

STUDENT MANAGEMENT SYSTEM

IC 1302 – Programming I Level 1 – Semester 1

> Jeyakrishnan Jeyapriya 2021/T/01213

Department of Information and Communication Technology Faculty of Technology University of Colombo

ı

TABLE OF CONTENTS

1. Introduction	1
2. Methodology	2
2.1. Main Menu	3
2.2. Add Student Details	4
2.3. Add Course Details	5
2.4. Display Student Details	6
2.5. Display Course Details	7
2.6. Update Student Details	8-9
2.7. Search Each Student Details	10-11
2.8. Exit the entire program	12
3. Appendix A: The C Program	13-27
4. Reference	28

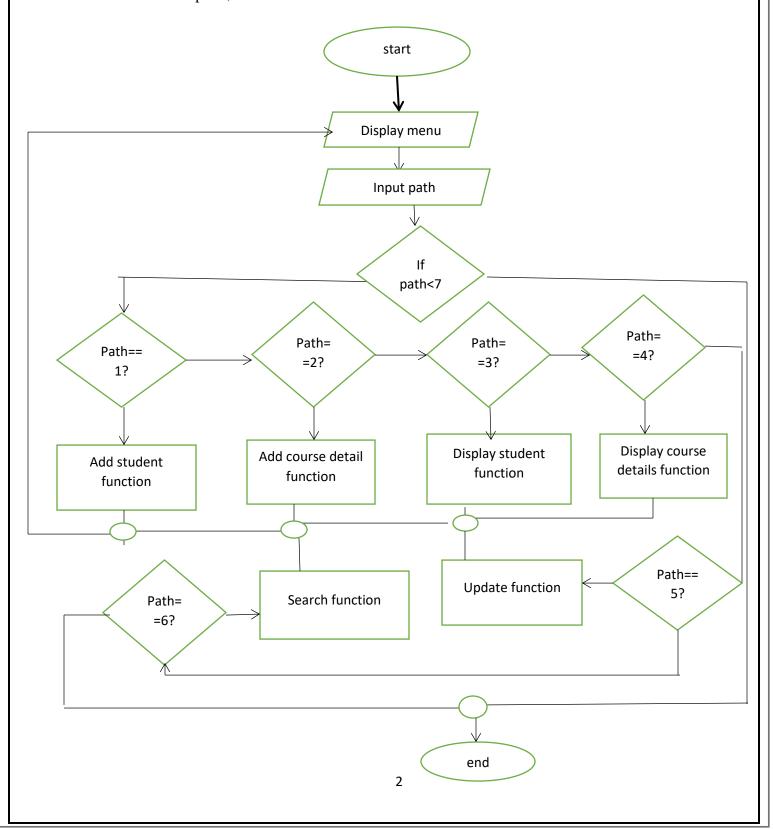
Introduction

This project is creating a Student Management System developed using c language to store information of students. In this console application is a solution tool that is designed to track, maintain and manage all the data generated by a School, University and institutes, including the grades of a student.

The student management system is used to manage the entire student's data. This project is used for making the process of managing information easier and to also make it accessible. To storing information, use binary file as a database in the project. This information could be the general details like student name, department, index number, and their grades for each course which is following or specific information like collection of data. The Student Management System project cover functionalities of add new student details, update student details, enter new course details, find student details and course details and list students information. Also to increase the readability, I have broken the application in different functions. Each function of the project extensively use in the file handing function, so it is also a great project to understand file handling in C.

Methodology

To develop this project mainly used notepad++ to write code used GCC compiler to compile codes. To find some code parts, error solutions to used internet.



Main menu

This is the first screen of the Student Management System. User can choose the menu item by inputting the menu number. Menu will be displayed again and again when a function is accomplished. There are 7 menu items in the menu. User can select the one item in the main menu. Ask the user to select the option. If the user selects 7, then the application will closed.

- 1. ADD NEW STUDENT DETAILS Add to the new student data in the student file
- 2. ADD NEW COURSE DETAILS Add to new course details in the course file
- 3. DISPLAY STUDENT DETAILS Display to the student details in the terminal as a report
- 4. DISPLAY COURSE DETAILS Display the course details in the terminal
- 5. UPDATE STUDENT DETAILS Update the student records
- 6. SEARCH EACH STUDENT DETAILS Search each student details and display in the terminal
- 7. EXIT THE ENTIRE PROGRAM To exit the entire program

```
STUDENT
MANAGEMENT SYSTEM

01. ADD NEW STUDENT DETAILS
02. ADD NEW COURSE DETAILS
03. DISPLAY STUDENT DETAILS
04. DISPLAY COURSE DETAILS
05. UPDATE STUDENT DETAILS
06. SURCH EACH STUDENT DETAILS
07. EXIT THE ENTIRE PROGRAM

ENTER THE PATH :
```

Figure 1: Main Manu

Add Student Details

This function opens the "student.txt" binary file in append mode and writes the student details in the binary file. If you wish to enter another student details input 1 to continue or exit the add student details function press 0.

- 1. Enter the name input the name of the student
- 2. Enter the index number input the student index number in to the system
- 3. Enter the department Enter the department to each student study
- 4. Enter the course count –Input the course count to the system each student following or followed
- 5. Enter the course name- Enter the course name
- 6. Enter the grade- Input the each course grade in the system.

```
ENTER THE NAME : Jeyapriya_Jeyakrishnan
ENTER THE INDEX NUMBER : 2021/T/0123
ENTER THE DEPARTMENT : Information_&_Communication_Technology
ENTER THE COURSE COUNT : 3

ENTER COURSE NAME : Programming
ENTER COURSE GRADE : A
ENTER COURSE NAME : Basic_Mathamatics
ENTER COURSE GRADE : B
ENTER COURSE GRADE : B
ENTER COURSE GRADE : A

(press 1 to add another student details
or press 0 : 0

ENTER THE PATH :
```

Figure 2: Add Student Details

Add to Course details

This function opens the "course.txt" binary file in append mode and writes the course details in the binary file. If you wish to enter another course details input 1 to continue or exit the add course details function press 0.

- 1. Enter the Course name- input the course name
- 2. Enter the Course ID input course id

```
ENTER THE COURSE NAME : Programming ENTER THE COURSE ID : IC_1302

press 1 to add another student press 0 to exit : 1

------ADD COURSE DETAILS------

ENTER THE COURSE NAME : Application_Lab ENTER THE COURSE ID : IC_1301

press 1 to add another student press 0 to exit :
```

Figure 3: Add Course Details

Display Student details

* Display in the terminal all student records as a report from the "student.txt" binary file.

```
STUDENT NAME :- Jeyapriya
DEPARTMENT :- ICT
INDEX NUMBER :- 2021t01213ICT

COURSE DETAILS

COURSE NAME :- Programming
COURSE GRADE :- A
CREDIT VALUE :- 4.00

COURSE NAME :- Basic_Mathamatics
COURSE GRADE :- B
CREDIT VALUE :- 3.00

GPA OF THE STUDENT :- 3.50
```

Figure 4: Display Student Details

Display course details

* Display all course details in the terminal from the course.txt" binary text file.

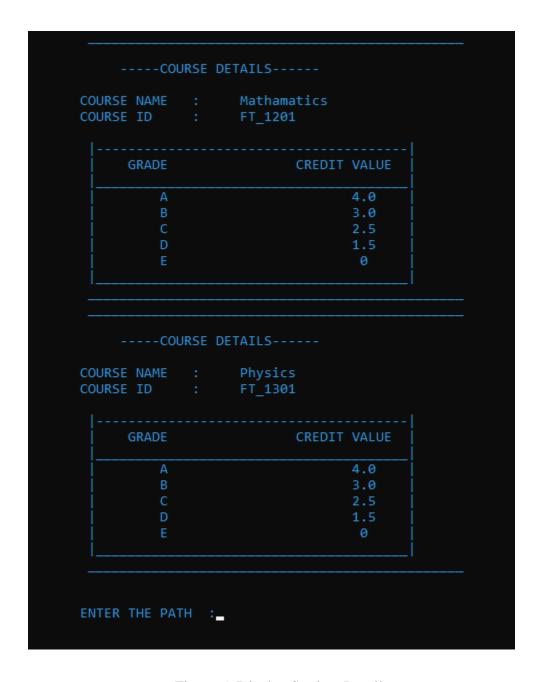


Figure 5: Display Student Details

Update Student Details

If any of the Student details have been mistyped or wrong, the user should be allowed to update the information in the text file using the terminal. The user first enters the index number of the student concurrently the system can search when the index number matches, and after the user can update the student's details.

* Users can enter the index number in the terminal

```
ENTER STUDENT INDEX NUMBER :123
```

Figure 6: enter the index number of the student

* Users can change the name, department, index number and course details of each student.

```
ENTER STUDENT INDEX NUMBER :123
------UPDATE STUDENT DETAILS-----
ENTER THE NAME :
```

Figure 7: Get new student details

* The user enters the new student details then the system gives a message to user "record updated successfully."

Figure 8: Get new student details

Search each student details

This function opens the binary file in reading mode and asks the user to enter the student index number he wants to search for. If the student information is unavailable on the list, it shows the message "record not found".

* User can enter the index number in the terminal he wants to search for.

```
ENETR THE INDEX NUMBER :
```

Figure 9: input index number

* If each record is not found display a message to the user "record not found."

```
ENETR THE INDEX NUMBER : 123

Record not found . . . .
```

Figure 10: display record note found

* If the specific record is found then display the student details in the terminal.

```
ENETR THE INDEX NUMBER : 147

-----STUDENT DETAILS-----

STUDENT NAME :- Kavin
DEPARTMENT :- ET
INDEX NUMBER :- 147

------COURSE DETAILS-----

COURSE NAME :- ET
COURSE GRADE :- B
CREDIT VALUE :- 3.00

GPA OF THE STUDENT :- 3.00

ENTER THE PATH :
```

Figure 11: display specific student details

Exit the entire program

When the user wants to exit the entire program input the number "7". Then user can exit the entire program.

```
STUDENT

MANAGEMENT SYSTEM

01. ADD NEW STUDENT DETAILS

02. ADD NEW COURSE DETAILS

03. DISPLAY STUDENT DETAILS

04. DISPLAY COURSE DETAILS

05. UPDATE STUDENT DETAILS

06. SURCH EACH STUDENT DETAILS

07. EXIT THE ENTIRE PROGRAM

ENTER THE PATH :7

C:\Users\DELL\Desktop>
```

Figure 12: Exit the program

Appendix A: The C program

The complete C Program with functions folded. (450-lines)

```
//---Header files----//
      #include<stdio.h>
      #include<string.h>
      #include<conio.h>
      #include<stdlib.h>
      //---structures---//
     struct course
     + (
 13
     struct student
     ±{
14
23
    #struct c{
28
      void add();
30
     void course details();
31
     void display();
      void display_c_details();
32
33
     void grade();
34
     void update();
35
     void search();
      int main()
37 +{
    void add()
93
94
198
     void course_details()
199
230
     void display()
231
    void display_c_details()
269
270
303
     void update()
304
420
     void search()
421
     ±{
```

Figure 13: entire program

Including necessary header files

- 1. stdio.h
- 2. string.h
- 3. conio.h
- 4. stdlib.h

```
//---Header files----//
#include<stdio.h>
#include<string.h>
#include<conio.h>
#include<stdlib.h>
```

Figure 14: header file

Definitions of database structure

* Generally, the database in C is handled by File. Here we declare the Structure which would store to a binary or regular file. Including structure can store course details and student details in binary files.

```
//---structures---//
       struct course
 8
 9
           char c_name[20];
10
          char grade;
11
          float c_value;
     L};
12
13
      struct student
14
15
          char name[20];
16
          char index[20];
17
          char dep[20];
18
          struct course c[10];
19
          int c_count;
20
          float total;
21
          float gpa;
22
    L);
   struct c{
23
24
          char name[20];
25
          char id[10];
26
     L};
27
28
```

Figure 15: structures

Including functions

- 1. Void add finding student details from the user and store them in the binary file.
- 2. Void course_details –finding course details from the user and store them in the binary file.
- 3. Void display display all student details in the terminal as a report.
- 4. Void display_c_details –display all course details in the structured way in the terminal.
- 5. Void update Update student details when the details wrong or mistyped.
- 6. Void search- display each student detail in the terminal.

```
28
      //including functions
29
      void add();
30
      void course_details();
31
      void display();
      void display_c_details();
32
33
      void update();
34
      void search();
35
      int main()
```

Figure 16: functions

Main function

```
int main()
35
36
   □{
37
         int n;
38
         system("cls");
39
         system("COLOR B");
40
41
         42
                        "======\n");
         printf( "\t\t\t
                                                STUDENT\n");
43
         \label{eq:printf( "} t\t \t
                                           MANAGEMENT SYSTEM\n");
44
45
46
         printf( "\t\t=======""
                        "======\n\n\n");
47
48
49
         printf("\n\t\t\t 01. ADD NEW STUDENT DETAILS ");
50
         printf("\n\t\t\t 02. ADD NEW COURSE DETAILS ");
         printf("\n\t\t\t 03. DISPLAY STUDENT DETAILS ");
51
         printf("\n\t\t\ 04. DISPLAY COURSE DETAILS ");
52
         printf("\n\t\t\t 05. UPDATE STUDENT DETAILS ");
53
         printf("\n\t\t\t 06. SURCH EACH STUDENT DETAILS ");
printf("\n\t\t\t 07. EXIT THE ENTIRE PROGRAM ");
54
55
56
```

Figure 17: main function

```
54
           printf("\n\t\t\t
                               06. SURCH EACH STUDENT DETAILS ");
55
           printf("\n\t\t\t
                               07. EXIT THE ENTIRE PROGRAM ");
56
57
           do{
58
               printf("\n\n\tENTER THE PATH :");
59
60
               scanf ("%d", &n);
61
               switch(n){
62
                   case 1:
63
                       add();
64
                       goto choice;
65
                       break;
66
67
                   case 2:
                       course_details();
68
69
                       goto choice;
70
                       break;
71
                   case 3:
72
                       display();
73
                       goto choice;
74
                       break;
75
                   case 4:
76
                       display_c_details();
77
                       goto choice;
78
                       break;
79
                   case 5:
80
                       update();
81
                       goto choice;
82
                       break;
83
                   case 6:
84
                       search();
85
                       goto choice;
86
                       break;
87
               } while (n>=1&&n<7);
88
89
```

Figure 18: main function

Add student detail function and calculate the GPA of the student

```
90
      void add()
91
    92
          system("cls");
93
          int a;
94
          FILE *fp;
95
96
          fp=fopen("student.txt", "a");
97
          if (fp == NULL)
98
99
            printf("Error!");
100
            exit(1);
101
102
103
          struct student s;
104
105
106
          do {
107
              s.gpa=0;
108
              s.total=0;
109
              float k[5]={4.0,3.0,2.5,1.5,0};
110
              system ("COLOR E");
111
              printf("\n\n\t-----\n\n");
              printf("\n\t ENTER THE NAME
112
                                            : ");
113
              scanf("%s",s.name);
114
              printf("\t ENTER THE INDEX NUMBER : ");
              scanf("%s",s.index);
115
116
              printf("\t ENTER THE DEPARTMENT
                                               : ");
117
              scanf("%s",s.dep);
118
              fflush (stdin);
119
              printf("\t ENTER THE COURSE COUNT : ");
120
              scanf("%d", &s.c_count);
121
              for(int i=0;i<s.c_count;i++)</pre>
122
```

Figure 19: add function

```
119
                printf("\t ENTER THE COURSE COUNT : ");
120
                scanf ("%d", &s.c_count);
121
                for(int i=0;i<s.c_count;i++)</pre>
122
123
                    printf("\t\t ENTER COURSE NAME : ");
124
                    scanf("%s",s.c[i].c_name);
125
                    fflush(stdin);
126
                    printf("\t\t ENTER COURSE GRADE : ");
127
                    scanf("%c",&s.c[i].grade);
128
                    if(s.c[i].grade=='a')
129
130
                        s.c[i].c_value=k[0];
131
                        s.total+=s.c[i].c_value;
132
133
134
                    else if(s.c[i].grade=='A')
135
136
                        s.c[i].c_value=k[0];
137
                        s.total+=s.c[i].c_value;
138
139
140
                    else if(s.c[i].grade=='B')
141
142
                        s.c[i].c_value=k[1];
143
                        s.total+=s.c[i].c_value;
144
145
146
                    else if(s.c[i].grade=='b')
147
148
                        s.c[i].c_value=k[1];
149
                        s.total+=s.c[i].c_value;
150
151
                    else if(s.c[i].grade=='C')
152
153
                        s.c[i].c_value=k[2];
154
                        s.total+=s.c[i].c_value;
155
```

Figure 20: add function

```
160
161
                    else if(s.c[i].grade=='D')
162
163
                        s.c[i].c_value=k[3];
164
                        s.total+=s.c[i].c_value;
165
166
                    else if(s.c[i].grade=='d')
167
168
                        s.c[i].c_value=k[3];
169
                        s.total+=s.c[i].c_value;
170
                    else if(s.c[i].grade=='E')
171
172
173
                        s.c[i].c_value=k[4];
174
                        s.total+=s.c[i].c_value;
175
176
                    else if(s.c[i].grade=='e')
177
178
                        s.c[i].c_value=k[4];
179
                        s.total+=s.c[i].c_value;
180
                    }
181
182
                }
183
184
                s.gpa=s.total/(float)s.c_count;
185
                fwrite(&s,sizeof(struct student),1,fp);
186
                printf("\n(press 1 to add another student details"
                          "\n or press 0 : ");
187
188
                scanf ("%d", &a);
189
190
191
            }while(a!=0);
192
            fclose(fp);
193
      L}
194
```

Figure 21: add function

Finding course details in the binary file

```
195
       void course_details()
     □ {
196
197
           system("cls");
198
           int a;
199
           FILE *fpl;
200
           fpl=fopen("course.txt", "a");
201
           if(fpl == NULL)
202
203
             printf ("Error!");
204
            exit(1);
205
206
           struct c cl;
207
208
209
           do {
210
               system("cls");
211
               system("COLOR C");
212
               printf("\n\n\t\t----\n\n");
213
214
               printf("\t\t ENTER THE COURSE NAME
                                                 : ");
215
              scanf("%s",cl.name);
216
              printf("\t\t ENTER THE COURSE ID
217
              scanf("%s",cl.id);
218
              fwrite(&cl,sizeof(struct c),1,fpl);
219
              printf("\n\n press 1 to add another student\n"
220
                       " press 0 to exit : ");
               scanf("%d",&a);
221
222
223
           }while(a!=0);
224
           fclose(fpl);
225
226
```

Figure 22: add course details function

Display student details in the terminal

```
227
      void display()
228
    \square{
229
          system("cls");
230
          FILE *fp;
231
          struct student s;
232
          fp=fopen("student.txt","r");
233
          if (fp == NULL)
234
235
            printf("Error!");
236
            exit(1);
237
238
          239
          printf("\n\t\t\t\tStudent Report");
240
          241
          while (fread (&s, size of (struct student), 1, fp))
242
243
              system("COLOR D");
244
             printf("\n\t ____
             printf("\n\n\t ------STUDENT DETAILS-----");
245
246
             printf("\n\n\t STUDENT NAME :- %s",s.name);
247
             printf("\n\t DEPARTMENT :- %s",s.dep);
248
             printf("\n\t INDEX NUMBER :- %s",s.index);
249
             printf("\n");
250
             printf("\n\n\t
                                    COURSE DETAILS
                                                  ");
251
              for(int i=0;i< s.c_count;i++)</pre>
252
253
                 printf("\n\n\t COURSE NAME :- %s",s.c[i].c_name);
254
                 printf("\n\t COURSE GRADE :- %c",s.c[i].grade);
255
                 printf("\n\t CREDIT VALUE :- %.2f\n",s.c[i].c_value);
256
257
             printf("\n\n\t GPA OF THE STUDENT :- %.2f",s.gpa);
258
             printf("\n\t _
259
260
          fclose(fp);
261
```

Figure 23: Display Student Details

Display course details in the terminal

```
262
       void display_c_details()
263
264
           system("cls");
265
           int a;
266
           FILE *fpl;
267
           fpl=fopen("course.txt", "r");
268
           if (fpl == NULL)
269
270
             printf("Error!");
271
            exit(1);
272
273
           struct c cl;
274
           while (fread (&cl, size of (struct c), l, fpl))
275
276
               system("COLOR 3");
277
               printf("\n\t
                                                                     ____");
               printf("\n\n\t ----COURSE DETAILS----\n");
278
               \label{eq:printf("\n\tCOURSE NAME : $s",cl.name);} \\
279
280
               printf("\n\tCOURSE ID :
                                              %s\n",cl.id);
281
               printf("\n\t |-----|");
                                           CREDIT VALUE |");
282
               printf("\n\t | GRADE
283
               printf("\n\t |____
                                                                    I");
              printf("\n\t | A
printf("\n\t | B
printf("\n\t | C
printf("\n\t | D
printf("\n\t | E
printf("\n\t | E
284
285
                                                             3.0 |");
286
                                                             2.5 |");
287
                                                             1.5 |");
288
                                                             0
                                                                   |");
289
290
                                                                            ");
               printf("\n\t
291
292
           fclose(fpl);
293
```

Figure 24: Display Course Details

Update student details

```
294
      void update()
295
    □{
296
           system("cls");
297
           system("COLOR B");
298
           FILE *fp,*temp;
299
           char temp_idex[10];
300
           struct student s;
301
           int found=0;
302
           fp=fopen("student.txt","r+");
303
           temp=fopen("temp.txt","a+");
304
           if (fp == NULL)
305
306
            printf("Error!");
307
             exit(1);
308
           printf("\nENTER STUDENT INDEX NUMBER :");
309
310
           scanf("%s",temp_idex);
311
312
           while(fread(&s,sizeof(struct student),1,fp))
313
314
               if (strcmp(temp_idex,s.index)!=0)
315
316
                   fwrite(&s,sizeof(struct student),1,temp);
317
318
              else if(strcmp(temp_idex,s.index)==0)
319
320
321
                   found=1;
322
                  s.gpa=0;
323
                  s.total=0;
324
                  float k[5]={4.0,3.0,2.5,1.5,0};
325
                  printf("\n\n\t-----\n\n");
                  printf("\n\n\t ENTER THE NAME
326
327
                  scanf("%s",s.name);
328
                  printf("\t ENTER THE INDEX NUMBER : ");
329
                   scanf("%s",s.index);
```

Figure 25: Update student Details

```
330
                    printf("\t ENTER THE DEPARTMENT
331
                    scanf("%s",s.dep);
332
                    fflush(stdin);
333
                    printf("\tENTER THE COURSE COUNT : ");
334
                    scanf("%d",&s.c_count);
335
                    for(int i=0;i<s.c_count;i++)</pre>
336
337
                        printf("\t\t\t ENTER COURSE NAME : ");
338
                        scanf("%s",s.c[i].c_name);
339
                        fflush (stdin);
340
                        printf("\t\t ENTER COURSE GRADE : ");
341
                        scanf("%c",&s.c[i].grade);
342
                        if(s.c[i].grade=='a')
343
344
                            s.c[i].c_value=k[0];
345
                            s.total+=s.c[i].c_value;
346
347
348
                        else if(s.c[i].grade=='A')
349
350
                            s.c[i].c_value=k[0];
351
                            s.total+=s.c[i].c_value;
352
353
354
                        else if (s.c[i].grade=='B')
355
356
                            s.c[i].c_value=k[l];
357
                            s.total+=s.c[i].c_value;
358
359
                        else if (s.c[i].grade=='b')
360
361
                            s.c[i].c_value=k[1];
362
                            s.total+=s.c[i].c_value;
363
364
                        else if(s.c[i].grade=='C')
365
```

Figure 26: Update student Details

```
374
                        else if(s.c[i].grade=='D')
375
376
                            s.c[i].c_value=k[3];
377
                            s.total+=s.c[i].c_value;
378
379
                        else if(s.c[i].grade=='d')
380
381
                            s.c[i].c value=k[3];
382
                            s.total+=s.c[i].c_value;
383
384
                        else if(s.c[i].grade=='E')
385
386
                            s.c[i].c_value=k[4];
387
                            s.total+=s.c[i].c_value;
388
389
                        else if(s.c[i].grade=='e')
390
391
                            s.c[i].c_value=k[4];
392
                            s.total+=s.c[i].c value;
393
                        }
394
                        else{
395
                            s.c[i].c_value=k[4];
396
                            s.total+=s.c[i].c_value;
397
                        }
398
399
400
                s.gpa=s.total/(float)s.c_count;
401
                fwrite(&s,sizeof(struct student),1,temp);
402
403
404
405
            fclose(fp);
406
            fclose(temp);
407
            remove ("student.txt");
408
            rename ("temp.txt", "student.txt");
            printf("\n\n\tRecord updated successfully....");
409
410
```

Figure 27: Update student Details

Search student details

```
411
      void search()
     □{
412
413
           system("cls");
414
           FILE *fp;
415
           struct student s;
416
           char temp[20];
417
           int templ=0;
418
           fp=fopen("student.txt", "r");
419
           if (fp == NULL)
420
421
            printf("Error!");
422
             exit(1);
423
424
           printf("\n\t ENETR THE INDEX NUMBER : ");
425
           scanf("%s",temp);
426
           while(fread(&s,sizeof(struct student),1,fp))
427
428
               if (strcmp (temp, s.index) == 0)
429
430
                  templ=1;
431
                  system("COLOR A");
                  printf("\n\t -----");
432
433
                  printf("\n\n\t STUDENT NAME :- %s",s.name);
                  printf("\n\t DEPARTMENT :- %s",s.dep);
434
435
                  printf("\n\t INDEX NUMBER :- %s",s.index);
436
                  printf("\n\n\t -----");
437
                  for(int i=0;i<s.c count;i++)</pre>
438
439
                      printf("\n\t COURSE NAME :- %s",s.c[i].c_name);
440
                      printf("\n\t COURSE GRADE :- %c",s.c[i].grade);
441
                      printf("\n\t CREDIT VALUE :- %.2f\n",s.c[i].c_value);
442
443
                  printf("\n\t GPA OF THE STUDENT :- %.2f",s.gpa);
444
                  printf("\n\t
                                                                      ");
445
               }
446
```

Figure 28: Search student Detail

```
445
446
447
448
448
449
450

| fclose(fp);
| fclose(fp);
```

Figure 29: Search student Details

St	ud	pr	r+	m	an	a	70	m	ρn	t c	vste	n
IJι	иu	CI	ΙL	III	un	uц	15	III	CII	ιs	ysle.	П

Reference

1. w3school -

https://www.w3schools.com/c/

2. YouTube videos-

 $\underline{https://www.youtube.com/results?search_query=how+to+use+colourse+in+c+program}\\ \underline{ming}$

https://www.youtube.com/watch?v=8nIilb2kiSU