Connecting SQL Database with C# in Windows Forms Application

Step 1: Open Visual Studio

- Launch **Visual Studio** on your computer.

Step 2: Create a New Windows Forms Project

1. Go to **File → New → Project**.
2. Select **Windows Forms App (.NET Framework)**.
3. Click **Next**.

Step 3: Rename and Save the Project

1. Enter an appropriate name for your project (e.g., `StudentDatabaseApp`).
2. Choose a proper location on your computer to save the project.
3. Click **Create**.

Step 4: Create the Interface

1. From the **Toolbox**, drag and drop:
    - **2 Labels**: Rename them to "Name" and "Address".
    - **2 Textboxes**: For entering name and address.
    - **2 Buttons**: Rename them to "Insert" and "Clear".
2. Arrange them neatly to create a user-friendly interface.

Step 5: Create a Database

1. In the **Solution Explorer**, right-click on your project name.
2. Select **Add → New Item → Service-based Database**.
3. Rename the database (e.g., `StudentDB.mdf`) and click **Add**.

Step 6: Create a Table

1. Open the **Server Explorer** (from the View menu).
2. Expand the newly created database under **Data Connections**.
3. Right-click on **Tables → Add New Table**.

Step 7: Add Columns

1. Define the following columns in the table designer:
    - **Name**: `varchar(50)`
    - **Gender**: `varchar(50)`
    - **Address**: `varchar(150)`
    - **DOB**: `date`
    - **ID**: Set as `Primary Key` and enable **Identity Specification**:
        - Select the **ID** column.
        - In the **Column Properties**, set **Identity Specification → (Is Identity)** to **True**.
2. Click **Update → Update Database** to save the table.

Step 8: Insert Sample Data (Optional)

1. In **Server Explorer**, right-click on **Tables → Refresh**.
2. Locate your newly created table, right-click, and select **Show Table Data**.
3. Add some sample rows directly into the database table.

Step 9: Create a Config Class

1. In the **Solution Explorer**, right-click on your project name.
2. Select **Add → New Item → Class**.
3. Rename the class to `config.cs`.
4. Add the following code in `config.cs`:

```
public SqlConnection con;
public string str;

public config()
{
   str = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=D:\Day02\db\db\form.mdf;Integrated
Security=True";

   try {
      con = new SqlConnection(str);
      con.Open();

   }
   catch (Exception e)
   {
      MessageBox.Show("Error: "+e.Message);
   }
}

}
```

Step 10: Get the Connection String

1. Go to the **Server Explorer**.
2. Select your database and view its **Properties**.
3. Copy the **Connection String** and paste it into the `str` variable in `Config.cs`.

Step 11: Add Form Code

1. Open **Form1.cs** in your project.
2. Replace or add the following code:

Step 12: Test the Application

1. Build and run the application by pressing **F5**.
2. Enter data in the textboxes and click the **Insert** button to add records to the database.

```
config o = new config();
    private void button1_Click(object sender, EventArgs e)
    {
      try
      {
        if(textBox1.Text!="" &&  textBox3.Text!="")
        {
          string qry = "  INSERT INTO [Table]
(name,gender,address,dob)VALUES('"+textBox1.Text+"','"+comboBox1.Text+"','"+textBox3.Text+"','"
+dateTimePicker1.Value.ToString("yyyy/MM/dd")+"')";
          SqlCommand cmd = new SqlCommand(qry, o.con);
          cmd.ExecuteNonQuery();
          MessageBox.Show("Data Stored");
        }
        else
        {
          MessageBox.Show("All FIELDS ARE REQUIRED");
        }

      }
      catch(Exception x)
      {
        MessageBox.Show(x.Message);
      }
    }
```