VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANASANGAMA, BELAGAVI – 590018

KARNATAKA

Assignment Report
On
"SOFTWARE FOR STUDENT INFORMATION MANAGEMENT SYSTEM"

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DATA STRUCTURES AND APPLICATIONS (BCS304) COURSE OF III SEMESTER

Submitted by:

SUPREETH M R

(1CG23CS106)

Guide:
Mr.Asif Ulla Khan .M. Tech.
Asst. Prof., Dept. of CSE
C.I. T, Gubbi.

HOD:
Dr. Shantala C P PhD.,
Head, Dept. of CSE
C.I.T, Gubbi.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Channabasaveshwara Institute of Technology

(Affiliated to VTU, Belgaum & Approved by AICTE, New Delhi)

(ISO 9001:2015 Certified Institution)

NH 206 (B.H. Road), Gubbi, Tumkur – 572 216. Karnataka.

2024-25

Rubric – B.E. Mini-Project [BCS304]

Course outcome	Rubric/Level	Excellent (91-100%)	Good (81-90%)	Average (61-80%)	Moderate (40-60%)	Score
CO1	Identification of project proposal (05 Marks)					
CO2	Design and Implementation (10 Marks)					
CO3	Presentation skill (05 Marks)					
CO4	Report (05 Marks)					
Total						

Course outcome:

CO 1: Identification of project proposal which is relevant to subject of engineering.

CO 2: Design and implement proposed project methodology.

CO 3: Effective communication skill to assimilate their project work.

CO 4: Understanding overall project progress and performance.

Student Signature **Faculty signature**

INDEX

ABSTRACT

A student management system is a software application that helps schools and universities to manage and maintain the data of their students. It allows the users to perform various operations such as adding, finding, and displaying the students based on their name, roll number, or course ID. The system also stores the students' grades, attendance, and other records. The system is implemented using the C programming language, and uses a menu-driven approach to provide a user-friendly interface.

CHAPTER 1:

INTRODUCTION

- The purpose of this project is to design and develop a student management system using the C programming language.
- The system is intended to help the educational institutions to reduce the workload of managing the student data manually, and to improve the efficiency and accuracy of the data processing.
- The system consists of a structure that defines the attributes of a student, such as first name, last name, roll number, CGPA, and course ID.
- The system also provides some functions that allow the users to perform various operations on the student data, such as adding a new student, finding a student by name, roll number, or course ID, and displaying the details of the students enrolled in a particular course.
- The system uses a menu-driven approach to provide a user-friendly interface, where the users can choose from different options to perform the desired operation.
- The system also validates the user input and handles the possible errors and exceptions.

  Approach: The idea is to form an individual functions for every operation. All the functions are unified together to form software.

1. Add Student Details: Get data from user and add a student to the list of students. While adding the students into 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.

- The student management system in the C program uses an array of structures to store and manipulate student data effectively.

CHAPTER 2:

PROBLEM STATEMENT

Write a program to build a simple Software for Student Information Management System which can perform the following operations:

- Store the First name of the student.
- Store the Last name of the student.
- Store the unique Roll number for every student.
- Store the CGPA of every student.
- Store the courses registered by the student.

CHAPTER 3:

IMPLEMENTATION

```
// C program for the implementation of
// menu driven program for student
// management system
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
// Variable to keep track of
// number of students
int i = 0;
int k,n;
// Structure to store the student
struct sinfo {
    char fname[50];
    char lname[50];
    int roll;
    float  cgpa;
    int cid[10];
} st[55];
// Function to add the student
void add_student()
{
    for(k=0;k<n;k++)
    {
    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 < 2; j++) {
        scanf("%d", &st[i].cid[j]);
    }
    i = i + 1;
    }
}
// Function to find the student
// by the roll number
void find_rl()
{
    int x;
    printf("Enter the Roll Number of the student\n");
    scanf("%d", &x);
    for (int j = 1; j <= i; j++) {
        if (x == st[i].roll) {
            printf("The Students Details are\n");
            printf("The First name is %s\n",st[i].fname);
            printf( "The Last name is %s\n", st[i].lname);
            printf("The CGPA is %f\n",st[i].cgpa);
            printf( "Enter the course ID of each course\n");
        }
        for (int j = 0; j < 5; j++) {
            printf( "The course ID are %d\n", st[i].cid[j]);
        }
        break;
    }
}
// Function to find the student
// by the first name
void find_fn()
{
    char a[50];
```

```
    printf("Enter the First Name of the student\n");
    scanf("%s", a);
    int c = 0;
    for (int j = 1; j <= i; j++) {
        if (!strcmp(st[j].fname, a)) {
            printf("The Students Details are\n");
            printf("The First name is %s\n", st[i].fname);
            printf( "The Last name is %s\n",st[i].lname);
            printf( "The Roll Number is %d\n ",st[i].roll);
            printf("The CGPA is %f\n", st[i].cgpa);
            printf( "Enter the course ID of each course\n");
            for (int j = 0; j < 5; j++) {
                printf("The course ID are %d\n",st[i].cid[j]);
            }
            c = 1;
        }
        else
            printf("The First Name not Found\n");
    }
}
// Function to find
// the students enrolled
// in a particular course
void find_c()
{
    int id;
    printf("Enter the course ID \n");
    scanf("%d", &id);
    int c = 0;
    for (int j = 1; 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[i].fname);
                printf( "The Last name is %s\n",st[i].lname);
```

```
            printf("The Roll Number is %d\n ",st[i].roll);
            printf("The CGPA is %f\n",st[i].cgpa);
            c = 1;
            break;
        }
        else
            printf("The First Name not Found\n");
    }
  }
}
// Function to print the total
// number of students
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);
}
// Function to delete a student
// by the roll number
void del_s()
{
  int a;
  printf("Enter the Roll Number which you want to delete\n");
  scanf("%d", &a);
  for (int j = 1; 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
```

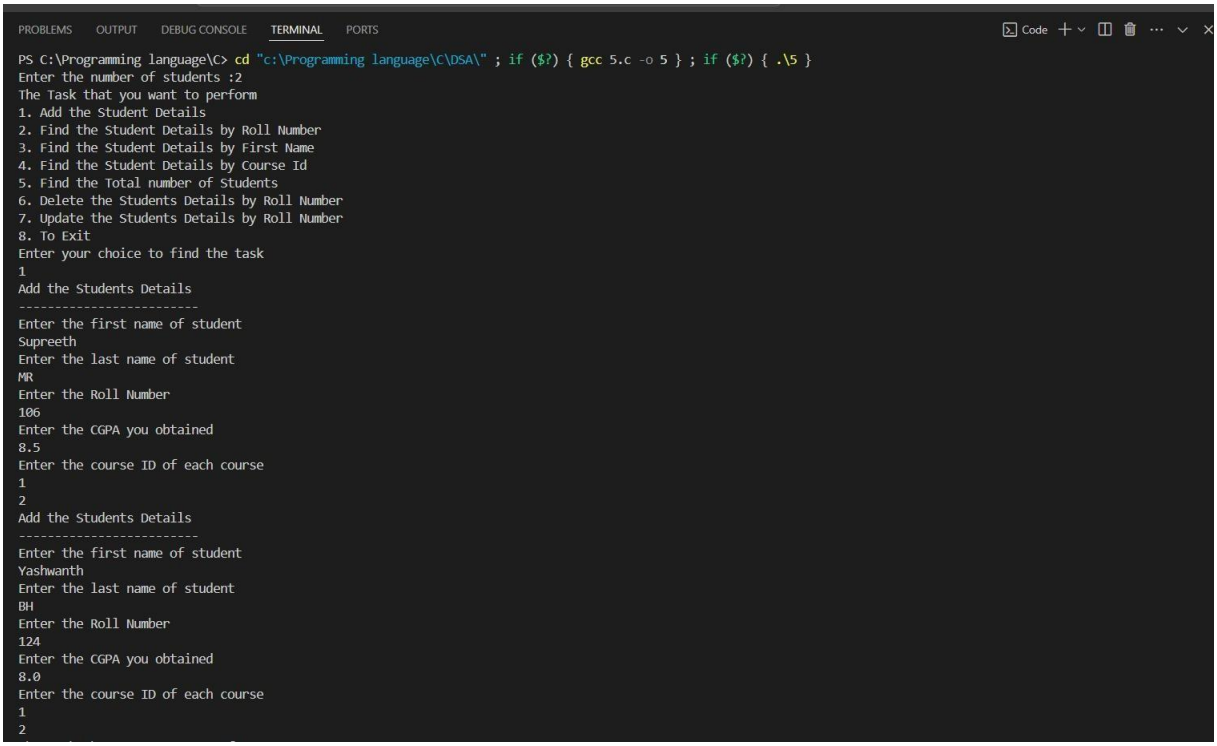
```
void up_s()
{
   printf("Enter the roll number to update the entry : ");
   long int x;
   scanf("%ld", &x);
   for (int j = 0; j < i; j++) {
      if (st[j].roll == x) {
         printf("1. first name\n2. last name\n3. roll no.\n4. CGPA\n5.courses\n");
         int z;
         scanf("%d", &z);
         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");
            scanf("%f", &st[j].cgpa);
            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");
      }
```

```
    }
}
// Driver code
void main()
{
    int choice, count;
    printf("Enter the number of students :");
    scanf("%d",&n);
    while (i = 1) {
        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;
        }
    }
}
```

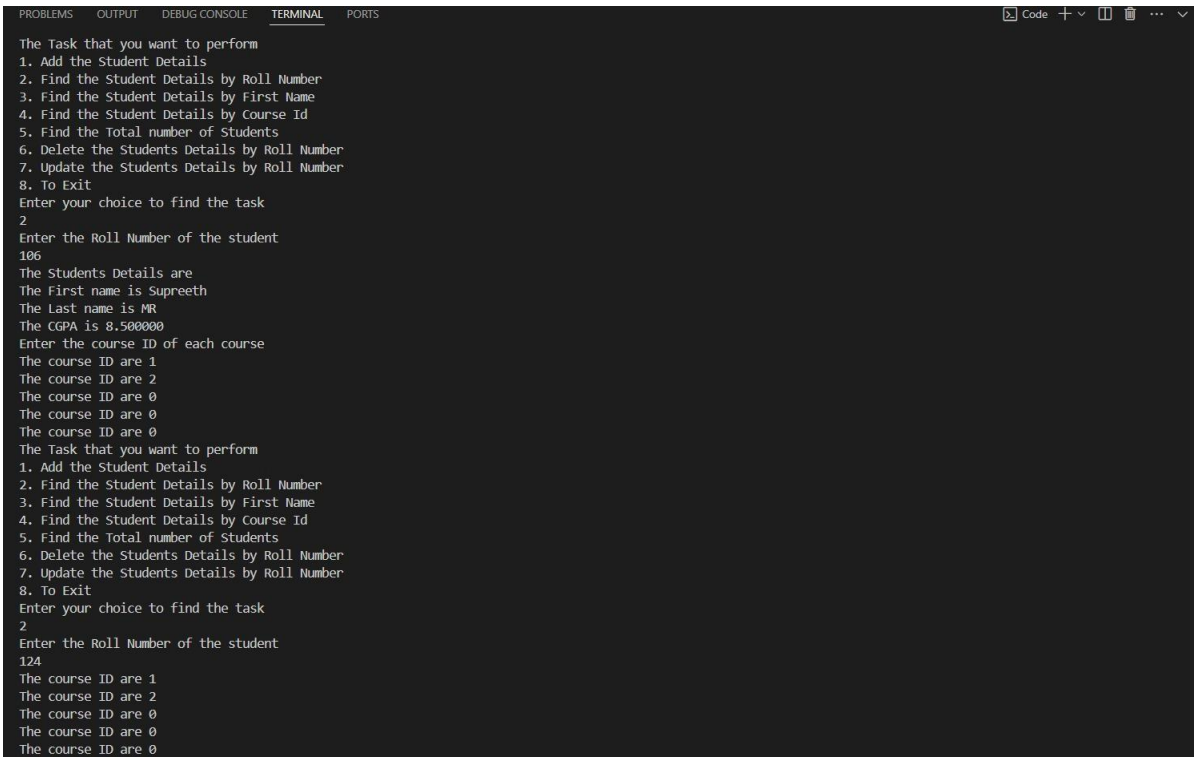
CHAPTER 4:

RESULTS/SCREEN SHOT



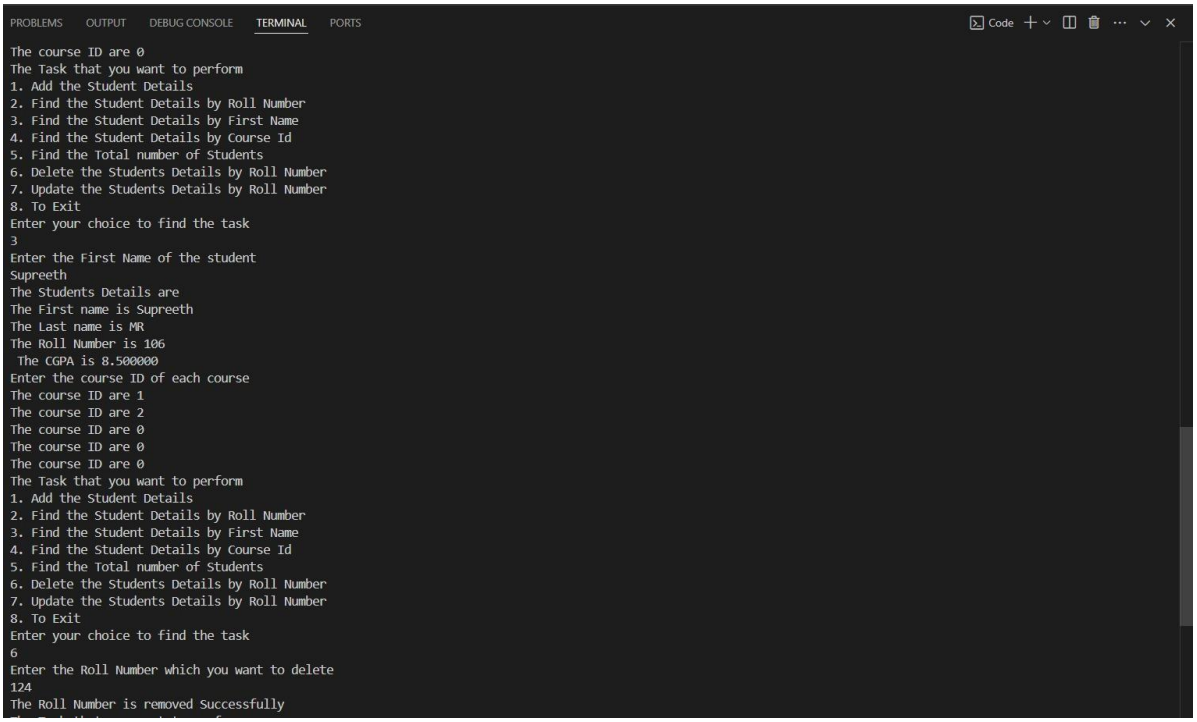
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                                   Code

PS C:\Programming language\C> cd "c:\Programming language\C\DSA\" ; if ($?) { gcc 5.c -o 5 } ; if ($?) { .\5 }
Enter the number of students :2
The Task that you want to perform
1. Add the Student Details
2. Find the Student Details by Roll Number
3. Find the Student Details by First Name
4. Find the Student Details by Course Id
5. Find the Total number of Students
6. Delete the Students Details by Roll Number
7. Update the Students Details by Roll Number
8. To Exit
Enter your choice to find the task
1
Add the Students Details
------------------------
Enter the first name of student
Supreeth
Enter the last name of student
MR
Enter the Roll Number
106
Enter the CGPA you obtained
8.5
Enter the course ID of each course
1
2
Add the Students Details
------------------------
Enter the first name of student
Yashwanth
Enter the last name of student
BH
Enter the Roll Number
124
Enter the CGPA you obtained
8.0
Enter the course ID of each course
1
2
```



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                                   Code

The Task that you want to perform
1. Add the Student Details
2. Find the Student Details by Roll Number
3. Find the Student Details by First Name
4. Find the Student Details by Course Id
5. Find the Total number of Students
6. Delete the Students Details by Roll Number
7. Update the Students Details by Roll Number
8. To Exit
Enter your choice to find the task
2
Enter the Roll Number of the student
106
The Students Details are
The First name is Supreeth
The Last name is MR
The CGPA is 8.500000
Enter the course ID of each course
The course ID are 1
The course ID are 2
The course ID are 0
The course ID are 0
The course ID are 0
The Task that you want to perform
1. Add the Student Details
2. Find the Student Details by Roll Number
3. Find the Student Details by First Name
4. Find the Student Details by Course Id
5. Find the Total number of Students
6. Delete the Students Details by Roll Number
7. Update the Students Details by Roll Number
8. To Exit
Enter your choice to find the task
2
Enter the Roll Number of the student
124
The course ID are 1
The course ID are 2
The course ID are 0
The course ID are 0
The course ID are 0
```


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Code
The course ID are 0
The Task that you want to perform
1. Add the Student Details
2. Find the Student Details by Roll Number
3. Find the Student Details by First Name
4. Find the Student Details by Course Id
5. Find the Total number of Students
6. Delete the Students Details by Roll Number
7. Update the Students Details by Roll Number
8. To Exit
Enter your choice to find the task
3
Enter the First Name of the student
Supreeth
The Students Details are
The First name is Supreeth
The Last name is MR
The Roll Number is 106
The CGPA is 8.500000
Enter the course ID of each course
The course ID are 1
The course ID are 2
The course ID are 0
The course ID are 0
The course ID are 0
The Task that you want to perform
1. Add the Student Details
2. Find the Student Details by Roll Number
3. Find the Student Details by First Name
4. Find the Student Details by Course Id
5. Find the Total number of Students
6. Delete the Students Details by Roll Number
7. Update the Students Details by Roll Number
8. To Exit
Enter your choice to find the task
6
Enter the Roll Number which you want to delete
124
The Roll Number is removed Successfully


The Task that you want to perform
1. Add the Student Details
2. Find the Student Details by Roll Number
3. Find the Student Details by First Name
4. Find the Student Details by Course Id
5. Find the Total number of Students
6. Delete the Students Details by Roll Number
7. Update the Students Details by Roll Number
8. To Exit
Enter your choice to find the task
5
The total number of Student is 1
you can have a max of 50 students
you can have 49 more students
The Task that you want to perform
1. Add the Student Details
2. Find the Student Details by Roll Number
3. Find the Student Details by First Name
4. Find the Student Details by Course Id
5. Find the Total number of Students
6. Delete the Students Details by Roll Number
7. Update the Students Details by Roll Number
8. To Exit
Enter your choice to find the task
8

CHAPTER 5:

CONCLUSIONS

The implementation of the menu-driven student management system in C offers a practical solution for managing student records in educational settings. Through the functionalities provided, including adding, searching, deleting, and updating student details, the program addresses the fundamental requirements of student administration.

The project aims to simplify and automate the tasks related to managing student records and improve the efficiency and accuracy of the system. The project is user-friendly, secure, and reliable. The project can be further enhanced by adding more features and functionalities, such as graphical user interface, database connectivity, and report generation.

Despite its simplicity, the program demonstrates key concepts of file handling, data structures, and user interaction, making it a valuable learning tool for students and developers alike. The modular design allows for easy scalability and customization to meet specific institutional needs. Overall, this project The serves as a foundational step towards developing more comprehensive and sophisticated student management systems, contributing to the improvement of educational administration processes.