



DEPARTMENT OF BASIC SCIENCE AND HUMANITIES
INSTITUTE OF ENGINEERING AND MANAGEMENT, KOLKATA.

STUDENT MANAGEMENT SYSTEM

SUBMITTED BY:

NAME: ANGIRAVO SAHA

ENTROLLMENT NUMBER: 12022002003129

SECTION: G

CLASS ROLL NUMBER: 23

STREAM: ECE

SUBJECT: PROGRAMMING FOR PROBLEM SOLVING USING C

SUBJECT CODE: ESC103(PR)

DEPARTMENT: BASIC SCIENCE AND HUMANITIES

UNDER THE SUPERVISION OF:

PROF: SWARNENDU GHOSH

ACADEMIC YEAR: 2022-2023

(PROJECT REPORT SUBMITTED IN FULFILLMENT OF THE
REQUIREMENTS FOR THE SECOND SEMESTER)



CERTIFICATE OF RECOMMENDATION

We hereby recommend that the project prepared under our supervision by Anushka Pandit, entitled “Student Management System” be accepted in fulfilment of the requirements for the degree of fulfilment of the second semester.

Head of the Department
IEM, Kolkata

Project Supervisor
Basic Science and Humanities

1.

1. INTRODUCTION:

This project is assigned to me for developing a Student Management System with the help of basic C programming language.

The basic aim of the project is to create a student management system where we

need to put up basic student details and thereby with the help of c programming, we have to create a portal (.exe file) for adding new student information, their roll number, marks obtained, reading and deleting student information, viewing all student list at a glance.

2. Variable Description:

The different variables used in this project are listed under:-

1. int- To store integer datatypes.
2. char- To store character datatypes.

3. Function Description:

The different functions (structures) used in this project are listed under:-

1. create_student- For creating the required student details vis. name,.
2. read_student- For reading the student details.
3. update_student- For updating any student information.
4. delete_student- To delete any of the student information.
5. struct_student – A value struct is a fixed size structure that contains only public data fields and is declared by using the value struct keyword.

5. PROGRAM CODE:

```
C > Users > Anushka > AppData > Local > Microsoft > Windows > INetCache > IE > A61FVT68 > C st
1  #include <stdio.h>
2  #include <string.h>
3
4  #define MAX_NAME_LENGTH 50
5  #define MAX_STUDENTS 100
6
7  struct student {
8      char name[MAX_NAME_LENGTH];
9      int roll_number;
10     float marks;
11     char grade;
12 };
13
14 void create_student(struct student *s) {
15     printf("Enter name: ");
16     scanf("%s", s->name);
17     printf("Enter roll number: ");
18     scanf("%d", &s->roll_number);
19     printf("Enter marks: ");
```

```

19     printf("Enter marks: ");
20     scanf("%f", &s->marks);
21     if (s->marks >= 90) {
22         s->grade = 'A';
23     } else if (s->marks >= 80) {
24         s->grade = 'B';
25     } else if (s->marks >= 70) {
26         s->grade = 'C';
27     } else if (s->marks >= 60) {
28         s->grade = 'D';
29     } else {
30         s->grade = 'F';
31     }
32 }
33
34 void read_student(struct student *s) {
35     printf("Name: %s\n", s->name);
36     printf("Roll Number: %d\n", s->roll_number);
37     printf("Marks: %.2f\n", s->marks);

```

```

C student[1].c X  Extension: C/C++ Extension Pack
C: > Users > Anushka > AppData > Local > Microsoft > Windows > INetCache > IE > A61FVT68 > C student[1].c >
38     printf("Grade: %c\n", s->grade);
39 }
40
41 void update_student(struct student *s) {
42     printf("Enter new name (or press enter to keep the same): ");
43     char new_name[MAX_NAME_LENGTH];
44     getchar();
45     fgets(new_name, MAX_NAME_LENGTH, stdin);
46     new_name[strcspn(new_name, "\n")] = '\0'; // remove newline character
47     if (strlen(new_name) > 0) {
48         strcpy(s->name, new_name);
49     }
50     printf("Enter new marks (or enter -1 to keep the same): ");
51     float new_marks;
52     scanf("%f", &new_marks);
53     if (new_marks != -1) {
54         s->marks = new_marks;
55         if (s->marks >= 90) {
56             s->grade = 'A';
57         } else if (s->marks >= 80) {
58             s->grade = 'B';
59         } else if (s->marks >= 70) {
60             s->grade = 'C';
61         } else if (s->marks >= 60) {
62             s->grade = 'D';
63         } else {
64             s->grade = 'F';
65         }
66     }
67 }
68
69 void delete_student(struct student *s) {
70     memset(s->name, 0, MAX_NAME_LENGTH);
71     s->roll_number = 0;
72     s->marks = 0.0;
73     s->grade = '\0';
74 }

```

```

C: > Users > Anushka > AppData > Local > Microsoft > Windows > INetCache > IE > A61FVT68 > C student[1].c > ...
75
76 int main() {
77     struct student students[MAX_STUDENTS];
78     int num_students = 0;
79     int choice = 0;
80     while (choice != 6) {
81         printf("Menu:\n");
82         printf("1. Create new student\n");
83         printf("2. Read student\n");
84         printf("3. Update student\n");
85         printf("4. Delete student\n");
86         printf("5. View all students list\n");
87         printf("6. Exit\n");
88         printf("Enter your choice: ");
89         scanf("%d", &choice);
90         printf("\n");
91         switch (choice) {
92             case 1:

```

```

93         if (num_students < MAX_STUDENTS) {
94             create_student(&students[num_students]);
95             num_students++;
96             printf("Student created successfully.\n\n");
97         } else {
98             printf("Maximum number of students reached.\n\n");
99         }
100         break;
101     case 2:
102         printf("Enter roll number of student to read: ");
103         int roll_number_to_read;
104         scanf("%d", &roll_number_to_read);
105         printf("\n");
106         int found_student_index = -1;
107         for (int i = 0; i < num_students; i++) {
108             if (students[i].roll_number == roll_number_to_read) {
109                 found_student_index = i;
110                 break;
111             }

```

```

C:\Users> Anushka > AppData > Local > Microsoft > Windows > INetCache > IE > A61FVT68 > C:\student[1].c > ...
112     }
113     if (found_student_index == -1) {
114         printf("Student with roll number %d not found.\n\n", roll_number_to_read);
115     } else {
116         read_student(&students[found_student_index]);
117         printf("\n");
118     }
119     break;
120     case 3:
121         printf("Enter roll number of student to update: ");
122         int roll_number_to_update;
123         scanf("%d", &roll_number_to_update);
124         printf("\n");
125         found_student_index = -1;
126         for (int i = 0; i < num_students; i++) {
127             if (students[i].roll_number == roll_number_to_update) {
128                 found_student_index = i;
129                 break;
130             }
131         }
132         if (found_student_index == -1) {
133             printf("Student with roll number %d not found.\n\n", roll_number_to_update);
134         } else {
135             update_student(&students[found_student_index]);
136             printf("Student updated successfully.\n\n");
137         }
138         break;
139     case 4:
140         printf("Enter roll number of student to delete: ");
141         int roll_number_to_delete;
142         scanf("%d", &roll_number_to_delete);
143         printf("\n");
144         found_student_index = -1;
145         for (int i = 0; i < num_students; i++) {
146             if (students[i].roll_number == roll_number_to_delete) {

```

```

C:\Users> Anushka > AppData > Local > Microsoft > Windows > INetCache > IE > A61FVT68 > C:\student[1].c > ...
149     }
150     }
151     if (found_student_index == -1) {
152         printf("Student with roll number %d not found.\n\n", roll_number_to_delete);
153     } else {
154         delete_student(&students[found_student_index]);
155         printf("Student deleted successfully.\n\n");
156     }
157     break;
158     case 5:
159         if (num_students == 0) {
160             printf("No students to display.\n\n");
161         } else {
162             printf("All Students List:\n");
163             for (int i = 0; i < num_students; i++) {
164                 printf("Student %d:\n", i+1);
165                 read_student(&students[i]);
166                 printf("\n");
167             }
168         }
169         break;
170     case 6:
171         printf("Exit\n\n");

```

```

171         printf("Exiting...\n");
172         break;
173     default:
174         printf("Invalid choice. Please try again.\n\n");
175         break;
176     }
177 }
178 return 0;
179 }
180

```

6. OUTPUT:

```

"C:\ALL C PROGRAMS\fp2.exe" x + v
Menu:
1. Create new student
2. Read student
3. Update student
4. Delete student
5. View all students list
6. Exit
Enter your choice: 1

Enter name: ron de
Enter roll number: Enter marks: Student created successfully.

Menu:
1. Create new student
2. Read student
3. Update student
4. Delete student
5. View all students list
6. Exit
Enter your choice:
Enter name: Enter roll number: |

```

```

"C:\ALL C PROGRAMS\fp2.exe" x + v
2. Read student
3. Update student
4. Delete student
5. View all students list
6. Exit
Enter your choice: 1

Enter name: ron de
Enter roll number: Enter marks: Student created successfully.

```

```
Menu:
1. Create new student
2. Read student
3. Update student
4. Delete student
5. View all students list
6. Exit
Enter your choice:
Enter name: Enter roll number: 5
Enter marks: 45
Student created successfully.
```

```
Menu:
1. Create new student
2. Read student
3. Update student
4. Delete student
5. View all students list
6. Exit
Enter your choice: |
```

THANK YOU!!!

