SEMESTER PROJECT.

TASK:

#include <iostream>

#include<string>

#include<fstream>

using namespace std;

class assesment

{

public:

string asses\_subject;

string assesment\_name;

int assesment\_marks;

string assesment\_weightage;

string student\_username;

void entr\_assesment()

{

cout << endl << "Enter Assesment Name (Quiz/Assignment/Project/Sessional):";

cin >> assesment\_name;

cout << endl << "Enter Subject For Which Creating Assesment:";

cin >> asses\_subject;

cout << endl << "Enter Assesment Marks:(int)";

cin >> assesment\_marks;

cout << endl << "Enter Assesment Weightage in Total:";

cin >> assesment\_weightage;

cout << endl << "Enter Student username for which you give:";

cin >> student\_username;

}

void view\_assesment() {

static int g = 0;

g++;

cout << endl << "Assesment No:" << g << " Data :";

cout << endl << assesment\_name;

cout << endl << assesment\_marks;

cout << endl << assesment\_weightage;

}

string return\_asses\_user() { return student\_username; }

assesment\* assesment\_nptr;

};

class attendance

{

public:

string username;

string subject;

string attendance\_status;

string date;

string slot;

void attendance\_fun()

{

cout << endl << "Enter Attendance status:";

cin >> attendance\_status;

cout << endl << "Enter Date:";

cin >> date;

cout << endl << "Enter slot:";

cin >> slot;

}

void view\_attendance()

{

cout << endl << attendance\_status << " " << date << " " << slot << endl << endl;

}

attendance\* attendnptr;

};

class studentcoursedata

{

public:

string courseName;

string studnet\_username;

string teacher;

string code;

int credithour;

int semester\_no;

studentcoursedata\* scdnptr;

};

class user

{

public:

string username;

string password;

};

class course

{

public:

string courseName;

string teacher;

string code;

int credithour;

int semester\_no;

public:

void entercoursedata()

{

cout << endl << "------------------" << endl << "Enter New Course Name:";

cin >> courseName;

cout << endl << "Enter course Teacher:";

cin >> teacher;

cout << endl << "Enter Course Code:";

cin >> code;

cout << endl << "Enter course Credit Hour:(int)";

cin >> credithour;

cout << endl << "Enter Semester Name In which You Offer This Course.(int)";

cin >> semester\_no;

}

int return\_sem()

{

return semester\_no;

}

string return\_course()

{

return courseName;

}

string return\_teacher() { return teacher; }

string return\_code() { return code; }

int return\_credit() {

return credithour;

}

course\* nextcourse;

};

class Admin :public user, public course

{

public:

string name;

void admin\_data()

{

cout << endl << "------------------" << endl << "Enter Admin Name Please Dont Use space:";

cin >> name;

cout << endl << "Enter Admin Username:";

cin >> username;

cout << endl << "Enter Admin Password:";

cin >> password;

}

void Displayadmin()

{

cout << endl << "The Admin Has Data:";

cout << endl << "Name:" << name;

cout << endl << "Username:" << username;

cout << endl << "Password:" << password;

}

string return\_username() { return username; }

string return\_password() { return password; }

Admin\* naptr;

};

class Faculty\_member :public user

{

public:

string name;

void faculty\_data()

{

cout << endl << "-----------" << endl << "Enter New Faculty Member Data:";

cout << endl << "Enter Faculty Name:";

cin >> name;

cout << endl << "Enter Faculty Username:";

cin >> username;

cout << endl << "Enter aculty Password:";

cin >> password;

}

void facultydisplay()

{

cout << endl << "Name : " << name;

cout << endl << "UserName : " << username;

cout << endl << "Password : " << password;

}

string re\_username() { return username; }

string re\_password() { return password; }

Faculty\_member\* nfptr;

};

class student :public user

{

public:

string roll\_no;

string stu\_name;

void student\_dataentry()

{

cout << endl << "-----------" << endl << "Enter New Student Data:";

cout << endl << "Enter Student Name:";

cin >> stu\_name;

cout << endl << "Enter Student User Name:";

cin >> username;

cout << endl << "Enter Student Username password:";

cin >> password;

cout << endl << "Enter Student RollNumber:";

cin >> roll\_no;

}

void studentdisplay()

{

cout << endl << " Student all Data ";

cout << endl << "Name: " << stu\_name;

cout << endl << "Uersname " << username;

cout << endl << "Password " << password;

cout << endl << "Roll\_NO: " << roll\_no;

}

string re\_user() { return username; }

string re\_pass() { return password; }

string re\_roll() { return roll\_no; }

student\* nsptr;

};

Faculty\_member\* ffptr = NULL;

student\* fsptr = NULL;

Admin\* faptr = NULL;

course\* firstcourse = NULL;

studentcoursedata\* scdfirst;

attendance\* attendfptr = NULL;

assesment\* assesment\_fptr = NULL;

void mainmenu();

void adminDataEntry();

void loginportal();

void storedata();

float credit\_2 = 0;

float credit\_3 = 0;

string viewattendance;

int main()

{

string line1, line2, line3,line4,line5,line6;

int value1, value2;

cout << endl << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Welcome To Pieas Portal\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

fstream afile;

afile.open("Admin.txt", ios::in);

if (afile)

{

while (!afile.eof())

{

getline(afile, line1);

getline(afile, line2);

getline(afile, line3);

Admin\* anode = new Admin;

anode->name = line1;

anode->username = line2;

anode->password = line3;

if (faptr == NULL)

{

anode->naptr = NULL;

faptr = anode;

}

else

{

Admin\* currenta = faptr;

while (currenta->naptr != NULL) { currenta = currenta->naptr; }

currenta->naptr = anode;

anode->naptr = NULL;

}

}

}

else { cout << " "; }

afile.close();

fstream ffile;

ffile.open("Faculty.txt", ios::in);

if (ffile)

{

while (!ffile.eof())

{

getline(ffile, line1);

getline(ffile, line2);

getline(ffile, line3);

Faculty\_member\* nodef = new Faculty\_member;

nodef->name = line1;

nodef->username = line2;

nodef->password = line3;

if (ffptr == NULL)

{

nodef->nfptr = NULL;

ffptr = nodef;

}

else

{

Faculty\_member\* currentf = ffptr;

while (currentf->nfptr != NULL) { currentf=currentf->nfptr; }

currentf->nfptr = nodef;

nodef->nfptr = NULL;

}

}

}

else { cout << " "; }

ffile.close();

fstream sfile;

sfile.open("Student.txt", ios::in);

if (sfile)

{

while (!sfile.eof())

{

getline(sfile, line1);

getline(sfile, line2);

getline(sfile, line3);

getline(sfile, line4);

student\* nodes = new student;

nodes->stu\_name = line1;

nodes->username = line2;

nodes->password = line3;

nodes->roll\_no = line4;

if (fsptr == NULL)

{

nodes->nsptr = NULL;

fsptr = nodes;

}

else

{

student\* currents = fsptr;

while (currents->nsptr != NULL) { currents = currents->nsptr; }

currents->nsptr = nodes;

nodes->nsptr = NULL;

}

}

}

else { cout << ""; }

sfile.close();

fstream cfile;

cfile.open("CourseData.txt", ios::in);

if (cfile)

{

while (!cfile.eof())

{

getline(cfile, line1);

getline(cfile, line2);

getline(cfile, line3);

getline(cfile, line4);

getline(cfile, line5);

course\* nodec = new course;

nodec->courseName = line1;

nodec->teacher = line2;

nodec->code = line3;

value1 = stoi(line4);

value2 = stoi(line5);

nodec->credithour = value1;

nodec->semester\_no = value2;

if (firstcourse == NULL)

{

nodec->nextcourse = NULL;

firstcourse = nodec;

}

else

{

course\* currentc = firstcourse;

while (currentc->nextcourse != NULL) { currentc = currentc->nextcourse; }

currentc->nextcourse = nodec;

nodec->nextcourse = NULL;

}

}

}

else { cout << ""; }

cfile.close();

fstream scdfile;

scdfile.open("Studentcoursedata.txt", ios::in);

if (scdfile)

{

while (!scdfile.eof())

{

getline(scdfile, line1);

getline(scdfile, line2);

getline(scdfile, line3);

getline(scdfile, line4);

getline(scdfile, line5);

getline(scdfile, line6);

studentcoursedata\* nodescd = new studentcoursedata;

nodescd->courseName = line1;

nodescd->studnet\_username = line2;

nodescd->teacher = line3;

nodescd->code = line4;

value1 = stoi(line5);

value2 = stoi(line6);

nodescd->credithour = value1;

nodescd->semester\_no = value2;

if (scdfirst == NULL)

{

nodescd->scdnptr = NULL;

scdfirst = nodescd;

}

else

{

studentcoursedata\* currentscd = scdfirst;

while (currentscd->scdnptr != NULL) { currentscd = currentscd->scdnptr; }

currentscd->scdnptr = nodescd;

nodescd->scdnptr = NULL;

}

}

}

else { cout <<" "; }

scdfile.close();

fstream attendfile;

attendfile.open("Attendance.txt", ios::in);

if (attendfile)

{

while (!attendfile.eof())

{

getline(attendfile, line1);

getline(attendfile, line2);

getline(attendfile, line3);

getline(attendfile, line4);

getline(attendfile, line5);

attendance\* nodeattend = new attendance;

nodeattend->username = line1;

nodeattend->subject = line2;

nodeattend->attendance\_status = line3;

nodeattend->date = line4;

nodeattend->slot = line5;

if (attendfptr == NULL)

{

nodeattend->attendnptr = NULL;

attendfptr = nodeattend;

}

else

{

attendance\* cattend = attendfptr;

while (cattend->attendnptr != NULL) { cattend = cattend->attendnptr; }

cattend->attendnptr = nodeattend;

nodeattend->attendnptr = NULL;

}

}

}

else { cout << ""; }

attendfile.close();

fstream assessfile;

assessfile.open("Assesement.txt", ios::in);

if (assessfile)

{

while (!assessfile.eof())

{

getline(assessfile, line1);

getline(assessfile, line2);

getline(assessfile, line3);

getline(assessfile, line4);

getline(assessfile, line5);

assesment\* nodeassess = new assesment;

nodeassess->asses\_subject=line1;

nodeassess->assesment\_name = line2;

value1 = stoi(line3);

nodeassess->assesment\_marks = value1;

nodeassess->assesment\_weightage = line4;

nodeassess->student\_username = line5;

if (assesment\_fptr == NULL)

{

nodeassess->assesment\_nptr = NULL;

assesment\_fptr = nodeassess;

}

else

{

assesment\* cassess = assesment\_fptr;

while (cassess->assesment\_nptr != NULL) { cassess = cassess->assesment\_nptr; }

cassess->assesment\_nptr = nodeassess;

nodeassess->assesment\_nptr = NULL;

}

}

}

else { cout << ""; }

assessfile.close();

mainmenu();

return 0;

}

void mainmenu()

{

int op;

cout << endl << "Press 1 To Enter Data of Users only Admin Can Use.(make sure to enter first Admin data SO further procedure continue):";

cout << endl << "Press 2 to Login To Your Account------Any one." << endl;

cout << endl << "press 3 to exit the programme (your data should save on this command.)";

cin >> op;

switch (op)

{

case 1:

{

adminDataEntry();

break;

}

case 2:

{

loginportal();

break;

}

case 3:

{

storedata();

exit(0);

}

}

}

void adminDataEntry()

{

int op, n, oop;

cout << endl << "Press 1 for Admin Data Entery." << endl << "Press 2 for Faculty Data Entery." << endl << "Press 3 for Students Data Entery.";

cin >> oop;

switch (oop)

{

case 1:

{

string usernam, pass;

Admin\* admin1;

admin1 = new Admin;

if (faptr == NULL)

{

cout << endl << "First Entry No Need Of ID.";

admin1->admin\_data();

admin1->naptr = NULL;

faptr = admin1;

}

else

{

cout << endl << "Here Need A login ID For Further Data Entry of Admin:";

cout << endl << "Enter Your Username:";

cin >> usernam;

cout << endl << "Enter Your Password:";

cin >> pass;

Admin\* currentptra = faptr;

while (currentptra != NULL) {

if (currentptra->return\_username() == usernam && currentptra->return\_password() == pass) {

break;

}

else { currentptra = currentptra->naptr; }

}

if (currentptra == NULL) {

cout << endl << "Wrong Username Or Password." << endl << "ReEnter:";

adminDataEntry();

}

if (usernam == currentptra->return\_username() && pass == currentptra->return\_password())

{

cout << endl << "How Many Enteries You want:";

cin >> n;

for (int i = 0; i < n; i++)

{

Admin\* currenta = faptr;

while (currenta->naptr != NULL)

{

currenta = currenta->naptr;

}

Admin\* admin1;

admin1 = new Admin;

admin1->admin\_data();

currenta->naptr = admin1;

admin1->naptr = NULL;

}

}

}

break;

}

case 2:

{

string usernam, pass;

cout << endl << "Here Need A login ID of Admin For Further Data Entry of Faculty Member:";

cout << endl << "Enter Your Username:";

cin >> usernam;

cout << endl << "Enter Your Password:";

cin >> pass;

Admin\* currentptra = faptr;

while (currentptra != NULL) {

if (currentptra->return\_username() == usernam && currentptra->return\_password() == pass) {

break;

}

else { currentptra = currentptra->naptr; }

}

if (currentptra == NULL) {

cout << endl << "Wrong Username Or Password." << endl << "ReEnter:";

adminDataEntry();

}

if (usernam == currentptra->return\_username() && pass == currentptra->return\_password())

{

cout << endl << "How Many Enteries Of New Faculty Member You Want.";

cin >> n;

Faculty\_member\* faculty1;

faculty1 = new Faculty\_member;

for (int i = 0; i < n; i++)

{

if (ffptr == NULL)

{

faculty1->faculty\_data();

faculty1->nfptr = NULL;

ffptr = faculty1;

}

else

{

Faculty\_member\* currentf = ffptr;

while (currentf->nfptr != NULL)

{

currentf = currentf->nfptr;

}

Faculty\_member\* faculty1;

faculty1 = new Faculty\_member;

faculty1->faculty\_data();

currentf->nfptr = faculty1;

faculty1->nfptr = NULL;

}

}

}

break;

}

case 3:

{

string usernam, pass;

cout << endl << "Here Need A login ID of Admin For Further Data Entry of Student:";

cout << endl << "Enter Your Username:";

cin >> usernam;

cout << endl << "Enter Your Password:";

cin >> pass;

Admin\* currentptra = faptr;

while (currentptra != NULL) {

if (currentptra->return\_username() == usernam && currentptra->return\_password() == pass) {

break;

}

else { currentptra = currentptra->naptr; }

}

if (currentptra == NULL) {

cout << endl << "Wrong Username Or Password." << endl << "Chose Again.";

adminDataEntry();

}

if (usernam == currentptra->return\_username() && pass == currentptra->return\_password())

{

cout << endl << "How Many Enteries Of New STUDENT You Want.";

cin >> n;

student\* stud1;

stud1 = new student;

for (int i = 0; i < n; i++)

{

if (fsptr == NULL)

{

stud1->student\_dataentry();

stud1->nsptr = NULL;

fsptr = stud1;

}

else

{

student\* currents = fsptr;

while (currents->nsptr != NULL)

{

currents = currents->nsptr;

}

student\* stud1;

stud1 = new student;

stud1->student\_dataentry();

currents->nsptr = stud1;

stud1->nsptr = NULL;

}

}

}

break;

}

default:

break;

}

cout << endl << "Press 1 for Mainmenu.:" << endl << "Press 2 for ReEntry" << endl;

cin >> op;

if (op == 1) { mainmenu(); }

else { adminDataEntry(); }

}

void loginportal()

{

int op;

string usernam, pass;

cout << endl << " Welcome To Login Portal. ";

cout << endl << "Press 1 if You are----------> Admin." << endl << "Press 2 if you are-----------> Faculty Member." << endl

<< "Press 3 if you Are-----------> student." << endl;

cin >> op;

switch (op)

{

case 1:

{

cout << endl << "Enter Your Credentials." << endl << "Enter Username:";

cin >> usernam;

cout << endl << "Enter Your Password:";

cin >> pass;

Admin\* currentptra = faptr;

while (currentptra != NULL) {

if (currentptra->return\_username() == usernam && currentptra->return\_password() == pass) {

break;

}

else { currentptra = currentptra->naptr; }

}

if (currentptra == NULL) {

cout << endl << "Wrong Username Or Password." << endl << "Choose Agian.";

loginportal();

}

if (usernam == currentptra->return\_username() && pass == currentptra->return\_password())

{

cout << endl << " Welcome Mrs. " << currentptra->name;

int g;

cout << endl << "Press 1 To Offer Semester:";

cin >> g;

if (g == 1)

{

int v;

cout << endl << " Enter All Courses Of Semesters You Offering For All Batches.";

cout << endl << "How Many Course You Want to OFFER:";

cin >> v;

course\* semestercoursedata;

semestercoursedata = new course;

for (int i = 0; i < v; i++)

{

if (firstcourse == NULL)

{

semestercoursedata->entercoursedata();

semestercoursedata->nextcourse = NULL;

firstcourse = semestercoursedata;

}

else

{

course\* currentcourse = firstcourse;

while (currentcourse->nextcourse != NULL)

{

currentcourse = currentcourse->nextcourse;

}

course\* semestercoursedata;

semestercoursedata = new course;

semestercoursedata->entercoursedata();

currentcourse->nextcourse = semestercoursedata;

semestercoursedata->nextcourse = NULL;

}

}

}

}

break;

}

case 2:

{

cout << endl << "Enter Your Credentials." << endl << "Enter Username:";

cin >> usernam;

cout << endl << "Enter Your Password:";

cin >> pass;

Faculty\_member\* currentptrf = ffptr;

while (currentptrf != NULL) {

if (currentptrf->re\_username() == usernam && currentptrf->re\_password() == pass) {

break;

}

else { currentptrf = currentptrf->nfptr; }

}

if (currentptrf == NULL) {

cout << endl << "Wrong Username Or Password." << endl << "Chose Again.";

loginportal();

}

if (usernam == currentptrf->re\_username() && pass == currentptrf->re\_password())

{

cout << endl << "Welcome Mrs." << currentptrf->name;

int p;

string subjectname, studentusername;

cout << endl << "Press 1 to Enter Atendance Of Student (make sure student enrolled Already.): ";

cout << endl << "Press 2 to Create Assesment:(make sure student enrolled Already.)";

cin >> p;

if (p == 1)

{

Mainmenu:

cout << endl << "Enter Subject Name for Attendance:";

cin >> subjectname;

cout << endl << "Enter Student Username : ";

cin >> studentusername;

studentcoursedata\* scdptr = scdfirst;

while (scdptr != NULL)

{

if (scdptr->studnet\_username == studentusername && scdptr->courseName == subjectname)

{

int u;

cout << endl << "How Many Attendance Entries You Want To Enter:";

cin >> u;

for (int i = 0; i < u; i++) {

attendance\* attendptr;

attendptr = new attendance;

if (attendfptr == NULL)

{

attendptr->username = studentusername;

attendptr->subject = subjectname;

attendptr->attendance\_fun();

attendptr->attendnptr = NULL;

attendfptr = attendptr;

}

else

{

attendance\* currentattendance = attendfptr;

while (currentattendance->attendnptr != NULL)

{

currentattendance = currentattendance->attendnptr;

}

attendance\* attendptr;

attendptr = new attendance;

attendptr->username = studentusername;

attendptr->subject = subjectname;

attendptr->attendance\_fun();

currentattendance->attendnptr = attendptr;

attendptr->attendnptr = NULL;

}

}

}

scdptr = scdptr->scdnptr;

}

int o;

cout << endl << "Another Entry Press 1 :" << endl << "For No Entry Press Any Integer Key:";

cin >> o;

if (o == 1) { goto Mainmenu; }

}

else if (p == 2)

{

int q;

cout << endl << "How many students assesment you want to create:";

cin >> q;

for (int i = 0; i < q; i++)

{

assesment\* asses\_ptr;

asses\_ptr = new assesment;

if (assesment\_fptr == NULL)

{

asses\_ptr->entr\_assesment();

asses\_ptr->assesment\_nptr = NULL;

assesment\_fptr = asses\_ptr;

}

else

{

assesment\* current\_asses = assesment\_fptr;

while (current\_asses->assesment\_nptr != NULL)

{

current\_asses = current\_asses->assesment\_nptr;

}

assesment\* asses\_ptr;

asses\_ptr = new assesment;

asses\_ptr->entr\_assesment();

current\_asses->assesment\_nptr = asses\_ptr;

asses\_ptr->assesment\_nptr = NULL;

}

}

}

}

break;

}

case 3:

{

cout << endl << "Enter Your Credentials." << endl << "Enter Username:";

cin >> usernam;

::viewattendance = usernam;

cout << endl << "Enter Your Password:";

cin >> pass;

student\* currentptrs = fsptr;

while (currentptrs != NULL) {

if (currentptrs->re\_user() == usernam && currentptrs->re\_pass() == pass) {

break;

}

else { currentptrs = currentptrs->nsptr; }

}

if (currentptrs == NULL) {

cout << endl << " Wrong Username Or Password." << endl << "Chose Again. ";

loginportal();

}

if (usernam == currentptrs->re\_user() && pass == currentptrs->re\_pass())

{

int en;

cout << endl << "Welcome Mrs." << currentptrs->stu\_name;

cout << endl << "Press 1 for Semester Enrollment:";

cout << endl << "Press 2 for Check Your Fee Vouture:(Make Sure Enrollment First.)";

cout << endl << "Press 3 for View Attendance:(If Faculty Update Then You see.)";

cout << endl << "Press 4 To ckeck Your Own Grading:(If Faculty Update Then You see.)";

cin >> en;

if (en == 1)

{

int s;

cout << endl << "In Which Semester You Want Enrollment:";

cin >> s;

course\* currentcourse = firstcourse;

while (currentcourse != NULL)

{

if (currentcourse->return\_sem() == s)

{

cout << endl << currentcourse->return\_course();

}

currentcourse = currentcourse->nextcourse;

}

if (currentcourse == NULL) { cout << endl << "Above Courses Are Available In Smester No < " << s << " >"; }

int l;

cout << endl << "Enter How many courses you want to Enroll:";

cin >> l;

for (int i = 0; i < l; i++)

{

string coursename, uname;

cout << endl << "Enter Course Name Those Which Are Above Mentoined in semester < " << s << " >" << endl;

cin >> coursename;

course\* currentcourse = firstcourse;

while (currentcourse != NULL)

{

if (currentcourse->return\_course() == coursename) {

student\* currentstudent = fsptr;

while (currentstudent != NULL)

{

if (currentstudent->re\_user() == ::viewattendance) {

studentcoursedata\* scdata;

scdata = new studentcoursedata;

if (scdfirst == NULL)

{

scdata->studnet\_username = ::viewattendance;

scdata->courseName = currentcourse->return\_course();

scdata->teacher = currentcourse->return\_teacher();

scdata->code = currentcourse->return\_code();

scdata->semester\_no = currentcourse->return\_sem();

scdata->credithour = currentcourse->return\_credit();

if (scdata->credithour == 2) { ::credit\_2 = ::credit\_2 + 5000; }

else if (scdata->credithour == 3) { ::credit\_3 = ::credit\_3 + 7500; }

scdata->scdnptr = NULL;

scdfirst = scdata;

}

else

{

studentcoursedata\* currentscdata = scdfirst;

while (currentscdata->scdnptr != NULL) { currentscdata = currentscdata->scdnptr; }

studentcoursedata\* scdata;

scdata = new studentcoursedata;

scdata->studnet\_username = ::viewattendance;

scdata->courseName = currentcourse->return\_course();

scdata->teacher = currentcourse->return\_teacher();

scdata->code = currentcourse->return\_code();

scdata->semester\_no = currentcourse->return\_sem();

scdata->credithour = currentcourse->return\_credit();

if (scdata->credithour == 2) { ::credit\_2 = ::credit\_2 + 5000; }

else if (scdata->credithour == 3) { ::credit\_3 = ::credit\_3 + 7500; }

currentscdata->scdnptr = scdata;

scdata->scdnptr = NULL;

}

}

currentstudent = currentstudent->nsptr;

}

}

currentcourse = currentcourse->nextcourse;

}

}

}

else if (en == 2)

{

cout << endl << " Your Total Fee Is: ";

cout << endl << " Transport Fee " << 15000 << " RS/- " <<

endl << "Hostel Charges " << 27000 << " RS/-" << endl << "IP FEE " << 1000 << " RS/-" << endl << "Library Fee " << 5000 << " RS/-";

cout << endl << "Course Enrollment Fee Is: " << ::credit\_2 + ::credit\_3;

float Total = ::credit\_3 + ::credit\_2 + 15000.0 + 1000.0 + 27000.0 + 5000.0;

cout << endl << "TOTAL FEE IS :" << Total << endl;

}

else if (en == 3)

{

gg:

string s;

cout << endl << "Enter Your Subject of which Subject you want To View Attendance:";

cin >> s;

attendance\* currentattend = attendfptr;

while (currentattend != NULL)

{

if (currentattend->subject == s && currentattend->username == ::viewattendance)

{

currentattend->view\_attendance();

}

currentattend = currentattend->attendnptr;

}int j;

cout << endl << "Press 1 for for another subject Or Press 0 for Back.";

cin >> j;

if (j == 1) { goto gg; }

}

else if (en == 4)

{

assesment\* current\_asses = assesment\_fptr;

while (current\_asses != NULL)

{

if (current\_asses->return\_asses\_user() == ::viewattendance)

{

current\_asses->view\_assesment();

}

current\_asses = current\_asses->assesment\_nptr;

}

}

}

break;

}

}

int oop;

cout << endl << "Press 1 for Exit " << endl << "Press 2 for Mainmenu.:" << endl << "Press 3 for Login Portal agian." << endl;

cin >> oop;

if (oop == 1) { exit(0); }

else if (oop == 2) { mainmenu(); }

else { loginportal(); }

}

void storedata()

{

fstream afile, ffile, sfile, cfile, scdfile, attendfile, assessfile;

afile.open("Admin.txt", ios::out);

ffile.open("Faculty.txt", ios::out);

sfile.open("Student.txt", ios::out);

cfile.open("CourseData.txt", ios::out);

scdfile.open("Studentcoursedata.txt", ios::out);

attendfile.open("Attendance.txt", ios::out);

assessfile.open("Assesement.txt", ios::out);

Admin\* cadmin = faptr;

Faculty\_member\* cfaculty = ffptr;

student\* cstudent = fsptr;

course\* ccourse = firstcourse;

studentcoursedata\* cstudentcoursedata = scdfirst;

attendance\* cattendance = attendfptr;

assesment\* cassesment = assesment\_fptr;

while (cadmin != NULL)

{

afile << cadmin->name << endl << cadmin->username << endl << cadmin->password<< endl;

cadmin = cadmin->naptr;

}

while (cfaculty != NULL)

{

ffile << cfaculty->name << endl << cfaculty->re\_username() << endl << cfaculty->re\_password() << endl;

cfaculty = cfaculty->nfptr;

}

while (cstudent != NULL)

{

sfile << cstudent->stu\_name << endl << cstudent->re\_user() << endl << cstudent->re\_pass() << endl << cstudent->re\_roll() << endl;

cstudent = cstudent->nsptr;

}

while (ccourse != NULL)

{

string str = to\_string(ccourse->credithour);

string str2 = to\_string(ccourse->semester\_no);

cfile << ccourse->courseName<< endl << ccourse->teacher << endl << ccourse->code << endl << str<<

endl <<str2 << endl;

ccourse = ccourse->nextcourse;

}

while (cstudentcoursedata != NULL)

{

string str = to\_string(cstudentcoursedata->credithour);

string str2 = to\_string(cstudentcoursedata->semester\_no);

scdfile << cstudentcoursedata->courseName << endl << cstudentcoursedata->studnet\_username << endl << cstudentcoursedata->teacher

<< endl << cstudentcoursedata->code << endl <<str<< endl << str2 << endl;

cstudentcoursedata = cstudentcoursedata->scdnptr;

}

while (cattendance != NULL)

{

attendfile << cattendance->username << endl << cattendance->subject << endl << cattendance->attendance\_status << endl << cattendance->date

<< endl << cattendance->slot << endl;

cattendance = cattendance->attendnptr;

}

while (cassesment != NULL)

{

string str =to\_string(cassesment->assesment\_marks);

assessfile << cassesment->asses\_subject << endl << cassesment->assesment\_name << endl << str << endl << cassesment->assesment\_weightage

<< endl << cassesment->student\_username << endl;

cassesment = cassesment->assesment\_nptr;

}

afile.close();

ffile.close();

sfile.close();

cfile.close();

scdfile.close();

attendfile.close();

assessfile.close();

}

OutPuts:





