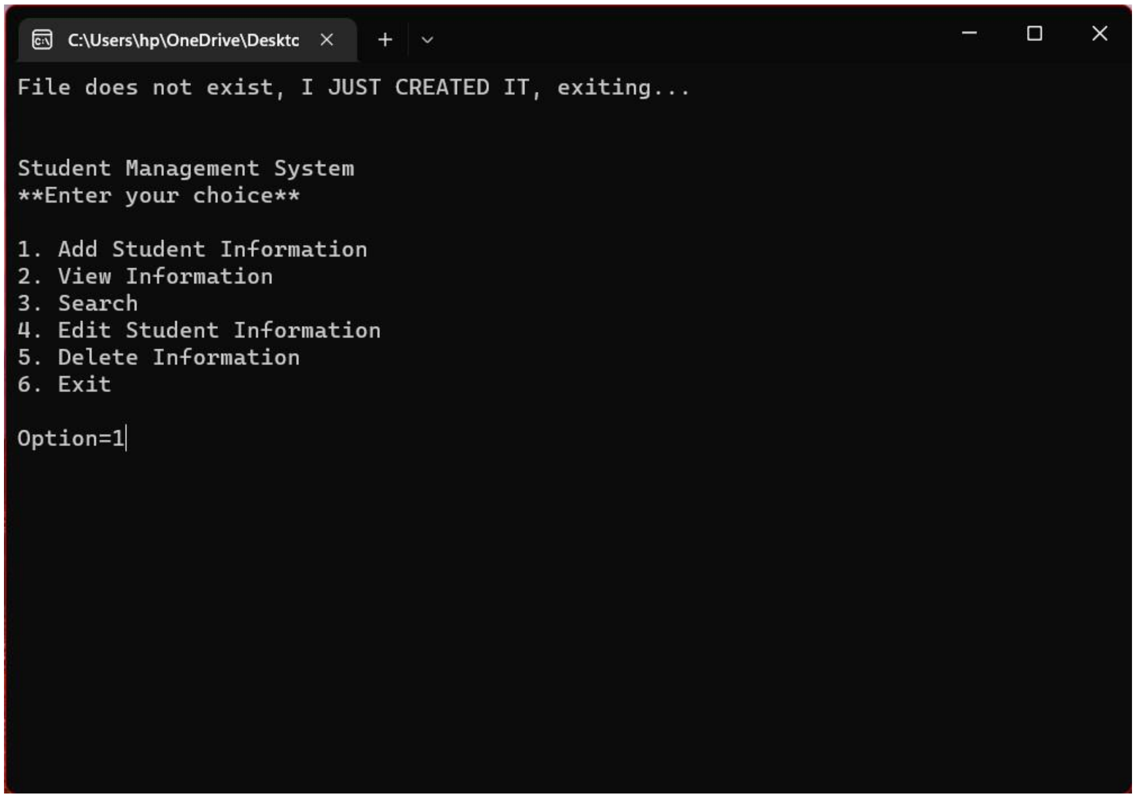
File Edit Selection View Go Run Terminal Help
student_management.c - 3rd sem project - Visual Studio Code
student_management.c
student_management.c > searchStudent()
Ln 16, Col 22 Spaces: 4 UTF-8 CRLF C Go Live Win32 ✓ Prettier
ENG IN
11:13 PM
24-01-2023

```c
    [sno].deLab, x[sno].phone, x[sno].age);
    printf("\n\n");
}
void read()
{
    FILE *fp = fopen("student.txt", "r");
    if (fp == NULL)
    {
        fp = fopen("student.txt", "w");
        fclose(fp);
        printf("File does not exist, I JUST CREATED IT, exiting...\n\n\n");
    }

    num = fread(x, sizeof(struct ad), 100, fp);
    fclose(fp);
}
void write()
{
    FILE *fp = fopen("student.txt", "w");
    if (fp == NULL)
    {
        printf("Error");
        exit(0);
    }
    fwrite(x, sizeof(struct ad), num, fp);

    fclose(fp);
}
```

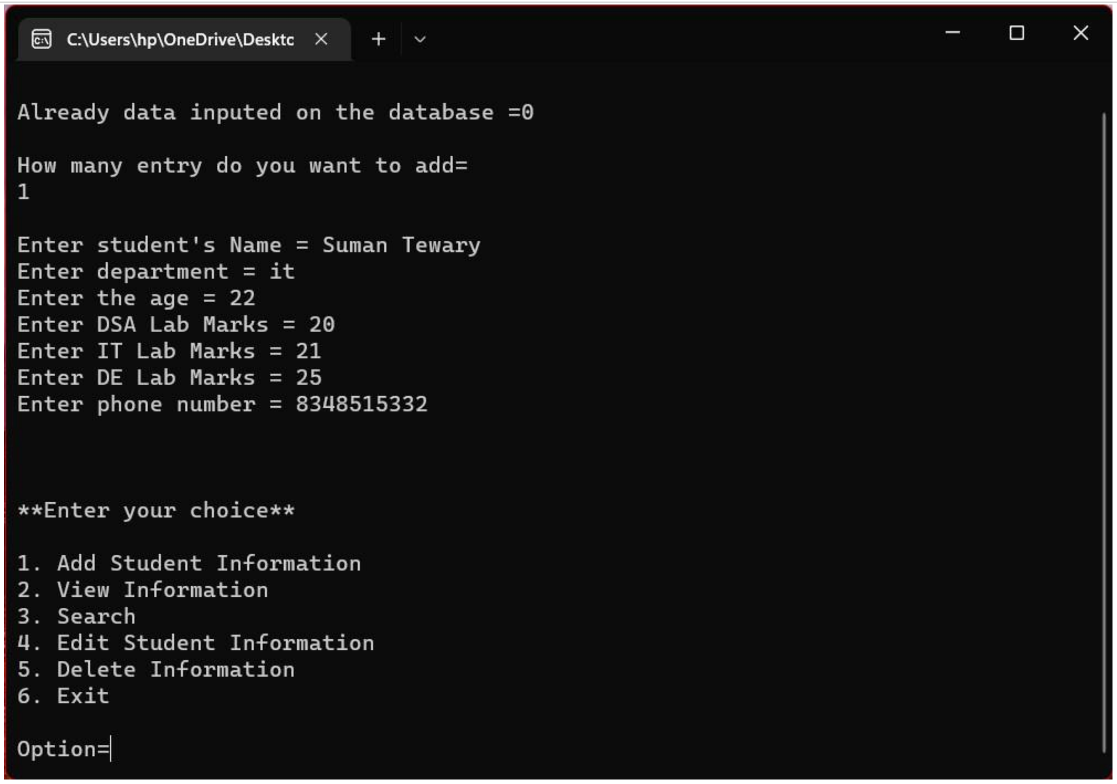
❖ Output:



```
Enter the serial number of the student that you want to delete=0
What do you want ?
1.Remove the whole record
2.Remove Name
3.Remove department
4.Remove age
5.Remove Marks
6.Remove phone number
Option = 1


**Enter your choice**

1. Add Student Information
2. View Information
3. Search
4. Edit Student Information
5. Delete Information
6. Exit

Option=
```

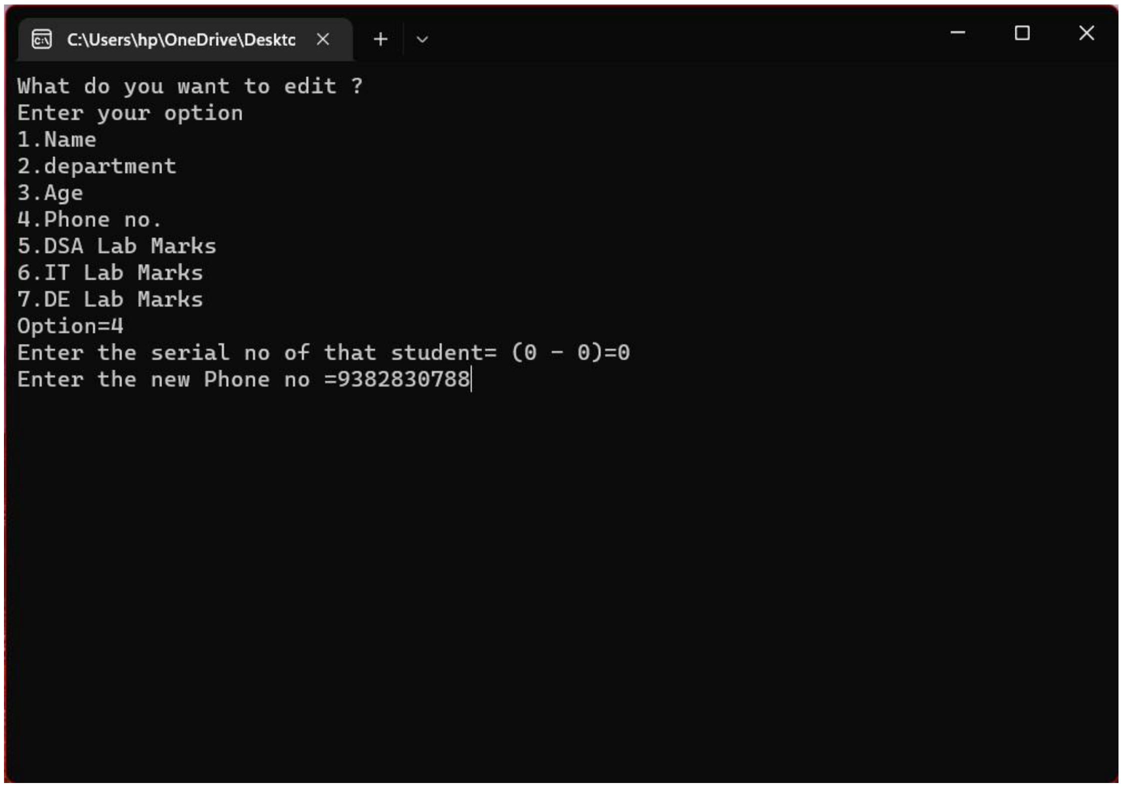


```
File does not exist, I JUST CREATED IT, exiting...


Student Management System
**Enter your choice**

1. Add Student Information
2. View Information
3. Search
4. Edit Student Information
5. Delete Information
6. Exit

Option=1
```



C:\Users\hp\OneDrive\Desktc

```
Already data inputed on the database =0

How many entry do you want to add=
1

Enter student's Name = Suman Tewary
Enter department = it
Enter the age = 22
Enter DSA Lab Marks = 20
Enter IT Lab Marks = 21
Enter DE Lab Marks = 25
Enter phone number = 8348515332



**Enter your choice**

1. Add Student Information
2. View Information
3. Search
4. Edit Student Information
5. Delete Information
6. Exit

Option=
```



C:\Users\hp\OneDrive\Desktc

```
What do you want to edit ?
Enter your option
1.Name
2.department
3.Age
4.Phone no.
5.DSA Lab Marks
6.IT Lab Marks
7.DE Lab Marks
Option=4
Enter the serial no of that student= (0 - 0)=0
Enter the new Phone no =9382830788
```