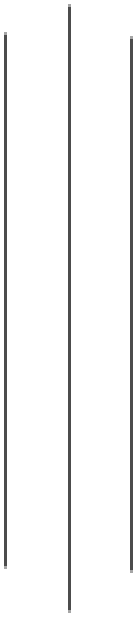

Program Name: (BCS)hons

Course Code: 2514

Course Name: Data Structure and Algorithm

Individual project

Date of Submission: 20th October 2021

Submitted By:

Student Name: Pradip Dhakal

IUKL ID: 042902900047

Semester: 4th

Intake: September 2019

Submitted To:

Faculty Name: Prakash Chandra

Department: LMS

Abstract

Any Education institute can use the Student Management System to easily keep track of their students' records. Manually achieving this goal is challenging since data is fragmented and redundant, and gathering relevant data could take a long period. All of these difficulties are addressed by this initiative.

Introduction

- PURPOSE
  The Student Information System's purpose is to give any organization's administrator access to a student's personal information while also allowing the student to keep his profile up to date.It will also make it easier to maintain track of all of a student's information, such as their ID, name, mailing address, phone number, and date of birth. As a result, all of a student's information will be available in a matter of seconds. Overall, it will make Student Information Management easy for any organization's administrator and students. The primary goal of this report is to demonstrate the project's requirements for the Student Information System and to assist any organization in maintaining and managing its students' personal data.

- Scope

  Managing and maintaining student information without a Student Information System is a time-consuming task for any organization. The Student Information System will contain all of the students' information, including their background, educational qualifications, personal information, and resume information.

Feature

The project does not include a graphical user interface. The project, on the other hand, is fully operational. In order to record the data entered by the students, the linked list idea is used in the project. The project is built using C. The application saves information such as the student's name, phone number, and roll number. Some of the program's features are given below:

- Terminal based Application
- The data is stored using the linked list concept.
- The application can perform basic CRUD operations by unse choice
- Proper menu based application

Testing

Main menu

```
                        STUDENT RECORD MANAGEMENT SYSTEM
 please enter carefully
.....................................................................


1 Insert details
2 Search details
3 Delete  details
4 Update details
5 Display all student details

Enter Choice:
```

Inserting details

Name

Contact num

Address

Age

```
Enter Choice:1
Enter roll number: 1
Enter name: Pradeep dhakal
Enter contact_num: 9861893027
Enter address: Kavrapalanchok


Enter Choice:1
Enter roll number: 2
Enter name: Anup Dhakal
Enter contact_num: 9861456789
Enter address: bhaktapur
```

Displaying Inserted detail

```
DISPLAYING THE RECORDED DETAILS
************************************************************************
Roll Number: 2
Name: Anup Dhakal
contact_num: 9861456789
addressage: bhaktapur
Roll Number: 1
Name: Pradeep dhakal
contact_num: 9861893027
addressage: Kavrapalanchok
```

Searching Details by roll number

```
Enter Choice:2
Enter roll number to search: 1
Roll Number: 1
Name: Pradeep dhakal
contact_num: 9861893027
addressage: Kavrapalanchok
```

If the given roll number is wrong

```
Enter Choice:2
Enter roll number to search: 5
Student with roll number 5 is not found !!!
```

Updating The record by using Roll Number

```
Enter Choice:4
Enter roll number to update: 1
Record with roll number 1 Found !!!
Enter a new name: Binu Bajgain
Enter a new contact_num number: 9840414243
Enter a new addressage: Banepa
Update done Successfully!!!


Enter Choice:2
Enter roll number to search: 1
Roll Number: 1
Name: Binu Bajgain
contact_num: 9840414243
addressage: Banepa
```

Deleting the record

```
Enter Choice:3
Enter roll number to delete: 2
Record with roll number 2 Found!!!
Record Deleted  Successfully !!!
```

Remaining Record after delete one record

```
Enter Choice:5

DISPLAYING THE RECORDED DETAILS
************************************************************************
Roll Number: 1
Name: Binu Bajgain
contact_num: 9840414243
addressage: Banepa
```

Program testing Successful

Conclusion

As a result, the modest project of a student management system has been completed. It is fully functional and meets all of the requirements. The application runs without errors and records all of the student data and essential information. Aside from the linked list concept, which is used in the project, all of the other functionalities for inserting, updating, deleting, and searching student information are also included.

Appendix

Complete code op application

StudentManagemant.C

```
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
struct Student
{
   int rollnumber;
   char name[100];
   char contact_num[100];
   char address[100];
   struct Student *next;

} * head;
void insert(int rollnumber, char *name, char *contact_num, char *address)
{

   struct Student *student = (struct Student *)malloc(sizeof(struct Student));
   student->rollnumber = rollnumber;
   strcpy(student->name, name);
   strcpy(student->contact_num, contact_num);
   strcpy(student->address, address);
   student->next = NULL;

   if (head == NULL)
   {

      head = student;
```

```
    }
    else
    {

        student->next = head;
        head = student;
    }
}
void search(int rollnumber)
{
    struct Student *temp = head;
    while (temp != NULL)
    {
        if (temp->rollnumber == rollnumber)
        {
            printf("Roll Number: %d\n", temp->rollnumber);
            printf("Name: %s\n", temp->name);
            printf("contact_num: %s\n", temp->contact_num);
            printf("addressage: %s\n", temp->address);
            return;
        }
        temp = temp->next;
    }
    printf("Student with roll number %d is not found !!!\n", rollnumber);
}
void update(int rollnumber)
{
    char tem;
    struct Student *temp = head;
    while (temp != NULL)
    {

        if (temp->rollnumber == rollnumber)
        {
            printf("Record with roll number %d Found !!!\n", rollnumber);
            printf("Enter a new name: ");
            scanf("%c", &tem);
            scanf("%[^\n]", temp->name);
            printf("Enter a new contact_num number: ");
            scanf("%s", temp->contact_num);
            printf("Enter a new addressage: ");
            scanf("%c", &tem);
            scanf("%[^\n]", &temp->address);
            printf("Update done Successfully!!!\n");
```

```
            return;
        }
        temp = temp->next;
    }
    printf("Student with roll number %d is not found !!!\n", rollnumber);
}
void Delete(int rollnumber)
{
    struct Student *temp1 = head;
    struct Student *temp2 = head;
    while (temp1 != NULL)
    {

        if (temp1->rollnumber == rollnumber)
        {

            printf("Record with roll number %d Found!!! \n", rollnumber);

            if (temp1 == temp2)
            {
                head = head->next;
                free(temp1);
            }
            else
            {
                // temp1 is the node we need to delete
                // temp2 is the node previous to temp1
                temp2->next = temp1->next;
                free(temp1);
            }

            printf("Record Deleted  Successfully !!!\n");
            return;
        }
        temp2 = temp1;
        temp1 = temp1->next;
    }
    printf("Student not found %d!!!\n", rollnumber);
}
void display()
{
    struct Student *temp = head;
    while (temp != NULL)
    {
```

```
        printf("Roll Number: %d\n", temp->rollnumber);
        printf("Name: %s\n", temp->name);
        printf("contact_num: %s\n", temp->contact_num);
        printf("addressage: %s\n", temp->address);
        temp = temp->next;
    }
}
int main()
{
    head = NULL;
    int choice;

    int rollnumber;
    char name[100];
    char contact_num[100];
    char address[100];

    char temp;
    printf("\n\n\t\t\t STUDENT RECORD MANAGEMENT SYSTEM\n ");
    printf("please enter carefully");

    printf("\n.....\n");

        printf("\n\n1 Insert details\n2 Search details\n3 Delete  details\n4 Update details\n5 Display all student details");

    do
    {
        printf("\n\nEnter Choice:");
        scanf("%d", &choice);
        switch (choice)
        {
        case 1:
            printf("Enter roll number: ");
            scanf("%d", &rollnumber);
            printf("Enter name: ");
            scanf("%c", &temp);
            scanf("%[^\n]", name);
            printf("Enter contact_num: ");
            scanf("%s", contact_num);
            printf("Enter address: ");
            scanf("%c", &temp);
            scanf("%[^\n]", address);
```

```
            insert(rollnumber, name, contact_num, address);
            break;
        case 2:
            printf("Enter roll number to search: ");
            scanf("%d", &rollnumber);
            search(rollnumber);
            break;
        case 3:
            printf("Enter roll number to delete: ");
            scanf("%d", &rollnumber);
            Delete(rollnumber);
            break;
        case 4:
            printf("Enter roll number to update: ");
            scanf("%d", &rollnumber);
            update(rollnumber);
            break;
        case 5:
            printf("\nDISPLAYING THE RECORDED DETAILS\n");

printf("**********************************************************************
****\n");
            display();
            break;
        }

    } while (choice != 0);
}
```

Thank You