Simple Login and Registration System

Introduction:

This document provides all the details required for a Simple Login and Registration System that uses C# and Windows Forms for the desktop app and SQL for the backend.

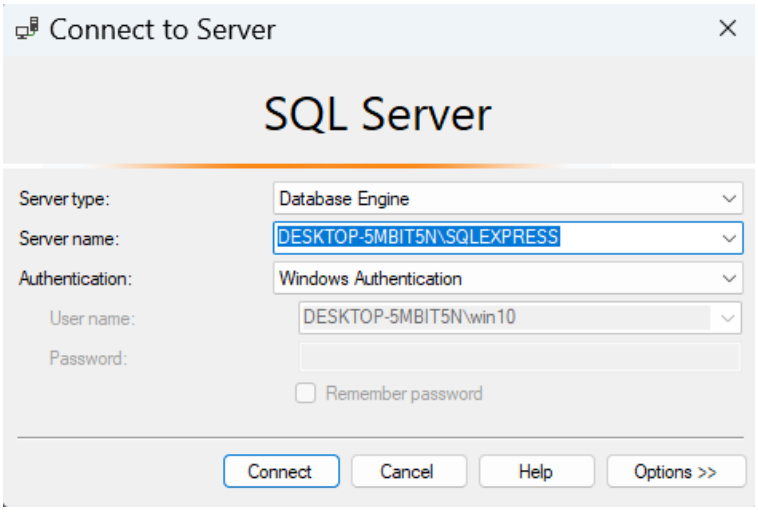
Setup Instructions:

Minimum Requirements:

- IDE that supports C# and Windows Form, preferably Visual Studio
- .NET versions 7.0 and above
- SQL Server Management Software

To set up a successful connection:

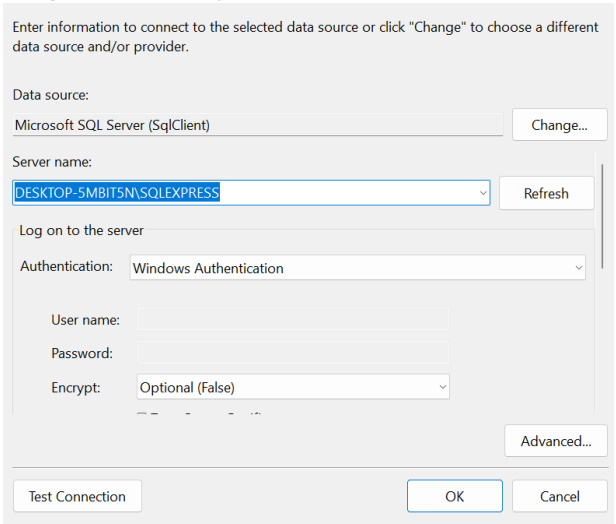
1. Open SQL Server Management Software such as Microsoft SQL Server Management Studio
2. Copy the Server name and connect to the server

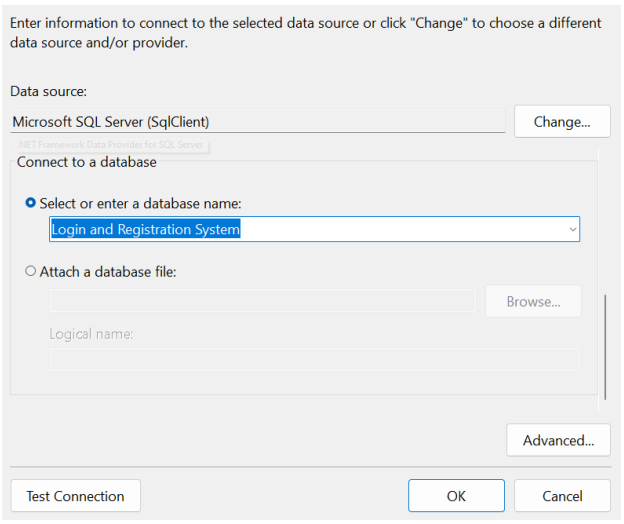


3. Open the solution on VS and open DB Manager.cs
4. In the DB Manager class, replace the highlighted text with your server name

```
namespace Login_and_Registration_System
{
    3 references
    public class DBManager
    {
        static string DB_Connection_String = @"Data Source=DESKTOP-5MBIT5N\SQLEXPRESS;Initial Catalog='Login and Registration System';Integrat
        SqlConnection myConnection;
        static bool check = false;
        1 reference
        public DBManager()
        {
```

5. Search Server Explorer -> Data Connections -> Add Connections and replace the Server name with your Server name and select the database “Login and Registration System”





6. Run the program and a message box will pop up indicating that the database has connected successfully.
7. If the SqlClient is not installed:

   Project -> Manage NuGet Packages -> Browse for SqlClient -> Install

Technology Stack:

- **.NET:** C# framework for building user interfaces as well as backend logic.
- **OOP:** Object-oriented programming to capsulate UIs as components, as well as making a database manager and API callers.
- **SQL:** Database for storing user information and running queries on the database.

Design decisions:

- Implemented a DB Manager that has the functions to execute the queries in the SQL server and ensure a valid connection
- Created a Controller that includes all the query functions called by the widgets

Assumptions and Limitations:

Assumptions:

- **Phone Validation:** Phone-number is a local Egyptian number.
- **Case Sensitivity**: The system assumes case-sensitive comparisons for usernames and passwords.
- **Use of strong passwords:** Strong passwords were provided by the users when registering, therefore there are no suggestions on strong passwords.

Limitations:

- **Security**: Storing and handling passwords in plaintext is insecure. This method is vulnerable to various attacks, including credential theft if the source code is exposed.
- **Error Handling**: Basic error handling only (e.g., notifying of incorrect credentials). Does not lock out users after multiple failed attempts.
- **No Advanced Features**: Lacks features like password recovery, user registration, or multi-factor authentication.