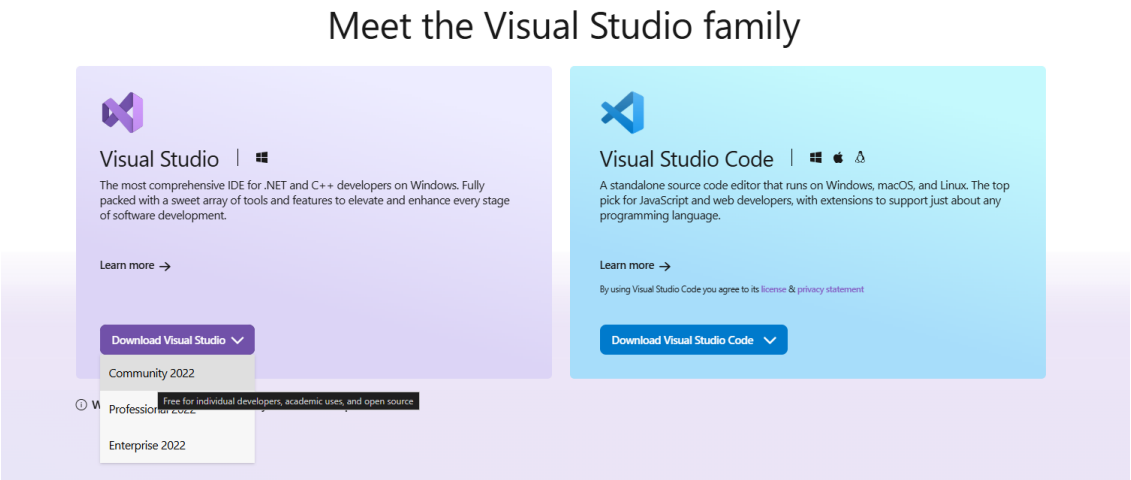
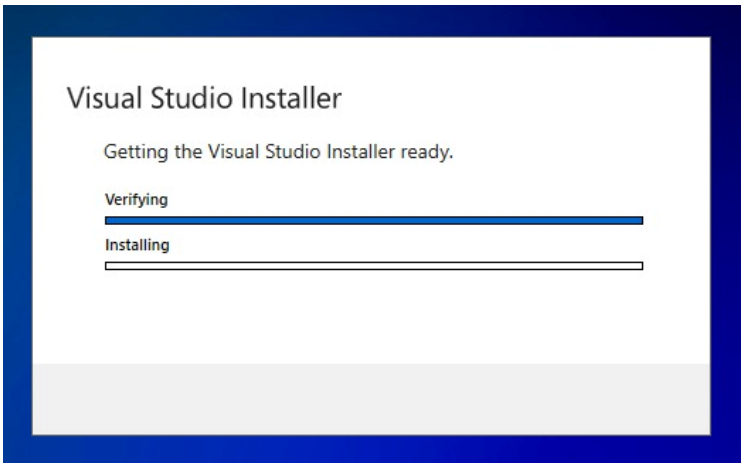
DBMS – Lab1

1. Go to https://visualstudio.microsoft.com/ and download **Visual Studio Community**.

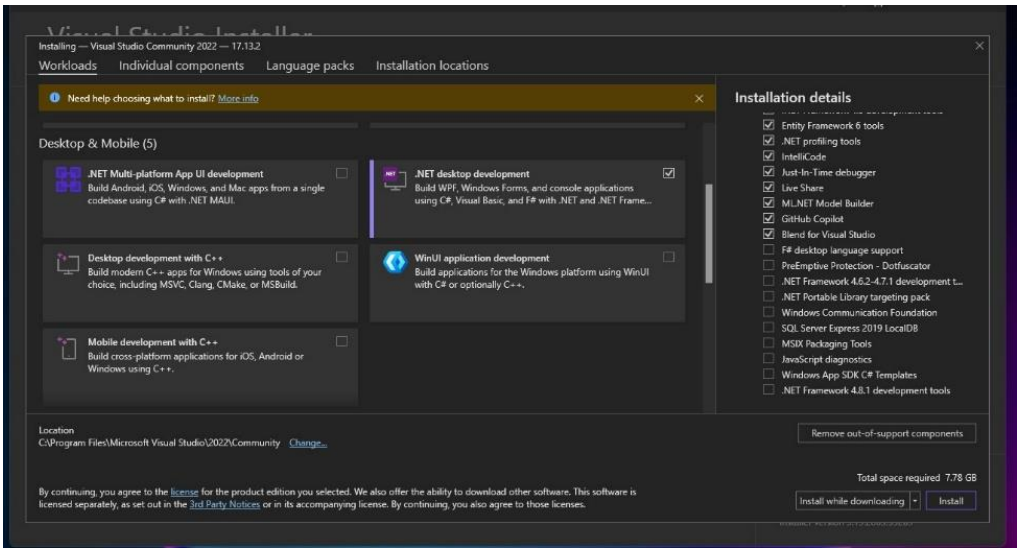


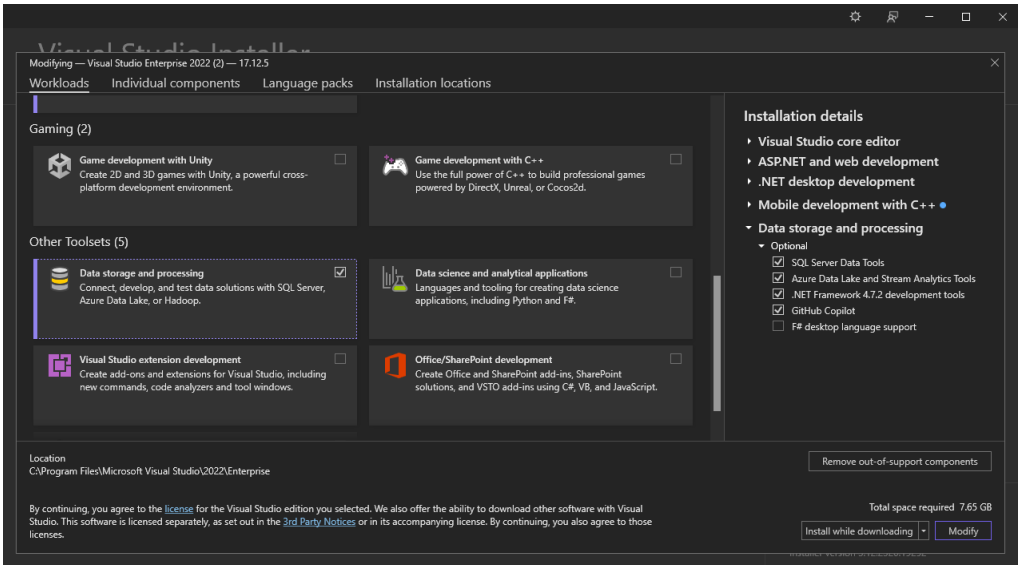
2. On your local computer, in the **Downloads** folder you will find the executable for Visual Studio Community Installer. Click on it so the installation will start.

3. After clicking on the now-downloaded executable file, the installation process will begin.

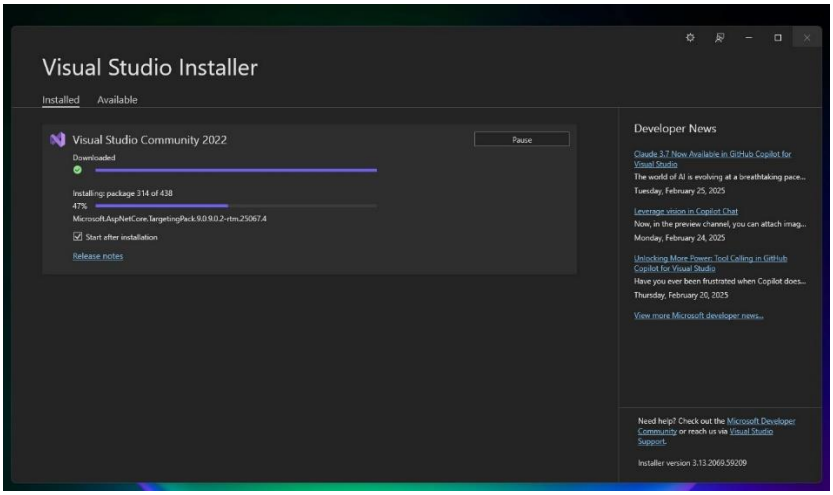


4. A new window will pop-up, asking you to select the packages you want to include. Make sure you select **“.NET desktop development”** + “**Data Storage and processing**”

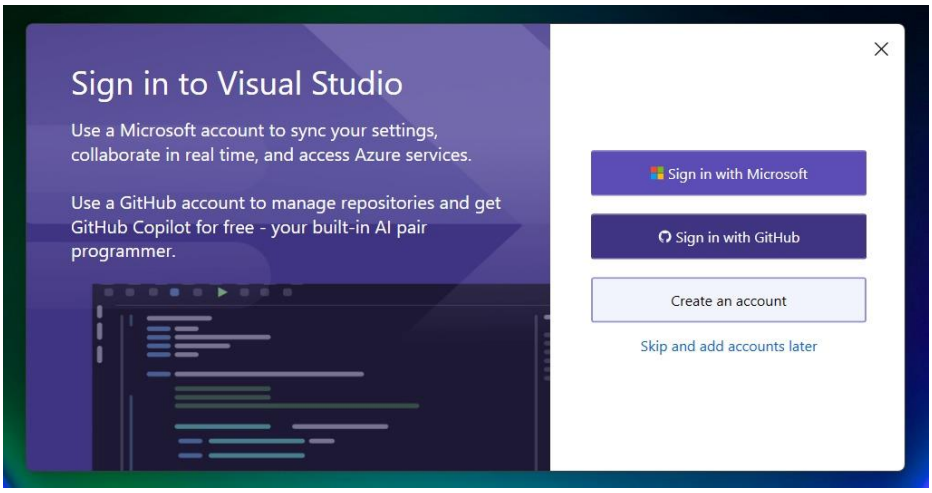


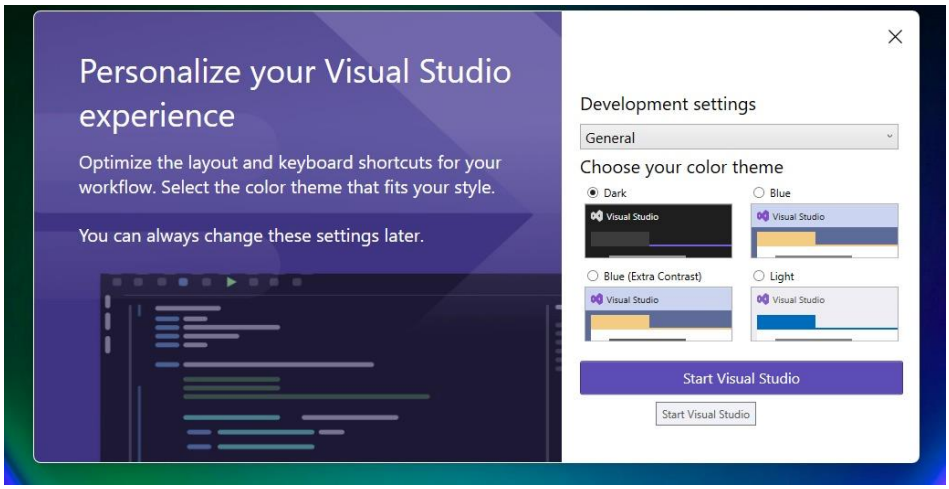


5. After selecting it, press **Install** and wait for the installation to complete. You might be asked to reboot your computer.

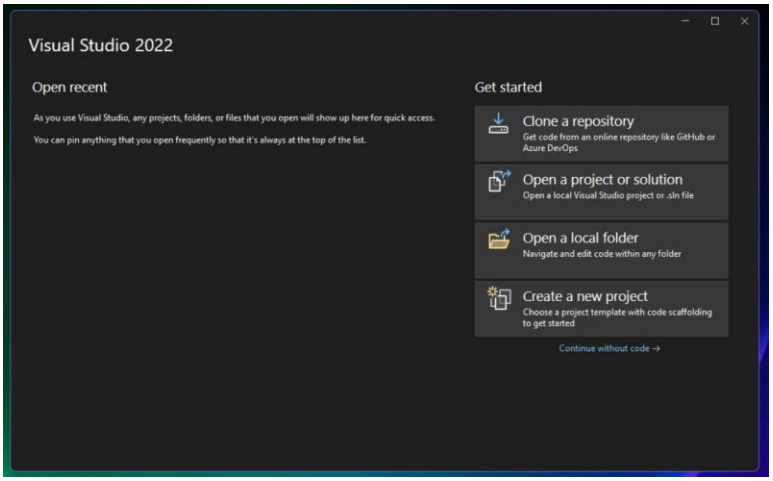


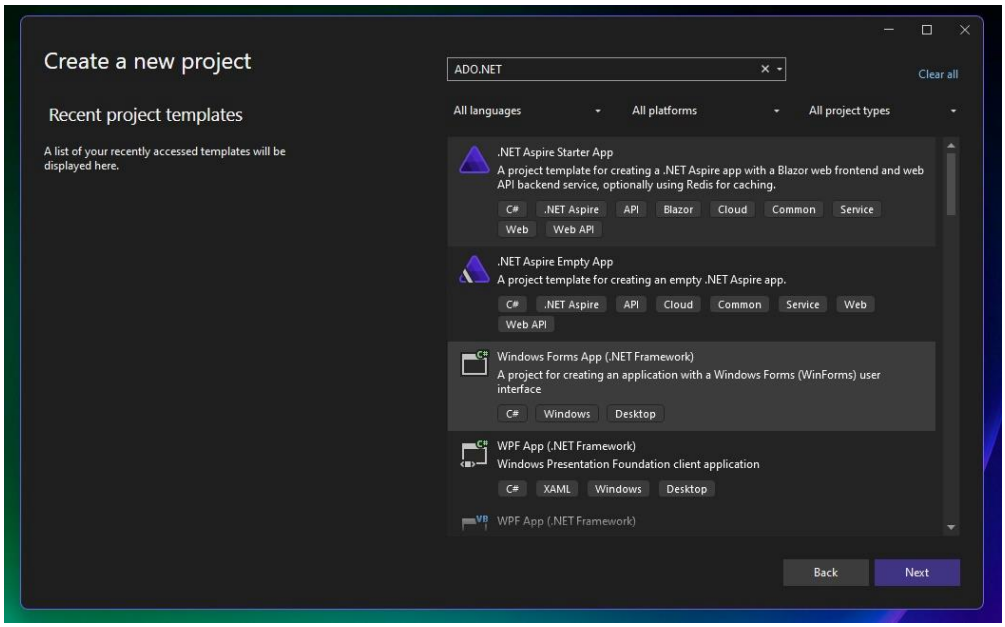
6. After rebooting, open the Visual Studio. You might be asked to Sign in with an account and personalize your IDE by choosing a color theme. We can skip it for now.



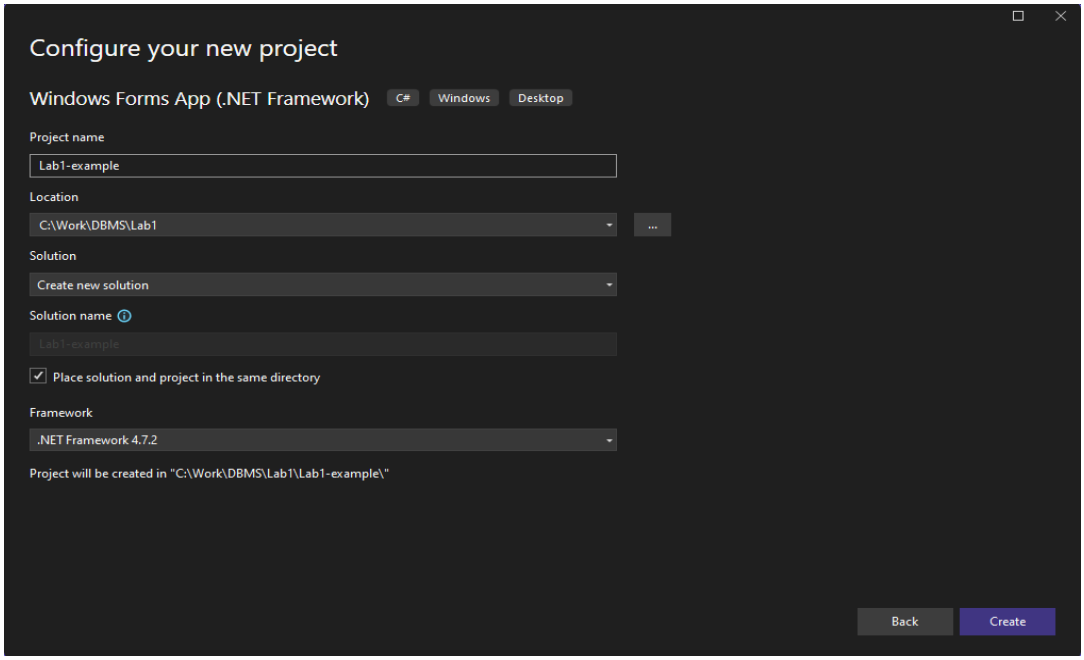


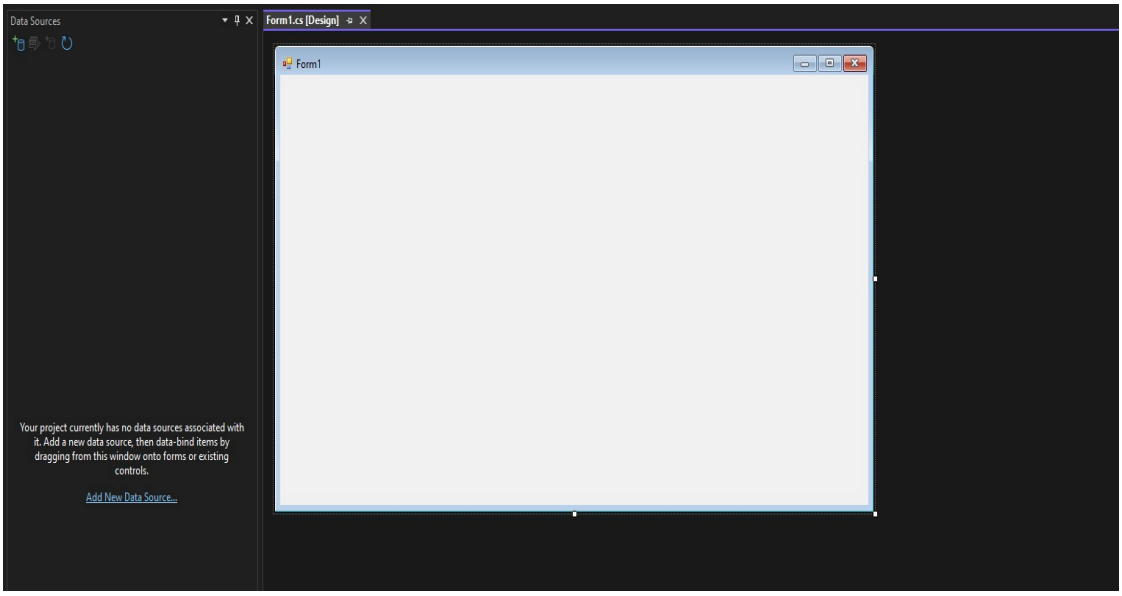
7. Let's create a new project. Make sure that after clicking "**Create a new Project**", you write in the search box ADO.NET and select "**Windows Forms App (.NET Framework)**" as a template.



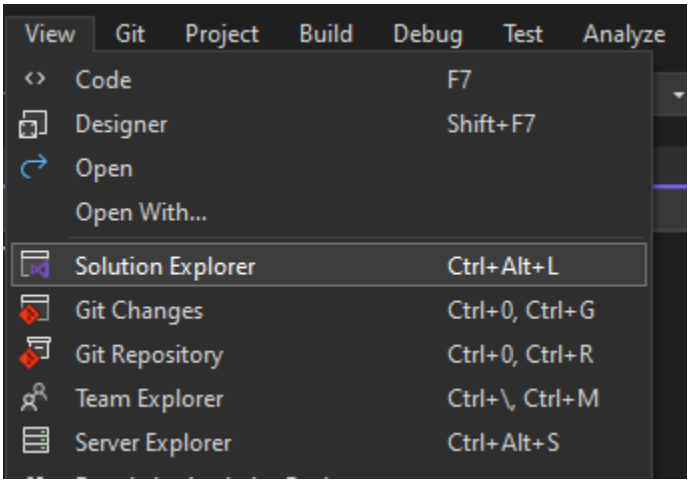


8. Choose a name and a location you will not forget about I and then your new Windows Forms application is ready.



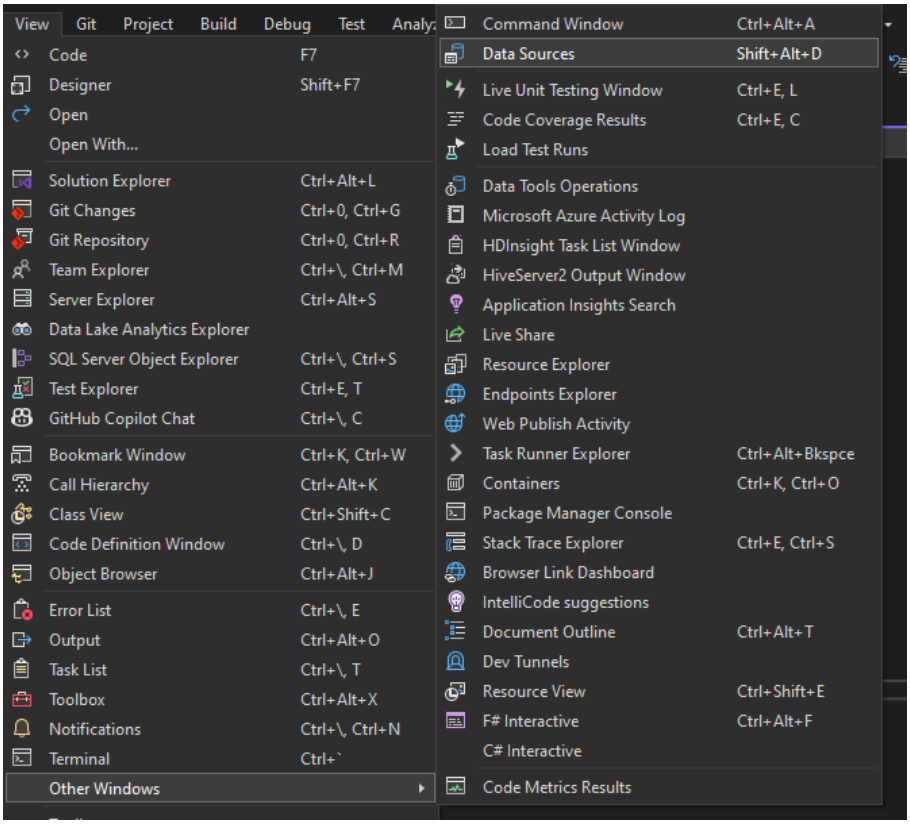


9. If you do not have the **Solution Explorer** in your right side, and you want it there you can find it on **View -> Solution Explorer**

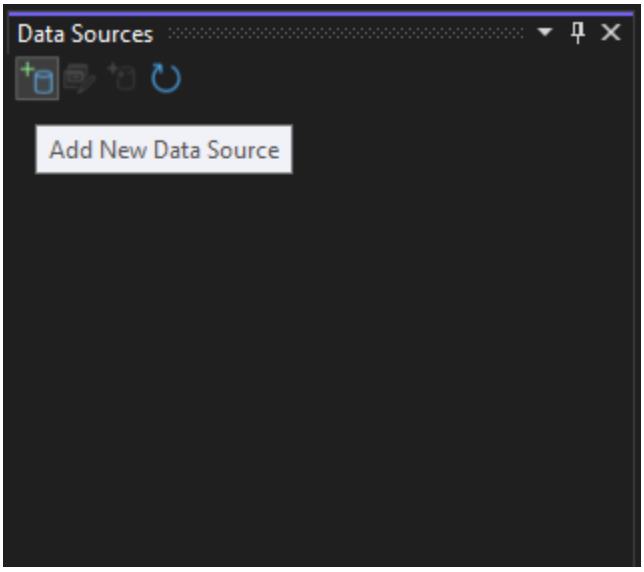


10. In order to connect to a database we need a Data Source configured. You can do it either manually, in code or from UI through Data Source Window.

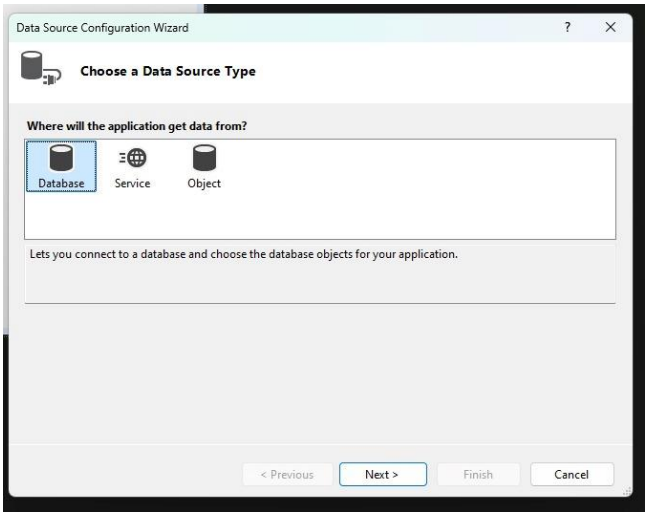
11. In your left side you should have de **Data Source** window. If you can't find it, it is in **View -> Other Windows -> Data Sources**



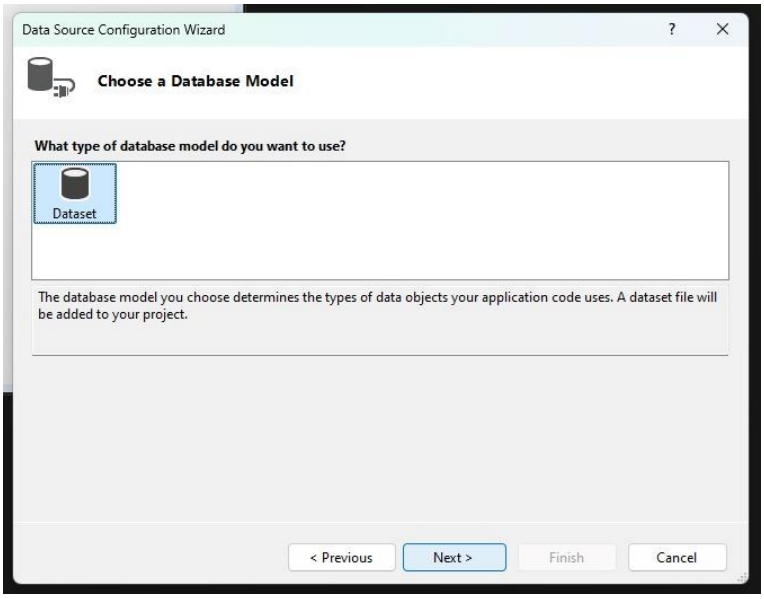
12. Now you have the Data Source window prepared. Clcik on "**Add New Data Source**".



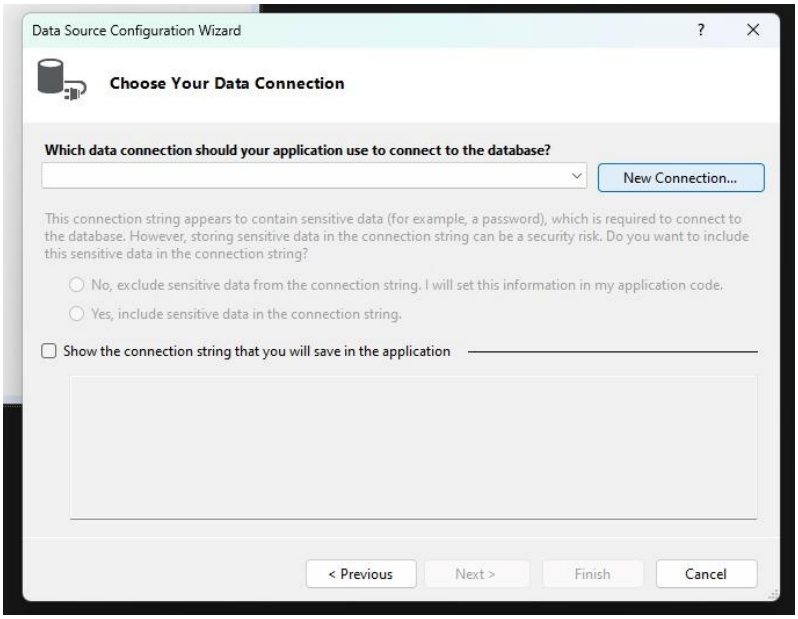
13. A pop-up will appear. Select the "**Database**" option to make your application able to read data from you DB. Click **Next**.



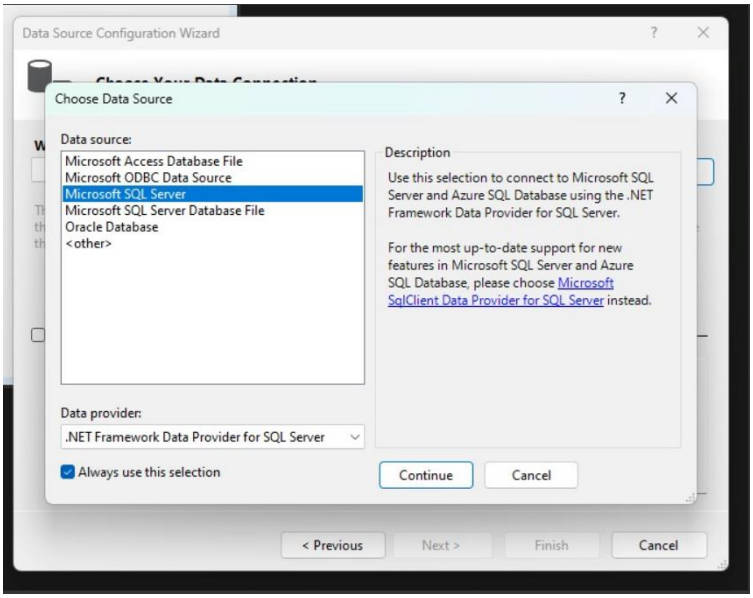
14. In order to fetch the data from database and store in in your application, you need to select "**Dataset**" structure as data model. Click **Next**.



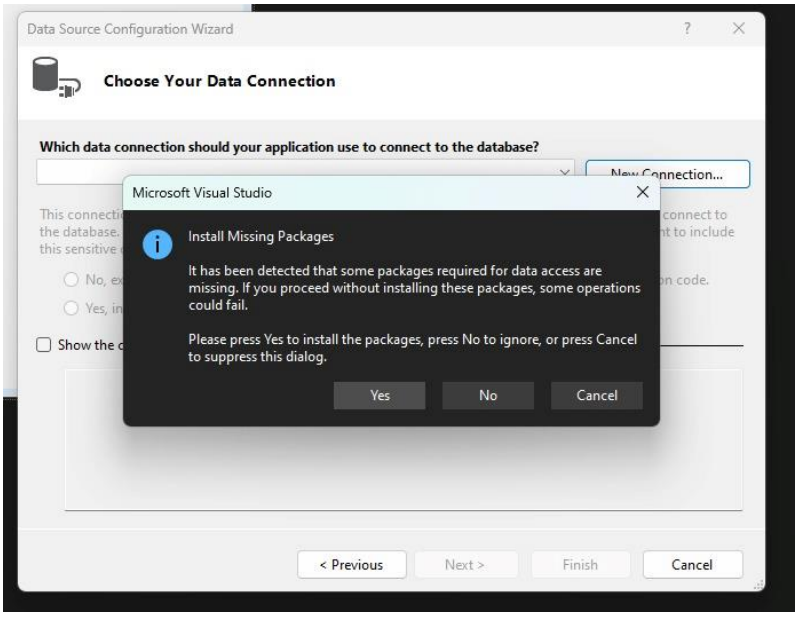
15. You will now be asked to connect to your database. Select "**New Connection…**".



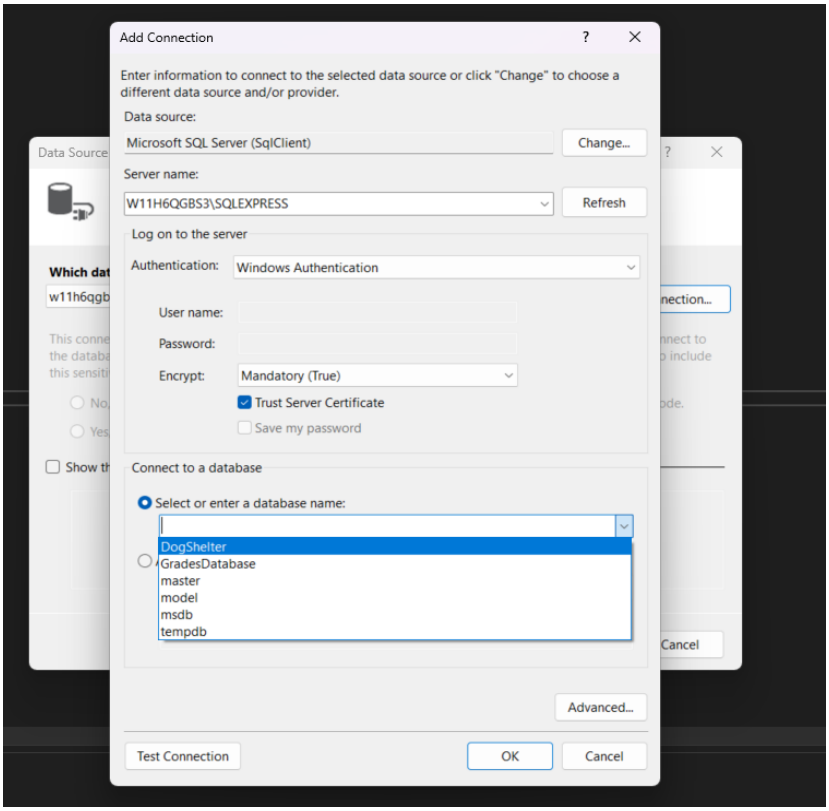
16. Now you need to choose the data source type. Because we are going to use last semesters' database, you will ne to select "**Microsoft SQL Server**". Click **Continue**.



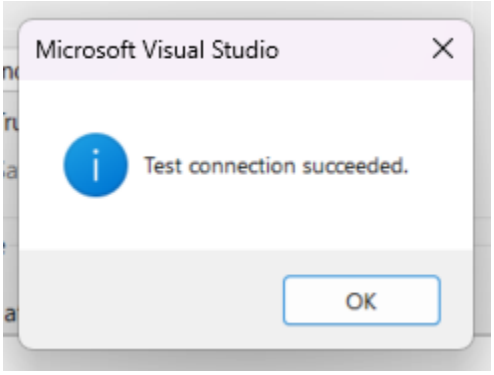
17. You might be asked to install a few missing packages. If so, install them.



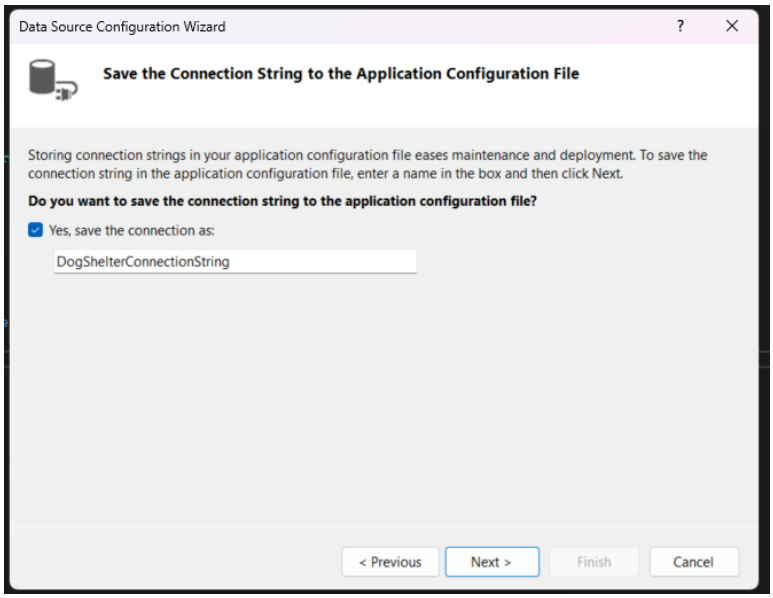
18. A new window will pop, asking for more details regarding your database connection. Make sure you input your **server name** (do not wait for the list to load because it will take too long) and your **database name**. Select "**Trust Server Certificate**"



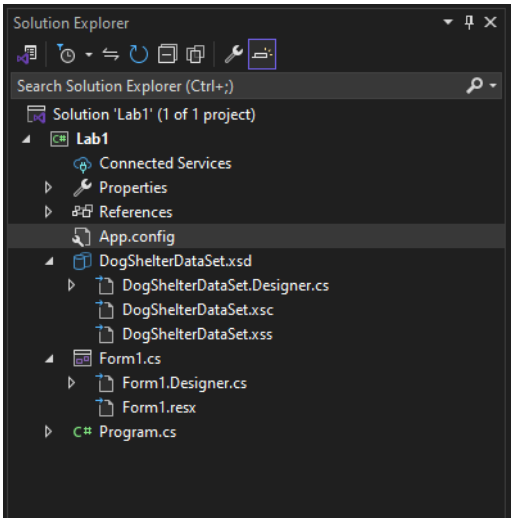
19. Click on “Test Connection” to see if that was successful.



20. The new connection is established. Click **Next**.
    * if you are curious, you can click on “**Show the connection string that you will save in the application**” so you can see how it looks like.
21. Next, you will be asked if you want to save the connection string in the applications’ configuration file. Select “**Yes**”.

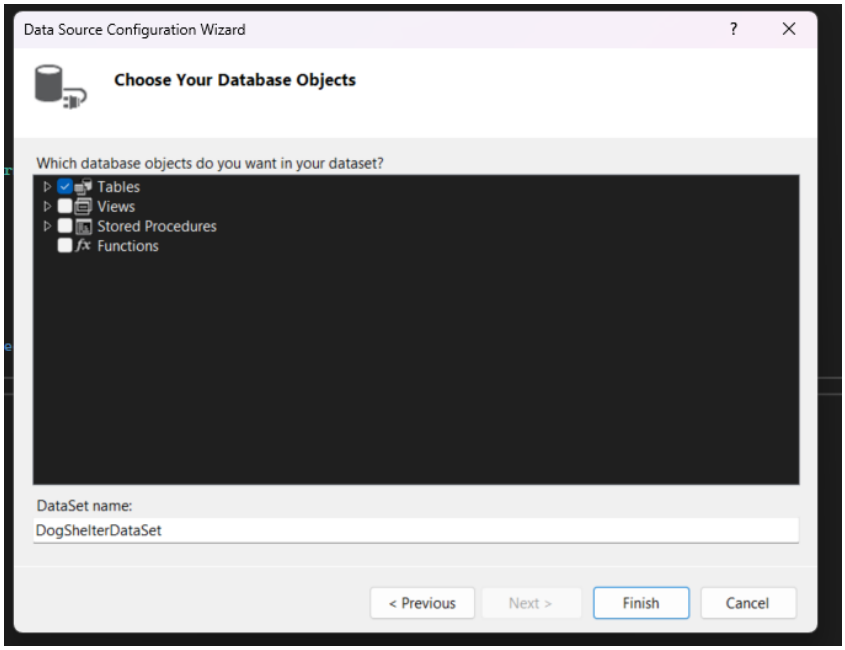


➔ On **App.cofig** file from **Solution Explorer** window, you can now find the embedded connection string.

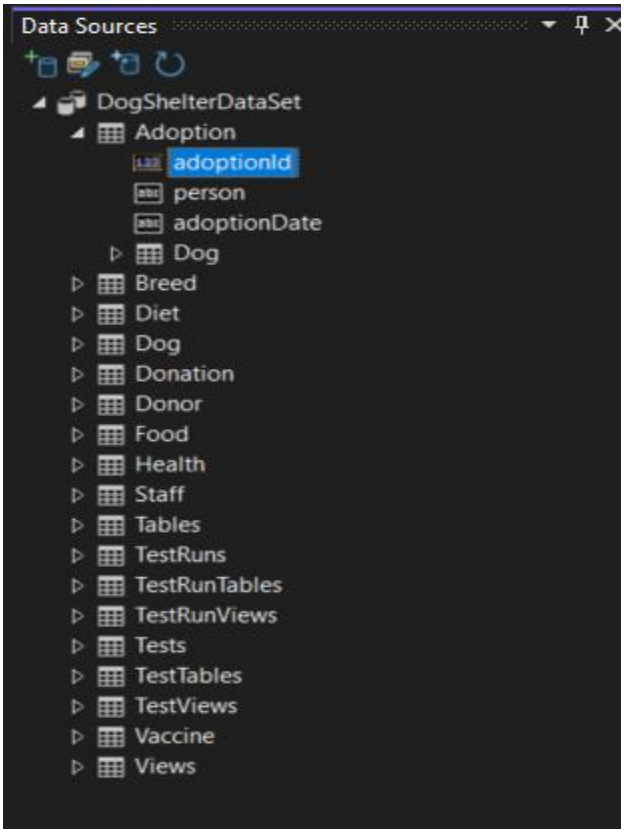



```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <configSections>
    </configSections>
    <connectionStrings>
        <add name="Lab1.Properties.Settings.DogShelterConnectionString"
            connectionString="Data Source=W11H6QGBS3\SQLEXPRESS;Initial Catalog=DogShelter;Integrated Security=True;Encrypt=True;TrustServerCertificate=True"
            providerName="System.Data.SqlClient" />
    </connectionStrings>
    <startup>
        <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
    </startup>
</configuration>
```

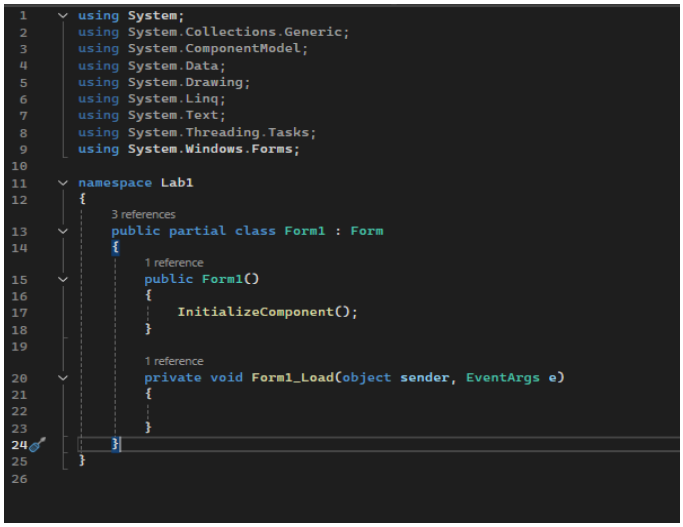
22. You will be asked to import objects from your database into your C# application. You cand select all of them or only ones you will need. You can also see the name of the dataset which will keep this information in "**DataSet name:**" box. Click **Finish**.



23. Now the connection is established, and you can find the tables in the Data Source window. Let's check the connection through C# code and forms.

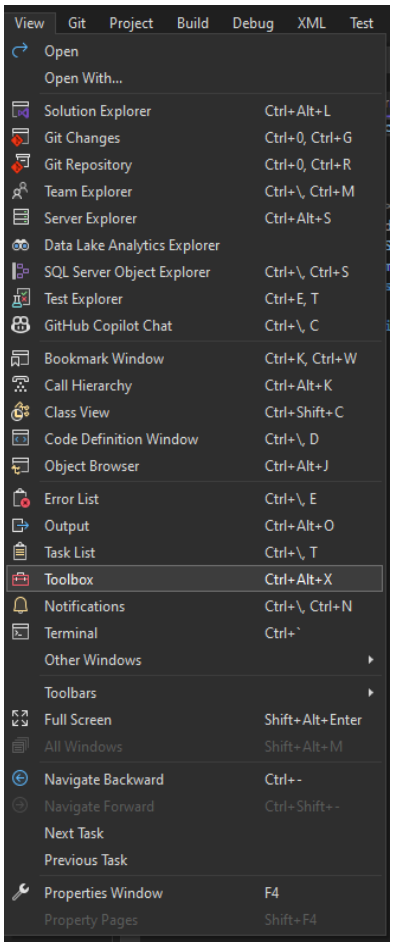


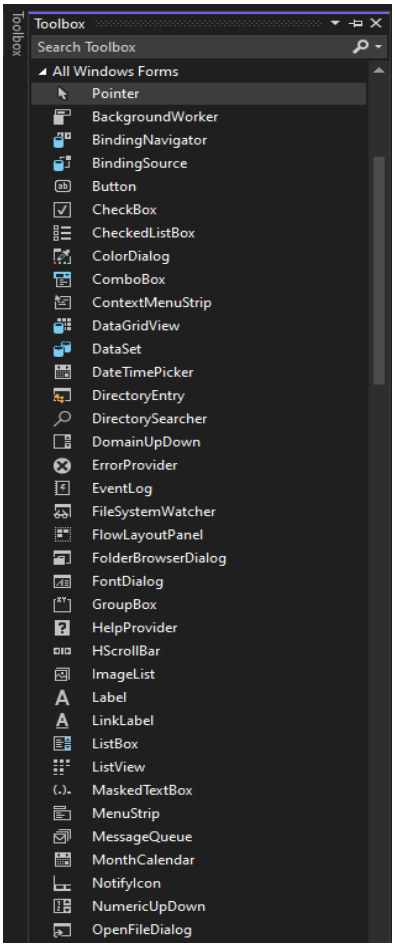
24. *Windows Forms represent the Windows User Interface (UI).* If you want to see the code behind the Form from your newly created application you can just **double-click** on it. This will take you to **Form1.cs** where you will create the logic.



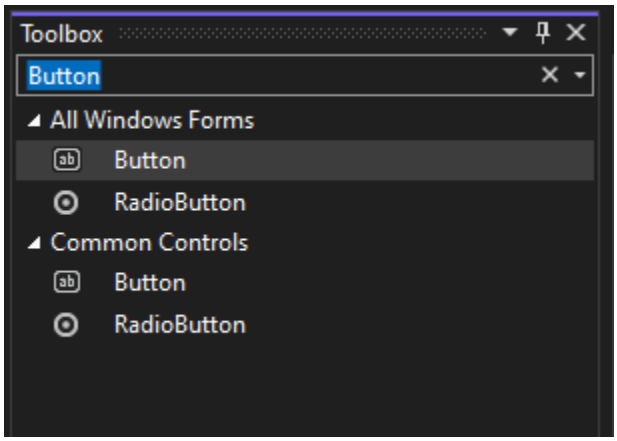
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Lab1
{
    3 references
    public partial class Form1 : Form
    {
        1 reference
        public Form1()
        {
            InitializeComponent();
        }

        1 reference
        private void Form1_Load(object sender, EventArgs e)
        {

        }
    }
}
```

25. In forms you can add any UI elements you might think of starting from labels, to buttons and even tables. To access these items, you need to go to **View->Toolbox**. This will bring the Toolbox window in the right side of your screen. You can pin it for later use.





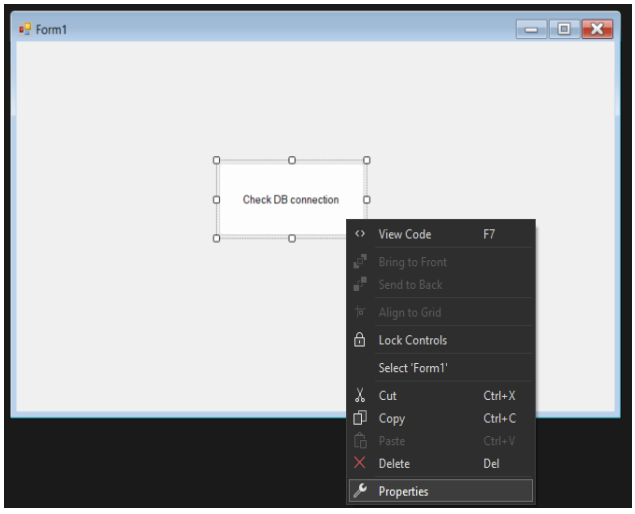
26. Lets add a Button to our form in order to check the database connection. Make sure to input "**Button**" in the search box inside ToolBox window from right side of your IDE.

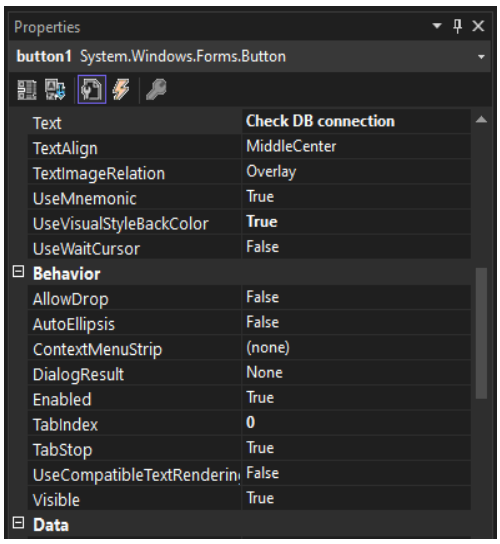


27. Select the Button UI element and drag it into your form. If you want to add logic to check the connection to the database, you can **double-click** on it. This will take you to the place in which you can implement it.

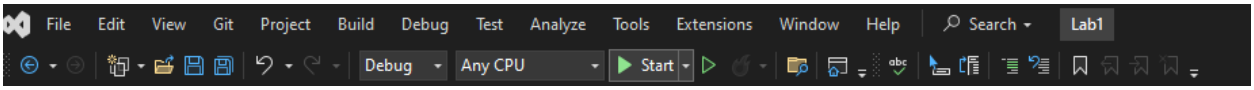
```
3 references
public partial class Form1 : Form
{

    string connectionString = @"Data Source=W11H6QGBS3\SQLEXPRESS;Initial Catalog=DogShelter;Integrated Security=True;Encrypt=True;TrustServerCertificate=True
    1 reference
    public Form1()
    {
        InitializeComponent();
    }

    1 reference
    private void Form1_Load(object sender, EventArgs e)
    {

    }

    1 reference
    private void button1_Click(object sender, EventArgs e)
    {
        try
        {
            SqlConnection cn = new SqlConnection(connectionString);
            if (cn.State == System.Data.ConnectionState.Closed)
                cn.Open();

                MessageBox.Show("Test connection succeeded.", "Message", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Message", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
```

28. If you want to modify the button's style, color, font, add text to the UI components, you can do that by **right-clicking** on the UI element and then press "**Properties**". A new window will appear in the lower right side of Visual Studio allowing you to make those changes.





29. Now that everything is in place, you can check if that works, you can run the application by clicking “Start” button on the top of the IDE.



30. Click the “**Check DB Connection**” button. If the connection is successful, a message box will appear.