

PROJECT 2

Student Management System



Presented By
Mohammed Hassan

Presented To
Eng.Keroles Shenouda

Contact Information



GitHub: [Click Here](#)



My Drive: [Click Here](#)



My progress Page: [Click Here](#)



mohasanbder@gmail.com

Table of Contents

❖ Problem Statement.....	3
❖ Approach.....	3
❖ Student_System.h	4
1- Student Structure	5
2- Database Structure.....	5
3- Function Prototypes	6
4- Buffer Status.....	6
❖ Main.c	7
❖ Student_System.c	8
1- Student_System_init.....	8
2- Add Student Manually	9
➤ Check Student ID	10
➤ FIFO Enqueue	11
➤ Check FIFO Status	12
3- Add Student from File.....	13
4- Find Student by Roll Number.....	14
5- Find Student by First Name	15
6- Find Student by Course.....	16
7- Delete Student by ID Number.....	17
8- Update Student by id	18
9- Show All Student Information.....	19
10- Print total Student Number	20

11- Print Student information	20
❖ Program Execution Snapshots	21
➤ Add Student Manually.....	21
➤ Add Student from File.....	22
➤ Find Student Details by ID Number	23
➤ Find Student details by First Name.....	24
➤ Find Student Details by Course ID	25
➤ Print Total Number of Students.....	26
➤ Delete Student by Roll Number.....	27
➤ Update Student Details by Roll Number	28
1- If Student ID Number Not found:.....	28
2- Update Student id:.....	29
3- update Student id failed.....	30
4- update Student First Name	31
5- update Student Last Name	32
6- Update Student GPA	33
7- update Student Course ID Number	34
8- Show All Details information of Students	35
9- Exit the Program.....	36

❖ Problem Statement

Write a Program to build a simple software for Student Information Management System which can perform the following operations:

- 1. Store the First name of the student.**
- 2. Store the Last name of the student.**
- 3. Store Unique ID number for each student.**
- 4. Store GPA for each student.**
- 5. Store Courses Registered by the student.**

❖ Approach

There are 10 options to form an individual functions for every operations use it to handle Student Information.

- 1. Add the Student Details Manually.**
- 2. Add the Student Details Form File.**
- 3. Find the Student Details by given id Number of Student.**
- 4. Find the Student Details by Given Student First Name.**
- 5. Find the Student Details by Given Student Last Name.**
- 6. Count total Number of Students.**
- 7. Delete Specific Student by Given his ID Number.**
- 8. Update Student Information by Given his ID Number.**
 - a. Change Student ID Number.**
 - b. Change Student First Name.**
 - c. Change Student Last Name**
 - d. Change Student Courses.**
- 9. Print All Student Information.**
- 10. Exit The Program.**

❖ Student_System.h

```
1 /*
2 =====
3 Name      : Student_System.h
4 Author    : Mohammed Hassan
5 Created on : Oct 30, 2023
6 Description : First Term Project 2
7 =====
8 */
9
10 #ifndef STUDENT_SYS_H_
11 #define STUDENT_SYS_H_
12
13 #include <stdio.h>
14 #include <stdint.h>
15 #include <string.h>
16 // =====> printf <=====
17 #define print(...) \
18     fflush(stdin);fflush(stdout); \
19     printf(_VA_ARGS__); \
20     fflush(stdin);fflush(stdout);
21
22
23 // =====> Macros <=====
24 #define Name_Length    20
25 #define Number_Courses 5
26 #define Max_Number_Students 50
27
28 // =====> Student information <=====
29 struct Studen_info_t
30 {
31     char first_name[Name_Length];
32     char last_name[Name_Length];
33     int roll ;
34     float GPA ;
35     int Course_id[Number_Courses];
36 };
37 typedef struct Studen_info_t SStudent_Node ;
38
39 // =====> Type Definition <=====
40 typedef struct
41 {
42     SStudent_Node* base ;
43     SStudent_Node* head ;
44     SStudent_Node* tail ;
45     uint32_t length ;
46     uint32_t count ;
47 } FIFO_Buf_t;
48
49 // =====> Buf Status <=====
50 typedef enum
51 {
52     FIFO_no_error,
53     FIFO_full,
54     FIFO_empty,
55     FIFO_Null
56 } FIFO_Buf_Status;
57
58 // =====> APIs <=====
59
60 FIFO_Buf_Status FIFO_init(FIFO_Buf_t* Student_Queue, SStudent_Node* Buf, uint32_t length );           // Student Queue Initialization
61 FIFO_Buf_Status FIFO_enqueue(FIFO_Buf_t* Student_Queue, SStudent_Node Student);                      // Check FIFO Full
62 FIFO_Buf_Status FIFO_is_Full(FIFO_Buf_t* Student_Queue);
63
64 void Add_Student_manually(FIFO_Buf_t* Student_Queue);
65
66 void Add_Student_From_File(FIFO_Buf_t* Student_Queue);
67
68 void Find_Student_By_id(FIFO_Buf_t* Student_Queue);
69
70 void Find_Student_By_FirstName(FIFO_Buf_t *Student_Queue);
71
72 void Find_Student_By_Course_id(FIFO_Buf_t *Student_Queue);
73
74 void Delete_Student_By_id(FIFO_Buf_t *Student_Queue);
75
76 void Update_Student_By_id(FIFO_Buf_t *Student_Queue);
77
78 void Print_Student_Count(FIFO_Buf_t *Student_Queue);                                              // print total Number of Students
79 void Show_Student_Information(FIFO_Buf_t *Student_Queue);                                         // View all Students information
80 void print_Student_info(SStudent_Node *student);                                                 // print information of Student
81
82 struct Studen_info_t *Check_Student_id(FIFO_Buf_t *Student_Queue, uint32_t id);                  // Check if student is exist or not
83
84 #endif /* STUDENT_SYS_H_ */
```

1-Student Structure

```
● ● ●  
1 // =====>> Student information <<=====  
2 struct Studen_info_t  
3 {  
4     char first_name[Name_Length];  
5     char last_name[Name_Length];  
6     int roll ;  
7     float GPA ;  
8     int Course_id[Number_Courses];  
9 };  
10 typedef struct Studen_info_t SStudent_Node ;
```

2-Database Structure

```
● ● ●  
1 // =====>> Type Definition <<=====  
2 typedef struct  
3 {  
4     SStudent_Node* base ;  
5     SStudent_Node* head ;  
6     SStudent_Node* tail ;  
7     uint32_t length ;  
8     uint32_t count ;  
9 } FIOF_Buf_t;
```

3-Function Prototypes

```
● ● ●
1 FIFO_Buf_Status FIFO_init(FIFO_Buf_t* Student_Queue, SStudent_Node* Buf, uint32_t length );           // Student Queue Initialization
2 FIFO_Buf_Status FIFO_enqueue(FIFO_Buf_t* Student_Queue, SStudent_Node Student);                         // Check FIFO Full
3 FIFO_Buf_Status FIFO_is_Full(FIFO_Buf_t* Student_Queue);
4
5 void Add_Student_manually(FIFO_Buf_t* Student_Queue);
6
7 void Add_Student_From_File(FIFO_Buf_t* Student_Queue);
8
9 void Find_Student_By_id(FIFO_Buf_t* Student_Queue);
10
11 void Find_Student_By_FirstName(FIFO_Buf_t *Student_Queue);
12
13 void Find_Student_By_Course_id(FIFO_Buf_t *Student_Queue);
14
15 void Delete_Student_By_id(FIFO_Buf_t *Student_Queue);
16
17 void Update_Student_By_id(FIFO_Buf_t *Student_Queue);
18
19 void Print_Student_Count(FIFO_Buf_t *Student_Queue);                                                 // print total Number of Students
20 void Show_Student_Information(FIFO_Buf_t *Student_Queue);                                         // View all Students information
21 void print_Student_info(SStudent_Node *student);                                              // print information of Student
22
23 struct Studen_info_t *Check_Student_id(FIFO_Buf_t *Student_Queue, uint32_t id);                  // Check if student is exist or not
24
25 #endif /* STUDENT_SYS_H_ */
```

4-Buffer Status

```
● ● ●
1 // ======>> Buf Status <<=====
2 typedef enum
3 {
4     FIFO_no_error,
5     FIFO_full,
6     FIFO_empty,
7     FIFO_Null
8 } FIFO_Buf_Status;
```

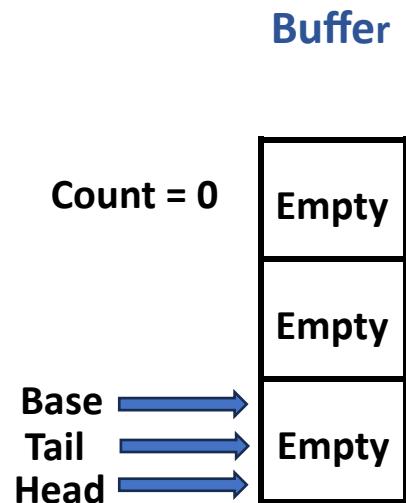
❖ Main.c

```
1  /*
2  =====
3  Name      : main.c
4  Author    : Mohammed Hassan
5  Created on : Oct 30, 2023
6  Description : First Term Project 2
7  =====
8 */
9 #include "Student_System.h"
10 int main()
11 {
12     int Select_Option;
13     FIFO_Buf_t Student_Queue;
14     struct Studen_info_t Student_Buffer[Max_Number_Students];
15
16     FIFO_init(&Student_Queue, Student_Buffer, Max_Number_Students);
17     while (1)
18     {
19         print("\n\t\t\t*****\n");
20         print("\t\t\t* Choose the Task that you want to Perform *");
21         print("\n\t\t\t*****\n\n");
22         print(" 1- Add the Student Details Manually\n\n");
23         print(" 2- Add the Student Details From Text File\n\n");
24         print(" 3- Find the Student Details by Roll Number\n\n");
25         print(" 4- Find the Student Details by First Name\n\n");
26         print(" 5- Find the Student Details by Course id\n\n");
27         print(" 6- Find the Total Number of Students\n\n");
28         print(" 7- Delete the Students Details by Roll Number\n\n");
29         print(" 8- Update the students Details by Roll Number\n\n");
30         print(" 9- Show All Information\n\n");
31         print(" 10- Exit\n\n");
32         print(" Enter your Choice to Perform the task: ");
33
34         scanf("%d", &Select_Option);
35         print("\n*****\n");
36         switch (Select_Option)
37         {
38             case 1:
39                 Add_Student_manually(&Student_Queue);
40                 break;
41             case 2:
42                 Add_Student_From_File(&Student_Queue);
43                 break;
44             case 3:
45                 Find_Student_By_id(&Student_Queue);
46                 break;
47             case 4:
48                 Find_Student_By_FirstName(&Student_Queue);
49                 break;
50             case 5:
51                 Find_Student_By_Course_id(&Student_Queue);
52                 break;
53             case 6:
54                 Print_Student_Count(&Student_Queue);
55                 break;
56             case 7:
57                 Delete_Student_By_id(&Student_Queue);
58                 break;
59             case 8:
60                 Update_Student_By_id(&Student_Queue);
61                 break;
62             case 9:
63                 Show_Student_Information(&Student_Queue);
64                 break;
65             case 10:
66                 print("\t\t\t***** Thank you *****\n");
67                 return 0;
68                 break;
69             default:
70                 print("\n Wrong Option Try Again\n");
71                 break;
72         }
73     }
74     return 0;
75 }
```

❖ Student_System.c

1- Student_System_init

- The function will Start by checking the input parameters validity.
- Start Initialize FIFO.



```
● ● ●
1 // =====> Initialize FIFO Buffer <=====
2 FIFO_Buf_Status FIFO_init(FIFO_Buf_t *Student_Queue, SStudent_Node *Buf, uint32_t length)
3 {
4     if (Buf == NULL)
5     {
6         return FIFO_Null;
7     }
8     // Initialize Students Data Base
9     Student_Queue->base = Buf;
10    Student_Queue->head = Buf;
11    Student_Queue->tail = Buf;
12    Student_Queue->length = length;
13    Student_Queue->count = 0;
14    return FIFO_no_error;
15 }
```

2- Add Student Manually

Take id number form user.
Check if this id is existed or not.
If not, start taking data from user.
After getting all data add it to the

After getting all data add it to the
queue

```
// =====> Add Student Manually <=====  
void Add_Student_manually(FIFO_Buf_t *Student_Queue)  
{  
    sStudent_Node New_Student;  
    int i;  
  
    // *Check if Student Data Base is valid  
    if (!Student_Queue->base || !Student_Queue->head || !Student_Queue->tail)  
    {  
        print("\n*****\n");  
        print("* [ERROR] Student Data Base Initialization ==> Failed\n");  
        print("*****\n");  
        return;  
    }  
  
    // *Check if Student Data Base is Full  
    if (Student_Queue->count == Student_Queue->length)  
    {  
        print("*****\n");  
        print("* [ERROR] Student Data Base is ==> Full\n");  
        print("*****\n");  
        return;  
    }  
  
    // * ADD STUDENT INFORMATION  
    print("\t\t=====\n");  
    print("\t\t\tEnter Student Data\t\t\t<==\n");  
    print("\t\t=====\n\n");  
  
    print(" Enter Student id : ");  
    scanf("%d", &New_Student.roll);  
  
    if (Check_Student_id(Student_Queue, New_Student.roll))  
    {  
        print("\n*****\n");  
        print("* [ERROR] Roll Number %d is Already Taken Before\n", New_Student.roll);  
        print("* [ERROR] Adding Student Manually Failed\n");  
        print("*****\n");  
        return ;  
    }  
    else  
    {  
        print(" Enter Student First Name : ");  
        scanf("%s", New_Student.first_name);  
  
        print(" Enter Student Last Name : ");  
        scanf("%s", New_Student.last_name);  
  
        print(" Enter Student GPA : ");  
        scanf("%f", &New_Student.GPA);  
  
        for (i = 0; i < Number_Courses; i++)  
        {  
            print(" Enter Course id Number %d : ", i + 1);  
            scanf("%d", &New_Student.Course_id[i]);  
        }  
        if (FIFO_enqueue(Student_Queue, New_Student) == FIFO_no_error)  
        {  
            print("\n*****\n");  
            print("* [INFO] Student Details is Added Successfully\n");  
        }  
        else  
        {  
            print("\n*****\n");  
            print("* [ERROR] Adding Student Manually Failed\n");  
            print("*****\n");  
            return;  
        }  
    }  
    Print_Student_Count(Student_Queue);  
}
```

➤ Check Student ID

- Scan if Student id existed or not.
- If not, return NULL.

```
● ● ●
1 // =====> Scan if Student id Existed or not <=====
2 struct Studen_info_t *Check_Student_id(FIOF_Buf_t *Student_Queue, uint32_t id)
3 {
4     int i;
5     SStudent_Node *New_Student = Student_Queue->tail;
6
7     for (i = 0; i < Student_Queue->count; i++)
8     {
9         if (New_Student->roll == id)
10         {
11             return New_Student;
12         }
13         else
14         {
15             New_Student++;
16         }
17     }
18     New_Student = NULL;
19     return New_Student;
20 }
```

➤ FIFO Enqueue

- Check buffer validity.
- Check if FIFO is Full.
- Add all data to the Buffer.
- Increment counter.
- Check if the head at the last element in the array.

```
1 // =====> FIFO Enqueue <=====
2 FIFO_Buf_Status FIFO_enqueue(FIFO_Buf_t *Student_Queue, SStudent_Node Student)
3 {
4     // Check FIFO Valid
5     if (!Student_Queue->base || !Student_Queue->head || !Student_Queue->tail)
6     {
7         return FIFO_Null;
8     }
9
10    // FIFO is Full
11    if (FIFO_is_Full(Student_Queue) == FIFO_full)
12    {
13        return FIFO_full;
14    }
15
16    *(Student_Queue->head) = Student;
17    Student_Queue->count++;
18    // Check if The head at the last element in the array.
19
20    if (Student_Queue->head == (Student_Queue->base + (Student_Queue->length * sizeof(SStudent_Node))))
21    {
22        Student_Queue->head = Student_Queue->base;
23    }
24    else
25    {
26        Student_Queue->head++;
27    }
28
29    return FIFO_no_error;
30 }
```

➤ Check FIFO Status

```
● ● ●
1 // =====> Check FIFO Status <=====
2 FIFO_Buf_Status FIFO_is_Full(FIFO_Buf_t *Student_Queue)
3 {
4     // Check FIFO Valid
5     if (!Student_Queue->base || !Student_Queue->head || !Student_Queue->tail)
6     {
7         print("*****\n");
8         print("* [ERROR] FIFO Enqueue Failed ==> FIFO is Null\n");
9         print("*****\n");
10        return FIFO_Null;
11    }
12
13    if (Student_Queue->count == Student_Queue->length)
14    {
15        print("*****\n");
16        print("* [ERROR] FIFO Enqueue Failed ==> FIFO is Full\n");
17        print("*****\n");
18        return FIFO_full;
19    }
20
21    return FIFO_no_error;
22 }
```

3- Add Student from File

- open Student_Data.txt with option read only.
- Check if the file is existed and include data or not.
- Reading file information until end of file using while loop.
- Start reading data from file and check if Student id exists before or not.

```
● ● ●
1 // =====> Add Student From File <=====
2 void Add_Student_From_File(FIFO_Buf_t *Student_Queue)
3 {
4     FILE *Student_file;
5     SStudent_Node New_Student;
6     int i;
7
8     /* Open Text file
9     Student_file = fopen("Student_Data.txt", "r");
10
11    /* check if this file is exist or not
12    if (Student_file == NULL)
13    {
14        print("*****\n");
15        print("* [ERROR] this File not found\n");
16        print("* [ERROR] Add Student From File Failed\n");
17        print("*****\n");
18        return;
19    }
20    /* reading file information until end of file
21    while (!feof(Student_file))
22    {
23        /* Reading Student id from file
24        fscanf(Student_file, "%d", &New_Student.roll);
25        if (Check_Student_id(Student_Queue, New_Student.roll))
26        {
27            print("*****\n");
28            print("* [ERROR] Roll Number %d is Already Taken Before\n", New_Student.roll);
29            print("*****\n\n");
30            fscanf(Student_file, "%*[^\n]");
31            continue ;
32        }
33
34        fscanf(Student_file, "%s", New_Student.first_name);
35        fscanf(Student_file, "%s", New_Student.last_name);
36        fscanf(Student_file, "%f", &New_Student.GPA);
37
38        for(i = 0 ; i < Number_Courses ; i++)
39        {
40            fscanf(Student_file, "%d", &New_Student.Course_id[i]);
41        }
42
43        if(FIFO_enqueue(Student_Queue, New_Student) == FIFO_no_error)
44        {
45            print("* [INFO] Roll Number %d is Saved Successfully\n", New_Student.roll);
46        }
47        else
48        {
49            print("*****\n");
50            print("* [ERROR] Adding Student From File Failed\n");
51            print("*****\n");
52            return ;
53        }
54    }
55    print("* [Info] Student Details Are Saved Successfully ");
56    Print_Student_Count(Student_Queue);
57
58    /* Close text file
59    fclose(Student_file);
60 }
```

4- Find Student by Roll Number

- Check if there is Students in the queue or not.
- Take an ID from the user to find it.
- Check if this id exists before.
- If Student was found print Student's information.

```
1 // =====>> Find Student by id <<=====
2 void Find_Student_By_id(FIFO_Buf_t* Student_Queue)
3 {
4     int Student_id;
5     SStudent_Node* Student ;
6     // *Check if Student Data Base is Empty
7     if (Student_Queue->count == 0)
8     {
9         print("*****\n");
10        print("* [ERROR] Student Data Base is Empty\n");
11        print("*****\n");
12        return;
13    }
14    print("\n*****\n");
15    print("\nEnter Student id number you want to find :")
16    scanf("%d", &Student_id);
17
18
19    Student = Check_Student_id(Student_Queue, Student_id);
20    if(Student == NULL)
21    {
22        print("*****\n");
23        print("* [ERROR] Student id number %d not found \n", Student_id);
24        print("*****\n");
25        return ;
26    }
27    else
28    {
29        print("Student id : %d is Found Successfully\n", Student_id);
30        print_Student_info(Student);
31    }
32 }
```

5- Find Student by First Name

```
● ● ●

1 // =====> Find Student by First Name <<=====
2 void Find_Student_By_FirstName(FIOF_Buf_t *Student_Queue)
3 {
4     SStudent_Node* Student = Student_Queue->tail ;
5     char First_Name[Name_Length] ;
6     int i, flag = 0 ;
7
8     // *Check if Student Data Base is Empty
9     if (Student_Queue->count == 0)
10    {
11        print("*****\n");
12        print("* [ERROR] Student Data Base is Empty\n");
13        print("*****\n");
14        return;
15    }
16    print("Enter Student First Name :")
17    scanf("%s",First_Name);
18
19    for(i = 0 ; i < Student_Queue->count ; i++)
20    {
21        if(strcmp(First_Name, Student->first_name) == 0)
22        {
23            print("\n*****\n");
24            print("Student : %s is found Successfully\n",First_Name);
25            print_Student_info(Student);
26            flag = 1 ;
27            return ;
28        }
29        // Check if we reach the last item in the queue
30        if((Student + 1) == (Student_Queue->base + Student_Queue->length))
31        {
32            Student = Student_Queue->base;
33        }
34        else
35        {
36            Student++;
37        }
38    }
39    if(!flag)
40    {
41        print("*****\n");
42        print("* [ERROR] Student %s Not found \n", First_Name );
43        print("*****\n");
44        return ;
45    }
46
47 }
```

6- Find Student by Course

```
● ● ●
1 // =====>> Find Student by course id <<=====
2 void Find_Student_By_Course_id(FIOF_Buf_t *Student_Queue)
3 {
4     SStudent_Node* Student = Student_Queue->tail ;
5     int i, j, counter = 0, Course_id;
6
7     if (Student_Queue->count == 0)
8     {
9         print("*****\n");
10        print("* [ERROR] Student Data Base is Empty\n");
11        print("*****\n");
12        return;
13    }
14
15    print("\nEnter course id: ");
16    scanf("%d", &Course_id);
17
18    for(i = 0 ; i < Student_Queue->count ; i++)
19    {
20        for(j = 0 ; j < Number_Courses ; j++)
21        {
22            if(Student->Course_id[j] == Course_id)
23            {
24                print_Student_info(Student);
25                counter++ ;
26            }
27        }
28        Student++;
29    }
30    if(counter == 0)
31    {
32        print("*****\n");
33        print("* [ERROR] Course ID %d Not found on the System\n", Course_id);
34        print("*****\n");
35        return;
36    }
37    else
38    {
39        print("*****\n");
40        print("[INFO] Number of Students enrolled in this course is : %d\n", counter);
41        print("*****\n");
42    }
43 }
```

7- Delete Student by ID Number

```
// =====> Delete Student By id <=====
void Delete_Student_By_id(FIOF_Buf_t *Student_Queue)
{
    int Student_id, i, flag = 0 ;
    SStudent_Node* Student = Student_Queue->base ;

    if (Student_Queue->count == 0)
    {
        print("*****\n");
        print("* [ERROR] Student Data Base is Empty\n");
        print("*****\n");
        return;
    }

    print("Enter Student id to Delete it from the system: ");
    scanf("%d", &Student_id);

    for(i = 0 ; i < Student_Queue->count ; i++)
    {
        if(Student->roll == Student_id)
        {
            *Student = *(Student_Queue->tail);
            Student_Queue->count-- ;

            if( (Student_Queue->tail+1) == (Student_Queue->base + Student_Queue->length) )
            {
                Student_Queue->tail = Student_Queue->base ;
            }
            else
            {
                Student_Queue->tail++;
            }
            flag = 1 ;
            break ;
        }

        else
        {
            flag = 0 ;
        }
        Student++ ;
    }

    if(flag == 1)
    {
        print("*****\n");
        print("* [INFO] Student ID Number %d Removed Successfully\n", Student_id);
        print("*****\n");
    }
    else
    {
        print("*****\n");
        print("* [ERROR] Student ID Number %d Not found\n", Student_id );
        print("*****\n");
    }
}
```

8- Update Student by id

```
1 // =====> Update Student Data <=====
2 void Update_Student_By_Id(FOIF_Buf_t *Student_Queue)
3 {
4     SStudent_Node* Student = Student_Queue->tail;
5     SStudent_Node *Student_Data_mod;
6     int option_number, Student_Id, New_Student_Id, i, flag = 0;
7
8     if (Student_Queue->count == 0)
9     {
10         print("*****\n");
11         print(" [ERROR] Student Data Base is Empty\n");
12         print("*****\n");
13         return;
14     }
15     print(" Enter Student id to update data: ");
16     scanf("%d", &student_Id);
17
18     Student_Data_mod = Check_Student_id(Student_Queue, Student_Id);
19     if(Student_Data_mod != NULL)
20     {
21         print("\n*****\n");
22         print("\n* Student Data *\n");
23         print("\n*****\n");
24         print_Student_info(Student_Data_mod);
25         print("\n*****\n");
26     }
27     else
28     {
29         print("*****\n");
30         print(" [ERROR] Student ID number %d Not found\n", Student_Id);
31         print("*****\n");
32         return;
33     }
34
35     print(" What do you want to change\n");
36     print(" 1: Student ID Number\n");
37     print(" 2: Student First Name\n");
38     print(" 3: Student Last Name\n");
39     print(" 4: Student GPA\n");
40     print(" 5: Student Course ID\n");
41     print("Enter the Number: ");
42     scanf("%d", &option_number);
43
44     switch(option_number)
45     {
46     case 1:
47         print(" \nEnter the New Student ID: ");
48         scanf("%d", &New_Student_Id);
49
50         for(i = 0 ; i < Student_Queue->count ; i++)
51         {
52             while(Student->roll == New_Student_Id)
53             {
54                 print("*****\n");
55                 print(" [ERROR] Roll Number %d is Exist\n", New_Student_Id);
56                 print("*****\n");
57                 flag = 1 ;
58                 break ;
59             }
60             if((Student + 1) == (Student_Queue->base + Student_Queue->length))
61             {
62                 Student = Student_Queue->base;
63             }
64             else
65             {
66                 Student++;
67             }
68         }
69         if(flag == 0)
70         {
71             Student_Data_mod->roll = New_Student_Id ;
72
73             print("\n* [INFO] Student ID Number %d Updated Successfully\n\n",New_Student_Id);
74
75             print("*****\n");
76             print(" Student Data After update *****\n");
77             Print_Student_info(Student_Data_mod);
78         }
79         break;
80     case 2:
81         print(" Enter the New Student First Name: ");
82         scanf("%s",Student_Data_mod->first_name);
83
84         print("* [INFO] Student %s Updated Successfully\n\n",Student_Data_mod->first_name);
85
86         print("*****\n");
87         print("***** Student Data After update *****\n");
88         print_Student_info(Student_Data_mod);
89         break ;
90     case 3:
91         print(" Enter The New Student Last Name: ");
92         scanf("%s",Student_Data_mod->last_name);
93
94         print("* [INFO] Student Last Name %s Updated Successfully\n\n",Student_Data_mod->last_name);
95
96         print("*****\n");
97         print("***** Student Data After update *****\n");
98         print_Student_info(Student_Data_mod);
99         break ;
100    case 4:
101        print(" Enter The New Student GPA: ");
102        scanf("%f",&Student_Data_mod->GPA);
103
104        print("* [INFO] Student GPA %.2f Updated Successfully\n\n",Student_Data_mod->GPA);
105
106        print("*****\n");
107        print("***** Student Data After update *****\n");
108        print_Student_info(Student_Data_mod);
109        break;
110    case 5:
111        print(" Enter Course Number From 1 to %d to update: ", Number_Courses);
112        scanf("%d", &option_number);
113        print(" Enter New Course ID: ");
114        scanf("%d", &Student_Data_mod->Course_id[option_number - 1]);
115
116        print("* [INFO] Student Course ID Number %d Updated Successfully\n\n",Student_Data_mod->Course_id[option_number - 1]);
117
118        print("*****\n");
119        print("***** Student Data After update *****\n");
120        print_Student_info(Student_Data_mod);
121        break;
122    default:
123        print("-----\n");
124        print(" [ERROR]Wrong Option\n");
125        print("*****\n");
126        break;
127    }
128 }
```

9- Show All Student Information

```
● ● ●
1 // =====> Show All Student Information <=====
2 void Show_Student_Information(FIFOBuf_t *Student_Queue)
3 {
4     int i;
5     SStudent_Node *Student_info;
6
7     // *Check if Student Data Base is Empty
8     if (Student_Queue->count == 0)
9     {
10         print("*****\n");
11         print("* [ERROR] Student Data Base is Empty\n");
12         print("*****\n");
13         return;
14     }
15
16     // *Check if Student Data Base is valid
17     if (!Student_Queue->base || !Student_Queue->head || !Student_Queue->tail)
18     {
19         print("*****\n");
20         print("* [ERROR] Student Data Base Initialization ==> Failed\n");
21         print("*****\n");
22         return;
23     }
24
25     Student_info = Student_Queue->tail;
26
27     for (i = 0; i < Student_Queue->count; i++)
28     {
29         print_Student_info(Student_info);
30         Student_info++;
31     }
32     Print_Student_Count(Student_Queue);
33 }
```

10- Print total Student Number

```
● ● ●
1 // =====>> Print Student Count Info <<=====
2 void Print_Student_Count(FIOF_Buf_t *Student_Queue)
3 {
4     print("\n***** Total Number Of Students *****\n")
5     print("* [INFO] The Total Number of Students is %d\n", Student_Queue->count);
6     print("* [INFO] You Can Add up to %d Students\n", Max_Number_Students);
7     print("* [INFO] You Can Add %d more Students\n", Max_Number_Students - Student_Queue->count);
8     print("*****\n");
9 }
```

11- Print Student information

```
● ● ●
1 // =====>> print Student info<<=====
2 void print_Student_info(SStudent_Node *student)
3 {
4     int i ;
5     print("*****\n");
6     print("Student ID: %d \n", student->roll);
7     print("Student First Name: %s \n", student->first_name);
8     print("Student Last Name: %s \n", student->last_name);
9     print("Student GPA Number: %.2f \n", student->GPA);
10
11    for (i = 0; i < Number_Courses; i++)
12    {
13        print("Course id Number %d are: %d \n", i + 1, student->Course_id[i]);
14    }
15 }
```

❖ Program Execution Snapshots

➤ Add Student Manually

```
*****  
* Choose the Task that you want to Perform *  
*****  
  
1- Add the Student Details Manually  
2- Add the Student Details From Text File  
3- Find the Student Details by Roll Number  
4- Find the Student Details by First Name  
5- Find the Student Details by Course id  
6- Find the Total Number of Students  
7- Delete the Students Details by Roll Number  
8- Update the students Details by Roll Number  
9- Show All Information  
10- Exit  
  
Enter your Choice to Perform the task: 1  
*****  
=====  
==>>      Enter Student Data      <<==  
=====  
  
Enter Student id : 100  
Enter Student First Name : Mohammed  
Enter Student Last Name : Hassan  
Enter Student GPA : 3.2  
Enter Course id Number 1 : 10  
Enter Course id Number 2 : 20  
Enter Course id Number 3 : 30  
Enter Course id Number 4 : 40  
Enter Course id Number 5 : 50  
  
=====  
* [INFO] Student Details is Added Successfully  
***** Total Number Of Students *****  
* [INFO] The Total Number of Students is 1  
* [INFO] You Can Add up to 50 Students  
* [INFO] You Can Add 49 more Students  
*****
```

➤ Add Student from File

Roll No.	Student Name	GPA	10	15	16	19	20
1	Mohammed Hassan	3.2	10	15	16	19	20
2	Khaled Alaa	2.7	12	16	20	14	13
3	Mohammed Goda	3.7	15	11	10	13	17
1	Yahya Amr	2.8	16	20	14	13	12
4	Mahmoed nour	3.9	19	20	12	15	16
5	Salma Ahmed	3.5	12	15	16	18	19
5	Mostafa Ali	3.4	11	12	10	20	15

```
*****  
* Choose the Task that you want to Perform *  
*****  
  
1- Add the Student Details Manually  
2- Add the Student Details From Text File  
3- Find the Student Details by Roll Number  
4- Find the Student Details by First Name  
5- Find the Student Details by Course id  
6- Find the Total Number of Students  
7- Delete the Students Details by Roll Number  
8- Update the students Details by Roll Number  
9- Show All Information  
10- Exit  
  
Enter your Choice to Perform the task: 2  
=====*[INFO] Roll Number 1 is Saved Successfully  
*[INFO] Roll Number 2 is Saved Successfully  
*[INFO] Roll Number 3 is Saved Successfully  
=====*[ERROR] Roll Number 1 is Already Taken Before  
=====*[INFO] Roll Number 4 is Saved Successfully  
*[INFO] Roll Number 5 is Saved Successfully  
=====*[ERROR] Roll Number 5 is Already Taken Before  
=====*[Info] Student Details Are Saved Successfully  
***** Total Number Of Students *****  
*[INFO] The Total Number of Students is 5  
*[INFO] You Can Add up to 50 Students  
*[INFO] You Can Add 45 more Students  
*****
```

➤ Find Student Details by ID Number

```
*****  
* Choose the Task that you want to Perform *  
*****  
  
1- Add the Student Details Manually  
2- Add the Student Details From Text File  
3- Find the Student Details by Roll Number  
4- Find the Student Details by First Name  
5- Find the Student Details by Course id  
6- Find the Total Number of Students  
7- Delete the Students Details by Roll Number  
8- Update the students Details by Roll Number  
9- Show All Information  
10- Exit  
  
Enter your Choice to Perform the task: 3  
  
*****  
Enter Student id number you want to find :4  
Student id : 4 is Found Successfully  
*****  
Student ID: 4  
Student First Name: Mahmoed  
Student Last Name: nour  
Student GPA Number: 3.30  
Course id Number 1 are: 19  
Course id Number 2 are: 20  
Course id Number 3 are: 12  
Course id Number 4 are: 15  
Course id Number 5 are: 16
```

➤ Find Student details by First Name

```
******  
* Choose the Task that you want to Perform *  
*****  
  
1- Add the Student Details Manually  
2- Add the Student Details From Text File  
3- Find the Student Details by Roll Number  
4- Find the Student Details by First Name  
5- Find the Student Details by Course id  
6- Find the Total Number of Students  
7- Delete the Students Details by Roll Number  
8- Update the students Details by Roll Number  
9- Show All Information  
10- Exit  
  
Enter your Choice to Perform the task: 4  
*****  
  
Enter Student First Name :Mai  
  
Student : Mai is found Successfully  
*****  
Student ID: 3  
Student First Name: Mai  
Student Last Name: Mohammed  
Student GPA Number: 2.50  
Course id Number 1 are: 15  
Course id Number 2 are: 11  
Course id Number 3 are: 10  
Course id Number 4 are: 13  
Course id Number 5 are: 17
```

➤ Find Student Details by Course ID

```
*****  
* Choose the Task that you want to Perform *  
*****  
  
1- Add the Student Details Manually  
2- Add the Student Details From Text File  
3- Find the Student Details by Roll Number  
4- Find the Student Details by First Name  
5- Find the Student Details by Course id  
6- Find the Total Number of Students  
7- Delete the Students Details by Roll Number  
8- Update the students Details by Roll Number  
9- Show All Information  
10- Exit  
  
Enter your Choice to Perform the task: 5  
  
*****  
Enter course id: 12  
*****  
Student ID: 2  
Student First Name: Khaled  
Student Last Name: Alaa  
Student GPA Number: 2.70  
Course id Number 1 are: 12  
Course id Number 2 are: 16  
Course id Number 3 are: 20  
Course id Number 4 are: 14  
Course id Number 5 are: 13  
*****  
Student ID: 4  
Student First Name: Mahmoed  
Student Last Name: nour  
Student GPA Number: 3.30  
Course id Number 1 are: 19  
Course id Number 2 are: 20  
Course id Number 3 are: 12  
Course id Number 4 are: 15  
Course id Number 5 are: 16  
*****  
Student ID: 5  
Student First Name: Salma  
Student Last Name: Ahmed  
Student GPA Number: 3.50  
Course id Number 1 are: 12  
Course id Number 2 are: 15  
Course id Number 3 are: 16  
Course id Number 4 are: 18  
Course id Number 5 are: 19  
===== [INFO] Number of Students enrolled in this course is : 3 =====
```

➤ Print Total Number of Students

```
*****  
* Choose the Task that you want to Perform *  
*****  
  
1- Add the Student Details Manually  
2- Add the Student Details From Text File  
3- Find the Student Details by Roll Number  
4- Find the Student Details by First Name  
5- Find the Student Details by Course id  
6- Find the Total Number of Students  
7- Delete the Students Details by Roll Number  
8- Update the students Details by Roll Number  
9- Show All Information  
10- Exit  
  
Enter your Choice to Perform the task: 6  
*****  
* [INFO] Total Number Of Students is 5  
* [INFO] You Can Add up to 50 Students  
* [INFO] You Can Add 45 more Students  
*****
```

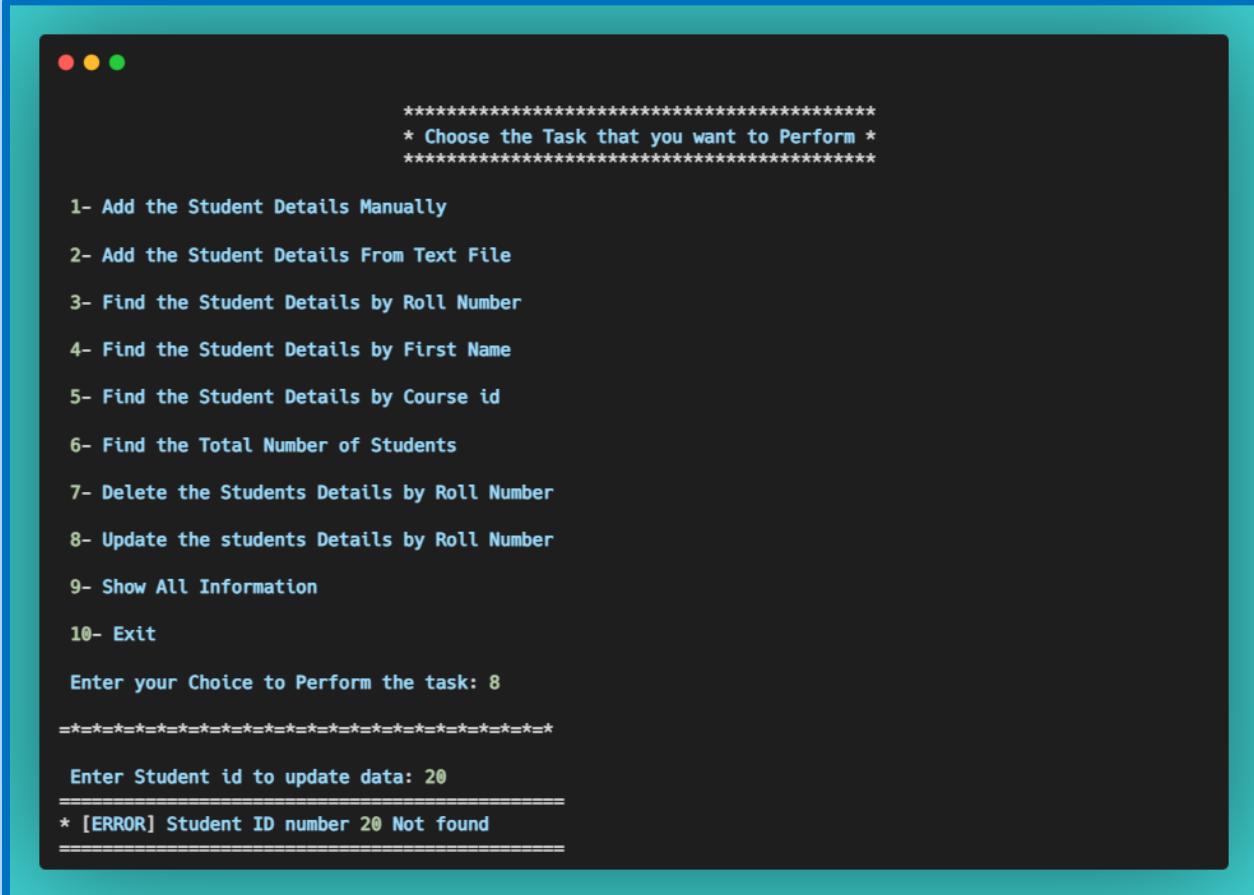
➤ Delete Student by Roll Number

```
*****  
* Choose the Task that you want to Perform *  
*****  
  
1- Add the Student Details Manually  
2- Add the Student Details From Text File  
3- Find the Student Details by Roll Number  
4- Find the Student Details by First Name  
5- Find the Student Details by Course id  
6- Find the Total Number of Students  
7- Delete the Students Details by Roll Number  
8- Update the students Details by Roll Number  
9- Show All Information  
10- Exit  
  
Enter your Choice to Perform the task: 7  
*****  
  
Enter Student id to Delete it from the system: 3  
=====
```

```
*****  
* Choose the Task that you want to Perform *  
*****  
  
1- Add the Student Details Manually  
2- Add the Student Details From Text File  
3- Find the Student Details by Roll Number  
4- Find the Student Details by First Name  
5- Find the Student Details by Course id  
6- Find the Total Number of Students  
7- Delete the Students Details by Roll Number  
8- Update the students Details by Roll Number  
9- Show All Information  
10- Exit  
  
Enter your Choice to Perform the task: 7  
  
=====  
Enter Student id to Delete it from the system: 15  
=====
```

➤ Update Student Details by Roll Number

1-If Student ID Number Not found:



```
*****  
* Choose the Task that you want to Perform *  
*****  
  
1- Add the Student Details Manually  
2- Add the Student Details From Text File  
3- Find the Student Details by Roll Number  
4- Find the Student Details by First Name  
5- Find the Student Details by Course id  
6- Find the Total Number of Students  
7- Delete the Students Details by Roll Number  
8- Update the students Details by Roll Number  
9- Show All Information  
10- Exit  
  
Enter your Choice to Perform the task: 8  
*****  
  
Enter Student id to update data: 20  
=====  
* [ERROR] Student ID number 20 Not found  
=====
```

2-Update Student id:

* Choose the Task that you want to Perform *

1- Add the Student Details Manually
2- Add the Student Details From Text File
3- Find the Student Details by Roll Number
4- Find the Student Details by First Name
5- Find the Student Details by Course id
6- Find the Total Number of Students
7- Delete the Students Details by Roll Number
8- Update the students Details by Roll Number
9- Show All Information
10- Exit

Enter your Choice to Perform the task: 8

Enter Student id to update data: 2

* Student Data *

Student ID: 2
Student First Name: Khaled
Student Last Name: Alaa
Student GPA Number: 2.70
Course id Number 1 are: 12
Course id Number 2 are: 16
Course id Number 3 are: 20
Course id Number 4 are: 14
Course id Number 5 are: 13

What do you want to change
1: Student ID Number
2: Student First Name
3: Student Last Name
4: Student GPA
5: Student Course ID

Enter the Number: 1
Enter the New Student ID: 10

* [INFO] Student ID Number 10 Updated Successfully

***** Student Data After updated *****

Student ID: 10
Student First Name: Khaled
Student Last Name: Alaa
Student GPA Number: 2.70
Course id Number 1 are: 12
Course id Number 2 are: 16
Course id Number 3 are: 20
Course id Number 4 are: 14
Course id Number 5 are: 13

3- update Student id failed

```
*****  
* Choose the Task that you want to Perform *  
*****  
  
1- Add the Student Details Manually  
2- Add the Student Details From Text File  
3- Find the Student Details by Roll Number  
4- Find the Student Details by First Name  
5- Find the Student Details by Course id  
6- Find the Total Number of Students  
7- Delete the Students Details by Roll Number  
8- Update the students Details by Roll Number  
9- Show All Information  
10- Exit  
  
Enter your Choice to Perform the task: 8  
*****  
  
Enter Student id to update data: 10  
*****  
* Student Data *  
*****  
*****  
Student ID: 10  
Student First Name: Khaled  
Student Last Name: Alaa  
Student GPA Number: 2.70  
Course id Number 1 are: 12  
Course id Number 2 are: 16  
Course id Number 3 are: 20  
Course id Number 4 are: 14  
Course id Number 5 are: 13  
*****  
What do you want to change  
1: Student ID Number  
2: Student First Name  
3: Student Last Name  
4: Student GPA  
5: Student Course ID  
  
Enter the Number: 1  
Enter the New Student ID: 3  
=====  
* [ERROR] Roll Number 3 is Exist  
=====
```

4-update Student First Name

* Choose the Task that you want to Perform *

1- Add the Student Details Manually
2- Add the Student Details From Text File
3- Find the Student Details by Roll Number
4- Find the Student Details by First Name
5- Find the Student Details by Course id
6- Find the Total Number of Students
7- Delete the Students Details by Roll Number
8- Update the students Details by Roll Number
9- Show All Information
10- Exit

Enter your Choice to Perform the task: 8

Enter Student id to update data: 4

* Student Data *

Student ID: 4
Student First Name: Mahmoed
Student Last Name: nour
Student GPA Number: 3.30
Course id Number 1 are: 19
Course id Number 2 are: 20
Course id Number 3 are: 12
Course id Number 4 are: 15
Course id Number 5 are: 16

What do you want to change
1: Student ID Number
2: Student First Name
3: Student Last Name
4: Student GPA
5: Student Course ID

Enter the Number: 2
Enter The New Student First Name: Osama

* [INFO] Student Osama Updated Successfully

***** Student Data After update *****

Student ID: 4
Student First Name: Osama
Student Last Name: nour
Student GPA Number: 3.30
Course id Number 1 are: 19
Course id Number 2 are: 20
Course id Number 3 are: 12
Course id Number 4 are: 15
Course id Number 5 are: 16

5-update Student Last Name

* Choose the Task that you want to Perform *

1- Add the Student Details Manually
2- Add the Student Details From Text File
3- Find the Student Details by Roll Number
4- Find the Student Details by First Name
5- Find the Student Details by Course id
6- Find the Total Number of Students
7- Delete the Students Details by Roll Number
8- Update the students Details by Roll Number
9- Show All Information
10- Exit

Enter your Choice to Perform the task: 8

Enter Student id to update data: 4

* Student Data *

Student ID: 4
Student First Name: Osama
Student Last Name: nour
Student GPA Number: 3.30
Course id Number 1 are: 19
Course id Number 2 are: 20
Course id Number 3 are: 12
Course id Number 4 are: 15
Course id Number 5 are: 16

What do you want to change
1: Student ID Number
2: Student First Name
3: Student Last Name
4: Student GPA
5: Student Course ID

Enter the Number: 3
Enter The New Student Last Name: Ali

* [INFO] Student Last Name Ali Updated Successfully

***** Student Data After update *****

Student ID: 4
Student First Name: Osama
Student Last Name: Ali
Student GPA Number: 3.30
Course id Number 1 are: 19
Course id Number 2 are: 20
Course id Number 3 are: 12
Course id Number 4 are: 15
Course id Number 5 are: 16

6-Update Student GPA

* Choose the Task that you want to Perform *

1- Add the Student Details Manually
2- Add the Student Details From Text File
3- Find the Student Details by Roll Number
4- Find the Student Details by First Name
5- Find the Student Details by Course id
6- Find the Total Number of Students
7- Delete the Students Details by Roll Number
8- Update the students Details by Roll Number
9- Show All Information
10- Exit

Enter your Choice to Perform the task: 8

Enter Student id to update data: 3

* Student Data *

Student ID: 3
Student First Name: Mai
Student Last Name: Mohammed
Student GPA Number: 2.50
Course id Number 1 are: 15
Course id Number 2 are: 11
Course id Number 3 are: 10
Course id Number 4 are: 13
Course id Number 5 are: 17

What do you want to change
1: Student ID Number
2: Student First Name
3: Student Last Name
4: Student GPA
5: Student Course ID

Enter the Number: 4
Enter The New Student GPA: 3.2

* [INFO] Student GPA 3.20 Updated Successfully

***** Student Data After update *****

Student ID: 3
Student First Name: Mai
Student Last Name: Mohammed
Student GPA Number: 3.20
Course id Number 1 are: 15
Course id Number 2 are: 11
Course id Number 3 are: 10
Course id Number 4 are: 13
Course id Number 5 are: 17

7-update Student Course ID Number

● ● ●

* Choose the Task that you want to Perform *

1- Add the Student Details Manually
2- Add the Student Details From Text File
3- Find the Student Details by Roll Number
4- Find the Student Details by First Name
5- Find the Student Details by Course id
6- Find the Total Number of Students
7- Delete the Students Details by Roll Number
8- Update the students Details by Roll Number
9- Show All Information
10- Exit

Enter your Choice to Perform the task: 8

Enter Student id to update data: 5

* Student Data *

Student ID: 5
Student First Name: Salma
Student Last Name: Ahmed
Student GPA Number: 3.50
Course id Number 1 are: 12
Course id Number 2 are: 15
Course id Number 3 are: 16
Course id Number 4 are: 18
Course id Number 5 are: 19

What do you want to change
1: Student ID Number
2: Student First Name
3: Student Last Name
4: Student GPA
5: Student Course ID

Enter the Number: 5
Enter Course Number From 1 to 5 to update: 4
Enter New Course ID: 10

* [INFO] Student Course ID Number 10 Updated Successfully

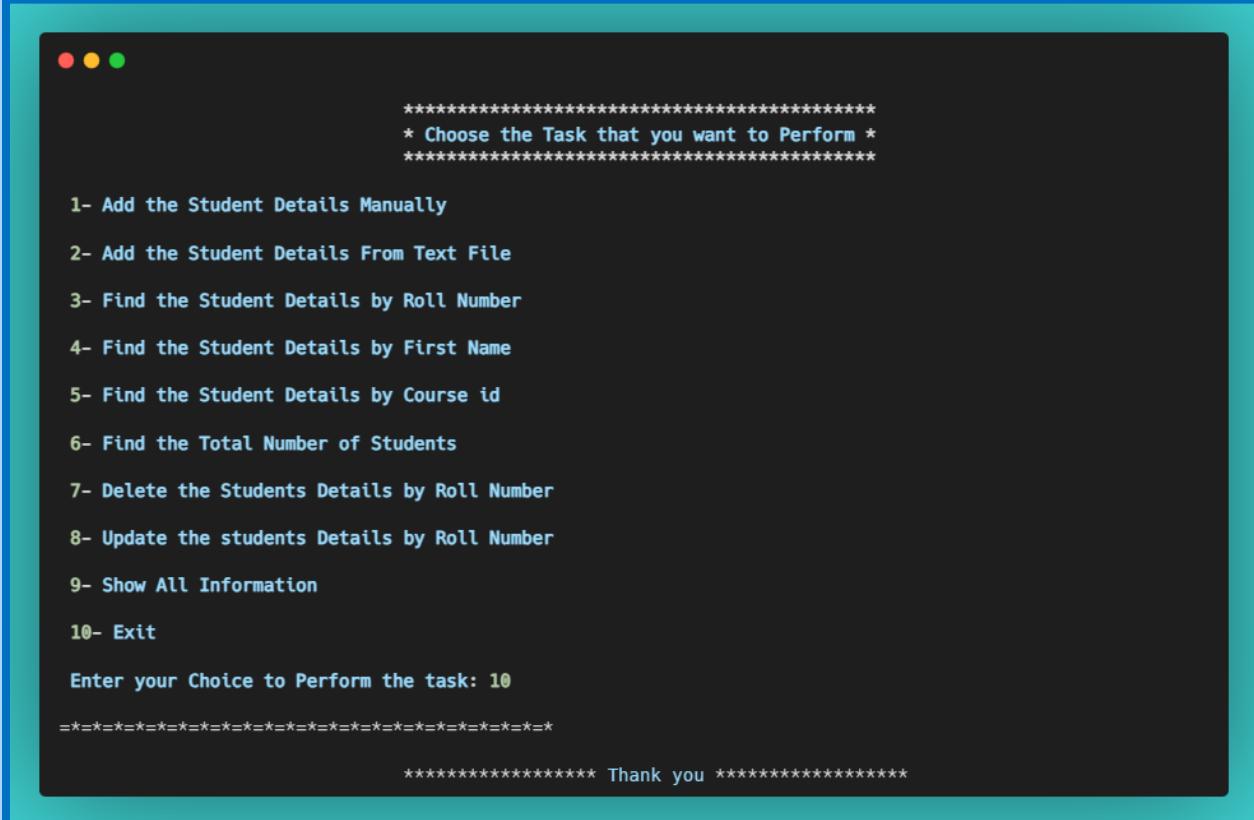
***** Student Data After update *****

Student ID: 5
Student First Name: Salma
Student Last Name: Ahmed
Student GPA Number: 3.50
Course id Number 1 are: 12
Course id Number 2 are: 15
Course id Number 3 are: 16
Course id Number 4 are: 10
Course id Number 5 are: 19

8-Show All Details information of Students

```
*****  
* Choose the Task that you want to Perform *  
*****  
  
1- Add the Student Details Manually  
2- Add the Student Details From Text File  
3- Find the Student Details by Roll Number  
4- Find the Student Details by First Name  
5- Find the Student Details by Course id  
6- Find the Total Number of Students  
7- Delete the Students Details by Roll Number  
8- Update the students Details by Roll Number  
9- Show All Information  
10- Exit  
  
Enter your Choice to Perform the task: 9  
*****  
*****  
Student ID: 1  
Student First Name: Mohammed  
Student Last Name: Hassan  
Student GPA Number: 3.20  
Course id Number 1 are: 10  
Course id Number 2 are: 15  
Course id Number 3 are: 16  
Course id Number 4 are: 19  
Course id Number 5 are: 20  
*****  
Student ID: 10  
Student First Name: Khaled  
Student Last Name: Alaa  
Student GPA Number: 2.70  
Course id Number 1 are: 12  
Course id Number 2 are: 16  
Course id Number 3 are: 20  
Course id Number 4 are: 14  
Course id Number 5 are: 13  
*****  
Student ID: 3  
Student First Name: Mai  
Student Last Name: Mohammed  
Student GPA Number: 3.20  
Course id Number 1 are: 15  
Course id Number 2 are: 11  
Course id Number 3 are: 10  
Course id Number 4 are: 13  
Course id Number 5 are: 17  
*****  
Student ID: 4  
Student First Name: Osama  
Student Last Name: Ali  
Student GPA Number: 3.30  
Course id Number 1 are: 19  
Course id Number 2 are: 20  
Course id Number 3 are: 12  
Course id Number 4 are: 15  
Course id Number 5 are: 16  
*****  
Student ID: 5  
Student First Name: Salma  
Student Last Name: Ahmed  
Student GPA Number: 3.50  
Course id Number 1 are: 12  
Course id Number 2 are: 15  
Course id Number 3 are: 16  
Course id Number 4 are: 10  
Course id Number 5 are: 19  
  
***** Total Number Of Students *****  
* [INFO] The Total Number of Students is 5  
* [INFO] You Can Add up to 50 Students  
* [INFO] You Can Add 45 more Students  
*****
```

9-Exit the Program



```
*****  
* Choose the Task that you want to Perform *  
*****  
  
1- Add the Student Details Manually  
2- Add the Student Details From Text File  
3- Find the Student Details by Roll Number  
4- Find the Student Details by First Name  
5- Find the Student Details by Course id  
6- Find the Total Number of Students  
7- Delete the Students Details by Roll Number  
8- Update the students Details by Roll Number  
9- Show All Information  
10- Exit  
  
Enter your Choice to Perform the task: 10  
***** Thank you *****
```

Thank You

Mohammed Hassam