

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>


#define STUDENT_FILE "students.txt"

#define CREDENTIAL_FILE "credentials.txt"


struct Student {

    int roll;

    char name[50];

    float marks;

};


// Global variables for logged user:

char currentUser[20];

char currentRole[10];


// Function declarations

int loginSystem();

void mainMenu();

void adminMenu();

void userMenu();

void addStudent();

void displayStudents();

void searchStudent();

void updateStudent();
```

```
void deleteStudent();
```

```
int main() {
```

```
    if (loginSystem()) {
```

```
        mainMenu();
```

```
    } else {
```

```
        printf("\nLogin failed. Exiting...\n");
```

```
    }
```

```
    return 0;
```

```
}
```

```
// ===== LOGIN SYSTEM =====
```

```
int loginSystem() {
```

```
    char username[20], password[20];
```

```
    char fileUser[20], filePass[20], fileRole[10];
```

```
    printf("===== Login Screen =====\n");
```

```
    printf("Username: ");
```

```
    scanf("%s", username);
```

```
    printf("Password: ");
```

```
    scanf("%s", password);
```

```
    FILE *fp = fopen(CREDENTIAL_FILE, "r");
```

```
    if (!fp) {
```

```
        printf("Error: credentials.txt not found!\n");
```

```
        return 0;
```

```
}
```

```
while (fscanf(fp, "%s %s %s", fileUser, filePass, fileRole) == 3) {  
    if (strcmp(username, fileUser) == 0 && strcmp(password, filePass) == 0) {  
        strcpy(currentRole, fileRole);  
        strcpy(currentUser, fileUser);  
        fclose(fp);  
        return 1;  
    }  
}
```

```
fclose(fp);  
return 0;  
}
```

```
// ===== MAIN MENU =====
```

```
void mainMenu() {  
    if (strcmp(currentRole, "ADMIN") == 0)  
        adminMenu();  
    else  
        userMenu();  
}
```

```
// ===== ADMIN MENU =====
```

```
void adminMenu() {  
    int choice;
```

```

do {

    printf("\n===== ADMIN MENU =====\n");

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

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

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

    printf("4. Update Student\n");

    printf("5. Delete Student\n");

    printf("6. Logout\n");

    printf("Enter choice: ");

    scanf("%d", &choice);


    switch (choice) {

        case 1: addStudent(); break;

        case 2: displayStudents(); break;

        case 3: searchStudent(); break;

        case 4: updateStudent(); break;

        case 5: deleteStudent(); break;

        case 6: return;

        default: printf("Invalid choice!\n");

    }

} while (1);

}


// ===== USER MENU =====

void userMenu() {

    printf("\n===== USER MENU =====\n");

```

```

printf("1. Display All Students\n");
printf("2. Search Student\n");
printf("3. Logout\n");

int choice;

scanf("%d", &choice);

switch (choice) {
    case 1: displayStudents(); break;
    case 2: searchStudent(); break;
    default: return;
}
}

// ===== ADD STUDENT =====

void addStudent() {
    FILE *fp = fopen(STUDENT_FILE, "a");
    if (!fp) {
        printf("Error opening file!\n");
        return;
    }

    struct Student st;

    printf("Enter Roll No: ");
    scanf("%d", &st.roll);

```

```

printf("Enter Name: ");
scanf("%s", st.name);
printf("Enter Marks: ");
scanf("%f", &st.marks);

fprintf(fp, "%d %s %.2f\n", st.roll, st.name, st.marks);
fclose(fp);

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

// ===== DISPLAY STUDENTS =====

void displayStudents() {
    FILE *fp = fopen(STUDENT_FILE, "r");
    if (!fp) {
        printf("No student records found!\n");
        return;
    }

    struct Student st;

    printf("\nROLL\tNAME\tMARKS\n");
    printf("-----\n");

    while (fscanf(fp, "%d %s %f", &st.roll, st.name, &st.marks) == 3) {
        printf("%d\t%s\t%.2f\n", st.roll, st.name, st.marks);
    }
}

```

```
    fclose(fp);
}

// ===== SEARCH STUDENT =====

void searchStudent() {
    int r, found = 0;
    struct Student st;

    printf("Enter roll number to search: ");
    scanf("%d", &r);

    FILE *fp = fopen(STUDENT_FILE, "r");
    if (!fp) {
        printf("File not found!\n");
        return;
    }

    while (fscanf(fp, "%d %s %f", &st.roll, st.name, &st.marks) == 3) {
        if (st.roll == r) {
            printf("\nRecord Found!\n");
            printf("Roll: %d\nName: %s\nMarks: %.2f\n", st.roll, st.name, st.marks);
            found = 1;
            break;
        }
    }
}
```

```

    if (!found)

        printf("No record found!\n");


    fclose(fp);
}


// ===== UPDATE STUDENT =====

void updateStudent() {

    int r, found = 0;

    struct Student st;


    printf("Enter roll number to update: ");

    scanf("%d", &r);


    FILE *fp = fopen(STUDENT_FILE, "r");

    FILE *temp = fopen("temp.txt", "w");


    while (fscanf(fp, "%d %s %f", &st.roll, st.name, &st.marks) == 3) {

        if (st.roll == r) {

            found = 1;

            printf("Enter new Name: ");

            scanf("%s", st.name);

            printf("Enter new Marks: ");

            scanf("%f", &st.marks);

        }
    }
}

```



```

        fprintf(temp, "%d %s %.2f\n", st.roll, st.name, st.marks);
    }

    fclose(fp);
    fclose(temp);

    remove(STUDENT_FILE);
    rename("temp.txt", STUDENT_FILE);

    if (found)
        printf("Record Updated!\n");
    else
        printf("Record Not Found!\n");
}

// ===== DELETE STUDENT =====

void deleteStudent() {
    int r, found = 0;
    struct Student st;

    printf("Enter roll number to delete: ");
    scanf("%d", &r);

    FILE *fp = fopen(STUDENT_FILE, "r");
    FILE *temp = fopen("temp.txt", "w");

```

```
while (fscanf(fp, "%d %s %f", &st.roll, st.name, &st.marks) == 3) {  
    if (st.roll != r) {  
        fprintf(temp, "%d %s %.2f\n", st.roll, st.name, st.marks);  
    } else {  
        found = 1;  
    }  
}  
  
fclose(fp);  
fclose(temp);  
  
remove(STUDENT_FILE);  
rename("temp.txt", STUDENT_FILE);  
  
if (found)  
    printf("Record Deleted!\n");  
else  
    printf("Record Not Found!\n");  
}
```