

المشروع النهائي
نظام جامعي



توثيق مشروع النظام جامعي

تم تطوير هذا النظام بلغة ++c لإدارة عمليات القبول والتسجيل في الجامعة ويتضمن الميزات الآتية

١. المكاتب المستخدمة في البرنامج

```
#include <iostream>
#include <string>
#include <vector>
#include <map>
// #include <limits>
```

استخدمنا مكتبة الإدخال والإخراج مكتبة السلاسل النصية ومكتبة vector
تقوم بتخزين البيانات بداخلها على شكل مصفوفات و مكتبة map
تخزن البيانات فيها بشكل مرتب تصاعدي

```
using namespace std;
```

٢. هياكل البيانات

١. تركيب خاص في التخصصات الخاصة بالجامعة ويحتوي على

```
struct Specialization {
    string name;
    string description;
    int duration;
    vector<string> courses;
```

- اسم التخصص
- وصف التخصص
- مدة الدراسة
- قائمة المقررات الدراسية

٢. تركيب خاص في كليات الجامعات ويحتوي على

```
struct College {
    string name;
    map<string, string> admissionRequirements;
    vector<Specialization> specializations;
```

```
};
```

- اسم الكلية
- متطلبات الكلية (خريطة من نوع مفتاح و قيمة)
- قائمة التخصصات المتاحة

٣. تركيب خاص في بيانات الطلاب ويحتوي على

```
struct Student {
    string name;
    string id;
    string college;
    string specialization;
    bool parallelProgram;
```

```
};
```

- اسم الطالب
- الرقم الجامعي
- الكلية المسجل بها
- التخصص المختار
- النظام الموازي

٣. قواعد بيانات النظام

استخدمنا قاعدتين بيانات لتخزين البيانات

```
vector<College> colleges;  
vector<Student> registeredStudents;
```

١. قاعدة بيانات خاصة بالكليات
٢. قاعدة بيانات خاصة بتخزين بيانات الطالب

٤. دالة لتصفية البيانات

قمنا بتهيئة النظام عن طريق إنشاء كليتين من نوع التركيب الخاص بالكليات

١. كلية الحاسبات

استخدمنا المتغير (cs) لتسمية الكلية عن طريق الاتي:

- اسم الكلية " كلية الحاسبات"
- متطلبات القبول {"الثانوية العامة، المعدل على الأقل ٨٠%"} {"اختبار القبول، المعدل على الأقل ٧٥%"} {" "}

❖ التخصص الأول لكلية الحاسبات

رمزنا له بالرمز (**csspec1**) ويحتوي علي

- اسم التخصص " علوم الحاسب "
- وصف التخصص " هو برنامج يغطي اساسيات البرمجة و هياكل البيانات والذكاء الصناعي "
- مدة الدراسة= ٤ سنوات
- المواد الأساسية" البرمجة و هياكل البيانات و قاعدة البيانات

❖ التخصص الثاني لكلية الحاسب

رمزنا له بالرمز (csspec2) ويحتوي علي

- اسم التخصص " هندسة البرمجيات "
- وصف التخصص " هو برنامج يركز على تطوير البرمجيات وإدارتها "
- مدة الدراسة = ٤ سنوات
- المواد الأساسية " هندسة البرمجيات و تطوير الويب و إدارة المشاريع..... "

وبعد التهيئة قمنا بإضافة البيانات الى قائمة التخصصات

push_back باستخدام دالة

وبعد ذلك نقوم بإضافة جميع البيانات الى قاعدة البيانات الخاصة بالكليات باستخدام نفس الدالة

Push_back

```
colleges.push_back(cs);
```

```
College eng;
```

```
eng.name = "College of  
Engineering";
```

```
eng.admissionRequirements = {  
    {"High School", "Minimum 85% in  
scientific track"},  
    {"Aptitude Test", "Minimum  
80%"},
```

٢. كلية الهندسة

استخدمنا المتغير (eng) لتسمية الكلية عن طريق الاتي:

- اسم الكلية " كلية هندسة الحاسوب"
- متطلبات القبول {"الثانوية العامة, المعدل على الأقل ٨٥%"} {"اختبار القبول, المعدل على الأقل ٨٠%"} {"Aptitude Test", "Minimum 80%"},

❖ التخصص الأول لكلية الهندسة

```
Specialization engSpec1;
```

```
engSpec1.name = "Computer Engineering";  
engSpec1.description = "Program combining  
electrical engineering and computer  
science\n\n";  
engSpec1.duration = 5;  
engSpec1.courses = {"Electrical Circuits",  
"Digital Electronics", "Signal Processing",  
"Computer Networks\n\n"};
```

رمزنا له بالرمز (engspec1) ويحتوي علي

- اسم التخصص "هندسة الحاسوب"
- وصف التخصص " هو برنامج يجمع بين هندسة الإلكترونيات و علوم الحاسوب"
- مدة الدراسة = ٥ سنوات
- المواد الأساسية " الدوائر الإلكترونية و الإلكترونيات الرقمية ومعالجة الإشارات و شبكات الحاسوب...."

❖ التخصص الثاني لكلية الهندسة

رمزنا له بالرمز (engspec2) ويحتوي علي

- اسم التخصص " الهندسة الصناعية"
- وصف التخصص " هو برنامج يركز على تحسين النظم والعمليات"
- مدة الدراسة = ٥ سنوات
- المواد الأساسية " بحوث العمليات و نظم التصنيع و هندسة العوامل البشرية و إدارة الجودة....."

```
Specialization engSpec2;
```

```
engSpec2.name = "Industrial Engineering";  
engSpec2.description = "Program focused on  
systems and process optimization\n\n";  
engSpec2.duration = 5;  
engSpec2.courses = {"Operations Research",  
"Human Factors Engineering", "Manufacturing  
Systems", "Quality Management\n\n"};
```

وبعد التهيئة قمنا بإضافة البيانات الى قائمة التخصصات

```
eng.specializations.push_back(engSpec1);
```

```
eng.specializations.push_back(engSpec2);
```

باستخدام دالة push_back

وبعد ذلك نقوم بإضافة جميع البيانات الى قاعدة

البيانات الخاصة بالكليات باستخدام نفس الدالة

```
colleges.push_back(eng);
```

Push_back

واستخدمنا دالة cin.ignore()

لتنظيف الإدخال من الأحرف الزائدة ومنع حدوث أخطاء في التبديل بين cin و getline() وضمان

قراءة الإدخال بشكل صحيح من المستخدم دون سلوك غير متوقع

٥. دالة عرض القائمة الرئيسية

هذه الدالة نقوم من خلالها بعرض لقائمة التي تمكنا من الوصول الى أي جزء من النظام الجامعي

١. عرض التخصصات

٢. متطلبات القبول

٣. عملية التسجيل

٤. النظام الموازي

٥. عرض بيانات الطلاب

٦. الخروج

```
void showMainMenu() {
    cout << "      University Admission System\n";
    cout << "===== \n";
    cout << "      1. View Available Specializations\n";
    cout << "      2. Admission Requirements\n";
    cout << "      3. Registration Process\n";
    cout << "      4. Parallel Program Information\n";
    cout << "      5. View Registered Students\n";
    cout << "      6. Exit\n";
    cout << "===== \n";
    cout << "      Enter your choice: ";
}
```

٦. دالة عرض التخصصات المتاحة

هذه الدالة تعرض جميع الكليات والتخصصات مع تفاصيل كل تخصص

عملنا دالة قمنا بتعريف داخلها

متغير (college) لتخزين اسم

الكلية المأخوذة من قاعدة البيانات (colleges)

```
void showSpecializations() {
    cout << "Available University Specializations:\n";

    for (const College& college : colleges) {
        cout << "\nCollege: " << college.name << "\n\n";
        cout << "Specializations:\n\n";

        for (const Specialization& spec :
            college.specializations) {
            cout << "- " << spec.name << " (" <<
                spec.duration << " years)\n";
            cout << "      *Description*: " <<
                spec.description << "\n";
            cout << "      *Core Courses*: ";

            for (const string& course : spec.courses) {
                cout << course << ", ";
            }
            cout << "\n";
        }
    }

    cout << "\nPress any key to return to main menu...";
    cin.ignore();
    cin.get();
}
```

وعرفنا ايضا دالة بداخل الدالة الاولى

وقمنا بتعريف متغير (spec) داخلها

لتخزين اسم و وصف التخصص ومدة دراسته

من قائمة التخصصات

وأیضا عرفنا دالة خاصة بالمواد بداخلها متغير

(course) لتخزين المواد من قائمة التخصص

١. دالة عرض متطلبات القبول

هذه الدالة تظهر كل متطلبات القبول لكل كلية

```
void showAdmissionRequirements() {  
    cout << "College Admission Requirements:\n\n";
```

عملنا دواراً قمنا بتعريف داخلها

```
    for (const College& college : colleges) {  
        cout << "\nCollege: " << college.name << "\n";  
        cout << "        *Requirements*:\n";
```

متغير (college) لتخزين اسم الكلية

```
        for (const auto& req : college.admissionRequirements) (colleges) المأخوذة من قاعدة البيانات
```

```
    {  
        cout << "- " << req.first << ": " << req.second  
        << "\n";  
    }  
}
```

عرفنا دواراً لعرض المتطلبات وعرفنا

بداخلها متغير من نوع auto

وخزنا بداخله المتطلبات

```
    cout << "\n*Note*\n: Requirements may change according to  
university policy each year.\n";  
    cout << "\nPress any key to return to main menu...";  
    cin.ignore();  
    cin.get();
```

وأيضاً طبعنا ملاحظة تنص على أن الشروط تتغير من كلية إلى أخرى

```
}
```

معلومات عن MAP , AUTO

Map: هي حاوية تخزن البيانات في أزواج (مفتاح و قيمة) في نظامنا سنخزن شروط القبول حيث المفتاح هو اسم الشرط والقيمة في التفاصيل.

الوصول إلى العناصر في map:

هناك طريقتين للوصول ١. الوصول باستخدام معامل [] , ٢. الوصول باستخدام المعامل () .at()

ال map في c++ يحفظ جميع العناصر مرتبة حسب المفتاح (بترتيب تصاعدي افتراضياً) مما يجعل البحث والوصول إلى العناصر سريعاً

مميزات loop في map: ١. الإيجاز/ تقليل عدد الكود المطلوب ٢. الأمان/ تجنب الأخطاء الشائعة في استخدام الكود.

استخدام auto في ال map:

تستخدم لاستنتاج نوع عناصر ال map تلقائياً في حالة map<string,string>

وبدون ال auto يمكن كتابة الكود يدوياً (for (const pair<string, const string>&req: admission...))

الفائدة من auto

يختصر الكود ويجعله أكثر قراءة وأيضاً يتجنب الأخطاء في كتابة النوع يدوياً

١. دالة لعملية التسجيل

في هذه الدالة نقوم بتسجيل الطالب في الجامعة من خلال الاتي

```
void registrationProcess() {
    cout << "Registration Process\n";
    if (colleges.empty()) {
        cout << "No colleges available for registration at this
time.\n";
        cout << "\nPress any key to return to main menu...";
        cin.ignore();
        cin.get();
        return;
    }
    Student newStudent;
    cout << "Full Name: ";
    getline(cin, newStudent.name);
    cout << "University ID: ";
    getline(cin, newStudent.id);
```

١. توجيه الطالب لإدخال البيانات الشخصية

في هذا الكود يتحقق في البداية من

وجود كليات متاحة إذا لم يكن هناك كليات

متاحة يطبع "لا يوجد كليات متاحة"

ثم يقوم بإدخال اسم الطالب ورقمه الجامعي

```
cout << "\nAvailable Colleges:\n";
for (int i = 0; i < colleges.size(); i++) {
    cout << i+1 << ". " << colleges[i].name << "\n";
}
int collegeChoice;
cout << "Select college (number): ";
cin >> collegeChoice;
if (collegeChoice < 1 || collegeChoice > colleges.size()) {
    cout << "Invalid selection!\n";
    cout << "\nPress any key to return to main menu...";
    cin.ignore();
    cin.get();
    return;
}
newStudent.college = colleges[collegeChoice-1].name;
```

٢. اختيار الطالب للكلية

استخدمنا دالة لاختيار كلية محدد في النظام

عرفنا متغير للاختيار

وعملنا شرط ينص على ان المستخدم

إذا ادخل رقم اقل من ١ او رقم اكبر

من حجم قاعدة البيانات الخاصة بالكليات

يطبع "الاختيار غير متوفر"

إذا لم يتحقق الشرط يقوم النظام بإضافة الكلية

المختارة الى سجل الطالب

```
cout << "\nSpecializations available in " <<
newStudent.college << ":\n";
for (int i = 0; i < colleges[collegeChoice-
1].specializations.size(); i++) {
    cout << i+1 << ". " << colleges[collegeChoice-
1].specializations[i].name << "\n";
}
int specChoice;
cout << "Select specialization (number): ";
cin >> specChoice;
if (specChoice < 1 || specChoice >
colleges[collegeChoice-1].specializations.size()) {
    cout << "Invalid selection!\n";
    cout << "\nPress any key to return to main menu...";
    cin.ignore();
    cin.get();
    return;
}
newStudent.specialization = colleges[collegeChoice-
1].specializations[specChoice-1].name;
```

٣. اختيار الطالب للتخصص

استخدمنا دالة لاختيار كلية محدد في النظام

عرفنا متغير للاختيار

وعملنا شرط ينص على ان المستخدم

إذا ادخل رقم اقل من ١ او رقم اكبر

من حجم قاعدة البيانات الخاصة بالتخصصات

يطبع "الاختيار غير متوفر"

إذا لم يتحقق الشرط يقوم النظام بإضافة التخصص

المختارة الى سجل الطالب

٤. اختيار الالتحاق بالنظام الموازي

```
char parallelChoice;  
cout << "Would you like to join the Parallel Program?"  
(y/n): ";  
cin >> parallelChoice;  
  
newStudent.parallelProgram = (parallelChoice == 'y' ||  
parallelChoice == 'Y');  
registeredStudents.push_back(newStudent);
```

يختار المستخدم إحدى الخيارين

(y,n)

ومن ثم يقوم النظام بإضافة بيانات الطالب
الجديد الى قاعدة البيانات الخاصة بالطالب
باستخدام دالة push_back

```
cout << "\nRegistration successful!\n";  
cout << "Your information:\n";  
cout << "Name: " << newStudent.name << "\n";  
cout << "University ID: " << newStudent.id << "\n";  
cout << "College: " << newStudent.college << "\n";  
cout << "Specialization: " <<  
newStudent.specialization << "\n";  
cout << "Parallel Program: " <<  
(newStudent.parallelProgram ? "Yes" : "No") << "\n";  
  
cout << "\nPress any key to return to main menu...";  
cin.ignore();  
cin.get();
```

عرض بيانات الطالب المسجل

- اسم الطالب
- الرقم الجامعي
- الكلية المختارة
- التخصص المختار

٩. دالة عرض معلومات النظام الموازي

```
void showParallelProgramInfo() {  
    cout << "Parallel Program\n";  
  
    cout << "What is the Parallel Program?\n";  
    cout << "The Parallel Program is an academic program that provides  
university education opportunities\n";  
    cout << "for students who didn't get admission in the regular program,  
with higher tuition fees.\n\n";  
  
    cout << "Program Features:\n";  
    cout << "- Same curriculum as the regular program\n";  
    cout << "- Same faculty members\n";  
    cout << "- Same awarded degrees\n\n";  
  
    cout << "Tuition Fees:\n";  
    cout << "- Fees vary by college and specialization\n";  
    cout << "- Average fees between 20,000 to 40,000 SAR annually\n\n";  
  
    cout << "Admission Requirements:\n";  
    cout << "- High school diploma\n";  
    cout << "- Meeting college requirements (typically 5-10% lower than  
regular program)\n\n";  
  
    cout << "Note: Transfer from parallel to regular program is possible  
according to university policies.\n";  
  
    cout << "\nPress any key to return to main menu...";
```

في هذه الدالة نقوم

بشرح ميزات النظام الموازي

والتكاليف

وشروط الالتحاق

١٠. دالة عرض الطلاب المسجلين في الجامعة

```
void showRegisteredStudents() {
```

```
    cout << "Registered Students\n\n";
```

في هذه الدالة نقوم بعرض الطلاب الذي تم تسجيلهم

```
    if (registeredStudents.empty()) {
        cout << "No students have registered yet.\n";
    } else {
```

في النظام وقبل ما يتم العرض نتحقق هل يوجد

```
        cout << "Total Registered Students: " <<
        registeredStudents.size() << "\n\n";
```

طلاب مسجلين ام لا

```
        for (const Student& student : registeredStudents) {
            cout << "Student Name: " << student.name << "\n";
            cout << "University ID: " << student.id << "\n";
            cout << "College: " << student.college << "\n";
            cout << "Specialization: " << student.specialization <<
```

في حالة وجود طلاب مسجلين

```
            "\n";
```

```
            cout << "Parallel Program: " << (student.parallelProgram
? "Yes" : "No") << "\n";
```

نقوم بعرض بياناتهم المخزنة في قاعدة

البيانات الخاصة بالطلاب المسجلين

مع تفاصيل كل طالب

```
        }
    }

    cout << "\nPress any key to return to main menu...";
    cin.ignore();
    cin.get();
}
```

}

```
int main() {
    initializeData();
```

١١. الدالة الرئيسية

```
    int choice;
    do {
        showMainMenu();
        cin >> choice;
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
```

```
    switch (choice) {
        case 1:
            showSpecializations();
            break;
        case 2:
            showAdmissionRequirements();
            break;
        case 3:
            registrationProcess();
            break;
        case 4:
            showParallelProgramInfo();
            break;
        case 5:
            showRegisteredStudents();
            break;
        case 6:
```

في هذه الدالة عملنا تهيئة للبيانات الموجودة في النظام

وقمنا بعمل حلقة تكرار لعرض القائمة الرئيسية وتنفيذ

الخيارات حتى يتم اختيار الخروج

وداخل حلقة التكرار استخدمنا switch للتحكم

في نوع الاختيار التي تم اختياره من المستخدم

```
        cout << "Thank you for using the University
Admission System. Goodbye!\n";
        break;
        default:
            cout << "Invalid choice, please try
again.\n";
            break;
    }
    while (choice != 6);

    return 0;
```