

```
/*Headers included in the project */
#include<iostream>
#include<string.h>
#include<stdlib.h>
#include<conio.h>
#include<iomanip>
#include<fstream>
using namespace std;

//GLOBAL VARIABLES
char ch;
fstream fin,fout,fio;

// prototype declaration of various functions
void main_menu();
void add_user();
void users_menu();
void del_user();
int verify_password();
void students_menu();
void teachers_menu();
void fee_menu();
void fee_add();
void fee_Modify();
void disp_stu_fee();
void fee_status();
void stu_search();
void stu_add();
void stu_displayall();
void Classwise();
void stu_Modify();
void stu_Delete();
void marks_entry();
void display_stu_marks();
void marks_menu();
void t_search();
void t_add();
void t_Modify();
void t_Delete();
void t_displayall();
```

```

//for security management
class Users
{
    char user[10],password[10];
public:
    int admin()
    {
        int ch;
        assign_user();
        if(strcmp(user,"KRITIKA")==0)
        {
            if(strcmp(password,"123")==0);
            ch=1;
        }
        return(ch);
    }
    void assign_user()
    {
        cout<<"\n\tEnter username : ";
        cin.get(ch);
        cin.getline(user,10);
        cout<<"\n\tEnter password : ";
        cin.get(ch);
        cin.getline(password,10);
    }
    /*accessor functions-*/
    char* getuser()
    {
        return user;
    }
    char* getpass()
    {
        return password;
    }
}U;    // object of class 'Users' is created as 'U'

void add_user()
{
    char ch='y';

```

```

    fin.open("USERS.dat",ios::out|ios::app);
    if(!fin)
    {
        cout<<"\nFile could not be opened !";
        exit(-1);
    }
    while(ch=='y' || ch=='Y')
    {
        U.assign_user();
        fin.write((char*)&U,sizeof(U));
        cout<<"\nData Appended and Transferred to
        file.\n";
        cout<<"\nDo you want to add more : ";
        cin>>ch;
    }
    fin.close();
    getch();
    system("cls");
    users_menu();
}
void del_user()
{
    char x[10];
    char found='n';
    char confirm='y';
    fin.open("USERS",ios::in);
    ofstream fout("temp.dat",ios::out);
    cout<<"\nEnter the Username to be deleted: ";
    cin.get(ch);
    gets(x);
    if(!fin)
    {
        cout<<"\nFile could not be opened !";
        exit(-1);
    }
    while(fin.read((char*)&U,sizeof(U)))
    {
        if(strcmp(U.getuser,x)==0)
        {
            found='y';

```

```

        cout<<"\nAre you sure you want to delete
this record(y/n) :";
        if(confirm=='n')
        {
            fout.write((char*)&U,sizeof(U));
        }
    }
    else
        fout.write((char*)&U,sizeof(U));
}
if(found=='n')
    cout<<"\nUser "<<x<<" not found.\n";
else
    cout<<"\nUser"<<x<<"is now invalidated";
fin.close();
fout.close();
remove("USERS");
rename("temp.dat","USERS");
getch();
system("cls");
users_menu();
}
int verify_password()
{
    fin.open("USERS",ios::in);
    char usernm[10],pwd[10];
    int found=0;
    cout<<"\n\n Enter user name : ";
    cin>>usernm;
    if(!fin)
    {
        cout<<"\nFile could not be opened !";
        exit(-1);
    }
    while(fin.read((char*)&U,sizeof(U)))
    {
        if(strcmp(usernm,U.getuser())==0)
        {
            cout<<"\n\n Enter Password : ";
            cin>>pwd;

```

```

        if(strcmp(pwd,U.getpass())==0)
            found=1;
    }
}
    fin.close();
    return found;
}

void users_menu()
{
    int x;
    cout<<"\n\t-----";
    cout<<"\n\t\tUsers Menu\n";
    cout<<"\n\t-----";
    cout<<"\n\t\t1. Assign new Users ";
    cout<<"\n\t\t2. Invalidate a User ";
    cout<<"\n\t\t3. Back to main menu";
    cout<<"\n\t-----";
    cout<<"\n\tEnter Your Choice (1-3) : ";
    cin>>x;
    while(x)
    {
        switch(x)
        {
case 1: system("cls");
            cout<<"\n\tNOTE : Remember ,only admin has
rights to add/invalidate users,\n\t so enter
only admin's username and password !\n ";
            cout<<"\t-----\n";
            if(U.admin()==1)
            {
                cout<<"\n\t\tACCESS GRANTED\n\tYou are now
allowed to add new users !\n ";
                add_user();
            }
            else
                cout<<"\n\tThe username or password you
entered is wrong,\n\t ACCESS DENIED !";
            cout<<"\n\t-----";
            getch();

```

```

        main_menu();
        break;
case 2:  system("cls");
        cout<<"\n\tNOTE : Remember ,only admin has
rights to add/invalidate users,\n\t so enter
only admin's username and password !\n ";
        cout<<"\t-----\n";
        if(U.admin()==1)
        {
            cout<<"\n\t\tACCESS GRANTED \n\tYou are
now allowed to invalid users !\n ";
            del_user();
        }
        else
            cout<<"\n\tThe username or password you
entered is wrong,\n\t ACCESS DENIED !";
        cout<<"\n\t-----";
        getch();
        main_menu();
        break;
case 3:  main_menu();
        break;
default: cout<<"Wrong choice";
        cout<<" Try again !";
        system("cls");
        users_menu();
        break;
    }
}
};

```

//class Fee starts here

```
class Fee : virtual protected Users
```

```

{
    public:
    int t1,t2,t3,c,s1,s2,d1,d2,d3,a,a1,a2,a3;
    int regno,paid,fee;
    char session[10];
    int dd,mm,yy;
    Fee()                                //constructor

```

```

    {
        fee=0;
    }
void enter_fee_details();
void print_fee_details();
void enter_status(int x)
{
    regno=x;
    cout<<"\n Enter amount paid :Rs.";
    cin>>paid;
    fee+=paid;
    cout<<"\n Enter month of
    submission(integer value) : ";
    cin>>mm;
    cout<<"\n Enter date of submission(integer
    value) : ";
    cin>>dd;
    cout<<"\n Enter year of submission(integer
    value) : ";
    cin>>yy;
}
int ret_regno()
{
    return regno;
}
char fsearch(int regno)
{
    char found='n';
    if(regno==ret_regno())
        found='y';
    return(found);
}
void displayfee()
{
    cout<<"\n\t Total Amount Paid : "<< fee;
    cout<<"\n\t Amount paid recently : "<<paid;
    cout<<"\n\t Fee last submitted on : "
    <<dd<<"-"<<mm<<"-"<<yy;
}
}ff;    //object of class Fee is created as 'ff'.

```

```

//class Marks starts here
class Marks :virtual protected Users
{
    public:
    int Id;
    float marks[10];
    float total_marks;    //gained by student
    float Max_marks;
    float per;            //percentage
    char subject[10][20];
    int x;                //no of subjects
    char grade;
    Marks()                //constructor
    {
        total_marks=0;
    };
    int ret_Id()
    {
        return Id;
    }
    char msearch(int regno)
    {
        char found='n';
        if(regno==ret_Id())
            found='y';
        return(found);
    }

    void get_marks(int rno)
    {
        Id=rno;
        cout<<"\n\tEnter number of subjects opted by the
        student : ";
        cin>>x;
        for(int i=0 ; i<x; i++)
        {
            total_marks=0;
            cout<<"\n\tEnter subject "<<i+1<<" name : ";

```



```

        cin.get(ch);          //to clear buffer
        cin.getline(subject[i],20);
        cout<<"\n\tEnter marks gained in "<<
        subject[i]<<" : ";
        cin>>marks[i];
        total_marks+=marks[i];
    }
    Max_marks=100*x;
}

void calculate()
{
    int y;
    y=(total_marks/Max_marks);
    per=y*100;
    if(per>=60)
        grade='A';
    else if(per>=50)
        grade='B';
    else if(per>=33)
        grade='C';
    else
        grade='F';
}

void disp_marks()
{
    calculate();
    cout<<"\n-----"
";
    cout<<"\n\t REPORT CARD ";
    cout<<"\n-----"
";
    cout<<"\n\tMarks in various subjects obtained by
student are as follows - ";
    for(int i=0;i<x;i++)
    {
        cout<<"\n\t"<<subject[i]<<" :
"<<marks[i]<<endl;
    }
}

```

```

        cout<<"\n\tTotal marks obtained:"
        <<total_marks<<"/"<<Max_marks;
        cout<<"\n\tPercentage : "<<per<<"%";
        cout<<"\n\tGrade\t: "<<grade<<"\n";
        cout<<"\n-----";
";
    }
}m;    //object of class Marks is created as 'm'.

// class student starts here

class Student : public Fee,public Marks
//hybrid inheritance
{
    int admno;
    char s_name[30];
    int std;           //standard
    char section;
    char dob[15];
    char father_name[30];
    char mother_name[30];
    char address[80];
    long long int contact;
    char aadhaar[15];
    char email[20];
    char religion[10];
    char nationality[10];
public:
    void stu_getdata();
    void stu_dis();
    void stu_mod();
    char search(int rno);
    // accessor functions -
    int get_admno() {return admno;}
    int get_Class() {return std;}
}s;    //object of class Student is created as 's'.

char Student::search(int rno)
{

```

```

        char found='n';
        if(s.get_admno()==rno)
            found='y';
        return(found);
    }

void Student::stu_getdata()
{
    cout<<"-----";
    cout<<"\nSchool management system - Enter Record";
    cout<<"\n-----";
    cout<<"\n\tEnter admission number of student : ";
    cin>>admno;
    cout<<"\n\tEnter name of student : ";
    cin.get(ch);        //to clear buffer
    cin.getline(s_name,30);
    cout<<"\n\tEnter the class(enter integers only, no
    roman numerals) : ";
    cin>>std;
    cout<<"\n \tEnter section(in single letters) : ";
    cin>>section;
    cin.get(ch);
    cout<<"\n\tEnter date of birth(dd/mm/yyyy) : ";
    cin.get(ch);        //to clear buffer
    cin.getline(dob,15);
    cout<<"\n\tEnter father's name : ";
    cin.getline(father_name,30);
    cout<<"\n\tEnter mother's name : ";
    cin.get(ch);
    cin.getline(mother_name,30);
    cout<<"\n\tEnter address : ";
    cin.getline(address,80);
    cout<<"\n\tEnter Mobile number : ";
    cin>>contact;
    cout<<"\n\tEnter Email Id : ";
    cin.get(ch);        //to clear buffer
    cin.getline(email,20);
    cout<<"\n\tEnter the religion : ";
    gets(religion);
    cout<<"\n\tEnter nationality : ";

```

```

        gets(nationality);
        cout<<"\n\tEnter aadhar card number : ";
        gets(aadhaar);

    }

void Student::stu_dis()
{
    cout<<"\n=====";
    cout<<"\n\t\tSTUDENT'S DETAILS";
    cout<<"\n=====";
    cout<<"\n\tAdmission no.   : "<<admno;
    cout<<"\n\tName           : "<<s_name;
    cout<<"\n\tClass          : "<<std;
    cout<<"\n\tSection         : "<<section;
    cout<<"\n\tDOB\t\t      : "<<dob;
    cout<<"\n\tFather's Name\t    : "<<father_name;
    cout<<"\n\tMother's Name \t   : "<<mother_name;
    cout<<"\n\tAddress          : "<<address;
    cout<<"\n\tMobile no       : "<<contact;
    cout<<"\n\tEmail Id   \t: "<<email;
    cout<<"\n\tAadhar Card no.   : "<<aadhaar;
    cout<<"\n\tNationality      : "<<nationality;
    cout<<"\n\tReligion        : "<<religion;
    cout<<"\n=====";
}

void Student::stu_mod()
{
    //temporary variables-
    int cl;
    char s[30],f[30],m[30],d[15],a[15],n[10];
    char r[10],e[20];
    char add[80],sec;
    long long int mob;
    cout<<"\n\tEnter the updated datails ";
    cout<<"\n\tName(press '$' for no changes) \t: ";
    cin.get(ch);
    gets(s);

```

```

cout<<"\n\tClass (press 0 for no changes, only
numerals)\t: ";
cin>>cl;
cout<<"\n\tSection (press '$' for no changes) \t: ";
cin>>section;
cout<<"\n\tMobile no (press 0 for no changes)\t: ";
cin>>mob;
cout<<"\n\tDOB (press '$' for no
changes, dd/mm/yyyy)\t: ";
cin.get(ch);
gets(dob);
    cout<<"\n\tFather's Name (press '$' for no
changes)\t: ";
gets(f);
cout<<"\n\tMother's Name (press '$' for no
changes)\t: ";
gets(m);
cout<<"\n\tAddress (press '$' for no changes)\t: ";
gets(add);
cout<<"\n\tEmail Id (press '$' for no changes)\t:";
cin.get(ch);
gets(e);
    cout<<"\n\tNationality (press '$' for no
changes)\t: ";
gets(n);
cout<<"\n\tReligion (press '$' for no changes)\t:";
gets(r);
cout<<"\n\tAadhar Card no. (press '$' for no
changes) \t: ";
gets(e);
char confirm;
cout<<"\nAre you sure you want to make these
changes (press 'y' for yes and 'n' for no):";
cin>>confirm;
if(confirm=='y' || 'Y')
{
    if(cl!=0)
        std=cl;
    if(mob!=0)
        contact=mob;
}

```

```

        if(sec!='$')
            section=sec;
        if(strcmp(s,"$")!=0)
            strcpy(s_name,s);
        if(strcmp(f,"$")!=0)
            strcpy(father_name,f);
        if(strcmp(m,"$")!=0)
            strcpy(mother_name,m);
        if(strcmp(d,"$")!=0)
            strcpy(dob,d);
        if(strcmp(a,"$")!=0)
            strcpy(aadhaar,a);
        if(strcmp(r,"$")!=0)
            strcpy(religion,r);
        if(strcmp(n,"$")!=0)
            strcpy(nationality,n);
        if(strcmp(e,"$")!=0)
            strcpy(email,e);
        if(strcmp(add,"$")!=0)
            strcpy(address,add);
    }
    else
        students_menu();
}
// end of class student

//For adding student's data into the file.
void stu_add()
{
    char ch='y';
    fout.open("STUDENTS.dat",ios::out|ios::app|
ios::binary);

    while(ch=='y' || ch=='Y')
    {
        s.stu_getdata();
        fout.write((char*)&s,sizeof(s));
        cout<<"\nData Appended and Transferred to
file.\n";
        cout<<"\nDo you want to add more : ";
    }
}

```

```

        cin>>ch;
    }
    fout.close();
    getch();
    students_menu();
}

//to search student's record
void stu_search()
{
    fin.open("STUDENTS.dat",ios::in);
    char found='n';
    int rno;
    if(!fin)
    {
        cout<<"\nFile could not be opened !";
        exit(-1);
    }
    cout<<"\nEnter the admission no. to be searched:";
    cin>>rno;
    while(fin.read((char*)&s,sizeof(s)))
    {
        if(s.search(rno)=='y')
        {
            found='y';
            s.stu_dis();
            break;
        }
    }
    if(found=='n')
    {
        cout<<"The record you are searching for,
        does not exists!";
    }
    fin.close();
    getch();
    students_menu();
}

//For displaying the record of students class wise
void Classwise()

```

```

{
    fin.open("STUDENTS.dat",ios::in);
    int std;
    char found='n';
    cout<<"\nEnter the class to be searched : ";
    cin>>std;
    if(!fin)
    {
        cout<<"\nFile could not be opened !";
        exit(-1);
    }
    while(fin.read((char*)&s,sizeof(s)))
    {
        if(s.get_Class()==std)
        {
            s.stu_dis();
            found='y';
        }
    }
    if(found=='n')
    {
        cout<<"\nThe Class "<<std<<" is not in the
        file.\n";
    };
    fin.close();
    getch();
    students_menu();
}

//to modify student's record
void stu_Modify()
{
    fin.open("STUDENTS.dat",ios::in|ios::out);
    char found='n';
    int rno;
    fin.seekg(0);
    cout<<"\nEnter the admission no. whose record is to
    be modified: ";
    cin>>rno;
    if(!fin)

```



```

    {
        cout<<"\nFile could not be opened !";
        exit(-1);
    }
while(fin.read((char*)&s,sizeof(s)))
{
    if(s.search(rno)=='y')
    {
        found='y';
        s.stu_mod();
        int pos = -1 * sizeof(s);
        fin.seekp(pos,ios::cur);
        fin.write((char*)&s,sizeof(s));
        cout<<"\nRecord of Admission no. "<<rno<<" has
        been updated.";
        cout<<"\nUpdated record is here -\n";
        s.stu_dis();
        break;
    }
}
if(found=='n')
{
    cout<<"\nThe record of admission no "<<rno<<"
    is not in the file.\n";
}
fin.close();
getch();
students_menu();
}

//For Displaying the all data of the students.
void stu_displayall()
{
    fin.open("STUDENTS.dat",ios::in);
    if(!fin)
    {
        cout<<"\nFile could not be opened !";
        exit(-1);
    }
while(fin.read((char*)&s,sizeof(s)))
{
    s.stu_dis();
}

```

```

    }
    fin.close();
    getch();
    students_menu();
}

//to delete a student's record
void stu_Delete()
{
    char found='n'; //initially the record is not
found
    char confirm='y'; //yes- initial confirmation
    int rno;
    fin.open("STUDENTS.dat",ios::in);
    if(!fin)
    {
        cout<<"\nFile could not be opened !";
        exit(-1);
    }
    fout.open("temp.dat",ios::out);
    cout<<"\nEnter the Admission no. whose record is to
be deleted : ";
    cin>>rno;
    while(fin.read((char*)&s,sizeof(s)))
    {
        if(s.get_admno()==rno)
        {
            s.stu_dis();
            found='y';
            cout<<"\nAre you sure you want to delete this
record(y/n) :";
            cin>>confirm;
            if(confirm=='n' || confirm=='N')
            {
                fout.write((char*)&s,sizeof(s));
            }
            else
            fout.write((char*)&s,sizeof(s));
        }
    }
}

```



```

        else
            cout<<"\nYOU ARE NOT AN AUTHORIZED
            USER !!\n YOU CAN'T ADD RECORDS.";
        getch();
        students_menu();
        break;
case 2: system("cls");
        stu_search();
        break;
case 3: system("cls");
        stu_displayall();
        break;
case 4: system("cls");
        Classwise();
        break;
case 5: if(verify_password()==1)
        {
            system("cls");
            stu_Modify();
        }
        else
            cout<<"\nYOU ARE NOT AN AUTHORIZED
            USER !!\n YOU CAN'T MODIFY RECORDS.";
        getch();
        students_menu();
        break;
case 6: if(verify_password()==1)
        {
            system("cls");
            stu_Delete();
        }
        else
            cout<<"\nYOU ARE NOT AN AUTHORIZED
            USER !!\n YOU CAN'T DELETE RECORDS.";
        getch();
        students_menu();
        break;
case 7: main_menu();
        break;

```

```

        default: cout<<"Wrong Input ! Please choose an
                  option from the list";
                  getch();
                  students_menu();
    }

    cin>>ch;
    system("cls");
}
}

void Fee::enter_fee_details()
{
    cout<<"\n-----";
    cout<<"\n\t Enter fee details for session : ";
    cin>>session;
    cout<<"\n-----";
    cout<<"\n\n Tution fee (monthly) for following
    groups - ";
    cout<<"\n      Classes I-V : Rs. ";
    cin>>t1;
    cout<<"\n      Classes VI-X : Rs.";
    cin>>t2;
    cout<<"\n      Classes XI-XII : Rs.";
    cin>>t3;
    cout<<"\n\n Computer fee (monthly) for following
    groups - ";
    cout<<"\n      Classes I-XII : Rs.";
    cin>>c;
    cout<<"\n\n Science fee (monthly) for following
    groups - ";
    cout<<"\n      Classes IX-X: Rs.";
    cin>>s1;
    cout<<"\n      Classes XI-XII: Rs.";
    cin>>s2;
    cout<<"\n\n Development fee (yearly) for following
    groups - ";
    cout<<"\n      Classes I-V : Rs. ";
    cin>>d1;
    cout<<"\n      Classes VI-X : Rs.";
    cin>>d2;

```

```

        cout<<"\n    Classes XI-XII : Rs.";
        cin>>d3;
        cout<<"\n\n Annual fee (yearly) for following
        groups - ";
        cout<<"\n    Classes I-XII : Rs.";
        cin>>a;
        cout<<"\n\n Admission charges (for one time) for
        following groups - ";
        cout<<"\n    Registration fee : Rs. ";
        cin>>a1;
        cout<<"\n    Admission fee : Rs.";
        cin>>a2;
        cout<<"\n    Security charges : Rs.";
        cin>>a3;
    }

void Fee::print_fee_details()
{
    cout<<"\n-----";
    cout<<"\n\tSCHOOL FEE STRUCTURE FOR SESSION"
    <<session;
    cout<<"\n-----";
    cout<<"\n    PARTICULARS\t\tFEE";
    cout<<"\n\n TUTION FEE(per month) ";
    cout<<"\n    Classes I-V\t\t:"<<t1;
    cout<<"\n    Classes VI-X\t\t:"<<t2;
    cout<<"\n    Classes XI-XII\t\t:"<<t3;
    cout<<"\n\n COMPUTER FEE(per month) ";
    cout<<"\n    Classes I-XII\t\t:"<<c;
    cout<<"\n\n SCIENCE FEE(per month,For science
    students only) ";
    cout<<"\n    Classes IX-X\t\t:"<<s1;
    cout<<"\n    Classes XI-XII\t\t:"<<s2;
    cout<<"\n\n DEVELOPMENT FEE(per year) ";
    cout<<"\n    Classes I-V\t\t:"<<d1;
    cout<<"\n    Classes VI-X\t\t:"<<d2;
    cout<<"\n    Classes XI-XII\t\t:"<<d3;
    cout<<"\n\n ANNUAL FEE(per year) ";
    cout<<"\n    Classes I-XII\t\t:"<<a;
    cout<<"\n\n ADMISSION CHARGES(one time)";

```

```

        cout<<"\n      Registration fee\t\t:"<<a1;
        cout<<"\n      Admission fee\t\t:"<<a2;
        cout<<"\n      Security (Refundable)\t\t:"<<a3;
        cout<<"\n-----";
    }

void fee_add()
{
    fout.open("FEES.dat",ios::out|ios::app);
    if(!fout)
    {
        cout<<"\nFile could not be opened !";
        exit(-1);
    }
    ff.enter_fee_details();
    fout.write((char*)&ff,sizeof(ff));
    fout.close();
    getch();
    fee_menu();
}

void fee_Modify()
{
    fin.open("FEES.dat",ios::out);
    fin.seekg(0);
    if(!fin)
    {
        cout<<"\nFile could not be opened !";
        exit(-1);
    }
    cout << "\nEnter the new details of fee - ";
    ff.enter_fee_details();
    int pos = -1 * sizeof(ff);
    fin.seekp(pos,ios::cur);
    fin.write((char*)&ff,sizeof(ff));
    cout<<"\nFee details has been updated.";
    fin.close();
    getch();
    fee_menu();
}

```

```

void fee_status()
{
    int regno;

    fin.open("STUDENTS.dat",ios::in|ios::out|ios::app);
    fout.open("FEES.dat",ios::out|ios::app);
    cout<<"\nEnter admission number of student : ";
    cin>>regno;
    char found='n';
    if(!fin)
    {
        cout<<"File could not be opened !";
        exit(-1);
    }
    while(!fin.eof())
    {
        fin.read((char*)&s,sizeof(s));
        if(s.search(regno)=='y')
        {
            s.enter_status(regno);
            fout.write((char*)&s,sizeof(Fee));
            found='y';
        }
    }

    if(found=='n')
    {
        cout<<"\nThe record of the student does not
        exists";
    }
    fin.close();
    fout.close();
    fee_menu();
}

void display_stu_fee()
{
    int regno;
    fin.open("STUDENTS.dat",ios::in|ios::out);
    fout.open("FEES.dat",ios::in|ios::out);

```



```

cout<<"\nEnter admission number of student : ";
cin>>regno;
char found='n';
// for searching in STUDENTS.dat's file
char got='n';    // for searching in FEES's file
while(!fin.eof())
{
    fin.read((char*)&s,sizeof(s));
    if(s.search(regno)=='y')
    {
        found='y';
        s.stu_dis();
        break;
    }
}
while(fout.read((char*)&s,sizeof(Fee)))
{
    if(s.fsearch(regno)=='y')
    {
        got='y';
        s.displayfee();
        break;
    }
}
if(got=='n')
{
    cout<<"\nFee records not found !";
}
if(found=='n')
{
    cout<<"\nThe record of the student does not
    exists";
}
fin.close();
fout.close();
getch();
fee_menu();
}

void marks_entry()
{

```

```

int regno;
fin.open("STUDENTS.dat",ios::in|ios::out|ios::app);
fout.open("MARKS.dat",ios::out|ios::app);
cout<<"\nEnter admission number of student : ";
cin>>regno;
char found='n';
if(!fin)
{
    cout<<"File could not be opened !";
    exit(-1);
}
while(!fin.eof())
{
    fin.read((char*)&s,sizeof(s));
    if(s.search(regno)=='y')
    {
        found='y';
        s.get_marks(regno);
        fout.write((char*)&s,sizeof(Marks));
        break;
    }
}
if(found=='n')
{
    cout<<"\nThe record of the student does not exists";
}
else
    cout<<"\nMarks entry saved.";
fin.close();
fout.close();
getch();
main_menu();
}

void display_stu_marks()
{
    int regno;
    fin.open("STUDENTS.dat",ios::in);
    fout.open("MARKS.dat",ios::in);
    cout<<"\nEnter admission number of student : ";

```

```

cin>>regno;
char found='n';
// for searching in STUDENTS.dat's file
char got='n';    // for searching in MARKS's file
while(!fin.eof())
{
    fin.read((char*)&s,sizeof(s));
    if(s.search(regno)=='y')
    {
        found='y';
        s.stu_dis();
        break;
    }
}
while(fout.read((char*)&s,sizeof(Marks)))
{
    if(s.msearch(regno)=='y')
    {
        s.disp_marks();
        got='y';
        break;
    }
}
fin.close();
fout.close();
getch();
main_menu();
}

void marks_menu()
{
    system("cls");
    int ch;
    cout<<"\n\t-----";
    cout<<"\n\t\t\tREPORT CARD MANAGEMENT\n";
    cout<<"\n\t-----";
    cout<<"\n\t\t1. Enter marks";
    cout<<"\n\t\t2. Display report card";
    cout<<"\n\t\t3. Back to main menu";
    cout<<"\n\t-----";
}

```

```

        cout<<"\n\tEnter Your Choice (1-3) : ";
        cin>>ch;
        while(ch)
        {
            switch(ch)
            {
                case 1: if(verify_password()==1)
                        {
                            system("cls");
                            marks_entry();
                        }
                        else
                        {
                            cout<<"\nYOU ARE NOT AN AUTHORIZED
                            USER !!\n YOU CAN'T ADD NEW DATA.";
                            getch();
                            marks_menu();
                        }
                        getch();
                        marks_menu();
                        break;
                case 2: display_stu_marks();
                        getch();
                        marks_menu();
                        break;
                case 3:  main_menu();
                        break;
                default: cout<<"Wrong Input ! Please choose an
                        option from the list";
                        getch();
                        marks_menu();

            }
            system("cls");
            cin>>ch;
        }
    }
    void fee_menu()
    {
        int ch;
    }

```

```

        system("cls");
        cout<<"\n\t-----
";
        cout<<"\n\t\t\tFEE MANAGEMENT SYSTEM\n";
        cout<<"\n\t-----
";
        cout<<"\n\t\t1. Enter new fee details";
        cout<<"\n\t\t2. View fee details";
        cout<<"\n\t\t3. Modify fee details";
        cout<<"\n\t\t4. Enter fee status of student";
        cout<<"\n\t\t5. View a student's fee detail";
        cout<<"\n\t\t6. Back to main menu";
        cout<<"\n\t-----
";

        cout<<"\n\tEnter Your Choice (1-6) : ";
        cin>>ch;
        while(ch)
        {
            switch(ch)
            {
                case 1: if(verify_password()==1)
                        {
                            system("cls");
                            fee_add();
                        }
                        else
                        {
                            cout<<"\nYOU ARE NOT AN AUTHORIZED
USER !!\n YOU CAN'T ADD NEW DATA.";
                            getch();
                            exit(0);
                        }
                        getch();
                        fee_menu();
                        break;
                case 2: ff.print_fee_details();
                        getch();
                        fee_menu();
                        break;
                case 3: if(verify_password()==1)

```

```

        {
            system("cls");
            fee_Modify();
        }
        else
        {
            cout<<"\nYOU ARE NOT AN AUTHORIZED
            USER !!\n YOU CAN'T MODIFY DATA.";
            getch();
            main_menu();
        }
        getch();
        fee_menu();
        break;
case 4: if(verify_password()==1)
        {
            system("cls");
            fee_status();
        }
        else
        {
            cout<<"\nYOU ARE NOT AN AUTHORIZED
            USER !!\n YOU CAN'T ADD DATA.";
            getch();
            main_menu();
        }
        getch();
        fee_menu();
        break;
case 5: display_stu_fee();
        getch();
        fee_menu();
        break;
case 6: main_menu();
        break;
default: cout<<"Wrong Input ! Please choose an
        option from the list";
        getch();
        fee_menu();
        break;

```

```

    }
    system("cls");
    cin>>ch;
    }
}

//class Teacher starts here
class Teachers
{
    private:
        int t_code;
        char tname[30];
        char design[10];
        char qual[20];
        char add[40] ;
        int exp_yr;
        int salary;
        long long int mobile_no;
    public :
        void t_getdata();
        void t_dis();
        void t_mod();
        int get_code()
            { return t_code; }    // accessor function
}T; //object of class Teachers is created as 'T'.

void Teachers::t_getdata()
{
    system("cls");
    cout<<"-----";
    cout<<"\nTeacher management system - Enter Record";
    cout<<"\n-----";
    cout<<"\n\tEnter teacher's code   : ";
    cin>>t_code;
    cout<<"\n\tEnter teacher's name : ";
    cin.get(ch);    //to clear buffer
    cin.getline(tname,30);
    cout<<"\n\tEnter designation      : ";
    gets(design);

```

```

        cout<<"\n\tEnter qualifications : ";
        gets(qual);
        cout<<"\n\tEnter salary given      : Rs." ;
        cin>>salary;
        cout<<"\n\tEnter experience          : ";
        cin>>exp_yr;
        cout<<"\n\tEnter Mobile no          : ";
        cin>>mobile_no;
        cout<<"\n\tEnter address              :";
        cin.get(ch); //to clear buffer
        gets(add);
};

void Teachers::t_dis()
{
    cout<<"\n===== ";
    cout<<"\n\t\tTEACHER'S DETAILS";
    cout<<"\n===== ";
    cout<<"\n\tTeacher's code      : "<<t_code;
    cout<<"\n\tName                : "<<tname;
    cout<<"\n\tDesignation          : "<<design;
    cout<<"\n\tQualifications        : "<<qual;
    cout<<"\n\tExperience            : "<<exp_yr;
    cout<<"\n\tSalary                : Rs."<<salary;
    cout<<"\n\tMobile no            : "<<mobile_no;
    cout<<"\n\tAddress                : "<<add;
    cout<<"\n===== ";
};

void Teachers::t_mod()
{
    char tnm[30],addr[40],dsg[10],ql[40];
    int exp;
    int sal;
    long long int mob;
    cout<<"\nEnter new data for following details-";
    cout<<"\nName(press '$' for no changes) : ";
    cin.get(ch);
    gets(tnm);
    cout<<"\nDesignation(press '$' for no changes): ";

```



```

        cin.get(ch);
        gets(dsg);
        cout<<"\nQualifications(press '$' for no changes):
";
        gets(ql);
        cout<<"\nExperience(press '0' for no changes): ";
        cin>>exp;
        cout<<"\nSalary(press '0' for no changes) : Rs.";
        cin>>sal;
        cout<<"\nMobile no(press '0' for no changes): ";
        cin>>mob;
        cout<<"\nAddress(press '$' for no changes): ";
        cin.get(ch);
        gets(addr);
        char confirm;
        cout<<"\nAre you sure you want to make these
changes(press'y'for yes and'n' for no):";
        cin>>confirm;
        if(confirm=='y' || 'Y')
        {
            if(strcmp(tnm,"$")!=0)
                strcpy(tname,tnm);
            if(strcmp(dsg,"$")!=0)
                strcpy(design,dsg);
            if(strcmp(ql,"$")!=0)
                strcpy(qual,ql);
            if(exp!=0)
                exp_yr=exp;
            if(sal!=0)
                salary=sal;
            if(mob!=0)
                mobile_no=mob;
            if(strcmp(addr,"$")!=0)
                strcpy(add,addr);
        }
        else
            teachers_menu();
    };

void teachers_menu()

```



```

        else
            cout<<"\nYOU ARE NOT AN AUTHORIZED
            USER !!\n YOU CAN'T MODIFY RECORDS.";
            getch();
            teachers_menu();
            break;
case 5: if(verify_password()==1)
    {
        system("cls");
        t_Delete();
    }
    else
        cout<<"\nYOU ARE NOT AN AUTHORIZED
        USER !!\n YOU CAN'T DELETE RECORDS.";
        getch();
        teachers_menu();
        break;
case 6: main_menu();
        break;
default: cout<<"Wrong Input ! Please choose a
        correct option";
        getch();
        teachers_menu();

    }
    system("cls");
    cin>>ch;
}

//For adding the data into the file.
void t_add()
{
    char ch='y';
    fin.open("TEACHERS.dat",ios::out|ios::app);
    if(!fin)
    {
        cout<<"\nFile could not be opened !";
        exit(-1);
    }
}

```

```

        while(ch=='y' || ch=='Y')
        {
            T.t_getdata();
            fin.write((char*)&T, sizeof(T));
            cout<<"\nData Appended and Transferred to
file.\n";
            cout<<"\nDo you want to add more : ";
            cin>>ch;
        }
        fin.close();
        getch();
        teachers_menu();
    }

    //For Displaying the all data of the teachers.
    void t_displayall()
    {
        fin.open("TEACHERS.dat", ios::in);
        if(!fin)
        {
            cout<<"\nFile could not be opened !";
            exit(-1);
        }
        while(!fin.eof())
        {
            fin.read((char*)&T, sizeof(T));
            T.t_dis();
        }
        fin.close();
        getch();
        teachers_menu();
    }

    //search teacher's record
    void t_search()
    {
        ifstream fin("TEACHERS.dat", ios::in);
        int tno;
        char found='n';    //initially record is not found
        cout<<"\nEnter Teacher's code to be searched : ";
    }

```

```

        cin>>tno;
        if(!fin)
        {
            cout<<"\nFile could not be opened !";
            exit(-1);
        }
        while(!fin.eof())
        {
            fin.read((char*)&T,sizeof(T));
            if(T.get_code()==tno)
            {
                T.t_dis();
                found='y';
                break;
            }
        }

        if(found=='n')
            { cout<<"\nThe Code "<<tno<<" is not in the
            file!\n";
            }
        getch();
        teachers_menu();
        fin.close();
    }

// to modify teacher's record
void t_Modify()
{
    fin.open("TEACHERS.dat",ios::in|ios::out);
    int tno;
    char found='n';
    fin.seekg(0);
    cout<<"\nEnter the Teacher's code whose record is
    to be modified: ";
    cin>>tno;
    if(!fin)
    {
        cout<<"\nFile could not be opened !";
        exit(-1);
    }

```

```

    }
    while(!fin.eof())
    {
        fin.read((char*)&T,sizeof(T));
        if(T.get_code()==tno)
        {
            T.t_mod();
            int pos=(-1)*sizeof(T);
            fin.seekp(pos,ios::cur);
            fin.write((char*)&T,sizeof(T));
            //overwritting
            cout<<"\nRecord of code "<<tno<<" has been
            updated.";
            cout<<"\nUpdated record is here -\n";
            T.t_dis();
            break;
        }
    }
    if(found=='n')
        cout<<"\nThe teacher's code "<<tno<<" is not in the
file!\n";
    fin.close();
    getch();
    teachers_menu();
}

//t delete a teacher's record
void t_Delete()
{
    int tno;
    char found='n'; //initially the record is not found
    char confirm='y';
    //yes-initially confirmation for deletion
    fin.open("TEACHERS.dat",ios::in);
    fout.open("temp.dat",ios::out);
    cout<<"\nEnter the teacher's code whose record is
to be deleted: ";
    cin>>tno;
    if(!fin)
    {

```

```

        cout<<"\nFile could not be opened !";
        exit(-1);
    }
    while(fin.read((char*)&T,sizeof(T)))
    {
        if(T.get_code()==tno)
        {
            T.t_dis();
            found='y';
            cout<<"\nAre you sure you want to delete
this record(y/n) :";
            cin>>confirm;
            if(confirm=='n')
            {
                fout.write((char*)&T,sizeof(T));
            }
        }
        else
            fout.write((char*)&T,sizeof(T));
    }
    if(found=='n')
        cout<<"\nThe teacher's code "<<tno<<" is not in
the file!\n";
    fin.close();
    fout.close();
    remove("TEACHERS.dat");
    rename("temp.dat","TEACHERS.dat");
    getch();
    teachers_menu();
}

//main menu of the project
void main_menu()
{
    system("cls");
    cout<<"\t-----";
    cout<<"\n\n\t\t\tSCHOOL MANAGEMENT PROGRAM\n\n";
    cout<<"\t-----";
    cout<<"\n\n\tWelcome to the main menu ";
    cout<<"\n\t 1.Student management system";
}

```

```

cout<<"\n\t 2.Report card management";
cout<<"\n\t 3.Fee management system";
cout<<"\n\t 4.Teacher management system";
cout<<"\n\t 5.Users menu";
cout<<"\n\t 6.Exit";
cout<<"\n\t-----";
cout<<"\n\n\t Enter your choice (1-6) : ";
char ch;//for getting choice
cin>>ch;
while(1)//outer loop
{
switch(ch)
{
    case '1' :    system("cls");
                  while(1)
                  {
                      students_menu();
                  }
                  break;
    case '2' :    system("cls");
                  while(1)
                  {
                      marks_menu();
                  }
                  break;
    case '3' :    system("cls");
                  while(1)
                  {
                      fee_menu();
                  }
                  break;
    case '4' :    system("cls");
                  while(1)
                  {
                      teachers_menu();
                  }
                  break;
    case '5' :    system("cls");
                  while(1)
                  {

```



```

                                users_menu();
                                }
        case '6' :    exit(0);
                    break;
        default  :    cout<<"Wrong choice. Please enter
                        again!";
                    }
    }

//MAIN FUNCTION
int main()
{
    cout<<"\n\n\n\n\t-----
-----\n\t\t\t\t PROJECT WORK\n\t\t\t\t ";
    cout<<"    BY\n\t\t\t\t KRITIKA YADAV\n\t\t\t\t CLASS-
12A\n\t\t\t\t UNDER THE GUIDANCE OF \n\t\t\t\t";
    cout<<"Mr. ASHRAF JAMAL\n\t-----
-----";
    getch();
    main_menu();
}
// END OF MAIN FUNCTION

```