





# GESTION D'ÉTABLISSEMENT

ALI HARIT  
ZAKARIA EL HAJJAM

ENCADRÉ PAR :  
ENNAJI FATIMA ZOHRA

# SOMMAIRE

01

Introduction

02

Bibliothèques

03

Connexion avec la base  
de donnée

04

Compilation

05

Espaces

06

Interfaces

07

Fonctions

08

Conclusion

# Introduction

Le domaine de l'éducation nécessite une gestion efficace des données académiques afin d'optimiser les processus d'apprentissage et de fournir aux enseignants et aux étudiants un environnement propice à l'épanouissement intellectuel. Dans cette optique, notre projet de gestion d'établissement en langage C++ vise à créer un système modulaire et convivial permettant aux professeurs et aux étudiants de gérer de manière transparente les informations liées aux cours, aux enseignants, aux étudiants et aux notes.

Le langage de programmation C++ a été choisi pour ses capacités robustes, sa performance et sa flexibilité, offrant ainsi un cadre idéal pour le développement d'une application complète de gestion éducative. Ce projet s'articule autour de principes de conception modulaire, d'interface utilisateur intuitive et d'une gestion sécurisée des données, garantissant ainsi la simplicité d'utilisation et la fiabilité du système.

# Bibliothèques

## iostream

La bibliothèque `<iostream>` fournit des fonctionnalités d'entrée/sortie standard en C++. Elle inclut les objets `cin` et `cout` pour la lecture depuis la console et l'affichage à l'écran, respectivement. Elle permet également d'utiliser des flux pour lire et écrire dans des fichiers.

## cstdlib

La bibliothèque `<cstdlib>` (C Standard Library) fournit des fonctions liées au traitement de chaînes de caractères, à la gestion de la mémoire dynamique, aux opérations sur des entiers, et d'autres fonctions générales. Par exemple, les fonctions `malloc` et `free` pour la gestion de la mémoire dynamique sont incluses dans cette bibliothèque.

## cstring

La bibliothèque `<cstring>` fournit des fonctions de manipulation de chaînes de caractères en langage C. Elle comprend des opérations telles que la copie, la concaténation et la comparaison de chaînes de caractères. Les fonctions populaires incluent `strcpy`, `strcat` et `strcmp`.

## Couleurs de texte

Ces définitions utilisent des codes de couleurs ANSI pour modifier la couleur du texte affiché dans la console. Ces codes sont utilisés avec des séquences d'échappement spécifiques pour changer la couleur du texte à l'écran. Par exemple, `red` permet d'afficher du texte en rouge.

## vector

La bibliothèque `<vector>` fait partie de la Standard Template Library (STL) en C++. Elle offre la classe `vector` qui permet de créer des tableaux dynamiques, avec une gestion automatisée de la mémoire et des fonctionnalités de manipulation étendues, comme l'ajout ou la suppression d'éléments.

# Connexion avec la base de donnée

## <sqlite3.h>

La bibliothèque <sqlite3.h> est la bibliothèque client de SQLite, un moteur de base de données relationnelles embarqué écrit en langage C++. SQLite est réputé pour sa légèreté, sa simplicité d'utilisation et sa polyvalence, ce qui en fait un choix populaire pour intégrer des bases de données dans des applications logicielles, y compris dans des applications mobiles et embarquées.

Cette connexion a été établie pour la création d'une base de données dédiée à la gestion complète, la manipulation, et la mise à jour des informations relatives aux étudiants et aux professeurs. Elle offre ainsi un cadre robuste permettant de stocker de manière structurée et sécurisée les données essentielles à la gestion académique. Cette base de données constitue un outil fondamental, fournissant une infrastructure solide pour enregistrer les détails des étudiants, les informations liées aux professeurs, et facilitant l'accès aux fonctionnalités de manipulation et de mise à jour des données en toute efficacité. En exploitant cette connexion, notre projet vise à optimiser la gestion des informations académiques au sein de l'établissement, offrant ainsi un environnement propice à la prise de décision informée et à la qualité des processus éducatifs.

# compilation

Pour compiler le code source `main.cpp` avec la bibliothèque SQLite3 et obtenir un exécutable nommé `projetcpp`, la commande suivante est utilisée :

```
c++ main.cpp -o projetcpp -lsqlite3
```

**c++:** Cela appelle le compilateur C++.

**main.cpp:** Le fichier source à compiler.

**-o projetcpp:** Spécifie le nom de l'exécutable généré.

**-lsqlite3:** Indique au compilateur de lier la bibliothèque SQLite3 pendant la phase de compilation.

Une fois la compilation terminée avec succès, un exécutable nommé `projetcpp` est généré. Vous pouvez exécuter cet exécutable pour vérifier le bon fonctionnement du programme résultant.

```
./projetcpp
```

# Espaces

Au sein de notre application, nous distinguons trois espaces principales, à savoir :

## Espace étudiant :

L'étudiant occupe une place centrale dans notre projet, avec un espace dédié comprenant une connexion sécurisée utilisant une adresse e-mail et un mot de passe uniques. Une fois connecté, l'étudiant a automatiquement accès à ses notes réparties dans différents modules. Ce processus implique un traitement détaillé où la filière et les modules spécifiques de l'étudiant sont sélectionnés minutieusement à partir de la base de données.

## Espace professeur :

Les enseignants, qui représentent les professeurs des étudiants, disposent d'un espace dédié comprenant leurs informations personnelles telles que le nom, prénom, adresse e-mail et mot de passe. Après une connexion sécurisée à l'aide de leur adresse e-mail et de leur mot de passe, les enseignants ont la possibilité de sélectionner le module qu'ils enseignent et d'accéder aux informations des étudiants associés à ce module.

## Espace directeur :

Le directeur, en tant que responsable de l'application, a la capacité de choisir une filière et un module spécifiques. Une fois cette sélection effectuée, il peut consulter les notes de tous les étudiants associés à cette filière et ce module particuliers. De plus, le directeur a le pouvoir de modifier les notes au besoin, assurant ainsi une gestion complète et efficace du suivi académique au sein de l'établissement.



# Interfaces

## Espace Etudiant :

ENSA  
KHOURIBGA

SERVICE SCOLARITÉ

Étudiant (E)      Professeur (P)      Directeur Scolarité (D)

>

ENSA  
KHOURIBGA

SERVICE SCOLARITÉ

E-mail :ali.harit@ensa.ma

Password :aliharit

ENSA  
KHOURIBGA

Bienvenue zakaria el hajjam

GI1

c++ Notes: 18  
python Notes: 18  
probabilite Notes: 16  
algorithmes Notes: 14  
java  
php Notes: 14

## Espace Professeurs :

ENSA  
KHOURIBGA

Bienvenue yassine motawakil

zakaria el hajjam | 18  
ali harit | 19  
morad imani | 18  
ayoub jaghni | 11  
anwar rachdi |  
rachad wadie | 17  
ahmed nour |  
yassine said | 15  
anas ichmawin | 15

ENSA  
KHOURIBGA

Bienvenue yassine motawakil

(1) GI1    c++  
  
(2) GI1    python  
  
(3) GI1    java

# Interfaces

## Espace directeur :

```

                                ENSA
                                KHOURIBGA
Bienvenue mohamed quadiri

(1) API1
(2) API2
(3) GI1
(4) GI2
(5) GI3
(6) IID1
(7) IID2
(8) IID3

```

```

                                ENSA
                                KHOURIBGA
Bienvenue mohamed quadiri

300 | zakaria el hajjam | 18
301 | ali harit | 19
302 | morad imani | 18
303 | ayoub jaghni | 11
304 | anwar rachdi |
305 | rachad wadie | 17
306 | ahmed nour |
307 | yassine said | 15
308 | anas ichmawin | 15
309 | ihab hillal | 16

donner id de l'etudiant :306
veuillez saisir la note :11

```

Les captures d'écran fournies précédemment englobent toutes les interfaces susceptibles de s'afficher lors de l'exécution du programme. Ces images offrent une représentation exhaustive de tous les espaces accessibles au cours de l'utilisation du logiciel, illustrant ainsi l'ensemble des fonctionnalités et des modules disponibles dans le cadre de l'application.

# Fonctions

## function.cpp

### recherche\_mail\_password

Cette fonction recherche un utilisateur dans la base de données en fonction de son adresse e-mail et de son mot de passe, en spécifiant le type d'utilisateur (professeur, directeur ou étudiant).

Elle renvoie l'identifiant de l'utilisateur trouvé.

### updateNotes

Cette fonction met à jour les notes d'un étudiant en fonction de la filière et du module spécifiés.

### affiche\_note\_etud

Cette fonction affiche les notes d'un étudiant en fonction de la filière d'un professeur et de l'identifiant de l'étudiant.

### search\_name\_etud

Cette fonction recherche le nom et le prénom d'un étudiant en fonction de la filière d'un professeur et de l'occurrence spécifiée.

# Fonctions

## main.cpp

## home\_page

Cette fonction recherche le nom et le prénom d'un étudiant en fonction de la filière d'un professeur et de l'occurrence spécifiée.

## login\_page

Gère le processus de connexion en demandant à l'utilisateur son adresse e-mail et son mot de passe. Elle valide les informations d'identification et, en cas de succès, redirige vers la page d'accueil respective de l'utilisateur (professeur, directeur ou étudiant).

## accueil\_professeur

Affiche la page d'accueil du professeur, lui permettant de choisir un cours pour voir les notes des étudiants. Ensuite, elle invite le professeur à sélectionner un cours et, après la sélection, affiche les notes de tous les étudiants dans ce cours.

## accueil\_directeur

Affiche la page d'accueil du directeur, lui permettant de choisir un département et un cours pour voir les notes des étudiants. Après avoir choisi un département et un cours, le directeur peut voir et mettre à jour les notes des étudiants.

# Fonctions

**main.cpp**

**accueil\_etudiant**

Affiche la page d'accueil de l'étudiant, montrant les informations personnelles de l'étudiant et les notes pour ses cours respectifs.

**les fonctions prédéfinies**

**sqlite3\_open**

retourne un code d'erreur. Si l'ouverture de la base de données est réussie, elle renverra SQLITE\_OK (égal à 0). Si elle échoue, elle renverra un autre code d'erreur qui indiquera la nature de l'échec.

**sqlite3\_prepare\_v2**

Cette fonction est utilisée pour compiler une requête SQL dans un objet de type sqlite3\_stmt afin de pouvoir l'exécuter par la suite.

**sqlite3\_bind\_text**

est utilisée pour lier des valeurs à des paramètres dans une instruction SQL préparée

# Fonctions

## sqlite3\_step

Cette fonction est utilisée pour exécuter l'instruction SQL préparée pointée par stmt. Elle est appelée de manière itérative jusqu'à ce que tous les résultats de la requête aient été récupérés. À chaque appel, la fonction effectue une seule étape du processus d'exécution

## sqlite\_row

Indique qu'une nouvelle ligne de résultat est disponible pour récupération à l'aide de fonctions telles que sqlite3\_column\_\*.

## sqlite3\_finalize

Cela libère les ressources associées à l'instruction préparée.

## sqlite3\_stmt

est utilisée dans l'API SQLite pour représenter une instruction SQL préparée. Une instruction SQL préparée est une représentation compilée d'une requête SQL.

## sqlite3\_column\_int

est utilisée pour récupérer la valeur d'une colonne sous forme d'entier à partir du résultat d'une requête SQLite. Cette fonction prend en paramètres un pointeur vers une structure sqlite3\_stmt (qui représente l'instruction préparée), ainsi que l'indice de la colonne dont vous souhaitez récupérer la valeur.

## Conclusion

En conclusion, ce projet de gestion d'établissement en C++ avec une base de données SQLite a été une expérience gratifiante. Nous avons réussi à mettre en place un système fonctionnel pour la gestion des étudiants et des professeurs, en exploitant efficacement les capacités de la base de données. Bien que le projet soit opérationnel, des améliorations potentielles subsistent, offrant des opportunités pour un développement futur. Cette expérience a renforcé nos compétences en programmation et a ouvert la porte à de nouvelles perspectives dans le domaine du développement logiciel.