

# **STUDENT RECORD SYSTEM**

**PREPARED BY:**  
**SARU PRADHAN**  
**SUMITA DANGAL**

**SUBMITTED TO:**  
**DEPARTMENT OF**  
**ELECTRONICS AND**  
**COMPUTER ENGINEERING**

# CONTENT

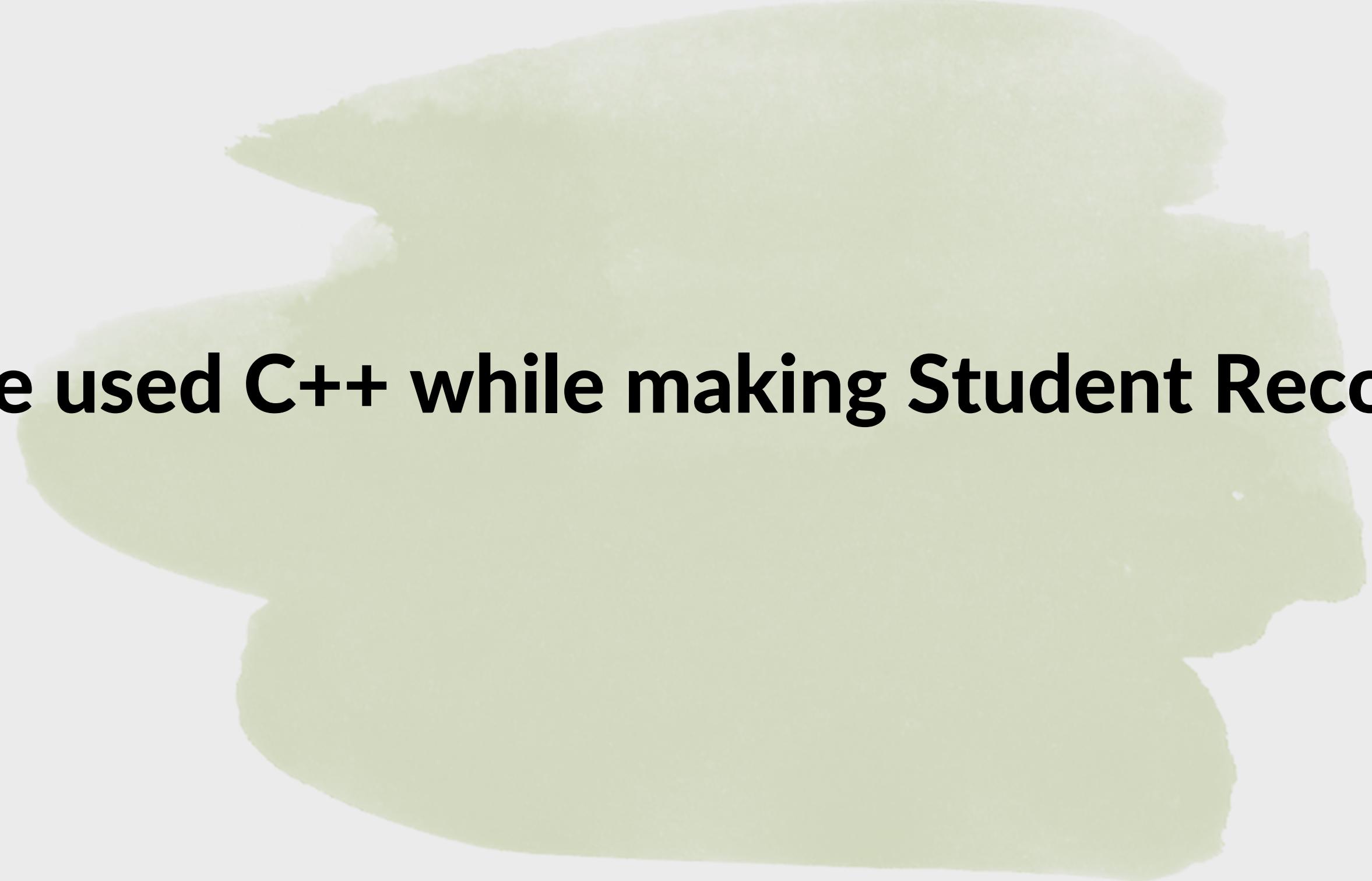
- Overview Of Project
- Programming Language Used
- Linked List
- Code Part
- Conclusion



# OVERVIEW OF PROJECT

## Student Record System

- A Student Record System works to coordinate scheduling and communications between faculty regarding students.
- Student Management System allows users to add, delete, and display student records and search for specific students by unique roll no.
- It is a solution tool designed to track, maintain and manage all the data generated by a school, including the name, roll no, contact number, scored marks and percentage, and so on.
- It allows school administrators to maintain accurate records and data on students, which can be used for various purposes such as tracking student progress, managing students, and monitoring academic performance.

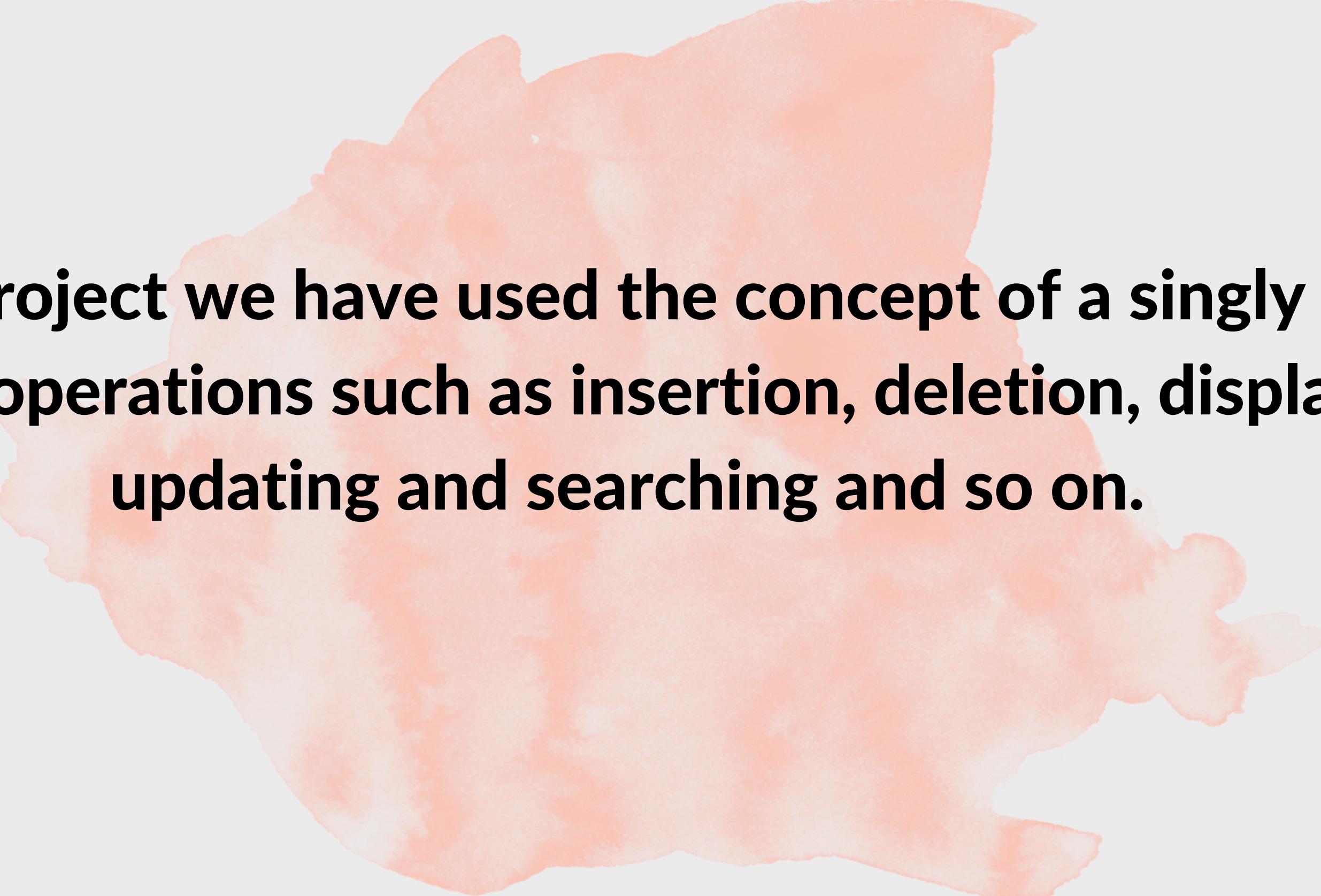


**We have used C++ while making Student Record System**

# PROGRAMMING LANGUAGE USED

## C++

- C++ is an object-oriented computer language.
- Developed by Bjarne Stroustrup.
- We used C++ while developing Student Record System.
- C++ is a cross-platform language, meaning that code written in C++ can be compiled on a variety of different platforms and operating systems, including Windows, Linux, and macOS. This makes it a good choice for developing software that needs to run on multiple platforms.
- It is a high-performance language that is compiled directly into machine code, making it very efficient in terms of speed and memory usage



**In this project we have used the concept of a singly linked list for operations such as insertion, deletion, displaying, updating and searching and so on.**

# LINKED LIST

- A linked list is a popular data structure used in computer science and data structures and algorithms.
- It is a dynamic data structure that can grow or shrink in size during program execution.
- It is a linear data structure.
- A linked list is made up of a series of nodes, each of which contains a data value and a reference (pointer) to the next node in the list.
- The first node is called the head of the list.
- While the last node points to a null value.
- There are two main types of linked lists: Singly linked lists and Doubly linked lists.

# Singly Linked List

- We use a singly linked list i.e Linear Data Structure in this project of STUDENT RECORD SYSTEM.
- In a singly linked list, each node contains two fields.
- One is the info field -> data items are stored here.
- Another is the link field -> to point next node in the list.
- We use various operations of singly linked lists in this project such as Insertion and Deletion.

```
21 void Insert(){
22     HANDLE h = GetStdHandle(STD_OUTPUT_HANDLE);
23     int r;
24     string n;
25     char g;
26     long long cn;
27     float m;
28
29     SetConsoleTextAttribute(h,6);
30     cout<<"\n\n Enter Roll no. : ";
31     cin>>r;
32     cout<<"\n\n Enter Name : ";
33     cin>>n;
34     cout<<"\n\n Enter 'M'for male and 'F'for female : ";
35     cin>>g;
36     cout<<"\n\n Enter Contact No. : ";
37     cin>>cn;
38     cout<<"\n Full Marks = 600";
39     cout<<"\n\n Enter Scored Marks : ";
40     cin>>m;
41     Node *new_node = new Node;
42     new_node -> rollno = r;
43     new_node -> name = n;
44     new_node -> marks = m;
45     new_node -> contactno = cn;
46     new_node -> gender = g;
47     new_node -> per = m/600*100;
48     new_node -> next_add = NULL;
49     if(head == NULL){
50         head = new_node;
51     }
52     else{
53         Node *ptr = head;
54         while(ptr-> next_add != NULL){
55             ptr = ptr->next_add;
56         }
57         ptr -> next_add = new_node;
58     }
59 }
60 cout<<"\n\n New Record Inserted Successfully....";
61 }
```

```
63 void Search()
64 {
65     HANDLE h = GetStdHandle(STD_OUTPUT_HANDLE);
66     if(head == NULL){
67         SetConsoleTextAttribute(h,4);
68         cout<<"\n\n !!! List is Empty !!!";
69     }
70     else{
71         int r, found = 0;
72         SetConsoleTextAttribute(h,4);
73         cout<<"\n\n Enter roll no for search: ";
74         cin>>r;
75         Node *ptr = head;
76         while(ptr !=NULL)
77         {
78             if(r == ptr -> rollno){
79                 SetConsoleTextAttribute(h,6);
80                 cout <<"\n\n Roll No :"<<ptr -> rollno;
81                 cout <<"\n\n Name : "<<ptr -> name;
82                 cout <<"\n\n Gender : "<<ptr -> gender;
83                 cout <<"\n\n Contact No : "<<ptr -> contactno;
84                 cout <<"\n\n Marks : "<< ptr -> marks;
85                 if(ptr -> marks > 600){
86                     cout << "\n\n Percentage % : Out of range";
87                 }
88                 else{
89                     cout <<"\n\n Percentage % : "<< ptr->per<<"%";
90                 }
91             }
92             found++;
93         }
94         ptr = ptr -> next_add;
95     }
96     if(found == 0)
97     {
98         cout <<"\n\n Search Roll No."<<r<<" Can't Found...";
99     }
100 }
101 }
```

```
103     void Count()
104     {
105         HANDLE h = GetStdHandle(STD_OUTPUT_HANDLE);
106         if(head == NULL)
107         {
108             SetConsoleTextAttribute(h,4);
109             cout<<"\n\n !!! List is Empty !!! ";
110         }
111         else
112         {
113             int c=0;
114             Node *ptr = head;
115             while(ptr != NULL)
116             {
117                 c++;
118                 ptr = ptr -> next_add;
119             }
120             SetConsoleTextAttribute(h,3);
121             cout << "\n\n Total No. of Students: "<<c;
122         }
123     }
```

```
124     void Update()
125     {
126         HANDLE h = GetStdHandle(STD_OUTPUT_HANDLE);
127         if(head == NULL){
128             SetConsoleTextAttribute(h,4);
129             cout<<"\n\n !!! List is Empty !!! ";
130         }
131         else
132         {
133             int r, found = 0;
134             cout<<"\n\n Enter roll no for updating: ";
135             cin>>r;
136             Node *ptr = head;
137             while(ptr !=NULL)
138             {
139                 if(r == ptr -> rollno)
140                 {
141                     cout<<"\n\n Enter New Roll no. : ";
142                     cin>>ptr -> rollno;
143                     cout<<"\n\n Enter Name: ";
144                     cin>>ptr -> name;
145                     cout<<"\n\n Enter Gender: ";
146                     cin>>ptr -> gender;
147                     cout<<"\n\n Enter Contact No.: ";
148                     cin>>ptr -> contactno;
149                     cout<<"\n Full Marks = 600";
150                     cout<<"\n\n Enter marks: ";
151                     cin>>ptr -> marks;
152                     ptr -> per = ptr -> marks/600 *100;
153                     cout <<"\n\n Record Updated Successfully...";
154                     found++;
155                 }
156             }
157             ptr = ptr -> next_add;
158         }
159         if(found == 0)
160         {
161             cout <<"\n\n Updated Roll No."<<r<<" Can't Found...";
162         }
163     }
164 }
```

```
165
166
167     void Delete()
168     {
169         HANDLE h = GetStdHandle(STD_OUTPUT_HANDLE);
170         if(head == NULL){
171             SetConsoleTextAttribute(h,4);
172             cout<<"\n\n !!! List is Empty !!! ";
173         }
174         else
175         {
176             int r, found = 0;
177             cout<<"\n\n Enter roll no for deletion: ";
178             cin>>r;
179             if(r == head -> rollno)
180             {
181                 Node *ptr = head;
182                 head = head -> next_add;
183                 cout<<"\n\n Record Deleted Successfully...";
184                 found++;
185                 delete ptr;
186             }
187             else
188             {
189                 Node *pre = head;
190                 Node *ptr = head -> next_add;
191                 while(ptr != NULL)
192                 {
193                     if(r == ptr -> rollno)
194                     {
195                         pre -> next_add = ptr -> next_add;
196                         cout<<"\n\n Record Deleted Successfully...";
197                         found++;
198                         delete ptr;
199                         break;
200                     }
201                     pre = ptr;
202                     ptr = ptr -> next_add;
203                 }
204                 if(found == 0)
205                 {
206                     cout <<"\n\n Delete Roll No."<<r<<"Can't Found...";
207                 }
208             }
209         }
```

```
210     void Display()
211     {
212         HANDLE h = GetStdHandle(STD_OUTPUT_HANDLE);
213         if(head == NULL){
214             SetConsoleTextAttribute(h,4);
215             cout<<"\n\n !!! List is Empty !!! ";
216         }
217         else
218         {
219             Node *ptr = head;
220             while(ptr !=NULL)
221             {
222                 SetConsoleTextAttribute(h,6);
223                 cout <<"\n\n Roll No. :"<<ptr -> rollno;
224                 cout <<"\n\n Name : "<<ptr -> name;
225                 cout <<"\n\n Gender : "<<ptr -> gender;
226                 cout <<"\n\n Contact No. : "<<ptr -> contactno;
227                 cout <<"\n\n Marks : "<< ptr -> marks;
228                 if(ptr -> marks > 600){
229                     cout << "\n\n Percentage % : Out of range";
230                 }else
231                 {
232                     cout <<"\n\n Percentage % : "<< ptr->per<<"%";
233                 }
234                 cout <<"\n\n*****";
235
236                 ptr = ptr -> next_add;
237             }
238         }
239     };
240 }
```

# OUTPUTS

## STUDENTS RECORD SYSTEM :

Enter:

1. Insert Record
2. Search Record
3. Count No of Students
4. Update Record
5. Delete Record
6. Display all record
7. exit

Enter Your Choice: 1

Enter Roll no. : 01

Enter Name : Aayusha

Enter 'M'for male and 'F'for female : F

Enter Contact No. : 9871456244

Full Marks = 600

Enter Scored Marks : 400

New Record Inserted Successfully....

Enter roll no for search: 2

Roll No :2

Name : Binit

Gender : M

Contact No : 985677412

Marks : 355

Percentage % : 59.1667%

Total No. of Students: 3

Enter roll no for deletion: 2

Record Deleted Successfully...

Roll No. :2

Name : Faisal

**Gender : M**

Contact No. : 9847562144

Marks : 444

Percentage % : 74%

Roll No. :2

Name : Binit

## Gender : M

Contact No. : 985677412

Marks : 355

Percentage % : 59.1667%

卷之三

Roll No. :3

Name : Cynthia

### Gender : F

Contact No. : 987564112

Marks : 555

Percentage % : 92.5%

## **CONCLUSION**

The Student Record System was developed using C++ programming language and the concept of Linked List specifically Singly Linked List i.e. Linear Data Structure. This recording system assists in inserting the details, updating the details, deleting past details when necessary, and displaying the details of students through their unique roll number.



COUT << "THANK YOU FOR  
YOUR UNDIVIDED  
ATTENTION";