**INDEX**

|  |  |  |
| --- | --- | --- |
| **No.** | **Contents** | **Page No.** |
| 1. | Introduction | 1 |
| 2. | Function Definition | 2 |
| 3. | Program Code | 7 |
| 4. | Input/Outputs | 61 |

**INTRODUCTION**

**SYSTEM DESCRIPTION**

This is a computerized college admission management system which primarily deals with admission and hostel procedures and stream and fee details.

This program checks whether the candidate applying for admission meets the requirements for the stream he chooses and only if he satisfies the requirement, the candidate is given admission. To apply for hostel facilities, the student must be a part of the college. Each student I allotted a room and is given a choice to select his diet.

Total fees is calculated with respect to his stream, quota, whether he is part of the hostel and on the basis of his diet preferences.

Deleting of a student record simultaneously delete him from the hostel.

The user is also allowed to add and modify streams and fee details.

**SELECTING STREAM DETAILS**

The user is given a choice of adding a new stream, viewing all the streams, displaying all the students enrolled in a particular stream, modifying the cut off mark or fee details of a particular stream and viewing fee details of the stream.

**SELECTING STUDENT DETAILS**

The user is given an option of adding and deleting a student, displaying all the student’s details and displaying a particular student’s details. On deleting a student’s record, his hostel records also gets deleted.

**SELECTING HOSTEL DETAILS**

The user is given the option of adding a hostelite details, modifying a hostelite’s details, displaying all hostelites details, displaying a particular hostelite’s details and deleting a hostelite’s record.

**FUNCTION DEFINITIONS**

1. chkseat() –

It takes parameter of type student. It compares the stream and quota of the student and checks if a seat is available in the respective quota in the mentioned stream. It returns 1 if seat is available otherwise returns 0

1. chckcutoff () –

It takes parameter of type student. It compares the stream and quota of the student and checks if the student’s percentage is above or equal to specified cut-off for the stream. If quota is SC/ST the their required cut-off is increased by 10. If the student satisfies the specified rank the function returns 1 otherwise 0

1. chkrank ()-

It takes a parameter of type student. It compares the stream and quota and checks whether the student’s AIEEE rank satisfies the specified cutoff rank. If the student satisfies the specified rank the function returns 1 otherwise 0.

1. modify ()-

It takes parametres of type student and a character. It compares the stream and modifies the number of seats occupied depending on the quota of the student. It increments the number of seats occupied in the specified quota if the parameter passed is ‘a’ otherwise decrements it.

1. Getfees ()-

It takes the stream and quota as parametres. It compares the stream and quota and returns the respective college fees.

1. Admn ()-

It takes the current year as parameter. It reads the name, admission number, AIEEE rank, class 12th grade percentage, the stream the student wishes to enrolling and year of birth. A student greater than 24 years will not be eligible for admission.

Thi function invokes functions chkseat(), chkrank() and chkcutoff() sending the required parameters. Only if all the functions returns a value 1, admission is granted. Function modify() is then invoked and the student’s details and character ‘a’ is sent as parameter. If any of the above functions returns value 0, the student is denied admission

1. Displayall ()-

This function displays all the student’s record in the file student.dat

1. Displstu ()-

This function displays the record of a particular student only.

1. Dispstr()-

This function takes the stream as a parameter and displays all the student’s records enrolled in this particular stream

1. Room\_modify()-

It takes the room number and the status if the room you wish to change as parameters. It compares the room number and changes the status of the specified room

1. Deletestudent()-

This function reads the admission number from the user and compares it to find the student’s record and deletes it. It also calls the hmodify() function to delete the student’s details from the hostel as well

1. Modifycutoff()-

This function reads the the stream from the user and compares it to find the stream’s record in the file streamchk.dat. It then modifies the cutoff percentage and rank to the values entered by the user

1. Addstream()-

This function adds new streams. It reads the stream name, total number of seats for the NRI, SC/ST, OBC and general quota, the cut off rank, percentage, and the fees for a general seat. It initializes the total number of occupied seats as 0 and calculates the fees for SC/ST seats as half of fees for general seats and fees for NRI seats as 4 times the fees for general fees

1. Showfeedet()-

This function displays all the fee details i.e. the stream name, fees for general, SC/ST, OBC and NRI seats

1. Modfeedet()-

This function compares the stream and modifies the fees as entered by the user

1. Displayseats()-

This function takes one’s admission number as a parameter and checks whether it exists and returns value 1 if it does otherwise returns 0

1. hcheckroom()-

This function reads records from the file rooms.dat and returns the room number of a vacant room.

1. writestudent()-

This function takes the student’s admission number and hostel fees as parameters. It writes the total fees and college fees.

1. hcount()-

This function returns the number of rooms occupied.

1. hread()-

This function displays all the students details in the hostel.

1. hdisplay()-

This function displays a particular student’s details in the hostel.

1. hmodify()-

This function invokes the function hcheckadmno() which checks whether the admission number entered by the user is valid or not. If valid, it modifies the diet preference as per the user’s choice and accordingly modifies the fees.

1. resetrooms()-

This function resets the status of all rooms to vacant.

1. hdelete()-

This function takes admission number as a parameter. It compares admission number and deletes the student’s details from the file hostel.dat. It also calls the function room\_modify() to reset the room status as vacant.

1. displayfee()-

This function takes admission as a parameter and displays the college fees, hostel fees and the total fees of the student if the admission number is entered valid.

1. streammenu()-

This function is called when the user wishes to view stream details. It displays the Stream Menu and invokes the required function as per the user’s choice.

Functions invoked are: addstream(), displayseat(), dispstr(), modifycutoff(), showfeedet(), modfeedet().

1. studentmenu()-

This function is called when the user wishes to view the student’s details.

It displays the student menu and invokes the required functions as per the user’s choice.

Functions invoked are: admn(), displayall(), displstu(), deletestudent() and displayfee().

1. hostelmenu()-

This function is called when the user wishes to view the hostel details. It displays the hostel menu and invokes the required functions as per the user’s choice.

Function invoked are: store(), hread(), hdisplay(), hmodify() and hdelete().

1. storeh()-

This function adds new student’s details for hostel to file hostel.dat . It calls function hcheckadmno() and sends the student’s admission number as a parameter. If it returns value 1, it then calls function hcheckroom(), which returns a room no of a vacant room. It reads the diet preference from the user and calculates mess fee as 1800 for veg. and 2000 for non-veg.

Hostel fee is 15000. Modify\_room() is invoked and the room number and status i.e, occupied is sent as parameters.

Total fees is calculated as the sum of mess fees and room fees.

1. intro()-

This function is called when the program first commences. It displays the name of the program and the names of the programmers.

Functions streammenu(), hostelmenu(), studentmenu(), resetrooms() and intro() are invoked from main

**Program code:**

#include<fstream>

#include<iostream>

#include<iomanip>

#include<stdlib.h>

#include<string.h>

#include<stdio.h>

#include<conio.h>

#include<windows.h>

#include<wincon.h>

#include<process.h>

using namespace std;

COORD coord={0,0};

void gotoxy(int x,int y)

{

coord.X=x;

coord.Y=y;

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE),coord);

}

struct feedata

{

char stream[25];

float NRI,SCST,OBC,general;

};

struct seats

{

char stream[25];

float NRI, SCST, OBC, general;

float tNRI,tSCST,tOBC,tgeneral;

float cutoff, rank;

feedata f;

};

struct student

{

long admno;

char name[50], quota[25], stream[25];

float rank,perc,cfees,hfees,tfees;

};

struct hostel

{

int admno,rno;

char food;

float mfees,rfees,tfees;

};

struct rooms

{

int rmno;

char status;

};

int chkseat(student st)

{

system("cls");

int found=0;

seats s;

ifstream fin("streamchk.dat",ios::binary);

while(fin.read((char\*)&s,sizeof(s)))

{

if(strcmpi(s.stream,st.stream)==0)

{

if(strcmpi(st.quota,"NRI")==0)

{

if(s.NRI<s.tNRI)

found=1;

}

else if(strcmpi(st.quota,"General")==0)

{

if(s.general<s.tgeneral)

found=1;}

else if(strcmpi(st.quota,"OBC")==0)

{

if(s.OBC<s.tOBC)

found=1;}

else if(strcmpi(st.quota,"SCST")==0)

{

if(s.SCST<s.tSCST)

found=1;}

}}

return found;

}

void gotoxytext(int x, int y, const char \*str)

{

gotoxy(x,y);

cout << str;

}

class CTextModeGrid

{

int m\_left, m\_top, m\_right, m\_bottom;

int m\_numRows, m\_numCols;

int m\_arrRowHeight[24], m\_arrColWidth[24];

public:

CTextModeGrid(int l, int t, int r, int b)

{

m\_left = l;

m\_top = t;

m\_right = r;

m\_bottom = b;

}

void SetRows(int rs) { m\_numRows = rs; }

void SetCols(int cs) { m\_numCols = cs; }

void SetRowHeight(int r, int rh) { m\_arrRowHeight[r] = rh; }

void SetColWidth(int c, int cw) { m\_arrColWidth[c] = cw; }

void DrawGrid();

void DrawSquare(int left, int top, int right, int bottom);

};

void CTextModeGrid::DrawGrid()

{

DrawSquare(m\_left, m\_top, m\_right, m\_bottom);

{

int pos = m\_left;

for(int i = 0; i < m\_numCols; i++)

{

pos += m\_arrColWidth[i];

if(pos > m\_right)

pos = m\_right;

for(int j = m\_top; j <= m\_bottom; j++)

{

gotoxy(pos, j);

cout << (char)(179);

}

if(pos == m\_left)

{

gotoxy(m\_left,m\_top);

cout << (char) (218);

gotoxy(m\_left,m\_bottom);

cout << (char) (192);

}

else if(pos == m\_right)

{

gotoxy(m\_right,m\_top);

cout << (char) (191);

gotoxy(m\_right,m\_bottom);

cout << (char) (217);

}

else

{

gotoxy(pos, m\_top);

cout << (char)(194);

gotoxy(pos, m\_bottom);

cout << (char)(193);

}

}

}

{

int pos = m\_top;

for(int i = 0; i < m\_numRows; i++)

{

pos += m\_arrRowHeight[i];

if(pos > m\_bottom)

pos = m\_bottom;

for(int j = m\_left; j <= m\_right; j++)

{

gotoxy(j, pos);

cout << (char)(196);

int xpos = m\_left;

for(int k = 0; k < m\_numCols; k++)

{

xpos += m\_arrColWidth[k];

if( xpos == j)

{

gotoxy(xpos, pos);

if(pos == m\_bottom)

{

cout << (char)(193);

}

else if(pos == m\_top)

{

cout << (char)(194);

}

else

{

cout << (char)(197);

}

}

}

}

if(pos == m\_top)

{

gotoxy(m\_left,m\_top);

cout << (char) (218);

gotoxy(m\_right,m\_top);

cout << (char) (191);

}

else if(pos == m\_bottom)

{

gotoxy(m\_left,m\_bottom);

cout << (char) (192);

gotoxy(m\_right,m\_bottom);

cout << (char) (217);

}

else

{

gotoxy(m\_left, pos);

cout << (char)(195);

gotoxy(m\_right, pos);

cout << (char)(180);

}

}

}

}

void CTextModeGrid::DrawSquare(int left, int top, int right, int bottom)

{

{

for(int i = left; i <= right; i++)

{

gotoxy(i,top);

cout << (char) (196);

gotoxy(i,bottom);

cout << (char) (196);

}

}

{

for(int i = top; i <= bottom; i++)

{

gotoxy(left,i);

cout << (char)(179);

gotoxy(right,i);

cout << (char)(179);

}

}

gotoxy(left,top);

cout << (char) (218);

gotoxy(right,top);

cout << (char) (191);

gotoxy(left,bottom);

cout << (char) (192);

gotoxy(right,bottom);

cout << (char) (217);

}

int chkcutoff(student st)

{

system("cls");

int found=0;

seats s;

ifstream fin("streamchk.dat",ios::binary);

while(fin.read((char\*)&s,sizeof(s)))

{

if(strcmpi(st.stream,s.stream)==0)

{

if(strcmpi(st.quota,"General")==0)

{

if(st.perc>=s.cutoff)

found=1;

}

else if(strcmpi(st.quota,"NRI")==0)

{

if(st.perc>=s.cutoff-10)

found=1;

}

else if(strcmpi(st.quota,"OBC")==0)

{

if(st.perc>=s.cutoff-20)

found=1;

}

else if(strcmpi(st.quota,"SCST")==0)

{

if(st.perc>=s.cutoff-30)

found=1;

}

}

}

return found;

}

int chkrank(student st)

{

system("cls");

int found=0;

seats s;

ifstream fin("streamchk.dat",ios::binary);

while(fin.read((char\*)&s,sizeof(s)))

{

if(strcmpi(st.stream,s.stream)==0)

{

if((strcmpi(st.quota,"General")==0))

{

if(st.rank<=s.rank)

found=1;

}

else if(strcmpi(st.quota,"NRI")==0)

{

if(st.rank-10<=s.rank)

found=1;

}

else if(strcmpi(st.quota,"OBC")==0)

{

if(st.rank-20<=s.rank)

found=1;

}

else if(strcmpi(st.quota,"SCST")==0)

{

if(st.rank-30<=s.rank)

found=1;

}

}

}

return found;

}

void modify(student st,char a)

{

system("cls");

fstream f("streamchk.dat",ios::binary|ios::in|ios::out);

seats s;

while(f.read((char\*)&s,sizeof(s)))

{

if(strcmpi(st.stream,s.stream)==0)

{

if(a=='A')

{

if(strcmpi(st.quota,"NRI")==0)

s.NRI++;

else if(strcmpi(st.quota,"General")==0)

s.general++;

else if(strcmpi(st.quota,"OBC")==0)

s.OBC++;

else if(strcmpi(st.quota,"SCST")==0)

s.SCST++;

}

else

{

if(strcmpi(st.quota,"NRI")==0)

s.NRI--;

else if(strcmpi(st.quota,"General")==0)

s.general--;

else if(strcmpi(st.quota,"OBC")==0)

s.OBC--;

else if(strcmpi(st.quota,"SCST")==0)

s.SCST--;

}

f.seekg(f.tellg()-sizeof(s));

f.write((char\*)&s,sizeof(s));

break;

}

}

f.close();

}

float getfees(char str[],char q[])

{

float f=0.0;

ifstream fin("streamchk.dat",ios::binary|ios::in);

seats st;

while(fin.read((char\*)&st,sizeof(st)))

{

if(strcmpi(st.stream,str)==0)

{

if(strcmpi(q,"NRI")==0)

f=st.f.NRI;

else if(strcmpi(q,"General")==0)

f=st.f.general;

else if(strcmpi(q,"OBC")==0)

f=st.f.OBC;

else if(strcmpi(q,"SCST")==0)

f=st.f.SCST;

}

}

return f;

}

void admn(float year)

{

system("cls");

student s;

int byear,admno;

gotoxy(9,2);

cout<<"Enter the AIEEE roll no: ";

cin>>admno;

ifstream fin("student.dat",ios::binary);

int found=0;

while(fin.read((char\*)&s,sizeof(s)))

if(s.admno==admno)

{found=1;

break;

}

fin.close();

if(found==0)

{

s.admno=admno;

gotoxy(9,3);

cout<<"Enter the students name: ";

gets(s.name);

gotoxy(9,4);

cout<<"Enter the students AIEEE rank: ";

cin>>s.rank;

gotoxy(9,5);

cout<<"Enter Class XII Grade Percentage: ";

cin>>s.perc;

gotoxy(9,6);

cout<<"Enter Stream choice: ";

gets(s.stream);

gotoxy(9,7);

cout<<"Enter which quota you wish to enroll under(General/NRI/SCST/OBC): ";

gets(s.quota);

s.cfees=getfees(s.stream,s.quota);

s.hfees=0;

s.tfees=s.cfees+s.hfees;

gotoxy(9,8);

cout<<"Enter Year of birth: ";

cin>>byear;

if(year-byear>24)

{

gotoxy(9,9);

cout<<"Overage for admission!";

}

else

{ found=chkseat(s);

if(found==1)

{

int f=chkcutoff(s);

int x=chkrank(s);

if(f==1)

{

if(x==1)

{

modify(s,'A');

gotoxy(9,10);

cout<<"Student has been admitted!";

ofstream fout("student.dat",ios::binary|ios::app);

fout.write((char\*)&s,sizeof(s));

fout.close();

}

else

{gotoxy(9,10);

cout<<"Unable to meet cutoff rank!"; }

}

else

{

gotoxy(9,10);

cout<<"Unable to meet cutoff percentage!";

}

}

else

{gotoxy(9,10);

cout<<"Seat Unavailable";

}

}

}

else

{

gotoxy(9,10);

cout<<" Duplicate Entry for student with rollno "<<admno;

}

getch();

}

void displayall()

{ system("cls");

student s;

ifstream fin("student.dat",ios::binary);

CTextModeGrid g1(1,1,80,21);

g1.SetRows(fin.tellg());

g1.SetRowHeight(0, 4);

g1.SetRowHeight(1, 14);

g1.SetRowHeight(2, 3);

g1.SetCols(6);

g1.SetColWidth(0, 10);

g1.SetColWidth(1, 15);

g1.SetColWidth(2, 22);

g1.SetColWidth(3, 10);

g1.SetColWidth(4, 10);

g1.SetColWidth(5, 20);

g1.DrawGrid();

gotoxytext(3,2,"Admno");

gotoxytext(15,2,"Name");

gotoxytext(32,2,"Quota");

gotoxytext(50,2,"Stream");

gotoxytext(61,2,"Rank");

gotoxytext(70,2,"Perc");

int i=4;

while(fin.read((char\*)&s,sizeof(s)))

{

gotoxy(3,i);

cout<<s.admno;

gotoxytext(15,i,s.name);

gotoxytext(32,i,s.quota);

gotoxytext(50,i,s.stream);

gotoxy(61,i);

cout<<s.rank;

gotoxy(70,i);

cout<<s.perc<<endl;

i++;

}

cout<<"\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n";

getch();

}

void disp1stu(long admno)

{

system("cls");

int found=0;

student s;

ifstream fin("student.dat",ios::binary);

while(fin.read((char\*)&s,sizeof(s)))

{

if(admno==s.admno)

{

found=1;

cout<<"\n\n"<<setw(8)<<"Admno"<<setw(14)<<"Name"<<setw(14)<<"Quota"<<setw(14)<<"Stream"<<setw(12)<<"Rank"<<setw(12)<<"Perc";

cout<<"\n"<<setw(8)<<s.admno<<setw(14)<<s.name<<setw(14)<<s.quota<<setw(14)<<s.stream<<setw(12)<<s.rank<<setw(12)<<s.perc;

}

}

if(found==0)

cout<<"\n\nInvalid admission number";

getch();

}

void dispstr(char stream[25])

{

student s;

system("cls");

ifstream fin("student.dat",ios::binary);

cout<<"\n\n"<<setw(10)<<"Admno"<<setw(15)<<"Name"<<setw(10)<<"Quota"<<setw(15)<<"Stream"<<setw(10)<<"Rank"<<setw(10)<<"Perc";

while(fin.read((char\*)&s,sizeof(s)))

{

if(strcmp(stream,s.stream)==0)

cout<<"\n\n"<<setw(10)<<s.admno<<setw(15)<<s.name<<setw(10)<<s.quota<<setw(15)<<s.stream<<setw(10)<<s.rank<<setw(10)<<s.perc;

}

getch();

}

void room\_modify(int no, char c)

{

system("cls");

rooms s;

int k=0;

ifstream fin("rooms.dat",ios::binary);

ofstream fout("temp.dat",ios::binary);

while(fin.read((char\*)&s,sizeof(s)))

{

if(s.rmno==no)

{s.status=c;

cout<<"\nDetails have been modified!";

k=1;

}

fout.write((char\*)&s,sizeof(s));

}

fout.close();

fin.close();

remove("rooms.dat");

rename("temp.dat","rooms.dat");

}

void deletestudent()

{

system("cls");

student s;

hostel h;

int found=0;

float admno;

ifstream fin("student.dat",ios::binary);

ofstream fout("temp.dat",ios::binary);

gotoxy(9,2);

cout<<"Enter the admission number of the student to be deleted: ";

cin>>admno;

while(fin.read((char\*)&s,sizeof(s)))

{

if(admno==s.admno)

{

modify(s,'D');

found++;

gotoxy(9,3);

cout<<"The name ";

cout<<s.name;

cout<<" is deleted from admission list";

}

else

fout.write((char\*)&s,sizeof(s));

}

fout.close();

fin.close();

remove("student.dat");

rename("temp.dat","student.dat");

fin.open("hostel.dat",ios::binary);

fout.open("temp.dat",ios::binary);

while(fin.read((char\*)&s,sizeof(s)))

{

if(admno==h.admno)

{

found++;

cout<<" and from hostel list"<<endl;

}

else

fout.write((char\*)&h,sizeof(h));

}

fin.close();

remove("hostel.dat");

rename("temp.dat","hostel.dat");

fout.close();

if(found==0)

{cout<<"\n\nInvalid admission number\n";

getch();}

else

room\_modify(admno,'V');

}

void modifycutoff()

{

system("cls");

char str[25];

seats s;

int found=0;

float perc;

long rank;

ifstream fin("streamchk.dat",ios::binary);

ofstream fout("temp.dat",ios::binary);

cout<<"\n\nEnter the name of the stream you wish to modify: ";

gets(str);

while(fin.read((char\*)&s,sizeof(s)))

{

if(strcmpi(str,s.stream)==0)

{

cout<<"\n\nEnter your new cutoff perc: ";

cin>>perc;

cout<<"\n\nEnter the rank required: ";

cin>>rank;

found=1;

s.cutoff=perc;

s.rank=rank;

}

fout.write((char\*)&s,sizeof(s));

}

if(found==0)

cout<<"\nStream not found\n";

fin.close();

fout.close();

remove("streamchk.dat");

rename("temp.dat","streamchk.dat");

}

void addstream()

{

system("cls");

seats s;

int found=0;

char stream[20];

ifstream fin("streamchk.dat",ios::binary);

gotoxy(9,2);

cout<<"Enter the stream name: ";

gets(stream);

while(fin.read((char\*)&s,sizeof(s)))

if(strcmpi(stream,s.stream)==0)

found=1;

fin.close();

if(found==0)

{

strcpy(s.stream,stream);

gotoxy(9,3);

cout<<"Enter the total number of NRI seats: ";

cin>>s.tNRI;

gotoxy(9,4);

cout<<"Enter the total number of seats for SC/ST: ";

cin>>s.tSCST;

gotoxy(9,5);

cout<<"Enter the total number of seats for OBC: ";

cin>>s.tOBC;

gotoxy(9,6);

cout<<"Enter total number of GENERAL seats: ";

cin>>s.tgeneral;

s.NRI=0;

s.SCST=0;

s.general=0;

s.OBC=0;

gotoxy(9,7);

cout<<"Enter the cutoff percentage: ";

cin>>s.cutoff;

gotoxy(9,8);

cout<<"Enter the required rank: ";

cin>>s.rank;

gotoxy(9,9);

cout<<"Enter the fees per year for general quota: ";

cin>>s.f.general;

s.f.NRI=4\*s.f.general;

s.f.SCST=(int)s.f.general/2;

s.f.OBC=(int)s.f.general/3;

ofstream fout("streamchk.dat",ios::binary|ios::app);

fout.write((char\*)&s,sizeof(s));

fout.close();

}

else

{

gotoxy(9,7);

cout<<"This Stream details already exists!!";

getch();

}

}

void showfeedata()

{

system("cls");

seats s;

ifstream fin("streamchk.dat",ios::binary);

cout<<"\n\n"<<setw(10)<<"Stream"<<setw(20)<<"NRI fee"<<setw(20)<<"SC/ST fee"<<setw(10)<<"OBC fee"<<setw(25)<<"GENERAL fee";

while(fin.read((char\*)&s,sizeof(s)))

cout<<"\n"<<setw(10)<<s.stream<<setw(20)<<s.f.NRI<<setw(20)<<s.f.SCST<<setw(10)<<s.f.OBC<<setw(25)<<s.f.general;

getch();

}

void modfeedata()

{

system("cls");

int found=0;

seats s;

ifstream fin("streamchk.dat",ios::binary);

ofstream fout("temp.dat",ios::binary);

char st[25];

cout<<"\nEnter the stream: ";

gets(st);

while(fin.read((char\*)&s,sizeof(s)))

{

if(strcmpi(s.stream,st)==0)

{

found=1;

cout<<"\n\nEnter the fees per year for General quota: ";

cin>>s.f.general;

s.f.NRI=4\*s.f.general;

s.f.SCST=(int)s.f.general/3;

s.f.OBC=(int)s.f.general/2;

}

fout.write((char\*)&s,sizeof(s));

}

fout.close();

fin.close();

remove("streamchk.dat");

rename("temp.dat","streamchk.dat");

if(found==0)

cout<<"\nInvalid stream";

getch();}

void displayseats()

{

system("cls");

seats s;

ifstream fin("streamchk.dat",ios::binary);

CTextModeGrid g1(1,1,80,21);

g1.SetRows(3);

g1.SetRowHeight(0, 2);

g1.SetRowHeight(1, 14);

g1.SetRowHeight(2, 3);

g1.SetCols(5);

g1.SetColWidth(0, 9);

g1.SetColWidth(1, 10);

g1.SetColWidth(2, 11);

g1.SetColWidth(3, 25);

g1.SetColWidth(4, 25);

g1.DrawGrid();

gotoxytext(3,2,"Stream");

gotoxytext(12,2,"Cutoff");

gotoxytext(22,2,"Rank-Req");

gotoxytext(36,2,"Seats Occupied");

gotoxytext(62,2,"Total seats");

gotoxytext(32,3,"GEN");

gotoxytext(39,3,"NRI");

gotoxytext(45,3,"SC/ST");

gotoxytext(52,3,"OBC");

gotoxytext(57,3,"GEN");

gotoxytext(62,3,"NRI");

gotoxytext(67,3,"SC/ST");

gotoxytext(75,3,"OBC");

int i=7;

while(fin.read((char\*)&s,sizeof(s)))

{gotoxy(3,i);

cout<<s.stream;

gotoxy(12,i);

cout<<s.cutoff;

gotoxy(22,i);

cout<<s.rank;

gotoxy(32,i);

cout<<s.general;

gotoxy(39,i);

cout<<s.NRI;

gotoxy(45,i);

cout<<s.SCST;

gotoxy(52,i);

cout<<s.OBC;

gotoxy(57,i);

cout<<s.tgeneral;

gotoxy(62,i);

cout<<s.tNRI;

gotoxy(67,i);

cout<<s.tSCST;

gotoxy(75,i);

cout<<s.tOBC;

i++; }

cout<<"\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n";

getch();

}

int hcheckadmno(long admn)

{

system("cls");

student s;

int found=0;

ifstream fin("student.dat",ios::binary);

while(fin.read((char\*)&s,sizeof(s)))

{

if(admn==s.admno)

found=1;

}

fin.close();

return found;

}

int hcheckroom()

{

system("cls");

rooms r;

int rn=0;

ifstream fin("rooms.dat",ios::binary);

while(fin.read((char\*)&r,sizeof(r)))

{

if(r.status=='V')

{

rn=r.rmno;

break;

}

}

return rn;

}

void writestudent(long ano,double hfee)

{

system("cls");

student s;

fstream file("student.dat",ios::binary|ios::in|ios::out);

while(file.read((char\*)&s,sizeof(s)))

{

if(ano==s.admno)

{

s.hfees=hfee;

s.tfees=s.cfees+s.hfees;

file.seekg(file.tellg()-sizeof(s));

}

file.write((char\*)&s,sizeof(s));

}

file.close();

}

void storeh()

{

system("cls");

hostel h;

int x;

cout<<"\n\nEnter admission number: ";

cin>>h.admno;

x=hcheckadmno(h.admno);

if(x==1)

{

h.rno=hcheckroom();

if(h.rno!=0)

{

cout<<"\n\nYour room number is: "<<h.rno;

h.rfees=15000;

cout<<"\n\nYour hostel fees for the 1st year is Rs"<<h.rfees;

cout<<"\n\nEnter your diet preference-Veg/Non-veg(v/n): ";

cin>>h.food;

if(h.food=='v'||h.food=='V')

h.mfees=1800;

else if(h.food=='n'||h.food=='N')

h.mfees=2000;

cout<<"\n\nThe mess fees for the 1st year is Rs"<<h.mfees;

h.tfees=h.rfees+h.mfees;

cout<<"\n\nTherefore The total hostel fees is Rs"<<h.tfees;

getch();

ofstream file("hostel.dat",ios::binary|ios::app);

file.write((char\*)&h,sizeof(h));

file.close();

room\_modify(h.rno,'O');

}

else

cout<<"\n\nNo room available\n";

}

else

cout<<"\n\nInvalid Admission Number";

getch();

}

int hccount()

{

system("cls");

ifstream fin("hostel.dat",ios::binary);

fin.seekg(0,ios::end);

long x=fin.tellg();

int nor=x/sizeof(hostel);

fin.close();

return nor;

}

void hread()

{ system("cls");

hostel h;

ifstream fin("hostel.dat",ios::binary);

cout<<"\n\n"<<setw(10)<<"Room No"<<setw(12)<<"Admno"<<setw(10)<<"Diet"<<setw(15)<<"Mess fee"<<setw(15)<<"Hostel fee"<<setw(15)<<"Total fee";

while(fin.read((char\*)&h,sizeof(h)))

cout<<"\n"<<setw(10)<<h.rno<<setw(12)<<h.admno<<setw(10)<<h.food<<setw(15)<<h.mfees<<setw(15)<<h.rfees<<setw(15)<<h.tfees;

fin.close();

getch();

}

void hdisplay()

{

system("cls");

hostel h;

long admno;

int x=0;

ifstream fin("hostel.dat",ios::binary);

cout<<"\n\nEnter the admission number of the hostelite whose details you wish to display: ";

cin>>admno;

while(fin.read((char\*)&h,sizeof(h)))

{if(h.admno==admno)

{

x=1;

cout<<"\n\n"<<setw(10)<<"Room No"<<setw(12)<<"Admno"<<setw(10)<<"Diet"<<setw(15)<<"Mess fee"<<setw(15)<<"Hostel fee"<<setw(15)<<"Total fee";

cout<<"\n"<<setw(10)<<h.rno<<setw(12)<<h.admno<<setw(10)<<h.food<<setw(15)<<h.mfees<<setw(15)<<h.rfees<<setw(15)<<h.tfees;

}

}

if(x==0)

cout<<"\n\nInvalid Admission Number\n";

fin.close();

getch();

}

void hmodify()

{

system("cls");

hostel h;

long admno;

int x;

ifstream fin("hostel.dat",ios::binary);

ofstream fout("temp.dat",ios::binary);

cout<<"\n\nEnter Admission number:";

cin>>admno;

x=hcheckadmno(admno);

if(x==1)

{

while(fin.read((char\*)&h,sizeof(h)))

{if(h.admno==admno)

{

cout<<"\n\nEnter Diet Preference(v/n):";

cin>>h.food;

if(h.food=='V'||h.food=='v')

h.mfees=1800;

else if(h.food=='N'||h.food=='n')

h.mfees=2000;

h.tfees=h.rfees+h.mfees;

cout<<"\n\nThe mess fees for 1st year is Rs"<<h.mfees;

}

fout.write((char\*)&h,sizeof(h));

}

fout.close();

fin.close();

remove("hostel.dat");

rename("temp.dat","hostel.dat");

}

else

cout<<"\n\nInvalid admission number\n";

}

void resetrooms()

{

system("cls");

rooms r;

ofstream fout("rooms.dat",ios::binary);

for(int i=1;i<=200;i++)

{

r.rmno=i;

r.status='V';

fout.write((char\*)&r,sizeof(r));

}

fout.close();

}

void hdelete(int admno)

{

system("cls");

int no;

hostel s;

int found=0;

ifstream fin("hostel.dat",ios::binary);

ofstream fout("temp.dat",ios::binary);

while(fin.read((char\*)&s,sizeof(s)))

{

if(admno==s.admno)

{

no=s.rno;

found=1;

cout<<"\nThe hostelite with admission number"<<s.admno<<"is deleted\n";

}

else

fout.write((char\*)&s,sizeof(s));

}

fin.close();

fout.close();

remove("hostel.dat");

rename("temp.dat","hostel.dat");

if(found==0)

cout<<"\n\nInvalid Admssion number\n";

else

room\_modify(no,'V');

getch();

}

void streamenu()

{textbackground(BLACK);

textcolor(YELLOW);

system("cls");

int c;

char str[25];

do

{

system("cls");

cout<<"\n!==============================================================================!\n\n";

gotoxy(30,4);

cout<<"STREAM MENU";

cout<<"\n\n!==============================================================================!";

gotoxy(20,8);

cout<<"1.Add new streams";

gotoxy(20,9);

cout<<"2.Display details of all streams available";

gotoxy(20,10);

cout<<"3.Display students of a particular stream";

gotoxy(20,11);

cout<<"4.Modify cut-off mark";

gotoxy(20,12);

cout<<"5.View fee detail of stream";

gotoxy(20,13);

cout<<"6.Modify fee detail of stream";

gotoxy(20,14);

cout<<"7.Back to Main Menu";

gotoxy(20,15);

cout<<"Enter your choice: ";

cout<<"\n\n\n!==============================================================================!";

gotoxy(39,15);

cin>>c;

switch(c)

{

case 1:addstream();

break;

case 2:displayseats();

break;

case 3:

system("cls");

gotoxy(21,16);

cout<<"Enter the stream: ";

gets(str);

dispstr(str);

break;

case 4:modifycutoff();

break;

case 5:showfeedata();

break;

case 6:modfeedata();

break;

case 7:break;

default:gotoxy(21,16);

cout<<"Invalid choice. Re-enter suitable option. ";

getch();

}

}while(c!=7);

}

void displayfee(int no)

{

system("cls");

int found=0;

student s;

ifstream fin("student.dat",ios::binary);

while(fin.read((char\*)&s,sizeof(s)))

{if(no==s.admno)

{found=1;

cout<<"\nName :"<<s.name;

cout<<"\nCollege Fees :"<<s.cfees;

if(s.hfees!=0)

{

cout<<"\nHostel Fee :"<<s.hfees;

cout<<"\nTotal Fees :"<<s.tfees;

}}}

if(found==0)

cout<<"\n\nInvalid Admission Number\n";

getch();

}

void studentmenu()

{textbackground(BLACK);

textcolor(YELLOW);

system("cls");

int date,c;

long admno,no;

do{

system("cls");

cout<<"\n!==============================================================================!\n\n";

gotoxy(30,4);

cout<<"STUDENT MENU";

cout<<"\n\n!==============================================================================!";

gotoxy(20,8);

cout<<"1.Enter new student details";

gotoxy(20,9);

cout<<"2.Display all student details";

gotoxy(20,10);

cout<<"3.Display one student's details";

gotoxy(20,11);

cout<<"4.Delete a student's details";

gotoxy(20,12);

cout<<"5.View Fee Details";

gotoxy(20,13);

cout<<"6.Back to Main Menu";

gotoxy(20,15);

cout<<"Enter your choice: ";

cout<<"\n\n\n!==============================================================================!";

gotoxy(39,15);

cin>>c;

switch(c)

{

case 1: gotoxy(21,16);

cout<<"Enter the current year: ";

cin>>date;

admn(date);

break;

case 2:displayall();

break;

case 3:system("cls");

gotoxy(21,16);

cout<<"Enter admission number: ";

cin>>admno;

disp1stu(admno);

break;

case 4:deletestudent();

break;

case 5:gotoxy(21,16);

cout<<"Enter admission number: ";

cin>>no;

displayfee(no);

case 6:break;

default:gotoxy(20,16);

cout<<"Invalid choice. Re-enter suitable option. ";

getch();;

}}while(c!=6);

}

void hostelmenu()

{textbackground(BLACK);

textcolor(YELLOW);

system("cls");

long admno;

int c;

do{

system("cls");

cout<<"\n!==============================================================================!\n\n";

gotoxy(30,4);

cout<<"HOSTEL MENU";

cout<<"\n\n!==============================================================================!";

gotoxy(20,8);

cout<<"1.Enter new hostelites details";

gotoxy(20,9);

cout<<"2.Display all hostelites details";

gotoxy(20,10);

cout<<"3.Display a hostelite's details";

gotoxy(20,11);

cout<<"4.Modify a hostelite's details";

gotoxy(20,12);

cout<<"5.Delete a hostelite's details";

gotoxy(20,13);

cout<<"6.Back to Main Menu";

gotoxy(20,15);

cout<<"Enter your choice: ";

cout<<"\n\n\n!==============================================================================!";

gotoxy(39,15);

cin>>c;

switch(c)

{

case 1:storeh();

break;

case 2:hread();

break;

case 3:hdisplay();

break;

case 4:hmodify();

break;

case 5:gotoxy(20,16);

cout<<"Enter admission number: ";

cin>>admno;

hdelete(admno);

break;

case 6:break;

default:gotoxy(20,16);

cout<<"Invalid choice. Re-enter suitable option. ";

getch();

}

}while(c!=6);

}

void setcursor(bool, DWORD);

HANDLE console = GetStdHandle(STD\_OUTPUT\_HANDLE);

void setcursor();

void setcursor(bool visible, DWORD size)

{

if(size == 0)

{

size = 20;

}

CONSOLE\_CURSOR\_INFO lpCursor;

lpCursor.bVisible = visible;

lpCursor.dwSize = size;

SetConsoleCursorInfo(console,&lpCursor);

}

void intro()

{

system("cls");

textbackground(BLACK);

textcolor(GREEN);

gotoxy(22,8);

cout<<"Computerized College Admission System";

for(int i=7;i<10;i++)

{gotoxy(20,i);

cout<<"|"; }

for(int i=7;i<10;i++)

{gotoxy(60,i);

cout<<"|"; }

gotoxy(20,6);

cout<<"-----------------------------------------";

gotoxy(20,10);

cout<<"-----------------------------------------";

gotoxy(32,17);

cout<<"PRESENTED BY: ";

gotoxy(32,18);

cout<<"ABEN THOMAS GEORGE";

gotoxy(32,19);

cout<<"SIDDHARTH PANDALAI";

gotoxy(32,20);

cout<<"SHANE ALEX";

gotoxy(24,12);

cout<<" \*\*\* PRESS ANY KEY TO CONTINUE \*\*\*";

setcursor(0,0);

getch();

}

int main()

{

textbackground(BLACK);

textcolor(GREEN);

int ch; char et;

intro();

do{

again:;

system("cls");

textbackground(BLACK);

cout<<"\n!==============================================================================!\n\n";

gotoxy(35,4);

cout<<"MENU";

cout<<"\n\n!==============================================================================!";

gotoxy(20,8);

cout<<"0.Reset Rooms";

gotoxy(20,9);

cout<<"1.Stream Details";

gotoxy(20,10);

cout<<"2.Student Details";

gotoxy(20,11);

cout<<"3.Hostel Details";

gotoxy(20,12);

cout<<"4.Exit";

gotoxy(20,16);

cout<<"Enter your choice: ";

setcursor(1,10);

cout<<"\n\n\n!==============================================================================!";

gotoxy(39,16);

cin>>ch;

switch(ch)

{case 0: resetrooms();

break;

case 1: streamenu();

break;

case 2: studentmenu();

break;

case 3: hostelmenu();

break;

case 4:system("cls");

cout<<"\n!==============================================================================!\n";

gotoxy(15,8);

cout<<"Do You Wish To Exit The Program(Y/N)? : ";

cout<<"\n\n\n\n\n\n\n\n\n\n!==============================================================================!";

gotoxy(58,8);

cin>>et;

if(et=='y'||et=='Y')

{

goto ex;

}

else

{

goto again;

}

ex:;

system("cls");

textcolor(YELLOW);

Sleep(1);

system("cls");

gotoxy(35,10);

cout<<" THANKS ";

Sleep(1);

system("cls");

gotoxy(35,20);

cout<<" Thank You ";

Sleep(5);

system("cls");

gotoxy(25,15);

textcolor(BLUE+BLINK);

gotoxy(25,8);

textcolor(YELLOW+BLINK);

Sleep(1000);

cout<<"HAVE A NICE DAY";

gotoxy(25,9);

textcolor(YELLOW+BLINK);

cout<<"Press ENTER to EXIT.......";

getch();

exit(0);

default: gotoxy(20,16);

cout<<"Invalid choice. Re-enter suitable option. ";

Sleep(2000);

}

}while(ch!=4);

getch();

return 0;

}

**Input/Outputs:**



























