

Student Information Management

Report



Mastering Embedded System Online Diploma

Prepared By : Sherif Ashraf Khedr

My Profile Link

Table Of Content

1 - System Requirement.....	4
1.1 - Add Student Details Manually (Task 1 and 2).....	4
1.2 - Find Student Details (Tasks 3, 4, 5, and 6).....	4
1.3 - Delete Student Details (Task 7).....	4
1.4 - Update Student Details (Task 8).....	4
1.5 - Show All Information (Task 9).....	5
1.6 - Exit the System (Task 10).....	5
1.7 - General Requirements.....	5
2 - System Implementation.....	6
2.1 - Fifo Functions Implementation.....	6
2.1.1 - FIFO_uddtInitBuffer.....	7
2.1.2 - FIFO_uddtPushData.....	8
2.1.3 - FIFO_uddtPopData.....	9
2.1.4 - FIFO_uddtIsFull.....	10
2.1.5 - FIFO_uddtPrintBuffer.....	11
2.1.6 - FIFO_uddtIsUniqueRollNumber.....	12
2.1.7 - FIFO_uddtShowDbStatistics.....	13
2.1.8 - FIFO_uddtFindByRollNumber.....	14
2.1.9 - FIFO_uddtFindByFirstName.....	15
2.1.10 - FIFO_uddtFindByCourseld.....	16
2.1.11 - FIFO_uddtDeleteByRollNumber.....	17
2.1.12 - FIFO_uddtUpdateRollNumber.....	18
2.1.13 - FIFO_uddtUpdateFirstName.....	19
2.1.14 - FIFO_uddtUpdateLastName.....	20
2.1.15 - FIFO_uddtUpdateGpa.....	21
2.1.16 - FIFO_uddtUpdateCourses.....	22
2.1.17 - FIFO_uddtIsValidRollNumber.....	23
2.2 - Student Functions Implementation.....	24

2.2.1 - ConvertToLower.....	25
2.2.2 - u32GetRollNumber.....	26
2.2.3 - vGetFirstName.....	27
2.2.4 - vGetLastName.....	27
2.2.5 - f32GetGpa.....	28
2.2.6 - vGetCoursesId.....	29
2.2.7 - STUDENT_vInitDataBase.....	30
2.2.8 - STUDENT_vAddUserByFile.....	31
2.2.9 - STUDENT_vAddUserByFile.....	32
2.2.10 - STUDENT_vFindByRollNumber.....	33
2.2.11 - STUDENT_vFindByFirstName.....	34
2.2.12 - STUDENT_vFindByLastName.....	35
2.2.13 - STUDENT_vPrintDbInfo.....	36
2.2.14 - STUDENT_vPrintAllStudents.....	36
2.2.15 - STUDENT_vDeleteUser.....	37
2.2.14 - STUDENT_vUpdateUser.....	38
2.3 - Main Functions Implementation.....	40
2.4 - Standard Data Type File.....	42
3 - Database Text File.....	43
4 - Program Execution.....	44
4.1 - Main Menu.....	44
4.2 - Add User Manual.....	44
4.3 - Add Student Automated.....	45
4.4 - Find By Roll Number.....	46
4.5 - Find By First Name.....	46
4.6 - Find By Course Id.....	47
4.7 - Print Count Of Student.....	48
4.8 - Delete By Roll Number.....	48
4.9 - Roll Number Update.....	49
4.10 - First Name Update.....	49
4.11 - Last Name Update.....	50
4.12 - Gpa Update.....	50
4.13 - Course Id Update.....	51

4.14 - Print All Student Data.....52

1 - System Requirement

1.1 - Add Student Details Manually (Task 1 and 2)

- The system should prompt the user to input a unique roll number, first name, last name, GPA, and 5 course IDs for a new student.
- The system should add this information to the database.

1.2 - Find Student Details (Tasks 3, 4, 5, and 6)

- For finding student details by roll number, first name, or course ID, the system should prompt the user to enter the corresponding information.
- The system should then search the database and display the details of the student(s) matching the entered criteria.
- For finding the total number of students, the system should count the number of records in the database and display the result.

1.3 - Delete Student Details (Task 7)

- The system should prompt the user to enter the roll number of the student whose details need to be deleted.
- The system should then remove the corresponding record from the database.

1.4 - Update Student Details (Task 8)

- The system should prompt the user to enter the roll number of the student whose details need to be updated.
- The system should then allow the user to modify any of the student's information (roll number, first name, last name, GPA, course IDs).

1.5 - Show All Information (Task 9)

- The system should display all the information stored in the database, including roll number, first name, last name, GPA, and course IDs, for all students.

1.6 - Exit the System (Task 10)

- The system should provide an option for the user to exit the program.

1.7 - General Requirements

- The system should handle input validation to ensure data integrity.
- It should provide appropriate error messages if the user enters invalid data or if a requested operation cannot be performed.
- The system should use proper data structures for storing and managing student information.
- Consider implementing file I/O operations if you are loading or saving data from/to a text file.
- Ensure the system is user-friendly, with clear prompts and instructions.
- Include proper comments and documentation to help others understand the code.

2 - System Implementation

2.1 - Fifo Functions Implementation

```
1  *****/
2 // Author      : Sherif Ashraf Khedr
3 // Project     : 01_FIFO_Buffer
4 // File        : fifo.h
5 // Date        : 27 Nov 2023
6 // GitHub       : https://github.com/sherifkhadr
7 *****/
8
9 #ifndef FIFO_H_
10 #define FIFO_H_
11
12 // user configuration
13 // select data width type (uint8 , uint16 , uint32 ...)
14
15 #include "stdTypes.h"
16 #define DATA_WIDTH      ST_studentInfo_t
17
18
19 typedef enum
20 {
21     FIFO_NOK = 0,
22     FIFO_OK,
23     FIFO_IS_FULL,
24     FIFO_IS_EMPTY,
25     FIFO_NEITHER_FULL_NOR_EMPTY,
26     FIFO_IS_NULL,
27     FIFO_IS_PTR_NULL,
28     FIFO_ID_IS_UNIQUE,
29     FIFO_ID_NOT_UNIQUE,
30     FIFO_ID_VALID,
31     FIFO_ID_NOT_VALID
32 }EN_FIFO_status_t;
33
34 typedef struct
35 {
36     uint8 length;
37     uint8 count;
38     DATA_WIDTH *head;
39     DATA_WIDTH *base;
40     DATA_WIDTH *tail;
41 }ST_FIFO_Buffer_t;
```

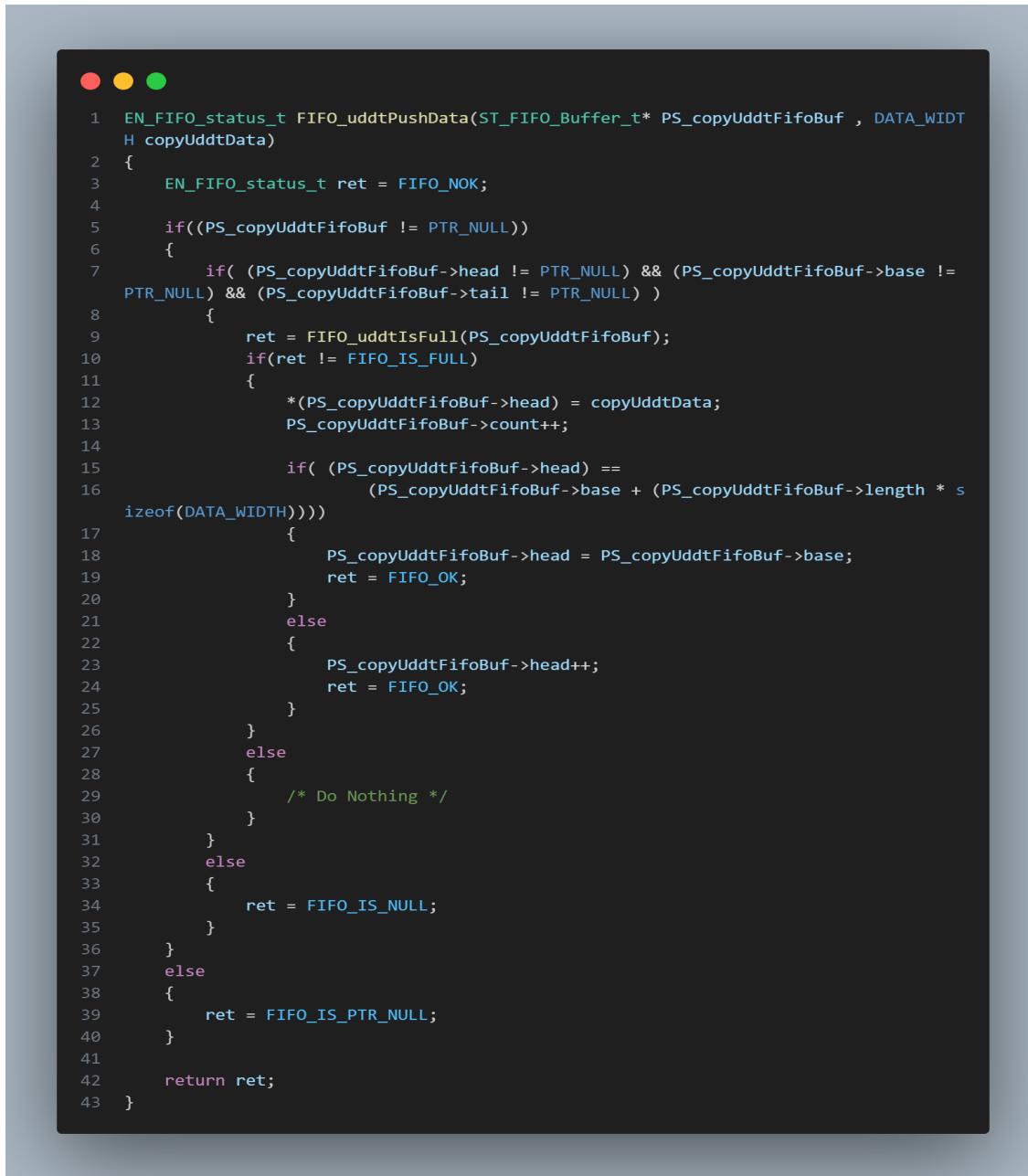
2.1.1 - FIFO_uddtInitBuffer

This function is responsible to initialise the buffer member head,tail,base and count also to assign the length that take as input

```
1 EN_FIFO_status_t FIFO_uddtInitBuffer(ST_FIFO_Buffer_t* PS_copyUddtFifoBuf , DATA_WI
2 DTH *P_copyUddtBuf , uint32 copy_U32BufLength)
3 {
4     EN_FIFO_status_t ret = FIFO_NOK;
5
6     if((PS_copyUddtFifoBuf != PTR_NULL) && (P_copyUddtBuf != PTR_NULL))
7     {
8         PS_copyUddtFifoBuf->base = P_copyUddtBuf;
9         PS_copyUddtFifoBuf->head = P_copyUddtBuf;
10        PS_copyUddtFifoBuf->tail = P_copyUddtBuf;
11        PS_copyUddtFifoBuf->length = copy_U32BufLength;
12        PS_copyUddtFifoBuf->count = 0;
13        ret = FIFO_OK;
14    }
15    else
16    {
17        ret = FIFO_IS_PTR_NULL;
18    }
19
20 }
```

2.1.2 - FIFO_udtPushData

This function is responsible to check if buffer is full or not and if not it will add the data in the address of head then will increment count



```
1 EN_FIFO_status_t FIFO_udtPushData(ST_FIFO_Buffer_t* PS_copyUddtFifoBuf , DATA_WIDT
2 H copyUddtData)
3 {
4     EN_FIFO_status_t ret = FIFO_NOK;
5
6     if((PS_copyUddtFifoBuf != PTR_NULL))
7     {
8         if( (PS_copyUddtFifoBuf->head != PTR_NULL) && (PS_copyUddtFifoBuf->base !=
9 PTR_NULL) && (PS_copyUddtFifoBuf->tail != PTR_NULL) )
10        {
11            if( FIFO_udtIsFull(PS_copyUddtFifoBuf);
12                if(ret != FIFO_IS_FULL)
13                {
14                    *(PS_copyUddtFifoBuf->head) = copyUddtData;
15                    PS_copyUddtFifoBuf->count++;
16
17                    if( (PS_copyUddtFifoBuf->head) ==
18                        (PS_copyUddtFifoBuf->base + (PS_copyUddtFifoBuf->length * s
19                         sizeof(DATA_WIDTH)))
20                    {
21                        PS_copyUddtFifoBuf->head = PS_copyUddtFifoBuf->base;
22                        ret = FIFO_OK;
23                    }
24                    else
25                    {
26                        PS_copyUddtFifoBuf->head++;
27                        ret = FIFO_OK;
28                    }
29                    /* Do Nothing */
30                }
31            }
32            else
33            {
34                ret = FIFO_IS_NULL;
35            }
36        }
37    else
38    {
39        ret = FIFO_IS_PTR_NULL;
40    }
41
42    return ret;
43 }
```

2.1.3 - FIFO_udtPopData

This function is responsible to check if buffer is empty or not and if not it will return the data in the passed argument then will decrement count



```
● ● ●
1 EN_FIFO_status_t FIFO_udtPopData(ST_FIFO_Buffer_t* PS_copyUddtFifoBuf , DATA_WIDTH
2 *copyUddtRetOfData)
3 {
4     EN_FIFO_status_t ret = FIFO_NOK;
5
6     if((PS_copyUddtFifoBuf != PTR_NULL))
7     {
8         if( (PS_copyUddtFifoBuf->head != PTR_NULL) && (PS_copyUddtFifoBuf->base != PTR_NULL) && (PS_copyUddtFifoBuf->tail != PTR_NULL) )
9         {
10            ret = FIFO_udtIsFull(PS_copyUddtFifoBuf);
11            if(ret == FIFO_IS_EMPTY)
12            {
13                /* Do Nothing */
14            }
15            else
16            {
17                *copyUddtRetOfData = *(PS_copyUddtFifoBuf->tail);
18                PS_copyUddtFifoBuf->count--;
19
20                if( (PS_copyUddtFifoBuf->tail) ==
21                    (PS_copyUddtFifoBuf->base + (PS_copyUddtFifoBuf->length * sizeof(DATA_WIDTH)))
22                {
23                    PS_copyUddtFifoBuf->tail = PS_copyUddtFifoBuf->base;
24                    ret = FIFO_OK;
25                }
26                else
27                {
28                    PS_copyUddtFifoBuf->tail++;
29                    ret = FIFO_OK;
30                }
31            }
32        else
33        {
34            ret = FIFO_IS_NULL;
35        }
36    }
37    else
38    {
39        ret = FIFO_IS_PTR_NULL;
40    }
41
42    return ret;
43 }
```

2.1.4 - FIFO_udtIsFull

This function to check the buffer is full or not or empty

```
● ○ ●
1 EN_FIFO_status_t FIFO_udtIsFull(ST_FIFO_Buffer_t* PS_copyUddtFifoBuf)
2 {
3     EN_FIFO_status_t ret = FIFO_NOK;
4
5     if((PS_copyUddtFifoBuf != PTR_NULL))
6     {
7         if( (PS_copyUddtFifoBuf->head != PTR_NULL) && (PS_copyUddtFifoBuf->base != PTR_NULL) && (PS_copyUddtFifoBuf->tail != PTR_NULL) )
8         {
9             if(PS_copyUddtFifoBuf->count == PS_copyUddtFifoBuf->length)
10            {
11                ret = FIFO_IS_FULL;
12            }
13            else if(PS_copyUddtFifoBuf->count < PS_copyUddtFifoBuf->length)
14            {
15                ret = FIFO_NEITHER_FULL_NOR_EMPTY;
16            }
17            else if(PS_copyUddtFifoBuf->count == 0)
18            {
19                ret = FIFO_IS_EMPTY;
20            }
21            else/* Do Nothing */;
22        }
23    else
24    {
25        ret = FIFO_IS_NULL;
26    }
27}
28else
29{
30    ret = FIFO_IS_PTR_NULL;
31}
32
33 return ret;
34}
35
```

2.1.5 - FIFO_udtPrintBuffer

This function to print all data in the buffer



```
1 EN_FIFO_status_t FIFO_udtPrintBuffer(ST_FIFO_Buffer_t* PS_copyUddtFifoBuf)
2 {
3     EN_FIFO_status_t ret = FIFO_NOK;
4
5     if((PS_copyUddtFifoBuf != PTR_NULL))
6     {
7         if( (PS_copyUddtFifoBuf->head != PTR_NULL) && (PS_copyUddtFifoBuf->base != PTR_NULL) && (PS_copyUddtFifoBuf->tail != PTR_NULL) )
8         {
9             DATA_WIDTH *temp;
10            uint32 i;
11
12            ret = FIFO_udtIsFull(PS_copyUddtFifoBuf);
13            if(ret == FIFO_IS_EMPTY)
14            {
15                DPRINTF("FIFO Is Empty \n");
16            }
17            else
18            {
19                temp = PS_copyUddtFifoBuf->tail;
20                for(i=0 ; i<PS_copyUddtFifoBuf->count ; i++)
21                {
22                    DPRINTF("Student Roll Number IS : %d\n",temp->u32RollNum
ber);
23                    DPRINTF("Student First Name IS : %s\n",temp->u8FirstName);
24                    DPRINTF("Student Last Name Is : %s\n",temp->u8LastName);
25                    DPRINTF("Student GPA Is : %.2f\n",temp->f32Gpa);
26                    for(uint8 j=0 ; j<5;j++)
27                    {
28                        DPRINTF("Student Course %d Id Is : %d\n",j+1,temp->u32CoursesIdNumber[j]);
29                    }
30                    temp++;
31                }
32            }
33        }
34        else
35        {
36            ret = FIFO_IS_NULL;
37        }
38    }
39    else
40    {
41        ret = FIFO_IS_PTR_NULL;
42    }
43
44    return ret;
45 }
```

2.1.6 - FIFO_udtIsUniqueRollNumber

This Function to check if the roll number is unique or not and return the state



```
● ● ●
1 EN_FIFO_status_t FIFO_udtIsUniqueRollNumber(ST_FIFO_Buffer_t* PS_copyUddtFifoBuf ,
2     uint32 copy_u32IdToCheck)
3 {
4     EN_FIFO_status_t ret = FIFO_NOK;
5
6     if((PS_copyUddtFifoBuf != PTR_NULL))
7     {
8         if( (PS_copyUddtFifoBuf->head != PTR_NULL) && (PS_copyUddtFifoBuf->base != PTR_NULL) && (PS_copyUddtFifoBuf->tail != PTR_NULL) )
9         {
10             ret = FIFO_udtIsFull(PS_copyUddtFifoBuf);
11             if(ret == FIFO_IS_EMPTY)
12             {
13                 DPRINTF("FIFO Is Empty \n");
14             }
15             else
16             {
17                 DATA_WIDTH *temp;
18                 uint8 u8IdFoundFlag = 0;
19                 uint32 i;
20                 temp = PS_copyUddtFifoBuf->tail;
21                 for(i=0 ; i<PS_copyUddtFifoBuf->count ; i++)
22                 {
23                     if(temp->u32RollNumber == copy_u32IdToCheck)
24                     {
25                         u8IdFoundFlag = 1;
26                         break;
27                     }
28                     temp++;
29                 }
30                 if(u8IdFoundFlag == 1)
31                 {
32                     ret = FIFO_ID_NOT_UNIQUE;
33                 }
34                 else
35                 {
36                     ret = FIFO_ID_IS_UNIQUE;
37                 }
38             }
39             else
40             {
41                 ret = FIFO_IS_NULL;
42             }
43         }
44         else
45         {
46             ret = FIFO_IS_PTR_NULL;
47         }
48     }
49     return ret;
50 }
```

2.1.7 - FIFO_udtShowDbStatistics

This function to print the count of student , count of free places to add new students and total size of the database

```
● ● ●

1 EN_FIFO_status_t FIFO_udtShowDbStatistics(ST_FIFO_Buffer_t* PS_copyUddtFifoBuf)
2 {
3     EN_FIFO_status_t ret = FIFO_NOK;
4
5     if((PS_copyUddtFifoBuf != PTR_NULL))
6     {
7         if( (PS_copyUddtFifoBuf->head != PTR_NULL) && (PS_copyUddtFifoBuf->base != PTR_NULL) && (PS_copyUddtFifoBuf->tail != PTR_NULL) )
8         {
9             ret = FIFO_udtIsFull(PS_copyUddtFifoBuf);
10            if(ret == FIFO_IS_EMPTY)
11            {
12                DPRINTF("FIFO Is Empty");
13            }
14            else
15            {
16                DPRINTF("[INFO] The Total Number Of Student Is %d \n",PS_copyUddtFifoBuf->count);
17                DPRINTF("[INFO] You Can Add Up To %d Student \n",PS_copyUddtFifoBuf->length);
18                DPRINTF("[INFO] You Can Add %d More Student \n",PS_copyUddtFifoBuf->length - PS_copyUddtFifoBuf->count);
19                ret = FIFO_OK;
20            }
21        }
22        else
23        {
24            ret = FIFO_IS_NULL;
25        }
26    }
27    else
28    {
29        ret = FIFO_IS_PTR_NULL;
30    }
31
32    return ret;
33 }
```

2.1.8 - FIFO_udtFindByRollNumber

This function search in database to find a student by roll number



```
1 EN_FIFO_status_t FIFO_udtFindByRollNumber(ST_FIFO_Buffer_t* PS_copyUddtFifoBuf , u
2 int32 copy_u32IdToCheck)
3 {
4     EN_FIFO_status_t ret = FIFO_NOK;
5
6     if((PS_copyUddtFifoBuf != PTR_NULL))
7     {
8         if( (PS_copyUddtFifoBuf->head != PTR_NULL) && (PS_copyUddtFifoBuf->base != PTR_NULL) && (PS_copyUddtFifoBuf->tail != PTR_NULL) )
9         {
10            ret = FIFO_udtIsFull(PS_copyUddtFifoBuf);
11            if(ret == FIFO_IS_EMPTY)
12            {
13                DPRINTF("FIFO Is Empty\n");
14            }
15            else
16            {
17                DATA_WIDTH *temp;
18                uint8 u8IdFoundFlag = 0;
19                uint32 i;
20                temp = PS_copyUddtFifoBuf->tail;
21                for(i=0 ; i<PS_copyUddtFifoBuf->count ; i++)
22                {
23                    if(temp->u32RollNumber == copy_u32IdToCheck)
24                    {
25                        DPRINTF("Student Roll Number IS : %d\n",temp->u32Rol
26 tName);
27                        DPRINTF("Student First Name IS : %s\n",temp->u8Firs
28 Name);
29                        DPRINTF("Student Last Name Is : %s\n",temp->u8Last
30 pa);
31                        for(uint8 j=0 ; j<5;j++)
32                        {
33                            DPRINTF("Student Course %d Id Is : %d\n",j+1,temp->u32C
34 oursesIdNumber[j]);
35                        }
36                    u8IdFoundFlag = 1;
37                    break;
38                }else{/* Do Nothing */}
39                temp++;
40            }
41            if(u8IdFoundFlag == 0)
42            {
43                DPRINTF("[ERROR] Roll Number %d Not Found \n",copy_u32IdToChec
44 k);
45            }
46            ret = FIFO_OK;
47        }
48    }
49    else
50    {
51        ret = FIFO_IS_NULL;
52    }
53}
54 return ret;
55 }
```

2.1.9 - FIFO_udtFindByName

This function search in database to find a student by first name



```
1 EN_FIFO_status_t FIFO_udtFindByName(ST_FIFO_Buffer_t* PS_copyUdtFifoBuf, uint8* copy_u8FirstNameToCheck)
2 {
3     EN_FIFO_status_t ret = FIFO_NOK;
4
5     if (PS_copyUdtFifoBuf != PTR_NULL)
6     {
7         if (PS_copyUdtFifoBuf->head != PTR_NULL && PS_copyUdtFifoBuf->base != PTR_NULL && PS_copyUdtFifoBuf->tail != PTR_NULL)
8         {
9             ret = FIFO_udtIsFull(PS_copyUdtFifoBuf);
10            if (ret == FIFO_IS_EMPTY)
11            {
12                DPRINT("FIFO Is Empty\n");
13            }
14            else
15            {
16                DATA_WIDTH *temp;
17                uint8 u8NameFoundFlag = 0;
18                uint32 i;
19                temp = PS_copyUdtFifoBuf->tail;
20
21                for (i = 0; i < PS_copyUdtFifoBuf->count; i++)
22                {
23                    if (strcmp((char*)temp->u8FirstName,(char *)copy_u8FirstNameToCheck) == 0)
24                    {
25                        DPRINT("Student Roll Number IS : %d\n", temp->u32RollNumber);
26                        DPRINT("Student First Name IS : %s\n", temp->u8FirstName);
27                        DPRINT("Student Last Name Is : %s\n", temp->u8LastName);
28                        DPRINT("Student GPA Is : %.2f\n", temp->f32Gpa);
29
30                        for (uint8 j = 0; j < 5; j++)
31                        {
32                            DPRINT("Student Course %d Id Is : %d\n", j + 1, temp->u32CoursesIdNumber[j]);
33                        }
34                        u8NameFoundFlag = 1;
35                    }
36                    else
37                    {
38                        /* Do Nothing */
39                    };
40                    temp++;
41                }
42                if (u8NameFoundFlag == 0)
43                {
44                    DPRINT("[ERROR] First Name %s Not Found\n", copy_u8FirstNameToCheck);
45                }
46
47                ret = FIFO_OK;
48            }
49        }
50        else
51        {
52            ret = FIFO_IS_NULL;
53        }
54    }
55    else
56    {
57        ret = FIFO_IS_PTR_NULL;
58    }
59
60    return ret;
61 }
```

2.1.10 - FIFO_udtFindByCourseId

This function print all student info that is register in this course id

```
● ● ●
1  EN_FIFO_status_t FIFO_udtFindByCourseId(ST_FIFO_Buffer_t* PS_copyUddtFifoBuf , uint32 copy_u32IdToCheck)
2  {
3      EN_FIFO_status_t ret = FIFO_NOK;
4
5      if((PS_copyUddtFifoBuf != PTR_NULL))
6      {
7          if( (PS_copyUddtFifoBuf->head != PTR_NULL) && (PS_copyUddtFifoBuf->base != PTR_NULL) && (PS_copyUddtFifoBuf->tail != PTR_NULL) )
8          {
9              ret = FIFO_udtIsFull(PS_copyUddtFifoBuf);
10             if(ret == FIFO_IS_EMPTY)
11             {
12                 DPRINTF("FIFO Is Empty\n");
13             }
14             else
15             {
16                 DATA_WIDTH *temp;
17                 uint8 u8IdFoundFlag = 0;
18                 uint32 i,j,studentCount=0;
19                 temp = PS_copyUddtFifoBuf->tail;
20                 for(i=0 ; i<PS_copyUddtFifoBuf->count ; i++)
21                 {
22                     for(j=0;j<5;j++)
23                     {
24                         if(temp->u32CoursesIdNumber[j] == copy_u32IdToCheck)
25                         {
26                             DPRINTF("The Student Details Is : \n");
27                             DPRINTF("Student Roll Number IS : %d\n",temp->u32RollNumber);
28                             DPRINTF("Student First Name IS : %s\n",temp->u8FirstName);
29                             DPRINTF("Student Last Name IS : %s\n",temp->u8LastName);
30                             DPRINTF("Student GPA Is : %.2f\n",temp->f32Gpa);
31                             u8IdFoundFlag = 1;
32                             studentCount++;
33                             break;
34                         }else{/* Do Nothing */}
35                     }
36                     temp++;
37                 }
38                 if(u8IdFoundFlag == 0)
39                 {
40                     DPRINTF("[ERROR] Course Id %d Not Found \n",copy_u32IdToCheck);
41                 }
42                 else
43                 {
44                     DPRINTF("[INFO] Total Number Of Student Enrolled Is : %d\n",studentCount);
45                 }
46                 ret = FIFO_OK;
47             }
48         }
49         else
50         {
51             ret = FIFO_IS_NULL;
52         }
53     }
54     else
55     {
56         ret = FIFO_IS_PTR_NULL;
57     }
58 }
59 return ret;
60 }
```

2.1.11 - FIFO_udtDeleteByRollNumber

This function search in database to find a student by roll number then it will delete this student



```
1 EN_FIFO_status_t FIFO_udtDeleteByRollNumber(ST_FIFO_Buffer_t* PS_copyUddtFifoBuf ,
2 {
3     EN_FIFO_status_t ret = FIFO_NOK;
4
5     if((PS_copyUddtFifoBuf != PTR_NULL))
6     {
7         if( (PS_copyUddtFifoBuf->head != PTR_NULL) && (PS_copyUddtFifoBuf->base != PTR_NULL) && (PS_copyUddtFifoBuf->tail != PTR_NULL) )
8         {
9             ret = FIFO_udtIsFull(PS_copyUddtFifoBuf);
10            if(ret == FIFO_IS_EMPTY)
11            {
12                DPRINTF("FIFO Is Empty\n");
13            }
14            else
15            {
16                DATA_WIDTH *temp;
17                uint8 u8IdFoundFlag = 0;
18                uint32 i;
19                temp = PS_copyUddtFifoBuf->tail;
20                for(i=0 ; i<PS_copyUddtFifoBuf->count ; i++)
21                {
22                    if(temp->u32RollNumber == copy_u32IdToCheck)
23                    {
24                        u8IdFoundFlag = 1;
25                        break;
26                    }/* Do Nothing */;
27                    temp++;
28                }
29                if(u8IdFoundFlag == 0)
30                {
31                    DPRINTF("[ERROR] Roll Number %d Not Found \n",copy_u32IdToChec
k);
32                }
33                else
34                {
35                    for(;i<PS_copyUddtFifoBuf->count ; i++)
36                    {
37                        *temp = *(temp + 1);
38                        temp++;
39                    }
40                    PS_copyUddtFifoBuf->count--;
41                    DPRINTF("[INFO] The Roll Number %d Is Removed Successfully \n",
42 copy_u32IdToCheck);
43                }
44            }
45        }
46        else
47        {
48            ret = FIFO_IS_NULL;
49        }
50    }
51    else
52    {
53        ret = FIFO_IS_PTR_NULL;
54    }
55
56    return ret;
57 }
```

2.1.12 - FIFO_udtUpdateRollNumber

This function search in database to find a student by roll number then it will update roll number of this student

```
1 EN_FIFO_status_t FIFO_udtUpdateRollNumber(ST_FIFO_Buffer_t* PS_copyUddtFifoBuf , u
2     int32 copy_u32IdToCheck , uint32 copy_u32NewRollNumber)
3 {
4     EN_FIFO_status_t ret = FIFO_NOK;
5
6     if((PS_copyUddtFifoBuf != PTR_NULL))
7     {
8         if( (PS_copyUddtFifoBuf->head != PTR_NULL) && (PS_copyUddtFifoBuf->base !=
9             PTR_NULL) && (PS_copyUddtFifoBuf->tail != PTR_NULL) )
10        {
11            if(FIFO_udtIsFull(PS_copyUddtFifoBuf);
12                if(ret == FIFO_IS_EMPTY)
13                {
14                    DPRINTF("FIFO Is Empty\n");
15                }
16            else
17            {
18                DATA_WIDTH *temp;
19                uint8 u8IdFoundFlag = 0;
20                uint32 i;
21                temp = PS_copyUddtFifoBuf->tail;
22                for(i=0 ; i<PS_copyUddtFifoBuf->count ; i++)
23                {
24                    if(temp->u32RollNumber == copy_u32IdToCheck)
25                    {
26                        temp->u32RollNumber = copy_u32NewRollNumber;
27                        u8IdFoundFlag = 1;
28                        break;
29                    }else/* Do Nothing */;
30                    temp++;
31                }
32                if(u8IdFoundFlag == 0)
33                {
34                    DPRINTF("[ERROR] Roll Number %d Not Found \n",copy_u32IdToChec
35 k);
36                }
37            }
38            ret = FIFO_OK;
39        }
40    }
41    else
42    {
43        ret = FIFO_IS_NULL;
44    }
45}
46
47 return ret;
48 }
```

2.1.13 - FIFO_udtUpdateFirstName

This function search in database to find a student by roll number then it will update first name of this student



```
1 EN_FIFO_status_t FIFO_udtUpdateFirstName(ST_FIFO_Buffer_t* PS_copyUddtFifoBuf , ui
nt32 copy_u32IdToCheck , uint8 *copy_u8NewFirstName)
2 {
3     EN_FIFO_status_t ret = FIFO_NOK;
4
5     if((PS_copyUddtFifoBuf != PTR_NULL))
6     {
7         if( (PS_copyUddtFifoBuf->head != PTR_NULL) && (PS_copyUddtFifoBuf->base != PTR_NULL) && (PS_copyUddtFifoBuf->tail != PTR_NULL) )
8         {
9             ret = FIFO_udtIsFull(PS_copyUddtFifoBuf);
10            if(ret == FIFO_IS_EMPTY)
11            {
12                DPRINTF("FIFO Is Empty\n");
13            }
14            else
15            {
16                DATA_WIDTH *temp;
17                uint8 u8IdFoundFlag = 0;
18                uint32 i;
19                temp = PS_copyUddtFifoBuf->tail;
20                for(i=0 ; i<PS_copyUddtFifoBuf->count ; i++)
21                {
22                    if(temp->u32RollNumber == copy_u32IdToCheck)
23                    {
24                        strcpy((char *)temp->u8FirstName, (char *)copy_u8NewFirstName);
25                        u8IdFoundFlag = 1;
26                        break;
27                    }else{/* Do Nothing */};
28                    temp++;
29                }
30                if(u8IdFoundFlag == 0)
31                {
32                    DPRINTF("[ERROR] Roll Number %d Not Found \n",copy_u32IdToCheck);
33                }
34                ret = FIFO_OK;
35            }
36        }
37        else
38        {
39            ret = FIFO_IS_NULL;
40        }
41    }
42    else
43    {
44        ret = FIFO_IS_PTR_NULL;
45    }
46
47    return ret;
48 }
```

2.1.14 - FIFO_udtUpdateLastName

This function search in database to find a student by roll number then it will update last name of this student



```
1 EN_FIFO_status_t FIFO_udtUpdateLastName(ST_FIFO_Buffer_t* PS_copyUddtFifoBuf , uint32 copy_u32IdToCheck , uint8 *copy_u8NewLasttName)
2 {
3     EN_FIFO_status_t ret = FIFO_NOK;
4
5     if((PS_copyUddtFifoBuf != PTR_NULL))
6     {
7         if( (PS_copyUddtFifoBuf->head != PTR_NULL) && (PS_copyUddtFifoBuf->base != PTR_NULL) && (PS_copyUddtFifoBuf->tail != PTR_NULL) )
8         {
9             ret = FIFO_udtIsFull(PS_copyUddtFifoBuf);
10            if(ret == FIFO_IS_EMPTY)
11            {
12                DPRINTF("FIFO Is Empty\n");
13            }
14            else
15            {
16                DATA_WIDTH *temp;
17                uint8 u8IdFoundFlag = 0;
18                uint32 i;
19                temp = PS_copyUddtFifoBuf->tail;
20                for(i=0 ; i<PS_copyUddtFifoBuf->count ; i++)
21                {
22                    if(temp->u32RollNumber == copy_u32IdToCheck)
23                    {
24                        strcpy((char *)temp->u8LastName, (char *)copy_u8NewLasttName);
25                        u8IdFoundFlag = 1;
26                        break;
27                    }else{/* Do Nothing */}
28                    temp++;
29                }
30                if(u8IdFoundFlag == 0)
31                {
32                    DPRINTF("[ERROR] Roll Number %d Not Found \n",copy_u32IdToChec
k);
33                }
34                ret = FIFO_OK;
35            }
36        }
37        else
38        {
39            ret = FIFO_IS_NULL;
40        }
41    }
42    else
43    {
44        ret = FIFO_IS_PTR_NULL;
45    }
46
47    return ret;
48 }
```

2.1.15 - FIFO_udtUpdateGpa

This function search in database to find a student by roll number then it will update Gpa of this student



```
1 EN_FIFO_status_t FIFO_udtUpdateGpa(ST_FIFO_Buffer_t* PS_copyUddtFifoBuf , uint32 copy_u32IdToCheck , float32 copy_f32NewGpa)
2 {
3     EN_FIFO_status_t ret = FIFO_NOK;
4
5     if((PS_copyUddtFifoBuf != PTR_NULL))
6     {
7         if( (PS_copyUddtFifoBuf->head != PTR_NULL) && (PS_copyUddtFifoBuf->base != PTR_NULL) && (PS_copyUddtFifoBuf->tail != PTR_NULL) )
8         {
9             ret = FIFO_udtIsFull(PS_copyUddtFifoBuf);
10            if(ret == FIFO_IS_EMPTY)
11            {
12                DPRINTF("FIFO Is Empty\n");
13            }
14            else
15            {
16                DATA_WIDTH *temp;
17                uint8 u8IdFoundFlag = 0;
18                uint32 i;
19                temp = PS_copyUddtFifoBuf->tail;
20                for(i=0 ; i<PS_copyUddtFifoBuf->count ; i++)
21                {
22                    if(temp->u32RollNumber == copy_u32IdToCheck)
23                    {
24                        temp->f32Gpa = copy_f32NewGpa;
25                        u8IdFoundFlag = 1;
26                        break;
27                    }/* Do Nothing */;
28                    temp++;
29                }
30                if(u8IdFoundFlag == 0)
31                {
32                    DPRINTF("[ERROR] Roll Number %d Not Found \n",copy_u32IdToCheck);
33                }
34                ret = FIFO_OK;
35            }
36        }
37        else
38        {
39            ret = FIFO_IS_NULL;
40        }
41    }
42    else
43    {
44        ret = FIFO_IS_PTR_NULL;
45    }
46
47    return ret;
48 }
```

2.1.16 - FIFO_udtUpdateCourses

This function search in database to find a student by roll number then it will update courses id of this student



```
1 EN_FIFO_status_t FIFO_udtUpdateCourses(ST_FIFO_Buffer_t* PS_copyUddtFifoBuf , uint
2 copy_u32IdToCheck ,uint32 *copy_u32NewCourses)
3 {
4     EN_FIFO_status_t ret = FIFO_NOK;
5
6     if((PS_copyUddtFifoBuf != PTR_NULL))
7     {
8         if( (PS_copyUddtFifoBuf->head != PTR_NULL) && (PS_copyUddtFifoBuf->base !=
9 PTR_NULL) && (PS_copyUddtFifoBuf->tail != PTR_NULL) )
10        {
11            if( FIFO_udtIsFull(PS_copyUddtFifoBuf);
12            if(ret == FIFO_IS_EMPTY)
13            {
14                DPRINTF("FIFO Is Empty\n");
15            }
16            else
17            {
18                DATA_WIDTH *temp;
19                uint8 u8IdFoundFlag = 0;
20                uint32 i;
21                temp = PS_copyUddtFifoBuf->tail;
22                for(i=0 ; i<PS_copyUddtFifoBuf->count ; i++)
23                {
24                    if(temp->u32RollNumber == copy_u32IdToCheck)
25                    {
26                        // Copy the values from u32ArrCourses to arrOfCourses
27                        for (uint8 i = 0; i < 5; i++)
28                        {
29                            temp->u32CoursesIdNumber[i] = copy_u32NewCourses[i];
30                        }
31                        u8IdFoundFlag = 1;
32                        break;
33                    }
34                }
35                if(u8IdFoundFlag == 0)
36                {
37                    DPRINTF("[ERROR] Roll Number %d Not Found \n",copy_u32IdToChec
38                }
39            }
40        }
41    }
42    else
43    {
44        ret = FIFO_IS_NULL;
45    }
46    else
47    {
48        ret = FIFO_IS_PTR_NULL;
49    }
50
51    return ret;
52 }
```

2.1.17 - FIFO_udtIsValidRollNumber

This Function responsible to check if this roll number is exist in database



```
1 EN_FIFO_status_t FIFO_udtIsValidRollNumber(ST_FIFO_Buffer_t* PS_copyUddtFifoBuf ,
2     uint32 copy_u32IdToCheck)
3 {
4     EN_FIFO_status_t ret = FIFO_NOK;
5
6     if((PS_copyUddtFifoBuf != PTR_NULL))
7     {
8         if( (PS_copyUddtFifoBuf->head != PTR_NULL) && (PS_copyUddtFifoBuf->base != PTR_NULL) && (PS_copyUddtFifoBuf->tail != PTR_NULL) )
9         {
10             ret = FIFO_udtIsFull(PS_copyUddtFifoBuf);
11             if(ret == FIFO_IS_EMPTY)
12             {
13                 DPRINTF("FIFO Is Empty\n");
14             }
15             else
16             {
17                 DATA_WIDTH *temp;
18                 uint32 i;
19                 temp = PS_copyUddtFifoBuf->tail;
20                 for(i=0 ; i<PS_copyUddtFifoBuf->count ; i++)
21                 {
22                     if(temp->u32RollNumber == copy_u32IdToCheck)
23                     {
24                         ret = FIFO_ID_VALID;
25                         break;
26                     }else{/* Do Nothing */};
27                     temp++;
28                 }
29             }
30         }
31     }
32     else
33     {
34         ret = FIFO_IS_NULL;
35     }
36 }
37     ret = FIFO_IS_PTR_NULL;
38 }
39
40 return ret;
41 }
```

2.2 - Student Functions Implementation

```
1  /*****  
2  // Author      : Sherif Ashraf Khedr  
3  // Project     : Student_Information_Management  
4  // File        : studentDbOptions.h  
5  // Date        : Dec 12, 2023  
6  // GitHub       : https://github.com/sherifkhadr  
7  *****/  
8  
9  
10 #ifndef STUDENTDBOPTIONS_H_  
11 #define STUDENTDBOPTIONS_H_  
12 #include "stdTypes.h"  
13  
14  
15 typedef struct  
16 {  
17     uint8 u8FirstName[50];  
18     uint8 u8LastName[50];  
19     uint32 u32RollNumber;  
20     float32 f32Gpa;  
21     uint32 u32CoursesIdNumber[5];  
22 }ST_studentInfo_t;
```



```
1 ST_FIFO_Buffer_t uddtStudentsBuffer;  
2 ST_studentInfo_t uddtStudentDb[50];  
3
```

2.2.1 - ConvertToLower

This function convert string to lower to make any string input or output form database in lower case



The screenshot shows a terminal window with three colored tabs at the top: red, yellow, and green. The terminal displays the following C code:

```
1 static void convertToLower(char *str) {  
2     while (*str) {  
3         *str = (*str >= 'A' && *str <= 'Z') ? (*str + ('a' - 'A')) : *str;  
4         str++;  
5     }  
6 }
```

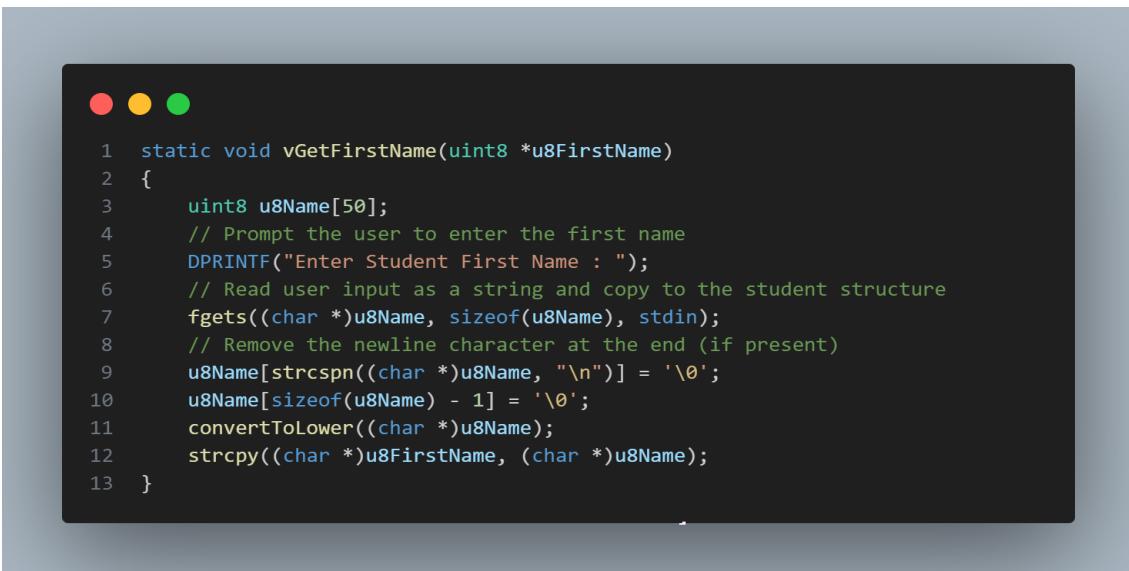
2.2.2 - u32GetRollNumber

This function to take the roll number from user and check that it is in a true formula

```
1 static uint32 u32GetRollNumber(void)
2 {
3     // Declare a temporary input buffer
4     uint8 u8TempInput[50];
5     uint32 u32RollNumber = 0;
6     // Declare a variable to store the result of FIFO functions
7     EN_FIFO_status_t FifoRet = FIFO_NOK;
8     // Prompt the user to enter the student's roll number
9     DPRINTF("Enter Student Roll Number : ");
10
11    // Loop until valid roll number is entered
12    while (1) {
13        // Read user input as a string
14        fgets((char *)u8TempInput, sizeof(u8TempInput), stdin);
15
16        // Convert the string to an integer
17        char *endptr;
18        u32RollNumber = (uint32)strtol((char *)u8TempInput, &endptr, 10);
19
20        // Check if the conversion was successful and roll number is unique
21        if (*endptr == '\0' || *endptr == '\n') {
22            FifoRet = FIFO_uddtIsUniqueRollNumber(&uddtStudentsBuffer,u32Rol
23            lNumber);
24            if (FifoRet == FIFO_ID_IS_UNIQUE) {
25                break; // Exit the loop if roll number is valid and unique
26            } else {
27                DPRINTF("[ERROR] Roll Number Is Not Unique. Try Again With a
28                Different Roll Number.\n");
29            }
30        } else {
31            DPRINTF("[ERROR] Invalid Roll Number. Please enter a valid integ
32            er.\n");
33        }
34    }
35    return u32RollNumber;
36 }
```

2.2.3 - vGetFirstName

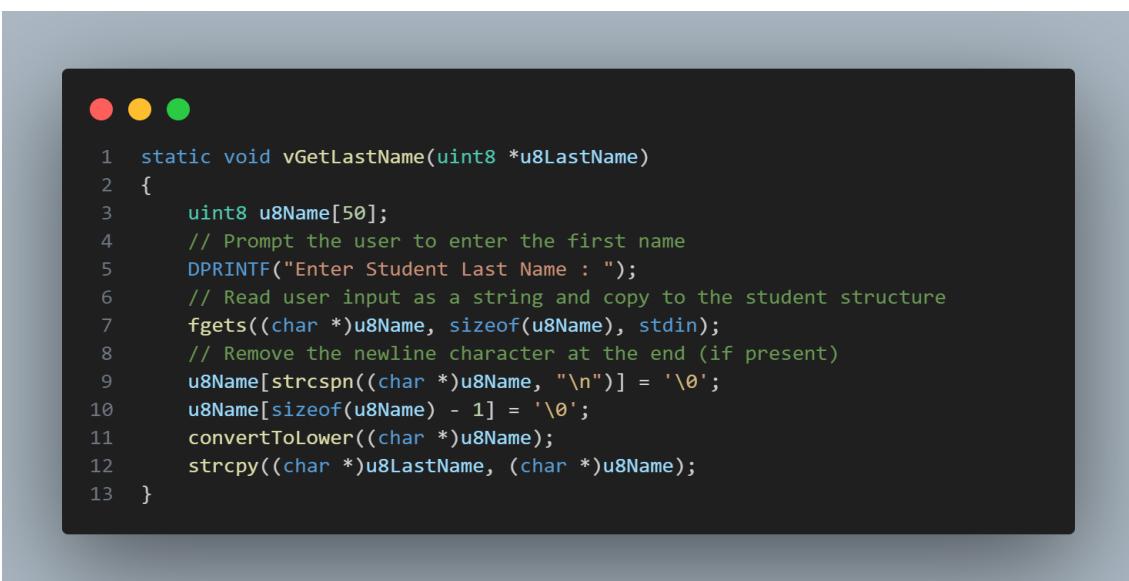
This function to take the first name from user and check that it is in a true formula



```
1 static void vGetFirstName(uint8 *u8FirstName)
2 {
3     uint8 u8Name[50];
4     // Prompt the user to enter the first name
5     DPRINTF("Enter Student First Name : ");
6     // Read user input as a string and copy to the student structure
7     fgets((char *)u8Name, sizeof(u8Name), stdin);
8     // Remove the newline character at the end (if present)
9     u8Name[strcspn((char *)u8Name, "\n")] = '\0';
10    u8Name[sizeof(u8Name) - 1] = '\0';
11    convertToLower((char *)u8Name);
12    strcpy((char *)u8FirstName, (char *)u8Name);
13 }
```

2.2.4 - vGetLastName

This function to take the last name from user and check that it is in a true formula



```
1 static void vGetLastName(uint8 *u8LastName)
2 {
3     uint8 u8Name[50];
4     // Prompt the user to enter the first name
5     DPRINTF("Enter Student Last Name : ");
6     // Read user input as a string and copy to the student structure
7     fgets((char *)u8Name, sizeof(u8Name), stdin);
8     // Remove the newline character at the end (if present)
9     u8Name[strcspn((char *)u8Name, "\n")] = '\0';
10    u8Name[sizeof(u8Name) - 1] = '\0';
11    convertToLower((char *)u8Name);
12    strcpy((char *)u8LastName, (char *)u8Name);
13 }
```

2.2.5 - f32GetGpa

This function to take the gpa from user and check that it is in a true formula



```
1 static float32 f32GetGpa(void)
2 {
3     // Declare a temporary input buffer
4     uint8 u8TempInput[50];
5     float32 u32Gpa = 0.0;
6     // Prompt the user to enter the GPA
7     DPRINTF("Enter Student GPA : ");
8
9     // Loop until a valid GPA is entered
10    while (1) {
11        // Read user input as a string and convert to float
12        fgets((char *)u8TempInput, sizeof(u8TempInput), stdin);
13        char *endptr;
14        u32Gpa = strtod((char *)u8TempInput, &endptr);
15
16        // Check if the conversion was successful
17        if (*endptr == '\0' || *endptr == '\n') {
18            break; // Exit the loop if GPA is valid
19        } else {
20            DPRINTF("[ERROR] Invalid GPA. Please enter a valid float.\n");
21        }
22    }
23    return u32Gpa;
24 }
25
```

2.2.6 - vGetCoursesId

This function to take the courses id from user and check that it is in a true formula

```
1 static void vGetCoursesId(uint32 *arrOfCourses)
2 {
3     // Declare a temporary input buffer
4     uint8 u8TempInput[50];
5     uint32 u32ArrCourses[5];
6     // Loop to get course codes from the user
7     for (uint8 i = 0; i < 5; i++) {
8         DPRINTF("Enter Course %d Id : ", i + 1);
9
10        // Loop until a valid course code is entered
11        while (1) {
12            // Read user input as a string and convert to integer
13            fgets((char *)u8TempInput, sizeof(u8TempInput), stdin);
14            char *endptr;
15            u32ArrCourses[i] = (uint32)strtol((char *)u8TempInput, &endptr,
16            10);
17
18            // Check if the conversion was successful
19            if (*endptr == '\0' || *endptr == '\n') {
20                break; // Exit the loop if course code is valid
21            } else {
22                DPRINTF("[ERROR] Invalid Course Code. Please enter a valid i
23                nteger.\n");
24            }
25        }
26        // Copy the values from u32ArrCourses to arrOfCourses
27        for (uint8 i = 0; i < 5; i++)
28        {
29            arrOfCourses[i] = u32ArrCourses[i];
30        }
31    }
32 }
```

2.2.7 - STUDENT_vInitDataBase

This function to initialise the database and assign its size to 50 student

```
1 void STUDENT_vInitDataBase(void)
2 {
3     /* initilaize the buffer */
4     FIFO_udtInitBuffer(&uddtStudentsBuffer , uddtStudentDb , 50);
5 }
```

2.2.8 - STUDENT_vAddUserByFile

This Function to add students from text file by automatic way

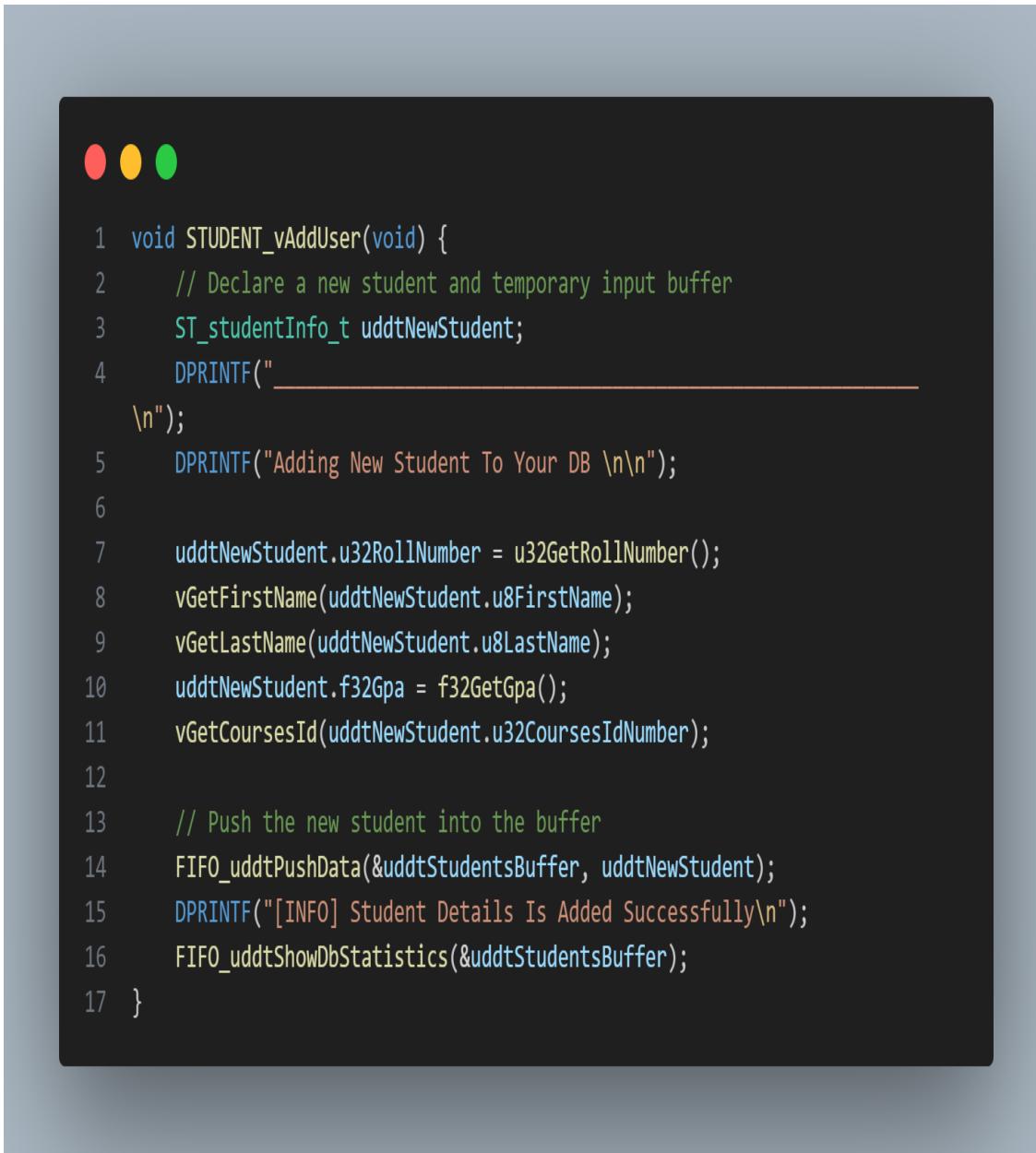


```
● ● ●

1 void STUDENT_vAddUsersByFile(void)
2 {
3     ST_studentInfo_t uddtNewStudent;
4     // Declare a variable to store the result of FIFO functions
5     EN_FIFO_status_t FifoRet = FIFO_NOK;
6     FILE* file = fopen("studentDb.txt", "r");
7     if (!file) {
8         DPRINTF("[ERROR] Unable to open file: %s\n", "studentDb.txt");
9         return;
10    }
11    // Read student information from the file and add to the database
12    while (fscanf(file, "%u %s %s %f %u %u %u %u %u",
13                  &uddtNewStudent.u32RollNumber,
14                  uddtNewStudent.u8FirstName,
15                  uddtNewStudent.u8LastName,
16                  &uddtNewStudent.f32Gpa,
17                  &uddtNewStudent.u32CoursesIdNumber[0],
18                  &uddtNewStudent.u32CoursesIdNumber[1],
19                  &uddtNewStudent.u32CoursesIdNumber[2],
20                  &uddtNewStudent.u32CoursesIdNumber[3],
21                  &uddtNewStudent.u32CoursesIdNumber[4]) == 9) {
22
23        // Add the student to the buffer
24        FifoRet = FIFO_uddtIsUniqueRollNumber(&uddtStudentsBuffer, uddtNewStudent.u32RollNumber);
25        if(FifoRet == FIFO_ID_IS_UNIQUE)
26        {
27            FIFO_uddtPushData(&uddtStudentsBuffer, uddtNewStudent);
28            DPRINTF("[INFO] Student With Roll Number %d Is Added Successfull
y\n", uddtNewStudent.u32RollNumber);
29            FIFO_uddtShowDbStatistics(&uddtStudentsBuffer);
30        }
31        else
32        {
33            DPRINTF("[ERROR] Student With Roll Number %d Is Not Added Taken
Roll Number\n", uddtNewStudent.u32RollNumber);
34        }
35    }
36
37    fclose(file);
38 }
```

2.2.9 - STUDENT_vAddUserByFile

This Function to add student by manually way by entering all student data



```
● ● ●

1 void STUDENT_vAddUser(void) {
2     // Declare a new student and temporary input buffer
3     ST_studentInfo_t uddtNewStudent;
4     DPRINTF("_____
\n");
5     DPRINTF("Adding New Student To Your DB \n\n");
6
7     uddtNewStudent.u32RollNumber = u32GetRollNumber();
8     vGetFirstName(uddtNewStudent.u8FirstName);
9     vGetLastName(uddtNewStudent.u8LastName);
10    uddtNewStudent.f32Gpa = f32GetGpa();
11    vGetCoursesId(uddtNewStudent.u32CoursesIdNumber);
12
13    // Push the new student into the buffer
14    FIFO_udtPushData(&uddtStudentsBuffer, uddtNewStudent);
15    DPRINTF("[INFO] Student Details Is Added Successfully\n");
16    FIFO_udtShowDbStatistics(&uddtStudentsBuffer);
17 }
```

2.2.10 - STUDENT_vFindByRollNumber

This function will take a roll number from user and then print data of this roll number



```
1 void STUDENT_vFindByRollNumber(void)
2 {
3     /* temp array to take user input from keyboard and store it */
4     uint8 u8TempInput[50];
5     /* temp uint32 variable to store integer id */
6     uint32 u32RollNumber = 0;
7     // Prompt the user to enter the student's roll number
8     DPRINTF("Enter Student Roll Number : ");
9
10    // Loop until valid roll number is entered
11    while (1) {
12        // Read user input as a string
13        fgets((char *)u8TempInput, sizeof(u8TempInput), stdin);
14
15        // Convert the string to an integer
16        char *endptr;
17        u32RollNumber = (uint32)strtol((char *)u8TempInput, &endptr, 10);
18
19        // Check if the conversion was successful and roll number is unique
20        if (*endptr == '\0' || *endptr == '\n') {
21            FIFO_uddtFindByRollNumber(&uddtStudentsBuffer ,u32RollNumber);
22            break;
23        }
24        else
25        {
26            DPRINTF("[ERROR] Invalid Roll Number. Please enter a valid integ
er.\n");
27        }
28    }
29 }
```

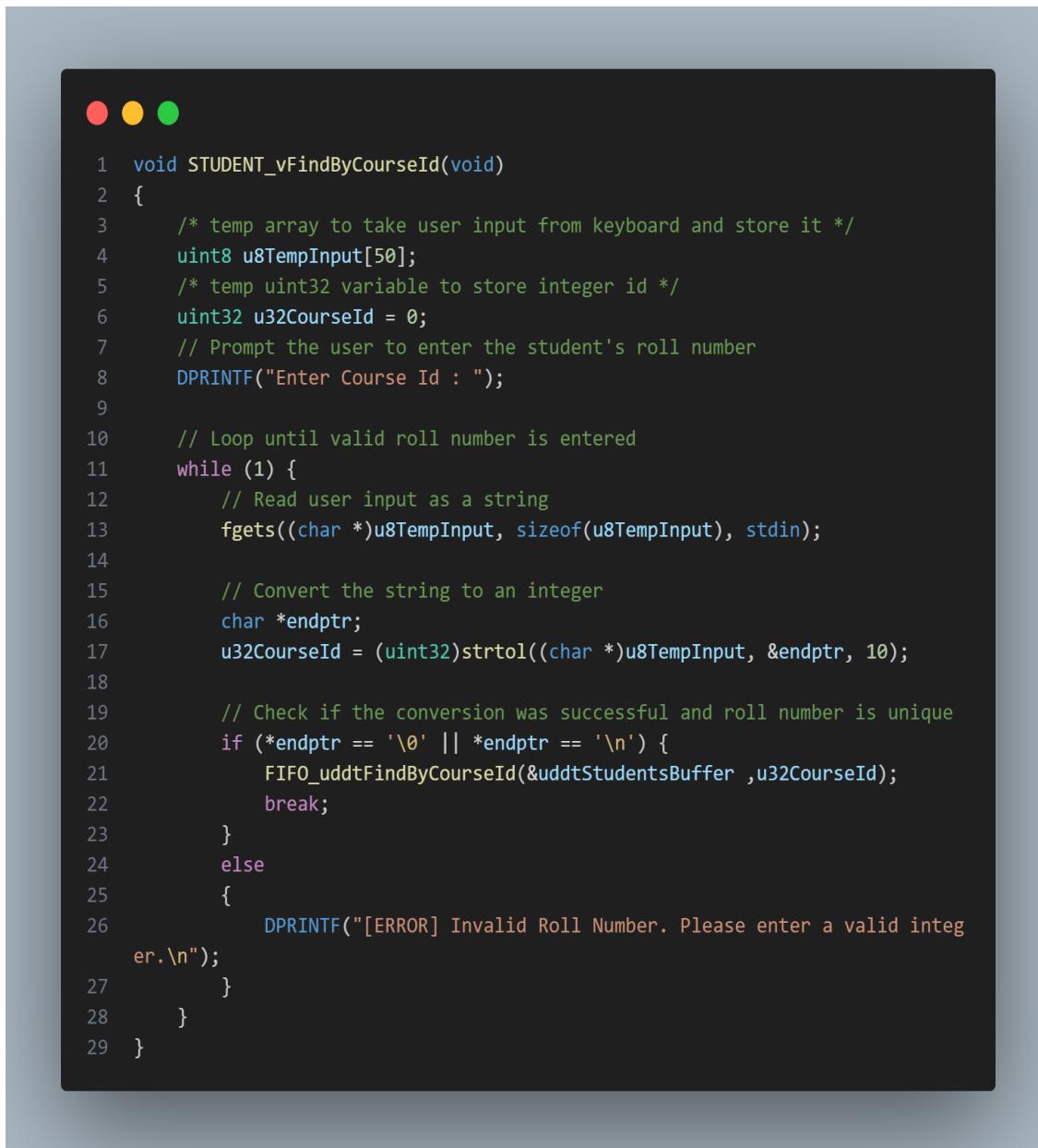
2.2.11 - STUDENT_vFindByFirstName

This function will take a first name from user and then print data of all student with this first name

```
1 void STUDENT_vFindByFirstName(void)
2 {
3     /* temp array to take user input from the keyboard and store it */
4     uint8 u8TempInput[50];
5
6     // Prompt the user to enter the student's first name
7     DPRINTF("Enter Student First Name : ");
8
9     // Loop until a valid first name is entered
10    while (1) {
11        // Read user input as a string
12        fgets((char *)u8TempInput, sizeof(u8TempInput), stdin);
13
14        // Remove the newline character at the end (if present)
15        u8TempInput[strcspn((char *)u8TempInput, "\n")] = '\0';
16        // Copy the input string to the first name variable
17        u8TempInput[sizeof(u8TempInput) - 1] = '\0';
18
19        // Check if the first name is not empty
20        if (strlen((char *)u8TempInput) > 0) {
21            convertToLower((char *)u8TempInput);
22            // Call the function to find and print student details by first
23            name
24            FIFO_udtFindByName(&udtStudentsBuffer, u8TempInput);
25            break;
26        } else {
27            DPRINTF("[ERROR] Invalid First Name. Please enter a non-empty st
28            ring.\n");
29        }
30    }
31 }
```

2.2.12 - STUDENT_vFindBylastName

This function will take a last name from user and then print data of all student with this last name



```
1 void STUDENT_vFindByCourseId(void)
2 {
3     /* temp array to take user input from keyboard and store it */
4     uint8 u8TempInput[50];
5     /* temp uint32 variable to store integer id */
6     uint32 u32CourseId = 0;
7     // Prompt the user to enter the student's roll number
8     DPRINTF("Enter Course Id : ");
9
10    // Loop until valid roll number is entered
11    while (1) {
12        // Read user input as a string
13        fgets((char *)u8TempInput, sizeof(u8TempInput), stdin);
14
15        // Convert the string to an integer
16        char *endptr;
17        u32CourseId = (uint32)strtol((char *)u8TempInput, &endptr, 10);
18
19        // Check if the conversion was successful and roll number is unique
20        if (*endptr == '\0' || *endptr == '\n') {
21            FIFO_uddtFindByCourseId(&uddtStudentsBuffer ,u32CourseId);
22            break;
23        }
24        else
25        {
26            DPRINTF("[ERROR] Invalid Roll Number. Please enter a valid integ
er.\n");
27        }
28    }
29 }
```

2.2.13 - STUDENT_vPrintDbInfo

This function will print the statics of database it will print count of student ,count of free places and count of total students

```
● ● ●  
1 void STUDENT_vPrintDbInfo(void)  
2 {  
3     FIFO_udtShowDbStatistics(&udtStudentsBuffer);  
4 }
```

2.2.14 - STUDENT_vPrintAllStudents

This function will print all students data in the database

```
● ● ●  
1 void STUDENT_vPrintAllStudents(void)  
2 {  
3     FIFO_udtPrintBuffer(&udtStudentsBuffer);  
4 }
```

2.2.15 - STUDENT_vDeleteUser

This function will take roll number from user and then this student will be deleted



```
1 void STUDENT_vDeleteUser(void)
2 {
3     /* temp array to take user input from keyboard and store it */
4     uint8 u8TempInput[50];
5     /* temp uint32 variable to store integer id */
6     uint32 u32RollNumber = 0;
7     // Prompt the user to enter the student's roll number
8     DPRINTF("Enter The Roll Number Which You Want To Delete : ");
9
10    // Loop until valid roll number is entered
11    while (1) {
12        // Read user input as a string
13        fgets((char *)u8TempInput, sizeof(u8TempInput), stdin);
14
15        // Convert the string to an integer
16        char *endptr;
17        u32RollNumber = (uint32)strtol((char *)u8TempInput, &endptr, 10);
18
19        // Check if the conversion was successful and roll number is unique
20        if (*endptr == '\0' || *endptr == '\n') {
21            FIFO_udtDeleteByRollNumber(&udtStudentsBuffer ,u32RollNumber);
22            break;
23        }
24        else
25        {
26            DPRINTF("[ERROR] Invalid Roll Number. Please enter a valid integ
er.\n");
27        }
28    }
29 }
```

2.2.14 - STUDENT_vUpdateUser

This function will take roll number from user and ask user what to update roll number ,first name ,last name ,gpa ,courses id then this student will be updated



```
1 void STUDENT_vUpdateUser(void)
2 {
3     /* temp array to take user input from keyboard and store it */
4     uint8 u8TempInput[50],u8Choice=0;
5     /* temp uint32 variable to store integer id */
6     uint32 u32RollNumber = 0;
7     // Declare a variable to store the result of FIFO functions
8     EN_FIFO_status_t FifoRet = FIFO_NOK;
9     // Prompt the user to enter the student's roll number
10    DPRINTF("Enter Student Roll Number To Update Info : ");
11
12    // Loop until valid roll number is entered
13    while (1) {
14        // Read user input as a string
15        fgets((char *)u8TempInput, sizeof(u8TempInput), stdin);
16
17        // Convert the string to an integer
18        char *endptr;
19        u32RollNumber = (uint32)strtol((char *)u8TempInput, &endptr, 10);
20
21        // Check if the conversion was successful and roll number is unique
22        if (*endptr == '\0' || *endptr == '\n') {
23
24            FifoRet = FIFO_udtIs_ValidRollNumber(&uddtStudentsBuffer,u32Roll
25 Number);
26
27            if(FifoRet == FIFO_ID_VALID)
28            {
29                DPRINTF("1 - Roll Number\n");
30                DPRINTF("2 - First Name\n");
31                DPRINTF("3 - Last Name\n");
32                DPRINTF("4 - GPA\n");
33                DPRINTF("5 - Courses\n");
34                DPRINTF("Enter Any Key To Exit\n");
35                while(1)
36                {
37                    // Read user input as a string
38                    fgets((char *)u8TempInput, sizeof(u8TempInput), stdin);
39                    // Convert the string to an integer
40                    char *endptr;
41                    u8Choice = (uint32)strtol((char *)u8TempInput, &endptr,
42 10);
43
44                    // Check if the conversion was successful
45                    if (*endptr == '\0' || *endptr == '\n')
46                    {
47                        switch(u8Choice)
48                        {
49                            case 1:
50                                FIFO_udtUpdateRollNumber(&uddtStudentsBuffer,u3
51 2RollNumber,u32GetRollNumber());
52                                DPRINTF("[INFO] Student With Roll Number %d Its
53 Roll Number Is Updated\n",u32RollNumber);
54                                break;
55                            }
56                            case 2:
57                            {
58                                uint8 u8TempOfName[50];
59                                vGetFirstName(u8TempOfName);
60                                FIFO_udtUpdateFirstName(&uddtStudentsBuffer,u32
61 RollNumber,u8TempOfName);
62                                DPRINTF("[INFO] Student With Roll Number %d Its
63 First Name Is Updated\n",u32RollNumber);
64                                break;
65                            }
66                        }
67                    }
68                }
69            }
70        }
71    }
72 }
```

```

1          case 3:
2          {
3              uint8 u8TempOfName[50];
4              vGetLastName(u8TempOfName);
5              FIFO_uddtUpdateLastName(&uddtStudentsBuffer,u32R
6          ollNumber,u8TempOfName);
7              DPRINTF("[INFO] Student With Roll Number %d Its
8          Last Name Is Updated\n",u32RollNumber);
9              break;
10         }
11         case 4:
12         {
13             FIFO_uddtUpdateGpa(&uddtStudentsBuffer,u32RollNu
14 mber,f32GetGpa());
15             DPRINTF("[INFO] Student With Roll Number %d Its
16 Gpa Is Updated\n",u32RollNumber);
17             break;
18         }
19         case 5:
20         {
21             uint32 u32TempOfCourses[5];
22             vGetCoursesId(u32TempOfCourses);
23             FIFO_uddtUpdateCourses(&uddtStudentsBuffer,u32Ro
24 llNumber,u32TempOfCourses);
25             DPRINTF("[INFO] Student With Roll Number %d Its
26 Courses Id Is Updated\n",u32RollNumber);
27             break;
28         }
29         default :
30         {
31             DPRINTF("[INFO] Process Is Cancelled \n");
32             break;
33         }
34     }
35     else
36     {
37         break;
38     }
39     else
40     {
41         DPRINTF("[ERROR] Roll Number %d Not Found \n",u32RollNumbe
42 r);
43     }
44     else
45     {
46         DPRINTF("[ERROR] Invalid Roll Number. Please enter a valid integ
47 er.\n");
48     }

```

2.3 - Main Functions Implementation

```
1  ****
2 // Author      : Sherif Ashraf Khedr
3 // Project     : Student_Information_Management
4 // File        : main.c
5 // Date        : Dec 12, 2023
6 // GitHub       : https://github.com/sherifkhadr
7 ****
8
9 #include <stdio.h>
10 #include <stdlib.h>
11 #include <conio.h>
12 #include "studentDbOptions.h"
13
14 int main(void)
15 {
16     uint8 choice;
17     STUDENT_vInitDataBase();
18     DPRINTF("Welcome To The Student Management System\n");
19     DPRINTF("_____\n");
20     while(1)
21     {
22         system("cls");
23         DPRINTF("Choose The Task That You Want To Perform :\n");
24         DPRINTF("1 - Add    The Student Details Manually\n");
25         DPRINTF("2 - Add    The Student Details From Text File\n");
26         DPRINTF("3 - Find   The Student Details By Roll Number\n");
27         DPRINTF("4 - Find   The Student Details By First Name\n");
28         DPRINTF("5 - Find   The Student Details By Course Id\n");
29         DPRINTF("6 - Find   The Total Number Of Students\n");
30         DPRINTF("7 - Delete  The Students Details By Roll Number\n");
31         DPRINTF("8 - Update  The Students Details By Roll Number\n");
32         DPRINTF("9 - Show    All Information\n");
33         DPRINTF("10 - To Exit\n");
34         DPRINTF("Enter Your Choice To Perform The Task :\n");
35         scanf("%d", (int *)&choice);
36         switch(choice){
37             case 1:
38                 system("cls");
39                 STUDENT_vAddUser();
40                 PAUSE();
41                 break;
42             case 2:
43                 system("cls");
44                 STUDENT_vAddUsersByFile();
45                 PAUSE();
46                 break;
```

```
1      case 3:
2          system("cls");
3          STUDENT_vFindByRollNumber();
4          PAUSE();
5          break;
6      case 4:
7          system("cls");
8          STUDENT_vFindByFirstName();
9          PAUSE();
10         break;
11     case 5:
12         system("cls");
13         STUDENT_vFindByCourseId();
14         PAUSE();
15         break;
16     case 6:
17         system("cls");
18         STUDENT_vPrintDbInfo();
19         PAUSE();
20         break;
21     case 7:
22         system("cls");
23         STUDENT_vDeleteUser();
24         PAUSE();
25         break;
26     case 8:
27         system("cls");
28         STUDENT_vUpdateUser();
29         PAUSE();
30         break;
31     case 9:
32         system("cls");
33         STUDENT_vPrintAllStudents();
34         PAUSE();
35         break;
36     case 10:
37         system("cls");
38         DPRINTF("Exiting.....\n\n");
39         exit(0);
40         break;
41     default:
42         DPRINTF("[ERROR] Invalid Choice Try Again..... \n\n");
43         PAUSE();
44         break;
45     }
46 }
47 return 0;
48 }
```

2.4 - Standard Data Type File

```
 1  /*************************************************************************/
 2 // Author      : Sherif Ashraf Khedr
 3 // Project     : Student_Information_Management
 4 // File       : stdTypes.h
 5 // Date        : Dec 12, 2023
 6 // GitHub      : https://github.com/sherifkhadr
 7 /*************************************************************************/
 8
 9
10 #ifndef STDTYPES_H_
11 #define STDTYPES_H_
12
13 #include <stdint.h>
14
15
16 #define DPRINTF(...) {fflush(stdout); \
17                         fflush(stdin); \
18                         printf(__VA_ARGS__); \
19                         fflush(stdout); \
20                         fflush(stdin);}
21
22 #define PAUSE() DPRINTF("\nPress Any Key To Continue.....\n"); \
23 getch();
24
25 typedef int8_t           sint8;
26 typedef uint8_t          uint8;
27
28 typedef int16_t          sint16;
29 typedef uint16_t          uint16;
30
31 typedef int32_t          sint32;
32 typedef uint32_t          uint32;
33
34 typedef int64_t          sint64;
35 typedef uint64_t          uint64;
36
37
38
39 typedef volatile int8_t    vsint8;
40 typedef volatile uint8_t   vuint8;
41
42 typedef volatile int16_t   vsint16;
43 typedef volatile uint16_t  vuint16;
44
45 typedef volatile int32_t   vsint32;
46 typedef volatile uint32_t  vuint32;
47
48 typedef volatile int64_t   vsint64;
49 typedef volatile uint64_t  vuint64;
50
51
52 typedef float             float32;
53 typedef double            float64;
54 typedef long double        float96;
55
56
57 #ifndef bool
58     typedef enum {
59         FALSE = 0,
60         TRUE
61     }bool;
62 #endif
63
64 #ifndef STR_NULL
65 #define STR_NULL     '\0'
66 #endif
67
68 #ifndef PTR_NULL
69 #define PTR_NULL     (void*)0
70 #endif
71
72
73 #endif
74
```

3 - Database Text File



```
1 123 Ahmed Ali 3.5 101 105 203 204 202
2 124 Fatima Khan 4.0 104 201 105 202 203
3 125 Aisha Rahman 2.8 104 201 203 102 105
4 126 Yusuf Abdullah 3.2 102 203 105 201 202
5 127 Mariam Ahmed 3.9 201 202 104 203 105
6 128 Omar Qureshi 2.5 203 201 105 102 202
7 129 Zainab Malik 3.7 202 105 203 101 102
8 130 Bilal Khattak 4.2 201 103 104 105 202
9 131 Sana Arif 2.9 105 101 202 203 204
10 132 Ilyas Hussain 3.6 203 204 201 202 205
11
```

4 - Program Execution

4.1 - Main Menu

```
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 - To Exit

Enter Your Choice To Perform The Task :
```

4.2 - Add User Manual

```
-----  
Adding New Student To Your DB

Enter Student Roll Number : 5
Enter Student First Name : sherif
Enter Student Last Name : ashraf
Enter Student GPA : 3.4
Enter Course 1 Id : 1
Enter Course 2 Id : 2
Enter Course 3 Id : 3
Enter Course 4 Id : 4
Enter Course 5 Id : 5
[INFO] Student Details Is Added Successfully
[INFO] The Total Number Of Student Is 1
[INFO] You Can Add Up To 50 Student
[INFO] You Can Add 49 More Student

Press Any Key To Continue.....
```

4.3 - Add Student Automated

```
[INFO] Student With Roll Number 123 Is Added Successfully
[INFO] The Total Number Of Student Is 1
[INFO] You Can Add Up To 50 Student
[INFO] You Can Add 49 More Student
[INFO] Student With Roll Number 124 Is Added Successfully
[INFO] The Total Number Of Student Is 2
[INFO] You Can Add Up To 50 Student
[INFO] You Can Add 48 More Student
[INFO] Student With Roll Number 125 Is Added Successfully
[INFO] The Total Number Of Student Is 3
[INFO] You Can Add Up To 50 Student
[INFO] You Can Add 47 More Student
[INFO] Student With Roll Number 126 Is Added Successfully
[INFO] The Total Number Of Student Is 4
[INFO] You Can Add Up To 50 Student
[INFO] You Can Add 46 More Student
[INFO] Student With Roll Number 127 Is Added Successfully
[INFO] The Total Number Of Student Is 5
[INFO] You Can Add Up To 50 Student
[INFO] You Can Add 45 More Student
[INFO] Student With Roll Number 128 Is Added Successfully
[INFO] The Total Number Of Student Is 6
[INFO] You Can Add Up To 50 Student
[INFO] You Can Add 44 More Student
[INFO] Student With Roll Number 129 Is Added Successfully
[INFO] The Total Number Of Student Is 7
[INFO] You Can Add Up To 50 Student
[INFO] You Can Add 43 More Student
[INFO] Student With Roll Number 130 Is Added Successfully
[INFO] The Total Number Of Student Is 8
[INFO] You Can Add Up To 50 Student
[INFO] You Can Add 42 More Student
[INFO] Student With Roll Number 131 Is Added Successfully
[INFO] The Total Number Of Student Is 9
[INFO] You Can Add Up To 50 Student
[INFO] You Can Add 41 More Student
[INFO] Student With Roll Number 132 Is Added Successfully
[INFO] The Total Number Of Student Is 10
[INFO] You Can Add Up To 50 Student
[INFO] You Can Add 40 More Student
```

Press Any Key To Continue.....

4.4 - Find By Roll Number

```
Enter Student Roll Number : 5
Student Roll Number IS : 5
Student First Name IS : sherif
Student Last Name Is : ashraf
Student GPA Is : 3.40
Student Course 1 Id Is : 1
Student Course 2 Id Is : 2
Student Course 3 Id Is : 3
Student Course 4 Id Is : 4
Student Course 5 Id Is : 5
```

```
Press Any Key To Continue.....
```

4.5 - Find By First Name

```
Enter Student First Name : sherif
Student Roll Number IS : 5
Student First Name IS : sherif
Student Last Name Is : ashraf
Student GPA Is : 3.40
Student Course 1 Id Is : 1
Student Course 2 Id Is : 2
Student Course 3 Id Is : 3
Student Course 4 Id Is : 4
Student Course 5 Id Is : 5
```

```
Press Any Key To Continue.....
```

4.6 - Find By Course Id

```
Enter Course Id : 105
The Student Details Is      :
Student Roll Number IS      : 123
Student First Name IS       : Ahmed
Student Last Name IS        : Ali
Student GPA IS              : 3.50
The Student Details Is      :
Student Roll Number IS      : 124
Student First Name IS       : Fatima
Student Last Name IS        : Khan
Student GPA IS              : 4.00
The Student Details Is      :
Student Roll Number IS      : 125
Student First Name IS       : Aisha
Student Last Name IS        : Rahman
Student GPA IS              : 2.80
The Student Details Is      :
Student Roll Number IS      : 126
Student First Name IS       : Yusuf
Student Last Name IS        : Abdullah
Student GPA IS              : 3.20
The Student Details Is      :
Student Roll Number IS      : 127
Student First Name IS       : Mariam
Student Last Name IS        : Ahmed
Student GPA IS              : 3.90
The Student Details Is      :
Student Roll Number IS      : 128
Student First Name IS       : Omar
Student Last Name IS        : Qureshi
Student GPA IS              : 2.50
The Student Details Is      :
Student Roll Number IS      : 129
Student First Name IS       : Zainab
Student Last Name IS        : Malik
Student GPA IS              : 3.70
The Student Details Is      :
Student Roll Number IS      : 130
Student First Name IS       : Bilal
Student Last Name IS        : Khattak
Student GPA IS              : 4.20
The Student Details Is      :
Student Roll Number IS      : 131
Student First Name IS       : Sana
Student Last Name IS        : Arif
Student GPA IS              : 2.90
[INFO] Total Number Of Student Enrolled Is : 9
```

Press Any Key To Continue.....

4.7 - Print Count Of Student

```
[INFO] The Total Number Of Student Is 11  
[INFO] You Can Add Up To 50 Student  
[INFO] You Can Add 39 More Student
```

```
Press Any Key To Continue.....
```

4.8 - Delete By Roll Number

```
Enter The Roll Number Which You Want To Delete : 130  
[INFO] The Roll Number 130 Is Removed Successfully
```

```
Press Any Key To Continue.....
```

4.9 - Roll Number Update

```
Enter Student Roll Number To Update Info : 5
1 - Roll Number
2 - First Name
3 - Last Name
4 - GPA
5 - Courses
Enter Any Key To Exit
1
Enter Student Roll Number : 0
[INFO] Student With Roll Number 5 Its Roll Number Is Updated

Press Any Key To Continue....
```

4.10 - First Name Update

```
Enter Student Roll Number To Update Info : 0
1 - Roll Number
2 - First Name
3 - Last Name
4 - GPA
5 - Courses
Enter Any Key To Exit
2
Enter Student First Name : ashraf
[INFO] Student With Roll Number 0 Its First Name Is Updated

Press Any Key To Continue....
```

4.11 - Last Name Update

```
Enter Student Roll Number To Update Info : 0
1 - Roll Number
2 - First Name
3 - Last Name
4 - GPA
5 - Courses
Enter Any Key To Exit
3
Enter Student Last Name : khedr
[INFO] Student With Roll Number 0 Its Last Name Is Updated

Press Any Key To Continue.....
```

4.12 - Gpa Update

```
Enter Student Roll Number To Update Info : 0
1 - Roll Number
2 - First Name
3 - Last Name
4 - GPA
5 - Courses
Enter Any Key To Exit
4
Enter Student GPA : 4.0
[INFO] Student With Roll Number 0 Its Gpa Is Updated

Press Any Key To Continue.....
```

4.13 - Course Id Update

```
Enter Student Roll Number To Update Info : 0
1 - Roll Number
2 - First Name
3 - Last Name
4 - GPA
5 - Courses
Enter Any Key To Exit
5
Enter Course 1 Id : 101
Enter Course 2 Id : 102
Enter Course 3 Id : 103
Enter Course 4 Id : 104
Enter Course 5 Id : 105
[INFO] Student With Roll Number 0 Its Courses Id Is Updated

Press Any Key To Continue....
```

4.14 - Print All Student Data

```
Student Roll Number IS      : 128
Student First Name IS       : Omar
Student Last Name Is        : Qureshi
Student GPA Is              : 2.50
Student Course 1 Id Is : 203
Student Course 2 Id Is : 201
Student Course 3 Id Is : 105
Student Course 4 Id Is : 102
Student Course 5 Id Is : 202
Student Roll Number IS      : 129
Student First Name IS       : Zainab
Student Last Name Is        : Malik
Student GPA Is              : 3.70
Student Course 1 Id Is : 202
Student Course 2 Id Is : 105
Student Course 3 Id Is : 203
Student Course 4 Id Is : 101
Student Course 5 Id Is : 102
Student Roll Number IS      : 131
Student First Name IS       : Sana
Student Last Name Is        : Arif
Student GPA Is              : 2.90
Student Course 1 Id Is : 105
Student Course 2 Id Is : 101
Student Course 3 Id Is : 202
Student Course 4 Id Is : 203
Student Course 5 Id Is : 204
Student Roll Number IS      : 132
Student First Name IS       : Ilyas
Student Last Name Is        : Hussain
Student GPA Is              : 3.60
Student Course 1 Id Is : 203
Student Course 2 Id Is : 204
Student Course 3 Id Is : 201
Student Course 4 Id Is : 202
Student Course 5 Id Is : 205
Student Roll Number IS      : 0
Student First Name IS       : ashraf
Student Last Name Is        : khedr
Student GPA Is              : 4.00
Student Course 1 Id Is : 101
Student Course 2 Id Is : 102
Student Course 3 Id Is : 103
Student Course 4 Id Is : 104
Student Course 5 Id Is : 105
```