|  |
| --- |
|  |

|  |
| --- |
| **المشروع النهائي** |
| **نظام جامعي** |
|  |
|  |
|  |
|  |

توثيق مشروع النظام جامعي

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

1. ***المكاتب المستخدمة في البرنامج***

**vector**

**#include <iostream>**

**#include <string>**

**#include <vector>**

**#include <map>**

**//#include <limits>**

using namespace std;

**استخدمنا مكتبة الإدخال والإخراج مكتبة السلاسل النصية ومكتبة**

**map**

**تقوم بتخزين البيانات بداخلها على شكل مصفوفات و مكتبة**

**تخزن البيانات فيها بشكل مرتب تصاعدي**

1. ***هياكل البيانات***
2. ***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**

1. ***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**

1. ***تركيب خاص في بيانات الطلاب ويحتوي على***

**truct Student {**

**string name;**

**string id;**

**string college;**

**string specialization;**

**bool parallelProgram;**

**};**

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

***استخدمنا قاعدتين بيانات لتخزين البيانات***

**vector<College> colleges;**

**vector<Student> registeredStudents;**

1. ***قواعد بيانات النظام***
2. **قاعدة بيانات خاصة بالكليات**
3. **قاعدة بيانات خاصة بتخزين بيانات الطالب**
4. ***دالة لتهيئة البيانات***

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

**void initializeData() {**

**// Computer Science College**

**College cs;**

**cs.name = "College of Computer Science and Information";**

**cs.admissionRequirements = {**

**{"High School", "Minimum 80% "},**

**{"Aptitude Test", "Minimum 75%"},**

**};**

1. **كلية الحاسبات**

**استخدمنا المتغير (cs) لتسمية الكلية عن طريف الاتي:**

* **اسم الكلية " كلية الحاسبات"**
* **متطلبات القبول }}" الثانوية العامة, المعدل على الأقل 80%"{**

**}"اختبار القبول, المعدل على الأقل 75%"{{**

* **التخصص الأول لكلية الحاسبات**

**Specialization csSpec1;**

**csSpec1.name = "Computer Science";**

**csSpec1.description = "Program covering programming fundamentals, data structures, and AI\n\n";**

**csSpec1.duration = 4;**

**csSpec1.courses = {"Programming 1", "Data Structures", "Algorithms", "Database Systems\n\n"};**

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

* اسم التخصص " **علوم الحاسب**"
* وصف التخصص" **هو برنامج يغطي اساسيات البرمجة و**

**هياكل البيانات والذكاء الصناعي**"

* مدة الدراسة= 4 سنوات
* المواد الأساسية" **البرمجة و هياكل البيانات و قاعدة البيانات...."**
* **التخصص الثاني لكلية الحاسب**

رمزنا له بالرمز(**csspec2**) ويحتوي علي

**Specialization csSpec2;**

**csSpec2.name = "Software Engineering";**

**csSpec2.description = "Program focused on software development and management\n\n";**

**csSpec2.duration = 4;**

**csSpec2.courses = {"Software Engineering", "Web Development", "Software Testing", "Project Management\n\n"};**

* اسم التخصص " **هندسة البرمجيات**"
* وصف التخصص" **هو برنامج يركز على تطوير البرمجيات**

**وإدارتها**"

* مدة الدراسة= 4 سنوات
* المواد الأساسية" **هندسة البرمجيات و تطوير الويب و**

**إدارة المشاريع**......."

**وبعد التهيئة قمنا بإضافة البيانات الى قائمة التخصصات**

**باستخدام دالة push\_back**

**colleges.push\_back(cs);**

**cs.specializations.push\_back(csSpec1);**

**cs.specializations.push\_back(csSpec2);**

**وبعد ذلك نقوم بإضافة جميع البيانات الى قاعدة**

**البيانات الخاصة بالكليات باستخدام نفس الدالة**

**Push\_back**

1. **كلية الهندسة**

**College eng;**

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

**eng.admissionRequirements = {**

**{"High School", "Minimum 85% in scientific track"},**

**{"Aptitude Test", "Minimum 80%"},**

**استخدمنا المتغير (eng) لتسمية الكلية عن طريف الاتي:**

* **اسم الكلية " كلية هندسة الحاسوب"**
* **متطلبات القبول }}" الثانوية العامة, المعدل على الأقل 85%"{**

**}"اختبار القبول, المعدل على الأقل 80%"{{**

* **التخصص الأول لكلية الهندسة**

رمزنا له بالرمز(**engspec1**) ويحتوي علي

**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"};**

* اسم التخصص "هندسة الحاسوب"
* وصف التخصص" **هو برنامج يجمع بين هندسة الإلكترونيات**

**و علوم الحاسوب**"

* مدة الدراسة= 5 سنوات
* المواد الأساسية" **الدوائر الإلكترونية و الإلكترونيات**

**الرقمية ومعالجة الإشارات و شبكات الحاسوب...."**

* **التخصص الثاني لكلية الهندسة**

رمزنا له بالرمز(**engspec2**) ويحتوي علي

**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"};**

* اسم التخصص " **الهندسة الصناعية**"
* وصف التخصص" **هو برنامج يركز على تحسين**

**النظم والعمليات**"

* مدة الدراسة= 5 سنوات
* المواد الأساسية" **بحوث العمليات و نظم التصنيع**

**وهندسة العوامل البشرية و إدارة الجودة**......."

**وبعد التهيئة قمنا بإضافة البيانات الى قائمة التخصصات**

**eng.specializations.push\_back(engSpec1);**

**eng.specializations.push\_back(engSpec2);**

**باستخدام دالة push\_back**

**colleges.push\_back(eng);**

**وبعد ذلك نقوم بإضافة جميع البيانات الى قاعدة**

**البيانات الخاصة بالكليات باستخدام نفس الدالة**

**Push\_back**

**واستخدمنا دالةcin.ignore()**

**لتنظيف الادخال من الاحرف الزائدة ومنع حدوث أخطاء في التبديل بين cin و getline() وضمان قراءة الادخال بشكل صحيح من المستخدم دون سلوك غير متوقع**

1. ***دالة عرض القائمة الرئيسية***

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

1. **عرض التخصصات**

**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: ";**

1. **متطلبات القبول**
2. **عملية التسجيل**
3. **النظام الموازي**
4. **عرض بيانات الطلاب**
5. **الخروج**
6. ***دالة عرض التخصصات المتاحة***

**هذه الدالة تعرض جميع الكليات والتخصصات مع تفاصيل كل تخصص**

**عملنا دوارة قمنا بتعريف داخلها**

**متغير (college) لتخزين اسم**

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();

}

**الكلية المأخوذة من قاعدة البيانات(colleges)**

**وعرفنا ايضا دوارة بداخل الدوارة الاولى**

**وقمنا بتعريف متغير (spec)داخلها**

**لتخزين اسم و وصف التخصص ومدة دراسته**

**من قائمة التخصصات**

**وأيضا عرفنا دوارة خاصه بالمواد بداخلها متغير**

**(course) لتخزين المواد من قائمة التخصص**

1. ***دالة عرض متطلبات القبول***

**هذه الدالة تظهر كل متطلبات القبول لكل كلية**

**void showAdmissionRequirements() {**

**cout << "College Admission Requirements:\n\n";**

**for (const College& college : colleges) {**

**cout << "\nCollege: " << college.name << "\n";**

**cout << " \*Requirements\*:\n";**

**for (const auto& req : college.admissionRequirements) {**

**cout << "- " << req.first << ": " << req.second << "\n";**

**}**

**}**

**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();**

**}**

**عملنا دوارة قمنا بتعريف داخلها**

**متغير (college) لتخزين اسم**

**الكلية**

**المأخوذة من قاعدة البيانات(colleges)**

**عرفنا دوارة لعرض المتطلبات وعرفنا**

**بداخلها متغير من نوع auto**

**وخزنا بداخلة المتطلبات**

**وأيضا طبعنا ملاحظة تنص على ان الشروط**

**تتغير من كلية الى أخرى**

**معلومات عن map , auto**

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

**الوصول الى العناصر في :map**

**هناك طريقتين للوصول 1. الوصول باستخدام معامل [] ,2. الوصول باستخدام المعامل .at ()**

**ال map في c++ يحفظ جميع العناصر مرتبة حسب المفتاح (بترتيب تصاعدي افتراضيا) مما يجعل البحث والوصول الى العناصر سريعا**

**مميزات loop في map: 1. الإيجاز/ تقليل عدد الكود المطلوب 2. الأمان/ تجنب الأخطاء الشائعة في استخدام الكود.**

**استخدام auto في ال map:**

**تستخدم لاستنتاج نوع عناصر الmap 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 اكثر قراءة وأيضا يتجنب الأخطاء في كتابة النوع يدوياً**

**في هذه الدالة نقوم بتسجيل الطالب في الجامعه 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);**

1. ***دالة لعملية التسجيل***
2. **توجيه الطالب لإدخال البيانات الشخصية**

**في هذا الكود يتحقق في البداية من**

**وجود كليات متاحه إذا لم يكن هناك كليات**

**متاحه يطبع "لا يوجد كليات متاحه"**

**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;**

1. **اختيار الطالب للكلية**

**استخدمنا دوارة لاختيار كلية محدد في النظام**

**عرفنا متغير للاختيار**

**وعملنا شرط ينص على ان المستخدم**

**إذا ادخل رقم اقل من 1 او رقم اكبر**

**من حجم قاعدة البيانات الخاصة بالكليات**

**يطبع " الاختيار غير متوفر"**

**اذا 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;

1. **اختيار الطالب للتخصص**

**استخدمنا دوارة لاختيار كلية محدد في النظام**

**عرفنا متغير للاختيار**

**وعملنا شرط ينص على ان المستخدم**

**إذا ادخل رقم اقل من 1 او رقم اكبر**

**من حجم قاعدة البيانات الخاصة بالتخصصات**

**يطبع " الاختيار غير متوفر"**

**اذا لم يتحقق الشرط يقوم النظام بإضافة التخصص**

**المختارة الى سجل الطالب**

1. **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**

1. ***دالة عرض معلومات النظام المواي***

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...";

cin.ignore();

cin.get();

**في هذه الدالة نقوم**

**بشرح ميزات النظام الموازي**

**والتكاليف**

**وشروط الالتحاق**

void showRegisteredStudents() {

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

if (registeredStudents.empty()) {

cout << "No students have registered yet.\n";

} else {

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

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 << "\n";

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

}

}

cout << "\nPress any key to return to main menu...";

cin.ignore();

cin.get();

}

1. ***دالة عرض الطلاب المسجلين في الجامعة***

**في هذه الدالة نقوم بعرض الطلاب الذي تم تسجيلهم**

**في النظام وقبل ما يتم العرض نتحقق هل يوجد**

**طلاب مسجلين ام لا**

**في حالة وجود طلاب مسجلين**

**نقوم بعرض بياناتهم المخزنة في قاعدة**

**البيانات الخاصة بالطلاب المسجلين**

**مع تفاصيل كل طالب**

int main() {

initializeData();

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:

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;

}

1. ***الدالة الرئيسية***

**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**