INSTITUTE OF ENGINEERING AND MANAGEMENT, KOLKATA

DEPARTMENT OF BASIC SCIENCE AND HUMANITIES

"Student Management System:-"

Submitted by:-

Name of the Student: Sankalan Samanta

Enrolment :12022002017081

Registration Number: 221040110483

Section: A

Class Roll Number: 83

Stream: CSE-IOTCSBT

Subject: Programming for Problem Solving Using C

Code:ESC(102)PR

Under the supervision of:**Prof. Swarnendu Ghosh**

Academic Year: 2022-26

(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 **SANKALAN SAMANTA,** entitled **Student Management System** be accepted in partial fulfillment of the requirements for the degree of partial fulfillment of the first year.

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 ca fixed size structure that contains only public data fields and is declared by using the value struct keyword.

CODE:-

```
#include <stdio.h>
#include <string.h>

#define MAX_NAME_LENGTH 50
#define MAX_STUDENTS 100

struct student {
    char name[MAX_NAME_LENGTH];
    int roll_number;
    float marks;
    char grade;
};

void create_student(struct student *s) {
    printf("Enter name: ");
    scanf("%s", s->name);
    printf("Enter roll number: ");
    scanf("%d", &s->roll_number);
    printf("Enter marks: ");
    scanf("%f", &s->marks);
    if (s->marks >= 90) {
        s->grade = 'A';
    } else if (s->marks >= 80) {
        s->grade = 'B';

    } else if (s->marks >= 70) {
        s->grade = 'C';
    } else if (s->marks >= 60) {
        s->grade = 'D';
    } else {
        s->grade = 'F';
    }
}

void read_student(struct student *s) {
    printf("Name: %s\n", s->name);
    printf("Roll Number: %d\n", s->roll_number);
    printf("Marks: %.2f\n", s->marks);
    printf("Grade: %c\n", s->grade);
}

void update_student(struct student *s) {
    printf("Enter new name (or press enter to keep the same): ");
    char new_name[MAX_NAME_LENGTH];
    getchar();
    fgets(new_name, MAX_NAME_LENGTH, stdin);
    new_name[strcspn(new_name, "\n")] = '\0'; // remove newline character
    if (strlen(new_name) > 0) {
        strcpy(s->name, new_name);
    }
    printf("Enter new marks (or enter -1 to keep the same): ");
    float new_marks;
    scanf("%f", &new_marks);
    if (new_marks != -1) {
        s->marks = new_marks;
        if (s->marks >= 90) {
```

```
            s->grade = 'A';
        } else if (s->marks >= 80) {
            s->grade = 'B';
        } else if (s->marks >= 70) {
            s->grade = 'C';
        } else if (s->marks >= 60) {
            s->grade = 'D';
        } else {
            s->grade = 'F';
        }
    }
}

void delete_student(struct student *s) {
    memset(s->name, 0, MAX_NAME_LENGTH);
    s->roll_number = 0;
    s->marks = 0.0;
    s->grade = '\0';
}

int main() {
    struct student students[MAX_STUDENTS];
    int num_students = 0;
    int choice = 0;
    while (choice != 6) {
        printf("Menu:\n");
        printf("1. Create new student\n");
        printf("2. Read student\n");
        printf("3. Update student\n");
        printf("4. Delete student\n");
        printf("5. View all students list\n");
        printf("6. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        printf("\n");
        switch (choice) {
            case 1:
                if (num_students < MAX_STUDENTS) {
                    create_student(&students[num_students]);
                    num_students++;
                printf("Student created successfully.\n\n");
                } else {
                    printf("Maximum number of students reached.\n\n");
                }
                break;
        case 2:
            printf("Enter roll number of student to read: ");
            int roll_number_to_read;
            scanf("%d", &roll_number_to_read);
            printf("\n");
            int found_student_index = -1;
            for (int i = 0; i < num_students; i++) {
                if (students[i].roll_number == roll_number_to_read) {
                    found_student_index = i;
                    break;
                }
            }
            if (found_student_index == -1) {
```

```
                printf("Student with roll number %d not found.\n\n",
roll_number_to_read);
            } else {
                read_student(&students[found_student_index]);
                printf("\n");
            }
            break;
        case 3:
            printf("Enter roll number of student to update: ");
            int roll_number_to_update;
            scanf("%d", &roll_number_to_update);
            printf("\n");
            found_student_index = -1;
            for (int i = 0; i < num_students; i++) {
                if (students[i].roll_number == roll_number_to_update) {
                    found_student_index = i;
                    break;
                }
            }
            if (found_student_index == -1) {
                printf("Student with roll number %d not found.\n\n",
roll_number_to_update);
            } else {
                update_student(&students[found_student_index]);
                printf("Student updated successfully.\n\n");
            }
            break;
        case 4:
            printf("Enter roll number of student to delete: ");
            int roll_number_to_delete;
            scanf("%d", &roll_number_to_delete);
            printf("\n");
            found_student_index = -1;
            for (int i = 0; i < num_students; i++) {
                if (students[i].roll_number == roll_number_to_delete) {
                    found_student_index = i;
                    break;
                }
            }
            if (found_student_index == -1) {
                printf("Student with roll number %d not found.\n\n",
roll_number_to_delete);
            } else {
                delete_student(&students[found_student_index]);
                printf("Student deleted successfully.\n\n");
            }
            break;
        case 5:
            if (num_students == 0) {
                printf("No students to display.\n\n");
            } else {
                printf("All Students List:\n");
                for (int i = 0; i < num_students; i++) {
                    printf("Student %d:\n", i+1);
                    read_student(&students[i]);
                    printf("\n");
                }
            }
```

```
            break;
        case 6:
            printf("Exiting...\n");
            break;
        default:
            printf("Invalid choice. Please try again.\n\n");
            break;
    }
}
return 0;
}
```

OUTPUTS:-

Sample outputs (screenshots) to demonstrate the functionalities in programs are listed below.