



# **Green University of Bangladesh**

## **Department of Computer Science and Engineering (CSE)**

**Faculty of Sciences and Engineering**  
**Semester: (Summer, Year: 2021), B.Sc. in CSE (Day)**

### **Project Report**

**Course Title: Data Structure Lab**  
**Course Code: CSE 106                      Section: 213DC**

**Lab Project Name: STUDENTS RECORDS MANAGEMENT SYSTEM.**

#### **Student Details**

	<b>Name</b>	<b>ID</b>
<b>1.</b>	Md Masud Rana	213902101

**Submission Date: 11/09/2022**

**Course Teacher's Name: Md. Sultanul Islam Ovi, Lecturer Dpt. Of CSE of GUB.**

**[For Teachers use only: **Don't Write Anything inside this box**]**

#### **Lab Project Status**

**Marks: .....**

**Signature: .....**

**Comments: .....**

**Date: .....**

# Table of Contents

<b>Chapter 1 Introduction.....</b>	<b>3</b>
1.1 Introduction.....	3
<b>Data Storage.....</b>	<b>3</b>
<b>Data Exchange.....</b>	<b>3</b>
<b>Resource and Service Management .....</b>	<b>3</b>
<b>Scalability.....</b>	<b>3</b>
1.1 Design Goals/Objective .....	4
<b>Chapter 2 .....</b>	<b>4</b>
<b>Implementation of the Project .....</b>	<b>4</b>
<b>Chapter 3 Conclusion .....</b>	<b>24</b>

# Chapter 1

## Introduction

### 1.1 Introduction

#### **What is data structure ?**

- ❖ A data structure is a particular way of organizing data in a computer so that it can be used effectively.

#### **Types of data structure:**

Basically, data structures are divided into two categories: **Linear** data structure. **Non-linear** data structure.

**Linear data structure** : array , linked list, stack and queue etc

**Non-linear data structure:** graph and trees etc

#### **Application of data structure:**

Data Storage

Data Exchange.

Resource and Service Management

Scalability.

#### **Data structure operation**

1. Searching
2. Sorting
3. Insertion
4. Deletion
5. Traverse
6. Update

## 1.1 Design Goals/Objective

The goal of the project is to design A STUDENT RECORDS MANAGEMENT SYSTEM.

Using this project anyone can perform some operation on student record management system. Admin can easily add a record with constant time and can manage large amount of records. When admin will need to find, he can easily do it. Just search by id and find out the record. Even he will show all the records with how many records are available in the database. Even he can delete any data any time just by id or any position. When admin need sort the records. Easily can be performed. When need to see how many records remain in file. Easily he can find out. Overall I said that as a mini project idea. It is best for above such kind of operation.

# Chapter 2

## Implementation of the Project

### 2.1 Implementation

#### C++ source code

```
#include<bits/stdc++.h>
using namespace std;
void create( );

void add_Records();
void insertAtBegin();
void insertAtEnd();
void insertAnyPosition();

void traverse_records();
void count_Records();
void search_linear();

void delete_Record();
void deleteAtBegin();
void deleteAtEnd();
void deleteAnyPosition();
void delete_By_ID();

void sort_selection();
```

```

void swap_INT(long int *,long int *);
void swap_String(char *, char *);
void swap_Float(float *x, float *y);

void decorationhomeWindow();
void decoration_on_all_students();
void decoration_on_searching_record();

//i have taken class
class node
{
    public: // class member we can used whole the program for doing
public
    long int ID;
    char name[30];
    char dpt[10];
    char section[20];
    float CGPA;
    char phone[12];
    node *next;
};
//globally i have declared
node *head,*temp,*tail;

int main()
{
    system("color 0a"); //this is for consol color.
    int choice=1;
    while(choice!=0)
    {
// i am using decoration for welcome window
        decorationhomeWindow();
        cout<<(" 1. Create Records\n");
        cout<<(" 2. Add Records\n");
        cout<<(" 3. Display All Records\n");
        cout<<(" 4. Number of Records\n");
        cout<<(" 5. Search Record\n");
        cout<<(" 6. Delete Record\n");
        cout<<(" 7. Sort Records\n");
        cout<<(" 0. Exit\n");
        cout<<("\nChoose Your Option\n");
    }
}

```

```

        cin>>choice;
        switch(choice)
        {
        case 1:
            create( );
            break;
        case 2:
            add_Records();
            break;
        case 3:
            traverse_records();
            break;
        case 4:
            count_Records();
            break;
        case 5:
            search_linear();
            break;
        case 6:
            delete_Record();
            break;
        case 7:
            sort_selection();
            break;

        case 0:
            exit(0);
            break;
        default:
            cout<<("\nWrong option: ");
        }
    }
}

// create records
void create( )
{
    system("cls");
    node *newnode;
    head=0;
    int i,choice;

```

```

cout<<("\nHow Many records want to Create? ");
cin>>choice;

for(i=1; i<=choice; i++)
{
    newnode = new node();

    cout<<("Enter ID: ");
    cin>>newnode->ID;
    fflush(stdin);

    cout<<("Enter Name : ");
    cin.get(newnode->name, 100);

    cout<<("Enter Depertment : ");
    cin>>newnode->dpt;

    cout<<("Enter Section : ");
    cin>>newnode->section;

    cout<<("Enter CGPA : ");
    cin>>newnode->CGPA;

    cout<<("Enter Contact Number : ");
    cin>>newnode->phone;

    newnode->next = 0;
    if(head==0)
    {
        head = tail = temp = newnode;
    }
    else
    {
        temp->next = newnode;
        temp = newnode;
        tail=tail->next;
    }
    system("cls");
}
cout<<("\nRecord Created Successfully\n")<<endl;
}

```

```

// add records
void add_Records()
{
    int pos;
    cout<<("\nWhich Position Want to Add?\n1. Add Begin\n2. Add
End\n3. Add Any Position\n")<<endl;
    cin>>pos;
    if(pos==1)
        insertAtBegin();
    else if(pos==2)
        insertAtEnd();
    else if(pos==3)
        insertAnyPosition();
    else
        cout<<("\nWrong Option\n")<<endl;
}

//insert record at beggining
void insertAtBegin()
{
    system("cls");
    node *newnode;
    newnode = new node();
    cout<<("Enter ID: ");
    cin>>newnode->ID;
    fflush(stdin);

    cout<<("Enter Name : ");
    cin.get(newnode->name, 100);

    cout<<("Enter Depertment : ");
    cin>>newnode->dpt;

    cout<<("Enter Section : ");
    cin>>newnode->section;

    cout<<("Enter CGPA : ");
    cin>>newnode->CGPA;
}

```



```

        cout<<("Enter Contact Number : ");
        cin>>newnode->phone;

newnode->next=head;
head=newnode;
system("cls");
cout<<("\nRecord Added Successfully\n");
}

//insert record end of the list
void insertAtEnd()
{
    system("cls");
    node *newnode;
    newnode = new node();

    cout<<("Enter ID: ");
        cin>>newnode->ID;
        fflush(stdin);

    cout<<("Enter Name : ");
        cin.get(newnode->name, 100);

    cout<<("Enter Depertment : ");
        cin>>newnode->dpt;

    cout<<("Enter Section : ");
        cin>>newnode->section;

    cout<<("Enter CGPA : ");
        cin>>newnode->CGPA;

    cout<<("Enter Contact Number : ");
        cin>>newnode->phone;

    newnode->next=0;
    tail->next = newnode;
    tail = newnode;
    system("cls");
    cout<<("\nRecord Added Successfully\n");
}

```

```

}

//insert record any position in the list
void insertAnyPosition()
{
    system("cls");
    int pos;
    cout<<("Enter position to Add: ");
    cin>>pos;
    if(pos<0)
    {
        cout<<("Invalid position\n");
    }
    else
    {
        node *newnode;
        newnode = new node();

        cout<<("Enter ID: ");
        cin>>newnode->ID;
        fflush(stdin);

        cout<<("Enter Name : ");
        cin.get(newnode->name, 100);

        cout<<("Enter Depertment : ");
        cin>>newnode->dpt;

        cout<<("Enter Section : ");
        cin>>newnode->section;

        cout<<("Enter CGPA : ");
        cin>>newnode->CGPA;

        cout<<("Enter Contact Number : ");
        cin>>newnode->phone;

        newnode->next=0;
        temp = head;
        int i=1;
        while(i<pos-1)

```

```

        {
            temp = temp->next;
            i++;
        }
        newnode->next=temp->next;
        temp->next = newnode;
    }
    system("cls");
    cout<<("\nRecord Added Succssfully\n");
}

//display all records
void traverse_records()
{
    system("cls");

    if(head==0)
    {
        cout<<("There are no Records!\n");
    }
    else
    {
        // call the decoration function
        decoration_on_all_students();
        temp = head;
        while(temp!=NULL)
        {
            cout << " "<<temp->ID << " \t"
                << temp->name << "\t"
                << temp->dpt << "\t\t"
                << temp->section << "\t  "
                << temp->CGPA << "\t  "
                << temp->phone << endl;
            temp = temp->next;
        }

        cout<<("\n");
    }
}

```

```

//count all records
void count_Records()
{
    system("cls");
    temp = head;
    int count_all_records=0;
    while(temp!=NULL)
    {
        count_all_records++;
        temp = temp->next;
    }
    cout<<"\nTotal Number of Records:"<<count_all_records<<endl;
}

// searching specific record by id
void search_linear()
{
    system("cls");
    int id,flag=0;
    cout<<("\nEnter ID to Search: ");
    cin>>id;
    temp = head;
    while(temp!=NULL)
    {
        if(id==temp->ID)
        {
            decoration_on_searching_record();
            flag=1;
            cout << " " <<temp->ID << " \t"
                << temp->name << "\t"
                << temp->dpt << "\t\t"
                << temp->section << "\t   "
                << temp->CGPA << "\t   "
                << temp->phone << endl;
            break;
        }
        temp=temp->next;
    }
    if(flag==0)
        cout<<("\nRecord Not Found!\n");
    // printf("\n\n\n");
}

```

```

}

// delete function
void delete_Record()
{
    int choice;
    cout<<("\n1. Delete Beginning of Records\n2. Delete At End of
Records\n3. Delete Any Position Of Records");
    cout<<("\n4. Delete By ID\n\nChoose Your Option\n");
    cin>>choice;
    if(choice==1)
        deleteAtBegin();
    else if(choice==2)
        deleteAtEnd();
    else if(choice==3)
        deleteAnyPosition();
    else if(choice==4)
        delete_By_ID();
    else
        cout<<("\nWrong Option!\n")<<endl;
}

// delete record at beggining of the records.
void deleteAtBegin()
{
    system("cls");
    if(head==0)
        cout<<("\nThere is No Record!\n")<<endl;
    else
    {
        temp = head;
        head = head->next;
        free(temp); // free the deleted node
        cout<<("Record Deleted Successfully\n");
    }
}

// delete record end of the records.
void deleteAtEnd()
{
    system("cls");

```

```

    if(head==0)
    {
        cout<<("\nThere is No Record!\n");
    }
    else
    {
        node *preNode;
        temp = head;
        while(temp->next!=0)
        {
            preNode = temp;
            temp = temp->next;
        }
        preNode->next=0;
        free(temp); //here free the deleted node
        cout<<("\nRecord Deleted Successfully\n");
    }
}

// delete record any position of the records
void deleteAnyPosition()
{
    system("cls");
    if(head==0)
    {
        cout<<("\nThere is No Record!\n");
    }
    else
    {
        int pos;
        cout<<("Enter position To Delete");
        cin>>pos;
        int i=1;
        temp = head;
        while(i<pos-1)
        {
            temp = temp->next;
            i++;
        }
        temp->next = temp->next->next;
        free(temp); //here done free deleted node
    }
}

```

```

        cout<<("\nRecord Deleted Successfully\n");
    }
}

// delete record by student id
void delete_By_ID()
{
    system("cls");
    if(head==0)
    {
        cout<<("\nThere is No Record!\n");
    }
    else
    {
        int id,flag=0;
        cout<<("\nInput ID To Delete :\n ");
        cin>>id;
        if(id==head->ID)
        {
            deleteAtBegin();
        }
        else
        {
            node *preNode;
            temp = head;
            while(temp!=NULL)
            {
                preNode = temp;
                temp=temp->next;
                if(id==temp->ID)
                {
                    flag=1;
                    preNode->next = temp->next;
                    cout<<("\nRecord Deleted Successfully\n");
                    break;
                }
            }
        }
        if(flag==0)
            cout<<("\nRecord Not Found!\n");
    }
}

```

```

}

// sorting the records according to id
void sort_selection()
{
    system("cls");
    node *curNode,*nextNode;
    curNode = head;
    while(curNode!=0)
    {
        nextNode = curNode->next;

        while(nextNode!=0)
        {
            if(curNode->ID>nextNode->ID)
            {
                swap_INT(&curNode->ID, &nextNode->ID);
                swap_String(curNode->name, nextNode->name);
                swap_String(curNode->dpt, nextNode->dpt);
                swap_String(curNode->section, nextNode->section);
                swap_Float(&curNode->CGPA, &nextNode->CGPA);
                swap_String(curNode->phone, nextNode->phone);
            }
            nextNode = nextNode->next;
        }
        curNode = curNode->next;
    }
    cout<<("\nRecord is Now Sorted\n");
}

// this function swap only long integer data type.
void swap_INT(long int *a,long int *b)
{
    int temp;
    temp = *a;
    *a = *b;
    *b = temp;
}

//this function swap only float data type.
void swap_Float(float *x, float *y)

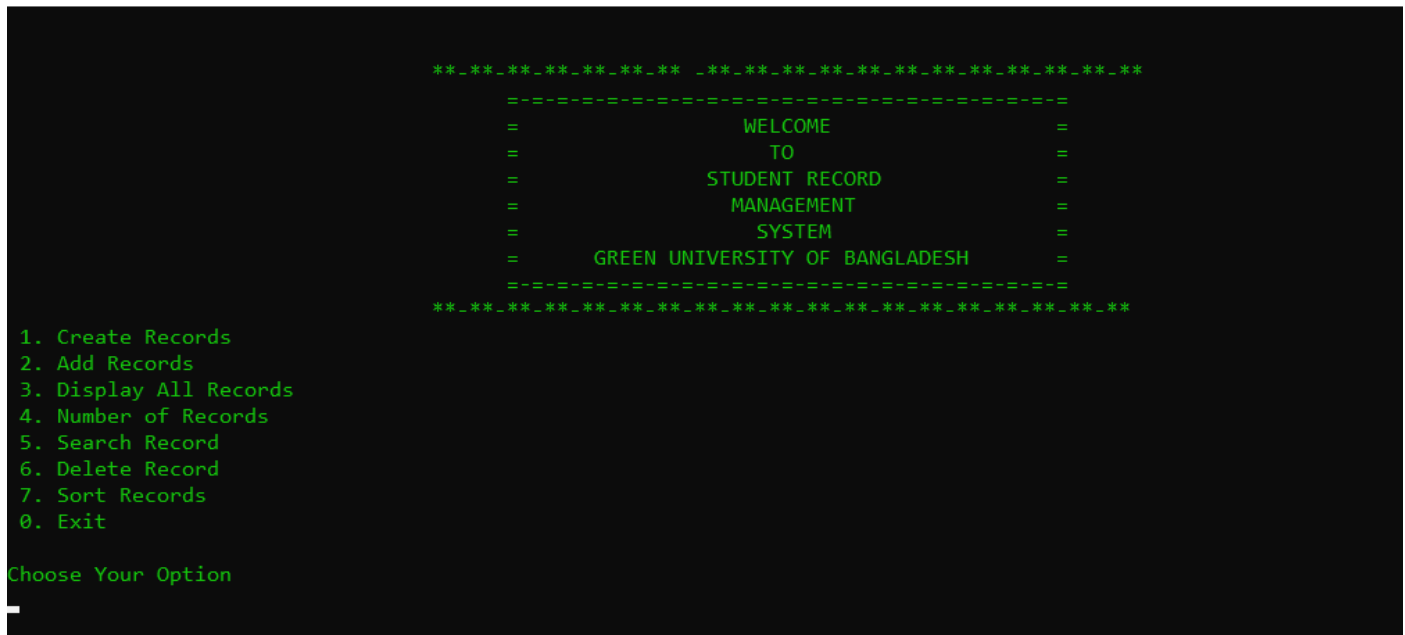
```





[illegible]

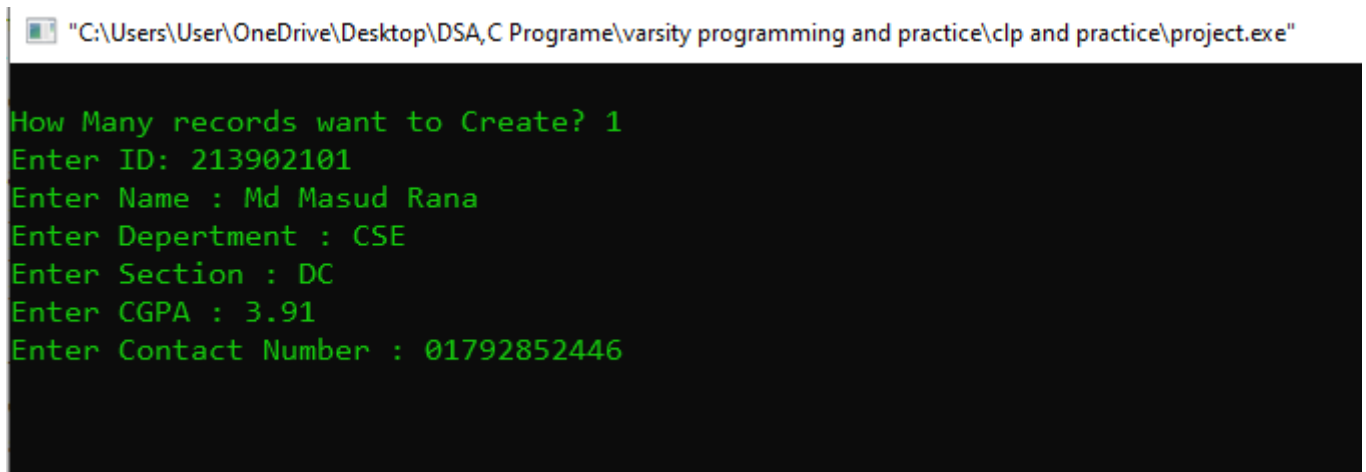
## Screenshots



*Figure 1: Main Menu interface*

When the program is executed, the user will be directed to the main menu interface. The program is introduced with a few lines of texts. Then seven selections are made for the user as the user can create record, insert record, display all records, search , delete record and sort the record.

User can insert record at begin, end and any position and user can delete any record by id or any position.



### Figure 2: Creates Record

If the user choice 1 then above interface will show . if user press 1 for number of record then 1 record will create and it will take student id, name, department, section , cgpa and contact number .

```
1. Create Records
2. Add Records
3. Display All Records
4. Number of Records
5. Search Record
6. Delete Record
7. Sort Records
0. Exit

Choose Your Option
2

Which Position Want to Add?
1. Add Begin
2. Add End
3. Add Any Position
```

*Figure 3: add records driven menu*

When the user press 2 for adding new record to the exiting record then above driven menu will show. There will show 3 option , adding begin, add end and any position. User what want he will able to do any operation just press the number.

```
***** LIST OF ALL STUDENTS RECORDS *****
```

ID	Name	Department	Section	CGPA	Phone No.
21399344	Shamoli Rahman	CSE	DC	4	01398343434
213902101	Md Masud Rana	CSE	DC	3.91	01792852446

```
*****  
===== WELCOME =====  
= TO =  
= STUDENT RECORD =  
= MANAGEMENT =  
= SYSTEM =  
= GREEN UNIVERSITY OF BANGLADESH =  
=====
```

```
*****
```

1. Create Records
2. Add Records
3. Display All Records
4. Number of Records
5. Search Record
6. Delete Record
7. Sort Records
0. Exit

```
Choose Your Option
```

**Figure 4: List of All Records**

As shown in the interface showed List of All records. When user will press 3 for displaying all records then this interface will be showed

```
C:\Users\User\OneDrive\Desktop\DSA_C Programs\varsity programming and practice\cp and practice\project.exe
```

```
Total Number of Records:2  
  
**_**_*_*_**_*_*_**_*_* _**_*_*_**_*_*_**_*_*_**_*_*_**_*_*  
======  
=                                WELCOME                               =  
=                                TO                                   =  
=                                STUDENT RECORD                       =  
=                                MANAGEMENT                           =  
=                                SYSTEM                              =  
=                                GREEN UNIVERSITY OF BANGLADESH       =  
======  
**_**_*_*_**_*_*_**_*_*_**_*_*_**_*_*_**_*_*_**_*_*_**_*_*  
  
1. Create Records  
2. Add Records  
3. Display All Records  
4. Number of Records  
5. Search Record  
6. Delete Record  
7. Sort Records  
0. Exit  
  
Choose Your Option
```

*Figure 5: when user press 4 then will show number of Records*

```
"C:\Users\User\OneDrive\Desktop\DSA,C Programme\varsity programming and practice\clp and practice\project.exe"
Enter ID to Search: 213902101

***** SEARCHING STUDENTS RECORD *****

ID          Name          Department  Section  CGPA  Phone No.
-----
213902101    Md Masud Rana  CSE        DC       3.91  01792852446
```

*Figure 6: Searching Result*

If The user want to find out the specific user, then just need to search by id. If id is valid then student record will show like above figure -6. If id is invalid or incorrect then it will show that “Record is not Found” is given bellow

```
"C:\Users\User\OneDrive\Desktop\DSA,C Programme\varsity programming and practice\clp and practice\project.exe"
Enter ID to Search: 213938434
Record Not Found!
```

*Figure 7: Searching id incorrect*

In figure 4 we see that two records having in the records. But searching id is not available in the records that is why when user will search by id the program will show that “Reocrd is not found “

```
1. Create Records
2. Add Records
3. Display All Records
4. Number of Records
5. Search Record
6. Delete Record
7. Sort Records
0. Exit

Choose Your Option
6

1. Delete Beginning of Records
2. Delete At End of Records
3. Delete Any Position Of Records
4. Delete By ID

Choose Your Option
```

*Figure 8: Delete Rocord menu driven*

If user press 6 for deleting any record then 4 option will show. User can perform one of them. He will can delete any user by his id or user can delete any position record.

```

"C:\Users\User\OneDrive\Desktop\DSA,C Programme\varsity programming and practice\clp and practice\project.exe"

Input ID To Delete :
213902101

Record Deleted Successfully
  
```

*Figure 9: delete record by id*

The above figure will show when user press 4 for deleting record by id. If id is valid then the record will delete and will show “ Record deleted Successfully” otherwise the program will show that “ record is not found”

```

***** LIST OF ALL STUDENTS RECORDS *****
  
```

ID	Name	Department	Section	CGPA	Phone No.
98234324	Soton Nodini	EEE	DB	4	0133343434
21399344	Shamoli Rahman	CSE	DC	4	01398343434

*Figure 10: Unsorted Records List*

Now all records are in unsorted but when user will press 7 for sorting . below interface with sorted records will show.

```

***** LIST OF ALL STUDENTS RECORDS *****
  
```

ID	Name	Department	Section	CGPA	Phone No.
21399344	Shamoli Rahman	CSE	DC	4	01398343434
98234324	Soton Nodini	EEE	DB	4	0133343434

*Figure 11: Sorted List*

# Chapter 3

## Conclusion

### Learning Outcome

The Student Records management system is designed for educational institution to create records and insert record, display all records, count , delete , search , delete and sorting the record. All the operation will be easily performed for large amount of data. This project is used data structure and algorithm and c++ language.

There are many function or operation had been implemented and some algorithm had been covered as well as linked list with c++ class was implemented. I have used c++ language and this projects is implemented in codeblocks IDE AND WINDOWS OPERATION SYSTEM. I have covered important operation on dsa and algorithm I hope this projects considered as not bad. I tired my best for doing this projects. I tired to avoid all the error.

### Future Scope

- We can update more thing like student profile , student marksheet, student record modify etc.
- We can add a file system for storing student records and we can add a database
- We can do many thing if we want and we can do large this project . the idea of this project not only for student record system but also person or any others things management system





