Project of C Programming

Student Management System Using C Programming

Name : Kaki Vishnu

Sap ID : 590021647

Batch : 53

Name : Anurag Yadav

Sap ID : 590027499

Batch : 53

Introduction :

- This project is a menu-driven Student Management System.
- It allows adding, searching, and displaying student details.
- It is built using **structures, arrays, and functions** in C.
- Designed to be simple, fast, and user-friendly.
- It allows the user to add new student records, search for specific students, view all stored records, and calculate the class average.

Project Objectives :

- To store and manage student information efficiently.
- To implement **structures** for grouping student data.
- To provide essential operations like:
    1. Add Student
    2. Display Students
    3. Search Student
    4. Calculate Average Marks
- To practice modular programming using functions.

Technologies Used :

- Programming Language: C
- Concepts Used:
    1. Structures
    2. Arrays
    3. Functions
    4. Loops &
    5. Conditional Statements
    6. Compiler Used: Online C Compiler (Programiz)

System Features :

- Add new student record
- View all student records
- Search student by roll number
- Calculate class average
- User-friendly menu-driven interface

System Flow :

- Add Student → Store Data → View/Search → Output Display
- Flowchart-style steps:
    1. Program starts
    2. Menu displayed
    3. User selects an option
    4. Corresponding function runs
    5. Result displayed
    6. Menu shown again

Data Structure Used :

Structure definition:

```
struct Student {
  int roll;
  char name[50];
  float marks;
};
```

Purpose:

- Groups roll number, name, and marks together
- Allows storing multiple students in an array

Code:

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

struct Student {
   int roll;
   char name[50];
   float marks;
};

struct Student students[100];   // store max 100 students
int count = 0;                  // number of students added

void addStudent() {
   if (count >= 100) {
      printf("Cannot add more students!\n");
      return;
   }

   printf("Enter Roll Number: ");
   scanf("%d", &students[count].roll);

   printf("Enter Name: ");
   scanf("%s", students[count].name);

   printf("Enter Marks: ");
   scanf("%f", &students[count].marks);
```

```
    count++;
    printf("Student added successfully!\n");
}

void displayStudents() {
    if (count == 0) {
        printf("No records to display!\n");
        return;
    }

    printf("\n--- Student Records ---\n");
    for (int i = 0; i < count; i++) {
        printf("Roll: %d | Name: %s | Marks: %.2f\n",
               students[i].roll, students[i].name, students[i].marks);
    }
}

void searchStudent() {
    int r, found = 0;
    printf("Enter Roll Number to Search: ");
    scanf("%d", &r);

    for (int i = 0; i < count; i++) {
        if (students[i].roll == r) {
            printf("\nRecord Found!\n");
            printf("Roll: %d | Name: %s | Marks: %.2f\n",
                   students[i].roll, students[i].name, students[i].marks);
            found = 1;
            break;
```

```
        }
    }

    if (!found) {
        printf("No student with roll number %d found!\n", r);
    }
}

void calculateAverage() {
    if (count == 0) {
        printf("No records to calculate average!\n");
        return;
    }

    float sum = 0;
    for (int i = 0; i < count; i++) {
        sum += students[i].marks;
    }

    printf("Class Average Marks = %.2f\n", sum / count);
}

int main() {
    int choice;

    while (1) {
        printf("\n--- Student Management System ---\n");
        printf("1. Add Student\n");
        printf("2. Display All Students\n");
```

```
        printf("3. Search Student\n");
        printf("4. Calculate Average Marks\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1: addStudent(); break;
            case 2: displayStudents(); break;
            case 3: searchStudent(); break;
            case 4: calculateAverage(); break;
            case 5: return 0;
            default: printf("Invalid choice! Try again.\n");
        }
    }
}
```

```
--- Student Management System ---
1. Add Student
2. Display All Students
3. Search Student
4. Calculate Average Marks
5. Exit
Enter your choice: 1
Enter Roll Number: 101
Enter Name: Vishnu
Enter Marks: 90
Student added successfully!

--- Student Management System ---
1. Add Student
2. Display All Students
3. Search Student
4. Calculate Average Marks
5. Exit
Enter your choice: 1
Enter Roll Number: 102
Enter Name: Anurag
Enter Marks: 85
Student added successfully!
```

Output-1 : Adding Students

Output-2 : Display all students

```
--- Student Management System ---
1. Add Student
2. Display All Students
3. Search Student
4. Calculate Average Marks
5. Exit
Enter your choice: 2

--- Student Records ---
Roll: 101 | Name: Vishnu | Marks: 90.00
Roll: 102 | Name: Anurag | Marks: 85.00
```

Output-3 : Search Student

```
--- Student Management System ---
1. Add Student
2. Display All Students
3. Search Student
4. Calculate Average Marks
5. Exit
Enter your choice: 3
Enter Roll Number to Search: 102

Record Found!
Roll: 102 | Name: Anurag | Marks: 85.00
```

Output-4 : Calculate Class Average

```
--- Student Management System ---
1. Add Student
2. Display All Students
3. Search Student
4. Calculate Average Marks
5. Exit
Enter your choice: 4
Class Average Marks = 87.50
```

Main Menu:

printf("1. Add Student\n");

printf("2. Display All Students\n");

printf("3. Search Student\n");

printf("4. Calculate Average Marks\n");

printf("5. Exit\n");

Explain:

The user selects an option

Switch-case executes the chosen functionality

Conclusion :

- **The project successfully demonstrates how to manage student data using C.**
- **Key concepts like structures, arrays, functions, and loops are effectively applied.**
- **The menu-driven approach makes the program simple and user-friendly.**
- **The system can handle multiple student records efficiently.**
- **The program can be easily expanded with features like update, delete, and file storage.**
- **This project shows how programming can solve real-life data management problems.**