PROJECT REVIEW 3

VIDEO LINK: https://youtu.be/f1uNF-yNuWM

VTOP 2.0

DATA STRUCTURE AND ALGORITHM
CSE2003

Team Members:
Anurag Pandey -19BCE0345
Jay Chadawar – 19BCE0360
Samyak Jain - 19BCE2516

ABSTRACT:

"Efficiency" is one of the most important aspects for any individual in a University, be it a student, professor or a warden. The Project **"VTOP 2.0"** is aimed at developing such an efficient program that helps in recording the **student, faculty, warden data** updating the data given and performing many other tasks whenever necessary. The idea behind this proposed project is **to develop consistent, robust and user-friendly program** that allows all the users to have **easy and fast access** to the required data. This service will work round the clock and thus will allow the users to perform their functions based on their convenience. It will be designed to **accommodate the large number of users** and facilitate on-the-go access to all.

KEYWORD:

Jump Search, Linear Search, File handling, Classes and Objects

AIM:

Our primary aim is to build a student management system wherein students, faculties and wardens can log in and have access to the various features designed for them.

OBJECTIVE:

Our objective is to use the concepts learnt in data structures and algorithms and apply it to implement a feasible project that allows the operation and working of the student management system in an efficient manner.

APPLICABILITY:

This algorithm may be of utmost use in creating an effective student management system for the benefit of the students at Vellore Institute of Technology to perform their daily academic and some non-academic activities that are necessary in the University. For example: apply leave, maintain attendance, view marks etc. It facilitates using these facilities on any device using an internet connection

Hence, this project seeks to give an effective management system to students at VIT.

INTRODUCTION:

This project is basically a student management system, where the faculty of the college, the students of the college and the hostel wardens have separate logins.

- Each of the above mentioned can login using their correct username and password and alter details.

- Students can view their profile details, can view their marks, can apply for leave. They can also modify the details that they wish to modify.

- Faculty of the college can view their faculty details; they can upload marks of the student under them. They can view the marks of all the students registered under them.

- The hostel wardens have the provision to approve or disapprove student’s leave request and view any student’s details. They also mark the hostel attendance for each day.

ALTERNATIVE METHOD AND DRAWBACKS:

In this system we have used file handling system which is not very secure and responsive. Hence this method is not being used nowadays. In place of file handling technique we can use any DBMS software like MySQL or Oracle SQL.

PROPOSED METHOD AND ADVANTAGE OF METHOD:

In place of file handling method we have proposed the method of database management system software such as structured query language.

Its primary advantages are :-

- No redundant data: Redundancy removed by data normalization. No data duplication saves storage and improves access time.
- Data Consistency and Integrity: As we discussed earlier the root cause of data inconsistency is data redundancy, since data normalization takes care of the data redundancy, data inconsistency also been taken care of as part of it
- Data Security: It is easier to apply access constraints in database systems so that only authorized user is able to access the data. Each user has a different set of access thus data is secured from the issues such as identity theft, data leaks and misuse of data.
- Privacy: Limited access means privacy of data.
- Easy access to data – Database systems manages data in such a way so that the data is easily accessible with fast response times.
- Easy recovery: Since database systems keeps the backup of data, it is easier to do a full recovery of data in case of a failure.
- Flexible: Database systems are more flexible than file processing systems.

ARCHITECTURE:

The entire project as a whole can be divided into 3 modules to make explanations simple, the three
modules being:

1.The student portal
2.The faculty portal
3.The warden portal

This “college database management system” can be seen as a database which contains details of every student and faculty of all sorts encrypted into binary files for safety and ease of access and also acts as a portal where various formalities such as applying for leave, updating of marks takes place.

Access of information is also limited to only the owner as a result of the security measures that are undertaken. The process starts off with a menu which allows the user to choose between the 3 modules to access.

MODULE: Student

This module begins off with the user requiring to enter the registration number of the student as well as the password. The entered data is then matched with all the student directories present which are stored in the form of a linked list. If no match appears, access is denied.

Upon successful authorization, the user is then shown a menu of the various features that are available, namely:

1.View profile
2.Modify profile
3.View marks
4.Apply for leave
5.View leave
6.View attendance
7.Return to main menu

When the profile related options are chosen, the details of the student are shown, all of which are stored in an encrypted binary file, ‘student.dat’. The details being, the student’s registration number, room number, email address, blood group, address etc.

Choosing the view marks option results in the marks of each semester of the student to be displayed in a tabular format.

When applying for leave, the user is asked to enter the from date and the to date, the dates entered are then checked for legitimacy, meaning any from date that happens to be after the to date is forbidden and any dates entered in the incorrect format are forbidden. In addition to this, the presence of extra days in leap and non-leap years are also taken into account.

The viewing attendance sub module displays the number of days the student has been present or absent in the hostel during the in-time of the students. The percentage attendance for the entire year is also calculated and displayed. Any other option entered by the user is rendered invalid and the same menu is displayed once again.

Data structures used: linked lists
Searching algorithms -Jump Search

MODULE: Faculty

The faculty portal also begins off in a similar manner to the student portal, with the user requiring to enter the faculty no. and corresponding password. The details of which are once again matched with a database of details stored in the form of a linked list.

Upon successful authorization, the next menu is then displayed to the user which contains the following features:

1.View profile
2.View student marks
3.Enter student marks
4.Return to main menu

The view profile is the same as the view profile of that of student module, the only difference being that the details of each faculty is

stored in a separate binary file called ‘faculty.dat’.
When the view student marks is chosen, the marks of all students of the class are then displayed on the screen in the order of registration number. The enter student marks works in a similar way to the view function. The marks of all the students are first displayed after which the faculty is required to enter the registration number of the student whose marks are to be entered. Upon successful entering of marks, the data collected by the user are written into the corresponding student’s mark structure in the binary file ‘student.dat’.

Data structures used -linked lists
Searching algorithms -Recursive linear search

MODULE: Warden

The beginning is the same as any other module, with the user being required to enter the warden number and password with the data entered being checked with data already present in the form of a linked list.
Upon successful authorization, the following menu is displayed to the user:

1.Mark attendance
2.Approve leave
3.View student details
4.Return to main menu

When the ‘mark attendance’ sub module is chosen, the registration number of each student who stays in the pertaining warden’s hostel is displayed one by one with the help and the warden then either marks the student as present or absent and the data collected from the user is reflected on the student structures present in the binary file ‘student.dat’.
When the ‘Approve leave’ sub module is chosen, the user is then displayed the list of all students along with their registration numbers and ‘from’ and ‘to’ dates, who have applied for leave, with the one who applied earliest being first. This is accomplished with the help of the queue data structure. The user is then required to enter a ‘y’ in order to

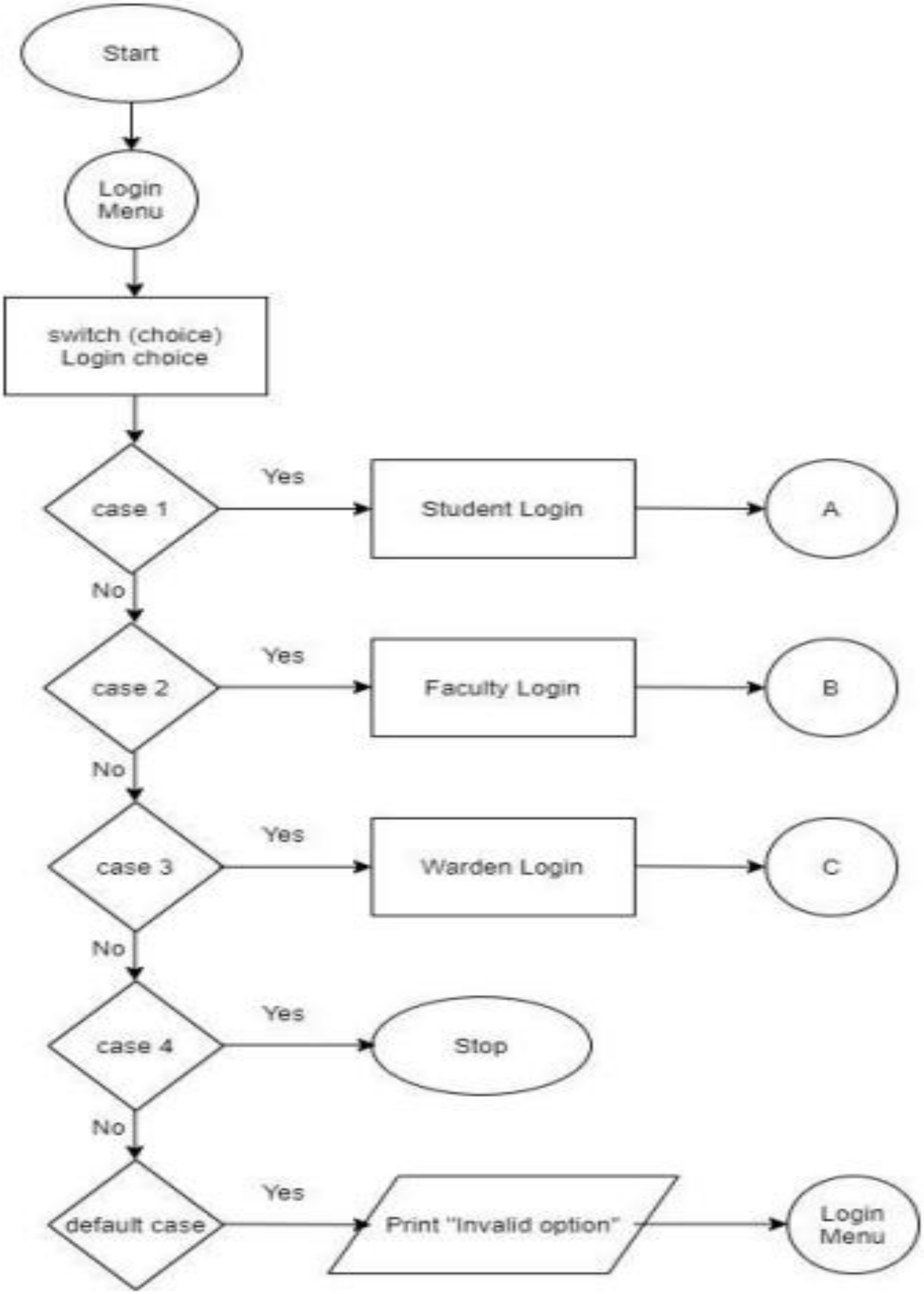
approve the leave or a ‘n’ in order to reject the leave. If there are no leave approvals, a message indicating the same is displayed. When the ‘View student details’sub module is invoked, the user is required to enter the registration number of the student to proceed. Upon entering valid credentials, the ‘student profile’ described in the student module is displayed along with the student’s room number as well as the attendance history of the student.

Data structures used -linked list, queues
Searching algorithms -Linear search

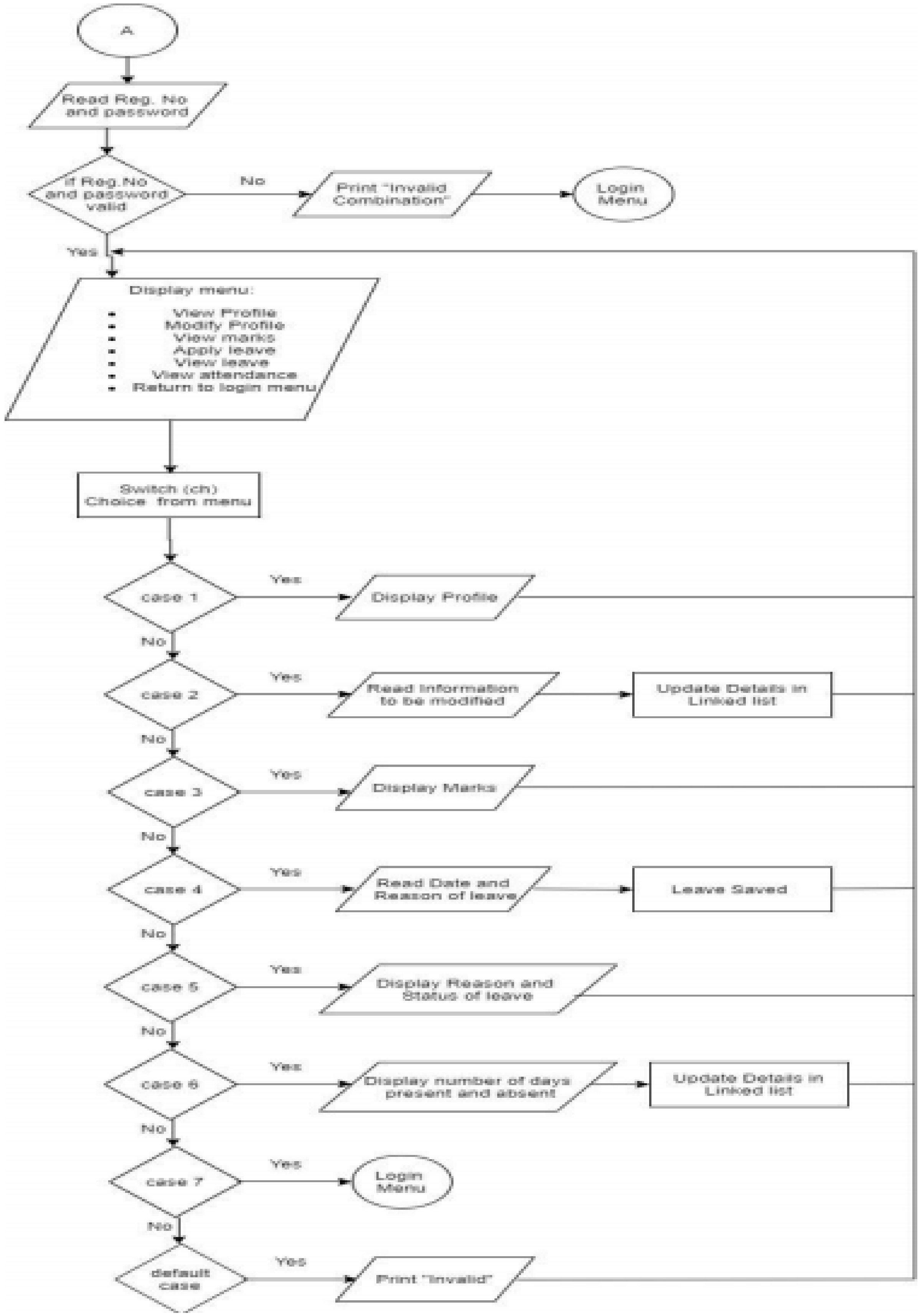
The above information gives a glimpse of the working of the program as well as the intricate details of each function, class and structure implemented in various parts and modules of the program. Though a ‘college database management system’ may sound lackluster, the data structures and modules implemented in this very program makes the running of any college that may use this program easy and efficient in a very user friendly way and with appropriate safety measures implemented.

IMPLEMENTATION DETAILS:

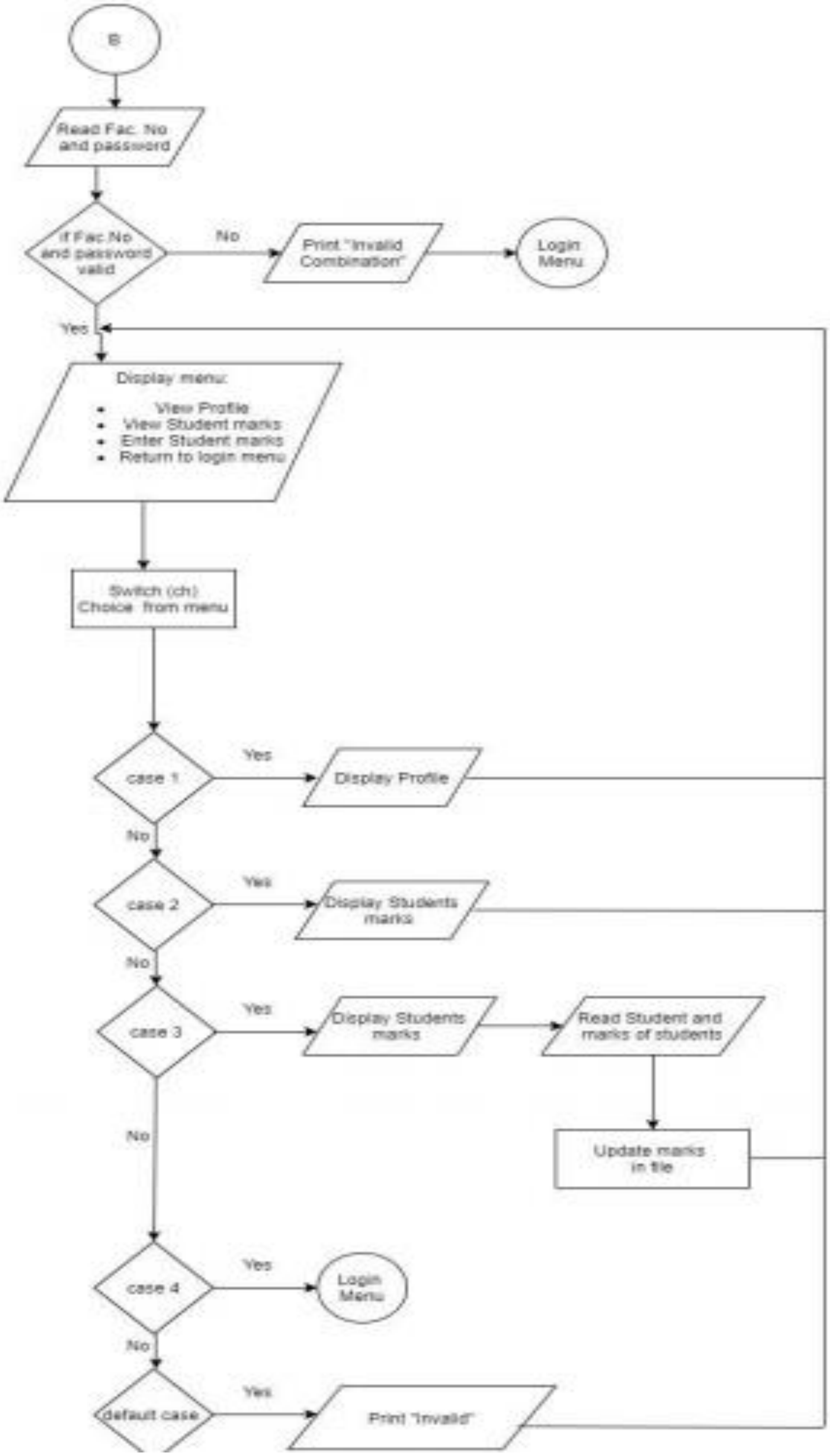
Choice of login:



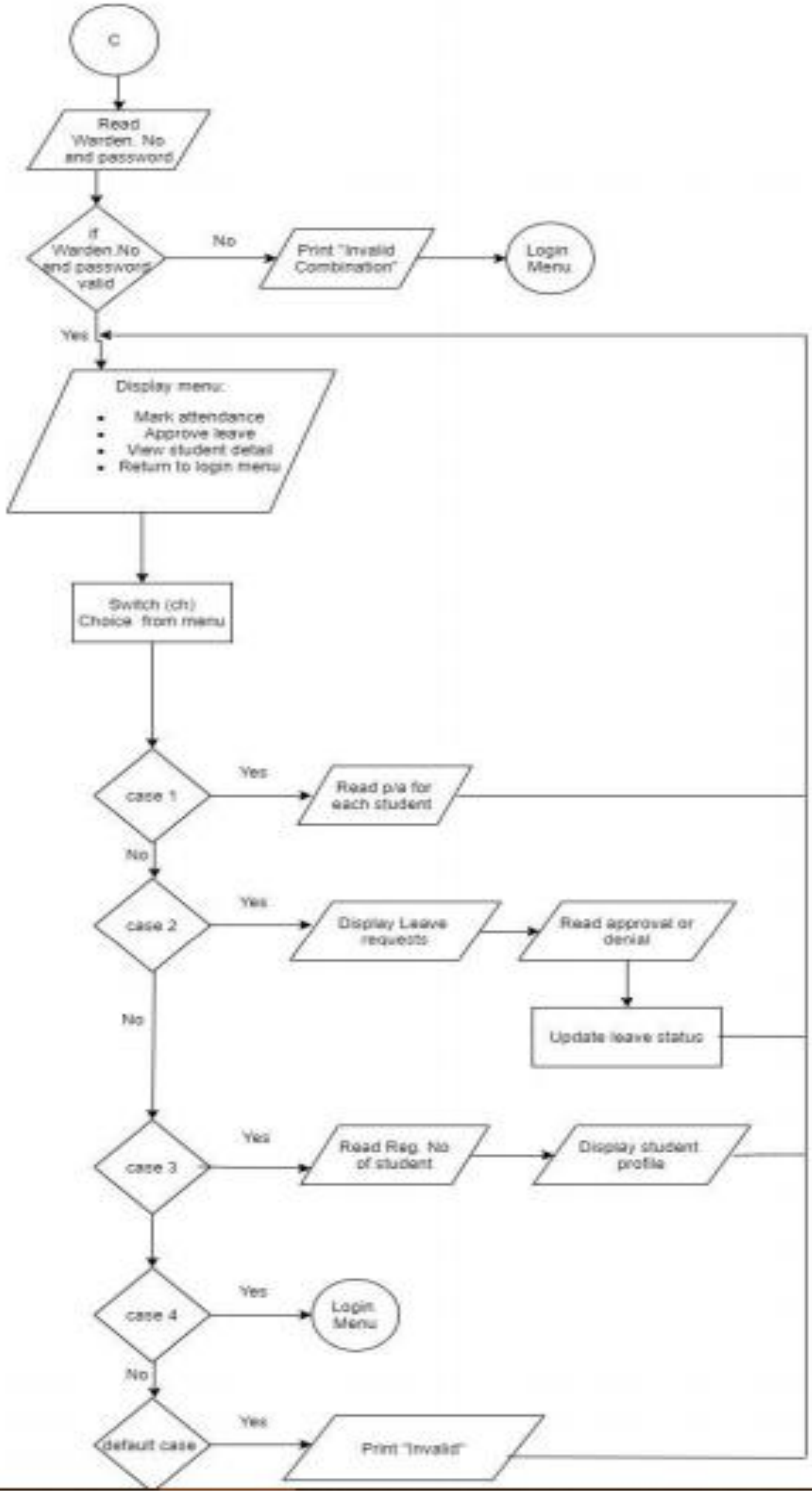
Student Login:


A
Read Reg. No and password
if Reg.No and password valid
No
Print "Invalid Combination"
Login Menu
Yes
Display menu:
View Profile
Modify Profile
View marks
Apply leave
View leave
View attendance
Return to login menu
Switch (ch) Choice from menu
case 1
Yes
Display Profile
No
case 2
Yes
Read Information to be modified
Update Details in Linked list
No
case 3
Yes
Display Marks
No
case 4
Yes
Read Date and Reason of leave
Leave Saved
No
case 5
Yes
Display Reason and Status of leave
No
case 6
Yes
Display number of days present and absent
Update Details in Linked list
No
case 7
Yes
Login Menu
No
default case
Yes
Print "Invalid"

Faculty Login:



Warden Login:



CODE:

```
Code:
#include<iostream>
#include<iomanip>
#include<stdlib.h>
#include<math.h>
#include<string.h>
#include<fstream>
#include<process.h>
#include<limits>
#include<ios>
using namespace std;

//Some variables needed for checking date format
const int MAX_VALID_YR = 9999;
const int MIN_VALID_YR = 1800;
int s_size=9;
int f_size=5;
int w_size=1;

//classes marks,leave and profile are part of class student
class marks
{
        char sub[5][10];
        float cat1[5],cat2[5],fat[5];
        public:


                marks()
                {
                        strcpy(sub[0],"MAT1002");
                        strcpy(sub[1],"CSE1001");
                        strcpy(sub[2],"CSE1002");
                        strcpy(sub[3],"ENG0001");
                        strcpy(sub[4],"PHY1001");

                        for(int i=0;i<5;i++)
                        {
                                cat1[i]=cat2[i]=fat[i]=-1;
                        }
                }


                void enter_marks(char c[])//for getting input from the teacher for the marks of a
particular subject
                {
                        char code[10];
                        int pos=-1;
                        strcpy(code,c);
                        for(int i=0;i<5;i++)
                        {
                                if(strcmp(code,sub[i])==0)
                                {
                                    pos=i;
                                    break;
                                }
                        }
                        if(pos>=0)
                        {
                                cout<<endl<<"Enter the marks for subject "<<sub[pos]<<endl;
                                int n;
                                cout<<"1:CAT 1\t\t2:CAT 2\t\t3:FAT"<<endl<<endl;
                                cout<<"Enter your option:";
                                cin>>n;
                                switch(n)
                                {
                                    case 1:
                                        while (cout << "Enter the marks:" && !(cin >>
cat1[pos])) {
                                    std::cin.clear(); //clear bad input flag
                                    std::cin.ignore(std::numeric_limits<std::streamsize>::max(),
'\n'); //discard input
                                    std::cout << "Invalid input; please re-enter.\n";
                                        }
                                        /*cout<<"Enter the marks:";
                                        cin>>cat1[pos];
                                        cin.clear();
                                        cin.ignore(numeric_limits<streamsize>::max(), '\n');
                                        cout<<"Invalid input";*/
                                        cout<<endl;
                                        break;
                                    case 2:
                                        while (cout << "Enter the marks:" && !(cin >>
cat2[pos])) {
                                    std::cin.clear(); //clear bad input flag
                                    std::cin.ignore(std::numeric_limits<std::streamsize>::max(),
'\n'); //discard input
                                    std::cout << "Invalid input; please re-enter.\n";
                                        }
                                        /* cout<<"Enter the marks:";
                                        cin>>cat2[pos]; */
                                        cout<<endl;
                                        break;
                                    case 3:
                                        while (cout << "Enter the marks:" && !(cin >>
fat[pos])) {
                                    std::cin.clear(); //clear bad input flag
                                    std::cin.ignore(std::numeric_limits<std::streamsize>::max(),
'\n'); //discard input
                                    std::cout << "Invalid input; please re-enter.\n";
                                        }
                                        /*cout<<"Enter the marks:";
                                        cin>>fat[pos]; */
                                        cout<<endl;
                                        break;
                                    default:
                                        cout<<"You have entered the wrong option. Pls start
from the begining."<<endl;
```

```
                        break;
                    }
                    //system("cls");

                }
                else
                    cout<<"You have entered the wrong option. Pls start from the
begining."<<endl;

            }//input marks


        void print_marks(char a[]="Nil")//a contains teacher code if displaying for teacher
        {
            int pos=-1;
            if(strcmp(a,"Nil")!=0)
            {
                for(int i=0;i<f_size;i++)
                {
                    if(strcmp(a,sub[i])==0)
                    {
                        pos=i;
                        break;
                    }
                }
                if(pos!=-1)
                {
                    if(cat1[pos]!=-1)
                    {
                        cout<<"CAT 1:"<<cat1[pos]<<'\t';
                    }
                    else
                    cout<<"CAT 1:"<<"Nil"<<'\t';

                if(cat2[pos]!=-1)
                    {
                        cout<<"CAT 2:"<<cat2[pos]<<'\t';
                    }
                    else
                    cout<<"CAT 2:"<<"Nil"<<'\t';
                if(fat[pos]!=-1)
                    {
                        cout<<"FAT :"<<fat[pos]<<endl;
                    }
                    else
                    cout<<"FAT :"<<"Nil"<<endl;

                }
                else
                    cout<<"You have entered the wrong subject code. Pls start
from the begining."<<endl;
            }
            else
            {
                for(int i=0;i<5;i++)
                {
```

```
                                        pos=i;
                                        cout<<sub[pos]<<'\t';
                                        if(cat1[pos]!=-1)
                                        {
                                                cout<<"CAT 1:"<<cat1[pos]<<'\t';
                                        }
                                        else
                                        cout<<"CAT 1:"<<"Nil"<<'\t';

                                if(cat2[pos]!=-1)
                                        {
                                                cout<<"CAT 2:"<<cat2[pos]<<'\t';
                                        }
                                        else
                                        cout<<"CAT 2:"<<"Nil"<<'\t';
                                if(fat[pos]!=-1)
                                        {
                                                cout<<"FAT :"<<fat[pos]<<endl;
                                        }
                                        else
                                        cout<<"FAT :"<<"Nil"<<endl;
                                }
                        }
                }
};
class leave
{
        public:
                char frm[10],tt[10];
        char from[10],to[10],reason[200];
        bool permission,applied;
                leave()
                {
                        permission=false;
                        applied=false;
                }
                /*check1(char a[],char b[])
                {
                        strcpy(from,a);
                        strcpy(to,b);
                }*/
                bool isLeap(int year)
                {
                 // Return true if year is a multiple pf 4 and
                 // not multiple of 100.
                 // OR year is multiple of 400.
                 return (((year%4==0) && (year%100!=0)) ||
                      (year%400==0));
                }

                // Returns true if given year is valid or not.
                bool check(char fr[])
                {
                        int d, m, y;
                        d=((fr[0]-48)*10)+(fr[1]-48);
                        m=((fr[3]-48)*10)+(fr[4]-48);
                y=((fr[6]-48)*1000)+((fr[7]-48)*100)+((fr[8]-48)*10)+(fr[9]-48);
           // If year, month and day are not in given range
           if (y > MAX_VALID_YR || y < MIN_VALID_YR)
            return false;
           if (m < 1 || m > 12)
            return false;
           if (d < 1 || d > 31)
            return false;

           // Handle February month with leap year
           if (m == 2)
           {
             if (isLeap(y))
               return (d <= 29);
             else
               return (d <= 28);
           }

           // Months of April, June, Sept and Nov
           // must have number of days less than
           // or equal to 30.
           if (m==4 || m==6 || m==9 || m==11)
             return (d <= 30);

           return true;
        }

        bool check_duration(char fr[],char to[])
        {
                int dayf,dayt,mf,mt;
                bool chk=true;
                dayf=((fr[0]-48)*10)+(fr[1]-48);
                dayt=((to[0]-48)*10)+(to[1]-48);
                mf=((fr[3]-48)*10)+(fr[4]-48);
                mt=((to[3]-48)*10)+(to[4]-48);
                if((mf==mt))
                {
                        if(dayf>dayt)
                                chk=false;
                }
                if(mt<mf)
                        chk=false;
                return chk;
        }
        bool find()
        {
if((check(from)==true)&&(check(to)==true)&&(check_duration(from,to)==true))
                {
                        return true;
                }
                else
                return false;
        }
        /*void copydate()
        {
            strcpy(from,date1);
            strcopy(to,date2);
        }*/


    void apply_leave(int n=0)
    {

    char date1[10];
    char temp[2],yr[4];
    char date2[10];
    int fd,fm,fy,td,tm,ty;
    if(n==1)
    {
    permission=false;
    cout<<"Enter the from date(dd): ";
        cin>>temp;
            strcpy(date1,temp);
            strcat(date1,"-");
            cout<<"Enter to month (mm): ";
            cin>>temp;
            strcat(date1,temp);
            strcat(date1,"-");
            cout<<"Enter to year (yyyy): ";
            cin>>yr;
            strcat(date1,yr);
            strcpy(from,date1);
        //strcpy(frm,from);
    cout<<"Enter the to date(dd): ";
        cin>>temp;
            strcpy(date2,temp);
            strcat(date2,"-");
            cout<<"Enter to month (mm): ";
            cin>>temp;
            strcat(date2,temp);
            strcat(date2,"-");
            cout<<"Enter to year (yyyy): ";
            cin>>yr;
            strcat(date2,yr);
            strcpy(from,date1);
            strcpy(to,date2);
            fd=((from[0]-48)*10)+(from[1]-48);
            fm=((from[3]-48)*10)+(from[4]-48);
            fy=((from[6]-48)*1000)+((from[7]-48)*100)+((from[8]-48)*10)+(from[9]-
48);
            td=((to[0]-48)*10)+(to[1]-48);
            tm=((to[3]-48)*10)+(to[4]-48);
            ty=((to[6]-48)*1000)+((to[7]-48)*100)+((to[8]-48)*10)+(to[9]-48);
            if(check(from)==false)
                {
                    cout<<endl<<"Wrong input for from date";
                    return;
                }
            if(check(to)==false)
                {
                    cout<<endl<<"Wrong input for to date";
                    return;
                }
            if(find()==true)
            {
                cout<<"Enter your reason for leave: ";
                cin.ignore();
                cin.getline(reason,200);
                applied=true;
                cout<<endl<<"Leave Saved."<<endl;
            }
            else
            {
                cout<<endl<<"From date cannot be greater than to date";
                return;
            }

    }


            if((find()==true))
            {
                //cout<<"From: "<<fd<<"-"<<fm<<"-"<<fy<<endl;
                //cout<<"To: "<<td<<"-"<<tm<<"-"<<ty<<endl;
                cout<<"Reason: "<<reason<<endl;
                cout<<"Leave status: "<<((permission==true)?"Approved":"Not
Approved")<<endl;
            }
            else
            cout<<"leave not yet appplied"<<endl;
    }

};

class profile
{
    public:
    char father[50],mother[50],address[100],phone[15],blood[4],email[50],name[50];


        //functios for input
        void get_profile()
        {
            cout<<"Enter The Details of the student mentioned below, in the same
order"<<endl;
            cout<<"Student's name,Father's name, Mother's name, Address, phone,
blood group, Email-ID"<<endl;
            cin.ignore();
            cin.getline(name,50);
            cin.getline(father,50);
            cin.getline(mother,50);
            cin.getline(address,100);
            cin.getline(phone,15);
            cin.getline(blood,4);
            cin.getline(email,50);
        }
        void print_profile()//??to print the details
```

```
{
    cout<<"Student's name: "<<name<<endl;
    cout<<"Father's name : "<<father<<endl;
    cout<<"Mother's name : "<<mother<<endl;
    cout<<"Address : "<<address<<endl;
    cout<<"Phone Number : "<<phone<<endl;
    cout<<"Blood group : "<<blood<<endl;
    cout<<"Email-ID : "<<email<<endl;
}
void modify_profile()
{
    char a[50];
    char detail[20];
    cout<<"Enter the detail you want to modify(Please enter the detail name as given in the profile): ";
    cin.ignore();
    cin.getline(detail,20);
    cout<<"Enter the change you want to make : ";
    //cin.ignore();
    cin.getline(a,50);
    cout<<endl;
    if(strcmpi(detail,"Father's name")==0)
    {
        int r=1;
        for(int p=0;a[p]!='\0';++p)
        {
            if(!isalpha(a[p]))
            {
                r=0;
                cout<<"Invalid format"<<endl;
                break;
            }
        }
    if(r==1)
    {
        strcpy(father,a);
        cout<<"Changes Saved."<<endl;
}
    }
    else if(strcmpi(detail,"mother's name")==0)
    {
        int r=1;
        for(int p=0;a[p]!='\0';++p)
        {
            if(!isalpha(a[p]))
            {
                r=0;
                cout<<"Invalid format"<<endl;
                break;
            }
        }
    if(r==1)
    {
        strcpy(mother,a);
        cout<<"Changes Saved."<<endl;
}
```

```
            }
            else if(strcmpi(detail,"Student's name")==0)
            {
                int r=1;
                for(int p=0;a[p]!='\0';++p)
                {
                    if(!isalpha(a[p]))
                    {
                        r=0;
                        cout<<"Invalid format"<<endl;
                        break;
                    }
                }
                if(r==1)
                {
                strcpy(name,a);
                cout<<"Changes Saved."<<endl;
            }
            }
            else if(strcmpi(detail,"Address")==0)
            {

                strcpy(address,a);
                cout<<"Changes Saved."<<endl;
            }
                else if(strcmpi(detail,"Phone number")==0)
            {
                int z=1;
                for(int q=0;a[q]!='\0';++q)
                {
                    if(!isdigit(a[q]))
                    {
                        z=0;
                        cout<<"Invalid number"<<endl;
                        break;
                    }

                }
                if(z==1)
                {

                strcpy(phone,a);
                cout<<"Changes Saved."<<endl;
            }
            }
                else if(strcmpi(detail,"Blood group")==0)
            {
                if(strcmp(a,"A-")!=0&&strcmp(a,"B-
")!=0&&strcmp(a,"A+")!=0&&strcmp(a,"B+")!=0&&strcmp(a,"AB+")!=0&&strcmp(a,"AB-
")!=0&&strcmp(a,"O+")!=0&&strcmp(a,"O+")!=0)
                    cout<<"Invalid type"<<endl;
                else
                {
                strcpy(blood,a);
                cout<<"Changes Saved."<<endl;
            }
            }
                else if(strcmpi(detail,"Email-ID")==0)
            {
                if(strlen(a)<9)

                    cout<<"Invalid format"<<endl;
                    else{

                    char b[50];
                    int i;
                    for(i=0;i<8;++i)
                    {
                        b[i]=a[strlen(a)-i-1];
                    }
                    b[i]='\0';

                if(strcmp(b,"moc.zyx@")!=0)
                    cout<<"Invalid format"<<endl;
                else{

                strcpy(email,a);
                cout<<"Changes Saved."<<endl; }}
            }

            else
            {
                cout<<"Entered detail not in the same as given in the profile"<<endl;

            }

        }
    };


//All student related functions and classes

//class student contains all student info
class student
{
    public:
    profile p;
    marks m;
    leave l;
    char reg[20];
    int room_no;
    char password[20];
    int pcount;
    int acount;
    student *next;
    student()
    {
        pcount=0;
        acount=0;
    }
    void get_student()
```

```
	{
		cout<<"reg:";
		cin>>reg;
		cout<<"password:";
		cin>>password;
		cout<<"room:";
		cin>>room_no;
		p.get_profile();


	}
	void print_attendence()
	{
		cout<<"Number of days present:"<<pcount<<endl;
		cout<<"Number of days absent:"<<acount<<endl;
		cout<<"Attendance Percentage:"<<((pcount*100)/(pcount+acount))<<"%";
	}


};

struct node
{
	student x;
	node *next;
}*top=NULL;

//extracts all the data in the student file to an array of type student(s[])
void extract_to(student s[])
{
	ifstream in("student.dat",ios::in|ios::binary|ios::app);
	in.seekg(ios::beg);
	int i;
	for(i=0;i<s_size;i++)
	{
		node *temp;
		temp=new node;
		in.read((char*)&s[i],sizeof(s[i]));
		temp->x=s[i];
		temp->next=top;
		top=temp;

	}
	in.close();

}

//returns podition of the object in student file with reg_no=rno
/* int find_student(int rno)
{
	student s;
	ifstream fin;
	fin.open("student.dat",ios::in|ios::binary);
	while(fin.read((char*)&s,sizeof(s)))
	{
		if(rno==s.reg)
```

```
                {
                        return(fin.tellg()-sizeof(s));
                }
        }
        fin.close();

} */
node* search(node *a,char r[])
{
        node* temp1=a;
        int n=s_size;
        int m=sqrt(n);
        int i;
        int order=0;
        //cout<<a->x.reg<<"a";
        a=a->next->next->next;

        if(strcmp(temp1->x.reg,temp1->next->x.reg)>=0)
         order=-1;
        else
         order=1;

        if(order>0)
        {
        while(m<n&&strcmp(r,a->x.reg)>=0)
        {
                //cout<<"entered asc while";
                for(i=0; i<sqrt(n); i++)
                {
                        if(a!=NULL)
                    a=a->next;
                  temp1=temp1->next;
                }
                m+=sqrt(n);
        }
        for(i=0; i<sqrt(n); i++)
        {
         if(strcmp(r,temp1->x.reg)==0)
                 return temp1;
                temp1=temp1->next;
        }
        }

  else
        {
        while(m<n&&strcmp(r,a->x.reg)<=0)
        {
                //cout<<"entered des while";
                //cout<<"value of temp1"<<temp1->x.reg<<"\n";
                for(i=0; i<sqrt(n); i++)
                {
                        if(a!=NULL)
                    a=a->next;
                  temp1=temp1->next;

                }
                //cout<<"for over";
                m+=sqrt(n);
        }
        //cout<<"value of temp1"<<temp1->x.reg<<"\n";
        for(i=0; i<sqrt(n); i++)
        {
         if(strcmp(r,temp1->x.reg)==0)
             return temp1;
                temp1=temp1->next;
        }
        }

        cout<<"Not found";
        system("pause");
        return NULL;
    }
    //the student login page(asks for password and shows the student menu)
    void s_login()
    {
        char r[20];char p[20],pos;
        student s1[s_size],s;
        cout<<"Reg no:"; cin>>r;

        /*ifstream in("student.dat",ios::in|ios::binary|ios::app);
        in.seekg(ios::beg);
        int i;
        for(i=0;i<3;i++)
        {
                in.read((char*)&s1[i],sizeof(s1[i]));
        }
        in.close();*/
        extract_to(s1);

        /*ifstream i("student.dat",ios::in|ios::binary);
        i.seekg(pos);
        i.read((char*)&s,sizeof(s));
        i.close();
        if(strcmp(s.p.name,"Nil")==0)
        {
                cout<<"Reg number not found!!";
                return;
        }*/
        node *temp=top;
        temp=search(temp,r);
        if(temp==NULL)
                return;
        s=temp->x;
        cout<<"Enter Password:";cin>>p;
        if(strcmp(p,s.password)!=0)
        {
                cout<<"Wrong Reg no/Password combination!!"<<endl;
                system("pause");
                return;
        }
                char ch='0';
                while(ch!='7')
```

```
              {

              system("cls");
              cout<<"\n\t\t\t\tSTUDENT LOGIN\n"<<endl;
              cout<<"Student reg no: "<<s.reg<<endl<<endl;
              cout<<"1-View profile\n2-Modify profile\n3-View marks\n4-Apply leave\n5-To view leave\n6-To view attendance\n7-Return to main menu\n\nEnter your choice : ";
              cin>>ch;
    switch(ch)
              {
                    case '1':
                          cout<<endl;
                          s.p.print_profile();
                          cout<<endl;
                          system("pause");
                          break;
                    case '2':
                          cout<<endl;
                          s.p.print_profile();
                          cout<<endl;
                          s.p.modify_profile();
                          system("pause");
                          cout<<endl;
                          break;
                    case '3':
                          cout<<endl;
                          s.m.print_marks();
                          cout<<endl;
                          system("pause");
                          break;
                    case '4':
                          cout<<endl;
                          s.l.apply_leave(1);
                          cout<<endl;
                          system("pause");
                      break;
                    case '5':
                          cout<<endl;
                          s.l.apply_leave();
                          cout<<endl;
                          system("pause");
                          break;
                    case '6':
                          cout<<endl;
                          s.print_attendence();
                          cout<<endl;
                          system("pause");
                          break;
                    case '7':
                      break;
                default:
                    //cout<<endl;
                    //cout<<"You have entered Invalid option";
                    //cout<<endl;
                    //system("pause");
                    break;
                }
  }
    temp->x=s;
                ofstream o("student.dat",ios::out|ios::binary);
                //o.seekp(pos);
                temp=top;
    while(temp)
    {
        s=temp->x;
                        o.write((char*)&s,sizeof(s));
                        temp=temp->next;
                }
                o.close();
}




//All faculty related functions and classes

//class faculty contains all faculty info
class faculty
{
        public:
        char fac_no[20];
        char sub_code[10];
        char name[50];
        char password[20];
        //student s[100];



        void get_faculty()
        {
                cout<<"reg no:"<<endl;
                cin>>fac_no;
                cout<<"subject code"<<endl;
                cin>>sub_code;
                cout<<"name"<<endl;
                cin.ignore();
                cin.getline(name,50);
                cout<<"password"<<endl;
                cin>>password;


        }

        void print_faculty()
        {

                cout<<"Faculty number : "<<fac_no<<endl;
                cout<<"Subject code : "<<sub_code<<endl;
          cout<<"Faculty name : "<<name<<endl;
    }

    void view_marks()
    {
        student s[s_size];
        extract_to(s);
        for(int i=0;i<s_size;i++)
            {
                cout<<"Reg no:"<<s[i].reg<<'\t';
                s[i].m.print_marks(sub_code);
            }
    }
    void enter_marks()
    {
        student s[s_size];
        extract_to(s);

        char reg_no[20]="0";
        while(strcmp(reg_no,"-1")!=0)
        {

            view_marks();
            cout<<endl<<"Enter the Reg no of the student to enter marks or -1 to quit:
";

            cin>>reg_no;
            for(int i=0;i<s_size;i++)
            {
                if(strcmp(reg_no,s[i].reg)==0)
                {
                    s[i].m.enter_marks(sub_code);
                }
            }
            ofstream o("student.dat",ios::out|ios::binary);
            int i;
            for(i=0;i<s_size;i++)
            {
                o.write((char*)&s[i],sizeof(s[i]));
            }
        }

    }
    //void check_reval()//mark correction
};

struct nodef
{
    faculty y;
    nodef *next;
}*topf=NULL;

nodef* search(nodef *tempf,char r[])
{
    if(tempf==NULL)
        return NULL;
    if(strcmp(tempf->y.fac_no,r)==0)
        return tempf;
    search(tempf->next,r);
}
//the faculty login page(asks for password and shows the faculty menu)
void f_login()
{
    /*ifstream in("faculty.dat",ios::in|ios::binary);
    faculty f1;
    in.seekg(ios::beg);
    for(int i=0;i<f_size;i++)
    {
        in.read((char*)&f1,sizeof(f1));
        f1.print_faculty();
    }
    in.close();
    system("pause");*/


    char p[20],r[20];
    faculty f;
    cout<<"Faculty no:"; cin>>r;
    nodef *tempf;
    ifstream in("faculty.dat",ios::in|ios::binary);
    in.seekg(ios::beg);
    int i;
    for(i=0;i<f_size;i++)
    {
        tempf=new nodef;
        in.read((char*)&f,sizeof(f));
        tempf->y=f;
        tempf->next=topf;
        topf=tempf;
            }
    in.close();

    tempf=search(tempf,r);
    f=tempf->y;
    cout<<"Enter Password:";cin>>p;
    if(strcmp(p,f.password)!=0)
    {
        cout<<"Wrong Reg no/Password combination!!"<<endl;
        system("pause");
        return;
    }

        char ch='0';
        while(ch!='4')
        {
        system("cls");
        cout<<"\n\t\t\t\tFACULTY LOGIN\n"<<endl;
        cout<<"Faculty no: "<<f.fac_no<<endl<<endl;
        cout<<"1-View profile\n2-View Students Marks\n3-Enter student marks\n4-
Return to main menu\n\nEnter your choice : ";
        cin>>ch;
  switch(ch)
        {
            case '1':
```

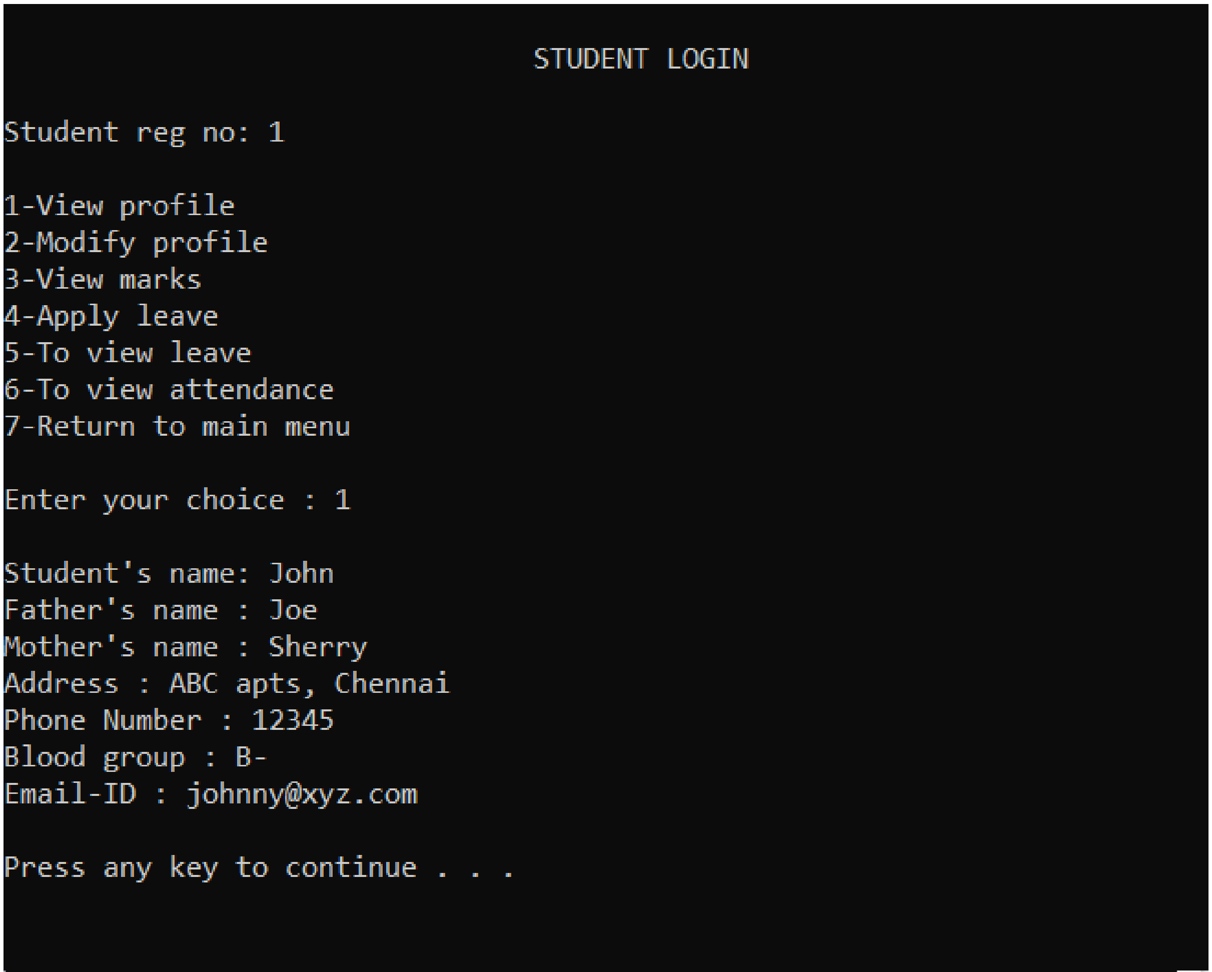
```
                                cout<<endl;
                                f.print_faculty();
                                cout<<endl;
                                system("pause");
                                break;
                        case '2':
                                cout<<endl;
                                f.view_marks();
                                cout<<endl;
                                system("pause");
                                break;
                        case '3':
                                cout<<endl;
                                f.enter_marks();
                                cout<<endl;
                                //system("pause");
                                break;
                        case '4':
                          break;
                  default: ;
                        //cout<<endl;
                        //cout<<"You have entered Invalid option";
                        //cout<<endl;
                                //system("pause");

                        }
     }
     /*ofstream o("faculty.dat",ios::out|ios::binary);
                o.seekp(pos);
                o.write((char*)&f,sizeof(f));
                o.close();*/
                tempf->y=f;
                ofstream o("faculty.dat",ios::out|ios::binary);
                tempf=topf;
                while(tempf)
                {
                        f=tempf->y;
                        o.write((char*)&f,sizeof(f));
                        tempf=tempf->next;
                }
                o.close();

}


//All warden related functions and classes
//class warden contains all warden info
class warden
{



        //student s[100];

        public:
        char ward_no[20];
        char name[20];
        char password[20];

        /*void extract_to(student& a[])
        {
        }*/
        void get_warden()
        {
                cout<<"Warden no:"; cin>>ward_no;
                cout<<"Name:";      cin>>name;
                cout<<"password:";  cin>>password;
        }

        void print_warden()
        {
                cout<<"Warden no: "<<ward_no<<endl;
                cout<<"Name: "<<name<<endl;
        }
        void mark_attendance()
        {
                struct q
                {
                        int pcount;
                        int acount;
                        q *next;
                }*f=NULL,*r=NULL;

                student s[s_size];
                extract_to(s);
                q *temp;

                //has the object of the students from the file
                //input to the student object is done here
                for(int i=0;i<s_size;i++)
                {
                        cout<<"Enter if Reg no: "<<s[i].reg<<" is present or absent (p/a): ";
                        char a;
                        temp=new q;
                        cin>>a;
                        if(a=='p')
                        {
                            temp->pcount=(++s[i].pcount);
                            temp->acount=s[i].acount;
                  }
                        else

                  {
                            temp->acount=(++s[i].acount);
                            temp->pcount=s[i].pcount;
                  }
                        temp->next=NULL;
                        if (r==NULL)
                            f=temp;
                        else
                            r->next=temp;
                        r=temp;
                }
                int i=0;
                q *temp1=f;
                ofstream o("student.dat",ios::out|ios::binary);
                o.seekp(ios::beg);
                while(temp1)
                {
                        s[i].pcount=temp1->pcount;
                        s[i].acount=temp1->acount;
                        o.write((char*)&s[i],sizeof(s[i]));
                        temp1=temp1->next;
                        ++i;
                }
                o.close();
        }
        void approve_leave()
        {
                student s[s_size];
                extract_to(s);
                char l;
                int flag=0;
                for(int i=0;i<s_size;i++)
                {
                        if(s[i].l.applied==true)
                        {
                            flag=1;
                            cout<<"Reg no : "<<s[i].reg<<" has applied leave."<<endl;
                            s[i].l.apply_leave();
                            cout<<"Do you want to approve the leave? (y/n) "<<endl;
                            cin>>l;
                            if(l=='y')
                            {
                                s[i].l.permission=true;
                                s[i].l.applied=false;
                            }
                            else
                            {
                                s[i].l.permission=false;
                                s[i].l.applied=false;
                            }

                        }
                }
                if(flag==0)
                {
                        cout<<endl<<"No leave applications yet"<<endl;
                }
                ofstream o("student.dat",ios::out|ios::binary);
                o.seekp(ios::beg);
                for(int i=0;i<s_size;i++)
                {
                o.write((char*)&s[i],sizeof(s[i]));

                }
                o.close();
        }
```

```
    void warden_search()
    {
        student s[s_size];
        extract_to(s);
        for(int i=0;i<s_size;i++)
        {
            cout<<"Reg no: "<<s[i].reg<<"\tName :"<<s[i].p.name<<endl;
        }
        cout<<endl<<"Enter the reg no of the student whose profile you want to view: ";
        char temp_reg[20];
        cin>>temp_reg;
        cout<<endl;
        for(int i=0;i<s_size;i++)
        {
            if(strcmp(s[i].reg,temp_reg)==0)
            {
                s[i].p.print_profile();
                cout<<"Room no: "<<s[i].room_no<<endl;
                s[i].print_attendence();
            }
        }

    }
};



struct nodew
{
    warden z;
    nodew *next;
}*topw=NULL;
//the warden login page(asks for password and shows the warden menu)
void w_login()
{
    char r[20];char p[20];
    warden w;
    cout<<"Warden no:"; cin>>r;
    nodew *tempw;
    ifstream in("warden.dat",ios::in|ios::binary);
    in.seekg(ios::beg);
    int i;
    for(i=0;i<w_size;i++)
    {
        tempw=new nodew;
        in.read((char*)&w,sizeof(w));
        tempw->z=w;
        tempw->next=topw;
        topw=tempw;
            }
    in.close();

    while(tempw)
    {
        if(strcmp(r,tempw->z.ward_no)==0)
        {
```

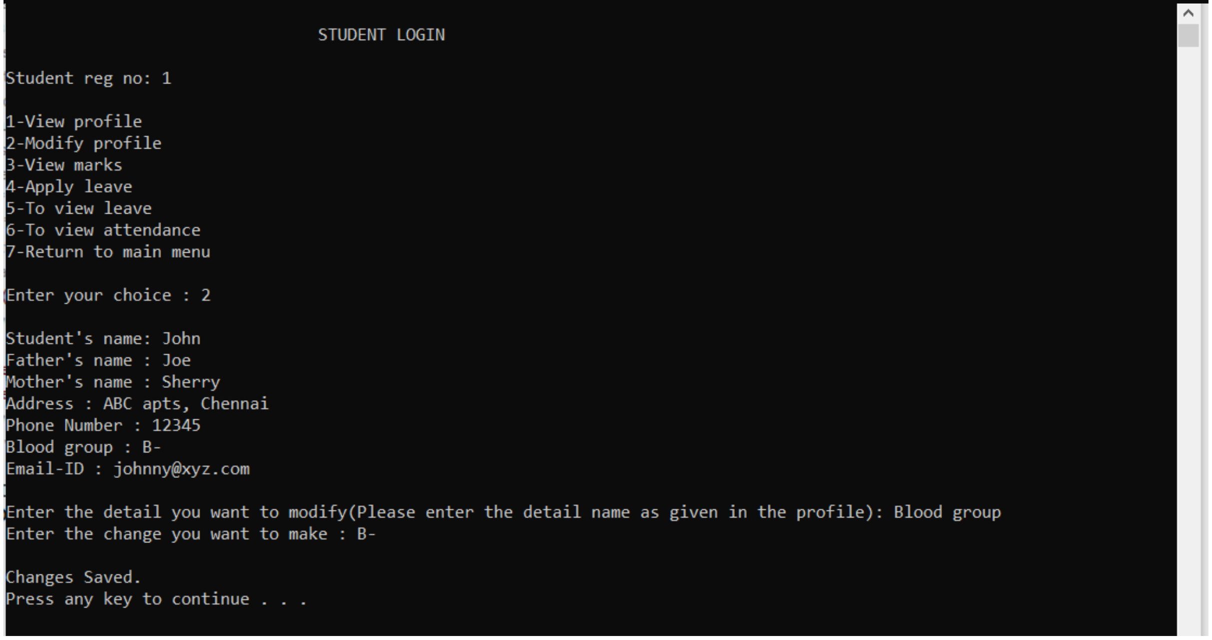
```
                        w=tempw->z;
                        break;
                }
                tempw=tempw->next;

        }
        cout<<"Enter Password:";cin>>p;
        if(strcmp(p,w.password)!=0)
        {
                cout<<"Wrong Reg no/Password combination!!"<<endl;
                system("pause");
                return;
        }

                char ch='0';

                while(ch!='4')
                {

                system("cls");
                cout<<"\n\t\t\t\tWARDEN LOGIN\n"<<endl;
                cout<<"Warden no: "<<w.ward_no<<endl<<endl;
                cout<<"1-Mark attendence\n2-Approve leave\n3-View student detail\n4-Return to
main menu\n\nEnter your choice : ";
                cin>>ch;
                switch(ch)
                {
                        case '1':
                                cout<<endl;
                                w.mark_attendance();
                                cout<<endl;
                                system("pause");
                                break;
                        case '2':
                                cout<<endl;
                                w.approve_leave();
                                cout<<endl;
                                system("pause");
                                break;
                        case '3':
                                cout<<endl;
                           w.warden_search();
                           cout<<endl;
                                system("pause");
                           break;
                        case '4':
                                break;
                   default: ;
                        //cout<<endl;
                        //cout<<"You have entered Invalid option";
                        //cout<<endl;
                                //system("pause");

                                }
    }
    ofstream o("warden.dat",ios::out|ios::binary);
```

```
                tempw->z=w;
                tempw=topw;
     while(tempw)
     {
        w=tempw->z;
                o.write((char*)&w,sizeof(w));
                tempw=tempw->next;
        }
                o.close();
}




//The main function shows initial menu to login as stud, fac or warden
int main()
{

        //To input student data to file

        /* ofstream o("student.dat",ios::out|ios::binary);
        int i;
        for(i=0;i<s_size;i++)
        {
                student s1;
                s1.get_student();
                o.write((char*)&s1,sizeof(s1));
        }
        o.close(); */

        /*       student ss[2];
                ifstream f("student.dat",ios::in|ios::binary);
                for(int i=0;i<2;++i)
                {
                        f.read((char*)&ss[i],sizeof(ss[i]));
                }
                f.close();
                ss[0].pcount=0;
                ss[0].acount=0;
                ss[1].pcount=0;
                ss[1].acount=0;
                ofstream o("student.dat",ios::out|ios::binary);
                for(int i=0;i<2;++i)
                {
                        o.write((char*)&ss[i],sizeof(ss[i]));
                }
                o.close(); */
        //to view contents of student file

        /* student s;
        ifstream in("student.dat",ios::in|ios::binary);
        in.seekg(ios::beg);
        for(int i=0;i<s_size;i++)
        {
                in.read((char*)&s,sizeof(s));
                cout<<s.reg<<endl;
                cout<<s.password<<endl;
                s.p.print_profile();
        }
        in.close();
        system("pause"); */


        //To input faculty data to file
        /* ofstream o("faculty.dat",ios::out|ios::binary);
        for(int i=0;i<f_size;i++)
        {
                faculty f;
                f.get_faculty();
                o.write((char*)&f,sizeof(f));
        }
        o.close(); */


        //To view contents of faculty file

        /* fstream in("faculty.dat",ios::in|ios::binary);
        faculty f;
        in.seekg(ios::beg);
        for(int i=0;i<f_size;i++)
        {
                in.read((char*)&f,sizeof(f));
                f.print_faculty();
        }
        in.close();
        system("pause"); */

        //To input warden data to file

        /* ofstream o("warden.dat",ios::out|ios::binary);
                warden w;
                w.get_warden();
                o.write((char*)&w,sizeof(w));
        o.close(); */


        //To view contents of warden file

        /* ifstream in("warden.dat",ios::in|ios::binary);
        warden w;
        in.seekg(ios::beg);
                in.read((char*)&w,sizeof(w));
                w.print_warden();
        in.close(); */


        char choice='0';
        while(choice!=4)
        {
                system("cls");
                cout<<"\n\t\t\t\t\t\tWELCOME\n\n"<<endl;
                cout<<"Enter 1 for Student , 2 for Faculty, 3 for Warden and 4 to exit"<<endl;
                cin>>choice;
                switch(choice)
                {
                    case '1':
                        system("cls");
                        cout<<"\n\t\t\t\tSTUDENT LOGIN\n"<<endl;
                        s_login();
                        break;
                    case '2':
                        system("cls");
                        cout<<"\n\t\t\t\tFACULTY LOGIN\n"<<endl;
                        f_login();
                        system("cls");
                        break;
                    case '3':
                        system("cls");
                        cout<<"\n\t\t\t\tWARDEN LOGIN\n"<<endl;
                        w_login();
                        system("cls");
                        break;
                    case '4': exit(0);
                        break;
                    default: ;
                    //  cout<<"You have entered an invalid option."<<endl;
                    //  system("pause");
                }
            }
        }
```

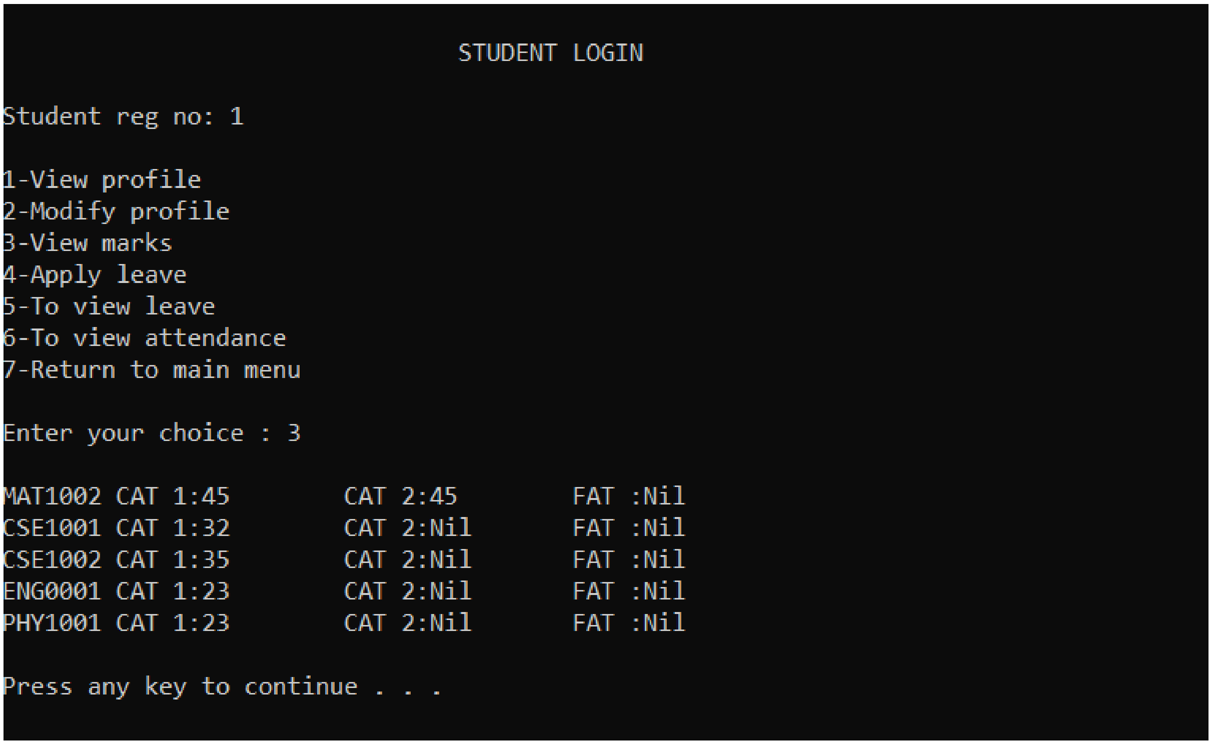
Output:



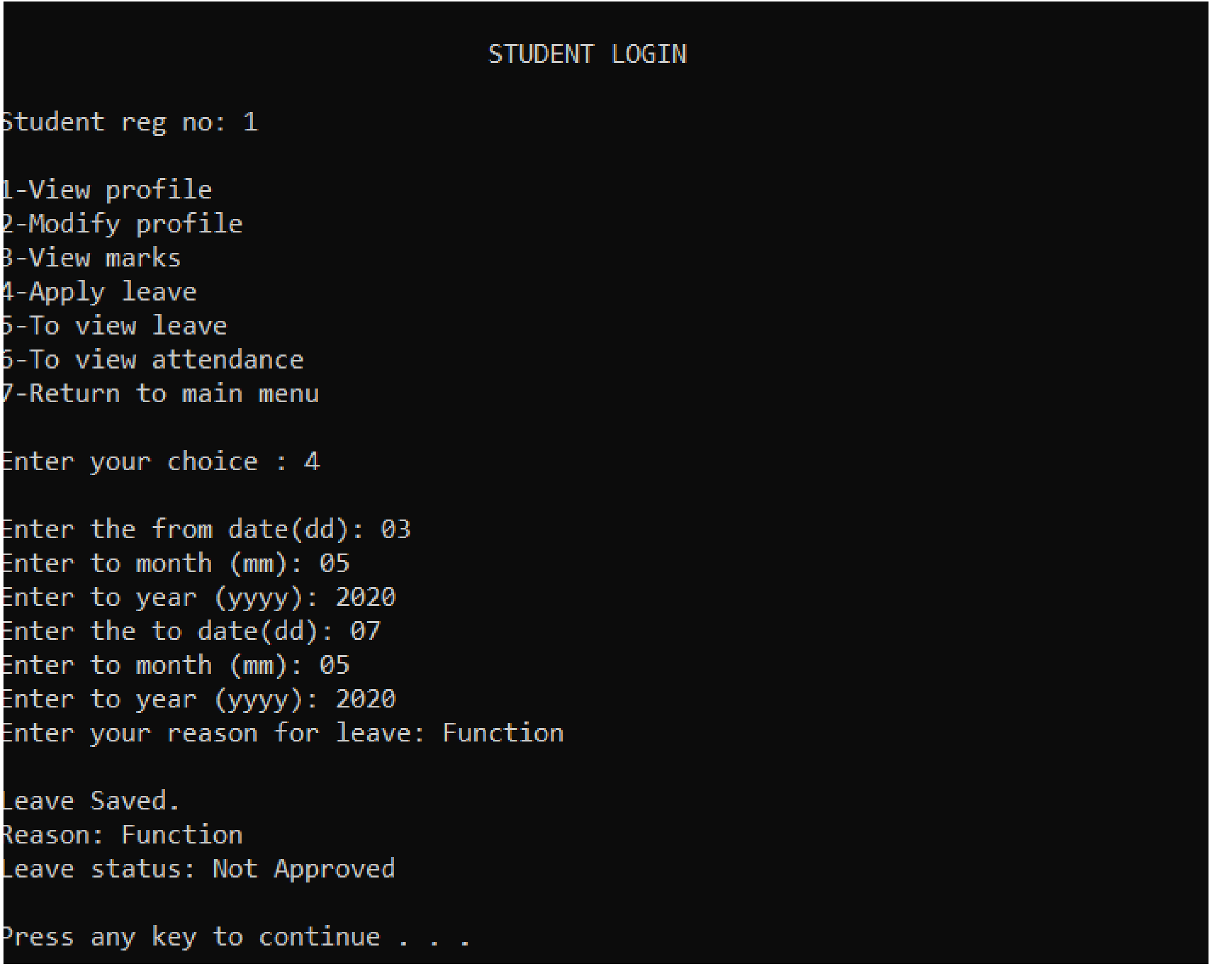
```
                                STUDENT LOGIN

Student reg no: 1

1-View profile
2-Modify profile
3-View marks
4-Apply leave
5-To view leave
6-To view attendance
7-Return to main menu

Enter your choice : 1

Student's name: John
Father's name : Joe
Mother's name : Sherry
Address : ABC apts, Chennai
Phone Number : 12345
Blood group : B-
Email-ID : johnny@xyz.com

Press any key to continue . . .
```



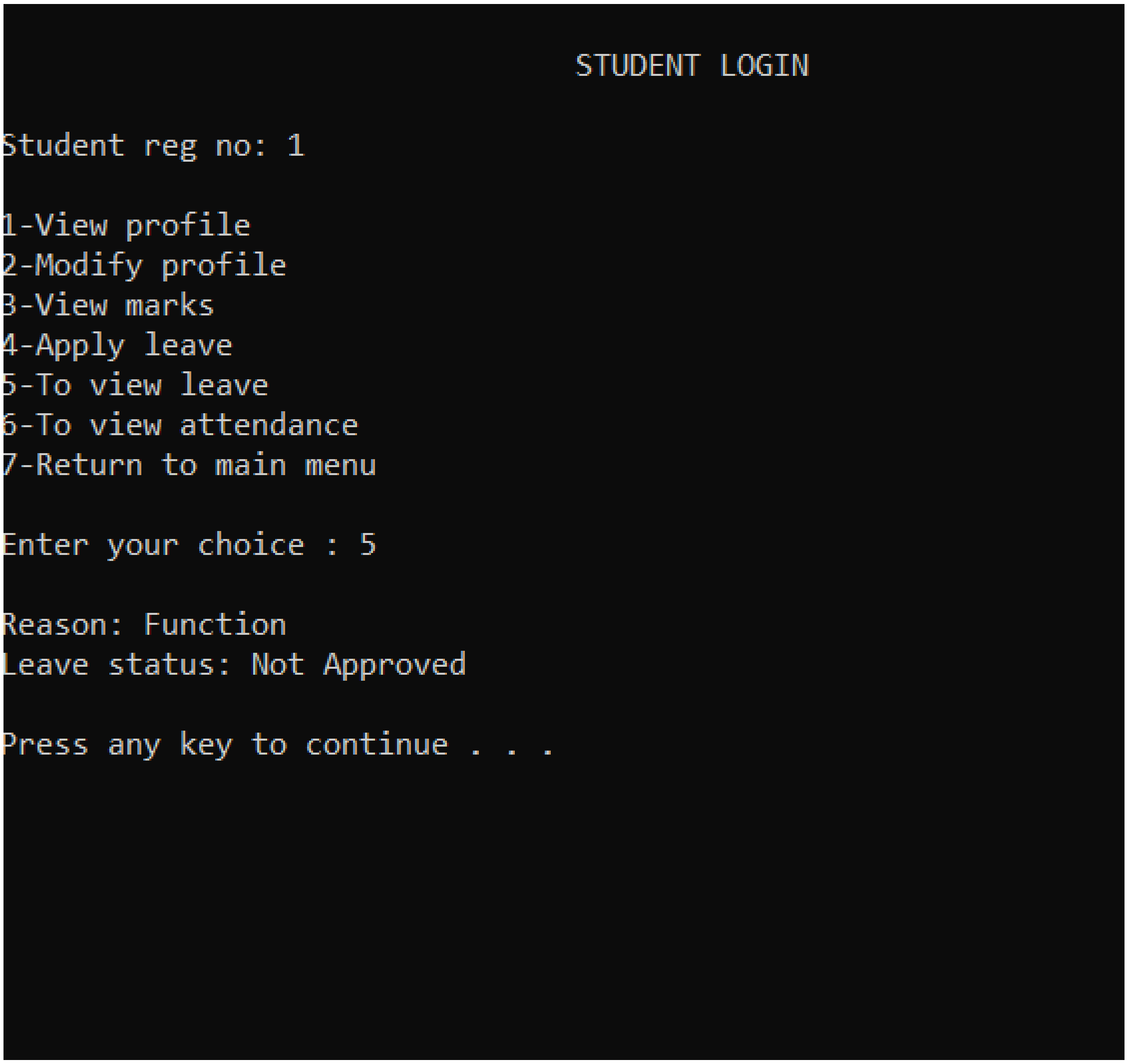
```
                                STUDENT LOGIN

Student reg no: 1

1-View profile
2-Modify profile
3-View marks
4-Apply leave
5-To view leave
6-To view attendance
7-Return to main menu

Enter your choice : 2

Student's name: John
Father's name : Joe
Mother's name : Sherry
Address : ABC apts, Chennai
Phone Number : 12345
Blood group : B-
Email-ID : johnny@xyz.com

Enter the detail you want to modify(Please enter the detail name as given in the profile): Blood group
Enter the change you want to make : B-

Changes Saved.
Press any key to continue . . .
```



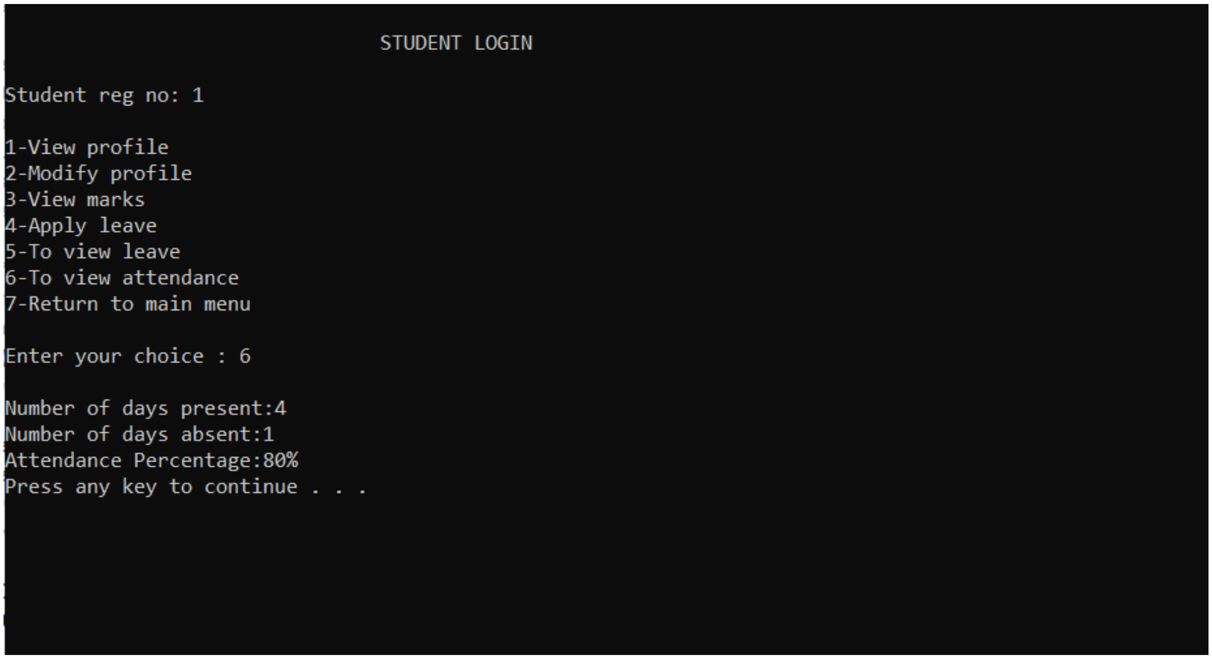
```
                                STUDENT LOGIN

Student reg no: 1

1-View profile
2-Modify profile
3-View marks
4-Apply leave
5-To view leave
6-To view attendance
7-Return to main menu

Enter your choice : 3

MAT1002 CAT 1:45        CAT 2:45        FAT :Nil
CSE1001 CAT 1:32        CAT 2:Nil       FAT :Nil
CSE1002 CAT 1:35        CAT 2:Nil       FAT :Nil
ENG0001 CAT 1:23        CAT 2:Nil       FAT :Nil
PHY1001 CAT 1:23        CAT 2:Nil       FAT :Nil

Press any key to continue . . .
```



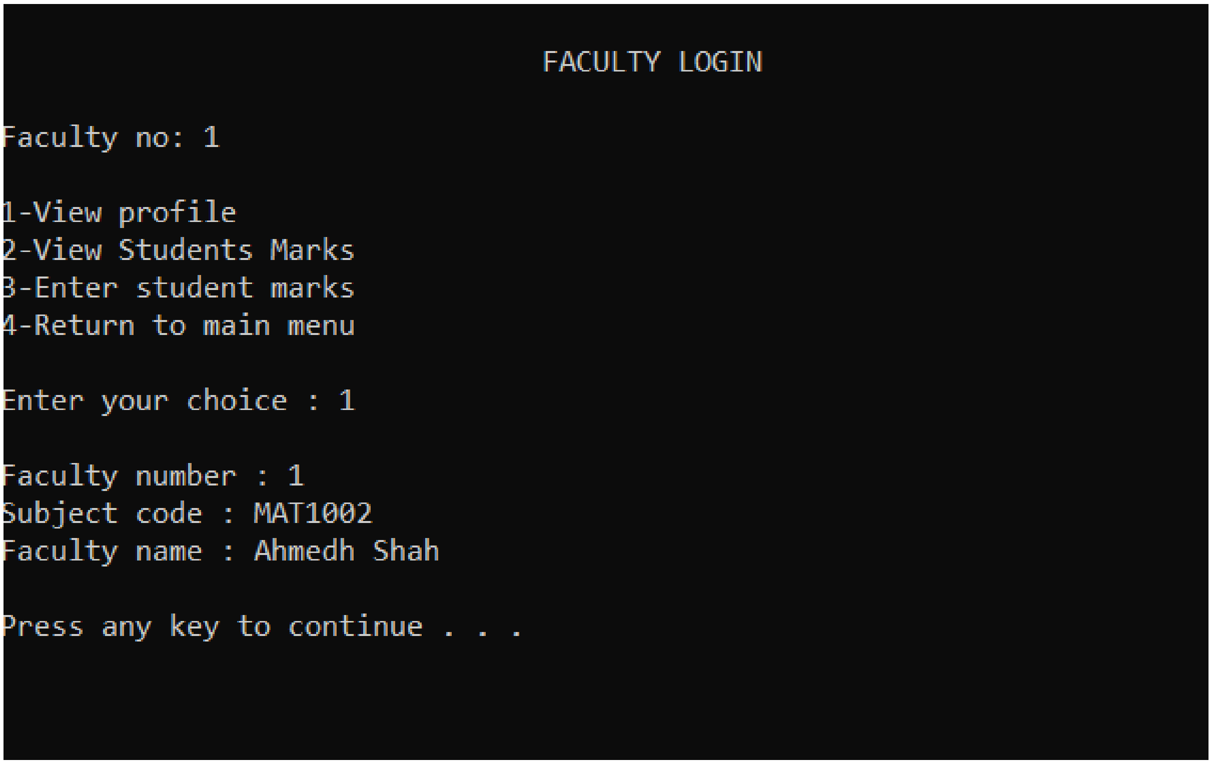
```
                                STUDENT LOGIN

Student reg no: 1

1-View profile
2-Modify profile
3-View marks
4-Apply leave
5-To view leave
6-To view attendance
7-Return to main menu

Enter your choice : 4

Enter the from date(dd): 03
Enter to month (mm): 05
Enter to year (yyyy): 2020
Enter the to date(dd): 07
Enter to month (mm): 05
Enter to year (yyyy): 2020
Enter your reason for leave: Function

Leave Saved.
Reason: Function
Leave status: Not Approved

Press any key to continue . . .
```



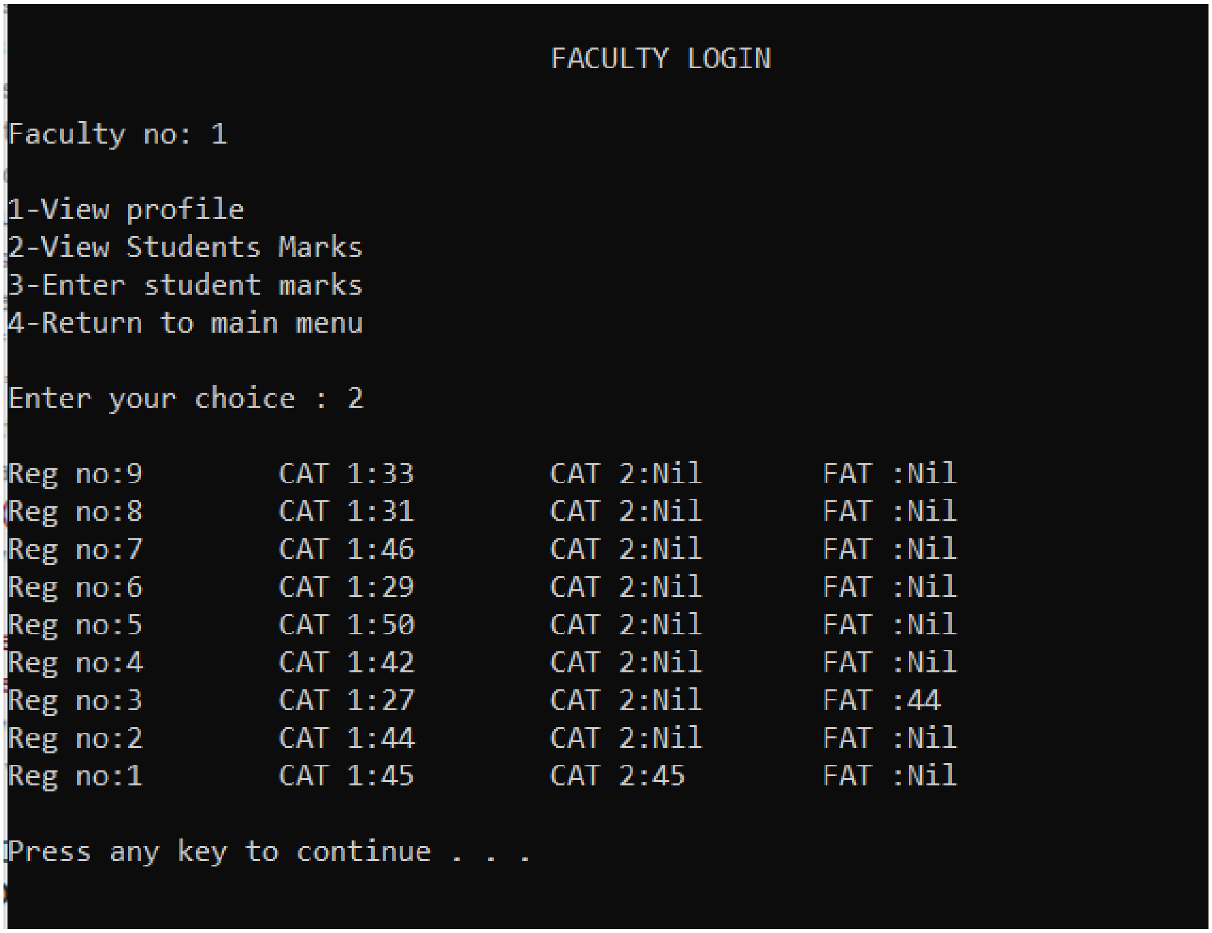
```
                              STUDENT LOGIN

Student reg no: 1

1-View profile
2-Modify profile
3-View marks
4-Apply leave
5-To view leave
6-To view attendance
7-Return to main menu

Enter your choice : 5

Reason: Function
Leave status: Not Approved

Press any key to continue . . .
```



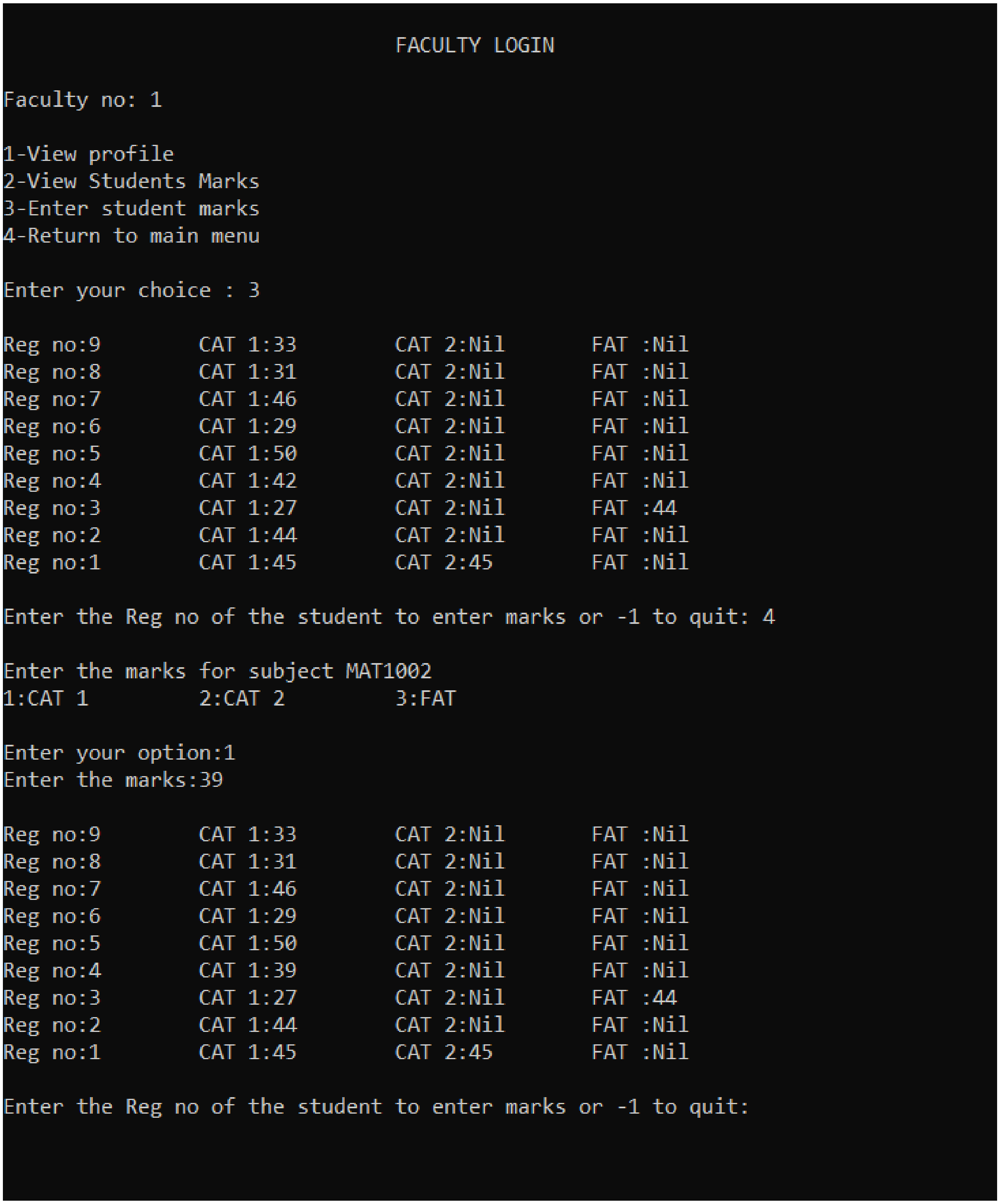
```
                                STUDENT LOGIN

Student reg no: 1

1-View profile
2-Modify profile
3-View marks
4-Apply leave
5-To view leave
6-To view attendance
7-Return to main menu

Enter your choice : 6

Number of days present:4
Number of days absent:1
Attendance Percentage:80%
Press any key to continue . . .
```



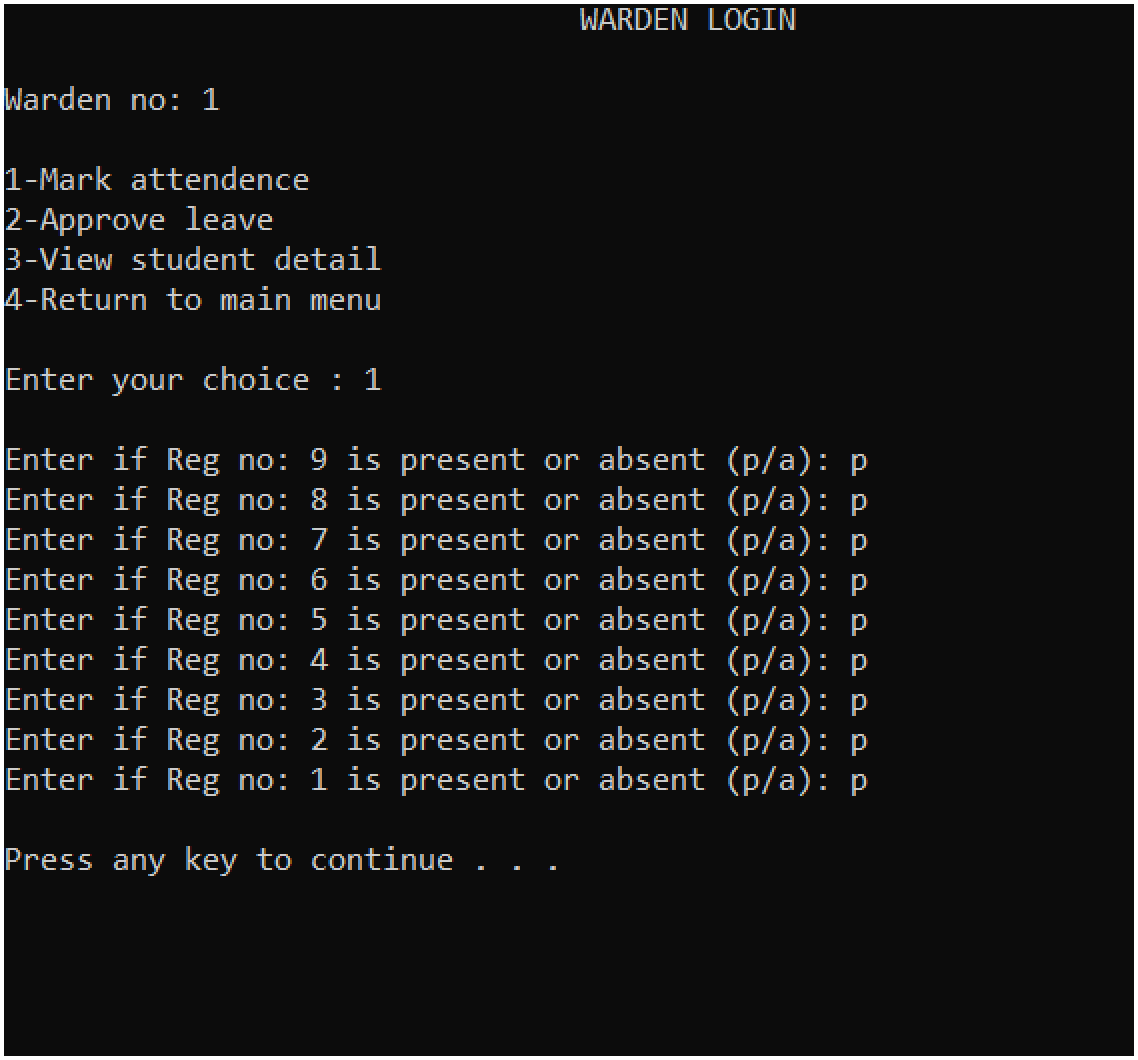
```
                                FACULTY LOGIN

Faculty no: 1

1-View profile
2-View Students Marks
3-Enter student marks
4-Return to main menu

Enter your choice : 1

Faculty number : 1
Subject code : MAT1002
Faculty name : Ahmedh Shah

Press any key to continue . . .
```



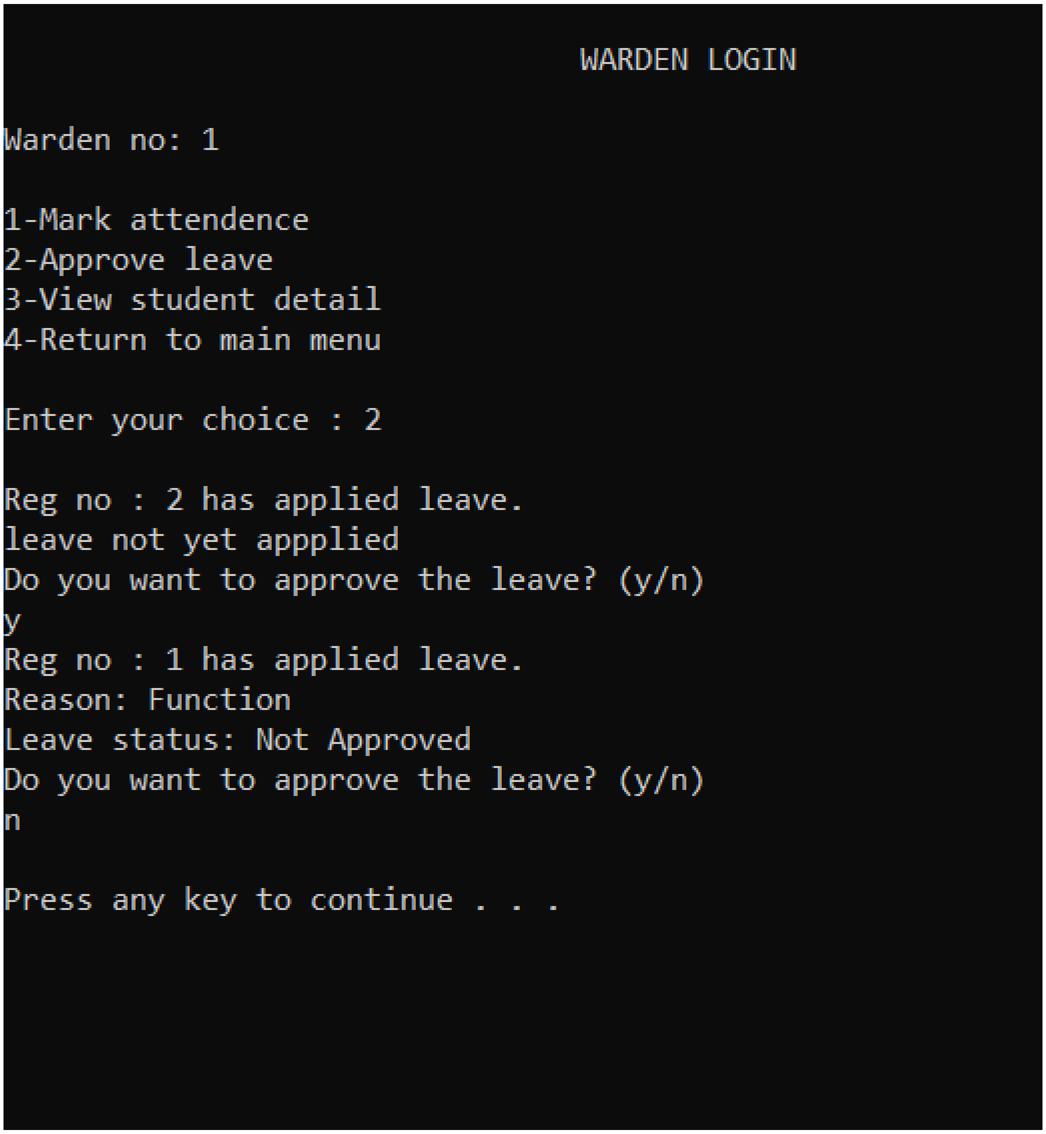
```
                                FACULTY LOGIN

Faculty no: 1

1-View profile
2-View Students Marks
3-Enter student marks
4-Return to main menu

Enter your choice : 2

Reg no:9        CAT 1:33        CAT 2:Nil       FAT :Nil
Reg no:8        CAT 1:31        CAT 2:Nil       FAT :Nil
Reg no:7        CAT 1:46        CAT 2:Nil       FAT :Nil
Reg no:6        CAT 1:29        CAT 2:Nil       FAT :Nil
Reg no:5        CAT 1:50        CAT 2:Nil       FAT :Nil
Reg no:4        CAT 1:42        CAT 2:Nil       FAT :Nil
Reg no:3        CAT 1:27        CAT 2:Nil       FAT :44
Reg no:2        CAT 1:44        CAT 2:Nil       FAT :Nil
Reg no:1        CAT 1:45        CAT 2:45        FAT :Nil

Press any key to continue . . .
```



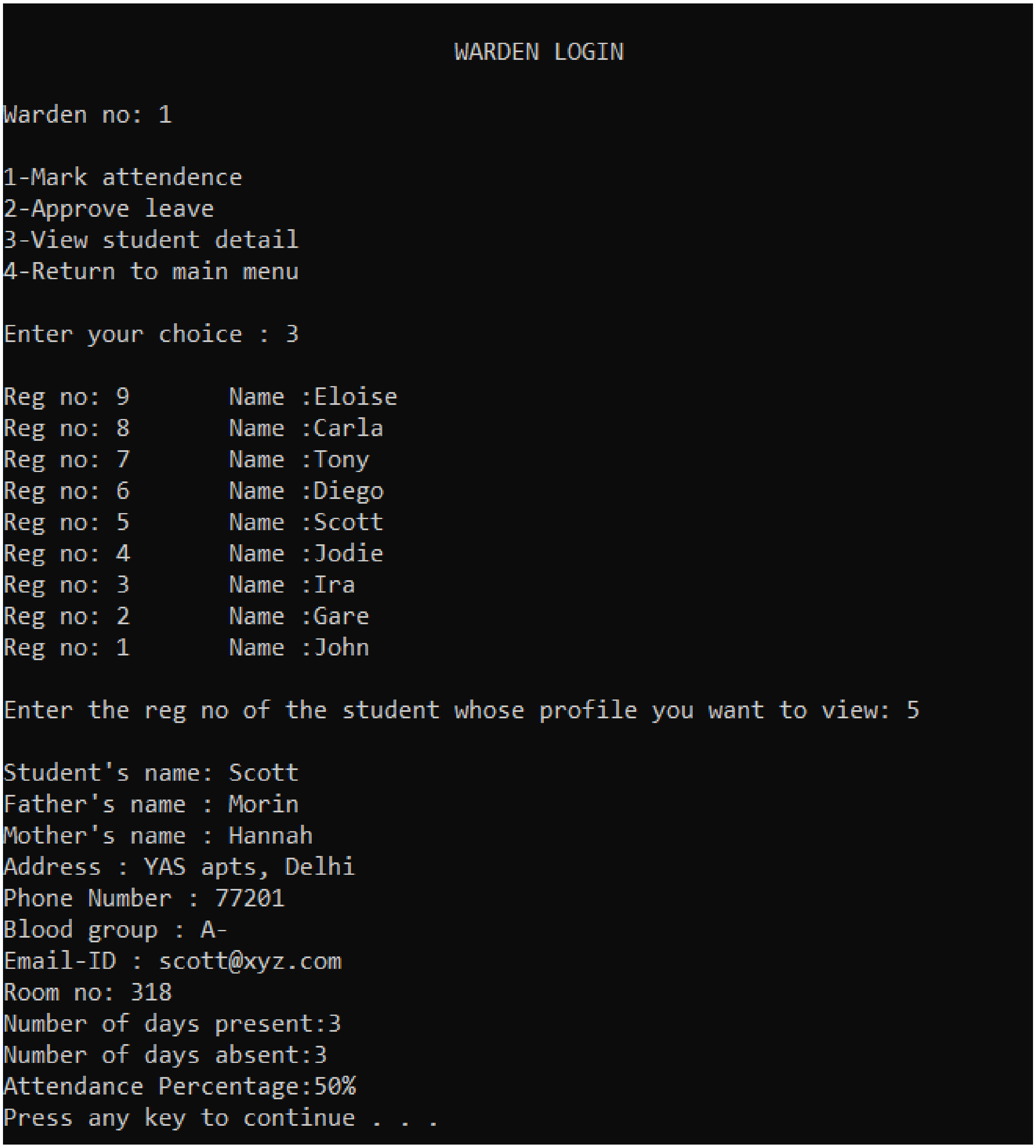
```
                                FACULTY LOGIN

Faculty no: 1

1-View profile
2-View Students Marks
3-Enter student marks
4-Return to main menu

Enter your choice : 3

Reg no:9        CAT 1:33        CAT 2:Nil       FAT :Nil
Reg no:8        CAT 1:31        CAT 2:Nil       FAT :Nil
Reg no:7        CAT 1:46        CAT 2:Nil       FAT :Nil
Reg no:6        CAT 1:29        CAT 2:Nil       FAT :Nil
Reg no:5        CAT 1:50        CAT 2:Nil       FAT :Nil
Reg no:4        CAT 1:42        CAT 2:Nil       FAT :Nil
Reg no:3        CAT 1:27        CAT 2:Nil       FAT :44
Reg no:2        CAT 1:44        CAT 2:Nil       FAT :Nil
Reg no:1        CAT 1:45        CAT 2:45        FAT :Nil

Enter the Reg no of the student to enter marks or -1 to quit: 4

Enter the marks for subject MAT1002
1:CAT 1         2:CAT 2         3:FAT

Enter your option:1
Enter the marks:39

Reg no:9        CAT 1:33        CAT 2:Nil       FAT :Nil
Reg no:8        CAT 1:31        CAT 2:Nil       FAT :Nil
Reg no:7        CAT 1:46        CAT 2:Nil       FAT :Nil
Reg no:6        CAT 1:29        CAT 2:Nil       FAT :Nil
Reg no:5        CAT 1:50        CAT 2:Nil       FAT :Nil
Reg no:4        CAT 1:39        CAT 2:Nil       FAT :Nil
Reg no:3        CAT 1:27        CAT 2:Nil       FAT :44
Reg no:2        CAT 1:44        CAT 2:Nil       FAT :Nil
Reg no:1        CAT 1:45        CAT 2:45        FAT :Nil

Enter the Reg no of the student to enter marks or -1 to quit:
```



```
                                WARDEN LOGIN

Warden no: 1

1-Mark attendence
2-Approve leave
3-View student detail
4-Return to main menu

Enter your choice : 1

Enter if Reg no: 9 is present or absent (p/a): p
Enter if Reg no: 8 is present or absent (p/a): p
Enter if Reg no: 7 is present or absent (p/a): p
Enter if Reg no: 6 is present or absent (p/a): p
Enter if Reg no: 5 is present or absent (p/a): p
Enter if Reg no: 4 is present or absent (p/a): p
Enter if Reg no: 3 is present or absent (p/a): p
Enter if Reg no: 2 is present or absent (p/a): p
Enter if Reg no: 1 is present or absent (p/a): p

Press any key to continue . . .
```



```
                                WARDEN LOGIN

Warden no: 1

1-Mark attendence
2-Approve leave
3-View student detail
4-Return to main menu

Enter your choice : 2

Reg no : 2 has applied leave.
leave not yet appplied
Do you want to approve the leave? (y/n)
y
Reg no : 1 has applied leave.
Reason: Function
Leave status: Not Approved
Do you want to approve the leave? (y/n)
n

Press any key to continue . . .
```



```
                                WARDEN LOGIN

Warden no: 1

1-Mark attendence
2-Approve leave
3-View student detail
4-Return to main menu

Enter your choice : 3

Reg no: 9       Name :Eloise
Reg no: 8       Name :Carla
Reg no: 7       Name :Tony
Reg no: 6       Name :Diego
Reg no: 5       Name :Scott
Reg no: 4       Name :Jodie
Reg no: 3       Name :Ira
Reg no: 2       Name :Gare
Reg no: 1       Name :John

Enter the reg no of the student whose profile you want to view: 5

Student's name: Scott
Father's name : Morin
Mother's name : Hannah
Address : YAS apts, Delhi
Phone Number : 77201
Blood group : A-
Email-ID : scott@xyz.com
Room no: 318
Number of days present:3
Number of days absent:3
Attendance Percentage:50%
Press any key to continue . . .
```

CONCLUSION AND FUTURE ENHANCEMENT:

1. With more knowledge of several coding domains we can go on further and actually made this project available on a proper website and develop an app for effective interface.

2.More features can be added to this system as per the feedback of student, faculties and wardens.

3.Better searching algorithms can be used to make this system fast and responsive

REFERENCES:

1. http://ijarcsse.com/Before_August_2017/docs/papers/Special_Issue/NCRTIT2K16/NCRTIT_12.pdf

2. https://www.geeksforgeeks.org/student-data-management-c

3. http://codingsiksha.blogspot.com/2015/06/student-information-system-mini-project.html

4. https://projectsgeek.com/2014/08/student-management-system-project-c.html