DSA [20ES117] COURSE PROJECT REPORT

On

STUDENT MANAGEMENT SYSTEM

Developed By:

H.T.NO :-2103A54023

STUDENT NAME :KOTHAKONDA SRISHANKA

Under the Guidance of

Dr. A. SIVA KRISHNA REDDY

Associate Professor

Submitted to

Department Computer Science and Artificial Intelligence

SR University

Ananthasagar(V), Hasanparthy(M), Hanamkonda(Dist.) – 506371

www.sru.edu.in

December 2022

Department of Computer Science and Artificial Intelligence

CERTIFICATE

This is to certify that the DSA course project report entitled **"STUDENT MANAGEMENT SYSTEM"** is a record of bonafide work carried out by the student **"KOTHAKONDA SRISHANKA"** bearing roll number **"2103A54023"** of Computer Science and Artificial Intelligence department during the academic year 2022-23.

Supervisor

(Dr. A. Siva Krishna Reddy)

ABSTRACT

- **This project is on "School student Management System" aiming to provide automation support of Instituition. It reduces complexity through automation of student details .It helps to improve management of student data with the help of this system the user of this system, which is the department i.e administrative will be able to maintain the data through the computer which will help in increasing his efficiency, accuracy of the department as well as help in electronic inter change of the data. Administrative department can use this system to check for the details of the student and study the past records to analyse the total no of students passed out from their respective organizations.**

PROBLEM STATEMENT:

For every school admins, the integral of their very existence lies on the most important issue of large paperwork .There is a problem with traditional Instituition management because it didn't have a systematic way to store information about students .When the staff or administrator wants to record the data of a student they have to fill out a paper form. By this process the data might be lost and there are a lot of problems at the same time.

- **Another problem that can happen is that it can be hard to find and update information about students and class arrangement.**

INTRODUCTION

Schools and universities are the foundation of knowledge and an educational body on which student rely upon. Therefore, they need to maintain a proper database of its students to keep all the updated records and easily share information with students. So inorder to make that happen I have developed"student Management System "which reduces the time complexity and workload of the management .

To developed this code I have used c language with the concept of linked list from data structures and algorithm.

LINKED LIST is a linear data structure. Every linked list has two parts , the data section and the address section that holds the address of the next element in the list which is called node. The size of linked list is not fixed and data items can be added at any location in the list, thus because of this reason it is friendly to develop student management system using this.

PROPOSED SYSTEM

To overcome these problems I have come up with a solution to store information in online mode by providing facilities :-

1.To insertion of student details.

2 .To search student details .

3 .To update particular student details.

4 .To maintain total count of students.

5.To delete student details.

By using this method information last forever without any loss of student details and it reduces workload of staff or administrator.

KNOWLEDGE REQUIRED TO DEVELOP THIS APPLICATION

- **Control statements (if ,else, switch case)**
- **Loop statements(while, for)**
- **String (to store student name).**
- **Functions (any type of use defined functions)**
- **Structure(store student details)**
- **Dynamic memory allocatio(malloc)**

Modules Included

In this application some variables and structure members are declared globally and some are declared locally so that these variables and structure members can be accessed in the program at any function call . We can choose any function by using function calls which are declared in switch-case. Inorder to repeat the loop control

statement while loop is used with condition .The memory allocation will be done in the program dynamically .The application asks the person who runs the program that whether they want to insert ,update ,search, count or delete student details.

IN THIS APPLICATION WE HAVE FIVE MODULES

1) **CREATE:-**

In this module first it asks to enter no of student details you wanted to create then followed by the first name ,last name ,class , marks scored in all six subject s.

2) **Insertion:-**

In this module first it ask whether you want to insert student details at begin ,end ,specified position. Then followed by the enter of student details.

3) **Search:-**

In this module it asks us to enter rollno of that particular student whose details you wanted to check and finally it display the details of student after searching , if that particular student details are not present then display as student don't belong to the school.

4) **Delete:-**

In this module it ask whether you wanted to delete student details of student at begin, end or specified position then delete that student details and displays the deleted details on screen.

5) **Count:-**

In this it count total no of students records stored by that particular school then displays that number on screen.

SOURCE CODE [HEADER FILE]

```
 void deleteatspecified();
void create_new();
void display();
void countrecord();
void searchrecord();
void updaterecord();
struct node
{
        int roll,class;
        char name[30],firstname[30];
        float total_marks,percentage;
        struct node *next;
};
struct node *start=NULL;
main()
{
```

```
int ch;
printf("1.create new student record \n2.insert student details\n3.search a student record\n4.count total students\n5.delete student record\n6.To show all records\n7.To update particular student details.\n8.exit\n");
while(1)
{
printf("\nenter your choice:");
scanf("%d",&ch);
switch(ch)
{
case 1:create_new();
break;
case 2:ins_record();
break;
case 3:searchrecord();
break;
case 4:countrecord();
break;
case 5:deleterecord();
```

```
                    break;

                case6:searchrecord();
                    break;
                case 7:updaterecord();
                    break;
                case 8: exit(0);
                default:printf("\n invalid choice");
            }
        }
}
void ins_record()
{
        int ch;
        printf("To insert a student record at\n1.Begin\n2.End\n3.Specified\n");
        {
            printf("enter your choice:");
          scanf("%d",&ch);
```

```
            switch(ch)
            {
                  case 1:ins_record_beggining();
                        break;
                  case 2:ins_record_end();
                        break;
                  case 3:ins_record_specified();
                        break;
                  default:printf("\n invalid choice");

            }
      }
}
void updaterecord()
{
      struct node *temp,*ptr;
      int loc,count=1;
      if(start==NULL)
```

```
{printf("\nstudent details are not entered\n");
}
else
{
   printf("enter the roll of student  to be updated:");
        scanf("%d",&loc);
        temp=start;
        count=1;
        while(temp!=NULL && temp->roll!=loc)
        {
                ptr=temp;
                count=count+1;
                temp=temp->next;
        }
                if(temp==NULL)
                {
                        printf("\nparticular student don't belongs to our school\n");
                }
```

```
else
{
        ptr->next=temp->next;
        int ch;
        printf("\nTo update press \n1.first name\n2.last name\n3.roll-no\n4.class\n5.total marks\n6.percentagecentage\n");
        printf("enter the user choice:");
        scanf("%d",&ch);
        switch(ch)
        {
            case 1:printf("\nenter the new first name\n");
                   scanf("%s",temp->firstname);
                   break;
            case 2:printf("enter the new last name\n");
                   scanf("%s",temp->name);
                   break;
            case 3:printf("enter the new roll\n");
                   scanf("%d",&temp->roll);
```

```
                    break;
                case 4:printf("enter the new class\n");
                    scanf("%d",&temp->class);
                    break;
                case 5:printf("enter the new total marks\n");
                    scanf("%f",&temp->total_marks);
                    break;
                case 6:printf("enter the new percentagecentage\n");
                    scanf("%f",&temp->percentage);
                    break;
                case 7:exit(0);
                default:printf("invalid choice\n");
            }
        }
    }
}
void display()
{
```

```
struct node *temp;
if(start==NULL)
{
        printf("\n list is empty");
}
else
{
        temp=start;
        printf("\nstudent details  are\n");
        int i=0;
        while(temp!=NULL)
        {
                i++;
                printf("\n\nStudent-%d",i);

printf("\nRoll.NO:%d\nFirstName:%s\nFirstName:%s\nClass:%d\nTotalMarks:%f\npercentage:%f\n",temp->roll,temp->firstname,temp->name,temp->class,temp->total_marks,((temp->percentage)/600)*100);
                temp=temp->next;
```

```
                }
        }
}
void countrecord()
{
        struct node *newnode,*temp,*ptr;
        int count=0;
        newnode=(struct node*)malloc(sizeof(struct node));
        if(start==NULL)
        {
                printf("\nstudent details are not entered\n");
        }
        else
        {
                temp=start;
                while(temp!=NULL)
                {
                        count++;
```

```
                temp=temp->next;
            }
            if(count==0)
            {
                printf("\nstudent have not yet joined\n");
            }
            else
            {
                printf("\nNumber of student :%d\n",count);
            }
        }
}
void searchrecord()
{
        struct node *temp,*ptr;
        int loc,count=1;
        if(start==NULL)
        {
```

```
        printf("\nstudent details are not entered\n");
}
else
{

        printf("\nenter the roll no of student");
        scanf("%d",&loc);
        temp=start;
        count=1;
        while(temp!=NULL && temp->roll!=loc)
        {
                ptr=temp;
                count=count+1;
                temp=temp->next;
        }
                if(temp==NULL)
                {
                        printf("\nparticular student don't belongs to our school\n");
```

```
                }
                  else
                  {
                  ptr->next=temp->next;
                  printf("\n\nRoll.NO:%d\nFirstName:%s\nFirst
Name:%s\nClass:%d\nTotal Marks:%f\npercentagecentage:%f\n", temp->roll,temp-
>firstname,temp->name,temp->class,temp->total_marks,((temp>percentage)/600)*100);
                        return;
                  }
            }
}
void ins_record_beggining()
{
      struct node *newnode;
      newnode=(struct node*)malloc(sizeof(struct node));
      if(newnode==NULL)
      {
            printf("\n memory is not allocated\n");
      }
```

```
else
{
    int m1,m2,m3,m4,m5,m6,r,a;
    printf("\n\n***enter student to be inserted details***\n\n");
     printf("roll:");
            scanf("%d",&r);
            printf("First Name:");
            scanf("%s",newnode->firstname);
            printf("Last Name:");
            scanf("%s",newnode->name);
            printf("Class:");
            scanf("%d",&newnode->class);
            printf("Marks in Order:");
            scanf("%d%d%d%d%d%d",&m1,&m2,&m3,&m4,&m5,&m6);
                    newnode->roll=r;
                    newnode->class=a;
                    newnode->total_marks=(m1+m2+m3+m4+m5+m6);
                newnode->percentage=((newnode->total_marks)/100)*100;
```

```
                newnode->next=NULL;
                    if(start==NULL)
                {
                    start=newnode;
                    }
                    else
                    {
                            newnode->next=start;
                            start=newnode;
                    }
   }
}
void create_new()
{
        int total,i;
        struct node *temp,*newnode;

          printf("enter the number of record you want to create:");
```

```
scanf("%d",&total);
printf("\n\n***enter student details***\n\n");
if(total<=0)
{
      printf("list must be greater than zero");
}
else
{
      int a,r,m1,m2,m3,m4,m5,m6;
 temp=(struct node*)malloc(sizeof(struct node));
        printf("roll:");
          scanf("%d",&r);
          printf("First Name:");
          scanf("%s",temp->firstname);
          printf("Last Name:");
          scanf("%s",temp->name);
          printf("Class:");
          scanf("%d",&a);
```

```
  printf("Marks in Order:");
  scanf("%d%d%d%d%d%d",&m1,&m2,&m3,&m4,&m5,&m6);
            temp->roll=r;
            temp->class=a;
            temp->total_marks=(m1+m2+m3+m4+m5+m6);
      temp->percentage=((temp->total_marks)/100)*100;
      temp->next=NULL;
   start=temp;
   for(i=2;i<=total;i++)
   {
         newnode=(struct node*)malloc(sizeof(struct node));
         int r,a,m1,m2,m3,m4,m5,m6;
printf("roll:");
  scanf("%d",&r);
  printf("First Name:");
  scanf("%s",newnode->firstname);
  printf("Last Name:");
  scanf("%s",newnode->name);
```

```
                printf("Class:");
                scanf("%d",&newnode->class);
                printf("Marks in Order:");
                scanf("%d%d%d%d%d%d",&m1,&m2,&m3,&m4,&m5,&m6);
                        newnode->roll=r;
                        newnode->class=a;
                        newnode->total_marks=(m1+m2+m3+m4+m5+m6);
                    newnode->percentage=((newnode->total_marks)/100)*100;
                    newnode->next=NULL;
                    temp->next=newnode;
                    temp=newnode;
                       }

            }
      }
void ins_record_end()
{
  struct node *newnode,*temp;
```

```
newnode=(struct node*)malloc(sizeof(struct node));
if(newnode==NULL)
{
        printf("\nmemory is not allocated");
}
else
{

                int r,m1,m2,m3,m4,m5,m6;
            float m;
            printf("roll:");
            scanf("%d",&r);
            printf("First Name:");
            scanf("%s",newnode->firstname);
            printf("Last Name:");
            scanf("%s",newnode->name);
            printf("Class:");
            scanf("%d",&newnode->class);
```

```
 printf("Marks in Order:");
 scanf("%d%d%d%d%d%d",&m1,&m2,&m3,&m4,&m5,&m6);
newnode->roll=r;
  newnode->total_marks=(m1+m2+m3+m4+m5+m6);
    newnode->percentage=((newnode->total_marks)/100)*100;
    newnode->next=NULL;
    if(start==NULL)
    {
      start=newnode;
              }
              else
              {
                      temp=start;
                      while(temp->next!=NULL)
                      {
                              temp=temp->next;
                       }
                       temp->next=newnode;
```

```
                                        }
 }
}
void ins_record_specified()
{
   struct node *newnode,*temp,*ptr;
    int item,loc,count=1,m1,m2,m3,m4,m5,m6;
    newnode=(struct node*)malloc(sizeof(struct node));
     if(newnode==NULL)
          {
                printf("\nmemory is not allocated");
          }
          else
          {
                    int r;
               printf("roll:");
                  scanf("%d",&r);
                  printf("First Name:");
```

```
            scanf("%s",newnode->firstname);
            printf("Last Name:");
            scanf("%s",newnode->name);
            printf("Class:");
            scanf("%d",&newnode->class);
            printf("Marks in Order:");
            scanf("%d%d%d%d%d%d",&m1,&m2,&m3,&m4,&m5,&m6);
              newnode->roll=r;
               newnode->total_marks=(m1+m2+m3+m4+m5+m6);
               newnode->percentage=((newnode->total_marks)/100)*100;
               newnode->next=NULL;
if(start==NULL)
{
        start=newnode;
}
else
{
        printf("\n enter position at which you want to insert particular student");
```

```
        scanf("%d",&loc);
        temp=start;
        while(temp!=NULL && count!=loc)
        {
                count++;
                ptr=temp;
                temp=temp->next;
        }
        if(temp==NULL)
        {
                printf("\nplease enter the correct position");
        }
        else
        {
                ptr->next=newnode;
                newnode->next=temp;
        }
}
```

```
}
}
void deleterecord()
{
        int ch;
        printf("\nTo delete a student record at\n1.begin\n2.end\n3.specified\n");
        {
                printf("enter your choice");
           scanf("%d",&ch);
                switch(ch)
                {
                        case 1:deleteatbegin();
                             break;
                        case 2:deleteatend();
                             break;
                        case 3:deleteatspecified();
                             break;
                        default:printf("\n invalid choice");
```

```
                }
        }
}
void deleteatbegin()
{
        struct node *temp;
        if(start==NULL)
        {
                printf("\n student details are not entered");
        }
        else
        {
                temp=start;
                printf("\nDeleted student\n");

        printf("\n\nRoll.NO:%d\nFirstName:%s\nFirstName:%s\nClass:%d\nTotal
Marks:%f\npercentagecentage:%f\n",  temp->roll,temp->firstname,temp->name,temp-
>class,temp->total_marks,((temp>percentage)/600)*100);
```

```
                start=start->next;
                free(temp);
                return;
        }
}
void deleteatend()
{
        struct node *temp;
        struct node *ptr;

        if(start==NULL)
        {
                printf("\nstudent details are not entered\n");
        }
   else
   {
        temp=start;
        if(start->next==NULL)
```

```
{
        start=NULL;
}
else
while(temp->next!=NULL)
{
        ptr=temp;
        temp=temp->next;
}
        ptr->next=NULL;
        printf("\ndeleted student details is\n");
        printf("\nDeleted student\n");

printf("\n\nRoll.NO:%d\nFirstName:%s\nFirstName:%s\nClass:%d\nTotal Marks:%f\npercentagecentage:%f\n",  temp->roll,temp->firstname,temp->name,temp->class, temp->total_marks,((temp->percentage)/600)*100);
        free(temp);
return;
```

```
}
}
void deleteatspecified()
{
        struct node *temp,*ptr;
        int loc,count=1;
        if(start==NULL)
        {
                printf("\nstudent details are not entered\n");
        }
        else
        {
                printf("\nenter the rollno of student\n");
                scanf("%d",&loc);
                temp=start;
                count=1;
                while(temp!=NULL && temp->roll!=loc)
                {
```

```
            ptr=temp;
            count=count+1;
            temp=temp->next;
        }
            if(temp==NULL)
            {
                printf("\nparticular student don't belongs to our school\n");
            }
            else
            {
                ptr->next=temp->next;
                printf("\nstudent\n");
        printf("\nDeleted student\n");

    printf("\n\nRoll.NO:%d\nFirstName:%s\nFirstName:%s\nClass:%d\nTotal
Marks:%f\npercentagecentage:%f\n",  temp->roll,temp->firstname,temp->name,temp-
>class, temp->total_marks,((temp->percentage)/600)*100);
                free(temp);
                return;
```

```
}
}
        }
```

OUTPUT:

1.create new student record

2.insert student details

3.search a student record

4.count total students

5.delete student record

6.To show all records

7.To update particular student details.

8.exit

enter your choice:1

enter the number of record you want to create:2

enter student details

roll:20

First Name:HARSHITHA

Last Name:PANDILLA

Class:8

Marks in Order:67

89

70

56

78

45

roll:21

First Name:VARSHINI

Last Name:ANNAM

Class:9

Marks in Order:56

78

90

45

34

23

enter your choice:2

To insert a student record at

1.Begin

2.End

3.Specified

enter your choice:1

enter student to be inserted details

roll:22

First Name:DIVIJA

Last Name:KANKALA

Class:9

Marks in Order:67

90

78

68

95

47

enter your choice:2

To insert a student record at

1.Begin

2.End

3.Specified

enter your choice:2

roll:25

First Name:DIVYA

Last Name:NISCHALA

Class:7

Marks in Order:56

78

90

23

45

67

enter your choice:2

To insert a student record at

1.Begin

2.End

3.Specified

enter your choice:3

roll:27

First Name:MALAVIKA

Last Name:MASADI

Class:7

Marks in Order:23

45

56

78

19

78

enter position at which you want to insert particular student2

enter your choice:3

enter the roll no of student21

Roll.NO:21

First Name:VARSHINI

First Name:ANNAM

Class:0

Total Marks:326.000000

percentagecentage:54.333336

enter your choice:4

Number of student :4

enter your choice:5

To delete a student record at

1.begin

2.end

3.specified

enter your choice1

Deleted student

Roll.NO:22

First Name:DIVIJA

First Name:KANKALA

Class:8

Total Marks:445.000000

percentagecentage:74.166664

enter your choice:5

To delete a student record at

1.begin

2.end

3.specified

enter your choice2

deleted student details is

Deleted student

Roll.NO:25

First Name:DIVYA

First Name:NISCHALA

Class:7

Total Marks:359.000000

percentagecentage:59.833336

enter your choice:5

To delete a student record at

1.begin

2.end

3.specified

enter your choice3

enter the rollno of student3

particular student don't belongs to our school

enter your choice:6

student details are

Student-1

Roll.NO:27

First Name:MALAVIKA

First Name:MASADI

Class:7

Total Marks:299.000000

percentage:49.833332

Student-2

Roll.NO:20

First Name:HARSHITHA

First Name:PANDILLA

Class:8

Total Marks:405.000000

percentage:67.500008

enter your choice:7

enter the roll of student to be updated:26

particular student don't belongs to our school

enter your choice:1

enter the number of record you want to create:1

enter student details

roll:29

First Name:AKSHITHA

Last Name:MUNJALA

Class:6

Marks in Order:45

67

89

90

23

45

enter your choice:7

enter the roll of student to be updated:29

To update press

1.first name

2.last name

3.roll-no

4.class

5.total marks

6.percentagecentage

enter the user choice:1

enter the new first name

AKSHAYA

enter your choice:3

enter the roll no of student29

Roll.NO:29

First Name:AKSHAYA

First Name:MUNJALA

Class:6

Total Marks:359.000000

percentagecentage:59.833336

enter your choice:8

Process exited after 437.7 seconds with return value 0

Press any key to continue . . .