

DATA STRUCTURES LAB

Project Title: STUDENT MANAGEMENT SYSTEM



**Jaypee Institute of Information Technology University,
Noida**

SUBMITTED TO:

Ankita Verma

SUBMITTED BY:

Aakash Mehta 20102004

Alisha Sethi 20102029

Paras Rawat 20102015

TABLE OF CONTENTS

1. Candidate's Declaration	3
2. Acknowledgement	4
3. Problem Statement	5
4. Introduction	5
5. List of Data Structures used in the project	6
6. Flowchart	7
7. Detailed Design of the project	8
8. Implementation details and results	9
9. Conclusion	20

CANDIDATES' DECLARATION

This is to certify that the work which is being presented in Data Structures Lab entitled “**Student Management System**”, submitted by “Aakash Mehta, Alisha Sethi and Paras Rawat”, in partial fulfillment of the requirements for the award of degree of **Bachelor of Technology in Electronics & Communication Engineering** and submitted to the Department of Electronics & Communication Engineering of Jaypee Institute of Information Technology, Noida is an authentic record of our own work carried out during a period from August 2022 to December 2022 under the supervision of “**Ankita Verma**”, ECE Department. The matter presented in this report has not been submitted by us for the award of any other degree elsewhere.

ACKNOWLEDGEMENT

We would like to express my profound gratitude to. Prof. Shweta Srivastava (HOD), of Electronics and Communication department, and Prof. D.K Rai (Dean) of Jaypee Institute of Information Technology for their contributions to the completion of my project titled “Student Management System”. We would like to express my special thanks to our subject teacher Ankita Verma for her time and efforts he provided throughout the year. Your useful advice and suggestions were really helpful to me during the project’s completion. In this aspect, we are eternally grateful to you. We would like to acknowledge that this project was completed entirely by us.

PROBLEM STATEMENT

Making STUDENT MANAGEMENT SYSTEM using file handling with the help of data structures

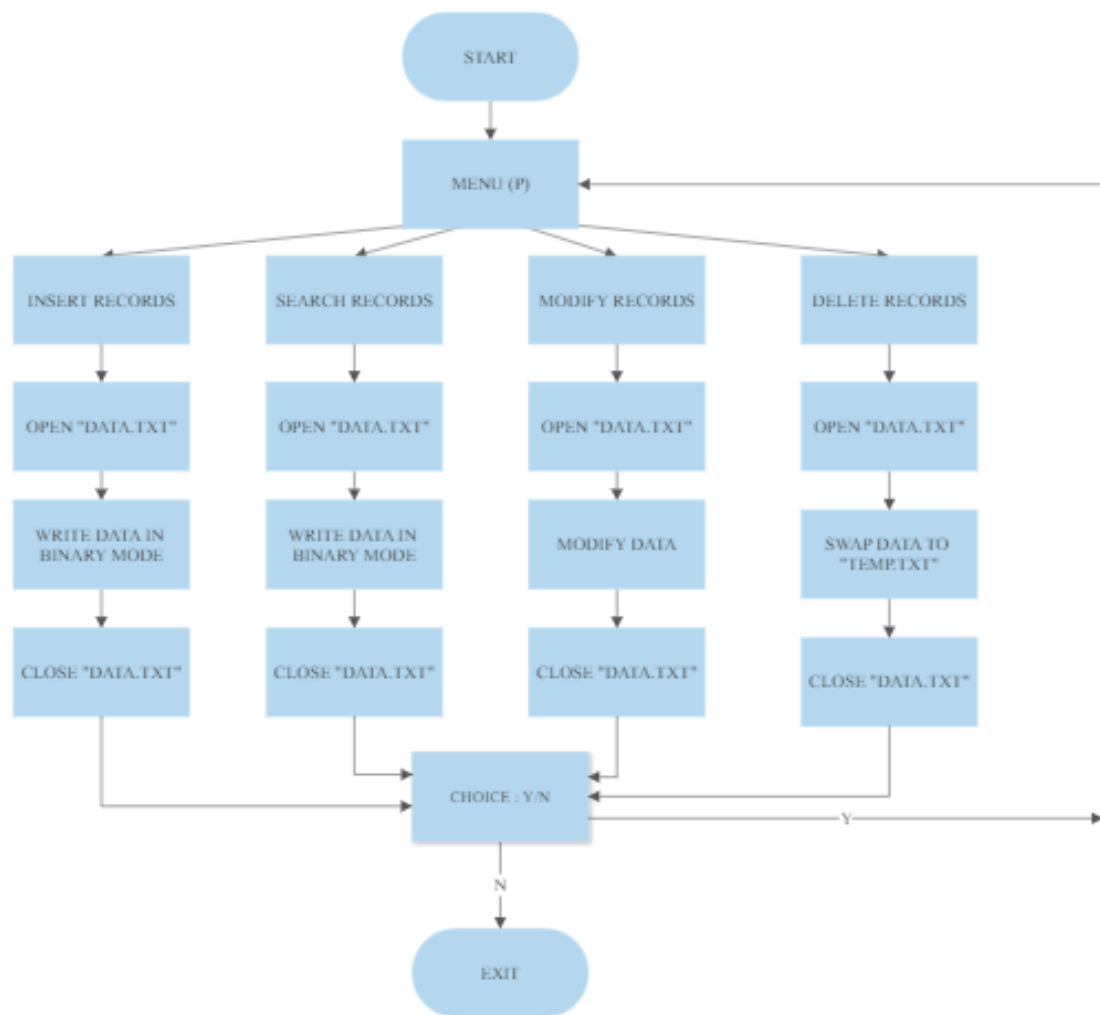
INTRODUCTION

A student management system (software) project is a web-based platform that allows schools and universities to take student data online for improved management and transparency. The main objective of the Student Management System is to manage the details of profiles, courses, logins, exams, fees. It manages all the information about file like inserting, searching, modifying, deleting or even showing any student record. Student Management System is a basic C++ program for education establishments to manage student data and manage many other student-related data needs in a school/college. The user can also modify all the available records easily. Before modifying the student's record, the user has to enter the respective name which shows the old available record. After that, the user can modify each and every field. Deleting a record is too simple here, the user just has to enter the full name of the student. In order to view student record, the user has to search for it. It is done by entering the full name of the student. Then the system displays each and every available detail.

DATA STRUCTURES USED

1. We will be using Link List in our program.
2. Link List is linear data structure.
3. Link List is a very flexible, dynamic data structure in which the element can be added or deleted from anywhere.
4. In the Link list each element is called a node.
5. Node: In node one part is data and another part is a pointer which points to the next node.
6. The last node is a NULL pointer.

FLOWCHART



DETAILED DESIGN OF THE PROJECT

1. Linked List data structure is used to insert the student records in our project
 - A linked list is a linear [data structure](#) that stores a collection of data elements dynamically. Nodes represent those data elements, and links or pointers connect each node. Each node consists of two fields, the information stored in a linked list and a pointer that stores the address of its next node. The last node contains null in its second field because it will point to no node. A linked list can grow and shrink its size, as per the requirement.
2. We used a variety of functions in order to open a file, read it, write more data, create a new file, close or delete a file, search for a file, etc. These are known as file handling operators in C which is a method of storing data in the C program in the form of an output or input that might have been generated while running a C program in a data file, i.e., a binary file or a text file for future analysis and reference in our program
3. We had created the structure in the starting to maintain the data records, we also used it to insert further through linked list we were able to execute several functions in our C program. Pointer to structure holds the add of the entire structure. It is used to create complex data structures such as linked lists, trees, graphs and so on.

IMPLEMENTATION DETAILS

```
#include<stdio.h>
#include<ctype.h>
#include<windows.h>
#include<stdlib.h>
#include<string.h>
```

```
FILE *fp,*tmp;
```

```
COORD coord = {0,0};
```

```
void gotoxy(int x, int y)
```

```
{
    coord.X=x;
    coord.Y=y;
};
```

```
struct data
```

```
{
    char id[20];
    char name[20];
    char add[20];
    char pname[20];
    char Batch[20];
    char phone_no[20];
}s1;
```

```
struct student
```

```

{
    struct data stud;
    struct student *next;
}*head,*temp,*curr;

void main()
{
    int pass, x=10;
    char z[5];
    while (x!=0)
    {
        printf("\nInput the password: ");
        scanf("%d",&pass);
        if (pass==1006)
        {
            printf("Correct password");
            x=0;
            int i,j,choice;
            p:
            system("cls");

            printf("\t\t\tSTUDENT MANAGEMENT SYSTEM \n\n");
            printf("\t\t\t1. Insert Student Record\n");
            printf("\t\t\t2. Search Student Record\n");
            printf("\t\t\t3. Modify Student Record\n");
            printf("\t\t\t4. Delete Student Record\n");
            printf("\t\t\t5. Show All Records\n");

```

```

printf("\t\t\t6. Exit\n\n");
printf("\t\t\tEnter your choice: ");
scanf("%d",&choice);
switch(choice)
{
    case 1:
        system("cls");
        do
        {
            insert();
            printf("\n\n\t\t\tAdd Another Student Record (Y, N): ");
            scanf("%c",z);
        }while(z=='y'||z=='Y');
        break;
    case 2:
        system("cls");
        search();
        break;
    case 3:
        system("cls");
        modify();
        break;
    case 4:
        system("cls");
        del();
        break;
    case 5:

```

```

        system("cls");
        show();
        break;
    case 6:
        exit(0);
        break;
    default:
        printf("INVALID!!");
        break;
    }
    getch();
    goto p;
}
else
{
    printf("Wrong password, try another!!");
}
printf("\n");
}
}

```

```

void insert()
{
    int i;
    struct student *new_node;
    new_node=(struct student*)malloc(sizeof(struct student));
    printf("\n\t\t\t\t\tEnrollment No.: ");

```

```

for(i=0;i<=29;i++)
{
    new_node->stud.id[i]='\0';
    new_node->stud.name[i]='\0';
    new_node->stud.add[i]='\0';
    new_node->stud.Batch[i]='\0';
    new_node->stud.phone_no[i]='\0';
}
scanf("%s",new_node->stud.id);
printf("\t\t\t\tName: ");
scanf("%s",new_node->stud.name);
printf("\t\t\t\tAddress: ");
scanf("%s",new_node->stud.add);
printf("\t\t\t\tParent's name: ");
scanf("%s",new_node->stud.pname);
printf("\t\t\t\tBatch: ");
scanf("%s",new_node->stud.Batch);
printf("\t\t\t\tPhone Number: ");
scanf("%s",new_node->stud.phone_no);
printf("\n\t\t\t\tThe record is successfully added");
new_node->next=NULL;
if(head==NULL)
{
    head=new_node;
    curr=new_node;
}
else

```

```

{
    curr->next=new_node;
    curr=new_node;
}
fp=fopen("data.txt","ab+");
fwrite(&curr->stud, sizeof(curr->stud), 1, fp);
fclose(fp);
}
void search()
{
    int i;
    char s_id[20];
    fp = fopen("data.txt","rb");
    if(fp==NULL)
    {
        printf("\n\t\t\t\tCannot Open File!!");
        return;
    }
    for(i=0;i<=19;i++)
    {
        s_id[i]='\0';
        s1.id[i]='\0';
    }
    int isFound = 0;
    printf("\t\t\t\tEnter ID to Search: ");
    scanf("%s",s_id);

```

```

i=0;
while(fread(&s1, sizeof(s1), 1, fp) == 1)
{
    if(strcmp(s_id,s1.id) == 0)
    {
        isFound = 1;
        break;
    }
}
if(isFound == 1)
{
    printf("\n\n\t\t\t\t\tRECORD FOUND!!\n");
    printf("\n\t\t\t\t\tID: %s",s1.id);
    printf("\n\t\t\t\t\tName: %s",s1.name);
    printf("\n\t\t\t\t\tAddress: %s",s1.add);
    printf("\n\t\t\t\t\tParent's Name: %s",s1.pname);
    printf("\n\t\t\t\t\tBatch: %s",s1.Batch);
    printf("\n\t\t\t\t\tPhone No: %s",s1.phone_no);
}
else
{
    printf("\n\n\t\t\t\t\tSORRY, RECORD NOT FOUND!!");
}
fclose(fp);
printf("%d",i);
return;
}

```

```

void del()
{
    int i,flag;
    int isFound = 0;
    char s_id[20];
    fp = fopen("data.txt","rb");
    if(fp==NULL)
    {
        printf("\n\t\t\t\t\tCannot Open File!!");
        return;
    }
    for(i=0;i<=19;i++)
    {
        s_id[i]='\0';
        s1.id[i]='\0';
    }
    printf("\t\t\t\t\tEnter enrollment no. to delete: ");
    scanf("%s",s_id);
    temp = fopen("temp.txt", "wb");
    i=0;
    if(strcmp(s_id,s1.id) == 0)
    {
        printf("\n\t\t\t\t\tNo Such Record Found!!");
    }
    while(fread(&s1, sizeof(s1), 1, fp) == 1)
    {
        i++;
    }
}

```



```

        if(strcmp(s_id,s1.id) != 0)
        {
            fwrite(&s1, sizeof(s1), 1, temp);
        }
        else
        {
            printf("\n\t\t\t\t\tNo Such Record Found!!");
        }
    }
    fclose(fp);
    fclose(temp);
    remove("data.txt");
    rename("temp.txt","data.txt");
    printf("\n\t\t\t\t\tRecord Deleted Successfully!!\n");
    return;
}

void modify()
{
    char s_id[19];
    int isFound = 0;
    printf("\t\t\t\t\tEnter enrollment no. to Modify: ");
    scanf("%s",s_id);
    fp = fopen("data.txt","rb+");
    while(fread(&s1, sizeof(s1),1,fp) == 1)
    {
        if(strcmp(s_id, s1.id) == 0)
        {

```

```

        printf("\n\t\t\t\tID: ");
        scanf("%s",s1.id);
        printf("\t\t\t\tName: ");
        scanf("%s",s1.name);
        printf("\t\t\t\tAddress: ");
        scanf("%s",s1.add);
        printf("\t\t\t\tParent's name: ");
        scanf("%s",s1.pname);
        printf("\t\t\t\tBatch: ");
        scanf("%s",s1.Batch);
        printf("\t\t\t\tPhone Number: ");
        scanf("%s",s1.phone_no);
        fseek(fp,-sizeof(s1), SEEK_CUR);
        fwrite(&s1,sizeof(s1), 1, fp);
        isFound = 1;
        break;
    }
}
if(!isFound)
{
    printf("\t\t\t\tNo Record Found");
}
else
{
    printf("\t\t\t\tRecord modified\n");
}
fclose(fp);

```

```

    return;
}
void show()
{
    fp=fopen("data.txt","rb");
    if(fp==NULL)
    {
        gotoxy(5,7);
        printf("\t\t\t\tERROR!!");
    }
    gotoxy(35,1);
    printf("\t\t\t\t----- STUDENT DETAILS ----- \n");
    gotoxy(18,3);
    printf("\n\nID\t\t\t\t\tNAME\t\t\t\t\tADDRESS\t\t\t\t\tParentsName\t\t\t\t\tBATCH\t\t\t\t\tCONTACT\n\n");
    gotoxy(18,6);
    while(fread(&s1,sizeof(s1),1,fp)==1)
    {
        printf("\n%-12s\t\t\t\t\t%-12s\t\t\t\t\t%-12s\t\t\t\t\t%-12s\t\t\t\t\t%-12s\t\t\t\t\t%-12s",s1.id,s1.name,s1.add,s1.pname,s1.Batch,s1.phone_no);
    }
    fclose(fp);
    getch();
}

```

EXPLANATION AND RESULTS

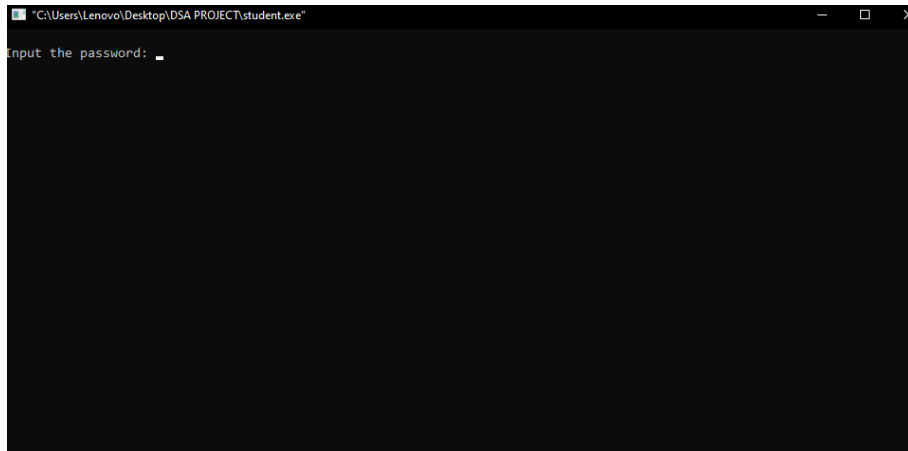


Fig1: Password Protection

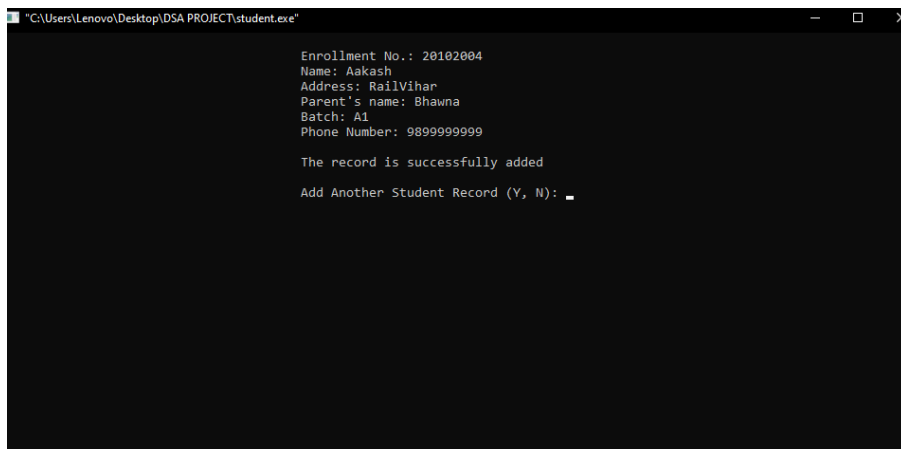


Fig2: Insert Data

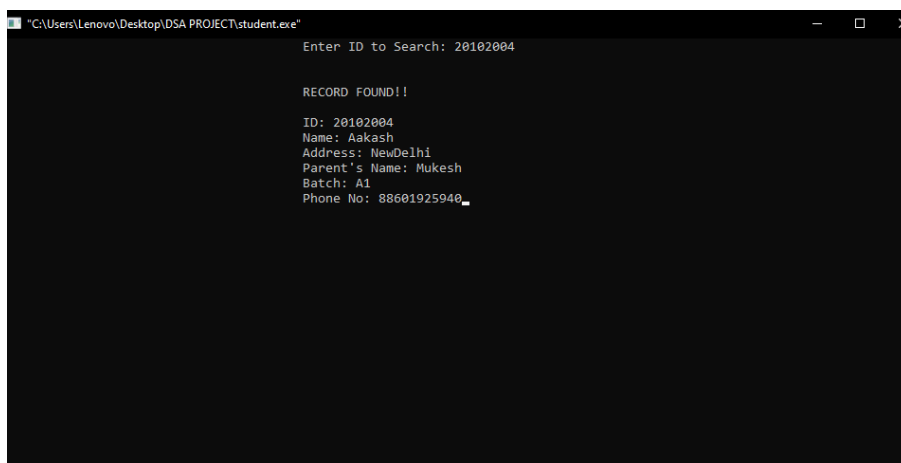


Fig3: Search Data

----- STUDENT DETAILS -----

ID	NAME	ADDRESS	ParentsName	BATCH	CONTACT
20102004	Aakash	NewDelhi	Mukesh	A1	8860192594
20102010	Abhay	YamunaNagar	Sanjay	A1	9999988888
20102029	Alisha	RailVihar	Ashok	A1	9810160994
20102017	Deevansh	Bhatinda	XYZ	A1	8888888889
23	3	33	3	3	3
6	6	6	6	6	6
7	7	7	7	7	7
20102004	Aakash	RailVihar	Bhawna	A1	9899999999

Fig4: Record (Before Modify)

Enter enrollment no. to Modify: 6

ID: ssss
 Name: ssss
 Address: ssss
 Parent's name: ssss
 Batch: sss
 Phone Number: ssssssss
 Record modified

Fig5: Modify Record

----- STUDENT DETAILS -----

ID	NAME	ADDRESS	ParentsName	BATCH	CONTACT
20102004	Aakash	NewDelhi	Mukesh	A1	8860192594
20102010	Abhay	YamunaNagar	Sanjay	A1	9999988888
20102029	Alisha	RailVihar	Ashok	A1	9810160994
20102017	Deevansh	Bhatinda	XYZ	A1	8888888889
23	3	33	3	3	3
sss	sss	sss	sss	sss	ssssssss
7	7	7	7	7	7
20102004	Aakash	RailVihar	Bhawna	A1	9899999999

Fig6: Record (After Modify)

ID	NAME	ADDRESS	ParentsName	BATCH	CONTACT
20102004	Aakash	NewDelhi	Mukesh	A1	8860192594
20102010	Abhay	YamunaNagar	Sanjay	A1	9999988888
20102029	Alisha	RailVihar	Ashok	A1	9810160994
20102017	Deevansh	Bhatinda	XYZ	A1	8888888889
23	3	33	3	3	3
ssss	ssss	ssss	ssss	sss	ssssssss
7	7	7	7	7	7
20102004	Aakash	RailVihar	Bhawna	A1	9899999999

Fig7: Record (Before Deletion)

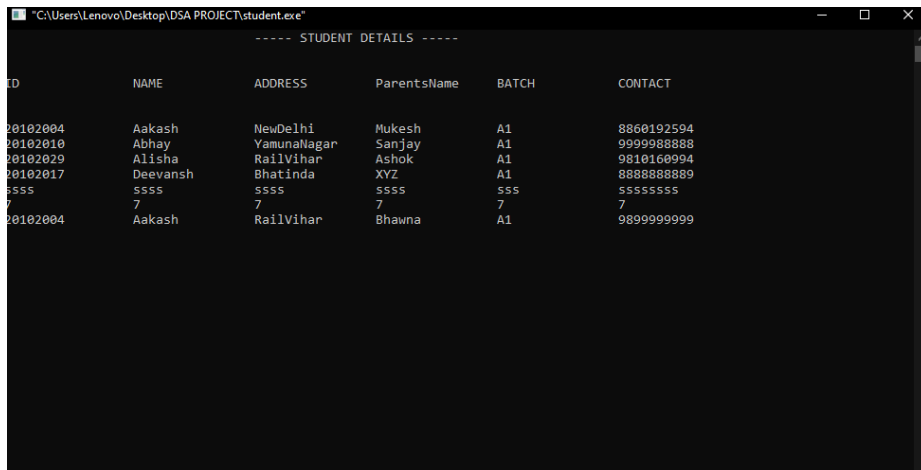
Enter enrollment no. to delete: 23

Record Deleted Successfully!!

Fig8: Deleting Record

ID	NAME	ADDRESS	ParentsName	BATCH	CONTACT
20102004	Aakash	NewDelhi	Mukesh	A1	8860192594
20102010	Abhay	YamunaNagar	Sanjay	A1	9999988888
20102029	Alisha	RailVihar	Ashok	A1	9810160994
20102017	Deevansh	Bhatinda	XYZ	A1	8888888889
ssss	ssss	ssss	ssss	sss	ssssssss
7	7	7	7	7	7
20102004	Aakash	RailVihar	Bhawna	A1	9899999999

Fig9: Record (After Deletion)



ID	NAME	ADDRESS	ParentsName	BATCH	CONTACT
00102004	Aakash	NewDelhi	Mukesh	A1	8860192594
00102010	Abhay	YamunaNagar	Sanjay	A1	9999988888
00102029	Alisha	RailVihar	Ashok	A1	9810160994
00102017	Deevansh	Bhatinda	XYZ	A1	8888888889
ssss	ssss	ssss	ssss	sss	ssssssss
7	7	7	7	7	7
00102004	Aakash	RailVihar	Bhawna	A1	9899999999

Fig10: All Records

CONCLUSION

The project entitled Student Management System was completed successfully. The system has been developed with much care and free of errors and at the same time it is efficient and less time consuming. The purpose of this project was to develop a web application for maintaining student records with much ease. This project helped us in gaining valuable information and practical knowledge on several topics. The entire system is secured. Also the project helped us understanding about the development phases of a project and software development life cycle. We learned how to test different features of a project. This project has given us great satisfaction. There is a scope for further development in our project to a great extent.