

**KIET Group of Institutions, Ghaziabad**

**COMPUTER SCIENCE AND INFORMATION TECHNOLOGY**

**PROJECT BASED LEARNING**

**On**

**STUDENT MANAGEMENT SYSTEM**

**SUBJECT: DATA STRUCTURE & USING C LAB**

# Submitted By:

**AMAN KUMAR ( 2100290110019)**

**DEEPAK KUMAR (2100290110049)**

**ANAS AIJAS (2100290110026)**

**CODE NO.(161)**

**ACKNOWLEDGEMENT**

## I’ve got this golden opportunity to express my kind gratitude and sincere thanks to my subject faculty **“Mr. VINAY KUMAR”**, Computer Science and Information Technology Department, **KIET GROUP OF INSTITUTIONS** for their kind support and necessary counselling in the preparation of this project report. I’m also indebted to each and every person responsible for the making up of this project directly or indirectly.

I must also acknowledge or deep debt of gratitude each one of my colleague who led this project come out in the way it is. It’s my hard work and untiring sincere efforts and mutual cooperation to bring out the project work. Last but not the least, I would like to thank my parents for their sound counselling and cheerful support. They have always inspired us and kept our spirit up.

**AIM**

To allow the administrator of any organization to edit and find out the personal details of a student and allows the student to keep up his details.

**OBJECTIVE**

The main objective of the Student Management System is to manage the details of Profiles, Cources, Logins, Exams, Fees

**ABSTRACT**

Student Management System is software which is helpful for students as well as the school authorities. In the current system all the activities are done manually. It is very time consuming and costly. Our Student Management System deals with the various activities related to the students. In the Software we can register as a user and user has of two types, student and administrator. Administrator has the power to add new user and can edit and delete a user. A student can register as user and can add edit and delete his profile. The administrator can add edit and delete marks for the student. All the users can see the marks.

**BASIC PRINCIPLE**

It uses the principle that the most occuring Ip address stays at the top and so thetime complexity of searching decreases eventually.

**METHODOLOGY**

Student management web-based system is the process of managing student's record in an institutional organization. It is done through the online method which traditionally, was prepared using papers and manual ledgers. It preserves student's and administrator's resources.

**INTRODUCTION**

Student Management System is software which is helpful for students as well as the school

authorities. In the current system all the activities are done manually. It is very time consuming and costly. Our Student Management System deals with the various activities related to the students.

There are mainly 3 modules in this software

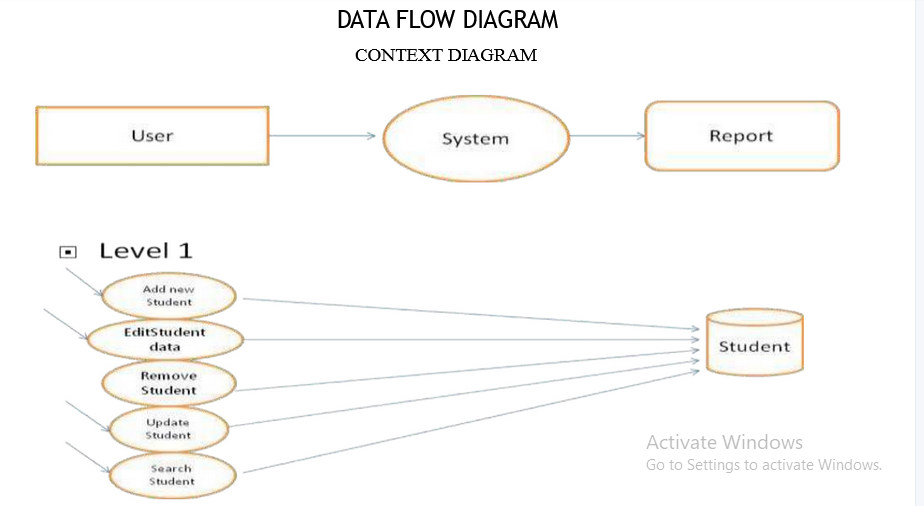
•User module

•Student Module

•Mark management

In the Software we can register as a user and user has of two types, student and administrator.

Administrator has the power to add new user and can edit and delete a user. A student can register as user and can add edit and delete his profile. The administrator can add edit and delete marks for thstudent. All the users can see the marks



# CODE IMPLEMENTATION

#include <iostream>

#include <conio.h>

using namespace std;

class Node

{

public:

int roll\_no;

string name ;

float marks,per;

Node \*next\_add;

};

class Linked\_List

{

public:

Node \*head = NULL;

void Insert()

{

int r;

string n;

float m;

cout<<"\n\n Enter Roll No. : ";

cin>>r;

cout<<"\n\n Enter Name : ";

cin>>n;

cout<<"\n\n Enter Marks : ";

cin>>m;

Node \*new\_node = new Node;

new\_node -> roll\_no = r;

new\_node -> name = n;

new\_node -> marks = m;

new\_node -> per = m/100\*100;

new\_node -> next\_add = NULL;

if(head==NULL)

{

head=new\_node;

}

else

{

Node \*ptr=head;

while(ptr->next\_add!=NULL)

{

ptr=ptr->next\_add;

}

ptr->next\_add=new\_node;

}

cout<<"\n\n\n New Node Inserted Successfully....";

}

void Search()

{

if(head==NULL)

{

cout<<"\n\n Linked List is Empty....";

}

else

{

int r,found=0;

cout<<"\n\n Enter Roll no. for Search : ";

cin>>r;

Node \*ptr =head;

while(ptr !=NULL)

{

if(r== ptr-> roll\_no)

{

cout<<"\n\n Roll No. :"<<ptr->roll\_no;

cout<<"\n\n Name : "<<ptr->name;

cout<<"\n\n Marks : "<<ptr->marks;

cout<<"\n\n Per % : "<<ptr->per;

found++;

}

ptr=ptr->next\_add;

}

if(found==0)

{

cout<<"\n\n Search Roll No."<<"Can't Found....";

}

}

}

void count()

{

if(head==NULL)

{

cout<<"\n\n Linked List is Empty....";

}

else

{

int c=0;

Node \*ptr =head;

while(ptr!=NULL)

{

c++;

ptr=ptr->next\_add;

}

cout<<"\n\n Total No. of Nodes :"<<c;

}

}

void Update()

{

if(head==NULL)

{

cout<<"\n\n Linked List is Empty....";

}

else

{

int r,found=0;

cout<<"\n\n Enter Roll no. for Updation : ";

cin>>r;

Node \*ptr =head;

while(ptr !=NULL)

{

if(r== ptr-> roll\_no)

{

cout<<"\n\n Enter New Roll No. : ";

cin>>ptr->roll\_no;

cout<<"\n\n Enter Name : ";

cin>>ptr->name;

cout<<"\n\n Enter Marks : ";

cin>>ptr->marks;

ptr->per= ptr->marks/100\*100;

cout<<"\n\n\n Record Updated Successfully";

found++;

}

ptr=ptr->next\_add;

}

if(found==0)

{

cout<<"\n\n Update Roll No."<<"Can't Found....";

}

}

}

void Del()

{

if(head==NULL)

{

cout<<"\n\n Linked List is Empty....";

}

else

{

int r,found=0;

cout<<"\n\n Enter Roll no. for Deletion : ";

cin>>r;

if(r==head->roll\_no)

{

Node \*ptr =head;

head=head->next\_add;

cout<<"\n\n Record Deleted Successfully...";

found++;

delete ptr;

}

else

{

Node \*pre =head;

Node \*ptr= head->next\_add;

while(ptr!=NULL)

{

if(r=ptr->roll\_no)

{

pre->next\_add=ptr->next\_add;

cout<<"\n\n Record Deleted Successfully...";

found++;

delete ptr;

break;

}

pre=ptr;

ptr=ptr->next\_add;

}

if(found==0)

{

cout<<"\n\n Delete Roll No."<<"Can't Found....";

}

}

}

}

void Show()

{

if(head==NULL)

{

cout<<"\n\n Linked List is Empty....";

}

else

{

Node \*ptr =head;

while(ptr !=NULL)

{

cout<<"\n\n Roll No. :"<<ptr->roll\_no;

cout<<"\n\n Name : "<<ptr->name;

cout<<"\n\n Marks : "<<ptr->marks;

cout<<"\n\n Per % : "<<ptr->per;

cout<<"\n\n \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_";

ptr=ptr->next\_add;

}

}

}

};

int main()

{

Linked\_List obj;

p:

system("cls");

int choice;

cout<<"\n\n 1. Insert Record";

cout<<"\n\n 2. Search Record";

cout<<"\n\n 3. Count Nodes";

cout<<"\n\n 4. Update Record";

cout<<"\n\n 5. Delete Record";

cout<<"\n\n 6. Show All Record";

cout<<"\n\n 7. Exit";

cout<<"\n\n\n Your Choice : ";

cin>>choice;

switch(choice)

{

case 1:

system("cls");

obj.Insert();

break;

case 2:

system("cls");

obj.Search();

break;

case 3:

system("cls");

obj.count();

break;

case 4:

system("cls");

obj.Update();

break;

case 5:

system("cls");

obj.Del();

break;

case 6:

system("cls");

obj.Show();

break;

case 7:

exit(0);

default:

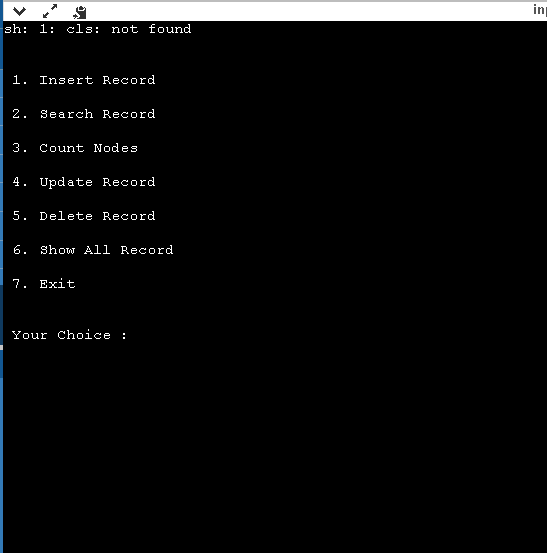
cout<<"\n\n\n Invalid Choice...Please Try Again";

}

getch();

goto p;

RESULT/OUTPUT



TIME COMPLEXITY**:** O(log n)

# SPACE COMPLEXITY: O(1)

**REFERENCE**

1. [https://www.youtube.com/watch?v=qMmqOHr75b8&ab\_channel=Jenn](https://www.youtube.com/watch?v=qMmqOHr75b8&ab_channel=Jenny%27slecturesCS%2FITNET%26JRF) [y%27slecturesCS%2FITNET%26JRF](https://www.youtube.com/watch?v=qMmqOHr75b8&ab_channel=Jenny%27slecturesCS%2FITNET%26JRF)
2. [https://www.youtube.com/watch?v=1HeIZNP3w4A&ab\_channel=Jenny](https://www.youtube.com/watch?v=1HeIZNP3w4A&ab_channel=Jenny%27slecturesCS%2FITNET%26JRF)

## [%27slecturesCS%2FITNET%26JRF](https://www.youtube.com/watch?v=1HeIZNP3w4A&ab_channel=Jenny%27slecturesCS%2FITNET%26JRF)