**Student Management System**

**A Mini Project**

**Academic Year: 2021 - 22 ODD SEMESTER**

**Department with Specialization : B.Tech. – Computer Science and**

**Engineering with specialization in**

**Artificial Intelligence and Machine**

**learning**

**Semester : I**

**Course Code : 18CSS101J**

**Course Title : Programming for Problem solving**

*Submitted by*

Vraj Shah [Reg No: RA2111026010275]

Devmalya Saha [Reg No: RA2111026010272]

*Under the Guidance of*

Dr. A. Suresh

(Associate Professor, NWC)



DEPARTMENT OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR- 603 203

JANUARY 2022

**AIM AND ABSTRACT**

A student management system is useful for managing scholars in a school. It is a very useful tool for maintaining students’ records. Although it has a command-line interface it is very easy to use and implement. It is a very useful project for beginners. This project has a lot of options. The project performs CRUD operations without errors.

**Features**

1. **Add Student Details:** Get data from the user and add a student to the list of students. While adding the students to the list, check for the uniqueness of the roll number.
2. **Find the student by the given roll number:** This function is to find the student record for the given roll number and print the details.
3. **Find the student by the given first name:** This function is to find all the students with the given first name and print their details.
4. **Find the students registered in a course:** This function is to find all the students who have registered for a given course.
5. **Count of Students:** This function is to print the total number of students in the system
6. **Delete a student:** This function is to delete the student record for the given roll number.
7. **Update Student:** This function is to update the student records. This function does not ask for new details for all fields but the user should be able to pick and choose what he wants to update.

**ALGORITHM**

1. STEP 1: - To declare the arrays required.
2. STEP 2: - To take the first name and the last name, roll number, CGPA, and course ID of the student
3. STEP 3: - To take the input and store it in an array.
4. STEP 4: - To take the roll number of the student as an input.
5. STEP 5: - To check the roll number and display the details like name, CGPA, course ID of the student whose roll number matches.
6. STEP 6: - To take the input from the user for the first name of the student whose details he wants to find.
7. STEP 7: - To check whose first name matches and to print the details of that student.
8. STEP 8: - To take the input from the user about the course ID.
9. STEP 9: - To check the course ID of the students present and if found display the details of the students whose course ID matches or else show no student is found with such course ID.
10. STEP 10: - To display the number of students that are counted in the class. And then subtract the number of students present in the class from 50 and print how many more the class can have.
11. STEP 11: - To take the value or the roll number of the student from the user whose data he or she wants to delete.
12. STEP 12: - To check whose roll number matches and delete every data available and display when it is done.
13. STEP 13: - To take the input from the user about the roll number whose details the user wants to update.
14. STEP 14: - To take check whose roll number matches and then take the input from the user about what he wants to update.
15. STEP 15: - To take the new data from the user and then display the updated and all the data available of the student whose data was updated.
16. STEP 16: - To take the input from the user on what he wants to do add find or delete or update or count the number of students and enter his choice.
17. STEP 17: -To check the case which matches with any of the functions like adding, deleting, updating, finding, or counting.
18. STEP 18: - To perform the function which the user chose and display the necessary output

FLOWCHART

**NO**

**LOGIN**

**YES**

SWITCH NUMBER

TRUE

BREAK STATEMENT

CASE 1

**ADD THE DETAILS OF A STUDENT**

**FIND THE DETAILS OF STUDENT BY ROLL NUMBER**

TRUE

BREAK STATEMENT

CASE 2

**FIND THE STUDENT DETAILS BY**

**FIRST NAME**

CASE 3

TRUE

BREAK STATEMENT

CASE 4

**FIND THE STUDENT DETAILS BY COURSE ID**

TRUE

BREAK STATEMENT

BREAK STATEMENT

TRUE

**FIND THE TOTAL NUMBER OF STUDENTS**

CASE 5

BREAK STATEMENT

TRUE

DELETE THE STUDENT DETAILS BY ROLL NUMBER

CASE 6

BREAK STATEMENT

TRUE

UPDATE STUDENT DETAILS BY ROLL NUMBER

CASE 7

TRUE

BREAK STATEMENT

EXIT

CASE 8

**SOURCE CODE**

#include <math.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int i = 0;

struct sinfo {

char fname[50];

char lname[50];

int roll;

float cgpa;

int cid[10];

} st[55];

void add\_student()

{

printf("Add the Students Details\n");

printf("-------------------------\n");

printf("Enter the first "

"name of student\n");

scanf("%s", st[i].fname);

printf("Enter the last name"

" of student\n");

scanf("%s", st[i].lname);

printf("Enter the Roll Number\n");

scanf("%d", &st[i].roll);

printf("Enter the CGPA "

"you obtained\n");

scanf("%f", &st[i].cgpa);

printf("Enter the course ID"

" of each course\n");

for (int j = 0; j < 5; j++) {

scanf("%d", &st[i].cid[j]);

}

i++;

}

void find\_rl()

{

int x;

printf("Enter the Roll Number"

" of the student\n");

scanf("%d",&x);

for (int j = 0; j < i; j++) {

if (x == st[j].roll) {

printf("The Students Details are\n");

printf("The First name is %s\n",

st[j].fname);

printf("The Last name is %s\n",

st[j].lname);

printf("The CGPA is %f\n",

st[j].cgpa);

for (int k = 0; k < 5; k++) {

printf(

"The course ID are %d\n",

st[j].cid[k]);

}

break;

}

}

}

void find\_fn()

{

char a[50];

printf("Enter the First Name"

" of the student\n");

scanf("%s",&a);

int c = 0;

for (int j = 0; j <= i; j++) {

// Compare the first names

if (!strcmp(st[j].fname, a)) {

printf(

"The Students Details are\n");

printf(

"The First name is %s\n",

st[j].fname);

printf(

"The Last name is %s\n",

st[j].lname);

printf(

"The Roll Number is %d\n ",

st[j].roll);

printf(

"The CGPA is %f\n",

st[j].cgpa);

printf("Enter the course ID "

"of each course\n");

// Print the course ID's

for (int k = 0; k < 5; k++) {

printf(

"The course ID are %d\n",

st[j].cid[k]);

}

c = 1;

}

}

}

void find\_c()

{

int id;

printf("Enter the course ID \n");

scanf("%d",&id);

int c = 0;

for (int j = 0; j <= i; j++) {

for (int d = 0; d < 5; d++) {

if (id == st[j].cid[d]) {

printf(

"The Students Details are\n");

printf(

"The First name is %s\n",

st[j].fname);

printf(

"The Last name is %s\n",

st[j].lname);

printf(

"The Roll Number is %d\n ",

st[j].roll);

printf(

"The CGPA is %f\n",

st[j].cgpa);

c = 1;

break;

}

}

}

}

void tot\_s()

{

printf("The total number of"

" Student is %d\n",

i);

printf("\n you can have a "

"max of 50 students\n");

printf("you can have %d "

"more students\n",

50 - i);

}

void del\_s()

{

int a;

printf("Enter the Roll Number"

" which you want to delete\n");

scanf("%d",&a);

for (int j = 0; j <= i; j++) {

if (a == st[j].roll) {

for (int k = j; k < 49; k++)

st[k] = st[k + 1];

i--;

}

}

printf("The Roll Number is "

"removed Successfully\n");

}

// Function to update a students data

void up\_s()

{

printf("Enter the roll number"

" to update the entry: ");

long int x;

scanf("%ld",&x);

for (int j = 0; j < i; j++) {

if (st[j].roll == x) {

printf("1. first name\n"

"2. last name\n"

"3. roll no.\n"

"4. CGPA\n"

"5. courses\n");

int z;

scanf("%d",&z);

switch (z) {

case 1:

printf("Enter the new first name : \n");

scanf("%s", st[j].fname);

break;

case 2:

printf("Enter the new last name : \n");

scanf("%s", st[j].lname);

break;

case 3:

printf("Enter the new roll number : \n");

scanf("%d", &st[j].roll);

break;

case 4:

printf("Enter the new CGPA : \n");

st[j].cgpa = 9;

break;

case 5:

printf("Enter the new courses \n");

scanf("%d%d%d%d%d", &st[j].cid[0],

&st[j].cid[1], &st[j].cid[2],

&st[j].cid[3], &st[j].cid[4]);

break;

}

printf("UPDATED SUCCESSFULLY.\n");

}

}

}

int main()

{

int choice, count;

while (i<=50)

{

printf("The Task that you "

"want to perform\n");

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

printf("2. Find the Student "

"Details by Roll Number\n");

printf("3. Find the Student "

"Details by First Name\n");

printf("4. Find the Student "

"Details by Course Id\n");

printf("5. Find the Total number"

" of Students\n");

printf("6. Delete the Students Details"

" by Roll Number\n");

printf("7. Update the Students Details"

" by Roll Number\n");

printf("8. To Exit\n");

printf("Enter your choice to "

"find the task\n");

scanf("%d", &choice);

switch (choice) {

case 1:

add\_student();

break;

case 2:

find\_rl();

break;

case 3:

find\_fn();

break;

case 4:

find\_c();

break;

case 5:

tot\_s();

break;

case 6:

del\_s();

break;

case 7:

up\_s();

break;

case 8:

exit(0);

break;

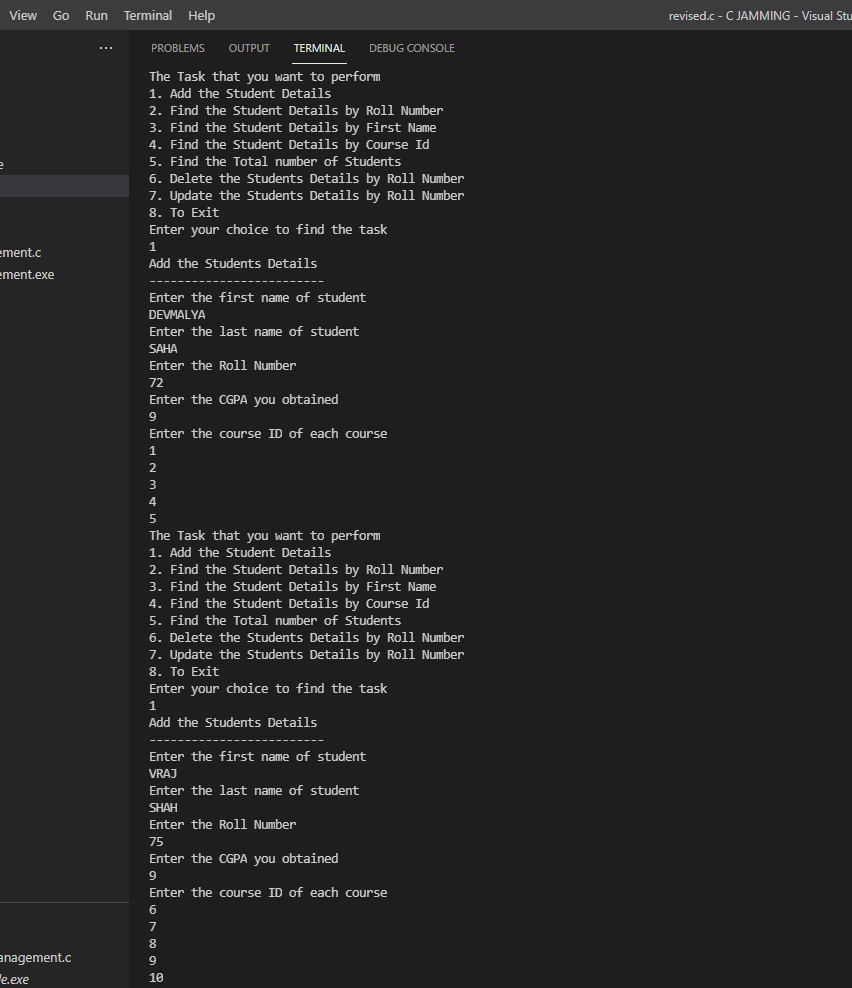
default:

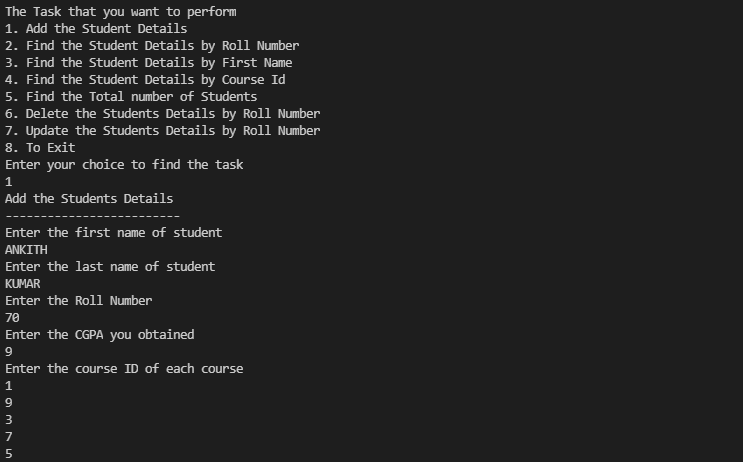
break;

}

}

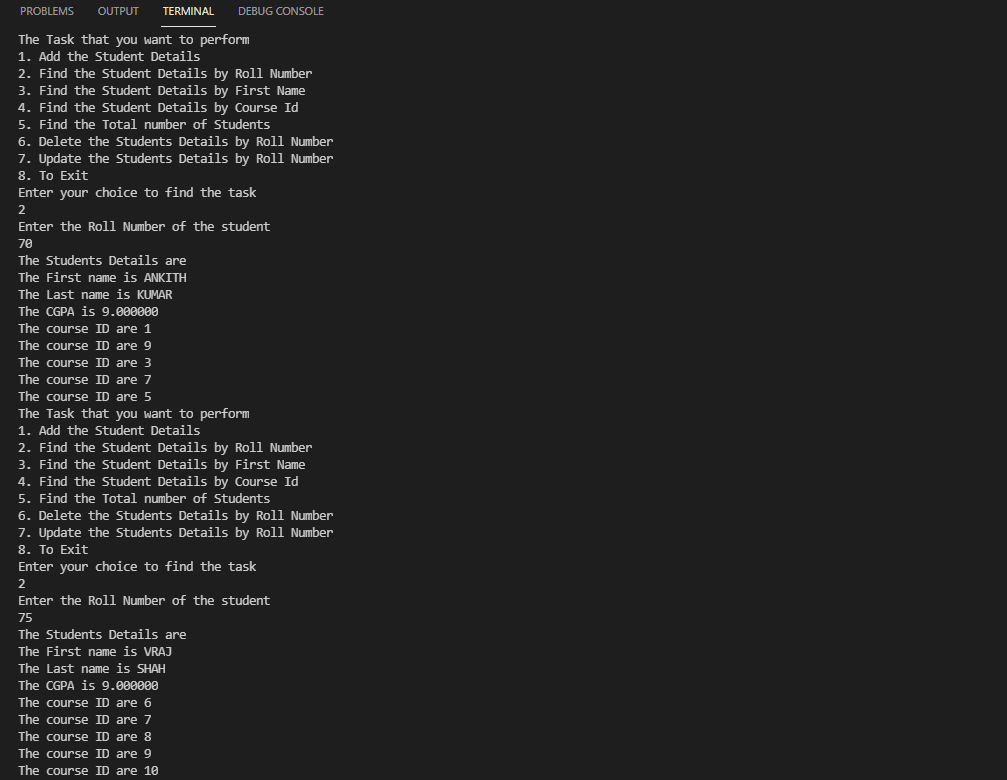
**SCREENSHOT AND OUTPUTS**

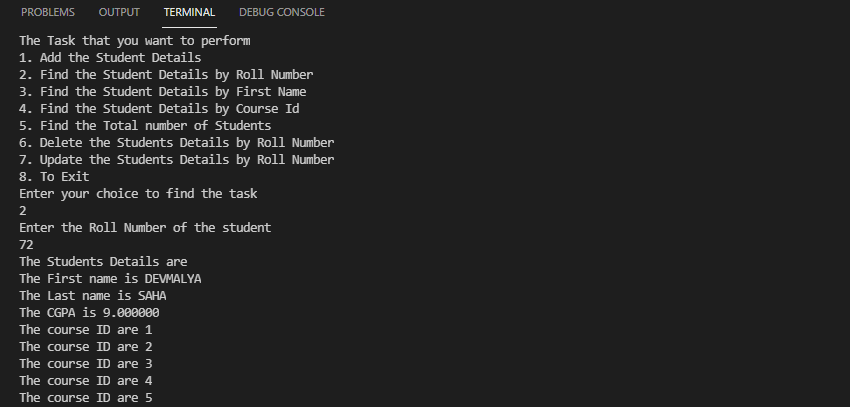
****

****

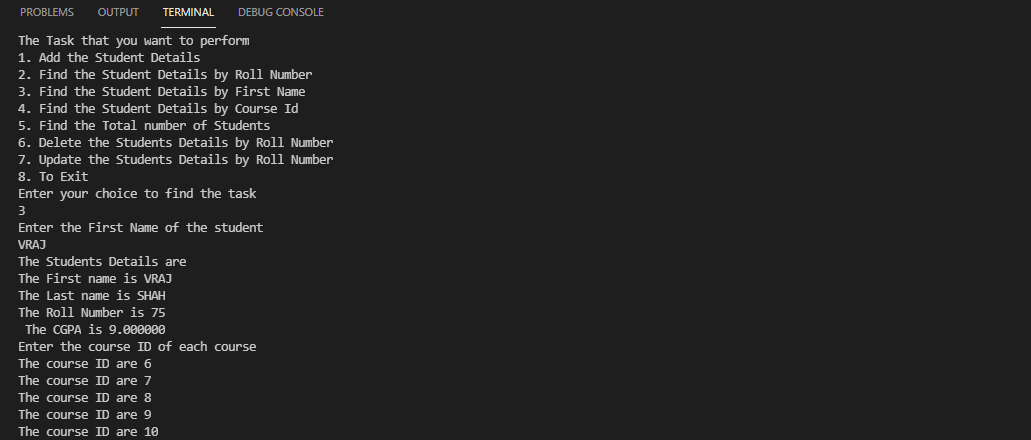
HERE, WE CAN SEE HOW THE FUNCTION TO ADD DETAILS OF THE STUDENTS IS RUNNING

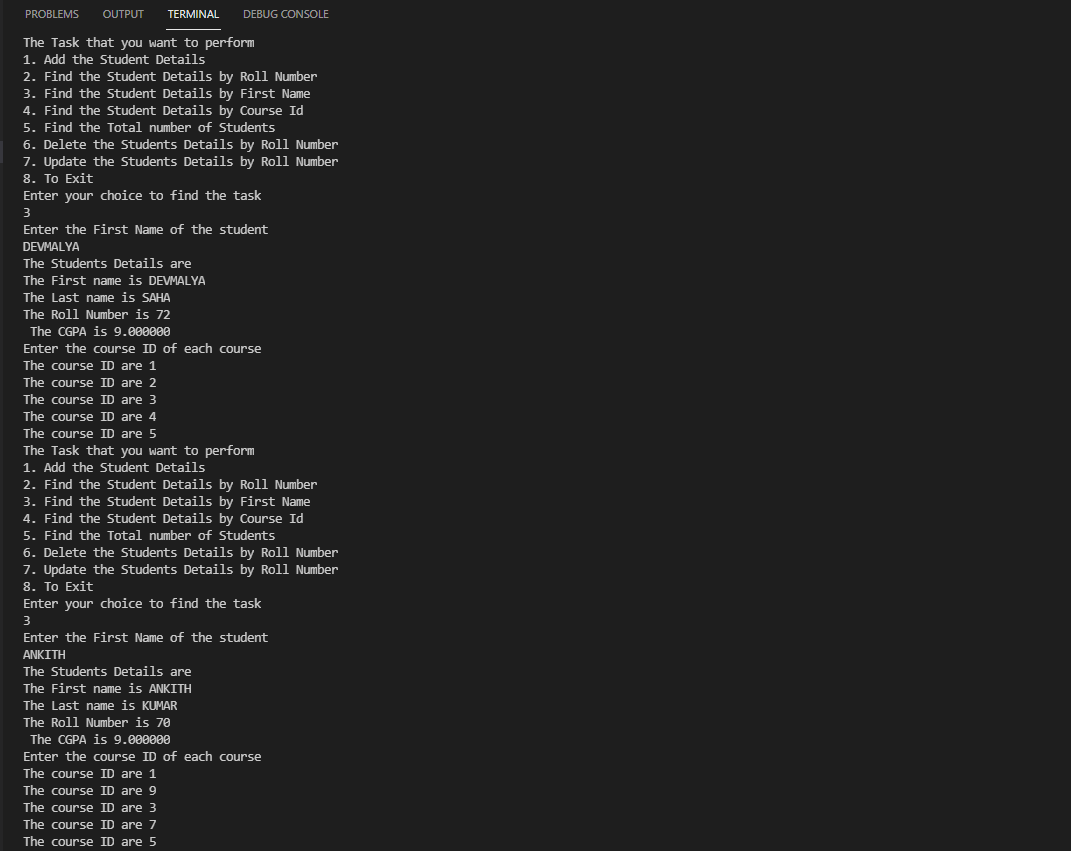
HERE WE CAN SEE HOW THE FUNCTION OF ADDING STUDENTS DETAILS IS RUNNING

****

****

HERE, WE CAN SEE HOW THE FUNCTION TO FIND THE STUDENT DETAIL BY THEIR ROLL NUMBER IS RUNNING SUCCESSFULLY

****

****

THE FUNCTION TO SEARCH STUDENT DETAILS BY THEIR FIRST NAME IS RUNNING SUCCESSFULLY

****

****

THE FUNCTION TO FIND THE STUDENT DETAILS BY ENTERING THEIR COURSE ID AND FIND THE STUDENTS REGISTERED IN THAT COURSE IS RUNNING SUCCESSFULLY

****

HERE WE CAN SEE THAT THE FUNCTION IS RUNNING PERFECTLY

****

SO, WE CAN SEE HERE THE FUNCTION TO FIND THE NUMBER OF STUDENTS IS WORKING PERFECTLY AND IT IS SHOWING 2 STUDENTS BECAUSE IN THE PREVIOUS FUNCTION WE HAVE REMOVED THE STUDENT DETAILS OF ROLL NUMBER 72 SO BOTH THE FUNCTIONS ARE RUNNING PERFECTLY

****

HERE, WE CAN SEE THAT THE FUNCTION TO UPDATE THE DETAILS OF THE STUDENTS AFTER SEARCHING BY THEIR ROLL NUMBER IS WORKING PERFECTLY AS AFTER SEARCHING WITH THE ROLL NUMBER THE STUDENT DETAILS WE CAN SEE THAT THE ROLL HAS CHANGED FROM 75 TO 63

**RESULT**

The given source code gives the result as expected during its runtime from the developers and users alike.

**CONCLUSION**

Student Management system can be used by educational institutions to maintain their student records easily. Achieving this objective is difficult using the manual system as the information is scattered, can be redundant, and collecting relevant information may be very time-consuming. All these problems are solved by this project.

This system helps in maintaining the information of pupils of the organization. It can be easily accessed by the manager and kept safe for a long period without any changes.