

DR.BR AMBEDKAR NATIONAL INSTITUTE **OF TECHNOLOGY , JALANDHAR**



DBMS File Handling Project

SUBMITTED TO:

Dr. Rajneesh Rani

ASST. PROF . DEPT. OF CSE

SUBMITTED BY:-

1. ANAMIKA SINGH (19104008)
2. BHUWAN SHARMA (19104018)
3. DINESH (19104028)
4. JAINAM SOGANI (19104038)
5. NIMIT JAIN (19104058)
6. POTTA LOKESH (19104068)
7. RAVI (19104078)
8. SARTHAK GUPTA (19104088)
9. SHIV KUMAR (19104098)
10. TEJAVATH NAVEEN (19104108)
11. YASH VIJAYVARGIYA (19104118)
12. SHREENAV KHANDELWAL (19106089 Branch Change)

INTRODUCTION

We developed this project on the basis of our learning in DBMS, SQL, File handling and programming language(C++). Given below is the source code for the file handling project.

C++ Code For the File Handling Project :

```
#include<bits/stdc++.h>
#include<conio.h>
using namespace std;
string dept[] = {"CS", "EC", "IT", "EE", "ME"};

class student
{
public:
    int reg, year, dob;
    long long contact_number;
    string name, fname, degree, address, branch;
    void input()
    {
        cout << "\n Enter name: ";
        //cin.ignore();
        getline(cin, name);
        cout << "\n Enter roll no: ";
```

```
cin >> reg;
cout << "\n Enter Date of birth (DDMMYYYY) :: ";
cin >> dob;

cout << "\n Enter Fathers Name :: ";
cin.ignore();
getline(cin, fname);
cout << "\nEnter Degree :: ";
cin.ignore();
cin >> degree;
cout << "\nEnter Year :: ";
cin >> year;

cout << "\nEnter Branch :: ";
cin.ignore();
cin >> branch;
cout << "\nEnter Address :: ";
cin.ignore();
getline(cin, address);
cout << "\nEnter Contact_Number :: ";
cin >> contact_number;
fflush(stdin);

name.resize(20), fname.resize(20), degree.resize(10),
address.resize(30), branch.resize(10);
```

```

}
void display()
{
    system("CLS");
    cout << "\t\tDisplay Records";
    cout << "\n";
    cout << "\n Name - " << name;
    cout << "\n Reg no. - " << reg;
    cout << "\n D.O.B - " << dob;
    cout << "\n Fathers Name - " << fname;
    cout << "\n Degree - " << degree;
    cout << "\n Year - " << year;
    cout << "\n Branch - " << branch;
    cout << "\n Address - " << address;
    cout << "\n Contact No. - " << contact_number;
    cout << "\n";
    system("PAUSE");
    system("CLS");
}
bool operator == (student a)
{
    if (reg == a.reg)
        return true;
    else

```

```
        return false;
    }
};

vector <student>v;

int search_reg(int reg, int &i);

void get_file()
{

    int i = 0;
    fstream f;
    int length;
    ifstream filestr;

    filestr.open("College.txt", ios::binary); // open your file
    filestr.seekg(0, ios::end); // put the "cursor" at the end of the file
    length = filestr.tellg(); // find the position of the cursor
    filestr.close();

    if (length != 0)
    {
        f.open("College.txt", ios::in);
        int n;
        f >> n;
        string s;
        getline(f, s);
```

```

while (n--)
{
    student x;

    cout << f.eof();

    f >> x.reg >> x.name >> x.fname >> x.dob >> x.degree >>
x.year >> x.branch >> x.address >> x.contact_number;

    replace(x.name.begin(), x.name.end(), '_', ' ');
    replace(x.address.begin(), x.address.end(), '_', ' ');
    replace(x.fname.begin(), x.fname.end(), '_', ' ');


    v.push_back(x);
}
cout << f.eof();
}
f.close();
}

void bubblesort()
{
    int i, j;
    student x;
    for (i = 0; i < v.size(); i++)
        for (j = 0; j < v.size() - i - 1; j++)
            if (v[j].reg > v[j + 1].reg)

```

```

        {
            x = v[j];
            v[j] = v[j + 1];
            v[j + 1] = x;
        }
    }

void insert_new()
{
    char ch = 'y';
    int ta;
    while (ch == 'y')
    {
        fflush(stdin);
        student x;
        x.input();
        if (search_reg(x.reg, ta) == 0)
            v.push_back(x);
        else
            cout << "\nTHE REGISTRATION NO. ALREADY  
EXIST!!!\nCANNOT ADD";
        cout << "\n Press [Y] to enter more: ";
        cin >> ch;
        fflush(stdin);
    }
}

```

```

void write_file()
{
    bubblesort();
    fstream f;
    f.open("College.txt", ios::out);

    f << v.size() << endl;
    f << "Roll No.\tName\tFathers
Name\tD.O.B\tDegree\tYear\tBranch\tAddress\tContact
Number\n";
    for (int i = v.size() - 1; i >= 0; i--)
    {
        student x = v[i];
        replace(x.name.begin(), x.name.end(), ' ', '_');
        replace(x.address.begin(), x.address.end(), ' ', '_');
        replace(x.fname.begin(), x.fname.end(), ' ', '_');

        f << x.reg << ' ' << x.name << ' ' << x.fname << ' ' << x.dob << ' '
<< x.degree << ' ' << x.year << ' ' << x.branch << ' ' << x.address << '
' << x.contact_number << endl;
    }
    f.close();
}

int search_reg(int reg, int &i)
{
    int ta = 0;

```



```
for (i = 0; i < v.size(); i++)
    if (v[i].reg == reg)
    {
        ta = 1;
        break;
    }
return ta;
}

int search_name(string name, vector<int> &vi)
{
    int i, ta = 0;
    for (i = 0; i < v.size(); i++)
        if (v[i].name == name)
        {
            ta = 1;
            vi.push_back(i);
        }
    return ta;
}

int search_branch(string branch, vector<int> &vj)
{
    int i, ta = 0;
    for (i = 0; i < v.size(); i++)
        if (v[i].branch == branch)
        {
```

```

        ta = 1;
        vj.push_back(i);
    }
    return ta;
}

void search_and_show()
{
    int ch, i, ta = 0;
    string name, branch;
    vector <int>vi;
    vector <int>vj;
    int reg;
poi:
    cout << "\n1.Press to search reg no."
        << "\n2.Press to search name"
        << "\n3.Press to search branch";
    cin >> ch;
    switch (ch)
    {
    case 1:
        cout << "\nEnter reg no.: ";
        cin >> reg;
        if (search_reg(reg, i) == 1)
            v[i].display();
        else

```

```
        cout << "\nRecord NOT FOUND!!!";  
        break;  
case 2:  
        cout << "\nEnter name: ";  
        cin.ignore();  
        getline(cin, name);  
        if (search_name(name, vi) == 1)  
        {  
            for (int j = 0; j < vi.size(); j++)  
                v[vi[j]].display();  
        }  
        else  
            cout << "\nRecord NOT FOUND!!!";  
        break;  
case 3:  
        cout << "\nEnter branch: ";  
        cin.ignore();  
        getline(cin, branch);  
        if (search_branch(branch, vj) == 1)  
        {  
            for (int j = 0; j < vj.size(); j++)  
                v[vj[j]].display();  
        }  
        else  
            cout << "\nRecord NOT FOUND!!!";
```

```
        break;
default:
    cout << "\nWrong CHOICE!!!";
    goto poi;
}
}

void show()
{
    int i;
    for (i = 0; i < v.size(); i++)
        v[i].display();
}

void delete_data()
{
    int i, j;
    int reg;
    vector <student>::iterator p = v.begin();
    cout << "\nEnter Reg. no.: ";
    cin >> reg;
    if (search_reg(reg, i) == 1)
    {
        student x = v[i];
        cout << "\nThe following data is being deleted";
        x.display();
        p += i;
```

```

        v.erase(p, p + 1);
    }
}

void edit_data()
{
    int i, j;
    int reg;
    vector <student>vi;
    cout << "\nEnter Reg. no.: ";
    cin >> reg;
    if (search_reg(reg, i) == 1)
    {
        cout << "\nEnter new data:";
        fflush(stdin);
        v[i].input();
    }
}

class Examination_Schedule
{
    int date[5];
public:
    void MakeExaminationSchedule()
    {
        // system("cls");
        // fstream file3;

```

```

cout << "\t\t\t\t\t-----" << endl;
cout << "\t\t\t\t\t| MAKE EXAMINATION SCHEDULE |" <<
endl;
cout << "\t\t\t\t\t-----" << endl;

cout << endl << endl << endl;

```

```

cout << "Enter the dates on which you want to conduct
examination\n";

```

```

for (int i = 0 ; i < 5 ; i++)
    cin >> date[i];
}

```

```

void Addtofile()

```

```

{
    system("cls");
    fstream file3;

```

```

    file3.open("Schedule.txt", ios::out | ios::trunc);

```

```

    for (int i = 1 ; i <= 5 ; i++)

```

```

    {
        file3 << dept[i - 1] << endl ;

```

```

        for (int j = 0 ; j <= 8 ; j += 2)

```

```

{
    if (j == 0)
    {
        string s = "Date";
        s.resize(20);
        file3 << s;
    }
    else
    {
        string s = "Sem ";
        int p = j;
        stringstream ss;
        ss << p;
        string t;
        ss >> t;
        s = s + t;
        s.resize(20);
        file3 << s;
    }
}

```

```

file3 << "MTech";

```

```

file3 << endl;

```

```

file3 << "-----\n";
-----\n";

```

```
int count = 5;
srand(time(NULL));

bool a[5] = {0, 0, 0, 0, 0};
while (count > 0)
{
    int k = rand() % 5;

    if (a[k] == 0)
        a[k] = 1;
    else
        continue;

    stringstream ss;
    ss << date[5 - count];
    string t;
    ss >> t;
    t.resize(20);
    file3 << t;
    for (int j = 0 ; j < 5 ; j++)
    {
        string s;
        s = s + dept[i - 1];
        int p = k + 1 + 5 * j;
```



```

        stringstream ss;
        ss << p;
        string t;
        ss >> t;
        s = s + t;
        s.resize(20);
        file3 << s;
    }

    file3 << endl;

    count--;
}
file3 << endl;
}
file3.close();
}

void ShowSchedule()
{
    system("cls");
    cout << "\t\t\t\t\t-----" << endl;
    cout << "\t\t\t\t\t| EXAMINATION SCHEDULE |" << endl;
    cout << "\t\t\t\t\t-----" << endl;
    cout << endl << endl << endl;
}

```



```
cout << "Enter your choice :";  
cin >> choice;  
string dec;  
switch (choice)  
{  
case 1: MakeExaminationSchedule(); Addtofile();  
    break;  
case 2: ShowSchedule(); getch();  
    break;  
case 3: return;  
default:  
    cout << "INVALID INPUT :(" << endl;  
}  
  
goto menuagain;  
fflush(stdin);  
}  
  
void studentview()  
{  
menuagain:  
system("cls");  
cout << "\t\t\t\t\t-----" << endl;  
cout << "\t\t\t\t\t WELCOME STUDENT |" << endl;
```

```

cout << "\t\t\t\t\t-----" << endl;
int choice;
cout << "\t\t\t\t\t1.SHOW EXAM SCHEDULE" << endl;
cout << "\t\t\t\t\t2.EXIT" << endl;
cout << "Enter your choice :";
cin >> choice;
switch (choice)
{
case 1: ShowSchedule(); getch();
    break;
case 2: write_file(); exit(1); return;
default:
    cout << "INVALID INPUT :(" << endl;
}
goto menuagain;
}
};
class Student_Result
{
public:
    long long int student_Id;
    float student_cgpa;

public:
    void SearchStudentRecord()

```

[illegible]

```
cin >> student_Id;
```

[illegible]

```
{  
    if (sId == student_Id)  
    {  
  
        cout << "ENTER UPDATED STUDENT CGPA" << endl;  
        cin >> student_cgpa;  
        file4 << student_Id << " " << student_cgpa << endl;  
  
    }  
    else  
    {  
        file4 << student_Id << " " << student_cgpa << endl;  
    }  
  
    file3 >> student_Id >> student_cgpa;  
  
}  
  
file3.close();  
file4.close();  
  
remove("Result.txt");  
rename("Temp.txt", "Result.txt");
```



```
void DeleteStudentRecord()
{
    system("cls");
    fstream file3, file4;

    cout << "\t\t\t\t\t-----" << endl;
    cout << "\t\t\t\t\t| DELETE STUDENT RECORD |" << endl;
    cout << "\t\t\t\t\t-----" << endl;

    cout << endl << endl << endl;

    long long int sId;

    cout << "ENTER THE ID OF THE STUDENT THAT YOU  
WANT TO UPDATE: ";

    cin >> sId;

    file3.open("Result.txt", ios::in);
    file3 >> student_Id >> student_cgpa;
    file4.open("Temp.txt", ios::out | ios::app);


    while (!file3.eof())
    {
        if (sId != student_Id)
```

$$\{$$

```
file4 << student_Id << student_cgpa << endl;
```

$$\}$$

```
file3 >> student_Id >> student_cgpa;
```

$$\}$$

```
file3.close();
```

```
file4.close();
```

```
remove("Result.txt");
```

```
rename("Temp.txt", "Result.txt");
```

$$\}$$

```
void ShowStudentRecords()
```

$$\{$$

```
system("cls");
```

```
fstream file3;
```

```
cout << "\\t\\t\\t\\t\\t\\t-----" << endl;
```

```
cout << "\\t\\t\\t\\t\\t\\tSTUDENT RECORDS|" << endl;
```

```
cout << "\\t\\t\\t\\t\\t\\t-----" << endl;
```

```
cout << endl << endl << endl;
```

```
file3.open("Result.txt", ios::in);
```

```
file3 >> student_Id >> student_cgpa;
```

```
student x;
```

```
for (auto i : v) {
```

```
    if (i.reg == student_Id) {
```

```
        x = i;
```

```
        break;
```

```
    }
```

```
}
```

```
while (!file3.eof())
```

```
{
```

```
    cout << setfill(' ') << setw(20) << student_Id << setfill(' ') <<  
    setw(20) << x.name << setfill(' ') << setw(20) << x.degree << setfill(''  
'') << setw(20) << x.year << setfill(' ') << setw(20) << x.branch <<  
    setfill(' ') << setw(20) << student_cgpa << endl << endl;
```

```
file3 >> student_Id >> student_cgpa;
```

```
for (auto i : v) {
```

```
    if (i.reg == student_Id) {
```

```
        x = i;
```

```
        break;
```

```

    }
}

}

file3.close();
}

void ShowResult()
{
    system("cls");
    cout << "\t\t\t\t\t-----" << endl;
    cout << "\t\t\t\t\t| RESULT PORTAL OF STUDENT |" << endl;
    cout << "\t\t\t\t\t-----" << endl;
    cout << endl << endl << endl;
    long long int sId;
    string sName;
    fstream file3;
    cout << "ENTER STUDENT ID :" << endl;
    cin >> sId;


    file3.open("Result.txt", ios::in);
    file3 >> student_Id >> student_cgpa;
    while (!file3.eof())
    {
        if (sId == student_Id )
        {

```

```
cout << student_cgpa << endl;
}
```

```
file3 >> student_Id >> student_cgpa;
```

$$\}$$

```
file3.close();
```

$$\}$$

```
void adminview()
```

$$\{$$

menuagain:

```
system("cls");
```

```
cout << "\\t\\t\\t\\t\\t-----" << endl;
```

```
cout << "\t\t\t\t\t WELCOME ADMINISTRATOR |" << endl;
```

```
cout << "\\t\\t\\t\\t\\t-----" << endl;
```

```
int choice;
```

```
cout << "\t\t\t\t\t1.ADD STUDENT RECORDS" << endl;
```

```
cout << "\t\t\t\t\t2.SHOW STUDENT RECORDS" << endl;
```

```
cout << "\t\t\t\t\t3.SEARCH STUDENT RECORDS" << endl;
```

```
cout << "\t\t\t\t\t4.UPDATE STUDENT RECORDS" << endl;
```

```
cout << "\t\t\t\t\t5.DELETE STUDENT RECORDS" << endl;
```

```
cout << "\\t\\t\\t\\t\\t\\t6.EXIT" << endl;
```

```
cout << "Enter your choice :";
cin >> choice;
string dec;
switch (choice)
{

case 1: do {
    AddStudentRecord();
    cout << "DO YOU WISH TO ADD ANOTHER RECORD :
[YES/NO]" << endl;
    cin >> dec;
    } while ((dec == "YES") || (dec == "yes"));
    break;
case 2: ShowStudentRecords();
    getch();
    break;
case 3: SearchStudentRecord();
    getch();
    break;
case 4: UpdateStudentRecord();
    break;
case 5: DeleteStudentRecord();
    break;
case 6: write_file(); exit(1);
```

default:

```
cout << "INVALID INPUT :(" << endl;
```

```
}
```

```
goto menuagain;
```

```
}
```

```
};
```

```
int main()
```

```
{
```

```
Examination_Schedule Authorisation;
```

```
Student_Result res;
```

```
int i = 1, mode;
```

```
get_file();
```

```
while (i)
```

```
{
```

```
system("CLS");
```

```

cout << "\t\t\t-----\n";
cout << "\t\t\t\t University Management System\n";
cout << "\t\t\t\t-----\n";
cout << "\n\t\tEnter <1> for Administrator Mode"
    << "\n\t\tEnter <2> for Student Mode : ";
cin >> mode;
if (mode == 1) {
    cout << "\n\t\t\tEnter <1> to Add new student"
        << "\n\t\t\tEnter <2> to Display all student"
        << "\n\t\t\tEnter <3> to Remove student"
        << "\n\t\t\tEnter <4> to Edit student"
        << "\n\t\t\tEnter <5> to Search student"
        << "\n\t\t\tEnter <6> to Make Examination Schedule"
        << "\n\t\t\tEnter <7> to Change Result"
        << "\n\t\t\tEnter <0> to Exit\n";
    cout << "\n\n\t\t\tEnter Your Choice:";
    cin >> i;
    switch (i)
    {
    case 1 :
        insert_new();
        break;
    case 2 :
        show();
        break;

```



```
case 3 :
    delete_data();
    break;
case 4 :
    edit_data();
    break;
case 5 :
    search_and_show();
    break;
case 6:
    Authorisation.adminview();
    break;
case 7: res.adminview();
    break;
case 0 :
    write_file();
    break;
default :
    cout << "\nWRONG CHOICE!!!\nTRY AGAIN";
}
}

//mode 1 ends

else if (mode == 2) {
    cout << "\n\t\t\tEnter <1> to Show Examination Schedule\n";
    cout << "\n\t\t\tEnter <2> to Show Result";
```

```
cout << "\n\n\t\tEnter Your Choice:";
cin >> i;
switch (i)
{
case 1:
    Authorisation.studentview();
    break;
case 2: res.ShowResult();
    getch();
    break;
default :
    cout << "\nWRONG CHOICE!!!\nTRY AGAIN";
}
}
}
return 0;
}
```
