



Université Abdelmalek Essaadi
Faculté des Sciences et Techniques de Tanger
Cycle Ingénieur Logiciels et Systèmes Intelligents
Année Universitaire 2023-2024



Mini Projet 1: “Générateur d’interface”

Réalisé par:
DAAOUAN Mohammed
FRIKH Said

Encadré par:
Pr. Hassan ZILI

Table de Matieres

I. Introduction.....	3
Objectif du projet.....	3
Présentation de l'application.....	3
Technologies utilisées.....	4
II. Application Desktop.....	5
Database connexion page.....	5
Affichage des tables.....	6
Ajouter un enregistrement.....	7
Modifier un enregistrement.....	8
Suppression d'un enregistrement.....	8
III. Conclusion.....	9
IV. Annexes.....	9
Code source.....	9

I. Introduction

Objectif du projet

L'objectif de ce mini-projet est de développer une application desktop capable de générer dynamiquement une interface utilisateur à partir de la structure d'une base de données existante. Cette application doit permettre aux utilisateurs de visualiser les tables et leur contenu, ainsi que d'effectuer des opérations de gestion des données telles que l'ajout, la mise à jour et la suppression d'enregistrements. L'application vise à simplifier l'interaction avec la base de données en fournissant une interface graphique intuitive et facile à utiliser, réduisant ainsi le besoin de compétences techniques avancées pour manipuler les données.

Présentation de l'application

L'application, nommée "Générateur d'Interface DB", est une solution logicielle desktop conçue pour interagir avec des bases de données relationnelles. Voici une vue d'ensemble de ses principales fonctionnalités :

1. Connexion à la Base de Données :

- L'application permet de se connecter à une base de données en fournissant les informations nécessaires telles que le type de base de données (par exemple, MySQL, PostgreSQL), l'adresse du serveur, le nom de la base de données, et les identifiants de connexion.

2. Affichage des Tables :

- Une fois connecté, l'application récupère et affiche la liste des tables présentes dans la base de données. Cette vue permet de sélectionner une table pour voir son contenu ou pour effectuer des opérations de gestion.

3. Visualisation du Contenu des Tables :

- Pour chaque table sélectionnée, l'application affiche son contenu sous forme de tableau, permettant de parcourir les enregistrements existants de manière visuelle et structurée.

4. Opérations CRUD :

- **Ajouter un Enregistrement** : Un formulaire dédié permet d'ajouter de nouveaux enregistrements dans une table, avec des champs dynamiquement générés en fonction de la structure de la table.
- **Éditer un Enregistrement** : L'application permet de sélectionner un enregistrement existant et de le mettre à jour via un formulaire similaire à celui de l'ajout.
- **Supprimer un Enregistrement** : Un bouton de suppression est disponible pour chaque enregistrement, permettant de le supprimer de la base de données.

5. Interface Utilisateur Intuitive :

- L'application est conçue avec une interface utilisateur claire et intuitive, facilitant la navigation et l'utilisation pour les utilisateurs non techniques. Les différentes fonctionnalités sont accessibles via des menus et des boutons bien disposés.

Technologies utilisées

Pour le développement de l'application "Générateur d'Interface DB", les technologies suivantes seront utilisées :

1. Langage de Programmation :

- **C#** : Un langage moderne et puissant, idéal pour le développement d'applications desktop robustes et performantes.



2. Environnement de Développement :

- **Visual Studio** : Un environnement de développement intégré (IDE) complet et riche en fonctionnalités, offrant des outils de développement avancés, un débogueur intégré, et une excellente prise en charge de C# et .NET.



3. Framework :

- **.NET** : Une plateforme de développement flexible et multi-plateforme, permettant de créer des applications desktop performantes avec une gestion efficace des ressources et des fonctionnalités avancées.



4. Bases de Données Supportées :

- L'application sera conçue pour être compatible avec plusieurs systèmes de gestion de bases de données relationnelles courants comme **MySQL**, **PostgreSQL**,,, offrant une flexibilité maximale.



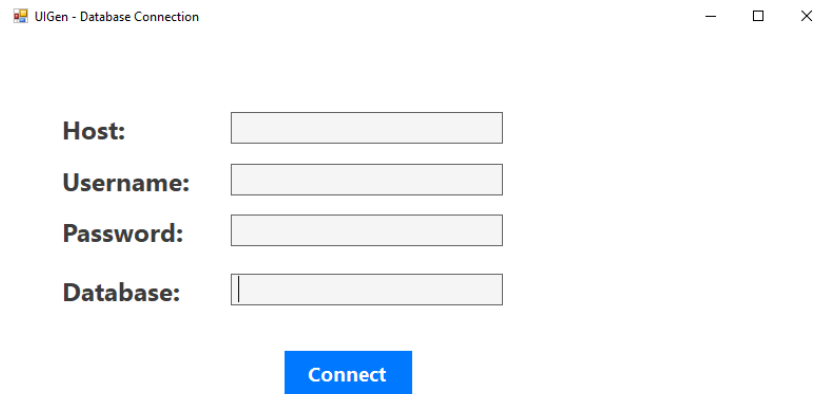
5. Contrôle de Version :

- **Git** : Un système de gestion de versions distribué, permettant de suivre les modifications du code source et de collaborer efficacement avec d'autres développeurs.
- **GitHub** : Une plateforme de partage et de collaboration sur le code source, facilitant la gestion des versions et le suivi des issues.



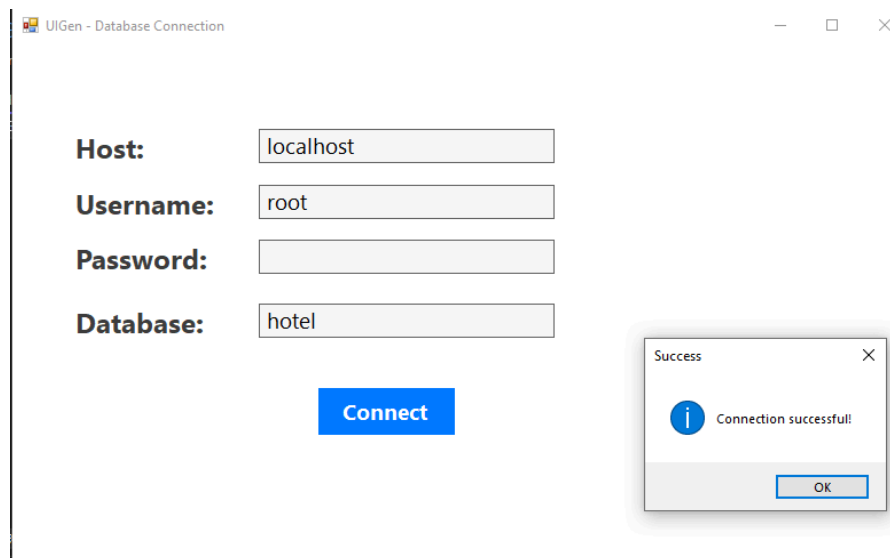
II. Application Desktop

Database connexion page



A screenshot of a desktop application window titled "UIGen - Database Connection". The window contains four input fields with labels: "Host:", "Username:", "Password:", and "Database:". Below these fields is a blue button labeled "Connect". The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

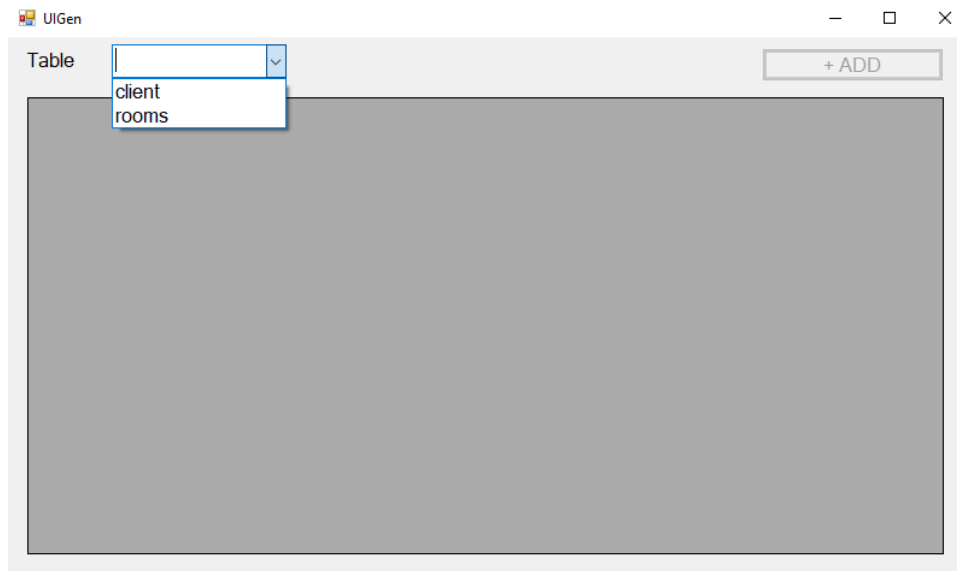
L'utilisateur peut entrer les détails de connexion à la base de données, tels que le type de base de données (par exemple, MySQL, PostgreSQL), l'adresse du serveur, le nom de la base de données et les informations d'identification de connexion. Cette étape est cruciale pour établir la communication entre l'application et la base de données.



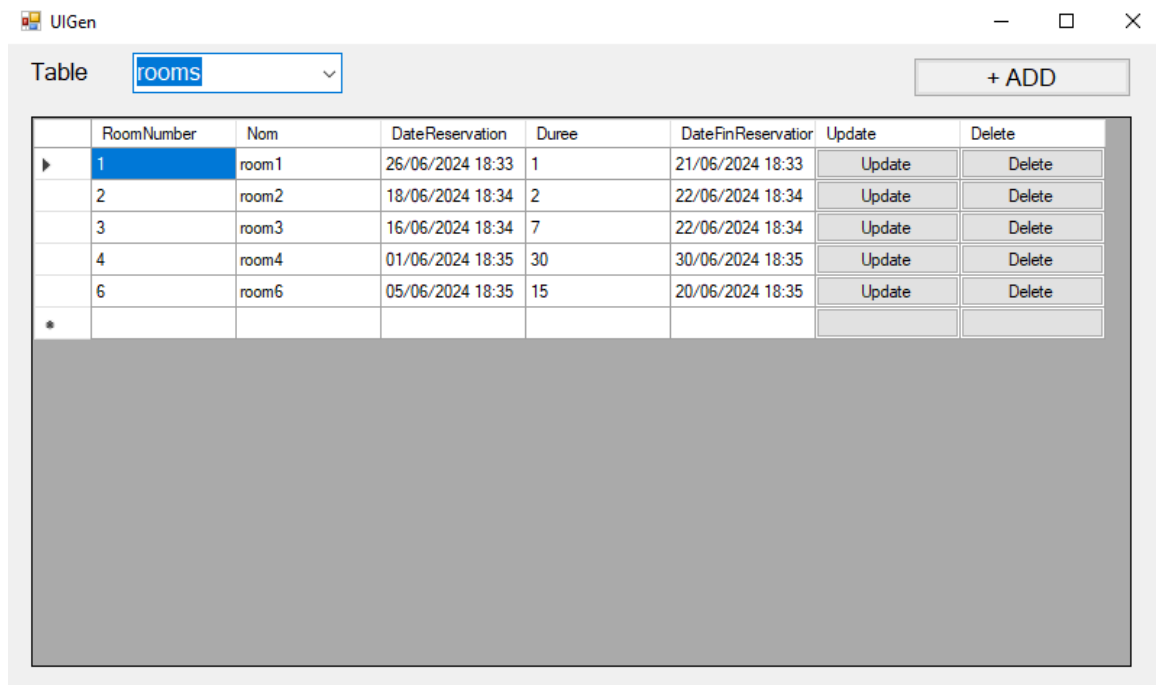
A screenshot of the "UIGen - Database Connection" window with the following values entered: Host: localhost, Username: root, Password: (empty), and Database: hotel. The "Connect" button is highlighted. A small "Success" dialog box is open in the bottom right corner, displaying a blue information icon and the text "Connection successful!". The dialog box has an "OK" button.

Affichage des tables

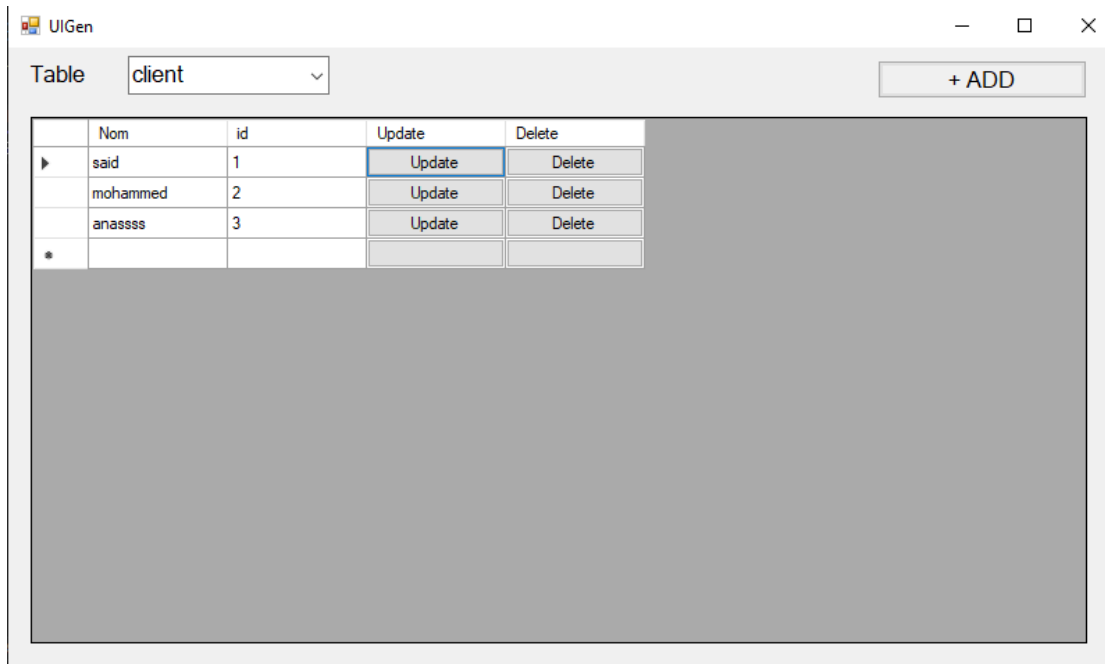
Les utilisateurs peuvent sélectionner une table pour en afficher le contenu ou effectuer des opérations de gestion. L'interface est conçue pour être intuitive, permettant une navigation facile à travers les différentes tables.



On a choisi la table rooms comme exemple



l'application affiche le contenu d'une table sélectionnée dans un format tabulaire structuré. Cela permet aux utilisateurs de parcourir les enregistrements existants de manière visuelle et organisée, facilitant ainsi la gestion des données.



Ajouter un enregistrement

Table client

Go Back

id

nom

frikh said

Data inserted successfully.

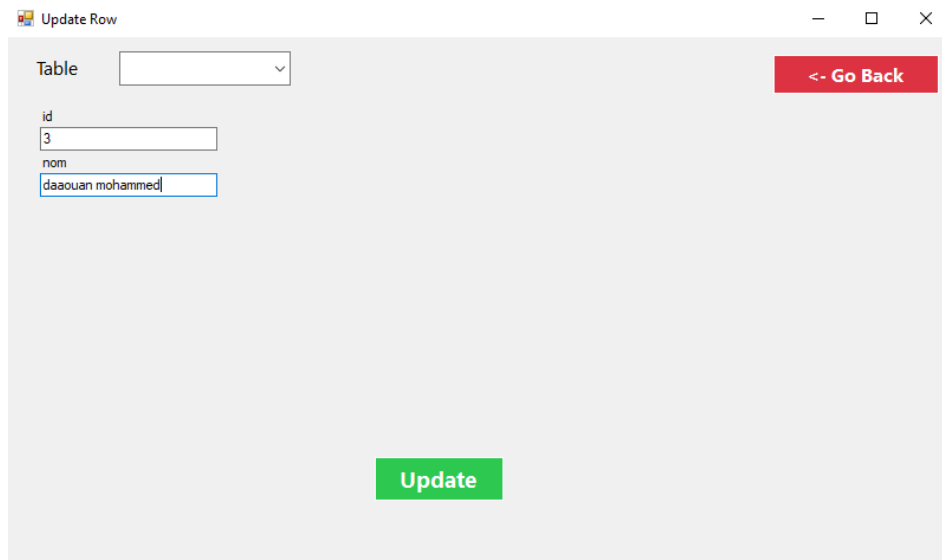
OK

Save

Ce formulaire est utilisé pour ajouter de nouveaux enregistrements dans une table. Les champs du formulaire sont générés dynamiquement en fonction de la structure de la table, ce qui permet aux utilisateurs de saisir facilement les informations nécessaires pour chaque champ.

Modifier un enregistrement

L'interface est similaire au formulaire d'ajout d'enregistrement, permettant une modification facile des enregistrements avec des champs dynamiquement générés en fonction de la structure de la table.



Update Row

Table

id

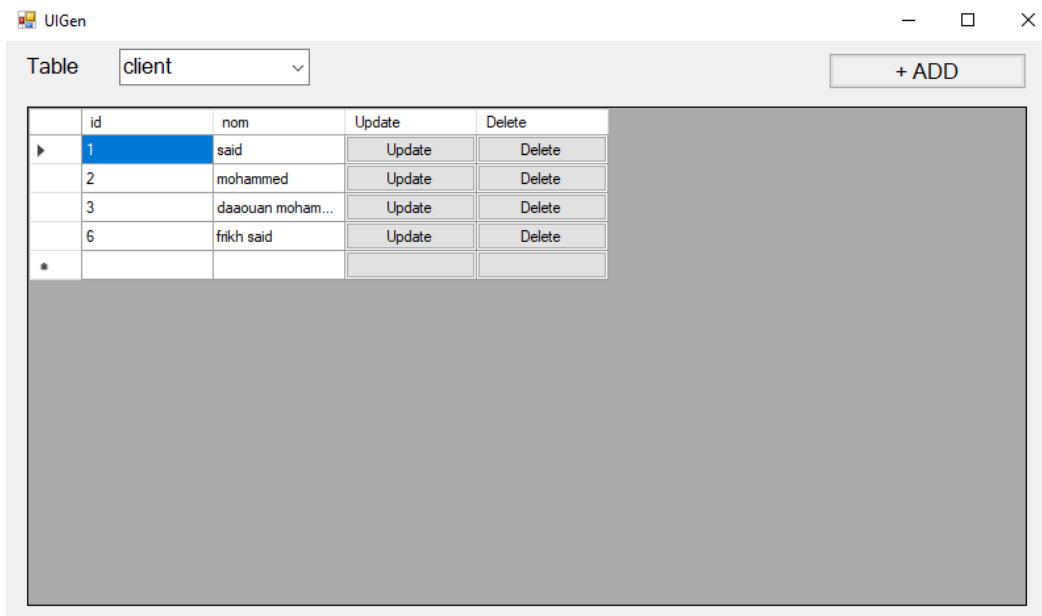
nom

<- Go Back

Update

Suppression d'un enregistrement

Cette image met en évidence le bouton de suppression disponible pour chaque enregistrement dans la vue de la table. Les utilisateurs peuvent cliquer sur ce bouton pour supprimer un enregistrement de la base de données. Cette fonctionnalité offre une gestion complète des données en permettant la suppression des enregistrements non désirés.



UIGen

Table

+ ADD

	id	nom	Update	Delete
▶	1	said	Update	Delete
	2	mohammed	Update	Delete
	3	daaouan moham...	Update	Delete
	6	frikh said	Update	Delete
*				

III. Conclusion

Le projet "Générateur d'Interface DB" a réussi à simplifier l'interaction avec les bases de données relationnelles grâce à une application desktop intuitive, utilisant des technologies modernes comme .NET et WPF. Il permet la gestion complète des données via des interfaces dynamiques et conviviales, tout en offrant une grande adaptabilité. Les perspectives futures incluent l'ajout de fonctionnalités avancées et l'amélioration continue de l'interface utilisateur et des performances. Ce projet constitue une solution robuste et évolutive pour la gestion des données.

IV. Annexes

Code source

https://github.com/Frikh-Said/Generateur_d_interface.git