STUDENT RECORD MANAGEMENT SYSTEM

Project in C Language

Name: Adarsh Singh

UID: 24BCA10542

SECTION: 8A (BCA)

Subject: DSA Using in C

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my faculty and mentors for their guidance and support in the successful completion of this mini project titled 'Student Record Management System'. This project has helped me gain valuable knowledge about programming in C and practical implementation of file handling and structures.

ABSTRACT

The Student Record Management System is a console-based mini project developed in the C programming language. It provides basic functionalities to manage student information such as adding, viewing, updating, searching, and deleting records. The system uses structures and binary file handling to store data permanently and is ideal for beginners to understand the practical use of file operations in C.

INTRODUCTION

Managing student information is essential in any educational institution. This system is designed to simplify the process of managing student records using a menudriven C program. It demonstrates basic concepts of structures, file handling, and modular programming in C.

PROBLEM STATEMENT

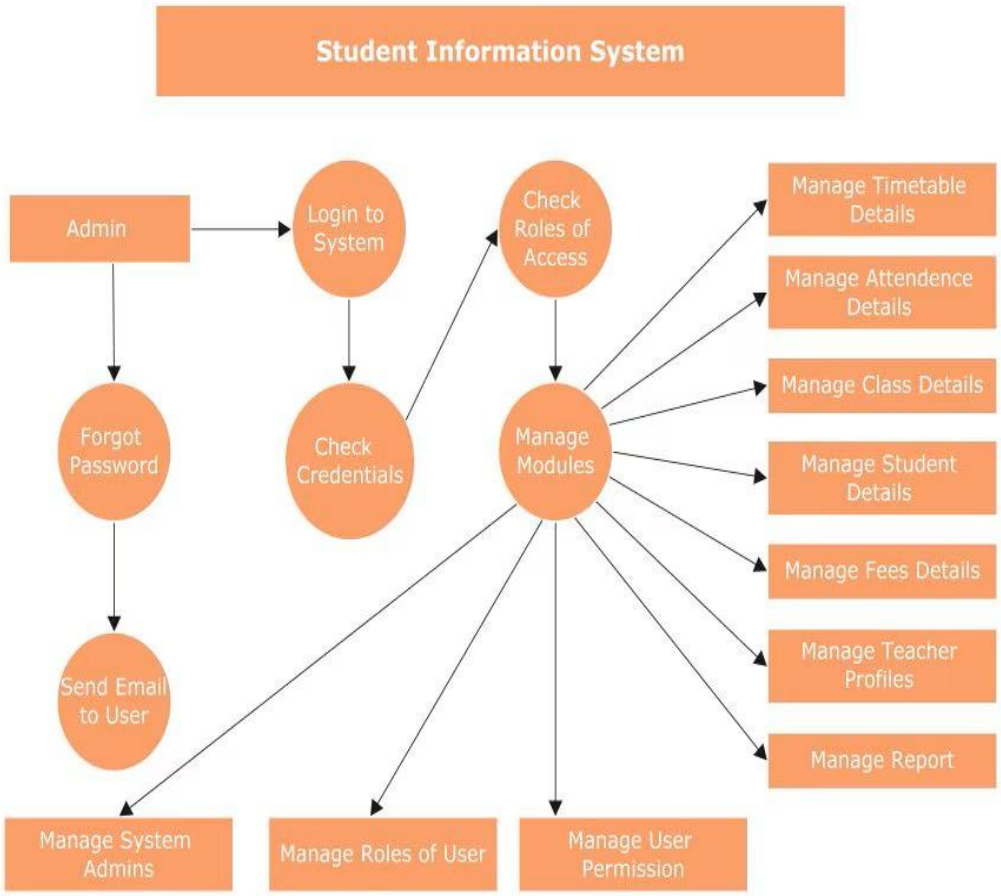
Manual record keeping of students is tedious and error-prone. The objective of this project is to develop an automated, simple, and efficient system using C programming to handle student data such as Roll Number, Name, and Marks. The system provides a menu-driven interface for ease of use and quick operations.

FLOW CHART

The following is the basic flow of the program:

1. Start
2. Display Menu
3. Choose Option
4. Perform Selected Operation
5. Return to Menu / Exit

FLOWCHART PICTURE:



SOURCE CODE

A simplified version of the source code is provided below:

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>


struct Student {

   int roll;

   char name[50];

float marks;

};


void addStudent() {

   FILE *fp = fopen("students.dat", "ab");

struct Student s;     printf("Enter Roll

Number: ");     scanf("%d", &s.roll);
```

```
printf("Enter Name: ");     scanf("
%[^\n]", s.name);     printf("Enter
Marks: ");     scanf("%f", &s.marks);

   fwrite(&s, sizeof(s), 1, fp);
fclose(fp);
   printf("Student record added successfully.\n");
}

void viewStudents() {
   FILE *fp = fopen("students.dat", "rb");
struct Student s;
   printf("\nAll Student Records:\n");
printf("Roll\tName\tMarks\n");     while
(fread(&s, sizeof(s), 1, fp)) {
printf("%d\t%s\t%.2f\n", s.roll, s.name,
s.marks);
```

```
    }
    fclose(fp);
}

void searchStudent() {
    FILE *fp = fopen("students.dat", "rb");
int roll, found = 0;     struct Student s;
    printf("Enter Roll Number to Search: ");
scanf("%d", &roll);
    while (fread(&s, sizeof(s), 1, fp)) {        if (s.roll
== roll) {            printf("Record Found:\nRoll:
%d\nName:
%s\nMarks: %.2f\n", s.roll, s.name, s.marks);
found = 1;            break;

        }
    }
    if (!found) {
```

```
        printf("No record found with Roll Number %d.\n", roll);
    }
    fclose(fp);
}

void updateStudent() {
    FILE *fp = fopen("students.dat", "rb+");
int roll, found = 0;     struct Student s;
    printf("Enter Roll Number to Update: ");
scanf("%d", &roll);

    while (fread(&s, sizeof(s), 1, fp)) {
        if (s.roll == roll) {
            found = 1;
            fseek(fp, -sizeof(s), SEEK_CUR);
printf("Enter New Name: ");
scanf(" %[^\n]", s.name);
```

```
printf("Enter New Marks: ");
scanf("%f", &s.marks);
fwrite(&s, sizeof(s), 1, fp);
            printf("Record updated successfully.\n");
break;
        }
    }
    if (!found) {
        printf("Record not found.\n");
    }
    fclose(fp);
}

void deleteStudent() {
    FILE *fp = fopen("students.dat", "rb");
FILE *temp = fopen("temp.dat", "wb");
int roll, found = 0;     struct Student s;
```

```
    printf("Enter Roll Number to Delete: ");
scanf("%d", &roll);

    while (fread(&s, sizeof(s), 1, fp)) {
if (s.roll != roll) {            fwrite(&s,
sizeof(s), 1, temp);
        } else {
found = 1;
        }
    }

    fclose(fp);     fclose(temp);
remove("students.dat");
rename("temp.dat", "students.dat");

    if (found) {
        printf("Record deleted successfully.\n");
    } else {
```

```
        printf("Record not found.\n");
    }
}

int main() {
int choice;
while (1) {
        printf("\n--- Student Record System ---\n");
printf("1. Add Student\n");
        printf("2. View Students\n");
printf("3. Search Student\n");
printf("4. Update Student\n");
printf("5. Delete Student\n");
printf("6. Exit\n");        printf("Enter
your choice: ");        scanf("%d",
&choice);
```

```
        switch (choice) {            case 1:
addStudent(); break;            case 2:
viewStudents(); break;            case 3:
searchStudent(); break;            case 4:
updateStudent(); break;            case 5:
deleteStudent(); break;            case 6:
exit(0);
            default: printf("Invalid choice. Try
again.\n");
        }
    }

    return 0; }
```

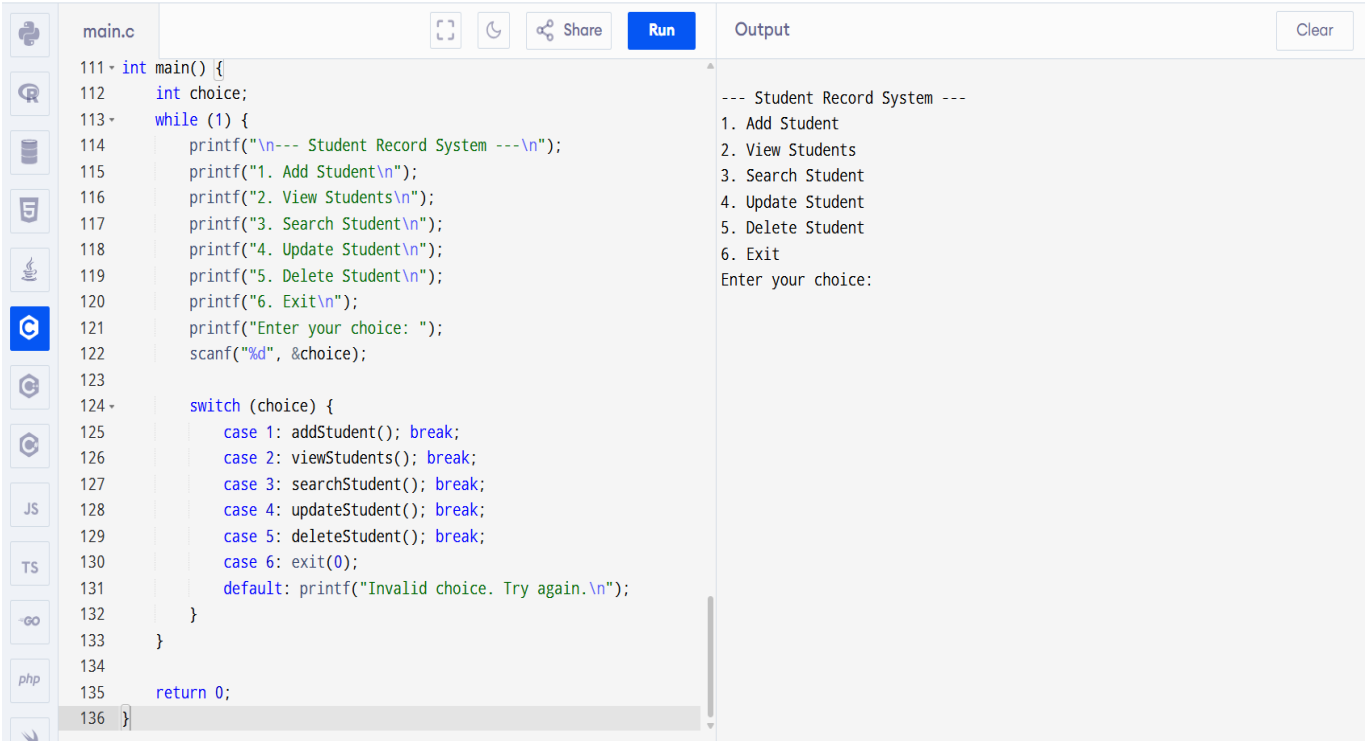
OUTPUT

The program provides a simple textbased menu where the user can:

--- Student Record System ---

1. Add Student
2. View Students
3. Search Student
4. Update Student
5. Delete Student
6. Exit

OUTPUT PICTURE:



FUTURE ENHANCEMENT

- Add GUI for better interaction - Store data in a database instead of binary files
- Add authentication for user access
- Export records to PDF or Excel - Add more fields like age, contact number, address, etc.

CONCLUSION

The Student Record Management System helped me understand file handling, structures, and user interaction in C programming. It provided a practical approach to data storage and manipulation using a simple yet efficient method.

REFERENCES

- Programming in ANSI C by E. Balagurusamy
- https://www.geeksforgeeks.org/
- https://www.tutorialspoint.com/cprogramming/
- Class notes and faculty guidance