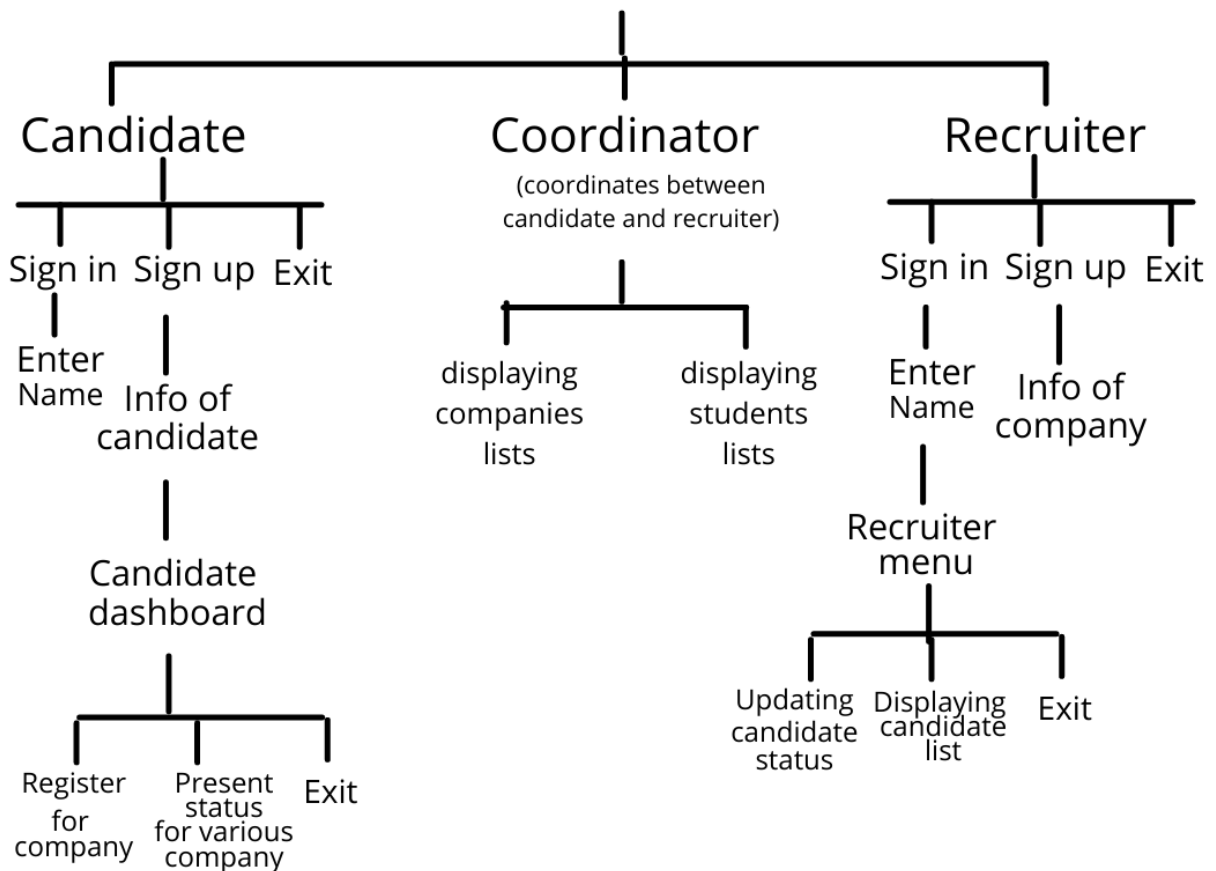


Placement Management System

Placement Management Program



Program Code

```
#include <bits/stdc++.h>
using namespace std;

class Candidate
{
private:
    string name;
    int rollNo;
    float cgpa;
    string branch;
    vector<pair<string, string>> myCompanies;

public:
    Candidate(string name, int rollNo, float cgpa, string branch)
    {
        this->name = name;
        this->rollNo = rollNo;
        this->cgpa = cgpa;
        this->branch = branch;
    }
    void registration();
    void displayCompanies();
    friend class coordinator;
    friend class Recruiter;
};

class Recruiter
{
private:
    string name;
    float MinCGPA;
    string JobProfile;
    vector<Candidate *> Students;

public:
    Recruiter(string N, float C, string j)
    {
        this->name = N;
```

```

        this->MinCGPA = C;
        this->JobProfile = j;
    }
    void StudentsEngaged();
    void StudentsUpdation();
    friend class coordinator;
};

class coordinator
{
public:
    vector<Candidate *> students;
    vector<Recruiter *> companies;
    bool checkEligibility(Candidate *c, Recruiter *r);
    void companiesList();
    void studentsList();
    void updation(Candidate *c, char status, string n);
    Recruiter *pointerToCompany(string nam);
    Candidate *findCand(string str);
    friend class Recruiter;
    friend class Candidate;
};

coordinator coord;

void Candidate ::registration()
{
    coord.companiesList();
    cout << "Enter company's index to register: " << endl;
    Candidate *ptr = this;
    int a;
    cin >> a;
    while (a != -1)
    {
        bool b = coord.checkEligibility(ptr, coord.companies[a - 1]);
        if (b)
        {
            cout << "Registered!" << endl;
        }
        else
    }
}

```

```

        {
            cout << "Not Eligible!" << endl;
        }

        cout << "Press -1 is you don't want to register for anymore
companies" << endl;
        cin >> a;
    }
}

void Candidate ::displayCompanies()
{
    for (auto it : myCompanies)
    {
        cout << it.first << " -> " << it.second << endl;
    }
}

void Recruiter ::StudentsEngaged()
{
    cout << endl;
    for (int i = 0; i < Students.size(); i++)
    {
        cout << Students[i]->name << endl;
    }
    cout << endl;
}

void Recruiter ::StudentsUpdation()
{
    // call function from coordinator
    int n = Students.size();
    for (int i = 0; i < n; i++)
    {
        cout << "Name of Candidate : " << Students[i]->name << endl;
        char status;
        cout << "press R for 'Rejected' and S for 'Selected for next
round' :";
        cin >> status;
        cout << endl;
    }
}

```

```

        vector<Candidate *>::iterator it1;
        it1 = Students.begin() + i;
        if (status == 'R')
            Students.erase(it1);
        coord.updation(Students[i], status, name);
    }
}

bool coordinator ::checkEligibility(Candidate *c, Recruiter *r)
{
    if (c->cgpa >= r->MinCGPA)
    {
        r->Students.push_back(c);
        c->myCompanies.push_back(make_pair(r->name, "Registered"));
        return true;
    }
    return false;
}

void coordinator ::companiesList()
{
    int count = 1;
    for (auto x : companies)
    {
        cout << "Index: " << count << " ";
        cout << x->name << endl;
        cout << "Min CGPA required : " << x->MinCGPA << endl;
        cout << "Job Profile: " << x->JobProfile << endl;
        cout << endl;
        count++;
    }
}

void coordinator::studentsList()
{
    for (auto a : students)
    {
        cout << "Name : " << a->name << "\n";
        cout << "Roll No.: " << a->rollNo << "\n";
        cout << "CGPA: " << a->cgpa << "\n";
    }
}

```

```

        cout << "Branch : " <<a->branch << "\n";
        cout << endl;
    }
}

void coordinator ::updation(Candidate *c, char status, string n)
{
    int ind = 0;
    for (int i = 0; i < c->myCompanies.size(); i++)
    {
        if (c->myCompanies[i].first == n)
        {
            ind = i;
        }
    }
    if (status == 'R')
    {
        c->myCompanies[ind].second = "Rejected";
    }
    else if (status == 'S')
    {
        c->myCompanies[ind].second = "Selected";
    }
}

Recruiter *coordinator::pointerToCompany(string nam)
{
    Recruiter *pt = NULL;
    for (auto it : companies)
    {
        if (nam == it->name)
        {
            pt = it;
            break;
        }
    }
    return pt;
}

Candidate *coordinator::findCand(string str)
{

```

```

Candidate* pt=NULL;
for (auto it : students)
{
    if (str == it->name)
    {
        pt = it;
        break;
    }
}
return pt;
}

void comenu()
{
    int choice;
    do
    {
        cout << "Welcome to Coordinator Menu" << endl;
        cout << "Enter 1 for displaying companies lists: " << endl;
        cout << "Enter 2 for displaying students lists: " << endl;
        cout << "Enter 0 to exit." << endl;
        cin >> choice;
        switch (choice)
        {
            case 1:
                coord.companiesList();
                break;
            case 2:
                coord.studentsList();
                break;
            case 0:
                return;
        }
    } while (choice);
}

void recmenu(Recruiter *pt)
{
    int ch;
    do

```

```

{
    cout << "Welcome to Recruiter Menu" << endl;
    cout << "Enter 1 for Updating Candidate: " << endl;
    cout << "Enter 2 for Displaying Candidate List: " << endl;
    cout << "Enter 0 for Exit: " << endl;
    cin >> ch;
    switch (ch)
    {
        case 1:
            pt->StudentsUpdation();
            break;

        case 2:
            pt->StudentsEngaged();
            break;

        case 0:
            return;

        default:
            cout << "Invalid List!" << endl;
            break;
    }
} while (ch);
}

Recruiter *makeRecruiter()
{
    cout<<"Enter company name : ";
    string n;
    cin >> n;
    cout<<"Enter minimum CGPA : ";
    float cg;
    cin >> cg;
    cout<<"Enter Job Title : ";
    string jb;
    cin >> jb;
    Recruiter *ptr = new Recruiter(n, cg, jb);
    coord.companies.push_back(ptr);
    return ptr;
}

```



```

void recmenuStarting()
{
    int ch;
    Recruiter *pt;
    string inputname;
    do
    {
        cout << "Enter 1 for Signing In: " << endl;
        cout << "Enter 2 for Signing Up: " << endl;
        cout << "Enter 3 to exit." << endl;

        cin >> ch;
        switch (ch)
        {
            case 1:
                cout << "Enter your name: " << endl;
                cin>>inputname;
                pt = coord.pointerToCompany(inputname);
                if (pt)
                    recmenu(pt);
                else
                    cout << "Not Found"<<endl;
                break;

            case 2:
                pt = makeRecruiter();
                recmenu(pt);
                break;

            case 3:
                return;

            default:
                cout << "Invalid Choice!" << endl;
                break;
        }
    } while (ch);
}

```

```

Candidate *getDetails(void)

```

```

{
    string name, branch;
    int rollNo;
    float cgpa;
    cout << "Enter name of a Candidate: ";
    cin >> name;
    cout << "Enter roll number of Candidate: ";
    cin >> rollNo;
    cout << "Enter cgpa of Candidate: ";
    cin >> cgpa;
    cout << "Enter branch of Candidate: ";
    cin >> branch;
    Candidate *p = new Candidate(name, rollNo, cgpa, branch);
    coord.students.push_back(p);
    return p;
}

void candidatemenu(Candidate *pt)
{
    int ch;
    do
    {
        cout << "Welcome to the candidate dashboard" << endl;
        cout << "Enter 1 for viewing your present status with the various companies: " << endl;
        cout << "Enter 2 for registering for a new company:" << endl;
        cout << "Enter 0 for Exit: " << endl;
        cin >> ch;
        switch (ch)
        {
            case 1:
                pt->displayCompanies();
                break;

            case 2:
                pt->registration();
                break;
            case 0:
                return;
        }
    }
}

```

```

    } while (ch);
}

void candMenuStarting()
{
    int ch;
    Candidate *ptrC;
    string st;
    do
    {
        cout << "Enter 1 for Signing In: " << endl;
        cout << "Enter 2 for Signing Up: " << endl;
        cout << "Enter 3 to exit." << endl;
        cin >> ch;
        switch (ch)
        {
            case 1:
                cout << "Enter your name: " << endl;
                cin>>st;
                ptrC = coord.findCand(st);
                if (ptrC)
                {
                    candidatemenu(ptrC);
                }
                else
                {
                    cout << "Not Found"<<endl;
                }
                break;

            case 2:
                ptrC = getDetails();
                candidatemenu(ptrC);
                break;

            case 3:
                return;

            default:
                cout << "Invalid Choice!" << endl;
                break;
        }
    } while (ch);
}

```

```
int main()
{
    int ch;
    do
    {
        cout << "Placement Management Program" << endl;
        cout << "Enter 1 for Candidate" << endl;
        cout << "Enter 2 for Recruiter" << endl;
        cout << "Enter 3 for Coordinator" << endl;
        cout << "Enter 0 to exit" << endl;
        cin >> ch;
        switch (ch)
        {
            case 1:
                candMenuStarting();
                break;

            case 2:
                recmenuStarting();
                break;

            case 3:
                comenu();
                break;

            case 0:
                exit(0);

            default:
                cout << "Invalid Choice!" << endl;
        }
    } while (ch);

    return 0;
}
```

PROGRAM OUTPUT

Main Menu

```
Placement Management Program
Enter 1 for Candidate
Enter 2 for Recruiter
Enter 3 for Coordinator
Enter 0 to exit
_
```

Recruiter's Starting Menu Along With Signing Up of Companies

```
Placement Management Program
Enter 1 for Candidate
Enter 2 for Recruiter
Enter 3 for Coordinator
Enter 0 to exit
2
Enter 1 for Signing In:
Enter 2 for Signing Up:
Enter 3 to exit.
2
Enter company name : DESIS
Enter minimum CGPA : 8.5
Enter Job Title : SDE
Welcome to Recruiter Menu
Enter 1 for Updating Candidate:
Enter 2 for Displaying Candidate List:
Enter 0 for Exit:
0
Enter 1 for Signing In:
Enter 2 for Signing Up:
Enter 3 to exit.
2
Enter company name : TCS
Enter minimum CGPA : 7.5
Enter Job Title : Analyst
Welcome to Recruiter Menu
Enter 1 for Updating Candidate:
Enter 2 for Displaying Candidate List:
Enter 0 for Exit:
0
```

Candidate Starting Menu Along With Signing Up of Candidates

```
Placement Management Program
Enter 1 for Candidate
Enter 2 for Recruiter
Enter 3 for Coordinator
Enter 0 to exit
1
Enter 1 for Signing In:
Enter 2 for Signing Up:
Enter 3 to exit.
2
Enter name of a Candidate: Nandini
Enter roll number of Candidate: 13
Enter cgpa of Candidate: 9.5
Enter branch of Candidate: CSE
Welcome to the candidate dashboard
Enter 1 for viewing your present status with the various companies:
Enter 2 for registering for a new company:
Enter 0 for Exit:
0
Enter 1 for Signing In:
Enter 2 for Signing Up:
Enter 3 to exit.
2
Enter name of a Candidate: Priyanka
Enter roll number of Candidate: 24
Enter cgpa of Candidate: 9.35
Enter branch of Candidate: IT
Welcome to the candidate dashboard
Enter 1 for viewing your present status with the various companies:
Enter 2 for registering for a new company:
Enter 0 for Exit:
0
```

Candidate Registering For Company

```
Enter 1 for Signing In:
Enter 2 for Signing Up:
Enter 3 to exit.
1
Enter your name:
Nandini
Welcome to the candidate dashboard
Enter 1 for viewing your present status with the various companies:
Enter 2 for registering for a new company:
Enter 0 for Exit:
2
Index: 1 DESIS
Min CGPA required : 8.5
Job Profile: SDE
|
Index: 2 TCS
Min CGPA required : 7.5
Job Profile: Analyst

Enter company's index to register:
1
Registered!
Press -1 is you don't want to register for anymore companies
2
Registered!
Press -1 is you don't want to register for anymore companies
-1
```

Candidate's Registered Companies List

```
Welcome to the candidate dashboard
Enter 1 for viewing your present status with the various companies:
Enter 2 for registering for a new company:
Enter 0 for Exit:
1
DESI -> Registered
TCS -> Registered
Welcome to the candidate dashboard
Enter 1 for viewing your present status with the various companies:
Enter 2 for registering for a new company:
Enter 0 for Exit:
0
```

Company's List of Registered Candidates

```
Placement Management Program
Enter 1 for Candidate
Enter 2 for Recruiter
Enter 3 for Coordinator
Enter 0 to exit
2
Enter 1 for Signing In:
Enter 2 for Signing Up:
Enter 3 to exit.
1
Enter your name:
TCS
Welcome to Recruiter Menu
Enter 1 for Updating Candidate:
Enter 2 for Displaying Candidate List:
Enter 0 for Exit:
2

Nandini
Priyanka
```

Company Updating its Registered Candidates List

```
Welcome to Recruiter Menu
Enter 1 for Updating Candidate:
Enter 2 for Displaying Candidate List:
Enter 0 for Exit:
1
Name of Candidate : Nandini
press R for 'Rejected' and S for 'Selected for next round' :R

Name of Candidate : Priyanka
press R for 'Rejected' and S for 'Selected for next round' :S

Welcome to Recruiter Menu
Enter 1 for Updating Candidate:
Enter 2 for Displaying Candidate List:
Enter 0 for Exit:
2|

Priyanka
```


Candidates' Updated Lists

```
Enter 1 for Signing In:
Enter 2 for Signing Up:
Enter 3 to exit.
1
Enter your name:
Nandini
Welcome to the candidate dashboard
Enter 1 for viewing your present status with the various companies:
Enter 2 for registering for a new company:
Enter 0 for Exit:
1
DESI -> Selected
TCS -> Registered
Welcome to the candidate dashboard
Enter 1 for viewing your present status with the various companies:
Enter 2 for registering for a new company:
Enter 0 for Exit:
0
Enter 1 for Signing In:
Enter 2 for Signing Up: |
Enter 3 to exit.
1
Enter your name:
Priyanka
Welcome to the candidate dashboard
Enter 1 for viewing your present status with the various companies:
Enter 2 for registering for a new company:
Enter 0 for Exit:
1
DESI -> Selected
TCS -> Selected
```

Coordinator Displaying Companies List

```
Placement Management Program
Enter 1 for Candidate
Enter 2 for Recruiter
Enter 3 for Coordinator
Enter 0 to exit
3
Welcome to Coordinator Menu
Enter 1 for displaying companies lists:
Enter 2 for displaying students lists:
Enter 0 to exit.
1
Index: 1 DESIS
Min CGPA required : 8.5
Job Profile: SDE

Index: 2 TCS
Min CGPA required : 7.5
Job Profile: Analyst
```

Coordinator Displaying Candidates List

```
Welcome to Coordinator Menu
Enter 1 for displaying companies lists:
Enter 2 for displaying students lists:
Enter 0 to exit.
2
Name : Nandini
Roll No.: 13
CGPA: 9.5
Branch : CSE

Name : Priyanka
Roll No.: 24
CGPA: 9.35
Branch : IT
```