

NATIONAL INSTITUTE OF TECHNOLOGY, SILCHAR

Dept:- Computer Science And Engineering



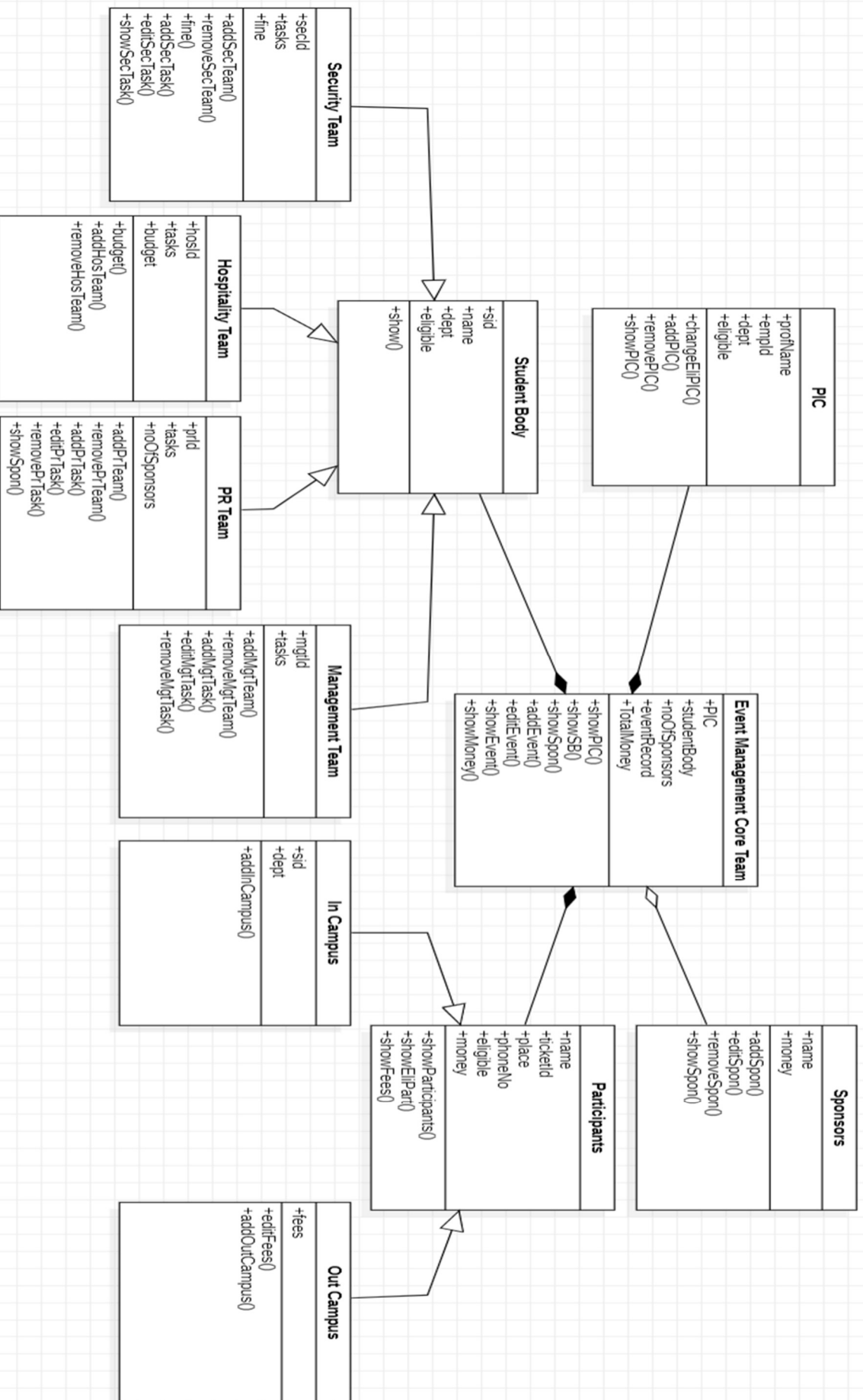
Object Oriented Design Lab

Event Management System



CS-315

Abhishek Ranjan 2112016 CSE 6th Sem



CODE

```
#include <bits/stdc++.h>
#include <conio.h>
#define nn "\n"
#define ll long long
#define pb push_back
#define pll pair<ll, ll>
#define vll vector<ll>
#define all(_obj) _obj.begin(), _obj.end()
#define fr(x, y) for (ll x = 0; x < y; x++)
#define nach_basanti \
    ios_base::sync_with_stdio(false); \
    cin.tie(0); \
    cout.tie(0);
#define inp(x) \
    ll x; \
    cin >> x;
#define inpv(v, n) \
    vll v(n); \
    fr(i, n) cin >> v[i];
#define printv(v) fr(i, v.size()) cout << v[i] << " ";
#define yes cout << "YES"
#define no cout << "NO"
#define home return;
const ll mod = 1e9 + 7;
using namespace std;
ll tot_money = 0;
ll tot_fine = 0;
class StudentBody
{
public:
    string sid;
    string name;
    string dept;
    int eli;

    StudentBody(string sid, string name, string dept, int eli)
    {
        this->sid = sid;
        this->name = name;
        this->dept = dept;
        this->eli = eli;
    }

    void intro()
    {
        cout << "Scholar ID : " << sid << nn;
        cout << "Name : " << name << nn;
        cout << "Department : " << dept << nn;
        cout << "Eligibility : " << (eli ? "Yes" : "No") << nn;
    }
};
class SecurityTeam : public StudentBody
{
public:
    string seclid;
    SecurityTeam(string sid, string name, string dept, int eli, string seclid) :
    StudentBody(sid, name, dept, eli)
    {
        this->seclid = seclid;
    }
    void intro()
    {
        cout << "Scholar ID : " << sid << nn;
        cout << "Name : " << name << nn;
        cout << "Department : " << dept << nn;
        cout << "Eligibility : " << (eli ? "Yes" : "No") << nn;
        cout << "Security ID : " << seclid << nn;
    }
};
class HospitalityTeam : public StudentBody
{
```

```
public:
    string hosld;
    HospitalityTeam(string sid, string name, string dept, int eli, string hosld) :
    StudentBody(sid, name, dept, eli)
    {
        this->hosld = hosld;
    }
    void intro()
    {
        cout << "Scholar ID : " << sid << nn;
        cout << "Name : " << name << nn;
        cout << "Department : " << dept << nn;
        cout << "Eligibility : " << (eli ? "Yes" : "No") << nn;
        cout << "Hospitality ID : " << hosld << nn;
    }
};
class PRTeam : public StudentBody
{
public:
    string prld;
    PRTeam(string sid, string name, string dept, int eli, string prld) :
    StudentBody(sid, name, dept, eli)
    {
        this->prld = prld;
    }
    void intro()
    {
        cout << "Scholar ID : " << sid << nn;
        cout << "Name : " << name << nn;
        cout << "Department : " << dept << nn;
        cout << "Eligibility : " << (eli ? "Yes" : "No") << nn;
        cout << "PR ID : " << prld << nn;
    }
};
class ManagementTeam : public StudentBody
{
public:
    string mgtld;
    ManagementTeam(string sid, string name, string dept, int eli, string
mgtld) : StudentBody(sid, name, dept, eli)
    {
        this->mgtld = mgtld;
    }
    void intro()
    {
        cout << "Scholar ID : " << sid << nn;
        cout << "Name : " << name << nn;
        cout << "Department : " << dept << nn;
        cout << "Eligibility : " << (eli ? "Yes" : "No") << nn;
        cout << "Management ID : " << mgtld << nn;
    }
};
class Participant
{
public:
    string name;
    string tid;
    string place;
    string pno;
    int eli;
    int mon_fees;

    Participant(string name, string tid, string place, string pno, int eli, int
mon_fees)
    {
        this->name = name;
        this->tid = tid;
        this->place = place;
        this->pno = pno;
        this->eli = eli;
        this->mon_fees = mon_fees;
    }

    void intro()
    {
        cout << "Name : " << name << nn;
        cout << "Ticket No. : " << tid << nn;
```

```

        cout << "Place : " << place << nn;
        cout << "Phone No. : " << pno << nn;
        cout << "Eligibility : " << (eli ? "Yes" : "No") << nn;
        cout << "Fees Given : " << mon_fees << nn;
    }
};
class InParticipant : public Participant
{
public:
    string sid;
    string dept;
    InParticipant(string name, string sid, string dept, string tid, string place,
string pno, int eli, int mon_fees) : Participant(name, tid, place, pno, eli,
mon_fees)
    {
        this->sid = sid;
        this->dept = dept;
    }
    void intro()
    {
        cout << "Name : " << name << nn;
        cout << "Scholar ID : " << sid << nn;
        cout << "Department : " << dept << nn;
        cout << "Ticket No. : " << tid << nn;
        cout << "Place : " << place << nn;
        cout << "Phone No. : " << pno << nn;
        cout << "Eligibility : " << (eli ? "Yes" : "No") << nn;
        cout << "Fees Given : " << mon_fees << nn;
    }
};
class OutParticipant : public Participant
{
public:
    int fees;
    OutParticipant(string name, int fees, string tid, string place, string pno, int
eli, int mon_fees) : Participant(name, tid, place, pno, eli, mon_fees)
    {
        this->fees = fees;
    }
    void intro()
    {
        cout << "Name : " << name << nn;
        cout << "Extraa fees collected : " << fees << nn;
        cout << "Ticket No. : " << tid << nn;
        cout << "Place : " << place << nn;
        cout << "Phone No. : " << pno << nn;
        cout << "Eligibility : " << (eli ? "Yes" : "No") << nn;
        cout << "Fees Given : " << mon_fees << nn;
    }
};
class Prof
{
public:
    string name;
    string dept;
    int eli;
    Prof(string name, string dept, int eli)
    {
        this->name = name;
        this->dept = dept;
        this->eli = eli;
    }
    void intro()
    {
        cout << "Name : " << name << nn;
        cout << "Department : " << dept << nn;
        cout << "Eligible : " << (eli == 0 ? "No" : "Yes") << nn;
    }
    void changeEli()
    {
        eli = !eli;
    }
};
class Sponsors
{
public:
    string name;

```

```

    ll money;
    Sponsors(string name, ll money)
    {
        this->name = name;
        this->money = money;
    }
    void intro()
    {
        cout << "Name : " << name << nn;
        cout << "Money Invested : " << money << nn;
    }
};

vector<Prof> profInCharge;
map<string, vector<string>> eventRecord;
vector<Sponsors> paisa;
vector<InParticipant> andar_wale;
vector<OutParticipant> bahar_wale;

vector<SecurityTeam> sec_team_wale;
vector<PRTeam> pr_team_wale;
vector<HospitalityTeam> hos_team_wale;
vector<ManagementTeam> mgt_team_wale;

map<string, vector<string>> mgtTask;
map<string, vector<string>> secTask;
map<string, vector<string>> prTask;

void decor()
{
    cout << "*****" << nn;
}
void decor_choice()
{
    cout << "Your Choice : ";
}
// Event MGT Part
void showPIC()
{
    if (profInCharge.size())
    {
        decor();
        cout << "The PICs are -" << endl;
        decor();
        for (auto &it : profInCharge)
        {
            it.intro();
            decor();
        }
    }
    else
    {
        cout << "No PICs";
    }

    getch();
}

void showSB()
{
    decor();
    cout << "Student Body Members : " << nn;
    decor();
    cout << nn;
    cout << "Security Team" << nn;
    decor();
    if (sec_team_wale.size())
    {
        for (auto &it : sec_team_wale)
        {
            it.intro();
            decor();
        }
        decor();
    }
    else
    {

```

```

        cout << "No Members" << nn;
    }
    decor();
    cout << "Management Team" << nn;
    decor();
    if (mgt_team_wale.size())
    {
        for (auto &it : mgt_team_wale)
        {
            it.intro();
            decor();
        }
        decor();
    }
    else
    {
        cout << "No Members" << nn;
    }
    decor();
    cout << "Hospitality Team" << nn;
    decor();
    if (hos_team_wale.size())
    {
        for (auto &it : hos_team_wale)
        {
            it.intro();
            decor();
        }
        decor();
    }
    else
    {
        cout << "No Members" << nn;
    }
    decor();
    cout << "PR Team" << nn;
    decor();
    if (pr_team_wale.size())
    {
        for (auto &it : pr_team_wale)
        {
            it.intro();
            decor();
        }
        decor();
    }
    else
    {
        cout << "No Members" << nn;
    }
    decor();
    getch();
}

void showSpon()
{
    ll sum = 0;
    if (paisa.size())
    {
        decor();
        cout << "The Sponsors are -" << endl;
        decor();
        for (auto &it : paisa)
        {
            it.intro();
            sum += it.money;
            decor();
        }
        cout << "Total Money Accumulated : " << sum << nn;
        decor();
    }
    else
    {
        cout << "No Sponsors";
    }
    getch();
}

```

```

void addEvent()
{
    string date, event;
    decor();
    cout << "Enter Date : ";
    fflush(stdin);
    getline(cin, date);
    cout << "Enter Event : ";
    fflush(stdin);
    getline(cin, event);
    decor();
    eventRecord[date].push_back(event);
}

void editEvent()
{
    decor();
    cout << "The Events are -" << endl;
    decor();
    for (auto &it : eventRecord)
    {
        cout << it.first << " : " << nn;
        for (auto &i : it.second)
        {
            cout << i << endl;
        }
        decor();
    }
    cout << "Enter date to edit : ";
    string date, event;
    fflush(stdin);
    getline(cin, date);

    if (eventRecord.find(date) == eventRecord.end())
    {
        cout << "Date is not present in the events";
    }
    else
    {
        cout << "Enter Event : ";
        fflush(stdin);
        getline(cin, event);
        eventRecord[date].clear();
        eventRecord[date].push_back(event);
        cout << "Event Edited Successfully" << nn;
    }
    getch();
}

void showEvent()
{
    if (eventRecord.size())
    {
        decor();
        cout << "The Events are -" << endl;
        decor();
        for (auto &it : eventRecord)
        {
            cout << it.first << " : " << nn;
            for (auto &i : it.second)
            {
                cout << i << endl;
            }
            decor();
        }
    }
    else
    {
        cout << "No Events";
    }

    getch();
}

void eventMgtCore()
{
    while (1)
    {
        system("cls");
    }
}

```

```

decor();
cout << "Event Management Core Team : " << nn;
decor();
cout << "1. Show PIC" << nn;
cout << "2. Show Student Body Members" << nn;
cout << "3. Show Sponsors" << nn;
cout << "4. Add Events" << nn;
cout << "5. Edit Events" << nn;
cout << "6. Show Events" << nn;
cout << "7. Go Back to Homepage" << nn;
decor_choice();

inp(ch);
switch (ch)
{
case 1:
    showPIC();
    break;
case 2:
    showSB();
    break;
case 3:
    showSpon();
    break;
case 4:
    addEvent();
    break;
case 5:
    editEvent();
    break;
case 6:
    showEvent();
    break;
case 7:
    cout << "Going to Homepage..." << nn;
    home;
    break;
}
}
// PIC part
void addPIC()
{
    string name, dept;
    int eli;
    decor();
    cout << "Enter Name : ";
    cin >> name;
    cout << "Enter Department : ";
    cin >> dept;
    cout << "Enter Eligibility : ";
    cin >> eli;
    decor();
    Prof p(name, dept, eli);
    profInCharge.push_back(p);
}
void changeEliPIC()
{
    if (!profInCharge.size())
    {
        cout << "No PICs" << endl;
        return;
    }
    ll choose;
    cout << "The PICs are -" << endl;
    decor();
    ll i = 1;
    for (auto &it : profInCharge)
    {
        cout << i << ". ";
        i++;
        it.intro();
        decor();
    }
    cout << "Enter Choice of Prof. to change Eligibility : ";
    cin >> choose;
    choose--;

```

```

    profInCharge[choose].changeEli();
}
void removePIC()
{
    if (!profInCharge.size())
    {
        cout << "No PICs" << endl;
        return;
    }
    ll choose;
    cout << "The PICs are -" << endl;
    decor();
    ll i = 1;
    for (auto &it : profInCharge)
    {
        cout << i << ". ";
        i++;
        it.intro();
        decor();
    }
    cout << "Enter Choice of Prof. to remove : ";
    cin >> choose;
    choose--;
    vector<Prof> temp;
    for (int i = 0; i < profInCharge.size(); i++)
    {
        if (i == choose)
            continue;
        temp.push_back(profInCharge[i]);
    }
    profInCharge = temp;
}
void pic()
{
    while (1)
    {
        system("cls");
        decor();
        cout << "Professor In Charge : " << nn;
        decor();
        cout << "1. Add PIC" << nn;
        cout << "2. Change Eligibility" << nn;
        cout << "3. Remove PIC" << nn;
        cout << "4. Show PIC" << nn;
        cout << "5. Go Back to Homepage" << nn;
        decor_choice();

        inp(ch);
        switch (ch)
        {
        case 1:
            addPIC();
            break;
        case 2:
            changeEliPIC();
            break;
        case 3:
            removePIC();
            break;
        case 4:
            showPIC();
            break;
        case 5:
            cout << "Going to Homepage..." << nn;
            home;
            break;
        }
    }
}

// Student Body Part
// Security Team
void addSecTeam()
{
    string sid;
    string name;
    string dept;

```

```

int eli;
string secId;

decor();
cout << "Enter Scholar ID : ";
cin >> sid;
cout << "Enter Name : ";
cin >> name;
cout << "Enter Department : ";
cin >> dept;
cout << "Enter Eligibility : ";
cin >> eli;
cout << "Enter Security Id : ";
cin >> secId;
decor();
SecurityTeam sec_team(sid, name, dept, eli, secId);
sec_team_wale.push_back(sec_team);
}
void removeSecTeam()
{
    if (!sec_team_wale.size())
    {
        cout << "No Security Team members" << endl;
        return;
    }
    // choose;
    cout << "The Security Team members are -" << endl;
    decor();
    // i = 1;
    for (auto &it : sec_team_wale)
    {
        cout << i << ". ";
        i++;
        it.intro();
        decor();
    }
    cout << "Enter Choice of Member to remove : ";
    cin >> choose;
    choose--;
    vector<SecurityTeam> temp;
    for (int i = 0; i < sec_team_wale.size(); i++)
    {
        if (i == choose)
            continue;
        temp.push_back(sec_team_wale[i]);
    }
    sec_team_wale = temp;
}
void fine()
{
    // fine;
    decor();
    cout << "Enter Fine to be collected : ";
    cin >> fine;
    tot_money += fine;
    cout << "Fine collected : " << fine;
    cout << nn;
    cout << "Total Money : " << tot_money;
    decor();
}
void addSecTask()
{
    string date, task;
    decor();
    cout << "Enter Date : ";
    fflush(stdin);
    getline(cin, date);
    cout << "Enter Task : ";
    fflush(stdin);
    getline(cin, task);
    decor();
    secTask[date].push_back(task);
}
void editSecTask()
{
    decor();
    cout << "The Security Tasks are -" << endl;

```

```

decor();
for (auto &it : secTask)
{
    cout << it.first << " : " << nn;
    for (auto &i : it.second)
    {
        cout << i << endl;
    }
    decor();
}
cout << "Enter date to edit : ";
string date, event;
fflush(stdin);
getline(cin, date);

if (secTask.find(date) == secTask.end())
{
    cout << "Date is not present in the events";
}
else
{
    cout << "Enter Task : ";
    fflush(stdin);
    getline(cin, event);
    secTask[date].clear();
    secTask[date].push_back(event);
    cout << "Task Edited Successfully" << nn;
}
getch();
}
void showSecTask()
{
    decor();
    cout << "The Security Tasks are -" << endl;
    decor();
    for (auto &it : secTask)
    {
        cout << it.first << " : " << nn;
        for (auto &i : it.second)
        {
            cout << i << endl;
        }
        decor();
    }
    decor();
    getch();
}
void secTeam()
{
    while (1)
    {
        system("cls");
        decor();
        cout << "Security Team : " << nn;
        decor();
        cout << "1. Add Team Member" << nn;
        cout << "2. Remove Team Member" << nn;
        cout << "3. Fine" << nn;
        cout << "4. Add Task" << nn;
        cout << "5. Edit Task" << nn;
        cout << "6. Show Task" << nn;
        cout << "7. Go to Student Body" << nn;
        decor_choice();

        inp(ch);
        switch (ch)
        {
            case 1:
                addSecTeam();
                break;
            case 2:
                removeSecTeam();
                break;
            case 3:
                fine();
                break;
            case 4:

```

```

        addSecTask();
        break;
    case 5:
        editSecTask();
        break;
    case 6:
        showSecTask();
        break;
    case 7:
        cout << "Going to Student Body..." << nn;
        home;
        break;
    }
}
}
// Hos Team
void addHosTeam()
{
    string sid;
    string name;
    string dept;
    int eli;
    string hosld;

    decor();
    cout << "Enter Scholar ID : ";
    cin >> sid;
    cout << "Enter Name : ";
    cin >> name;
    cout << "Enter Department : ";
    cin >> dept;
    cout << "Enter Eligibility : ";
    cin >> eli;
    cout << "Enter Hospitality Id : ";
    cin >> hosld;
    decor();
    HospitalityTeam hos_team(sid, name, dept, eli, hosld);
    hos_team_wale.push_back(hos_team);
}
void removeHosTeam()
{
    if (!hos_team_wale.size())
    {
        cout << "No Hospitality Team members" << endl;
        return;
    }
    // choose;
    cout << "The Hospitality Team members are -" << endl;
    decor();
    // i = 1;
    for (auto &it : hos_team_wale)
    {
        cout << i << ". ";
        i++;
        it.intro();
        decor();
    }
    cout << "Enter Choice of Member to remove : ";
    cin >> choose;
    choose--;
    vector<HospitalityTeam> temp;
    for (int i = 0; i < hos_team_wale.size(); i++)
    {
        if (i == choose)
            continue;
        temp.push_back(hos_team_wale[i]);
    }
    hos_team_wale = temp;
}
void budget()
{
    decor();
    cout << "Enter the Budget : ";
    // budget;
    cin >> budget;
    tot_money += budget;
    cout << "Total Money : " << tot_money << nn;
}

```

```

    decor();
}
void hosTeam()
{
    while (1)
    {
        system("cls");
        decor();
        cout << "Hospitality Team : " << nn;
        decor();
        cout << "1. Add Team Member" << nn;
        cout << "2. Remove Team Member" << nn;
        cout << "3. Budget" << nn;
        cout << "4. Go to Student Body" << nn;
        decor_choice();

        inp(ch);
        switch (ch)
        {
            case 1:
                addHosTeam();
                break;
            case 2:
                removeHosTeam();
                break;
            case 3:
                budget();
                break;
            case 4:
                cout << "Going to Student Body..." << nn;
                home;
                break;
        }
    }
}
// Mgt Team
void addMgtTeam()
{
    string sid;
    string name;
    string dept;
    int eli;
    string mgtld;

    decor();
    cout << "Enter Scholar ID : ";
    cin >> sid;
    cout << "Enter Name : ";
    cin >> name;
    cout << "Enter Department : ";
    cin >> dept;
    cout << "Enter Eligibility : ";
    cin >> eli;
    cout << "Enter Management Id : ";
    cin >> mgtld;
    decor();
    ManagementTeam mg_team(sid, name, dept, eli, mgtld);
    mgt_team_wale.push_back(mg_team);
}
void removeMgtTeam()
{
    if (!mgt_team_wale.size())
    {
        cout << "No Management Team members" << endl;
        return;
    }
    // choose;
    cout << "The Management Team members are -" << endl;
    decor();
    // i = 1;
    for (auto &it : mgt_team_wale)
    {
        cout << i << ". ";
        i++;
        it.intro();
        decor();
    }
}

```



```

cout << "Enter Choice of Member to remove : ";
cin >> choose;
choose--;
vector<ManagementTeam> temp;
for (int i = 0; i < mgt_team_wale.size(); i++)
{
    if (i == choose)
        continue;
    temp.push_back(mgt_team_wale[i]);
}
mgt_team_wale = temp;
}
void addMgtTask()
{
    string date, task;
    decor();
    cout << "Enter Date : ";
    fflush(stdin);
    getline(cin, date);
    cout << "Enter Task : ";
    fflush(stdin);
    getline(cin, task);
    decor();
    mgtTask[date].push_back(task);
}
void editMgtTask()
{
    decor();
    cout << "The Management Tasks are -" << endl;
    decor();
    for (auto &it : mgtTask)
    {
        cout << it.first << " : " << nn;
        for (auto &i : it.second)
        {
            cout << i << endl;
        }
        decor();
    }
    cout << "Enter date to edit : ";
    string date, event;
    fflush(stdin);
    getline(cin, date);

    if (mgtTask.find(date) == mgtTask.end())
    {
        cout << "Date is not present in the events";
    }
    else
    {
        cout << "Enter Task : ";
        fflush(stdin);
        getline(cin, event);
        mgtTask[date].clear();
        mgtTask[date].push_back(event);
        cout << "Task Edited Successfully" << nn;
    }
    getch();
}
void removeMgtTask()
{
    decor();
    cout << "The Management Tasks are -" << endl;
    decor();
    for (auto &it : mgtTask)
    {
        cout << it.first << " : " << nn;
        for (auto &i : it.second)
        {
            cout << i << endl;
        }
        decor();
    }
    cout << "Enter date to remove : ";
    string date, event;
    fflush(stdin);
    getline(cin, date);

```

```

if (mgtTask.find(date) == mgtTask.end())
{
    cout << "Date is not present in the events";
}
else
{
    mgtTask[date].clear();
    cout << "Task Removed Successfully" << nn;
}
getch();
}
void mgtTeam()
{
    while (1)
    {
        system("cls");
        decor();
        cout << "Management Team : " << nn;
        decor();
        cout << "1. Add Team Member" << nn;
        cout << "2. Remove Team Member" << nn;
        cout << "3. Add Task" << nn;
        cout << "4. Edit Task" << nn;
        cout << "5. Remove Task" << nn;
        cout << "6. Go to Student Body" << nn;
        decor_choice();

        inp(ch);
        switch (ch)
        {
            case 1:
                addMgtTeam();
                break;
            case 2:
                removeMgtTeam();
                break;
            case 3:
                addMgtTask();
                break;
            case 4:
                editMgtTask();
                break;
            case 5:
                removeMgtTask();
                break;
            case 6:
                cout << "Going to Student Body..." << nn;
                home;
                break;
        }
    }
}
// PR Team
void addPrTeam()
{
    string sid;
    string name;
    string dept;
    int eli;
    string prld;

    decor();
    cout << "Enter Scholar ID : ";
    cin >> sid;
    cout << "Enter Name : ";
    cin >> name;
    cout << "Enter Department : ";
    cin >> dept;
    cout << "Enter Eligibility : ";
    cin >> eli;
    cout << "Enter PR Id : ";
    cin >> prld;
    decor();
    PRTeam pr_team(sid, name, dept, eli, prld);
    pr_team_wale.push_back(pr_team);
}

```

```

void removePrTeam()
{
    if (!pr_team_wale.size())
    {
        cout << "No PR Team members" << endl;
        return;
    }
    ll choose;
    cout << "The PR Team members are -" << endl;
    decor();
    ll i = 1;
    for (auto &it : pr_team_wale)
    {
        cout << i << ". ";
        i++;
        it.intro();
        decor();
    }
    cout << "Enter Choice of Member to remove : ";
    cin >> choose;
    choose--;
    vector<PRTeam> temp;
    for (int i = 0; i < pr_team_wale.size(); i++)
    {
        if (i == choose)
            continue;
        temp.push_back(pr_team_wale[i]);
    }
    pr_team_wale = temp;
}

void addPrTask()
{
    string date, task;
    decor();
    cout << "Enter Date : ";
    fflush(stdin);
    getline(cin, date);
    cout << "Enter Task : ";
    fflush(stdin);
    getline(cin, task);
    decor();
    prTask[date].push_back(task);
}

void editPrTask()
{
    decor();
    cout << "The PR Tasks are -" << endl;
    decor();
    for (auto &it : prTask)
    {
        cout << it.first << " : " << nn;
        for (auto &i : it.second)
        {
            cout << i << endl;
        }
        decor();
    }
    cout << "Enter date to edit : ";
    string date, event;
    fflush(stdin);
    getline(cin, date);

    if (prTask.find(date) == prTask.end())
    {
        cout << "Date is not present in the events";
    }
    else
    {
        cout << "Enter Task : ";
        fflush(stdin);
        getline(cin, event);
        prTask[date].clear();
        prTask[date].push_back(event);
        cout << "Task Edited Successfully" << nn;
    }
    getch();
}

```

```

void removePrTask()
{
    decor();
    cout << "The PR Tasks are -" << endl;
    decor();
    for (auto &it : prTask)
    {
        cout << it.first << " : " << nn;
        for (auto &i : it.second)
        {
            cout << i << endl;
        }
        decor();
    }
    cout << "Enter date to remove : ";
    string date, event;
    fflush(stdin);
    getline(cin, date);

    if (prTask.find(date) == prTask.end())
    {
        cout << "Date is not present in the events";
    }
    else
    {
        prTask[date].clear();
        cout << "Task Removed Successfully" << nn;
    }
    getch();
}

void prTeam()
{
    while (1)
    {
        system("cls");
        decor();
        cout << "PR Team : " << nn;
        decor();
        cout << "1. Add Team Member" << nn;
        cout << "2. Remove Team Member" << nn;
        cout << "3. Add Task" << nn;
        cout << "4. Edit Task" << nn;
        cout << "5. Remove Task" << nn;
        cout << "6. Show Sponsors" << nn;
        cout << "7. Go to Student Body" << nn;
        decor_choice();

        inp(ch);
        switch (ch)
        {
            case 1:
                addPrTeam();
                break;
            case 2:
                removePrTeam();
                break;
            case 3:
                addPrTask();
                break;
            case 4:
                editPrTask();
                break;
            case 5:
                removePrTask();
                break;
            case 6:
                showSpon();
                break;
            case 7:
                cout << "Going to Student Body..." << nn;
                home;
                break;
        }
    }
}

void studentBody()
{

```

```

while (1)
{
    system("cls");
    decor();
    cout << "Student Body : " << nn;
    decor();
    cout << "1. Security Team" << nn;
    cout << "2. Hospitality Team" << nn;
    cout << "3. Management Team" << nn;
    cout << "4. PR Team" << nn;
    cout << "5. Go Back to Homepage" << nn;
    decor_choice();

    inp(ch);
    switch (ch)
    {
        case 1:
            secTeam();
            break;
        case 2:
            hosTeam();
            break;
        case 3:
            mgtTeam();
            break;
        case 4:
            prTeam();
            break;
        case 5:
            cout << "Going to Homepage..." << nn;
            home;
            break;
    }
}

// Sponsors Part
void addSpon()
{
    string name;
    ll money;
    decor();
    cout << "Enter Name : ";
    cin >> name;
    cout << "Enter Money : ";
    cin >> money;
    decor();
    Sponsors s(name, money);
    paisa.push_back(s);
}

void editSpon()
{
    if (!paisa.size())
    {
        cout << "No Sponsors" << endl;
        return;
    }
    ll choose, money;
    cout << "The Sponsors are -" << endl;
    decor();
    ll i = 1;
    for (auto &it : paisa)
    {
        cout << i << ". ";
        i++;
        it.intro();
        decor();
    }
    cout << "Enter Choice of Sponsor to change money : ";
    cin >> choose;
    cout << "Enter Money : ";
    cin >> money;
    choose--;
    paisa[choose].money = money;
}

void removeSpon()
{

```

```

if (!paisa.size())
{
    cout << "No Sponsors" << endl;
    return;
}
ll choose;
cout << "The Sponsors are -" << endl;
decor();
ll i = 1;
for (auto &it : paisa)
{
    cout << i << ". ";
    i++;
    it.intro();
    decor();
}
cout << "Enter Choice of Sponsor to remove : ";
cin >> choose;
choose--;
vector<Sponsors> temp;
for (int i = 0; i < paisa.size(); i++)
{
    if (i == choose)
        continue;
    temp.push_back(paisa[i]);
}
paisa = temp;
}

void sponsors()
{
    while (1)
    {
        system("cls");
        decor();
        cout << "Sponsors : " << nn;
        decor();
        cout << "1. Add Sponsors" << nn;
        cout << "2. Edit Sponsors" << nn;
        cout << "3. Remove Sponsors" << nn;
        cout << "4. Show Sponsors" << nn;
        cout << "5. Go Back to Homepage" << nn;
        decor_choice();

        inp(ch);
        switch (ch)
        {
            case 1:
                addSpon();
                break;
            case 2:
                editSpon();
                break;
            case 3:
                removeSpon();
                break;
            case 4:
                showSpon();
                break;
            case 5:
                cout << "Going to Homepage..." << nn;
                home;
                break;
        }
    }
}

// Participants Part
void addInCampus()
{
    string name;
    string sid;
    string dept;
    string tid;
    string place;
    string pno;
    int eli;
    int mon_fees;

```

```

decor();
cout << "Enter Name : ";
cin >> name;
cout << "Enter Scholar ID : ";
cin >> sid;
cout << "Enter Department : ";
cin >> dept;
cout << "Enter Ticket ID : ";
cin >> tid;
cout << "Enter Place : ";
cin >> place;
cout << "Enter Phone No. : ";
cin >> pno;
cout << "Enter Eligibility : ";
cin >> eli;
cout << "Enter Primary Fees : ";
cin >> mon_fees;
decor();
InParticipant ip(name, sid, dept, tid, place, pno, eli, mon_fees);
andar_wale.push_back(ip);
}
void addOutCampus()
{
    string name;
    int extra_fee;
    string tid;
    string place;
    string pno;
    int eli;
    int mon_fees;

    decor();
    cout << "Enter Name : ";
    cin >> name;
    cout << "Enter Extra Fees : ";
    cin >> extra_fee;
    cout << "Enter Ticket ID : ";
    cin >> tid;
    cout << "Enter Place : ";
    cin >> place;
    cout << "Enter Phone No. : ";
    cin >> pno;
    cout << "Enter Eligibility : ";
    cin >> eli;
    cout << "Enter Primary Fees : ";
    cin >> mon_fees;
    decor();
    OutParticipant op(name, extra_fee, tid, place, pno, eli, mon_fees);
    bahar_wale.push_back(op);
}
void showParticipants()
{
    if (andar_wale.size())
    {
        decor();
        cout << "The Insiders are -" << endl;
        decor();
        for (auto &it : andar_wale)
        {
            it.intro();
            decor();
        }
    }
    else
    {
        cout << "No Insiders" << nn;
        decor();
    }
    if (bahar_wale.size())
    {
        decor();
        cout << "The Outsiders are -" << endl;
        decor();
        for (auto &it : bahar_wale)
        {
            it.intro();

```

```

            decor();
        }
    }
    else
    {
        cout << "No Outsiders" << nn;
        decor();
    }
    getch();
}
void showEliPart()
{
    if (andar_wale.size())
    {
        decor();
        cout << "The Insiders are -" << endl;
        decor();
        for (auto &it : andar_wale)
        {
            if (!it.eli)
                continue;
            it.intro();
            decor();
        }
    }
    else
    {
        cout << "No Insiders" << nn;
        decor();
    }
    if (bahar_wale.size())
    {
        decor();
        cout << "The Outsiders are -" << endl;
        decor();
        for (auto &it : bahar_wale)
        {
            if (!it.eli)
                continue;
            it.intro();
            decor();
        }
    }
    else
    {
        cout << "No Outsiders" << nn;
        decor();
    }
    getch();
}
void showFees()
{
    int sum1 = 0, sum2 = 0;
    int extra = 0;
    if (andar_wale.size())
    {
        decor();
        cout << "The Insiders are -" << endl;
        decor();
        for (auto &it : andar_wale)
        {
            if (!it.eli)
                continue;
            it.intro();
            decor();
            sum1 += it.mon_fees;
        }
    }
    else
    {
        cout << "No Insiders" << nn;
        decor();
    }
    cout << "Fees Collected from Insiders : " << sum1 << nn;
    decor();
    if (bahar_wale.size())
    {

```

```

decor();
cout << "The Outsiders are -" << endl;
decor();
for (auto &it : bahar_wale)
{
    if (!it.eli)
        continue;
    it.intro();
    decor();
    sum2 += it.mon_fees;
    extra += it.fees;
}
}
else
{
    cout << "No Outsiders" << nn;
    decor();
}
cout << "Fees Collected from Outsiders and extra fees : " << sum2 << " & "
<< extra << nn;
decor();
tot_money = sum1 + sum2 + extra;
getch();
}
void participants()
{
    while (1)
    {
        system("cls");
        decor();
        cout << "Participants : " << nn;
        decor();
        cout << "1. Add In Campus Participants" << nn;
        cout << "2. Add Out Campus Participants" << nn;
        cout << "3. Show Participants" << nn;
        cout << "4. Show Eligible Participants" << nn;
        cout << "5. Show Fees Generated" << nn;
        cout << "6. Go Back to Homepage" << nn;
        decor_choice();

        inp(ch);
        switch (ch)
        {
            case 1:
                addInCampus();
                break;
            case 2:
                addOutCampus();
                break;
            case 3:
                showParticipants();
                break;
            case 4:
                showEliPart();
                break;
            case 5:
                showFees();
                break;
            case 6:
                cout << "Going to Homepage..." << nn;
                home;
                break;
        }
    }
}
void mainMenu()
{
    cout << "Choose appropriate option : " << nn;
    cout << "1. Event Management Core Team" << nn;
    cout << "2. Professor In Charge" << nn;
    cout << "3. Student Body" << nn;
    cout << "4. Sponsors" << nn;
    cout << "5. Participants" << nn;
    cout << "6. Total Money" << nn;
    cout << "7. To Exit" << nn;
    decor_choice();
}

```

```

void solve()
{
    system("cls");
    ll purah_money = 0;
    while (1)
    {
        system("cls");
        decor();
        cout << "Welcome to Event Management Software" << endl;
        decor();
        mainMenu();
        inp(ch);
        switch (ch)
        {
            case 1:
                eventMgtCore();
                break;
            case 2:
                pic();
                break;
            case 3:
                studentBody();
                break;
            case 4:
                sponsors();
                break;
            case 5:
                participants();
                break;
            case 6:
                for (auto &it : paisa)
                    purah_money += it.money;
                purah_money += tot_money;
                cout << "Total Money : " << purah_money << nn;
                getch();
                break;
            case 7:
                decor();
                cout << "Good Bye" << nn;
                decor();
                home;
                break;
        }
    }
}
int32_t main()
{
    solve();
    return 0;
}

```

OUTPUTS

// Starting Homepage

Welcome to Event Management Software

Choose appropriate option :

1. Event Management Core Team
2. Professor In Charge
3. Student Body
4. Sponsors
5. Participants
6. Total Money
7. To Exit

Your Choice : 1

//Showing Registered Events

Event Management Core Team :

1. Show PIC
2. Show Student Body Members
3. Show Sponsors
4. Add Events
5. Edit Events
6. Show Events
7. Go Back to Homepage

Your Choice : 6

The Events are -

5th april 24 :
dance krna hai

6th april 24 :
maggi khana hai

//Displaying Professors in charge of the event

Professor In Charge :

1. Add PIC
2. Change Eligibility
3. Remove PIC
4. Show PIC
5. Go Back to Homepage

Your Choice : 4

The PICs are -

Name : Raju

Department : CSE

Eligible : Yes

Name : Ramesh

Department : CSE

Eligible : Yes

Name : Sameer

Department : CSE

Eligible : Yes

Name : Rajiv

Department : CSE

Eligible : Yes

Name : Jehnab

Department : ME

Eligible : Yes

//Displaying the sponsors for the event

Sponsors :

1. Add Sponsors
2. Edit Sponsors
3. Remove Sponsors
4. Show Sponsors
5. Go Back to Homepage

Your Choice : 4

The Sponsors are -

Name : pepsi

Money Invested : 1000

Name : cococola

Money Invested : 10

Name : philips

Money Invested : 199

Name : citydhaba

Money Invested : 10

Total Money Accumulated : 1219

//Displaying the participants of the event

Participants :

1. Add In Campus Participants
2. Add Out Campus Participants
3. Show Participants
4. Show Eligible Participants
5. Show Fees Generated
6. Go Back to Homepage

Your Choice : 3

The Insiders are -

Name : Abhishek
Scholar ID : 2112016
Department : CSE
Ticket No. : 1001
Place : bihar
Phone No. : 11111
Eligibility : Yes
Fees Given : 1000

Name : Agraz
Scholar ID : 2112030
Department : CSE
Ticket No. : 1330
Place : bijni
Phone No. : 23233
Eligibility : Yes
Fees Given : 110

Name : Tushar
Scholar ID : 2112064
Department : CSE
Ticket No. : 1201
Place : tezpur
Phone No. : 32221
Eligibility : Yes
Fees Given : 10000

The Outsiders are -

Name : Akash
Extraa fees collected : 10000
Ticket No. : 13411
Place : dekhiajuli
Phone No. : 23454
Eligibility : Yes
Fees Given : 1000000

Name : Ramesh

Extraa fees collected : 100

Ticket No. : 12342

Place : assam

Phone No. : 34342

Eligibility : Yes

Fees Given : 1000

//Displaying the fees generated in total

Participants :

1. Add In Campus Participants
2. Add Out Campus Participants
3. Show Participants
4. Show Eligible Participants
5. Show Fees Generated
6. Go Back to Homepage

Your Choice : 5

The Insiders are -

Name : Abhishek
Scholar ID : 2112016
Department : CSE
Ticket No. : 1001
Place : bihar
Phone No. : 11111
Eligibility : Yes
Fees Given : 1000

Name : Agraz
Scholar ID : 2112030
Department : CSE
Ticket No. : 1330
Place : bijni
Phone No. : 23233
Eligibility : Yes
Fees Given : 110

Name : Tushar
Scholar ID : 2112064
Department : CSE
Ticket No. : 1201
Place : tezpur
Phone No. : 32221
Eligibility : Yes
Fees Given : 10000

Fees Collected from Insiders : 11110

The Outsiders are -

Name : Akash

Extraa fees collected : 10000

Ticket No. : 13411

Place : dekhiajuli

Phone No. : 23454

Eligibility : Yes

Fees Given : 1000000

Name : Ramesh

Extraa fees collected : 100

Ticket No. : 12342

Place : assam

Phone No. : 34342

Eligibility : Yes

Fees Given : 1000

Fees Collected from Outsiders and extra fees :

1001000 & 10100

//Displaying the Core Student Body Team members

Event Management Core Team :

1. Show PIC

2. Show Student Body Members

3. Show Sponsors

4. Add Events

5. Edit Events

6. Show Events

7. Go Back to Homepage

Your Choice : 2

Student Body Members :

Security Team

Scholar ID : 16

Name : Ranjan

Department : CSE

Eligibility : Yes

Security ID : 1016

Scholar ID : 18

Name : Toppo

Department : CSE

Eligibility : Yes

Security ID : 1018

Management Team

Scholar ID : 30

Name : Agraz

Department : CSE

Eligibility : Yes

Management ID : 3030

Scholar ID : 42

Name : Hirak

Department : CSE

Eligibility : Yes

Management ID : 3042

Scholar ID : 73

Name : Deepjyoti

Department : CSE

Eligibility : Yes

Management ID : 3073

Hospitality Team

Scholar ID : 20

Name : Ray

Department : CSE

Eligibility : Yes

Hospitality ID : 2020

Scholar ID : 23

Name : Rishant

Department : CSE

Eligibility : Yes

Hospitality ID : 2023

PR Team

Scholar ID : 14

Name : Priyajit

Department : CSE

Eligibility : Yes

PR ID : 4014

Scholar ID : 39

Name : Anupam

Department : CSE

Eligibility : Yes

PR ID : 4039
