**Header Filer**

#pragma once

#include<iostream>

#include<string>

#include<fstream>

using namespace std;

class person {

protected:

string name;

string id;

string password;

public:

void writeonfile();

void Displayfile();

void setname(string a) {

name = a;

}

void setid(string a) {

id = a;

}

void setpassword(string a) {

password = a;

}

string getname() {

return name;

}

string getid() {

return id;

}

string getpass() {

return password;

}

};

class teacher :public person {

char tsection;

string tcourse;

friend class admin;

public:

char getsection() {

return tsection;

}

void settsection(char a) {

tsection = a;

}

string gettcourse() {

return tcourse;

}

void settcourse(string a) {

tcourse = a;

}

};

class student :public person {

int semester = 0;

float cgpa = 0.0;

char ssection;

string asssolution = "";

string quizsolution = "";

char assgrade = '\0';

char quizgrade = '\0';

public:

friend class admin;

void setsemester(int a) {

semester = a;

}

void setcgpa(float a) {

cgpa = a;

}

void setsection(char a) {

ssection = a;

}

int getsemester() {

return semester;

}

float getcgpa() {

return cgpa;

}

char getsection() {

return ssection;

}

void setasssoltion(string a) {

asssolution = a;

}

string getasssolution() {

return asssolution;

}

void setquizsolution(string a) {

quizsolution = a;

}

string getquiozsolution() {

return quizsolution;

}

void setassgrade(char a) {

assgrade = a;

}

char getassgrade() {

return assgrade;

}

void setquizgrade(char a) {

quizgrade = a;

}

char getquizgrade() {

return quizgrade;

}

};

class assignment {

char asssection;

string assname;

string asscourse;

string asscontent;

string quizname;

string quizcontent;

string quizcourse;

char quizsection;

public:

void setquizname(string a) {

quizname = a;

}

string getquizname() {

return quizname;

}

void setquizcontent(string a) {

quizcontent = a;

}

string getquizcontent() {

return quizcontent;

}

void setquizcourse(string a) {

quizcourse = a;

}

string getquizcourse() {

return quizcourse;

}

void setasssection(char a) {

asssection = a;

}

char getasssection() {

return asssection;

}

void setquizsection(char a) {

quizsection = a;

}

char getquizsection() {

return quizsection;

}

void setassname(string a) {

assname = a;

}

string getassname() {

return assname;

}

void setasscourse(string a) {

asscourse = a;

}

string getasscourse() {

return asscourse;

}

void setasscontent(string a) {

asscontent = a;

}

string getasscontent() {

return asscontent;

}

};

class course {

string cname;

int credithour;

int semester;

char csection;

friend class admin;

public:

void setcname(string a) {

cname = a;

}

void setcredithour(int a) {

credithour = a;

}

void setcsemester(int a) {

semester = a;

}

void setcsection(char a) {

csection = a;

}

char getcsection() {

return csection;

}

string getcname() {

return cname;

}

int getsemester() {

return semester;

}

int getcredithour() {

return credithour;

}

};

student stds[100];

course courses[100];

teacher teachers[100];

assignment assignments[100];

fstream sfile;

fstream tfile;

fstream afile;

class admin :public person {

public:

void setstdgpa(student& obj, float a) {

obj.cgpa = a;

}

void setstdsemester(student& obj, int a) {

obj.semester = a;

}

void setstdssection(student& obj, char a) {

obj.ssection = a;

}

void setstdid(student& obj, string a) {

obj.id = a;

}

void setstdname(student& obj, string a) {

obj.name = a;

}

void setstdpassword(student& obj, string a) {

obj.password = a;

}

int getstdsemester(student& obj) {

return obj.semester;

}

float getstdcgpa(student& obj) {

return obj.cgpa;

}

char getstdsection(student& obj) {

return obj.ssection;

}

void settecid(teacher& obj, string a) {

obj.id = a;

}

void settecname(teacher& obj, string a) {

obj.name = a;

}

string gettecname(teacher& obj) {

return obj.name;

}

string gettecid(teacher& obj) {

return obj.id;

}

string gettecpass(teacher& obj) {

return obj.password;

}

string getteccourse(teacher& obj) {

return obj.tcourse;

}

char gettecsectrion(teacher& obj) {

return obj.tsection;

}

void settecsection(teacher& obj, char a) {

obj.tsection = a;

}

void settecpass(teacher& obj, string a) {

obj.password = a;

}

void setteccourse(teacher& obj, string a) {

obj.tcourse = a;

}

string getcorname(course& obj) {

return obj.cname;

}

int getcorsemester(course& obj) {

return obj.semester;

}

int getcorcredithour(course& obj) {

return obj.credithour;

}

void setcorname(course& obj, string a) {

obj.cname = a;

}

void setcorcredithour(course& obj, int a) {

obj.credithour = a;

}

void setcorsemester(course& obj, int a) {

obj.semester = a;

}

void displayst(int totalst, int code) {

cout << "We have total " << totalst << " students and their details are: " << endl;

cout << "NAME\t\tID\t\tSEMESTER\t\tCGPA\t\tPASSWORD\t\tSECTION " << endl;

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

cout << stds[i].getname() << "\t\t" << stds[i].getid() << "\t\t" << stds[i].getsemester() << "\t\t" << stds[i].getcgpa() << "\t\t" << stds[i].getpass() << "\t\t" << stds[i].getsection() << endl;

cout << "\n";

}

}

void displaytc(int totaltc, int code) {

cout << "We have total " << totaltc << " teachers and their details are: " << endl;

cout << "Name\t\tid\t\tpassword\t\tSECTION" << endl;

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

cout << teachers[i].getname() << "\t\t" << teachers[i].getid() << "\t\t" << teachers[i].getpass() << "\t\t" << teachers[i].getsection() << endl;

cout << "\n";

}

}

void displaycc(int totalcc, int code) {

cout << "We have total " << totalcc << " courses and their details are: " << endl;

cout << "Name\t\tCredithour\t\tSemester" << endl;

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

cout << courses[i].getcname() << "\t\t" << courses[i].getcredithour() << "\t\t" << courses[i].getsemester() << endl;

cout << "\n";

}

}

void findcourse(int totalcc, int code, int& g, bool& df) {

cout << "Enter name of the course to deete " << endl;

string n;

cin >> n;

for (int k = 0; k < totalcc; k++) {

if (courses[k].getcname() == n) {

g = k;

df = true;

break;

}

}

}

};

void writeonstdfile(string a) {

sfile.open("STUDENTS.txt", ios::out | ios::app);

sfile << a << "\n";

sfile.close();

}

int checkonstdfile(string a) {

sfile.open("STUDENTS.txt", ios::out | ios::app);

string c;

int d = 0;

string arr[100];

int arr1 = 0;

while (getline(sfile, c)) {

arr[arr1] = c;

arr1 += 1;

}

for (int i = 0; i < 101; i++) {

if (arr[i] == a) {

d = 1;

}

}

return d;

}

admin admins[100];

string menu() {

system("cls");

cout << "\t\t\t\t----------------------------------------------\n";

cout << "\t\t\t\t\t Learning Management System \n";

cout << "\t\t\t\t\tWelcome to \"LMS\" PLEASE LOG IN : " << endl;

cout << "\t\t\t\t----------------------------------------------\n";

string a;

cout << "\t\t\t\tID: ";

cin >> a;

return a;

}

**Source file**

#include"Header.h"

int main() {

admins[0].setid("A1");

admins[0].setname("Abdul Rehman");

admins[0].setpassword("ar");

teachers[0].setid("T1");

teachers[0].setname("Ahmed");

teachers[0].setpassword("ah");

teachers[0].settsection('A');

teachers[0].settcourse("MATH101");

stds[0].setname("Shehram");

stds[0].setid("S1");

stds[0].setpassword("sh");

stds[0].setsemester(2);

stds[0].setcgpa(3.4);

stds[0].setsection('A');

stds[1].setname("Umair Ali");

stds[1].setid("S2");

stds[1].setpassword("um");

stds[1].setsemester(2);

stds[1].setcgpa(3.1);

stds[1].setsection('B');

courses[0].setcname("MATH101");

courses[0].setcredithour(3);

courses[0].setcsemester(2);

courses[0].setcsection('A');

courses[1].setcname("ENGLISH101");

courses[1].setcredithour(2);

courses[1].setcsemester(1);

courses[1].setcsection('A');

int totalst = 2;

int totaltc = 1;

int totalad = 1;

int totalcc = 2;

int totalass = 0;

bool x = false;

while (x == false) {

string a;

a = menu();

string b;

int code;

bool ch = true;

for (int i = 0; i < 100; i++) {

if (teachers[i].getid() == a) {

b = "teacher";

code = i;

ch = false;

break;

}

else if (stds[i].getid() == a) {

code = i;

ch = false;

b = "student";

break;

}

else if (admins[i].getid() == a) {

code = i;

ch = false;

b = "admin";

break;

}

}

if (ch == false) {

if (b == "admin") {

bool el = false;

cout << "\t\t\t\t----------------------------------------------\n";

cout << "\t\t\t\t\t Welcome \"ADMIN\" " << endl;

cout << "\t\t\t\t----------------------------------------------\n";

string c;

cout << "\t\t\t\tPASSWORD: ";

cin >> c;

while (el == false) {

if (admins[code].getpass() == c) {

cout << "\n\n\t\t\t\t----------------------------------------------\n";

cout << "\t\t\t\t\tWelcome " << admins[code].getname() << endl;

cout << "\t\t\t\t----------------------------------------------\n";

cout << "\n\n\t\t\t\t\tWhat do you want to do? " << endl;

cout << " \n\n\n\t\t1.Change details of a student\n\t\t2.Change details of teacher \n\t\t3.Add a teacher \n\t\t4.Add a student \n\t\t5.See details of students \n\t\t6.See details of teachers\n\t\t7.Check the detials of courses \n\t\t8.Add a course\n\t\t9.Delete a course \n\t\t10.LOG OUT of LMS \n\t\t11.End the program" << endl;

int f;

cin >> f;

if (f == 1) {

bool sd = false;

while (sd == false) {

bool dd = false;

cout << "Enter id of student to change the details : " << endl;

string sid;

cin >> sid;

for (int k = 0; k < totalst; k++) {

if (stds[k].getid() == sid) {

dd = true;

cout << "The student has \n\tName:" << stds[k].getname() << "\n\tID:" << stds[k].getid() << "\n\tSEMESTER: " << stds[k].getsemester() << "\n\tCGPA: " << stds[k].getcgpa() << "\n\tPASSWORD:" << stds[k].getpass() << "\n\tSECTION:" << admins[code].getstdsection(stds[k]) << endl;

cout << "What do you want to change: " << "\n1.Name\n2.Id\n3.Semester\n4.Cgpa\n5.Password\n6.Section" << endl;

int chng;

cin >> chng;

if (chng == 1) {

string before, change;

cout << "Enter new name for the student: " << endl;

cin >> change;

before = stds[k].getname();

admins[code].setstdname(stds[k], change);

cout << "The name has been changed!" << "\n Previos name: " << before << " Current name: " << stds[k].getname() << endl;

}

else if (chng == 2) {

string before, change;

cout << "Enter new ID for the student: " << endl;

cin >> change;

bool drt = false;

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

if (stds[i].getid() == change) {

drt = true;

}

}

if (drt == true) {

cout << "The id already exists!\n";

}

else if (drt == false) {

before = stds[k].getid();

admins[code].setstdid(stds[k], change);

cout << "The ID has been changed!" << "\n Previos ID: " << before << " Current ID: " << stds[k].getid() << endl;

}

}

else if (chng == 3) {

int before, change;

cout << "Enter new Semesterr for the student: " << endl;

cin >> change;

before = stds[k].getsemester();

admins[code].setstdsemester(stds[k], change);

cout << "The Semester has been changed!" << "\n Previos Semester: " << before << " Current Smester: " << stds[k].getsemester() << endl;

}

else if (chng == 4) {

float before, change;

cout << "Enter new CGPA for the student: " << endl;

cin >> change;

before = stds[k].getcgpa();

admins[code].setstdgpa(stds[k], change);

cout << "The CGPA has been changed!" << "\n Previos CGPA: " << before << " Current CGPA: " << stds[k].getcgpa() << endl;

}

else if (chng == 5) {

string before, change;

cout << "Enter new Password for the student: " << endl;

cin >> change;

before = stds[k].getpass();

admins[code].setstdpassword(stds[k], change);

cout << "The Password has been changed!" << "\n Previos password: " << before << " Current password: " << stds[k].getpass() << endl;

}

else if (chng == 6) {

char before, change;

cout << "Enter new Section for the student: " << endl;

cin >> change;

before = stds[k].getsection();

admins[code].setstdssection(stds[k], change);

cout << "The Section has been changed!" << "\n Previos section: " << before << " Current section: " << stds[k].getsection() << endl;

}

}

}

if (dd = false) {

cout << "The student is not found!" << endl;

cout << "Press 'y' to enter id again\nPress 'n' to go to main menu" << endl;

char p;

cin >> p;

if (p == 'y') {

continue;

}

else {

sd = true;

break;

}

}

cout << "Press 'y' to enter id of student again\nPress 'n' to go to main menu" << endl;

char p;

cin >> p;

if (p == 'y') {

continue;

}

else {

sd = true;

break;

}

}

}

if (f == 2) {

bool sd = false;

int ddty = 0;

while (sd == false) {

cout << "Enter id of teacher to change the details : " << endl;

string sid;

cin >> sid;

for (int k = 0; k < totalst; k++) {

if (teachers[k].getid() == sid) {

sd = true;

ddty = 1;

cout << "The teacher has Name:" << admins[code].gettecname(teachers[k]) << "\nID:" << admins[code].gettecid(teachers[k]) << "\nCourse: " << admins[code].getteccourse(teachers[k]) << "\nPASSWORD:" << admins[code].gettecpass(teachers[k]) << "\nSECTION:" << admins[code].gettecsectrion(teachers[k]) << endl;

cout << "What do you want to change: " << "\n1.Name\n2.Id\n3.Course\n4.Password\n5.Section" << endl;

int chng;

cin >> chng;

if (chng == 1) {

ddty = 1;

string before, change;

cout << "Enter new name for the teacher: " << endl;

cin >> change;

before = admins[code].gettecname(teachers[k]);

admins[code].settecname(teachers[k], change);

cout << "The name has been changed!" << "\n Previous name: " << before << " Current name: " << admins[code].gettecname(teachers[k]) << endl;

}

else if (chng == 2) {

ddty = 1;

string before, change;

cout << "Enter new ID for the teacher: " << endl;

cin >> change;

bool drt = false;

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

if (stds[i].getid() == change) {

drt = true;

}

}

if (drt == true) {

cout << "The id already exists!\n";

}

else if (drt == false) {

before = admins[code].gettecid(teachers[k]);

admins[code].settecid(teachers[k], change);

cout << "The ID has been changed!" << "\n Previos ID: " << before << " Current ID: " << admins[code].gettecid(teachers[k]) << endl;

}

}

else if (chng == 3) {

ddty = 1;

string before, change;

cout << "Enter new Course for the teacher: " << endl;

cin >> change;

before = admins[code].getteccourse(teachers[k]);

admins[code].setteccourse(teachers[k], change);

cout << "The Course has been changed!" << "\n Previous Course: " << before << " Current Course: " << admins[code].getteccourse(teachers[k]) << endl;

}

else if (chng == 4) {

ddty = 1;

string before, change;

cout << "Enter new Password for the teacher: " << endl;

cin >> change;

before = admins[code].gettecpass(teachers[k]);

admins[code].settecpass(teachers[k], change);

cout << "The Password has been changed!" << "\n Previos password: " << before << " Current password: " << admins[code].gettecpass(teachers[k]) << endl;

}

else if (chng == 5) {

ddty = 1;

char before, change;

cout << "Enter new Section for the student: " << endl;

cin >> change;

before = admins[code].gettecsectrion(teachers[k]);

admins[code].settecsection(teachers[k], change);

cout << "The Section has been changed!" << "\n Previos section: " << before << " Current section: " << admins[code].gettecsectrion(teachers[k]) << endl;

}

}

}

if (ddty == 0) {

cout << "The id you entered is not in our data base \n1.Enter again\n2.o to admin command\n";

int ppp;

cin >> ppp;

if (ppp == 1) {

continue;

}

else {

system("cls");

break;

}

}

cout << "\n\n\tPress 'y' to enter id of student again\n\tPress 'n' to go to main menu" << endl;

char p;

cin >> p;

if (p == 'y') {

continue;

}

else {

sd = true;

break;

}

}

}

else if (f == 4) {

cout << "Enter how many students you want to add: " << endl;

int j;

cin >> j;

bool sd = false;

while (true) {

for (int i = 1; i < j + 1; i++) {

bool df = false;

string n, id, p;

int s;

float gp;

char sce;

cout << "Enter id of new student " << i << endl;

cin >> id;

for (int k = 0; k < totalst; k++) {

if (stds[k].getid() == id) {

df = true;

break;

}

}

if (df == false) {

int g = totalst;

cout << "\tEnter name of new student " << i << endl;

cin >> n;

cout << "\tEnter semester of new student " << i << endl;

cin >> s;

cout << "\tEnter password of new student " << i << endl;

cin >> p;

cout << "\tEnter cgpa of new student " << i << endl;

cin >> gp;

cout << "\tEnter section of new student " << i << endl;

cin >> sce;

admins[code].setstdname(stds[g], n);

admins[code].setstdgpa(stds[g], gp);

admins[code].setstdid(stds[g], id);

admins[code].setstdpassword(stds[g], p);

admins[code].setstdsemester(stds[g], s);

admins[code].setstdssection(stds[g], sce);

totalst = totalst + 1;

sd = true;

cout << "The student has benn added successfully!" << endl;

cout << "\n";

}

else if (df = true) {

system("cls");

cout << "Student with same id already eixts" << endl;

cout << "You want to go back or enter data again ? (y for again n to go back)" << endl;

char w;

cin >> w;

if (w == 'y') {

continue;

}

else {

system("cls");

sd = true;

}

}

}

if (sd == true) {

cout << "Do you want to perform more operations or end program? \n 1.End the program \n 2.Go to main menu \n 3.Enter a admin command " << endl;

int w;

cin >> w;

if (w == 3) {

break;

}

else if (w == 1) {

system("cls");

cout << "The program has ended thankyou for using LMS" << endl;

x = true;

}

else {

el = true;

break;

}

}

}

}

else if (f == 9) {

bool sd = false;

while (true) {

int g;

bool df = false;

admins[code].findcourse(totalcc, code, g, df);

if (df == false) {

cout << "The course is not found!" << endl;

cout << "Do you want to perform more operations or end program? \n 1.End the program \n 2.Go to main menu \n 3.Search for course again " << endl;

int w;

cin >> w;

if (w == 3) {

continue;

}

else if (w == 1) {

system("cls");

cout << "The program has ended thankyou for using LMS" << endl;

x = true;

}

else {

el = true;

}

}

else {

cout << "The course has been deleted successfully!" << endl;

admins[code].setcorname(courses[g], "\0");

admins[code].setcorcredithour(courses[g], 0);

admins[code].setcorsemester(courses[g], 0);

cout << "Do you want to perform more operations or end program? \n 1.End the program \n 2.Go to main menu \n 3.Delete another course" << endl;

int w;

cin >> w;

if (w == 3) {

continue;

}

else if (w == 1) {

system("cls");

cout << "The program has ended thankyou for using LMS" << endl;

x = true;

}

else {

el = true;

break;

}

}

}

}

else if (f == 3) {

cout << "Enter how many teachers you want to add: " << endl;

int j;

cin >> j;

bool sd = false;

while (true) {

for (int i = 1; i < j + 1; i++) {

bool df = false;

string n, id, p;

cout << "Enter id of new teacher " << i << endl;

cin >> id;

for (int k = 0; k < totalst; k++) {

if (teachers[k].getid() == id) {

df = true;

break;

}

}

if (df == false) {

int g = totaltc;

cout << "Enter name of new teacher " << i << endl;

cin >> n;

cout << "Enter password of new student " << i << endl;

cin >> p;

cout << "Enter the course teacher will be teaching: " << endl;

string tcc;

cin >> tcc;

cout << "Enter the section the teacher will be teaching to: " << endl;

char tccu;

cin >> tccu;

teachers[g].settsection(tccu);

teachers[g].settcourse(tcc);

teachers[g].setname(n);

teachers[g].setid(id);

teachers[g].setpassword(p);

totaltc = totaltc + 1;

sd = true;

cout << "The teacher has benn added successfully!" << endl;

cout << "\n";

}

else if (df = true) {

system("cls");

cout << "Teacher with same id already eixts" << endl;

cout << "You want to go back or enter data again ? (y for again n to go back)" << endl;

char w;

cin >> w;

if (w == 'y') {

continue;

}

else {

system("cls");

sd = true;

}

}

}

if (sd == true) {

cout << "Do you want to perform more operations or end program? \n 1.End the program \n 2.Go to main menu \n 3.Enter a admin command " << endl;

int w;

cin >> w;

if (w == 3) {

break;

}

else if (w == 1) {

system("cls");

cout << "The program has ended thankyou for using LMS" << endl;

x = true;

}

else {

el = true;

break;

}

}

}

}

else if (f == 8) {

cout << "Enter how many courses you want to add: " << endl;

int j;

cin >> j;

bool sd = false;

while (true) {

for (int i = 1; i < j + 1; i++) {

bool df = false;

string n;

int sem, cr;

cout << "Enter name of new course " << i << endl;

cin >> n;

for (int k = 0; k < totalcc; k++) {

if (admins[code].getcorname(courses[k]) == n) {

df = true;

break;

}

}

if (df == false) {

int g = totalcc;

cout << "Enter credithour of new course " << i << endl;

cin >> cr;

cout << "Enter semester to offer new course to " << i << endl;

cin >> sem;

admins[code].setcorname(courses[g], n);

admins[code].setcorcredithour(courses[g], cr);

admins[code].setcorsemester(courses[g], sem);

totalcc = totalcc + 1;

sd = true;

cout << "The course has benn added successfully!" << endl;

cout << "\n";

}

else if (df = true) {

system("cls");

cout << "Course with same name already eixts" << endl;

cout << "You want to go back or enter data again ? (y for again n to go back)" << endl;

char w;

cin >> w;

if (w == 'y') {

continue;

}

else {

system("cls");

sd = true;

}

}

}

if (sd == true) {

cout << "Do you want to perform more operations or end program? \n 1.End the program \n 2.Go to main menu \n 3.Enter a admin command " << endl;

int w;

cin >> w;

if (w == 3) {

break;

}

else if (w == 1) {

system("cls");

cout << "The program has ended thankyou for using LMS" << endl;

x = true;

}

else {

el = true;

break;

}

}

}

}

else if (f == 5) {

admins[0].displayst(totalst, code);

}

else if (f == 6) {

admins[code].displaytc(totaltc, code);

}

else if (f == 7) {

admins[code].displaycc(totalcc, code);

}

else if (f == 10) {

system("cls");

el = true;

}

else if (f == 11) {

system("cls");

x = true;

cout << "Thankyou for using LMS! " << endl;

break;

}

}

else {

cout << "The password you entered is wrong \n Please write again!" << endl;

cin >> c;

continue;

}

cout << "\n\n\t\tPlease choose one option \n\t1.Go to admin commands\n\t2.LOG OUT of LMS\n";

int ppt;

cin >> ppt;

if (ppt == 2) {

break;

}

else {

continue;

}

}

}

else if (b == "teacher") {

cout << "\t\t\t\t----------------------------------------------\n";

cout << "\t\t\t\t\t Welcome \"TEACHER\" " << endl;

cout << "\t\t\t\t----------------------------------------------\n";

bool el = false;

cout << "\n\n\n\t\t\t\t\tEnter Password: ";

string c;

cin >> c;

while (el == false) {

if (teachers[code].getpass() == c) {

cout << "\n\n\t\t\t\t----------------------------------------------\n";

cout << "\t\t\t\t\tWelcome " << teachers[code].getname() << endl;

cout << "\t\t\t\t----------------------------------------------\n";

cout << "\n\n\t\t\t\t\tWhat do you want to do? " << endl;

cout << "\n\n\n\t\t1.Check the list of enrolled students \n\t\t2.Give students assignments \n\t\t3.Give quiz to students \n\t\t4.Go to main menue\n\t\t5.Grade the submitted assignments\n\t\t6.Grade the submitted quizzez\n\t\t7.Logout of LMS\n\t\t8.End program" << endl;

int f;

cout << "\t\t";

cin >> f;

if (f == 1) {

int totalenrolled = 0;

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

if (admins[0].getstdsection(stds[i]) == teachers[code].getsection()) {

totalenrolled = totalenrolled + 1;

}

}

cout << "You have total " << totalenrolled << " students and their details are: " << endl;

cout << "NAME\t\tID\t\tSEMESTER\t\tCGPA\t\tSECTION " << endl;

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

if (admins[0].getstdsection(stds[i]) == teachers[code].getsection()) {

cout << stds[i].getname() << "\t\t" << stds[i].getid() << "\t\t " << stds[i].getsemester() << "\t\t" << stds[i].getcgpa() << "\t\t" << admins[code].getstdsection(stds[i]) << endl;

cout << "\n";

}

}

}

else if (f == 7) {

el = true;

break;

}

else if (f == 8) {

cout << "\n\n\t\t\t\tThank you for using LMS\n";

x = true;

break;

}

else if (f == 5) {

int ass = 0;

for (int i = 0; i < totalass + 1; i++) {

if (assignments[i].getasssection() == teachers[code].getsection()) {

ass += 1;

}

}

if (ass > 0) {

bool agg = false;

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

if (admins[0].getstdsection(stds[i]) == teachers[code].getsection()) {

if (stds[i].getasssolution() != "") {

system("cls");

agg = true;

cout << "Student name " << stds[i].getname() << " has uploaded the assignment: " << endl;

cout << "The solution is: " << stds[i].getasssolution() << endl;

cout << "Please enter a grade for the assignment" << endl;

char assg;

cin >> assg;

stds[i].setassgrade(assg);

cout << "The grade has been successfully updated!" << endl;

}

}

}

if (agg == false) {

cout << "No student has yet uploaded the assignment! " << endl;

}

}

else {

cout << "You have not uploaded assignmment for your course! " << endl;

}

cout << "What do you want to do next \n1.To go to main menu \n To exit the program 2 \n To enter id again press 3" << endl;

int p2;

cin >> p2;

if (p2 == 1) {

continue;

}

else if (p2 == 3) {

el = true;

continue;

}

else if (p2 == 2) {

x = true;

cout << "Thankyou for using LMS" << endl;

break;

}

}

else if (f == 2) {

cout << "Enter The name of assignment: " << endl;

string ass;

cin >> ass;

assignments[totalass].setassname(ass);

assignments[totalass].setasssection(teachers[code].getsection());

assignments[totalass].setasscourse(teachers[code].gettcourse());

cout << "Enter the assignemnt you want to give to students: " << endl;

string ass1;

cin >> ass1;

assignments[totalass].setasscontent(ass1);

totalass += 1;

cout << "The assignments has benn uploaded successfully!" << endl;

}

else if (f == 3) {

cout << "Enter The name of Quiz: " << endl;

string ass;

cin >> ass;

assignments[totalass].setquizname(ass);

assignments[totalass].setquizsection(teachers[code].getsection());

assignments[totalass].setquizcourse(teachers[code].gettcourse());

cout << "Enter the Quiz you want to give to students: " << endl;

string ass1;

cin >> ass1;

assignments[totalass].setquizcontent(ass1);

totalass += 1;

cout << "The quiz has benn uploaded successfully!" << endl;

}

else if (f == 4) {

system("cls");

el = true;

break;

}

else if (f == 6) {

int ass = 0;

for (int i = 0; i < totalass + 1; i++) {

if (assignments[i].getquizsection() == teachers[code].getsection()) {

ass += 1;

}

}

if (ass > 0) {

bool agg = false;

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

if (admins[0].getstdsection(stds[i]) == teachers[code].getsection()) {

if (stds[i].getquiozsolution() != "") {

system("cls");

agg = true;

cout << "Student name " << stds[i].getname() << " has uploaded the Quiz: " << endl;

cout << "The solution is: " << stds[i].getquiozsolution() << endl;

cout << "Please enter a grade for the Quiz" << endl;

char assg;

cin >> assg;

stds[i].setquizgrade(assg);

cout << "The grade has been successfully updated!" << endl;

}

}

}

if (agg == false) {

cout << "No student has yet attempted the quiz! " << endl;

}

}

else {

cout << "You have not uploaded Quiz for your course! " << endl;

}

cout << "What do you want to do next \n1.To go to main menu \n To exit the program 2 \n To enter id again press 3" << endl;

int p2;

cin >> p2;

if (p2 == 1) {

continue;

}

else if (p2 == 3) {

el = true;

continue;

}

else if (p2 == 2) {

x = true;

cout << "Thankyou for using LMS" << endl;

break;

}

}

}

else {

cout << "The password you entered is wrong \n1.To enter again \n2.To exit the program \n3.To enter id again " << endl;

int p2;

cin >> p2;

if (p2 == 1) {

cout << "Please write your password again: " << endl;

continue;

}

else if (p2 == 3) {

el = true;

continue;

}

else if (p2 == 2) {

x = true;

cout << "Thankyou for using LMS" << endl;

break;

}

}

cout << "What do you want to do next \n1.To go to main menu \n To exit the program 2 \n To enter id again press 3" << endl;

int p2;

cin >> p2;

if (p2 == 1) {

continue;

}

else if (p2 == 3) {

el = true;

continue;

}

else if (p2 == 2) {

x = true;

system("cls");

cout << "\n\n\t\t\t\t\t\tThankyou for using LMS\n\n\n\n" << endl;

break;

}

}

}

else if (b == "student") {

bool el = false;

while (el == false) {

bool loop = false;

cout << "\t\t\t\t----------------------------------------------\n";

cout << "\t\t\t\t\t Welcome \"STUDENT\" " << endl;

cout << "\t\t\t\t----------------------------------------------\n";

cout << "\n\n\n\t\t\t\t\tEnter Password: ";

string c;

cin >> c;

if (stds[code].getpass() == c) {

cout << "\n\n\t\t\t\t----------------------------------------------\n";

cout << "\t\t\t\t\tWelcome " << stds[code].getname() << endl;

cout << "\t\t\t\t----------------------------------------------\n";

cout << "\n\n\t\t\t\t\tWhat do you want to do? " << endl;

cout << "\t\t1.See the details of you registered courses\n\t\t2.See assignments\n\t\t3.See quizzez\n\t\t4.See the grade of assignments\n\t\t5.See the grade of quizzez\n\t\t6.Logout of LMS\n\t\t7.End the program" << endl;

int stt;

cout << "\t\t";

cin >> stt;

int ass = 0;

int cours = 0;

if (stt == 1) {

for (int i = 0; i < totalass + 1; i++) {

if (assignments[i].getasssection() == stds[code].getsection()) {

ass += 1;

}

}

for (int i = 0; i < totalcc + 1; i++) {

if (courses[i].getcsection() == stds[code].getsection()) {

cours += 1;

}

}

cout << "You have total " << cours << " Courses" << " and " << ass << " of them have assignments" << endl;

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

if (courses[i].getcsection() == stds[code].getsection()) {

cout << "NAME: " << courses[i].getcname() << "\tCREDIT HOURS: " << courses[i].getcredithour() << endl;

}

}

}

else if (stt == 6) {

loop = true;

break;

}

else if (stt == 7) {

cout << "\n\n\t\t\t\tThank you for using LMS\n";

x = true;

break;

}

else if (stt == 4) {

for (int i = 0; i < totalass + 1; i++) {

if (assignments[i].getasssection() == stds[code].getsection()) {

ass += 1;

}

}

if (ass > 0) {

for (int i = 0; i < totalass + 1; i++) {

if (assignments[i].getasssection() == stds[code].getsection()) {

if (stds[code].getassgrade() != '\0') {

cout << "You have been graded " << stds[code].getassgrade() << " Against your assignment " << assignments[i].getassname() << " For the course " << assignments[i].getasscourse() << endl;

}

else {

cout << "Your assignment has not yet been graded!" << endl;

}

}

}

}

else {

cout << "Sorry you have no assignments!" << endl;

}

}

else if (stt == 5) {

for (int i = 0; i < totalass + 1; i++) {

if (assignments[i].getasssection() == stds[code].getsection()) {

ass += 1;

}

}

if (ass > 0) {

for (int i = 0; i < totalass + 1; i++) {

if (assignments[i].getasssection() == stds[code].getsection()) {

if (stds[code].getquizgrade() != '\0') {

cout << "You have been graded " << stds[code].getquizgrade() << " Against your quiz " << assignments[i].getquizname() << " For the course " << assignments[i].getquizcourse() << endl;

}

else {

cout << "Your Quiz has not yet been graded!" << endl;

}

}

}

}

else {

cout << "Sorry you have no Quiz!" << endl;

}

}

else if (stt == 2) {

for (int i = 0; i < totalass + 1; i++) {

if (assignments[i].getasssection() == stds[code].getsection()) {

ass += 1;

}

}

cout << "You have total " << ass << " assignments" << endl;

if (ass > 0) {

cout << "The courses with assignemnts are: " << endl;

for (int j = 0; j < totalass + 1; j++) {

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

if (assignments[j].getasscourse() == courses[i].getcname()) {

cout << "\tName: " << courses[i].getcname() << "\tCredit hours: " << courses[i].getcredithour() << "\tHave assignment name" << assignments[j].getassname() << endl;

}

}

}

cout << "Enter the name of the ASSIGNMENT to see assignment: " << endl;

string ass2;

cin >> ass2;

bool as2 = false;

for (int i = 0; i < totalass + 1; i++) {

if (assignments[i].getassname() == ass2) {

as2 = true;

cout << "The name of assignmment is: " << assignments[i].getassname() << "The content is: " << assignments[i].getasscontent() << endl;

cout << "Do you want to upload a solution? " << endl;

char bg;

cout << "Enter Y for yes and N for no" << endl;

cin >> bg;

if (bg == 'Y' || bg == 'y') {

cout << "Enter your solution: " << endl;

string abf;

cin >> abf;

stds[code].setasssoltion(abf);

cout << "Your solution has been added successfully!" << endl;

cout << "What do you want to do next" << endl;

cout << "1.Exit the program\n2.Perform other operations\n3.Logout of the LMS" << endl;

int abg;

bool ngt = false;

while (ngt = false) {

cin >> abg;

if (abg == 1) {

ngt = true;

x = true;

loop = true;

system("cls");

cout << "Thankyou for using LMS" << endl;

break;

}

else if (abg == 2) {

ngt = true;

break;

}

else if (abg == 3) {

loop = true;

ngt = true;

break;

}

else {

cout << "Wrong input please write again! " << endl;

}

}

}

else if (bg == 'N' || bg == 'n') {

cout << "What do you want to do next" << endl;

cout << "1.Exit the program\n2.Perform other operations\n3.Logout of the LMS" << endl;

int abg;

bool ngt = false;

while (ngt = false) {

cin >> abg;

if (abg == 1) {

ngt = true;

x = true;

el = true;

system("cls");

cout << "Thankyou for using LMS" << endl;

break;

}

else if (abg == 2) {

ngt = true;

break;

}

else if (abg == 3) {

loop = true;

el = true;

ngt = true;

break;

}

else {

cout << "Wrong input please write again! " << endl;

}

}

}

}

}

if (as2 = false) {

cout << "The assignment is not found: " << endl;

cout << "What do you want to do next" << endl;

cout << "1.Exit the program\n2.Perform other operations\n3.Logout of the LMS" << endl;

int abg;

bool ngt = false;

while (ngt = false) {

cin >> abg;

if (abg == 1) {

ngt = true;

x = true;

el = true;

system("cls");

cout << "Thankyou for using LMS" << endl;

break;

}

else if (abg == 2) {

ngt = true;

break;

}

else if (abg == 3) {

loop = true;

el = true;

ngt = true;

break;

}

else {

cout << "Wrong input please write again! " << endl;

}

}

}

}

}

else if (stt == 3) {

cout << totalass;

for (int i = 0; i < totalass + 1; i++) {

if (assignments[i].getquizsection() == stds[code].getsection()) {

ass += 1;

}

}

cout << "You have total " << ass << " Quizez" << endl;

if (ass > 0) {

cout << "The courses with Quiz are: " << endl;

for (int j = 0; j < totalass + 1; j++) {

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

if (assignments[j].getquizcourse() == courses[i].getcname()) {

cout << "Name: " << courses[i].getcname() << "Credit hours: " << courses[i].getcredithour() << " Have Quiz name" << assignments[j].getquizname() << endl;

}

}

}

cout << "Enter the name of the quiz to see quiz: " << endl;

string ass2;

cin >> ass2;

bool as2 = false;

for (int i = 0; i < totalass + 1; i++) {

if (assignments[i].getquizname() == ass2) {

as2 = true;

cout << "The name of Quiz is: " << assignments[i].getquizname() << "The content is: " << assignments[i].getquizcontent() << endl;

cout << "Do you want to upload a solution? " << endl;

char bg;

cout << "Enter Y for yes and N for no" << endl;

cin >> bg;

if (bg == 'Y' || bg == 'y') {

cout << "Enter your solution: " << endl;

string abf;

cin >> abf;

stds[code].setquizsolution(abf);

cout << "Your solution has been added successfully!" << endl;

cout << "What do you want to do next" << endl;

cout << "1.Exit the program\n2.Perform other operations\n3.Logout of the LMS" << endl;

int abg;

bool ngt = false;

while (ngt = false) {

cin >> abg;

if (abg == 1) {

ngt = true;

x = true;

loop = true;

system("cls");

cout << "Thankyou for using LMS" << endl;

break;

}

else if (abg == 2) {

ngt = true;

break;

}

else if (abg == 3) {

loop = true;

ngt = true;

break;

}

else {

cout << "Wrong input please write again! " << endl;

}

}

}

else if (bg == 'N' || bg == 'n') {

cout << "What do you want to do next" << endl;

cout << "1.Exit the program\n2.Perform other operations\n3.Logout of the LMS" << endl;

int abg;

bool ngt = false;

while (ngt = false) {

cin >> abg;

if (abg == 1) {

ngt = true;

x = true;

el = true;

system("cls");

cout << "Thankyou for using LMS" << endl;

break;

}

else if (abg == 2) {

ngt = true;

break;

}

else if (abg == 3) {

loop = true;

el = true;

ngt = true;

break;

}

else {

cout << "Wrong input please write again! " << endl;

}

}

}

}

}

if (as2 = false) {

cout << "The Quiz is not found: " << endl;

cout << "What do you want to do next" << endl;

cout << "1.Exit the program\n2.Perform other operations\n3.Logout of the LMS" << endl;

int abg;

bool ngt = false;

while (ngt = false) {

cin >> abg;

if (abg == 1) {

ngt = true;

x = true;

el = true;

system("cls");

cout << "Thankyou for using LMS" << endl;

break;

}

else if (abg == 2) {

ngt = true;

break;

}

else if (abg == 3) {

loop = true;

el = true;

ngt = true;

break;

}

else {

cout << "Wrong input please write again! " << endl;

}

}

}

}

}

}

else {

cout << "The password you entered is wrong \n Please write again!" << endl;

}

cout << "Do you want to end the program or enter again(y to enter end program n to enter command again\n";

char aty;

cin >> aty;

if (aty == 'y') {

x = true;

system("cls");

cout << "\n\n\n\t\t\t\tThank you for using LMS \n\n\n\n";

break;

}

else {

continue;

}

}

}

}

else if (ch == true) {

cout << "The id is not found!" << endl;

cout << "Do you want to enter id again or end program? (Enter small y for yes and small n for no) " << endl;

char w;

cin >> w;

if (w == 'y') {

continue;

}

else {

system("cls");

cout << "The program has ended thankyou for using LMS" << endl;

x = true;

}

}

string names;

names = "STUDENTS.txt";

sfile.open(names, ios::out | ios::app);

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

sfile << stds[i].getid() << "\n";

}

sfile.close();

int fileint = checkonstdfile("S1");

}

system("pause");

return 0;

}