أساسيات ومفاهيم البرمجة
تصميم واجهات برامج ويندوز
Windows Forms App
باستخدام برنامج Visual Studio 2022
بالعربي CSharp 4
FUNCTION()
if
VAR
Write("HELLO WORLD")
WHILE
DO
لغة C#
مجتمع المبرمجين العرب

تصميم واجهات البرامج

لغة C#

Windows Forms App

سلسلة

سي شارب بالعربي

الجزء 4

مجتمع المبرمجين العرب

تصميم واجهات البرامج، لغة C#

Windows Forms App

باستخدام برنامج Visual Studio 2022

CSharp بالعربي

المحتويات:

١- ملخص عن برنامج Visual Studio
٢- إنشاء أول تطبيق وحدة تحكم calculator
٣- إنشاء أول تطبيق بسيط Windows Form
٤- إنشاء تطبيق عارض الصور Windows Form
٥- إنشاء تطبيق اختبار الرياضيات Windows Form
٦- إنشاء تطبيق لعبة مطابقة matching game
٧- الاتصال بقاعدة بيانات Access في تطبيقات .NET Framework
٨- البدء في استخدام ASP.NET Core لإنشاء أول تطبيق ويب
٩- مميزات برنامج Visual Studio

١ ملخص عن برنامج Visual Studio

مرحبًا بك في برنامج Visual Studio IDE

بيئة التطوير المتكاملة integrated development environment (IDE) عبارة عن برنامج غني بالميزات، يدعم العديد من جوانب تطوير البرامج، تعد بيئة التطوير المتكاملة Visual Studio IDE بمثابة منصة إطلاق إبداعية، يمكنك استخدامها لتحرير وتصحيح أخطاء وبناء التعليمات البرمجية، ثم نشر التطبيق. بالإضافة إلى المحرر القياسي وتصحيح الأخطاء الذي توفره معظم بيئات التطوير المتكاملة، يتضمن Visual Studio برامج تجميع وأدوات استكمال التعليمات البرمجية ومصممي الرسوم والعديد من الميزات الأخرى لتحسين عملية تطوير البرامج.

تثبيت Visual Studio 2022

مرحبًا بك في Visual Studio 2022 في هذا الإصدار، من السهل اختيار وتثبيت الميزات التي تحتاجها فقط.

ملحوظة:

إذا كنت بحاجة إلى تثبيت Visual Studio برمجيًا، راجع استخدام معلمات سطر الأوامر

هل أنت مستعد للتثبيت؟ ستوضح لك هذا القسم كيفية التثبيت خطوة بخطوة.

الخطوة1 تأكد من أن جهاز الكمبيوتر الخاص بك جاهز لاستخدام Visual Studio

قبل البدء في تثبيت Visual Studio

- تحقق من متطلبات النظام تساعدك هذه المتطلبات على معرفة ما إذا كان جهاز الكمبيوتر الخاص بك يدعم Visual Studio 2022
- تأكد من أن المستخدم الذي يقوم بالتثبيت الأولي لديه أذونات المسؤول على الجهاز. لمزيد من المعلومات، راجع أذونات المستخدم وVisual Studio
- قم بتطبيق أحدث تحديثات Windows تضمن هذه التحديثات أن لديك جهاز كمبيوتر يحتوي على أحدث تحديثات الأمان والمكونات النظامية المطلوبة لبرنامج Visual Studio
- Restart تضمن إعادة التشغيل عدم عرقلة أي تثبيتات أو تحديثات معلقة لتثبيت Visual Studio

- تحرير مساحة قم بإزالة الملفات والتطبيقات غير الضرورية من محرك أقراص النظام لديك، على سبيل المثال، عن طريق تشغيل تطبيق Disk Cleanup تنظيف القرص

يمكنك تثبيت Visual Studio 2022 جنبًا إلى جنب مع إصدارات أخرى. لمزيد من المعلومات، راجع Visual Studio 2022 platform targeting and compatibility وتثبيت Install Visual Studio versions side-by-side

الخطوة2 تحديد الإصدار والنسخة من Visual Studio المراد تثبيتها

حدد الإصدار والنسخة التي تريد تثبيتها من Visual Studio الخيارات الأكثر شيوعًا هي:

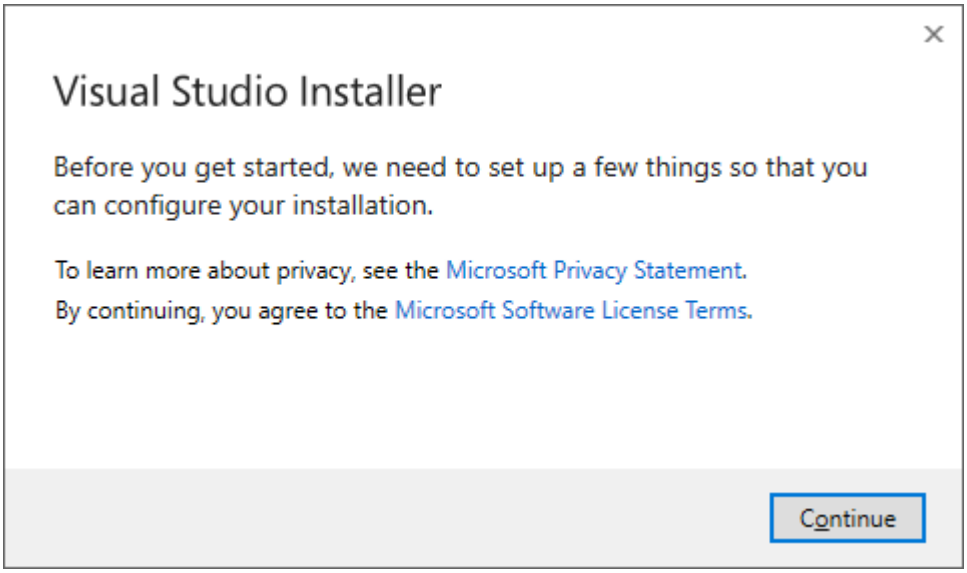
- أحدث إصدار من Visual Studio 2022 يتم استضافته على خوادم Microsoft لتثبيت هذا الإصدار، حدد Download Visual Studio ثم اختر الإصدار الذي تريده. يقوم المثبت بتنزيل برنامج تمهيد صغير إلى مجلد التنزيلات لديك.
- إذا كان لديك Visual Studio مثبتًا بالفعل، فيمكنك تثبيت إصدار آخر بجانبه عن طريق اختيار الإصدار المتوفر في علامة التبويب المتوفرة في مثبت Visual Studio
- يمكنك تنزيل برنامج التمهيد لإصدار معين من صفحة سجل إصدارات Visual Studio 2022 واستخدامه لتثبيت Visual Studio

الخطوة3 بدء التثبيت

إذا قمت بتنزيل ملف bootstrapper فيمكنك استخدامه لتثبيت Visual Studio ستحتاج إلى أذونات المسؤول. يقوم bootstrapper بتثبيت أحدث إصدار من Visual Studio Installer المثبت هو برنامج منفصل يوفر كل ما تحتاجه لتثبيت Visual Studio وتخصيصه.

١. من مجلد التنزيلات، انقر نقرًا مزدوجًا فوق برنامج التمهيد المسمى VisualStudioSetup.exe أو باسم مثل vs_community.exe لبدء التثبيت.

٢. إذا تلقيت إشعارًا بالتحكم في حساب المستخدم، فاختر "نعم يطلب منك مربع الحوار الإقرار بشروط ترخيص وبيان خصوصية Microsoft اختر متابعة **Continue**

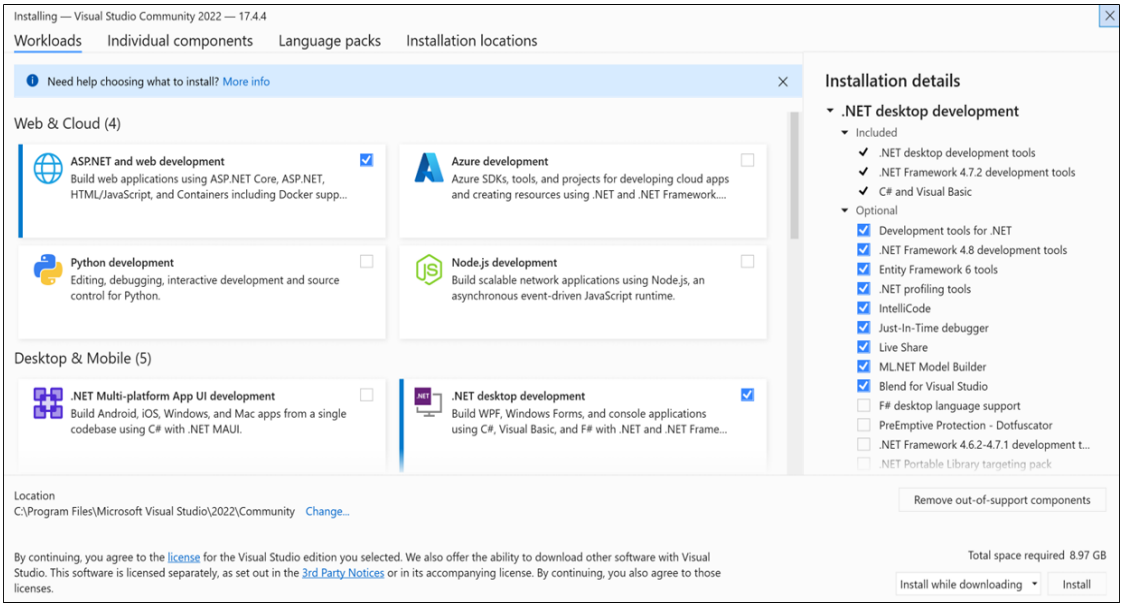


يتم فتح برنامج Visual Studio Installer يمكنك أيضًا تثبيت أي منتج تقدمه علامة التبويب Available "متوفر" في برنامج التثبيت Visual Studio Installer's Available tab

الخطوة4 اختيار أحمال أو عبء العمل workloads

بعد تثبيت Visual Studio Installer يمكنك استخدامه لتخصيص التثبيت الخاص بك، عن طريق تحديد مجموعات الميزات أو أحمال العمل التي تريدها. وإليك الطريقة.

١. حدد حجم العمل الذي تريده في Visual Studio Installer



راجع ملخصات أحمال العمل لتحديد أحمال العمل التي تدعم الميزات التي تحتاج إليها. على سبيل المثال، اختر أحمال العمل الخاصة بتطوير ASP.NET and web development لتحرير صفحات الويب ASP.NET باستخدام Web Live

Previewأو إنشاء تطبيقات ويب سريعة الاستجابة باستخدام Blazor يمكنك الاختيار من أحمال العمل الخاصة بسطح المكتب والهواتف المحمولة Desktop & Mobile لتطوير تطبيقات متعددة الأنظمة باستخدام C# أو مشاريع C++ التي تستهدف C++20

٢. بعد اختيار أحمال العمل التي تريدها، حدد تثبيت **Install**

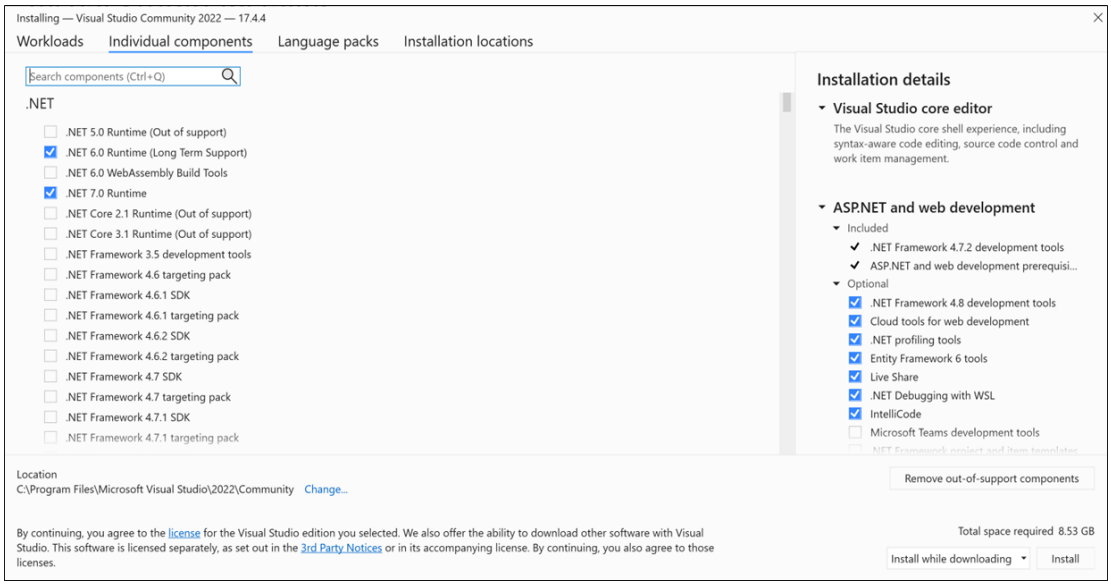
بعد ذلك، تظهر شاشات الحالة التي تعرض تقدم تثبيت Visual Studio الخاص بك.

نصيحة:

في أي وقت بعد التثبيت، يمكنك تثبيت أحمال العمل أو المكونات التي لم تقم بتثبيتها في البداية. إذا كان برنامج Visual Studio مفتوحًا، فانتقل إلى الأدوات، الحصول على الأدوات والميزات Tools > Get Tools and Features مما يفتح برنامج Visual Studio Installer أو افتح برنامج Visual Studio Installer من قائمة ابدأ Start ومن هناك، يمكنك اختيار أحمال العمل أو المكونات التي ترغب في تثبيتها. ثم اختر تعديل Modify

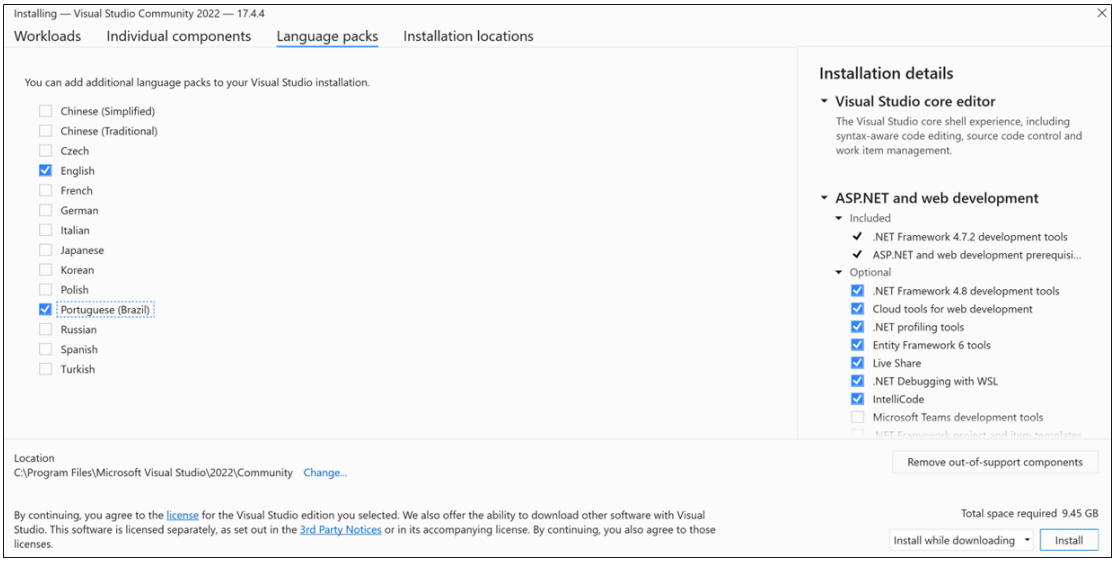
الخطوة5 اختيار المكونات الفردية (اختياري) individual components

إذا كنت لا تريد استخدام ميزة "أحمال العمل" لتخصيص تثبيت Visual Studio أو كنت تريد إضافة مكونات أكثر من تلك التي يتم تثبيتها بواسطة أحمال العمل، فيمكنك تثبيت أو إضافة مكونات فردية من علامة التبويب Individual components "المكونات الفردية" اختر ما تريده، ثم اتبع التعليمات.



الخطوة6 تثبيت حزم اللغات (اختياري) language packs

افتراضيًا، يحاول برنامج التثبيت مطابقة لغة نظام التشغيل عند تشغيله لأول مرة. لتثبيت Visual Studio باللغة التي تختارها، اختر علامة التبويب Language packs حزم اللغات من برنامج تثبيت Visual Studio ثم اتبع التعليمات.



تغيير لغة المثبت installer language من سطر الأوامر command line

هناك طريقة أخرى يمكنك من خلالها تغيير اللغة الافتراضية وهي تشغيل برنامج التثبيت من سطر الأوامر. على سبيل المثال، يمكنك إجبار برنامج التثبيت على التشغيل باللغة الإنجليزية باستخدام الأمر التالي:

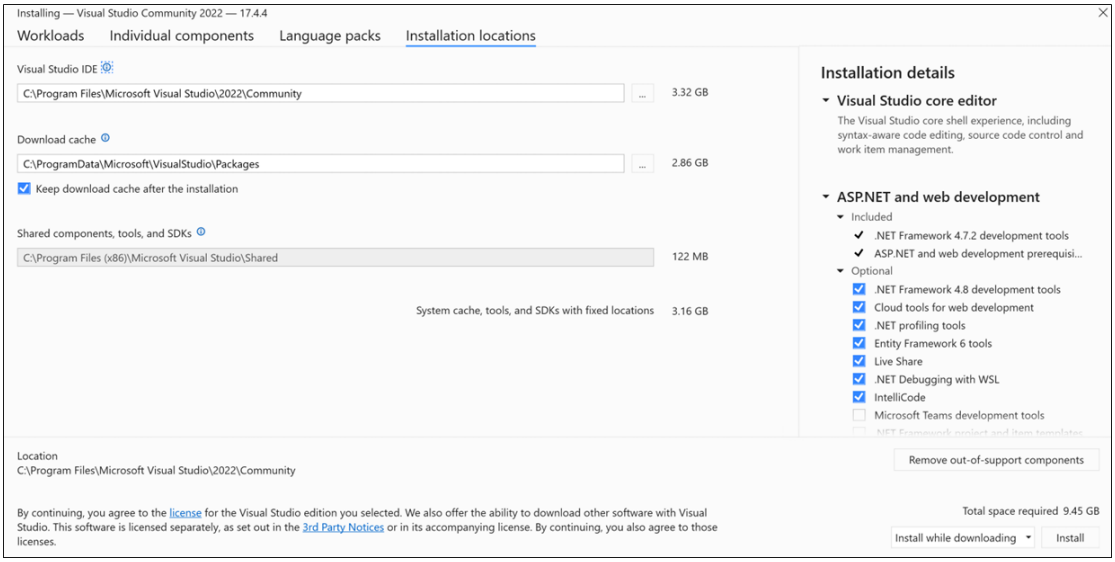
```
vs_installer.exe --locale en-US
```

يتذكر برنامج التثبيت هذا الإعداد عند تشغيله مرة أخرى. يدعم برنامج التثبيت اللغات المحلية التالية:

zh-cn, zh-tw, cs-cz, en-us, es-es, fr-fr, de-de, it-it, ja-jp, ko-kr, pl-pl, pt-br, ru-ru, and tr-tr.

الخطوة7 حدد موقع التثبيت (اختياري) installation location

يمكنك تقليل حجم تثبيت Visual Studio على محرك النظام لديك. لمزيد من المعلومات، راجع تحديد مواقع التثبيت.



مهم:

يمكنك تحديد محرك أقراص مختلف لـ Visual Studio IDE أو ذاكرة التخزين المؤقت للتنزيل فقط عند تثبيت Visual Studio لأول مرة. إذا قمت بالفعل بتثبيته وتريد تغيير محركات الأقراص، فيجب إلغاء تثبيت Visual Studio ثم إعادة تثبيته.

إذا قمت بتثبيت Visual Studio على جهاز الكمبيوتر الخاص بك من قبل، فلن تتمكن من تغيير مسار المكونات والأدوات ومجموعات تطوير البرامج المشتركة Shared components, tools, and SDKs يظهر باللون الرمادي. تتم مشاركة هذا الموقع بواسطة جميع تثبيتات Visual Studio

الخطوة8 تسجيل الدخول إلى حسابك (اختياري) Sign in to your account

على الرغم من عدم ضرورة تسجيل الدخول، إلا أن هناك العديد من المزايا للقيام بذلك.

يمكنك تقييم نسخة تجريبية مجانية من ة مدة 30 يومًا. إذا قمت بتسجيل الدخول، فيمكنك تمديد الفترة التجريبية إلى 90 يومًا. يعمل تمديد الفترة التجريبية لمدة 90 يومًا مرة واحدة فقط. لمواصلة استخدام Visual Studio بعد انتهاء الفترة التجريبية، قم بإلغاء قفلها باستخدام اشتراك عبر الإنترنت أو مفتاح المنتج.

لا تتطلب نسخة Visual Studio Community منك تسجيل الدخول. ومع ذلك، إذا طالبك التثبيت بتسجيل الدخول بشكل دوري، فقم بتسجيل الدخول لمواصلة استخدام Visual Studio Community دون انقطاع.

الخطوة9 البدء في التطوير developing

بعد اكتمال التثبيت، يمكنك البدء في التطوير باستخدام Visual Studio

١. حدد زر التشغيل Launch

٢. في نافذة البدء، اختر إنشاء مشروع جديد Create a new project

٣. في مربع البحث عن القالب، أدخل نوع التطبيق الذي تريد إنشاءه لعرض قائمة بالقالب المتاح، تعتمد قائمة القوالب على أحمال العمل التي اخترتها أثناء التثبيت. لعرض قوالب مختلفة، اختر أحمال عمل مختلفة.

ملحوظة:

سوف يتم شرح كيفية إنشاء برامج Windows Forms بالتفصيل في الوحدات القادمة.

يمكنك أيضًا تصفية بحثك عن لغة برمجة معينة باستخدام القائمة المنسدلة اللغة Language ويمكنك أيضًا تصفية البحث باستخدام قائمة المنصة Platform وقائمة نوع المشروع Project type

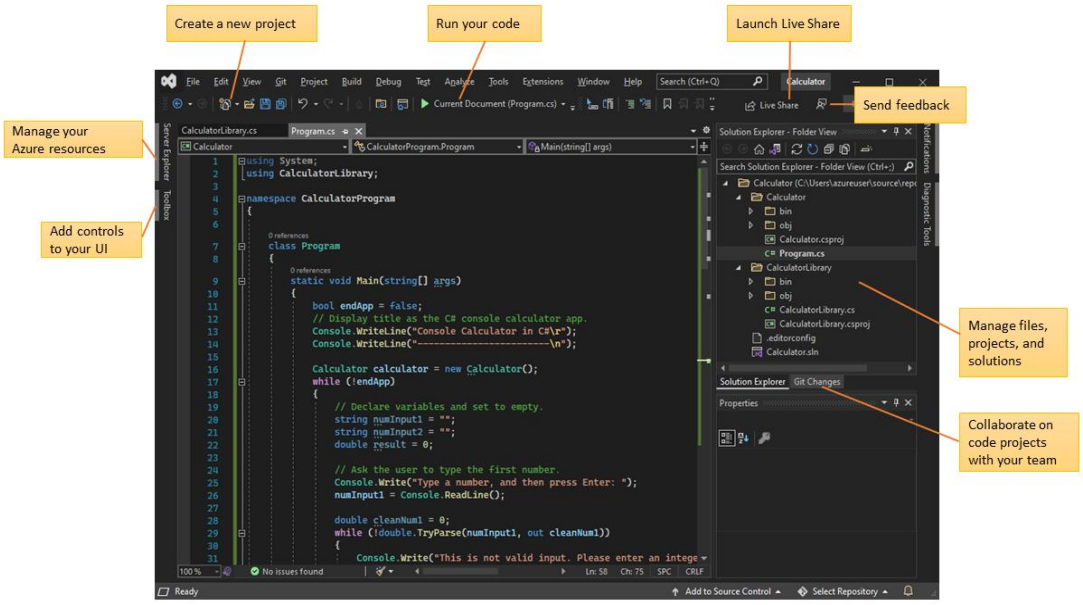
٤. حدد التالي Next قم بتوفير معلومات أخرى في مربعات الحوار التالية، ثم حدد إنشاء Create

يفتح Visual Studio مشروعك الجديد، وستكون جاهزًا لكتابة التعليمات البرمجية!

الدعم أو استكشاف الأخطاء وإصلاحها Support or troubleshooting

في بعض الأحيان، قد تسوء الأمور. إذا فشلت عملية تثبيت Visual Studio راجع استكشاف مشكلات تثبيت Visual Studio وترقيته وإصلاحها للحصول على إرشادات خطوة بخطوة.

- خيار دعم الدردشة أثناء التثبيت (باللغة الإنجليزية فقط) للمشاكل المتعلقة بالتثبيت.
- أبلغنا عن مشكلات المنتج من خلال أداة الإبلاغ عن مشكلة التي تظهر في كل من Visual Studio Installer and the Visual Studio IDE إذا كنت مسؤولاً عن تكنولوجيا المعلومات ولم تقم بتثبيت Visual Studio فيمكنك إرسال ملاحظات مسؤول تكنولوجيا المعلومات هنا
- اقترح ميزة، وتتبع مشكلات المنتج، وابحث عن إجابات في مجتمع مطوري Visual Studio



تُظهر الصورة السابقة برنامج Visual Studio مع مشروع مفتوح يعرض النوافذ الرئيسية ووظائفها:

- في مستكشف الحلول Solution Explorer في الزاوية اليمنى العليا، يمكنك عرض ملفات التعليمات البرمجية، والتنقل فيها وإدارتها. يمكن لمستكشف الحلول المساعدة في تنظيم التعليمات البرمجية عن طريق تجميع الملفات في حلول ومشاريع.
- تعرض نافذة المحرر المركزية editor window التي من المحتمل أن تقضي فيها معظم وقتك، محتويات الملف، في نافذة المحرر، يمكنك تحرير التعليمات البرمجية أو تصميم واجهة مستخدم مثل نافذة تحتوي على أزرار ومربعات نصية.
- في Git Changes الموجودة في أسفل اليمين، يمكنك تتبع عناصر العمل ومشاركة التعليمات مع الآخرين، باستخدام تقنيات التحكم في الإصدار مثل Git and GitHub

الإصدارات **Editions**

يتوفر Visual Studio لنظام تشغيل Windows

هناك ثلاث إصدارات هي **Community, Professional, and Enterprise** راجع مقارنة Compare Visual Studio editions لمعرفة الميزات المدعومة في كل إصدار.

الميزات الإنتاجية الشائعة

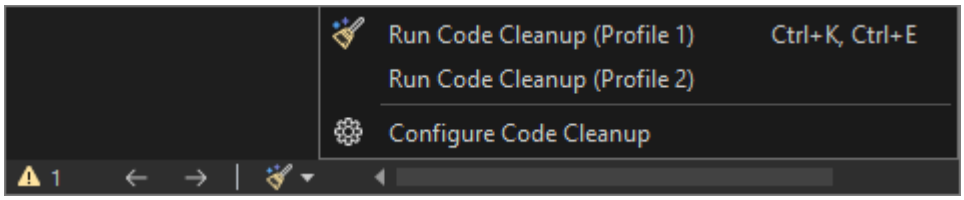
تتضمن بعض الميزات الشائعة في Visual Studio التي تعمل على تحسين إنتاجيتك عند تطوير البرامج ما يلي:

- الخطوط المتعرجة والإجراءات السريعة Squiggles and Quick Actions

الخطوط المتعرجة هي خطوط سفلية متموجة تنبهك إلى الأخطاء أو المشكلات المحتملة في تعليماتك أثناء الكتابة، تساعدك هذه الإشارات المرئية على إصلاح المشكلات على الفور، دون انتظار اكتشاف الأخطاء أثناء البناء أو وقت التشغيل، إذا مررت فوق خط متعرج، فسترى مزيدًا من المعلومات حول الخطأ، قد يظهر أيضًا أيقونة مصباح كهربائي في الهامش الأيسر يوضح الإجراءات السريعة التي يمكنك اتخاذها لإصلاح الخطأ.

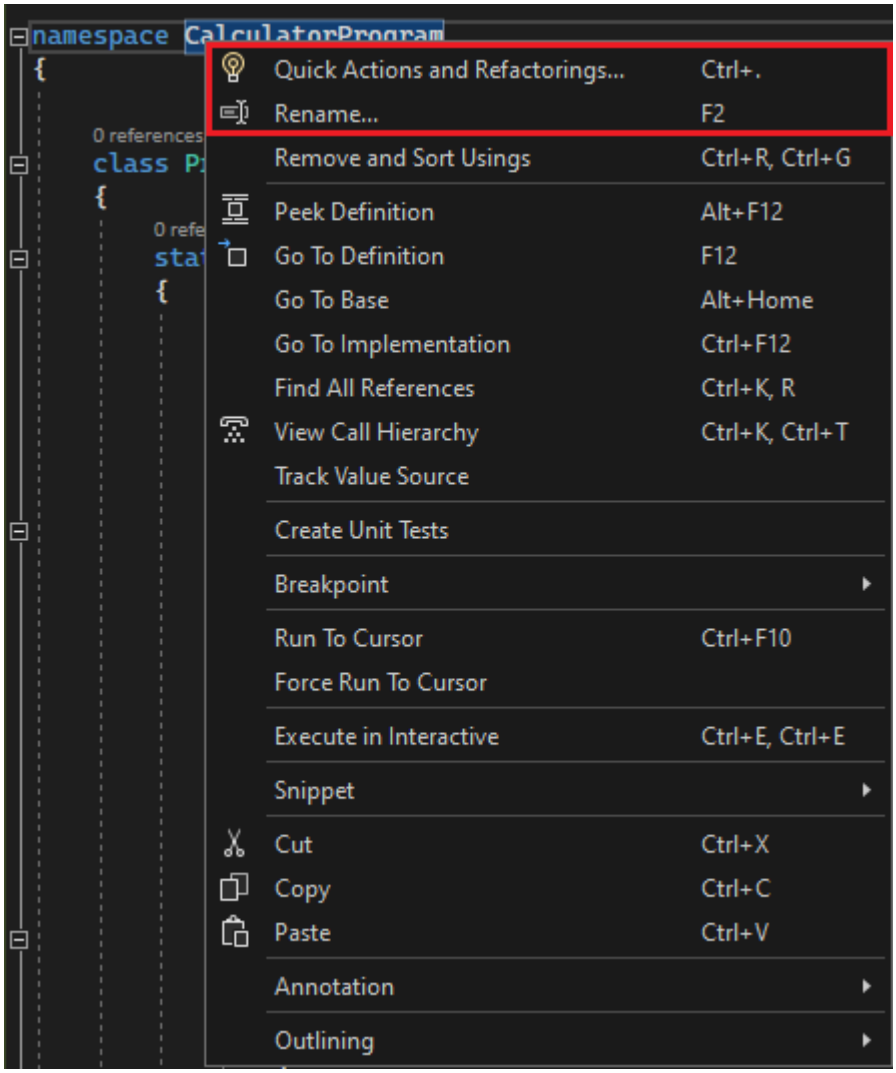
- تنظيف التعليمات Code Cleanup

بنقرة زر واحدة، يمكنك تنسيق تعليماتك وتطبيق أي إصلاحات مقترحة للتعليمات بواسطة إعدادات نمط الكود, code style settings, .editorconfig conventions and Roslyn analyzers يساعدك تنظيف التعليمات Code Cleanup المتوفر حاليًا فقط ل C# على حل المشكلات في تعليماتك، قبل إرسالها إلى المراجعة.



- إعادة الهيكلة Refactoring

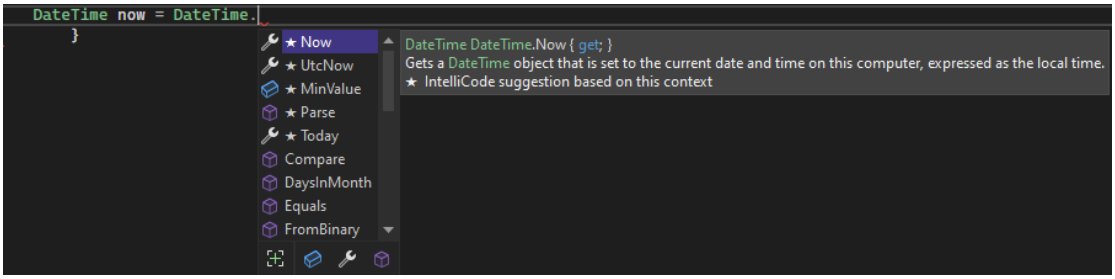
تتضمن إعادة الهيكلة عمليات مثل إعادة تسمية المتغيرات بذكاء، واستخراج سطر أو أكثر من التعليمات البرمجية في أسلوب جديد، وتغيير ترتيب معلمات الأسلوب method parameters



- التلميحات الذكية أو المتحسس الذكي IntelliSense

IntelliSense عبارة عن مجموعة من الميزات التي تعرض معلومات حول تعليماتك مباشرة في المحرر، وفي بعض الحالات، تكتب أجزاء صغيرة من التعليمات لك، إنه مثل وجود وثائق أساسية مضمنة في المحرر، لذلك لا يتعين عليك البحث عن معلومات النوع في مكان آخر.

يوضح الرسم التوضيحي التالي كيفية عرض IntelliSense لقائمة الأعضاء لنوع ما:

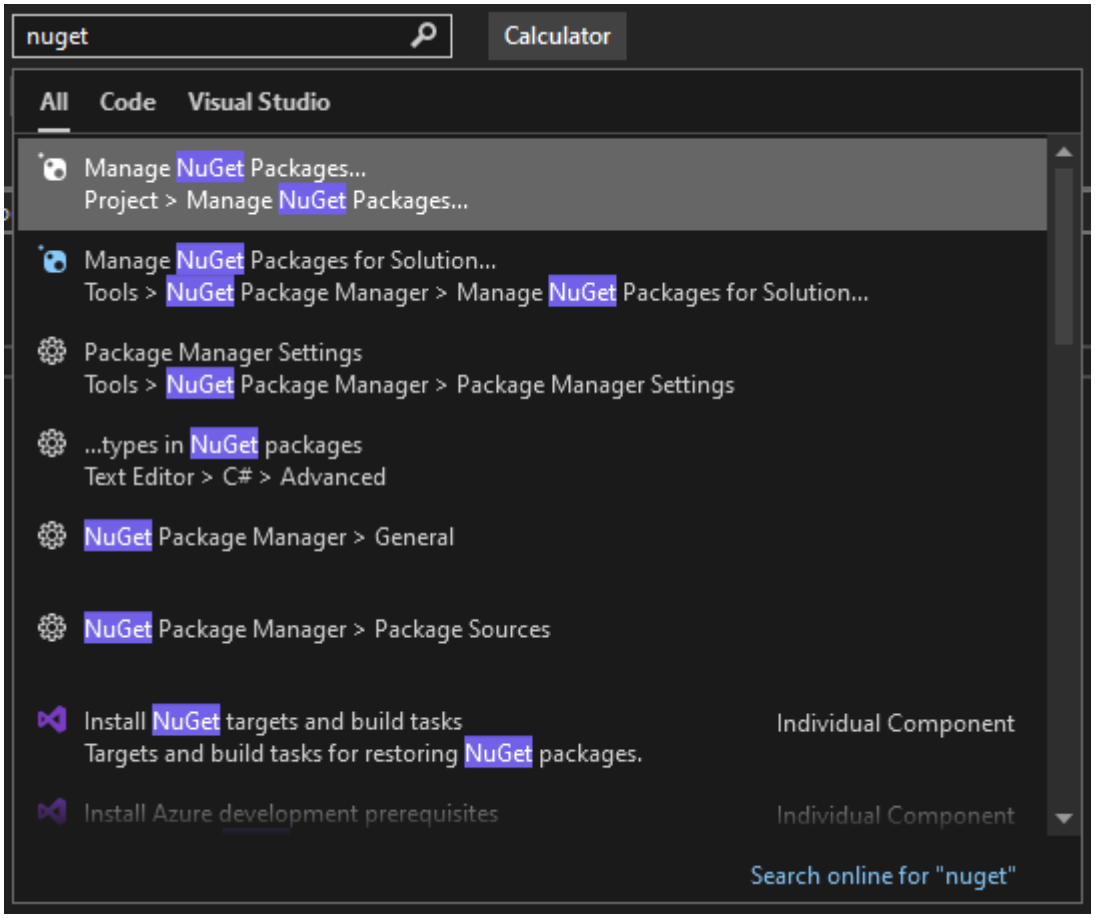


تختلف ميزات IntelliSense باختلاف اللغة، لمزيد من المعلومات، راجع:

C# IntelliSense, Visual C++ IntelliSense, JavaScript IntelliSense, and Visual Basic IntelliSense

- البحث Visual Studio search

قد تبدو قوائم وخياراتها وخصائصها Visual Studio menus, options, and properties مربكة في بعض الأحيان، يعد البحث Visual Studio search, or Ctrl+Q طريقة رائعة للعثور بسرعة على ميزات IDE والتعليمات الخاصة بها في مكان واحد.



للحصول على معلومات ونصائح حول الإنتاجية، راجع:

How to use Visual Studio search

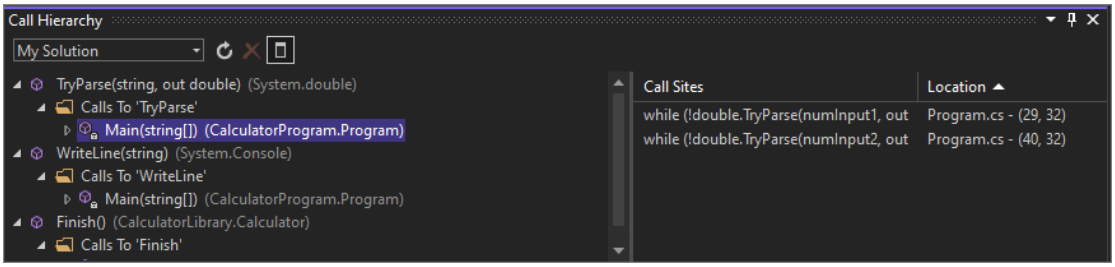
- المشاركة المباشرة Live Share

يمكنك تحرير وتصحيح الأخطاء بشكل تعاوني مع الآخرين في الوقت الفعلي، بغض النظر عن نوع التطبيق أو لغة البرمجة، يمكنك مشاركة مشروعك بشكل فوري وآمن. يمكنك أيضًا مشاركة جلسات التصحيح، ونسخ المحطة الطرفية وتطبيقات الويب

debugging sessions, terminal instances, localhost web apps والمكالمات الصوتية، والمزيد.

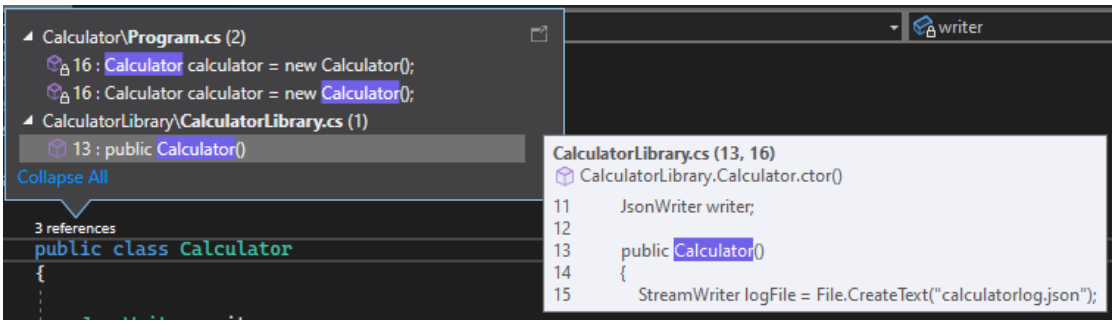
- تسلسل الاستدعاءات Call Hierarchy

تعرض نافذة التسلسل الهرمي لاستدعاءات الأساليب methods التي تستدعي أسلوب محدد، يمكن أن تكون هذه المعلومات مفيدة عندما تفكر في تغيير الأسلوب أو إزالته، أو عندما تحاول تعقب خطأ ما.



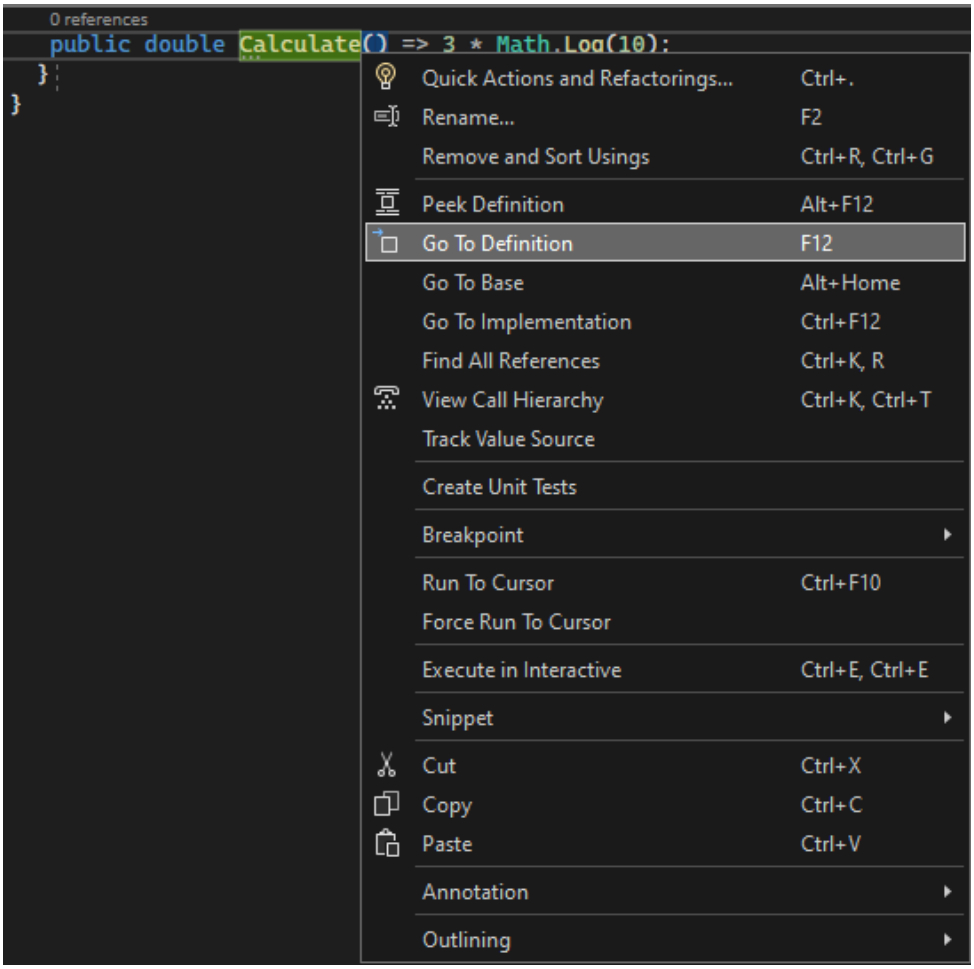
- عدسة التعليمات CodeLens

يساعدك CodeLens في العثور على مراجع التعليمات، وتغييرات التعليمات، والأخطاء المرتبطة، وعناصر العمل، ومراجعات التعليمات، واختبارات الوحدة، دون مغادرة المحرر.



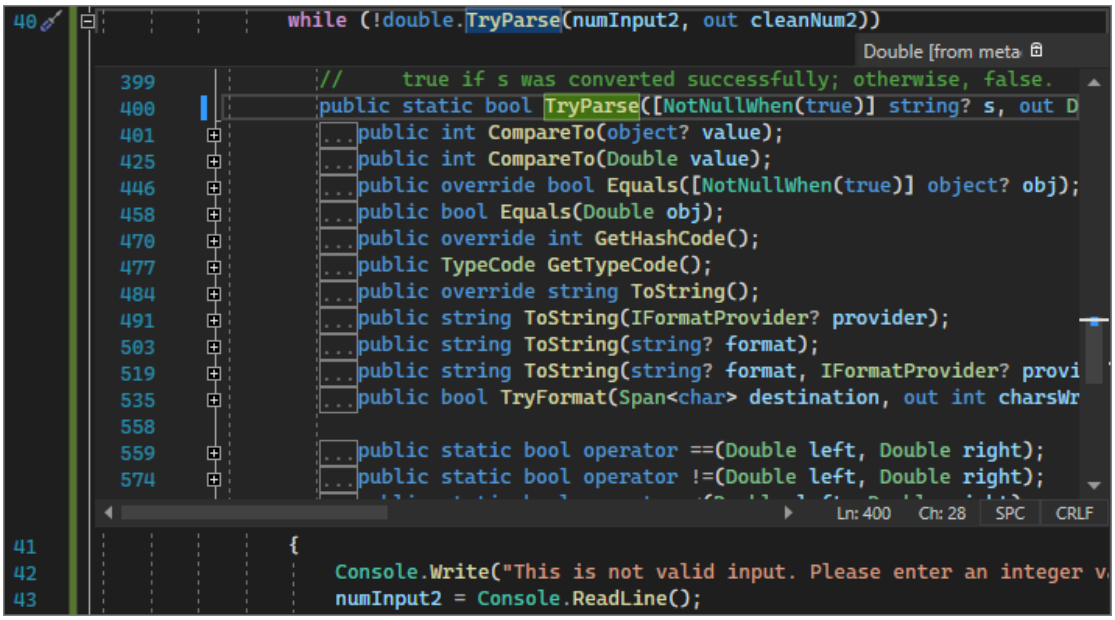
- انتقل إلى التعريف Go To Definition

تأخذك ميزة الانتقال إلى التعريف مباشرةً إلى موقع تعريف الوظيفة أو النوع function or type definition



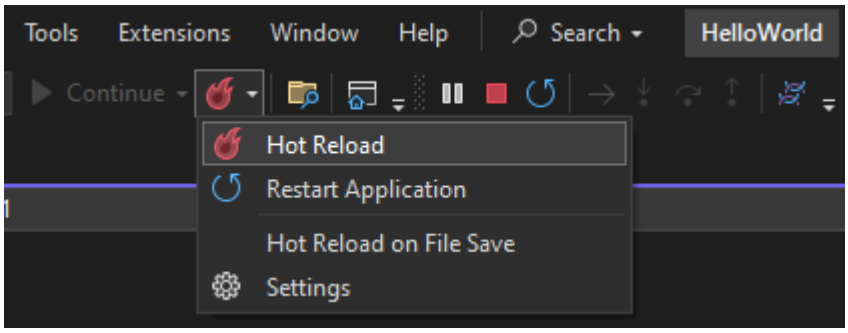
- تعريف Peek Definition

تُظهر نافذة Peek Definition تعريفًا للأسلوب أو النوع دون فتح ملف منفصل، تقريبًا كأنه طي تعليمات الأسلوب داخل قائمة منسدلة، ويمكنك إلقاء نظرة خاطفة عليها.



- إعادة التعديل الساخن Hot Reload

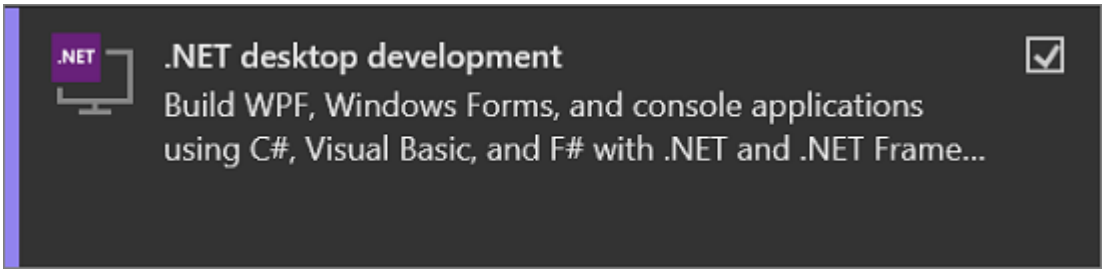
يتيح لك Hot Reload تحرير ملفات التعليمات البرمجية، وتطبيق التغييرات فورًا على التطبيق قيد التشغيل.



برنامج Visual Studio

في هذا القسم، يمكنك إنشاء مشروع بسيط لتجربة بعض الأشياء التي يمكنك القيام بها باستخدام Visual Studio يمكنك استخدام IntelliSense كمساعد في الترميز، وتصحيح أخطاء التطبيق لرؤية قيمة متغيرة أثناء تنفيذ التطبيق، وتغيير سمة اللون.

في المثبت المعياري، يمكنك اختيار أحمال العمل وتثبيتها، وهي عبارة عن مجموعات من الميزات التي تحتاجها للغات البرمجة أو المنصات التي تريدها، لاستخدام الخطوات التالية لإنشاء برنامج create a program تأكد من تحديد .NET desktop development أثناء التثبيت.



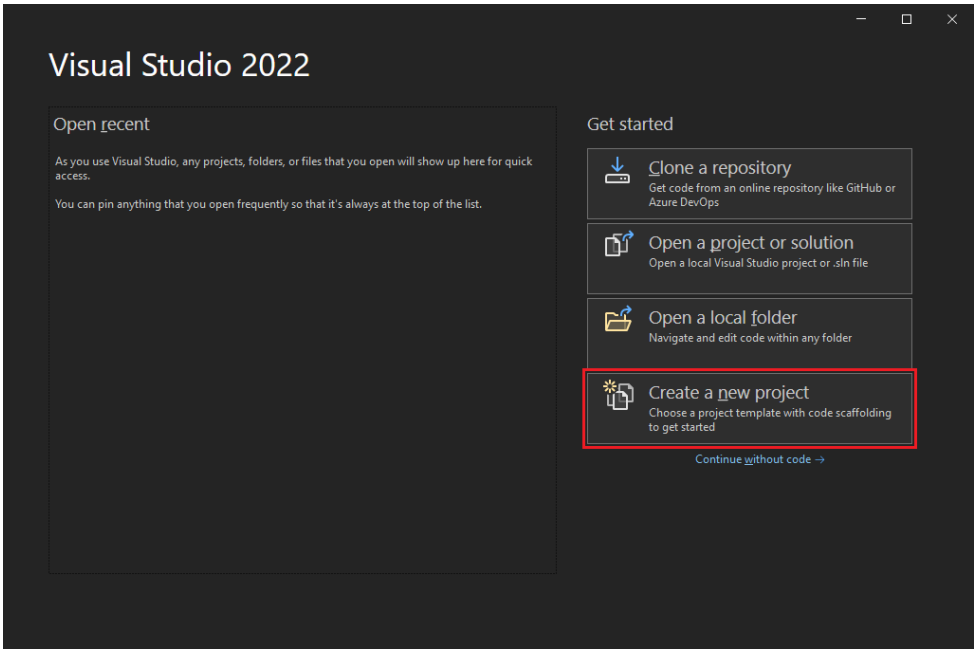
عند فتح Visual Studio لأول مرة، يمكنك تسجيل الدخول باستخدام حساب Microsoft أو حساب العمل أو المدرسة.

إنشاء برنامج Create a program

انغمس في إنشاء برنامج بسيط.

١. ابدأ تشغيل Visual Studio تظهر نافذة البدء مع خيارات لاستنساخ مستودع، أو فتح مشروع حديث، أو إنشاء مشروع جديد.

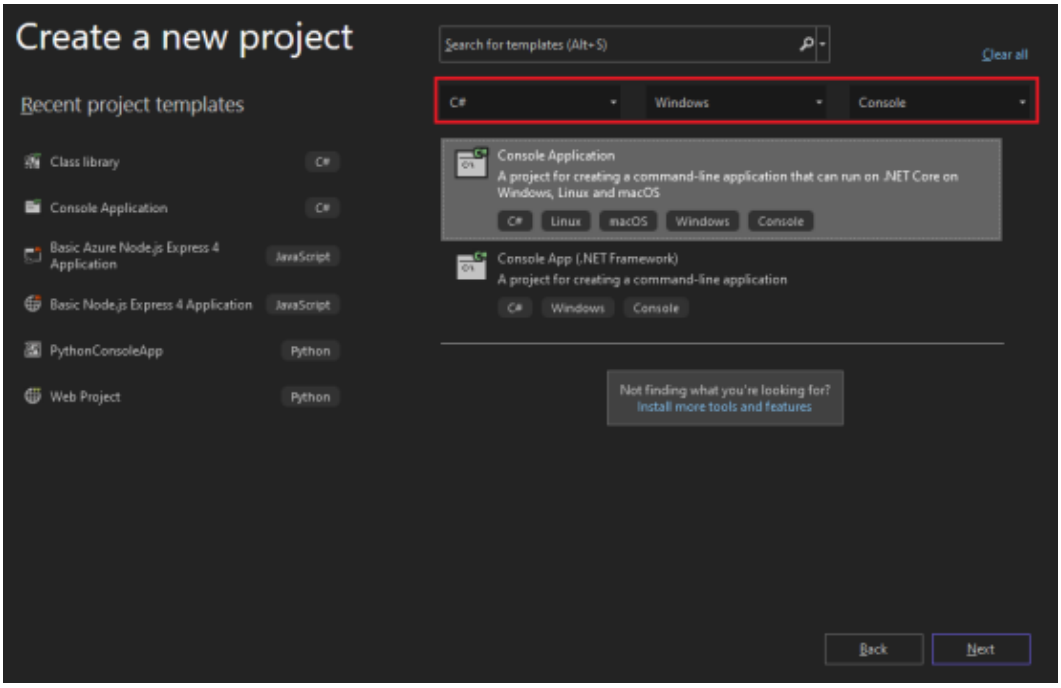
٢. اختر إنشاء مشروع جديد Create a new project



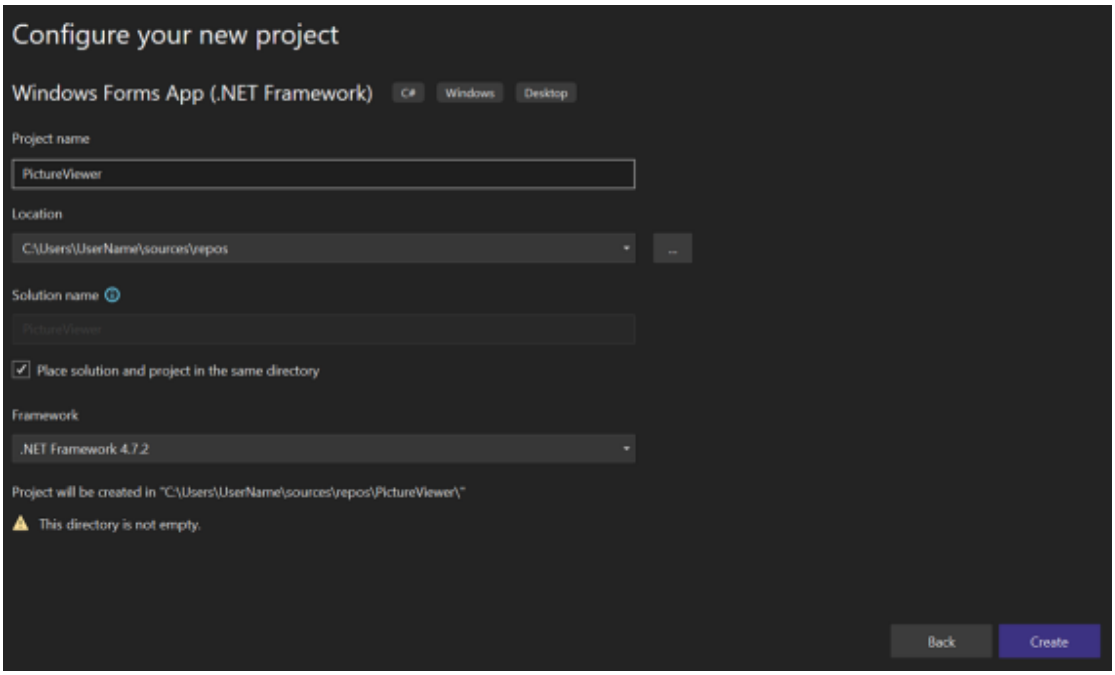
تفتح نافذة إنشاء مشروع جديد وتعرض عدة قوالب للمشروع، يحتوي القالب على الملفات الأساسية والإعدادات المطلوبة لنوع مشروع معين.

٣. للعثور على قالب، يمكنك كتابة أو إدخال كلمات رئيسية في مربع البحث، تقوم قائمة القوالب المتاحة بالتصفية بناءً على الكلمات الرئيسية التي تدخلها، يمكنك تصفية نتائج القالب بشكل أكبر عن طريق اختيار C# من القائمة المنسدلة كافة اللغات All languages ثم اختر Windows من قائمة كافة المنصات All platforms واختر وحدة التحكم Console من قائمة كافة أنواع المشاريع All project types

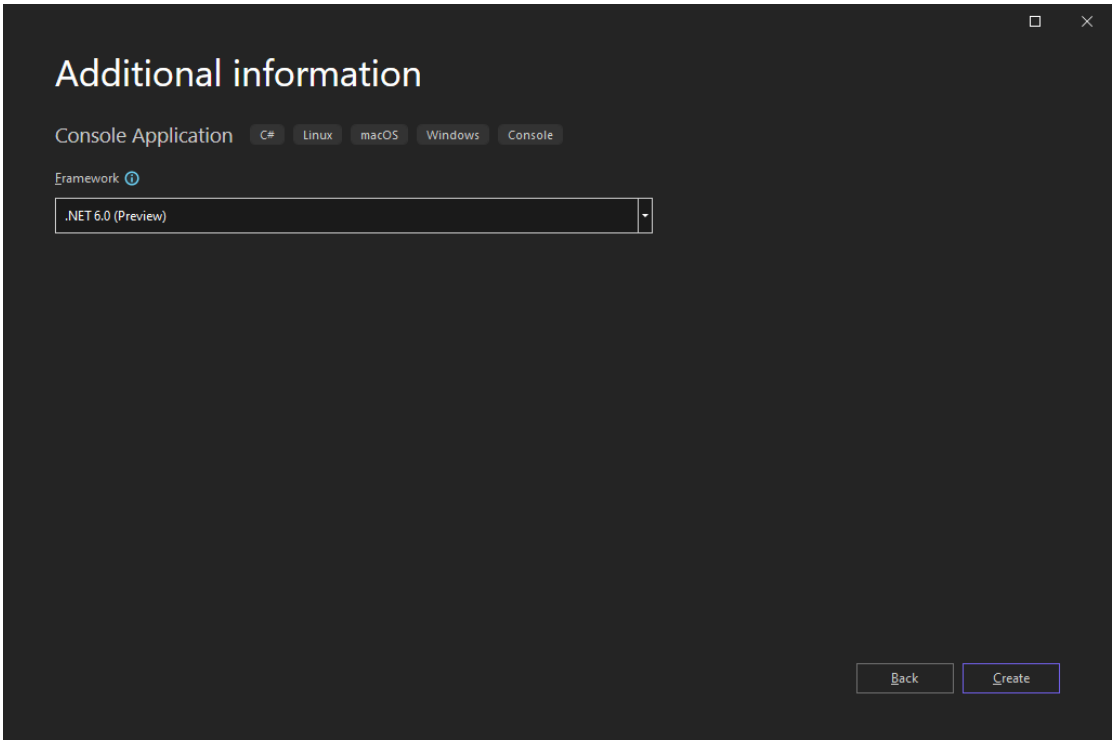
حدد قالب تطبيق وحدة التحكم Console Application ثم حدد التالي Next



٤. من نافذة تكوين مشروع جديد Configure your new project اكتب HelloWorld في مربع اسم المشروع Project name اختياريًا، يمكنك تغيير موقع مسار المشروع من الموقع الافتراضي C:\Users\<name>\source\repos وحدد التالي Next

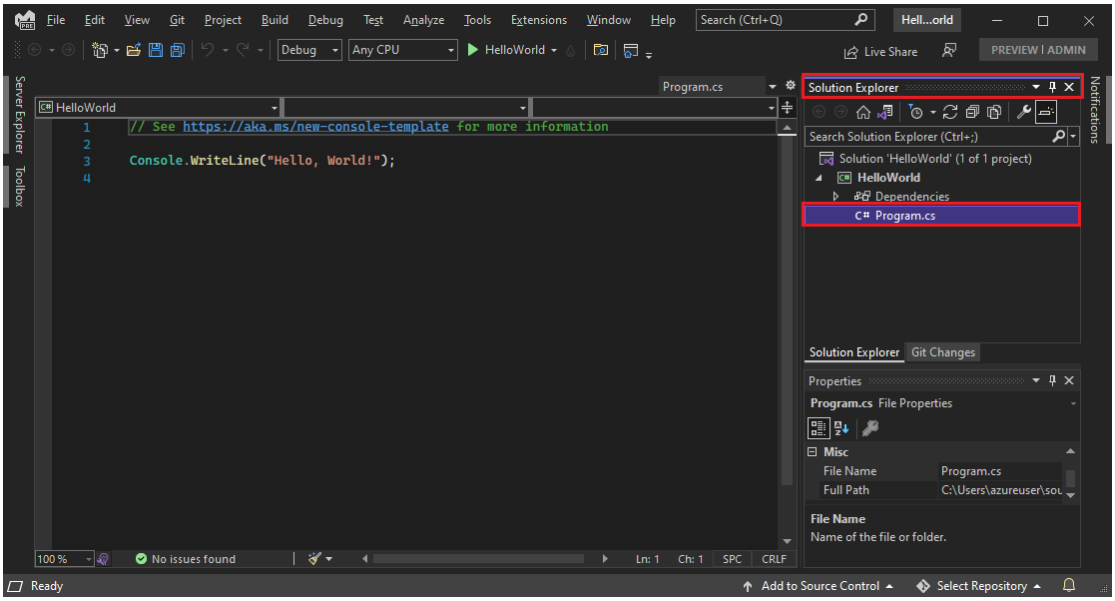


٥. من نافذة المعلومات الإضافية Additional information تأكد من ظهور .NET 8.0 في القائمة المنسدلة Target Framework ثم حدد إنشاء Create



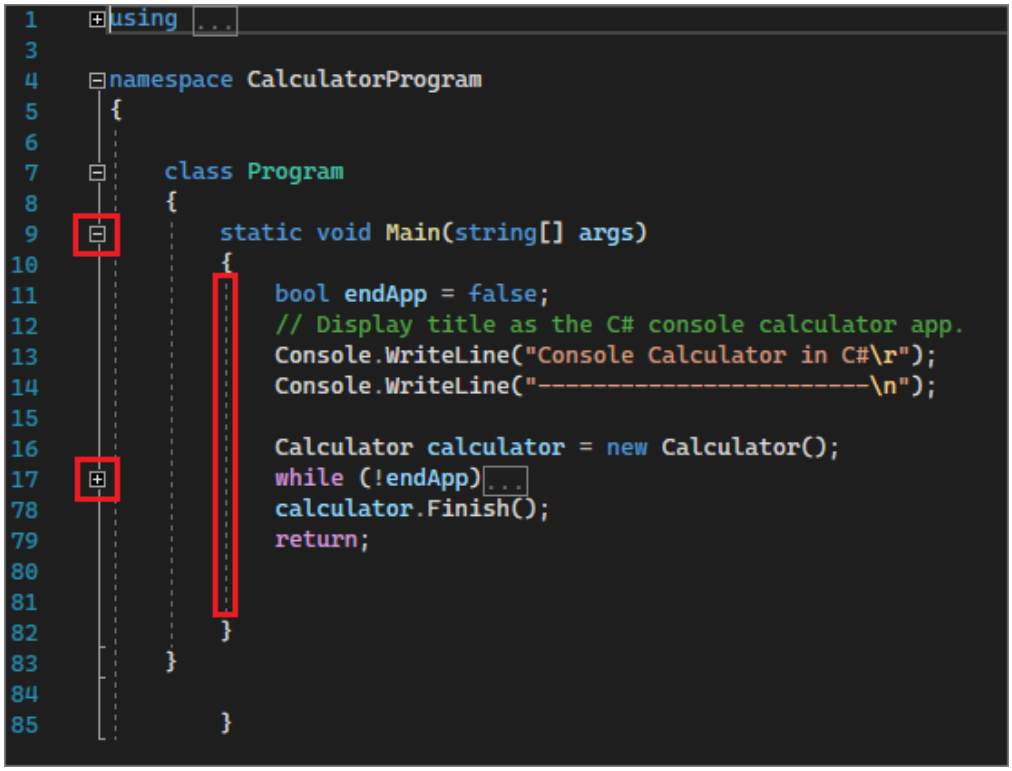
يقوم Visual Studio بإنشاء المشروع، هو عبارة عن تطبيق بسيط Hello World يستدعي أسلوب Console.WriteLine() لعرض رسالة نصية **Hello, World!** في وحدة التحكم console window

تظهر ملفات المشروع على الجانب الأيمن من Visual Studio IDE في نافذة تسمى Solution Explorer من هذه النافذة Solution Explorer اختر **Program.cs** يفتح ملف تعليمات C# البرمجية، الخاص بتطبيقك في نافذة المحرر المركزية، والتي تشغل معظم المساحة.



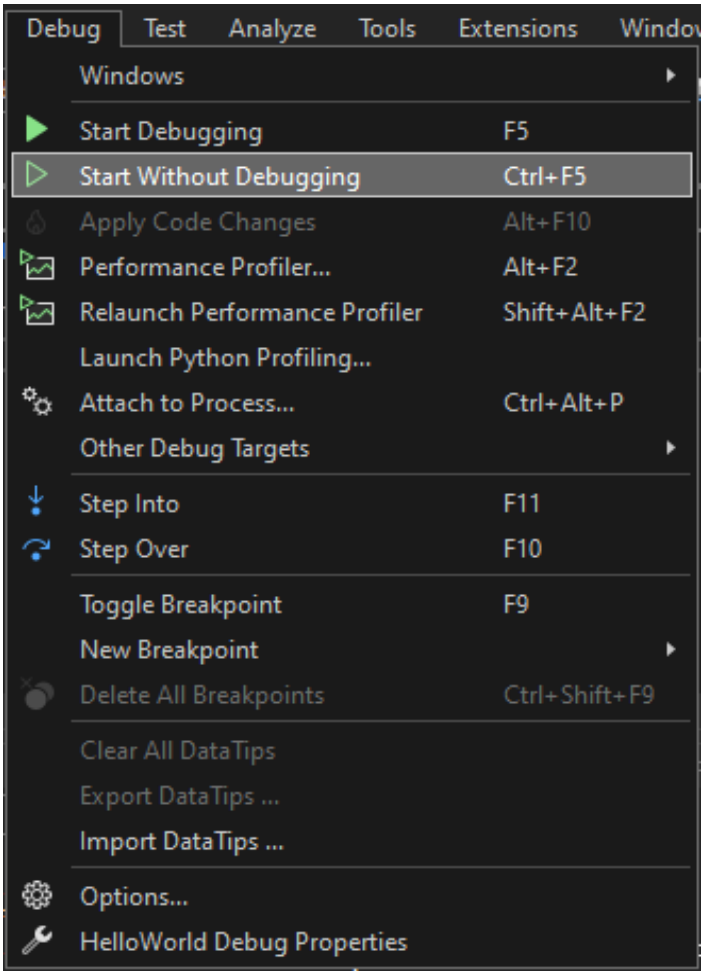
يتم تلوين التعليمات تلقائيًا للإشارة إلى أجزاء مختلفة، مثل الكلمات الأساسية keywords والأنواع types تساعدك أرقام الأسطر في تحديد موقع التعليمات.

في الصورة التالية، تشير الخطوط الصغيرة المتقطعة العمودية في التعليمات البرمجية إلى الأقواس التي تتطابق مع بعضها البعض، يمكنك أيضًا اختيار علامات ناقص أو زائد الصغيرة والمربعة، لتقليص أو توسيع كتل التعليمات البرمجية، تتيح لك ميزة تحديد الخطوط العريضة للتعليمات البرمجية إخفاء التعليمات التي لا تحتاج إلى رؤيتها، مما يساعد على تقليل الفوضى التي تظهر على الشاشة.

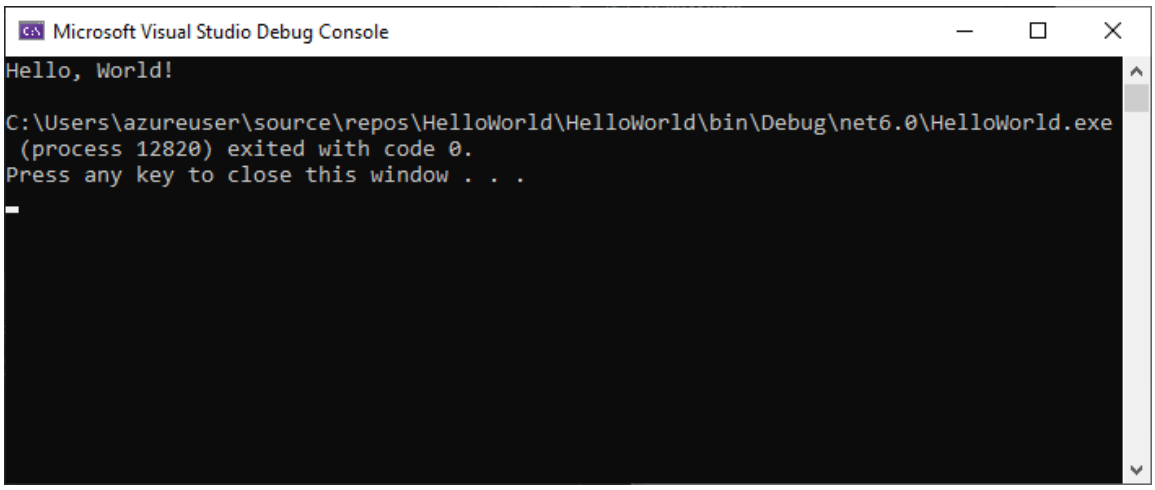


تتوفر العديد من القوائم ونوافذ الأدوات الأخرى.

٦. ابدأ تشغيل التطبيق من خلال اختيار Debug > Start Without Debugging من القائمة العلوية يمكنك أيًضا الضغط على **Ctrl+F5**



يقوم برنامج Visual Studio ببناء التطبيق، ويتم فتح نافذة وحدة التحكم مع رسالة Hello, World! لديك الآن تطبيق قيد التشغيل.



٧. لإغلاق نافذة وحدة التحكم، اضغط على أي مفتاح.

٨. دعنا نضيف بعض التعليمات البرمجية الإضافية إلى التطبيق، أضف التعليمات التالية، قبل السطر الذي يقول `Console.WriteLine("Hello World!");`

```
Console.WriteLine("\nWhat is your name?");
var name = Console.ReadLine();
```

تعرض هذه التعليمات رسالة What is your name? في نافذة وحدة التحكم، ثم ينتظر حتى يقوم المستخدم بإدخال أي نص.

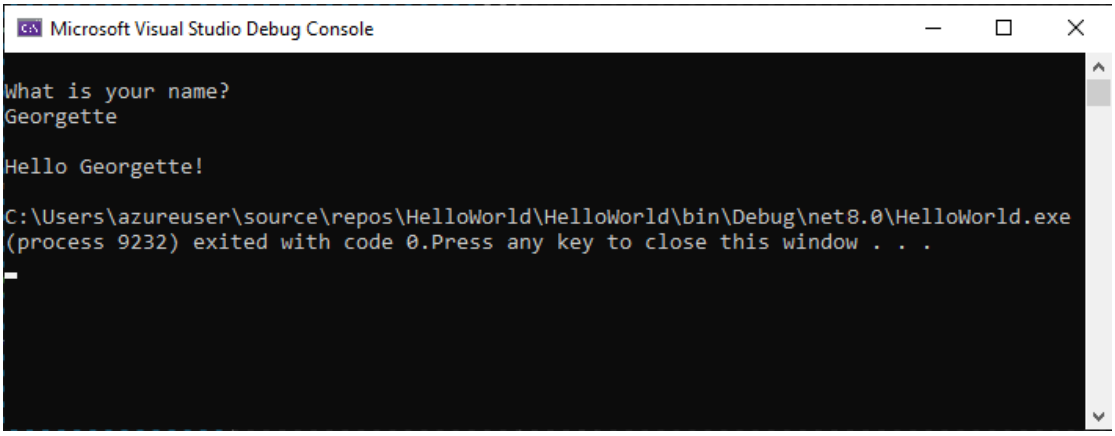
٩. قم بتغيير السطر الذي يقول `Console.WriteLine("Hello World!");` بالسطر التالي:

```
Console.WriteLine($"\nHello {name}!");
```

١٠. قم بتشغيل التطبيق مرة أخرى عن طريق تحديد Debug > Start Without Debugging أو الضغط على Ctrl+F5

يقوم Visual Studio بإعادة بناء التطبيق، ويتم فتح نافذة وحدة التحكم وتطلب منك اسمك.

١١. اكتب اسمك في نافذة وحدة التحكم واضغط على Enter



١٢. اضغط على أي مفتاح لإغلاق نافذة وحدة التحكم وإيقاف تشغيل البرنامج قيد التشغيل.

استخدام إعادة الهيكلة والتلميحات الذكية refactoring and IntelliSense

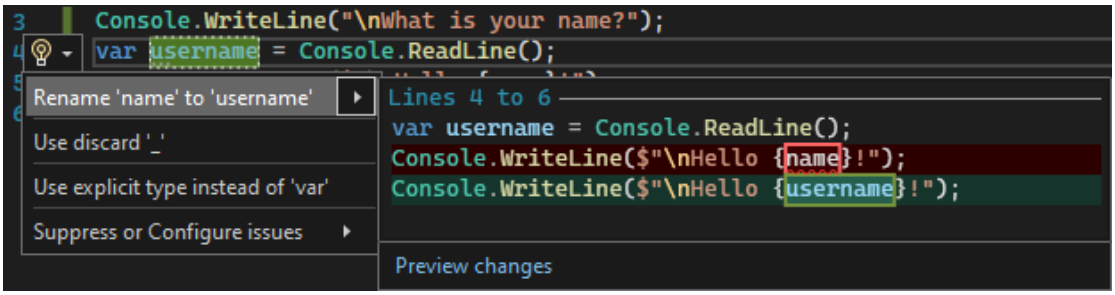
دعونا نلقي نظرة على بعض الطرق التي يمكن أن تساعدك بها، في refactoring and IntelliSense في كتابة التعليمات البرمجية بكفاءة أكبر.

أولًا، قم بإعادة تسمية المتغير `name`

١. انقر نقرًا مزدوجًا فوق المتغير `name` ثم اكتب الاسم الجديد للمتغير `username`

يظهر مربع حول المتغير، ويظهر مصباح كهربائي في الهامش.

٢. حدد رمز المصباح الكهربائي لإظهار الإجراءات السريعة المتاحة. حدد إعادة تسمية `name` إلى `username`



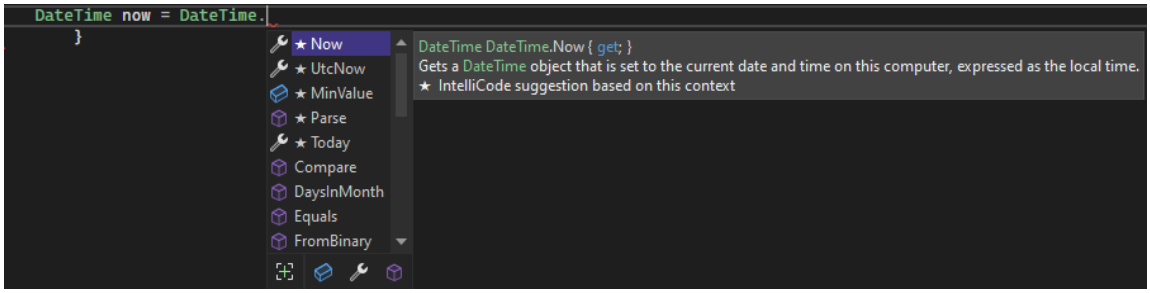
تمت إعادة تسمية المتغير عبر المشروع، والذي في حالتنا هو مكانين فقط.

٣. الآن ألق نظرة على IntelliSense أسفل السطر الذي يقول:

```
Console.WriteLine($"\nHello {username}!");
```

اكتب `DateTime now = DateTime.`

يعرض المربع عناصر فئة DateTime class كما يتم عرض وصف العنصر المحدد حاليًا في مربع منفصل.



٤. حدد العنصر المسمى **Now** والذي يعد خاصية للفئة بالنقر المزدوج عليه أو الضغط على **Tab** أكمل سطر التعليمات بإضافة فاصلة منقوطة إلى نهاية السطر:

```
DateTime now = DateTime.Now;
```

٥. أسفل هذا السطر، أدخل أسطر التعليمات البرمجية التالية:

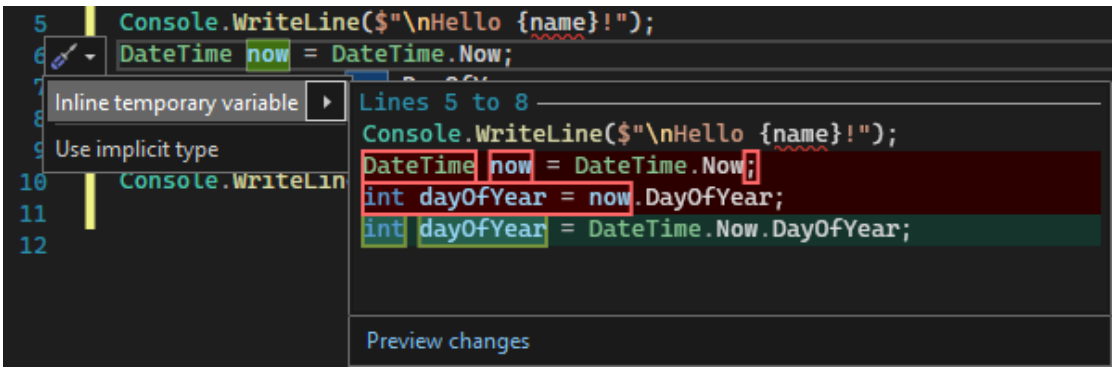
```
int dayOfYear = now.DayOfYear;
Console.Write("Day of year: ");
Console.WriteLine(dayOfYear);
```

تلميح:

تختلف Console.Write عن Console.WriteLine في أنها لا تضيف علامة إنهاء السطر بعد طباعتها، وهذا يعني أن القطعة التالية من النص، المرسلة إلى الإخراج ستتم طباعتها على نفس السطر الحالي. يمكنك التمرير فوق كل من هذه الأساليب في تعليماتك لرؤية أوصافها.

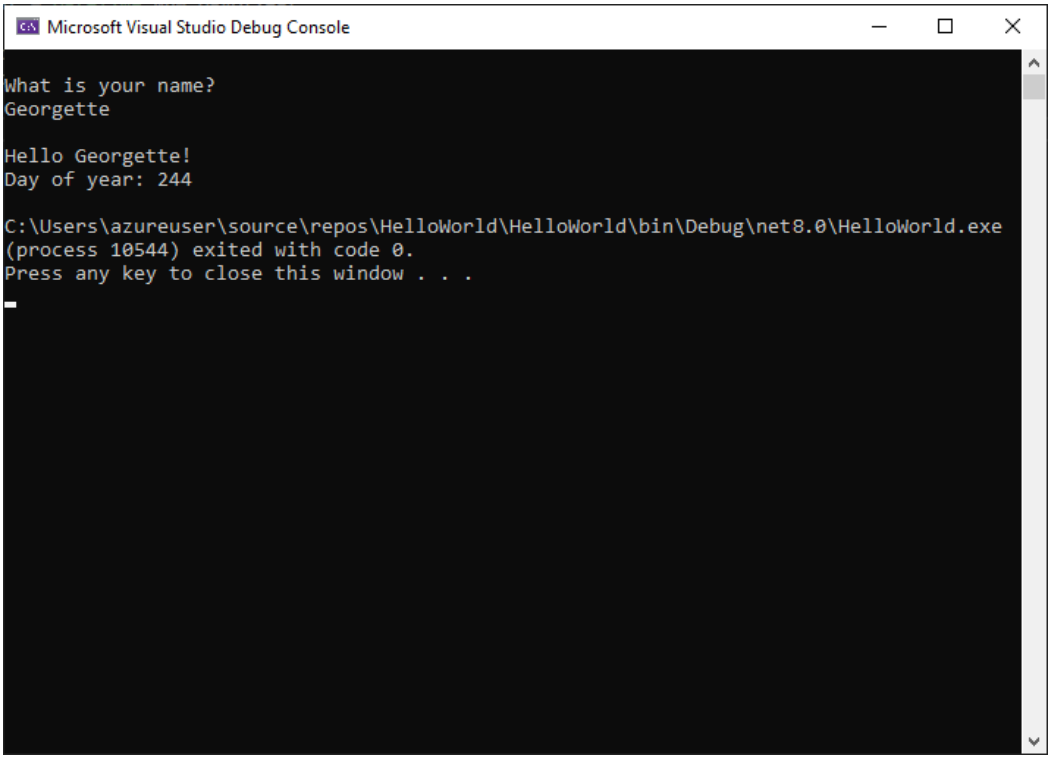
٦. استخدم إعادة الهيكلة refactoring مرة أخرى لجعل التعليمات أكثر إيجازًا، حدد المتغير `now` في السطر `DateTime now = DateTime.Now;` يظهر رمز مفك براغي في الهامش على هذا السطر.

٧. حدد أيقونة مفك البراغي لرؤية الاقتراحات المتاحة من Visual Studio توضح هذه الحالة إعادة هيكلة المتغير المؤقت المضمن Inline temporary variable لإزالة سطر من التعليمات دون تغيير سلوك التعليمات الإجمالي.



٨. حدد المتغير المؤقت المضمن Inline temporary variable لإعادة صياغة التعليمات.

٩. قم بتشغيل البرنامج مرة أخرى بالضغط على **Ctrl+F5** يبدو الناتج كالتالي:



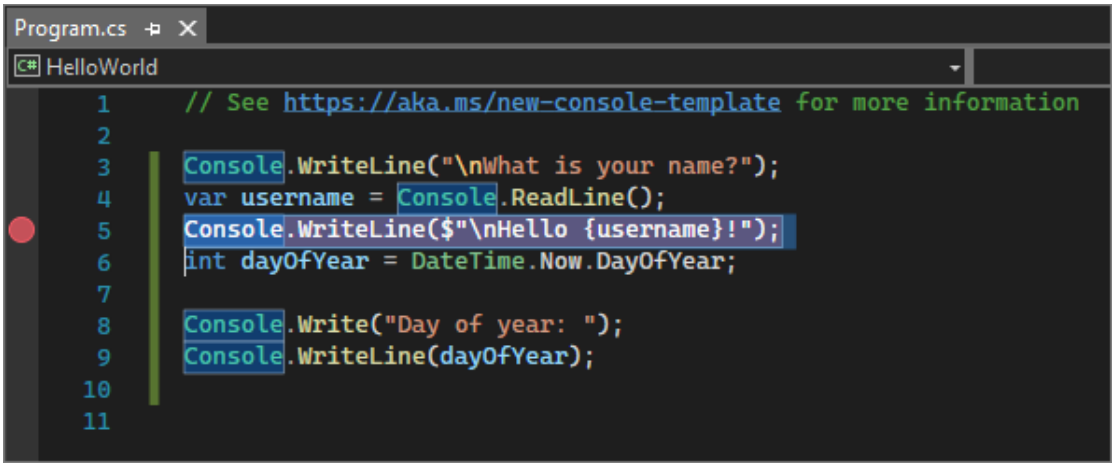
تصحيح التعليمات Debug code

عند كتابة التعليمات البرمجية، يجب عليك تشغيلها واختبارها بحثًا عن الأخطاء، يتيح لك نظام تصحيح الأخطاء في Visual Studio's debugging الانتقال عبر التعليمات البرمجية بجملة واحدة في كل مرة، وفحص المتغيرات أثناء العمل، يمكنك تعيين نقاط توقف breakpoints لإيقاف تنفيذ التعليمات البرمجية عند سطر معين، ومراقبة كيفية تغير قيمة المتغير أثناء تشغيل التعليمات البرمجية

تعيين نقطة توقف breakpoint لرؤية قيمة المتغير `username` أثناء تشغيل البرنامج.

١. قم بتعيين نقطة توقف على سطر التعليمات البرمجية التالي `Console.WriteLine($"\nHello {username}!");` بالنقر فوق الهامش الأيسر الأقصى، أو الحافة بجوار السطر. يمكنك أيضًا تحديد سطر التعليمات البرمجية ثم الضغط على F9

تظهر دائرة حمراء في الهامش، ويتم تسليط الضوء على الخط.

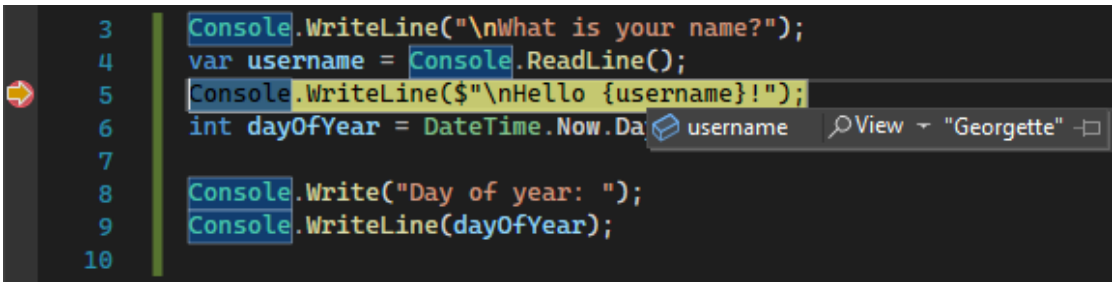


٢. ابدأ عملية تصحيح الأخطاء، عن طريق تحديد تصحيح الأخطاء واختيار بدء تصحيح الأخطاء Debug > Start Debugging أو الضغط على F5

٣. عندما تظهر نافذة وحدة التحكم وتطلب اسمك، أدخل اسمك.

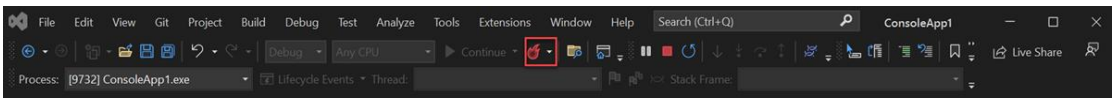
يعود التركيز إلى محرر التعليمات البرمجية في Visual Studio ويتم تمييز سطر التعليمات الذي يحتوي على نقطة التوقف باللون الأصفر، ويعني التمييز باللون الأصفر أن سطر التعليمات هذا سيتم تنفيذه بعد ذلك، وتجعل نقطة التوقف التطبيق يوقف التنفيذ مؤقتًا عند هذا السطر.

٤. مرر الماوس فوق المتغير `username` لمعرفة قيمته، يمكنك أيضًا النقر بزر الماوس الأيمن فوق `username` وتحديد Add Watch "إضافة مراقبة" لإضافة المتغير إلى نافذة المراقبة Watch window حيث يمكنك أيضًا رؤية قيمته.



٥. اضغط على F5 مرة أخرى لإكمال تشغيل التطبيق.

بمجرد تشغيل التطبيق، يمكنك تطبيق تعديل التعليمات على التطبيق قيد التشغيل من خلال النقر فوق زر إعادة التعديل أو التحميل السريع Hot Reload



لمزيد من المعلومات حول تصحيح الأخطاء في Visual Studio راجع ميزات المصحح Debugger feature tour

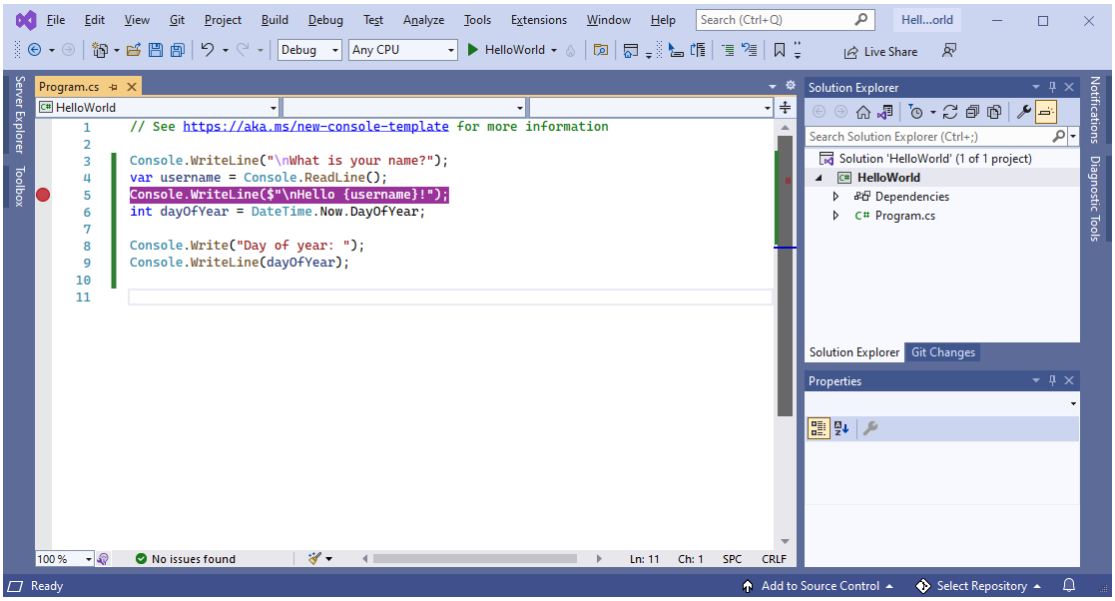
تخصيص Customize Visual Studio

يمكنك تخصيص واجهة مستخدم Visual Studio بما في ذلك تغيير سمة اللون الافتراضية. لتغيير سمة اللون:

١. في شريط القائمة، اختر أدوات ثم خيارات Tools > Options لفتح مربع الخيارات Options dialog

٢. في صفحة البيئة ثم الخيارات العامة Environment > General options قم بتغيير اختيار سمة اللون Color Theme إلى أزرق أو فاتح Blue or Light ثم اضغط موافق Ok

يتغير لون سمة بيئة التطوير المتكاملة بالكامل وفقًا لذلك تُظهر لقطة الشاشة التالية السمة الزرقاء.



لمعرفة المزيد عن طرق أخرى يمكنك من خلالها تخصيص IDE راجع تخصيص Personalize Visual Studio

تحديد إعدادات البيئة environment settings

يمكنك تكوين Visual Studio لاستخدام إعدادات البيئة المخصصة لمطوري C#

١. من شريط القائمة اختر أدوات ثم إعدادات الاستيراد والتصدير Tools > Import and Export Settings

٢. في معالج إعدادات الاستيراد والتصدير Import and Export Settings Wizard حدد إعادة تعيين جميع الإعدادات Reset all settings ثم حدد التالي Next

٣. في صفحة حفظ الإعدادات الحالية Save Current Settings اختر ما إذا كنت تريد حفظ إعداداتك الحالية قبل إعادة التعيين، إذا لم تقم بتخصيص أي إعدادات، فحدد لا، فقط إعادة تعيين الإعدادات، واستبدال إعداداتي الحالية No, just reset settings, overwriting my current settings ثم حدد التالي Next

٤. في صفحة اختيار مجموعة افتراضية من الإعدادات Choose a Default Collection of Settings اختر Visual C# ثم حدد إنهاء Finish

٥. في صفحة إعادة التعيين المكتملة Reset Complete اختر إغلاق Close

كيفية استخدام محرر التعليمات code editor مع C#

في هذ القسم سنضيف التعليمات البرمجية إلى ملف، للنظر في بعض الطرق في Visual Studio التي تجعل كتابة التعليمات والتنقل خلالها وفهمها أسهل.

إنشاء ملف كود جديد Create a new code file

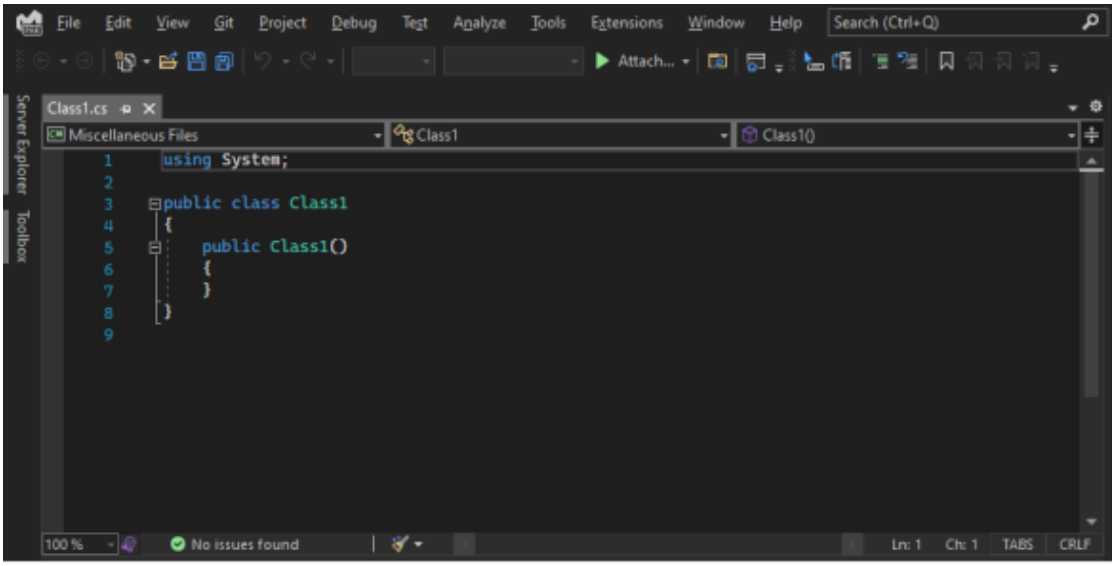
ابدأ بإنشاء ملف جديد وإضافة بعض التعليمات البرمجية إليه

١. افتح Visual Studio اضغط على Esc أو اختر Continue without code في نافذة البدء لفتح بيئة التطوير.

٢. من قائمة الملف File في شريط القائمة، اختر جديد ثم ملف New > File أو اضغط على Ctrl+N

٣. في مربع الحوار ملف جديد New File ضمن الفئة العامة General اختر Visual C# Class ثم اختر فتح Open

يتم فتح ملف جديد في المحرر بهيكل فئة C# لا يتعين عليك إنشاء مشروع Visual Studio كامل للاستفادة من بعض المزايا التي يوفرها محرر التعليمات البرمجية - كل ما تحتاجه هو ملف تعليمات برمجية.



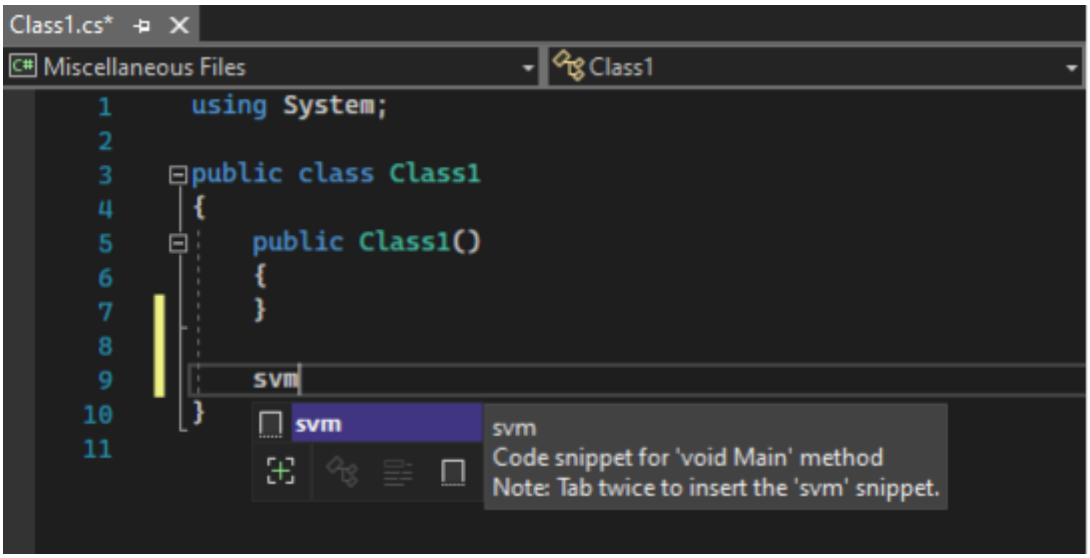
استخدم مقتطفات التعليمات code snippets

يوفر Visual Studio مقتطفات تعليمات مفيدة code snippets يمكنك استخدامها لإنشاء كتل التعليمات شائعة الاستخدام، بسرعة وسهولة، تتوفر مقتطفات تعليمات للغات برمجة مختلفة، بما في ذلك C#, Visual Basic, C++

دعونا نضيف مقتطف `void Main` إلى ملفنا.

١. ضع المؤشر قبل قوس الإغلاق الأخير } في الملف، واكتب الأحرف svm التي يرمز إليها `static void Main` لا تقلق إذا كنت لا تعرف ماذا يعني ذلك الآن.

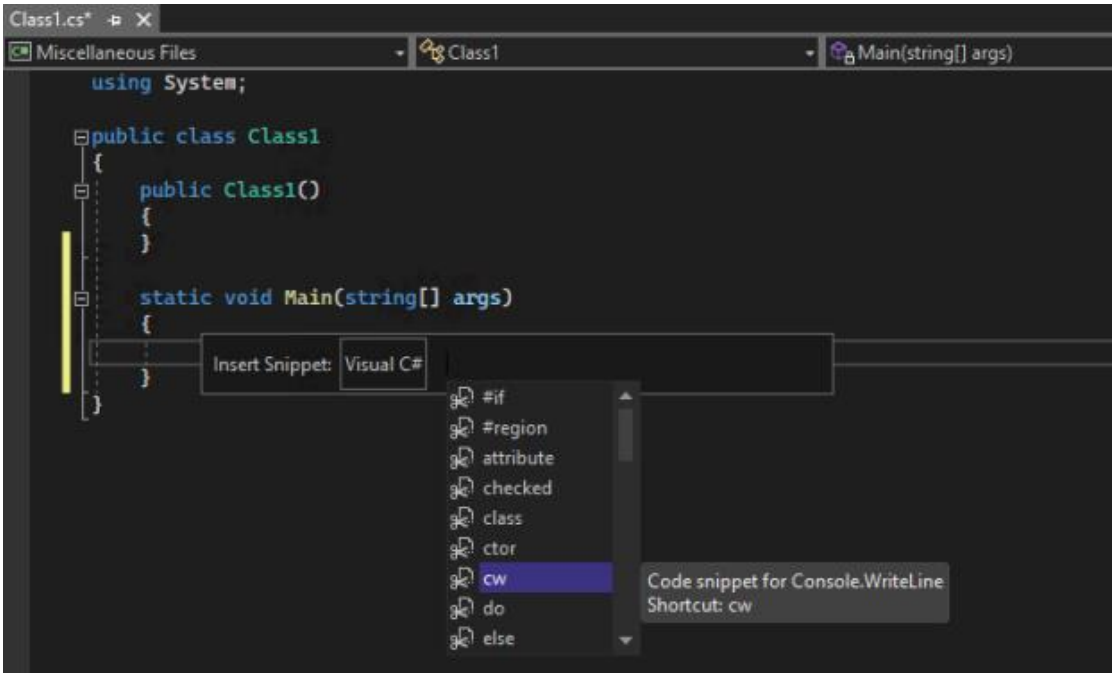
يظهر مربع حوار منبثق يحتوي على معلومات حول مقتطف التعليمات svm



٢. اضغط على Tab مرتين لإدراج مقتطف التعليمات البرمجية.

سوف تري static void Main() توقيع الأسلوب سيضاف إلى الملف، أسلوب Main() هي نقطة الدخول لتطبيقات C#

تختلف مقتطفات التعليمات البرمجية المتاحة باختلاف لغات البرمجة، يمكنك الاطلاع على مقتطفات التعليمات المتاحة للغتك من خلال اختيار تحرير ثم المتحسس الذكي ثم إدراج مقتطف Edit > IntelliSense > Insert Snippet أو اضغط على Ctrl+K, Ctrl+X ثم اختيار المجلد الخاص بلغة البرمجة الخاصة بك، بالنسبة إلى C# تبدو قائمة المقتطفات على النحو التالي:



تتضمن القائمة مقتطفات لإنشاء a class, a constructor, a for loop, an if or switch statement والمزيد.

تعليقات التعليمات Comment out code

يساعدك شريط الأدوات، وهو عبارة عن صف من الأزرار أسفل شريط القائمة في Visual Studio على زيادة إنتاجيتك أثناء كتابة التعليمات البرمجية. على سبيل المثال، يمكنك تبديل وضع إكمال IntelliSense أو زيادة أو تقليل المسافة البادئة للأسطر، أو التعليق على التعليمات البرمجية التي لا تريد تجميعها.

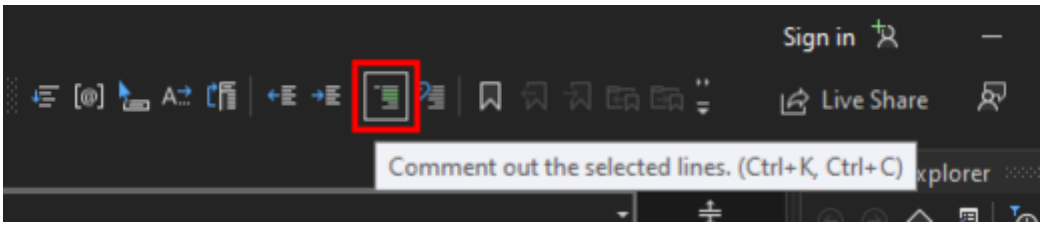
دعونا نعلق على بعض التعليمات البرمجية.

١. قم بلصق التعليمات التالية في نص الأسلوب Main()

```
// someWords is a string array.
string[] someWords = {
    "the",
    "quick",
    "brown",
    "fox",
    "jumps"
    };
```

```
string[] moreWords = {
    "over",
    "the",
    "lazy",
    "dog"
};
// Alphabetically sort the words.
IEnumerable<string> query = from word in someWords
                            orderby word
                            select word;
```

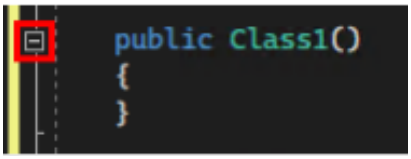
٢. نحن لا نستخدم المتغير moreWords ولكننا قد نستخدمه لاحقًا، لذا لا نريد حذفه، بدلًا من ذلك سنعلق على تلك الأسطر، حدد التعريف الكامل لـ moreWords حتى الفاصلة المنقوطة الختامية، ثم اختر زر التعليق على الأسطر المحددة Comment out the selected lines من شريط الأدوات، إذا كنت تفضل استخدام لوحة المفاتيح، فاضغط على Ctrl+E, Ctrl+C



يتم إضافة أحرف التعليق في C# // إلى بداية كل سطر محدد، للتعليق على التعليمات.

طي كتل التعليمات Collapse code blocks

لا نريد أن نرى المنشئ constructor الفارغ الذي تم إنشاؤه لـ Class1 لذا لإزالة الفوضى في عرض التعليمات، دعنا نطويه. اختر المربع الرمادي الصغير الذي يحتوي على علامة الطرح - بداخله في هامش السطر الأول من المنشئ، أو إذا كنت تفضل استخدام لوحة المفاتيح، ضع المؤشر في أي مكان في تعليمات المنشئ واضغط على Ctrl+M, Ctrl+M



يتم ضغط كتلة التعليمات إلى السطر الأول فقط، متبوعًا بعلامة (...) لتوسيع كتلة التعليمات مرة أخرى، حدد نفس المربع الرمادي الذي يحتوي الآن على علامة زائد + أو اضغط على Ctrl+M, Ctrl+M مرة أخرى، تسمى هذه الميزة Outlining وهي مفيدة بشكل خاص عند ضغط الأساليب الطويلة أو الفئات بأكملها.

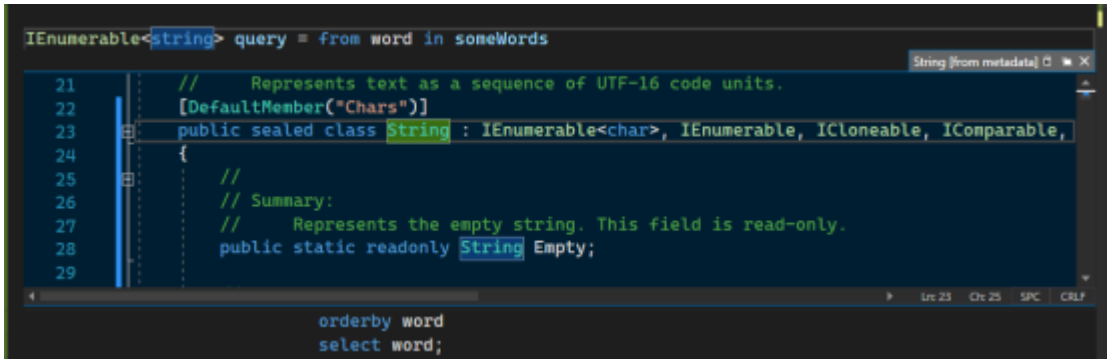
عرض تعريفات الرموز View symbol definitions

يجعل محرر Visual Studio فحص تعريف النوع أو الاسلوب أو المتغير أمرًا سهلًا إحدى الطرق هي الانتقال إلى definition of a type, method, or variable التعريف، في أي ملف يحتوي عليه، عن طريق اختيار الانتقال إلى التعريف Go to Definition أو الضغط على F12 في أي مكان يتم فيه الإشارة إلى التعليمات، هناك طريقة أسرع لا تحول تركيزك بعيدًا عن التعليمات التي تعمل عليها وهي استخدام Peek Definition

دعونا نلقي نظرة على تعريف النوع string type

١. انقر بزر الماوس الأيمن على أي ظهور string واختر Peek Definition من قائمة المحتوى أو اضغط على Alt+F12

تظهر نافذة منبثقة تحتوي على تعريف الفئة String يمكنك التمرير داخل النافذة المنبثقة، أو حتى إلقاء نظرة على تعريف نوع آخر من التعليمات التي تم الاطلاع عليها.



٢. قم بإغلاق نافذة تعريف النظرة الخاطفة peek definition عن طريق اختيار المربع الصغير الذي يحتوي على علامة **X** في الجزء العلوي الأيمن من النافذة المنبثقة.

استخدم IntelliSense لإكمال الكلمات

يعد IntelliSense موردًا لا يقدر بثمن عند كتابة التعليمات البرمجية، فيمكنه أن يعرض لك معلومات حول العناصر المتاحة لنوع ما، أو تفاصيل المعلمات لعمليات التحميل الزائد المختلفة لأسلوب ما parameter details for different overloads of a method

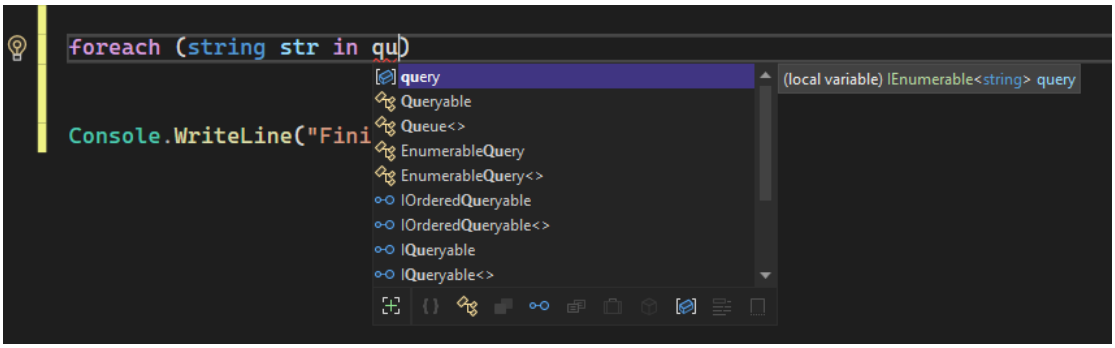
كما يمكنك استخدام IntelliSense لإكمال كلمة بعد كتابة عدد كافٍ من الأحرف لإزالة الغموض عنها.

دعنا نضيف سطرًا من التعليمات لطباعة السلاسل النصية المرتبة the ordered strings إلى نافذة وحدة التحكم، وهو المكان القياسي الذي سيتم وضع مخرجات البرنامج فيه.

١. أسفل المتغير `query` ابدأ بكتابة التعليمات التالية:

```
foreach (string str in qu
```

ستظهر لك نافذة منبثقة من IntelliSense تحتوي على معلومات حول الرمز `query`



٢. لإدراج بقية الكلمة `query` باستخدام إكمال IntelliSense للكلمات او اضغط على Tab

٣. قم بإنهاء كتلة التعليمات بحيث تبدو مثل التعليمات التالية. يمكنك التدرب بشكل أكبر على مقتطفات التعليمات البرمجية عن طريق إدخال `cw` ثم الضغط مرتين على Tab لتوليد عبارة `Console.WriteLine`

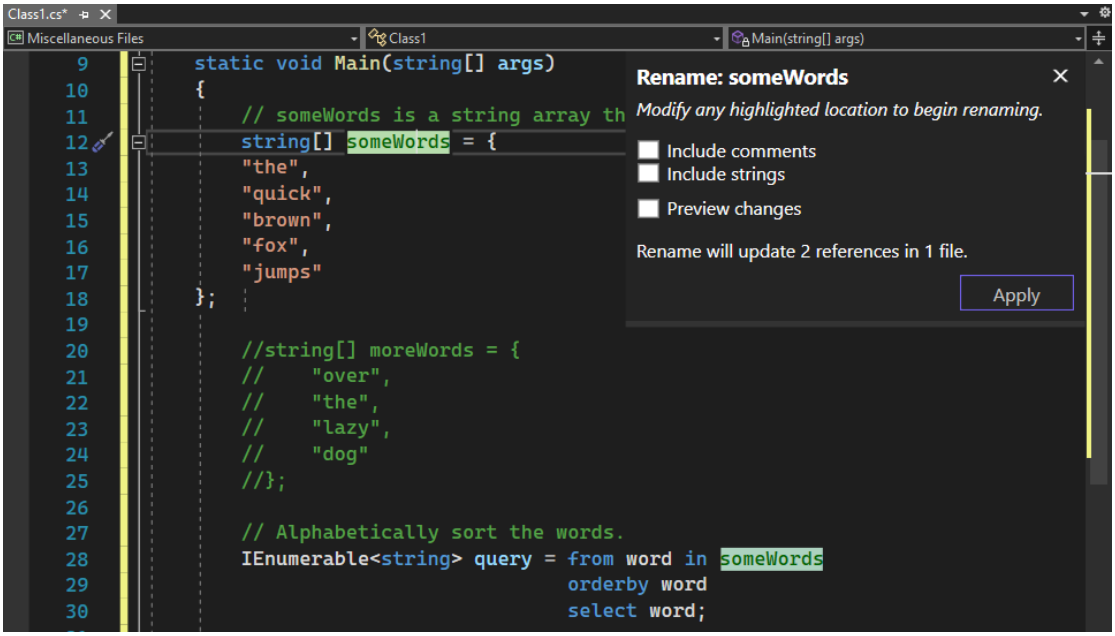
```
foreach (string str in query)
{
    Console.WriteLine(str);
}
```

إعادة الهيكلة أو صياغة الاسم Refactor a name

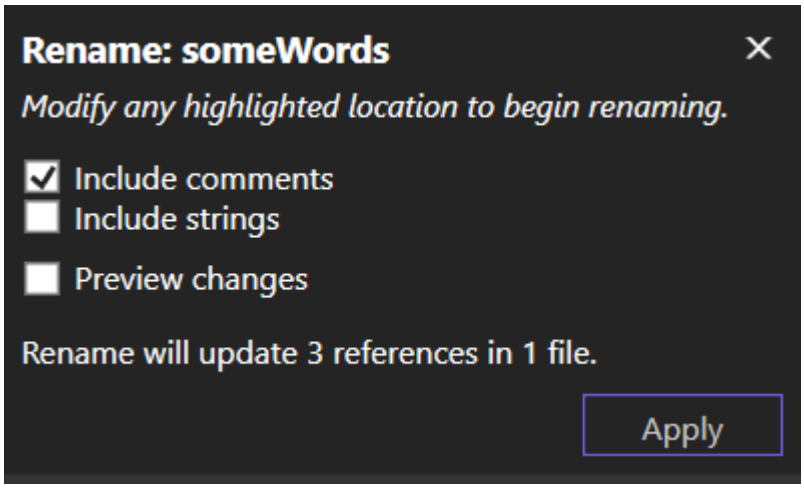
لا أحد يكتب التعليمات بشكل صحيح من المرة الأولى، وأحد الأشياء التي قد تحتاج إلى تغييرها هو اسم المتغير أو الأسلوب، دعنا نجرب وظيفة إعادة الهيكلة Refactor لإعادة تسمية المتغير `someWords` إلى `unsortedWords`

١. ضع المؤشر فوق تعريف المتغير `someWords` واختر إعادة التسمية Rename من النقر بزر الماوس الأيمن أو قائمة السياق، أو اضغط على F2

يظهر مربع حوار إعادة التسمية Rename في الجزء العلوي الأيمن من المحرر.



٢. أدخل الاسم المطلوب unsortedWords سترى أن المرجع إلى `unsortedWords` في بيان التعيين `query` يتم إعادة تسميته تلقائيًا أيضًا، قبل الضغط على Enter حدد مربع الاختيار تضمين التعليقات Include comments في المربع المنبثق إعادة التسمية Rename



اضغط على Enter أو اختر Apply في مربع الحوار إعادة التسمية Rename

لقد تمت إعادة تسمية كلا ظهوري `someWords` في تعليماتك، بالإضافة إلى النص `someWords` الموجود في تعليق تعليماتك.

٢ إنشاء أول تطبيق وحدة تحكم

في هذه الوحدة، يمكنك استخدام Visual Studio لإنشاء وتشغيل تطبيق وحدة تحكم C# واستكشاف بعض ميزات بيئة التطوير المتكاملة (IDE) لبرنامج Visual Studio

ملحوظة:

سوف يتم شرح كيفية إنشاء برامج Windows Forms بالتفصيل في الوحدات القادمة.

في هذا القسم الأول، يمكنك إكمال المهام التالية:

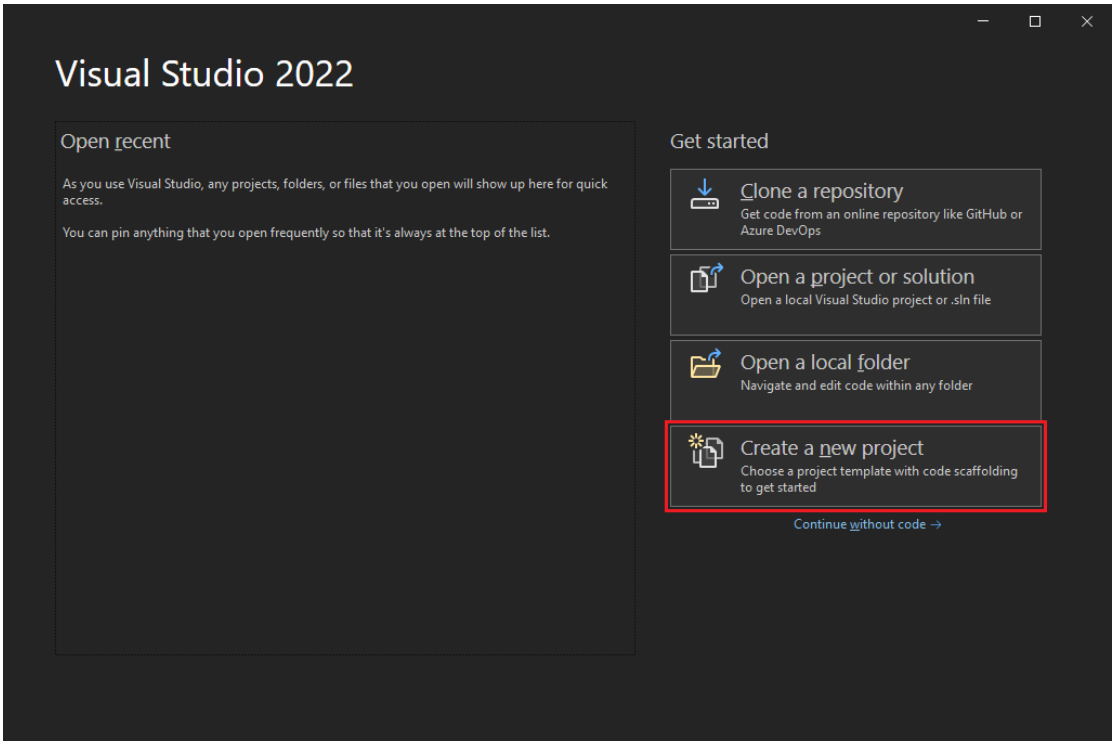
- إنشاء مشروع Visual Studio
- إنشاء تطبيق وحدة التحكم C# console app
- تصحيح أخطاء تطبيقك Debug your app
- أغلق تطبيقك Close your app
- افحص تعليماتك بالكامل Inspect your complete code

في القسم الثاني يمكنك توسيع هذا التطبيق لإضافة المزيد من المشاريع، وتعلم حيل تصحيح الأخطاء debugging tricks والرجوع إلى حزم الطرف الثالث reference third-party packages

إنشاء مشروع

للبدء، قم بإنشاء مشروع تطبيق C# يأتي نوع المشروع مع جميع ملفات القالب التي تحتاجها.

١. افتح Visual Studio ثم حدد إنشاء مشروع جديد Create a new project في نافذة البدء.

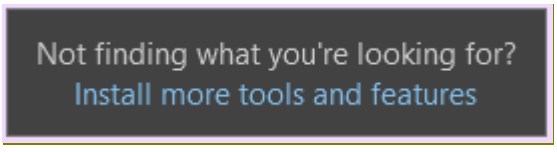


٢. في نافذة إنشاء مشروع جديد Create a new project حدد كل اللغات All languages ثم اختر C# من القائمة المنسدلة. اختر Windows من قائمة كل الأنظمة الأساسية All platforms واختر Console من قائمة كل أنواع المشاريع All project types

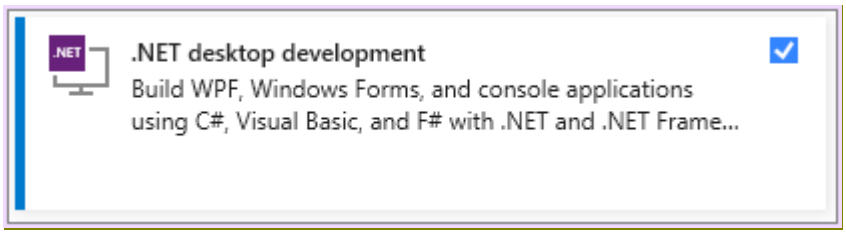
بعد تطبيق مرشحات اللغة والمنصة ونوع المشروع language, platform, project type اختر قالب تطبيق وحدة التحكم Console App ثم حدد التالي Next

ملحوظة:

إذا لم تتمكن من رؤية قالب تطبيق وحدة التحكم Console App فحدد تثبيت المزيد من الأدوات والميزات Install more tools and features

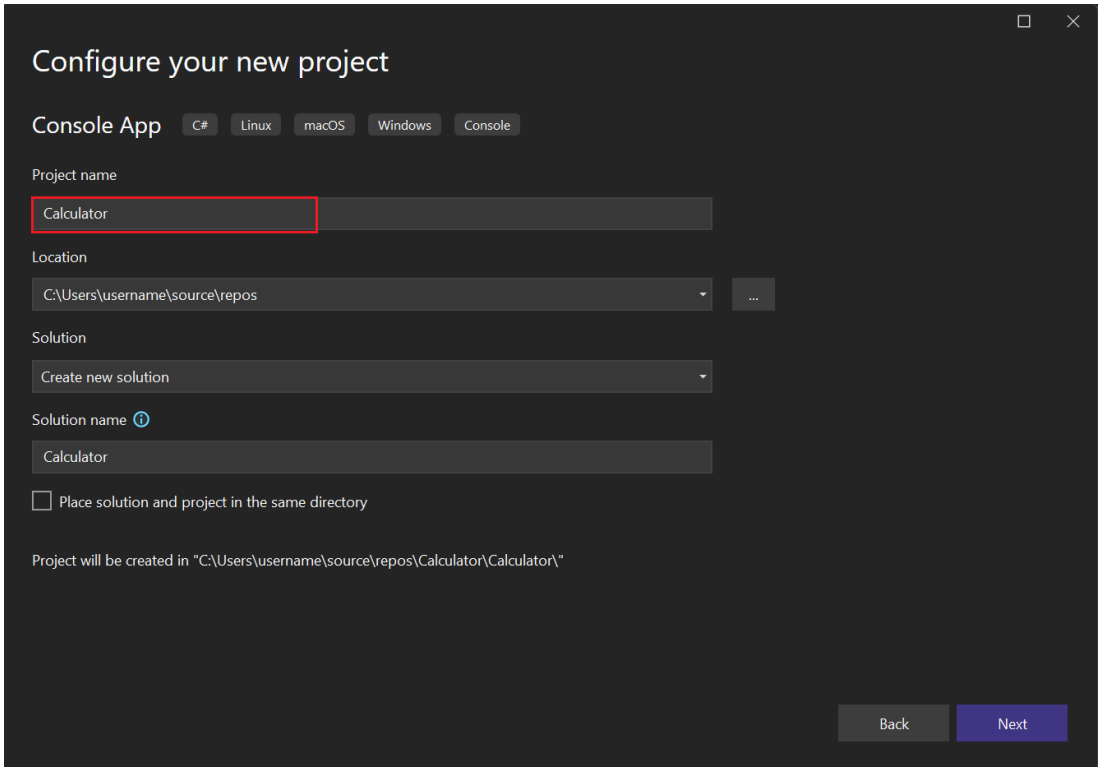


في برنامج Visual Studio Installer حدد عبء عمل تطوير سطح مكتب .NET desktop development

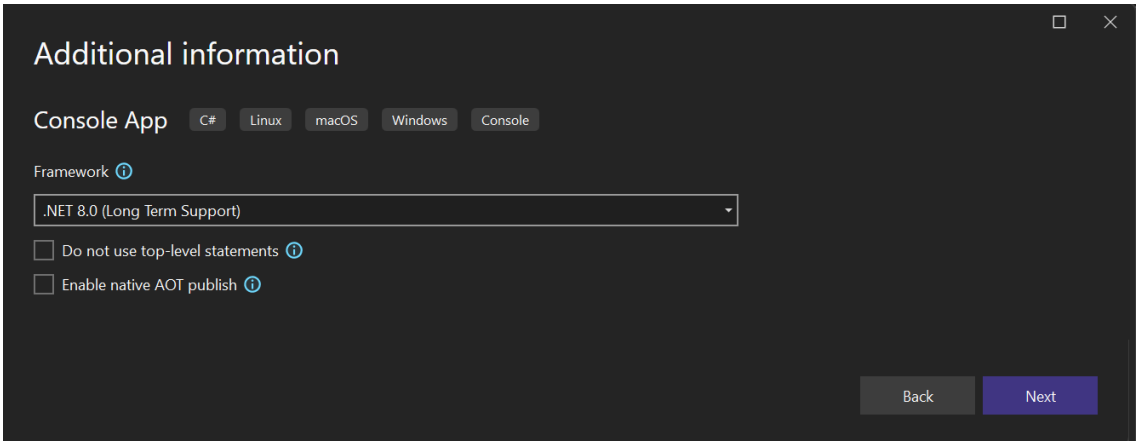


حدد تعديل Modify في برنامج Visual Studio Installer قد يُطلب منك حفظ عملك، حدد متابعة Continue لتثبيت عبء العمل.

٣. في نافذة تكوين مشروعك الجديد Configure your new project اكتب أو أكتب في مربع اسم المشروع Calculator ثم حدد التالي Next



٤. في نافذة المعلومات الإضافية Additional information حدد .NET 8.0 لحقل الإطار المستهدف Target Framework ثم حدد إنشاء Create



يفتح مشروعك الجديد، والذي يتضمن التعليمات الافتراضية "Hello World" لعرضها في المحرر، حدد ملف التعليمات Program.cs في نافذة Solution Explorer والتي توجد عادةً على الجانب الأيمن من Visual Studio

يستدعي بيان التعليمات الفردية أسلوب WriteLine لعرض السلسلة الحرفية "Hello, World!" في نافذة وحدة التحكم، إذا ضغطت على F5 فيمكنك تشغيل البرنامج الافتراضي في وضع التصحيح. بعد تشغيل التطبيق في مصحح الأخطاء، تظل نافذة وحدة التحكم مفتوحة. اضغط على أي مفتاح لإغلاق نافذة وحدة التحكم.

ملحوظة:

بدءًا من NET 6. تقوم المشاريع الجديدة التي تستخدم قالب وحدة التحكم بإنشاء تعليمات مختلفة عن الإصدارات السابقة. لمزيد من المعلومات راجع New C# templates generate top-level statements

إنشاء التطبيق

يمكنك إكمال المهام التالية:

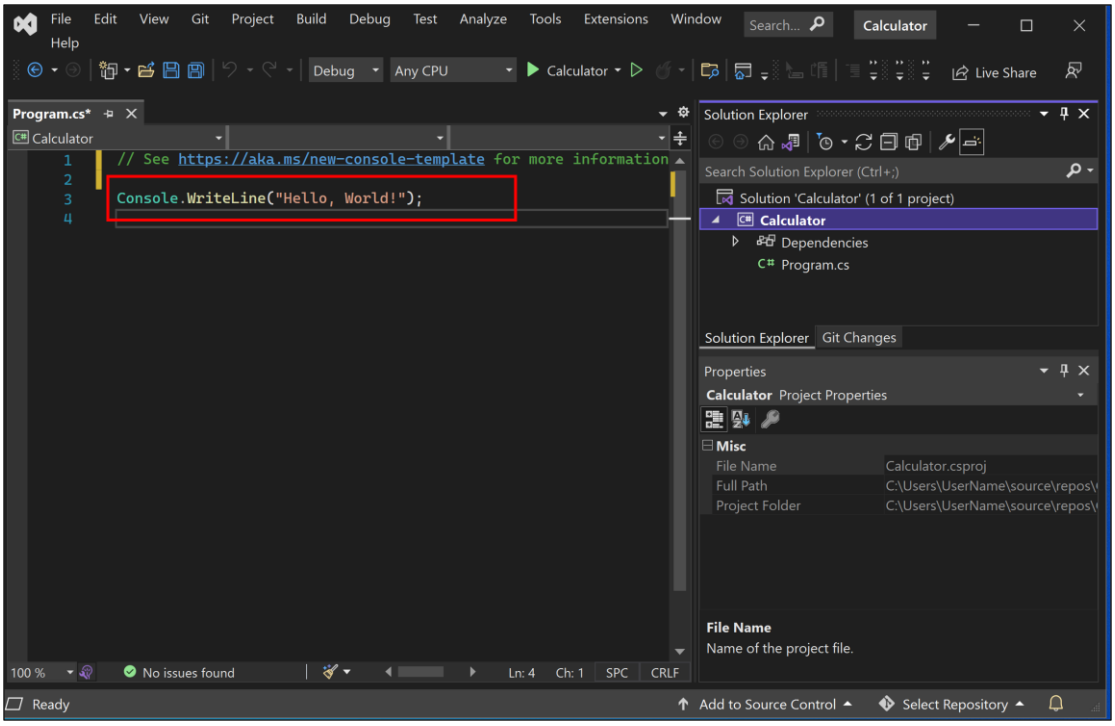
- استكشاف بعض الرياضيات الصحيحة الأساسية.
- إضافة التعليمات لإنشاء تطبيق الآلة الحاسبة الأساسي.
- تصحيح أخطاء التطبيق للعثور على الأخطاء وإصلاحها.
- تحسين التعليمات لجعلها أكثر كفاءة.

استكشاف الرياضيات الصحيحة

ابدأ ببعض الرياضيات الصحيحة الأساسية في C#

١. في مستكشف الحلول Solution Explorer في الجزء الأيمن، حدد Program.cs لعرض الملف في محرر التعليمات البرمجية.

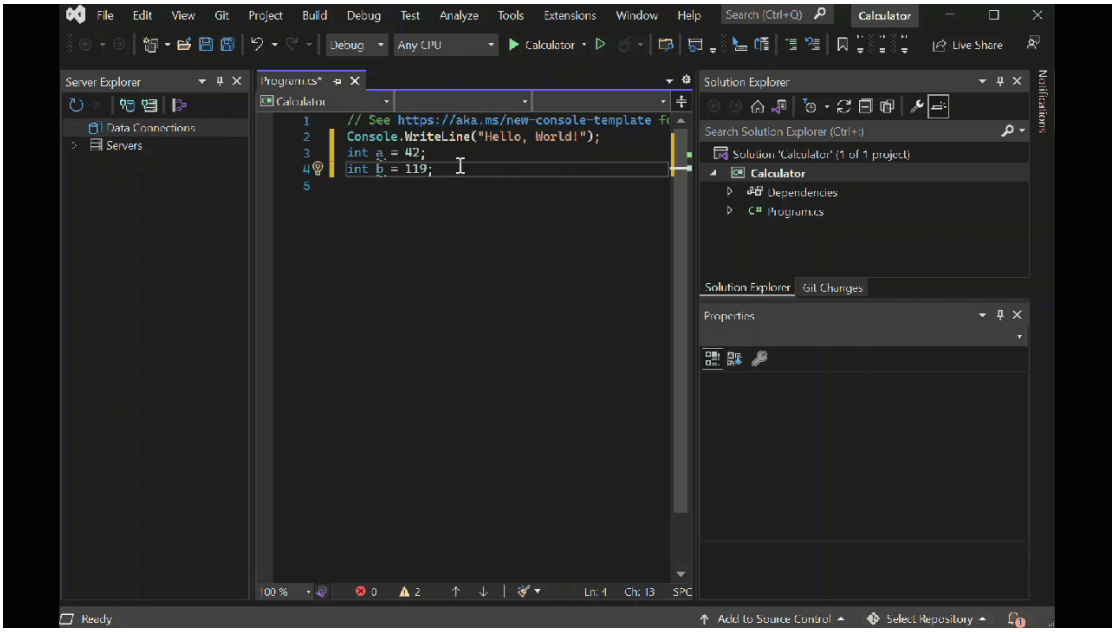
٢. في محرر التعليمات استبدل التعليمات الافتراضية "Hello World" التي تقول `Console.WriteLine("Hello World!");`



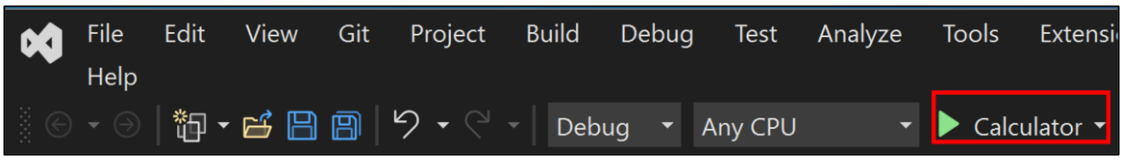
استبدل السطر بالتعليمات التالية:

```
int a = 42;
    int b = 119;
    int c = a + b;
    Console.WriteLine(c);
    Console.ReadKey();
```

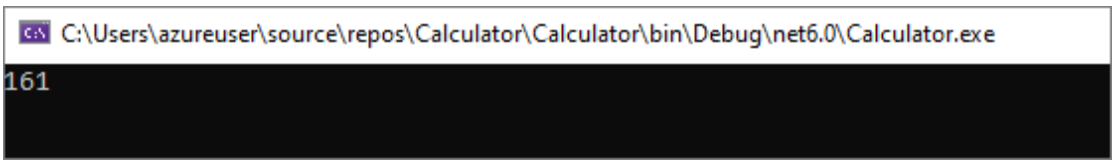
إذا قمت بإدخال التعليمات يدويًا فإن ميزة Visual Studio IntelliSense توفر لك خيار إكمال الإدخال تلقائيًا.



٣. لبناء تطبيقك وتشغيله، اضغط على **F5** أو حدد السهم الأخضر بجوار اسم Calculator في شريط الأدوات العلوي.



تفتح نافذة وحدة التحكم التي تعرض مجموع 119 + 42 وهو 161



٤. إغلاق نافذة وحدة التحكم

٥. اختياريًا، يمكنك تغيير المشغل operator لتغيير النتيجة. على سبيل المثال، يمكنك تغيير المشغل `+` في سطر التعليمات البرمجية `int c = a + b;` للطرح أو الضرب أو القسمة - for subtraction, * for multiplication, or / for division عند تشغيل التطبيق، تتغير النتيجة وفقًا لذلك.

إضافة التعليمات لإنشاء الآلة الحاسبة Calculator

واصل إضافة مجموعة أكثر تعقيدًا من تعليمات الآلة الحاسبة إلى مشروعك.

١. في محرر التعليمات، استبدل كافة التعليمات الموجودة في Program.cs بالتعليمات الجديدة التالية:

```
// Declare variables and then initialize to zero.
```

```
int num1 = 0; int num2 = 0;

// Display title as the C# console calculator
app.
Console.WriteLine("Console Calculator in
C#\r");
Console.WriteLine("-------------------------
\n");

// Ask the user to type the first number.
Console.WriteLine("Type a number, and then
press Enter");
num1 = Convert.ToInt32(Console.ReadLine());

// Ask the user to type the second number.
Console.WriteLine("Type another number, and
then press Enter");
num2 = Convert.ToInt32(Console.ReadLine());

// Ask the user to choose an option.
Console.WriteLine("Choose an option from the
following list:");
Console.WriteLine("\ta - Add");
Console.WriteLine("\ts - Subtract");
Console.WriteLine("\tm - Multiply");
Console.WriteLine("\td - Divide");
Console.Write("Your option? ");

// Use a switch statement to do the math.
  switch (Console.ReadLine())
  {
      case "a":
        Console.WriteLine($"Your result: {num1} +
{num2} = " + (num1 + num2));
          break;
      case "s":
```

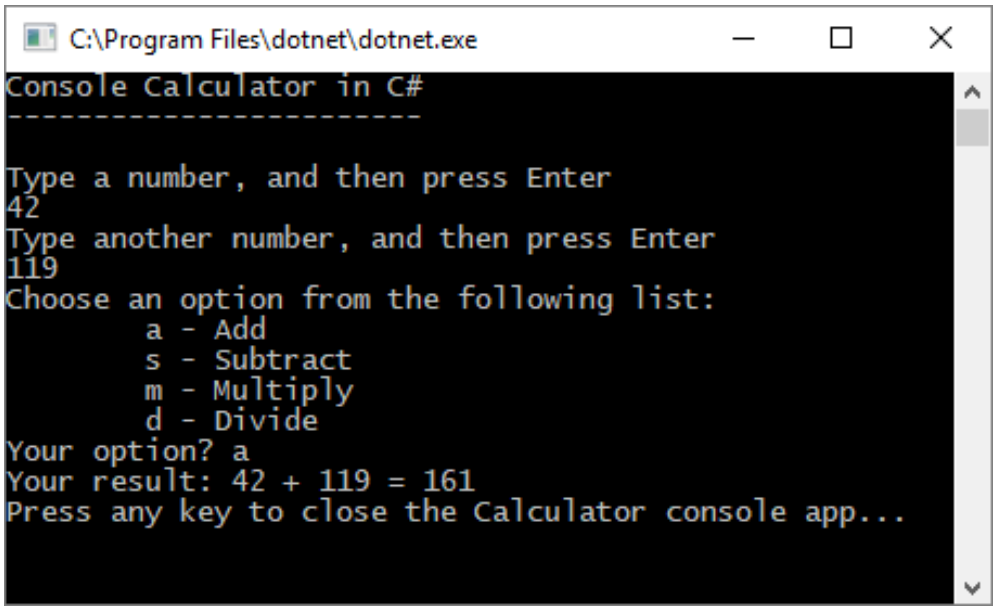
```
            Console.WriteLine($"Your result: {num1} -
{num2} = " + (num1 - num2));
             break;
        case "m":
            Console.WriteLine($"Your result: {num1} *
{num2} = " + (num1 * num2));
             break;
        case "d":
            Console.WriteLine($"Your result: {num1} /
{num2} = " + (num1 / num2));
             break;
    }
// Wait for the user to respond before closing.
Console.Write("Press any key to close the
Calculator console app...");
Console.ReadKey();
```

٢. لتشغيل التطبيق حدد زر Calculator أو اضغط على F5

يتم فتح نافذة وحدة التحكم.

٣. في نافذة وحدة التحكم، اتبع التعليمات لإضافة الرقمين 42, 119 معًا.

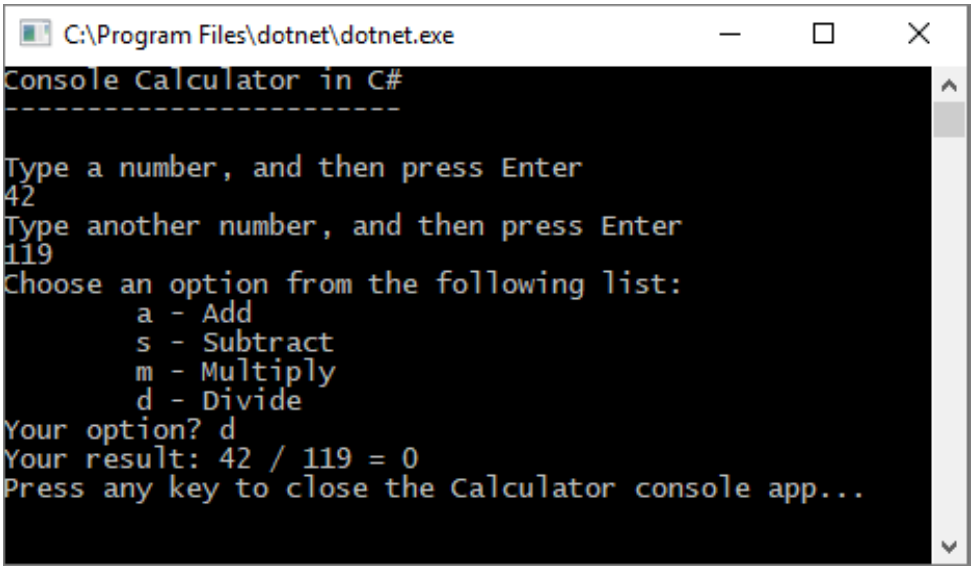
يجب أن يبدو تطبيقك مشابهًا للصورة التالية:



إضافة وظيفة عشرية decimal functionality

الآن، قم بتعديل التعليمات لإضافة المزيد من الوظائف.

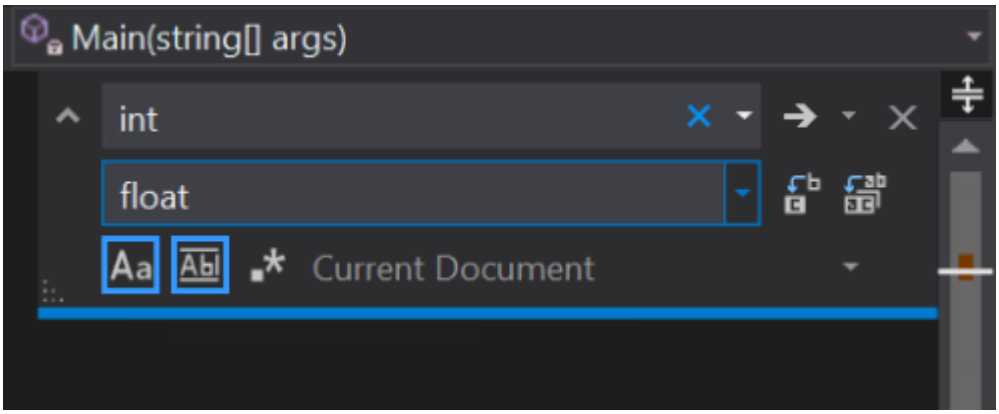
لا يقبل تطبيق الآلة الحاسبة الحالي سوى الأعداد الصحيحة ويعيدها. على سبيل المثال، إذا قمت بتشغيل التطبيق وقسمت الرقم 42 على الرقم 119 فستكون النتيجة صفرًا، وهو ما ليس دقيقًا.



إصلاح التعليمات لتحسين الدقة من خلال التعامل مع الكسور العشرية:

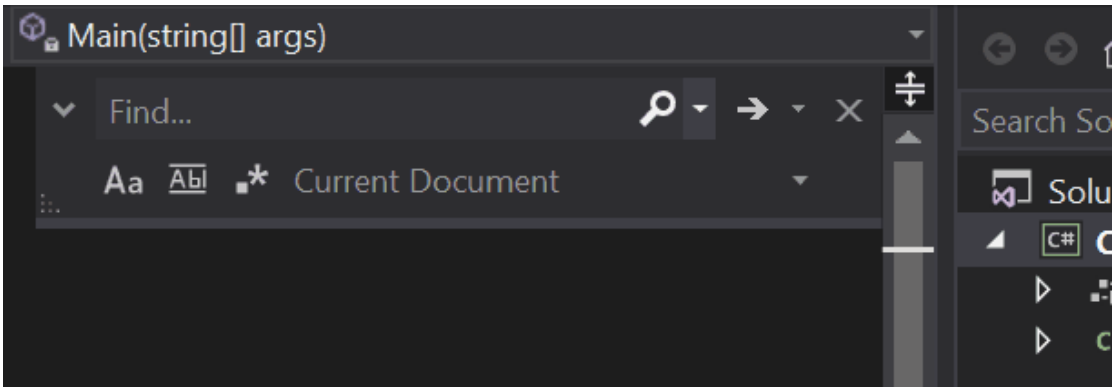
١. من محرر Program.cs اضغط على Ctrl+H لفتح عنصر التحكم والاستبدال Find and Replace

٢. اكتب int في عنصر التحكم the control واكتب float في حقل الاستبدال Replace field



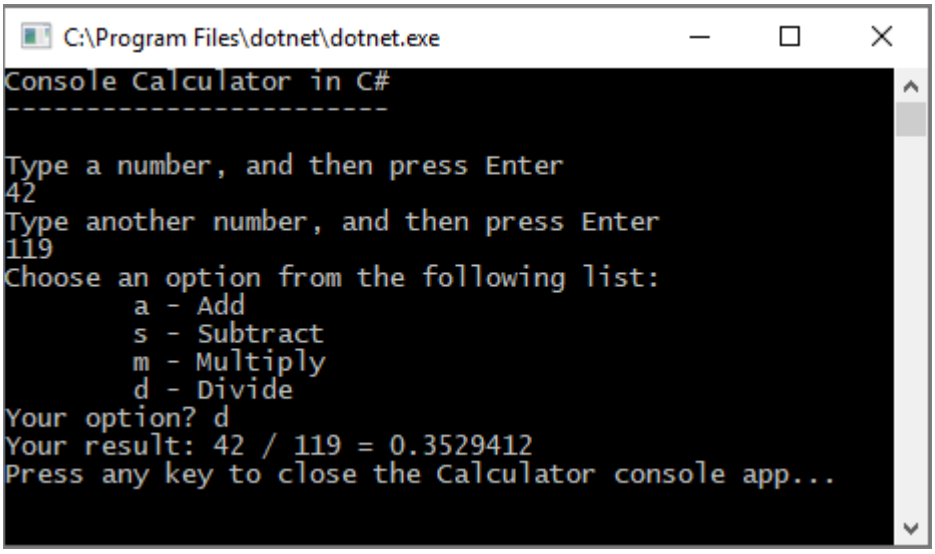
٣. قم بتحديد أيقونات مطابقة حالة الأحرف ومطابقة الكلمة بأكملها Match case and Match whole word في عنصر التحكم، أو اضغط على Alt+C, Alt+W

٤. حدد أيقونة استبدال الكل Replace all أو اضغط على Alt+A لتشغيل البحث والاستبدال search and replace



٥. قم بتشغيل تطبيق calculator مرة أخرى، ثم قسّم العدد 42 على العدد 119

يعيد التطبيق الآن رقمًا عشريًا بدلاً من الصفر.

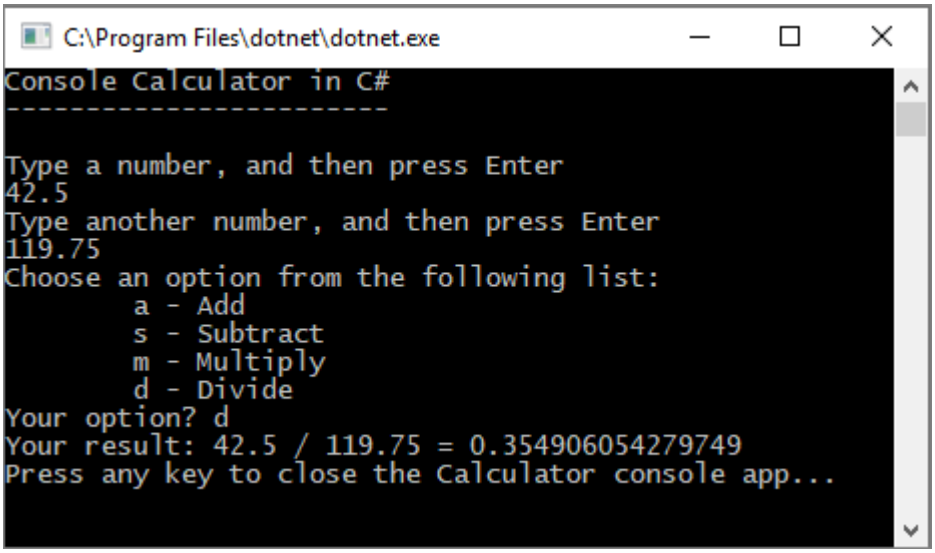


الآن أصبح التطبيق قادرًا على إنتاج نتائج عشرية. قم بإجراء بعض التعديلات الإضافية على التعليمات حتى يتمكن التطبيق من حساب الأعداد العشرية أيضًا.

٦. استخدم عنصر التحكم Find and Replace لتغيير كل مثيل للمتغير `float` إلى `double` ولتغيير كل مثيل للأسلوب `Convert.ToInt32` إلى `Convert.ToDouble`

٧. قم بتشغيل تطبيق calculator ثم قسّم الرقم 42.5 على الرقم 119.75

يقبل التطبيق الآن القيم العشرية، ويعيد رقمًا عشريًا أطول كنتيجة له.



في قسم Revise the code يمكنك تقليل عدد الأماكن العشرية في النتائج.

تصحيح أخطاء التطبيق Debug

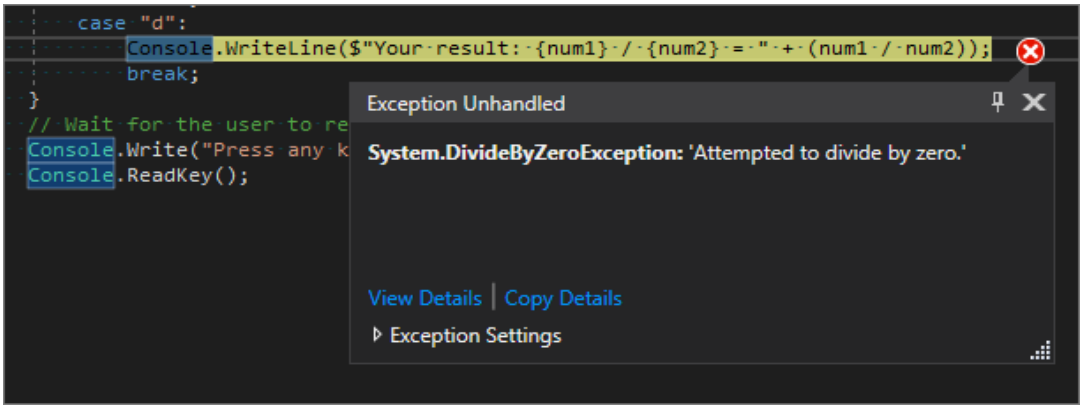
لقد قمت بتحسين تطبيق الآلة الحاسبة الأساسي الخاص بك، ولكن التطبيق لا يتعامل بعد مع الاستثناءات handle exceptions مثل أخطاء إدخال المستخدم. على سبيل المثال، إذا حاول المستخدمون القسمة على الصفر، أو إدخال حرف غير متوقع، فقد يتوقف التطبيق عن العمل، أو يعرض خطأ، أو يعرض نتيجة غير رقمية non-numeric result غير متوقعة.

دعونا نلقي نظرة على بعض أخطاء إدخال المستخدم الشائعة، ونحدد موقعها في المصحح إذا ظهرت هناك، ونصلحها في التعليمات

نصيحة: لمزيد من المعلومات حول مصحح الأخطاء وكيفية عمله، راجع النظرة الأولى على مصحح أخطاء Visual Studio

إصلاح خطأ القسمة على الصفر divide by zero

إذا حاولت قسمة رقم على الصفر، فقد يتجمد تطبيق وحدة التحكم، ثم يظهر لك الخطأ في محرر التعليمات البرمجية.



ملحوظة:

في بعض الأحيان لا يتجمد التطبيق ولا يعرض المصحح خطأ القسمة على الصفر. وبدلًا من ذلك، قد يعيد التطبيق نتيجة غير رقيمة، غير متوقعة، مثل رمز اللانهاية. لا يزال إصلاح التعليمات التالي ساريًا.

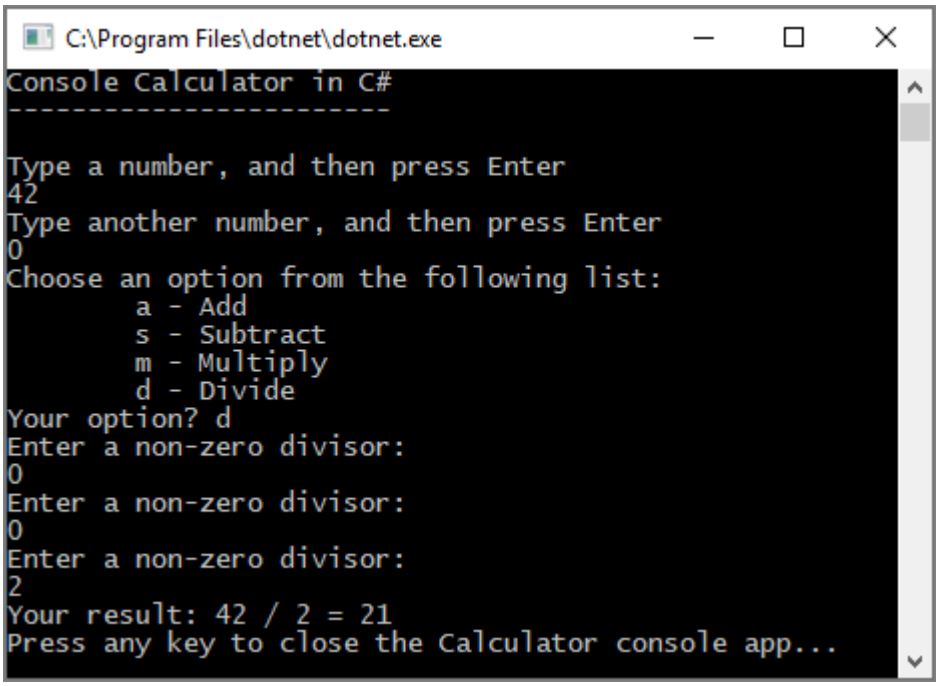
دعنا نغير التعليمات للتعامل مع هذا الخطأ. في Program.cs استبدل التعليمات الخاصة به `case "d":` بالتعليمات التالية:

```
// Ask the user to enter a non-zero divisor until they do so.
    while (num2 == 0)
   {
    Console.WriteLine("Enter a non-zero divisor: ");
    num2 = Convert.ToDouble(Console.ReadLine());
   }
 Console.WriteLine($"Your result: {num1} / {num2} = "
+ (num1 / num2));
 break;
}
```

بعد استبدال التعليمات، يجب أن يبدو القسم الذي يحتوي على العبارة `switch` مشابهًا للصورة التالية:

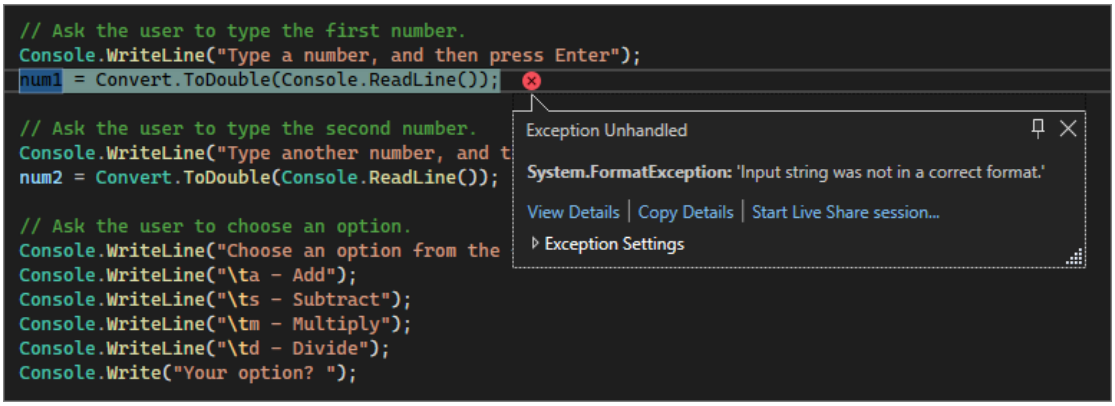
```
// Use a switch statement to do the math
switch (Console.ReadLine())
{
    case "a":
        Console.WriteLine($"Your result: {num1} + {num2} = " + (num1 + num2));
        break;
    case "s":
        Console.WriteLine($"Your result: {num1} - {num2} = " + (num1 - num2));
        break;
    case "m":
        Console.WriteLine($"Your result: {num1} * {num2} = " + (num1 * num2));
        break;
    case "d":
        // Ask the user to enter a non-zero divisor until they do so
        while (num2 == 0)
        {
            Console.WriteLine("Enter a non-zero divisor: ");
            num2 = Convert.ToInt32(Console.ReadLine());
        }
        Console.WriteLine($"Your result: {num1} / {num2} = " + (num1 / num2));
        break;
}
// Wait for the user to respond before closing
Console.Write("Press any key to close the Calculator console app...");
Console.ReadKey();
```

الآن، عندما تقوم بقسمة أي رقم على الصفر، يطلب التطبيق رقمًا آخر، ويستمر في السؤال حتى تقدم رقمًا غير صفري nonzero number



إصلاح خطأ التنسيق format

إذا أدخلت حرفًا أبجديًا بينما يتوقع التطبيق حرفًا رقميًا، يتجمد التطبيق. يعرض لك الخطأ التالي في محرر التعليمات البرمجية.



لمنع هذا الاستثناء، يمكنك إعادة صياغة التعليمات التي أدخلتها مسبقًا.

مراجعة التعليمات Revise the code

بدلاً من الاعتماد على فئة `Program` للتعامل مع كافة التعليمات البرمجية، يمكنك تقسيم تطبيقك إلى فئتين `Calculator and Program`

تتولى الفئة `Calculator` التعامل مع الجزء الأكبر من عمل الحساب bulk of the calculation وتتولى الفئة `Program` التعامل مع واجهة المستخدم ومعالجة الأخطاء user interface and error-handling

١. احذف كل شيء في Program.cs وأضف الفئة الجديدة Calculator التالية:

```
class Calculator
{
public static double DoOperation(double num1, double
num2, string op)
  {
   double result = double.NaN; // Default value is "not-a-
number" if an operation, such as division, could result in an error.

  // Use a switch statement to do the math.
   switch (op)
   {
     case "a":
         result = num1 + num2;
         break;
     case "s":
         result = num1 - num2;
         break;
     case "m":
         result = num1 * num2;
         break;
     case "d":
        // Ask the user to enter a non-zero divisor.
        if (num2 != 0)
         {
             result = num1 / num2;
         }
         break;
    // Return text for an incorrect option entry.
           default:
         break;
   }
    return result;
  }
}
```

٢. أضف أيضًا فئة جديدة Program على النحو التالي:

```
class Program
{
   static void Main(string[] args)
  {
   bool endApp = false;
   // Display title as the C# console calculator app.
   Console.WriteLine("Console Calculator in C#\r");
   Console.WriteLine("------------------------\n");

   while (!endApp)
   {
   // Declare variables and set to empty.
  // Use Nullable types (with ?) to match type of
System.Console.ReadLine
     string? numInput1 = "";
     string? numInput2 = "";
     double result = 0;

     // Ask the user to type the first number.
     Console.Write("Type a number, and then press
Enter: ");
     numInput1 = Console.ReadLine();

   double cleanNum1 = 0;
   while (!double.TryParse(numInput1, out cleanNum1))
    {
     Console.Write("This is not valid input.
Please enter a numeric value: ");
       numInput1 = Console.ReadLine();
      }

     // Ask the user to type the second number.
     Console.Write("Type another number, and then
press Enter: ");
     numInput2 = Console.ReadLine();

   double cleanNum2 = 0;
   while (!double.TryParse(numInput2, out cleanNum2))
        {
```

```
   Console.Write("This is not valid input. Please
enter a numeric value: ");
    numInput2 = Console.ReadLine();
        }

 // Ask the user to choose an operator.
  Console.WriteLine("Choose an operator from the
following list:");
  Console.WriteLine("\ta - Add");
  Console.WriteLine("\ts - Subtract");
  Console.WriteLine("\tm - Multiply");
  Console.WriteLine("\td - Divide");
  Console.Write("Your option? ");

  string? op = Console.ReadLine();

  // Validate input is not null, and matches the pattern
  if (op == null || ! Regex.IsMatch(op, "[a|s|m|d]"))
  {
  Console.WriteLine("Error: Unrecognized input.");
  }
  else
  {
    try
  {
  result = Calculator.DoOperation(cleanNum1,
cleanNum2, op);
      if (double.IsNaN(result))
       {
    Console.WriteLine("This operation will result
in a mathematical error.\n");
       }
     else Console.WriteLine("Your result:
{0:0.##}\n", result);
       }
     catch (Exception e)
      {
  Console.WriteLine("Oh no! An exception occurred
trying to do the math.\n - Details: " + e.Message);
     }
```

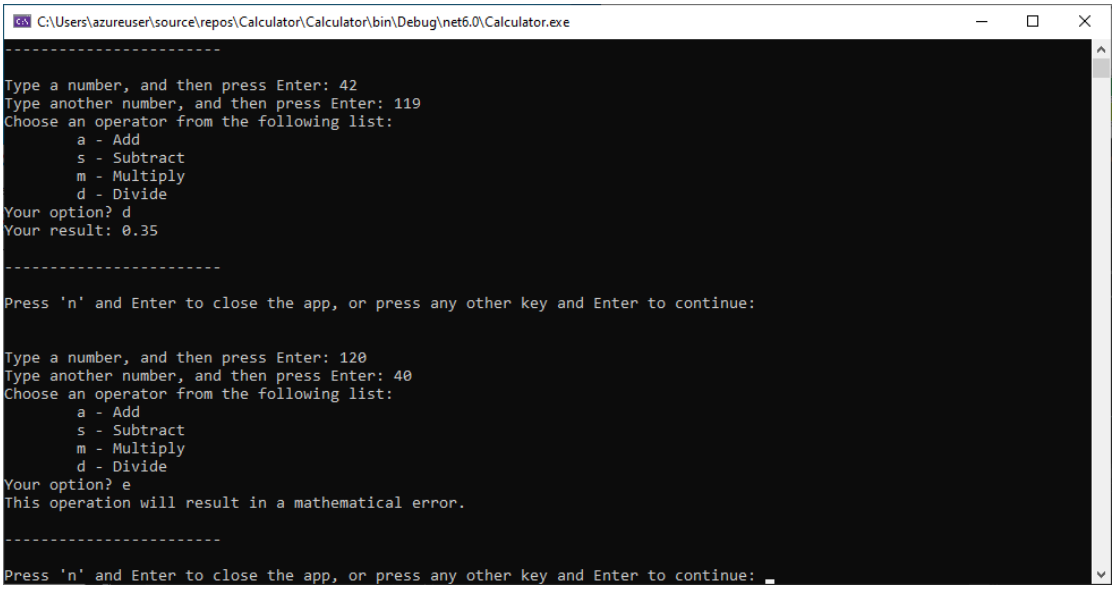
```
      }
  Console.WriteLine("------------------------\n");

 // Wait for the user to respond before closing.
 Console.Write("Press 'n' and Enter to close the app,
or press any other key and Enter to continue: ");
 if (Console.ReadLine() == "n") endApp = true;

 Console.WriteLine("\n"); // Friendly linespacing.
  }
   return;
  }
}
```

ملحوظة:

من الأفضل استخدام أنواع قابلة للعدم (مع الرمز ?) لسلاسل الإدخال input strings حيث System.Console.ReadLine يتم إرجاع نوع مرجعي قابل للعدم

٣. حدد زر **Calculator** أو اضغط على F5 لتشغيل تطبيقك.

٤. اتبع التعليمات وقم بقسمة الرقم **42** على الرقم **119** يجب أن تبدو نتائجك مشابهة للقطة الشاشة التالية:

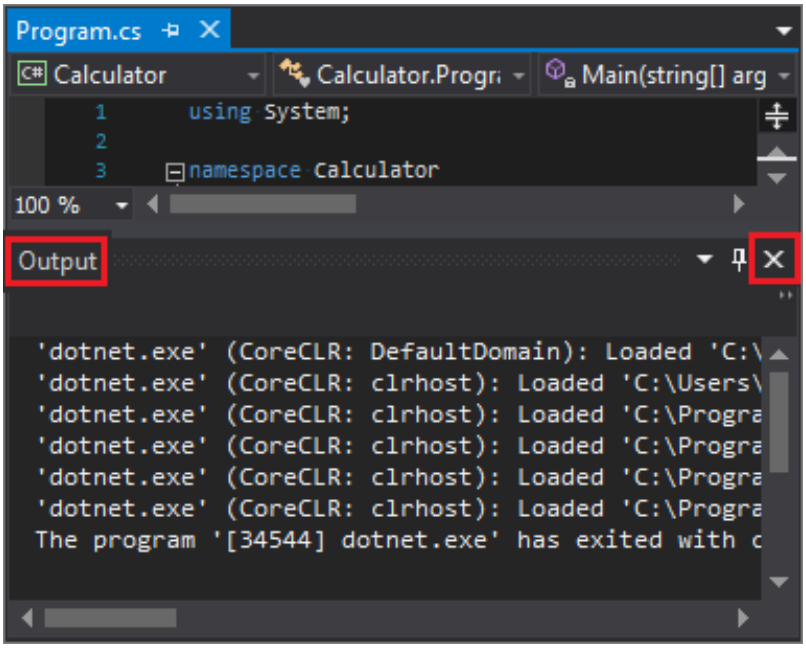


يمكنك الآن إجراء المزيد من العمليات الحسابية حتى تختار إغلاق تطبيق وحدة التحكم. كما يوجد عدد أقل من الأماكن العشرية في النتائج. وإذا أدخلت حرفًا غير صحيح، فستحصل على استجابة خطأ مناسبة.

اغلاق التطبيق Close the app

١. إذا لم تقم بإغلاقه بالفعل، فأغلق تطبيق Calculator

٢. أغلق جزء الإخراج Output في Visual Studio



٣. في Visual Studio اضغط على Ctrl+S احفظ تطبيقك.

إضافة عنصر التحكم في مصدرGit (Add Git source control)

الآن بعد أن أصبح لديك تطبيق، قد ترغب في إضافته إلى مستودع Git يجعل Visual Studio هذه العملية سهلة باستخدام أدوات Git التي يمكنك استخدامها مباشرة من IDE

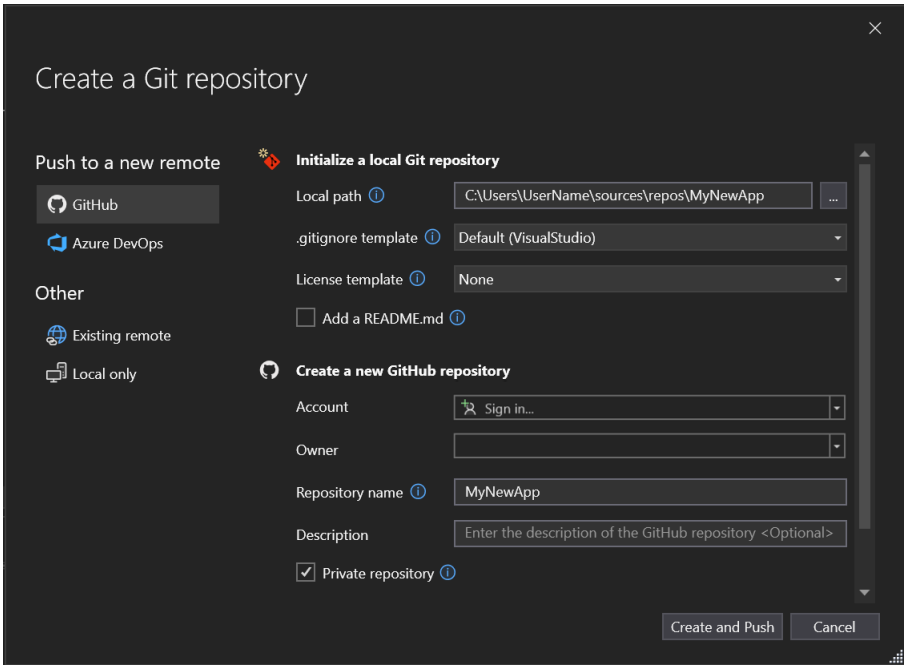
ملحوظة:

Git هو نظام التحكم في الإصدارات الأكثر استخدامًا على الإطلاق. سواء كنت مطورًا محترفًا أو تتعلم كيفية البرمجة، فإن Git يمكن أن يكون مفيدًا للغاية. إذا كنت جديدًا على Git فإن موقع الويب https://git-scm.com/ هو مكان جيد للبدء. يمكنك العثور على أوراق الغش وكتب شائعة عبر الإنترنت ومقاطع فيديو حول أساسيات Git

لربط تعليماتك بـ Git ابدأ بإنشاء مستودع Git جديد حيث توجد تعليماتك:

١. في شريط الحالة الموجود في أسفل يمين Visual Studio حدد إضافة إلى عنصر التحكم في المصدر Add to Source Control ثم حدد Git

٢. في مربع الحوار Create a Git repository قم بتسجيل الدخول إلى GitHub

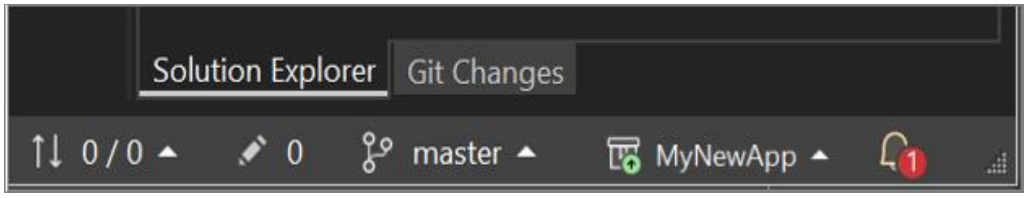


يتم ملء اسم المستودع تلقائيًا استنادًا إلى موقع المجلد الخاص بك. المستودع الجديد خاص بشكل افتراضي، مما يعني أنك الشخص الوحيد الذي يمكنه الوصول إليه.

نصيحة

سواء كان مستودعك عامًا أو خاصًا، فمن الأفضل أن يكون لديك نسخة احتياطية عن بُعد من تعليماتك مخزنة بشكل آمن على GitHub حتى إذا لم تكن تعمل مع فريق، فإن المستودع البعيد يجعل تعليماتك متاحة لك من أي جهاز كمبيوتر.

٣. حدد إنشاء ودفع Create and Push بعد إنشاء مستودعك، سترى تفاصيل الحالة في شريط الحالة:



استخدام إجراءات Git actions في Visual Studio

فيما يلي ملخص موجز لإجراءات Git المتوفرة في شريط حالة Visual Studio

- تُظهر الأسهم لأعلى/لأسفل Up/Down عدد الالتزامات الصادرة/الواردة outgoing/incoming commits في فرعك الحالي. يمكنك استخدام هذا الرمز لسحب أي التزامات واردة أو دفع أي التزامات صادرة.
- لعرض التزام محدد، حدد السهم لأعلى/لأسفل Up/Down ثم حدد عرض الصادر/الوارد View Outgoing/Incoming
- يعرض لك القلم الرصاص Pencil عدد التغييرات غير الملتزمة في تعليماتك. يمكنك تحديد هذا الرمز لعرض تلك التغييرات في نافذة تغييرات Git

توفر قائمة **Git** أدوات لإجراءات المستودع على ملفاتك. يمكنك استخدام use git fetch, pull, push, and sync for version control in Visual Studio

لمعرفة المزيد حول كيفية استخدام Git مع تطبيقك، راجع وثائق التحكم في إصدار Visual Studio

المراجعة: التعليمات البرمجية المكتملة Code complete

في هذه الوحدة، قمت بإجراء العديد من التغييرات على تطبيق Calculator يتعامل التطبيق الآن مع موارد الحوسبة handles computing resources بكفاءة أكبر، ويتعامل مع معظم أخطاء إدخال المستخدم user input errors

إليك التعليمات الكاملة، كلها في مكان واحد:

```
class Calculator
{
    public static double DoOperation(double num1,
double num2, string op)
    {
        double result = double.NaN; // Default value
is "not-a-number" which we use if an operation, such
as division, could result in an error.

        // Use a switch statement to do the math.
        switch (op)
        {
            case "a":
                result = num1 + num2;
                break;
            case "s":
                result = num1 - num2;
                break;
            case "m":
                result = num1 * num2;
                break;
            case "d":
             // Ask the user to enter a non-zero divisor.
                if (num2 != 0)
                {
                    result = num1 / num2;
                }
                break;
        // Return text for an incorrect option entry.
            default:
                break;
        }
        return result;
    }
}
```

```
class Program
{
    static void Main(string[] args)
    {
        bool endApp = false;
   // Display title as the C# console calculator app.
     Console.WriteLine("Console Calculator in C#\r");
     Console.WriteLine("------------------------\n");

        while (!endApp)
        {
    // Declare variables and set to empty.
   // Use Nullable types (with ?) to match type of
System.Console.ReadLine
            string? numInput1 = "";
            string? numInput2 = "";
            double result = 0;

            // Ask the user to type the first number.
      Console.Write("Type a number, and then press
Enter: ");
            numInput1 = Console.ReadLine();

            double cleanNum1 = 0;
   while (!double.TryParse(numInput1, out cleanNum1))
            {
              Console.Write("This is not valid input.
Please enter a numeric value: ");
                numInput1 = Console.ReadLine();
            }

         // Ask the user to type the second number.
         Console.Write("Type another number, and then
press Enter: ");
            numInput2 = Console.ReadLine();

            double cleanNum2 = 0;
   while (!double.TryParse(numInput2, out cleanNum2))
            {
         Console.Write("This is not valid input.
Please enter a numeric value: ");
                numInput2 = Console.ReadLine();
```

١٠

```
            }

            // Ask the user to choose an operator.
            Console.WriteLine("Choose an operator
from the following list:");
            Console.WriteLine("\ta - Add");
            Console.WriteLine("\ts - Subtract");
            Console.WriteLine("\tm - Multiply");
            Console.WriteLine("\td - Divide");
            Console.Write("Your option? ");

            string? op = Console.ReadLine();

    // Validate input is not null, and matches the pattern
  if (op == null || ! Regex.IsMatch(op, "[a|s|m|d]"))
            {
     Console.WriteLine("Error: Unrecognized input.");
            }
            else
            {
                try
                {
result = Calculator.DoOperation(cleanNum1, cleanNum2, op);
                    if (double.IsNaN(result))
                    {
                Console.WriteLine("This operation will
result in a mathematical error.\n");
                    }
    else Console.WriteLine("Your result: {0:0.##}\n",
result);
                }
                catch (Exception e)
                {
    Console.WriteLine("Oh no! An exception occurred
trying to do the math.\n - Details: " + e.Message);
                }
            }
   Console.WriteLine("-----------------------\n");

   // Wait for the user to respond before closing.
 Console.Write("Press 'n' and Enter to close the app,
or press any other key and Enter to continue: ");
```

```
        if (Console.ReadLine() == "n") endApp = true;

    Console.WriteLine("\n"); // Friendly linespacing.
        }
        return;
    }
}
```

توسيع تطبيق وحدة التحكم calculator وتصحيح الأخطاء

في القسم الثاني من تطبيق calculator ستتعرف بشكل أعمق على ميزات البناء وتصحيح الأخطاء في Visual Studio التي تحتاجها للتطوير اليومي. تتضمن هذه الميزات إدارة مشاريع متعددة وتصحيح الأخطاء والإشارة إلى حزم الجهات الخارجية. يمكنك تشغيل تطبيق وحدة التحكم C# الذي أنشأته، واستكشاف بعض ميزات بيئة التطوير المتكاملة (IDE) في Visual Studio

في القسم الثاني، يمكنك إكمال المهام التالية:

- أضف مشروعًات ثانيًا.
- المكتبات المرجعية وإضافة الحزم.
- تصحيح أخطاء تعليماتك.
- افحص تعليماتك المكتملة.

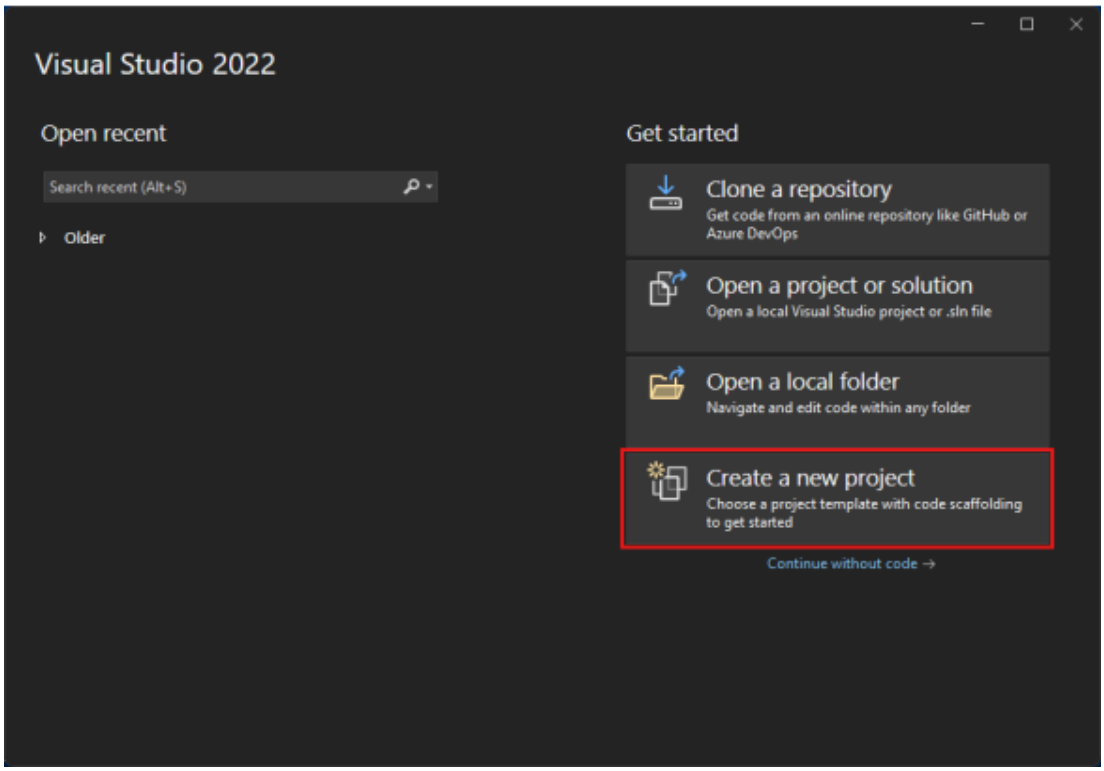
٣ إنشاء أول تطبيق Windows

إنشاء المشروع

أولًا، قم بإنشاء مشروع تطبيق C# يأتي نوع المشروع مع جميع ملفات القالب التي تحتاجها لإنشاء التطبيق.

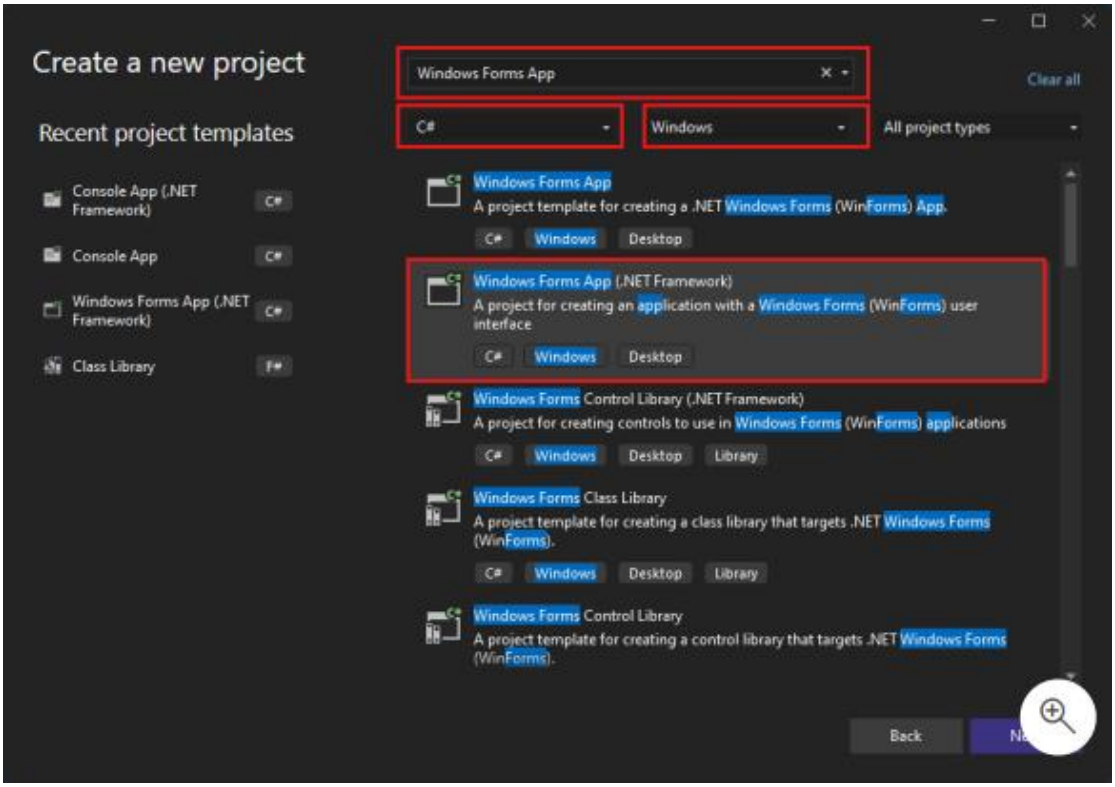
١. افتح Visual Studio 2022

٢. في نافذة البدء، حدد إنشاء مشروع جديد Create a new project

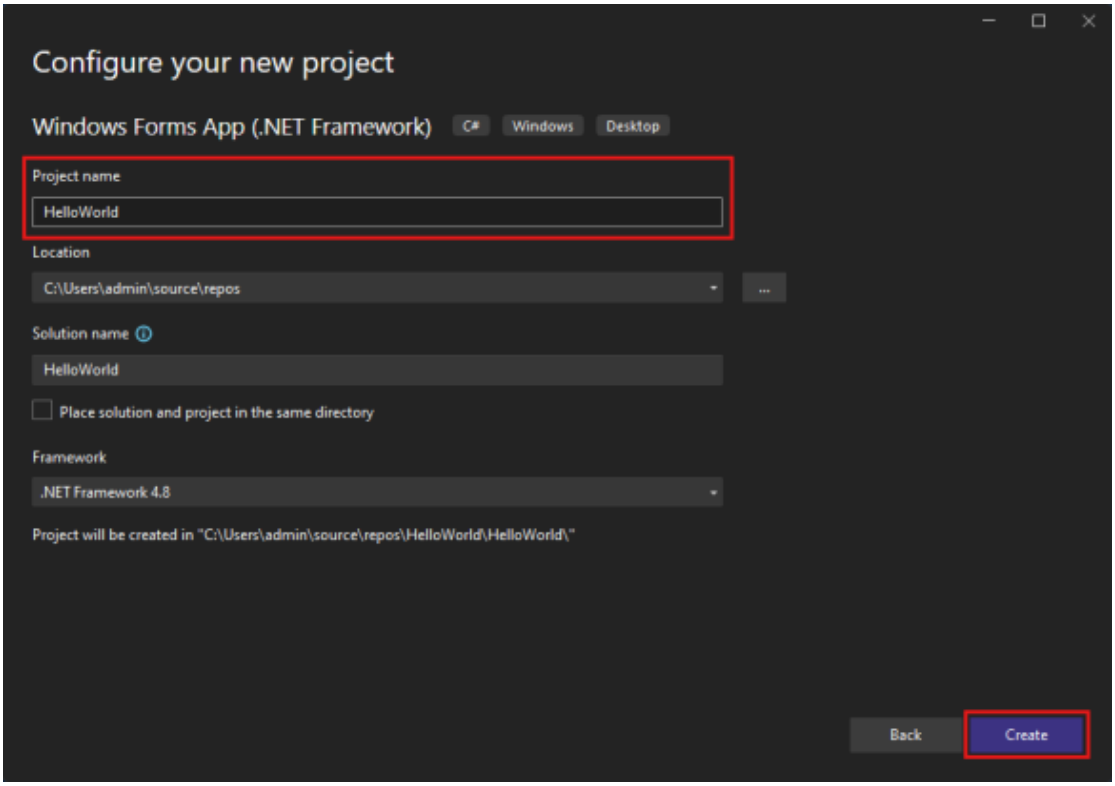


٣. في إنشاء مشروع جديد، حدد Windows Forms App(.NET Framework) ل C#

يمكنك تحسين عملية البحث للوصول بسرعة إلى القالب الذي تريده، على سبيل المثال، اكتب في مربع البحث Windows Forms App من قائمة اللغات حدد C# بعد ذلك، من قائمة المنصات حدد Windows ثم التالي Next



٤. في نافذة تكوين مشروعك الجديد، أكتب في اسم المشروع HelloWorld ثم حدد إنشاء Create



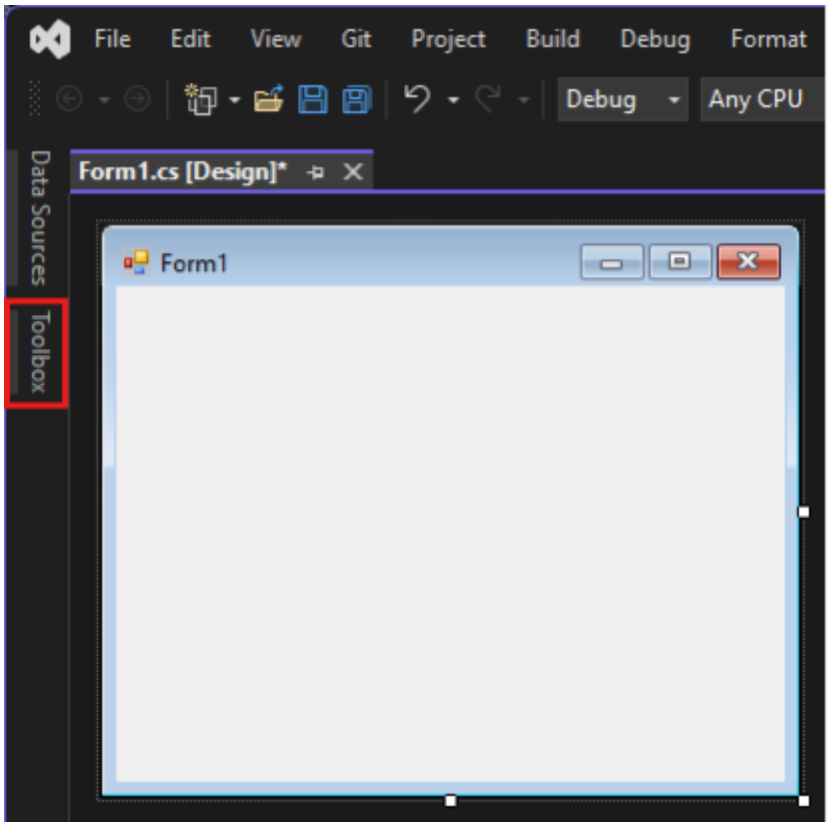
يفتح Visual Studio مشروعك الجديد.

إنشاء التطبيق

يفتح لك Studio Visual نموذجاً Form النموذج هو واجهة وتسمية المشروع، يتيح لك إضافة عناصر التحكم.

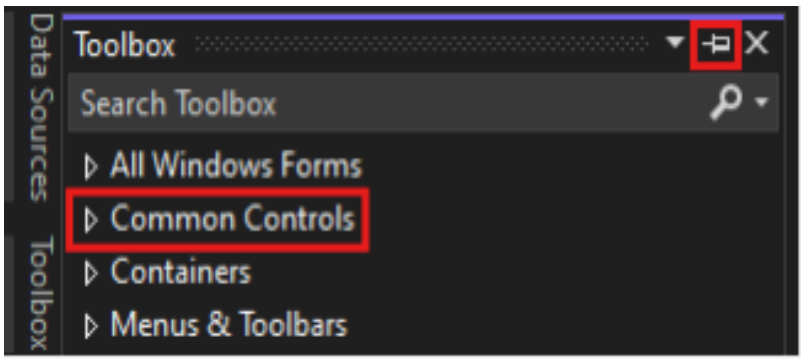
إضافة زر Button إلى النموذج Form

١. حدد Toolbox "صندوق الأدوات" لفتح نافذة Common Controls "مجموعة الأدوات" المنبثقة.

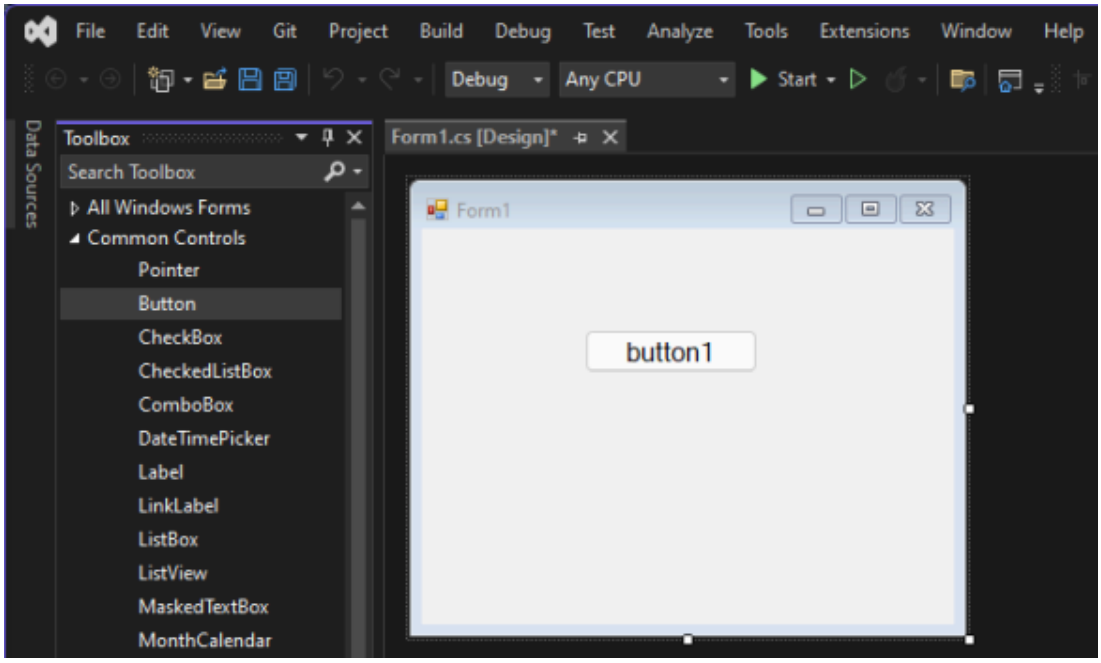


إذا لم تتمكن من رؤية خيار Toolbox "صندوق الأدوات" فيمكنك فتحه من شريط القائمة. حدد View اختر Toolbox أو اضغط **Ctrl + Alt + X**

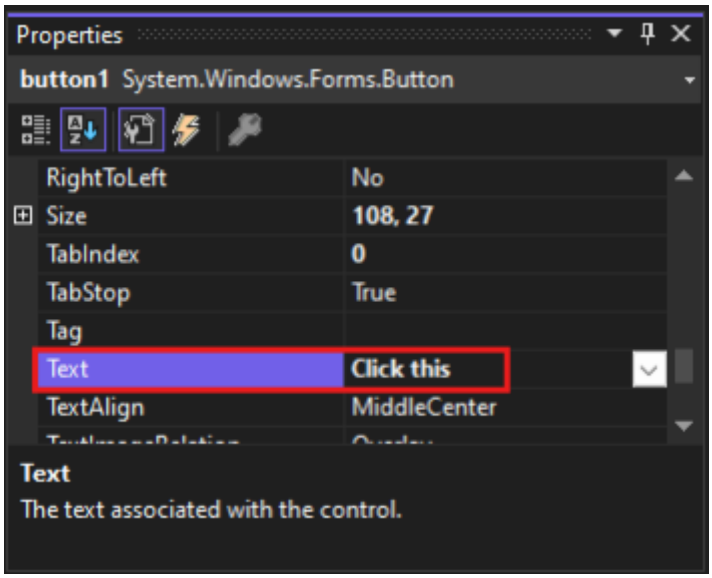
٢. حدد أيقونة الدبوس لتثبيت نافذة صندوق الأدوات Toolbox



٣. قم بتحديد عنصر التحكم Button "الزر" ثم اسحبه إلى النموذج Form

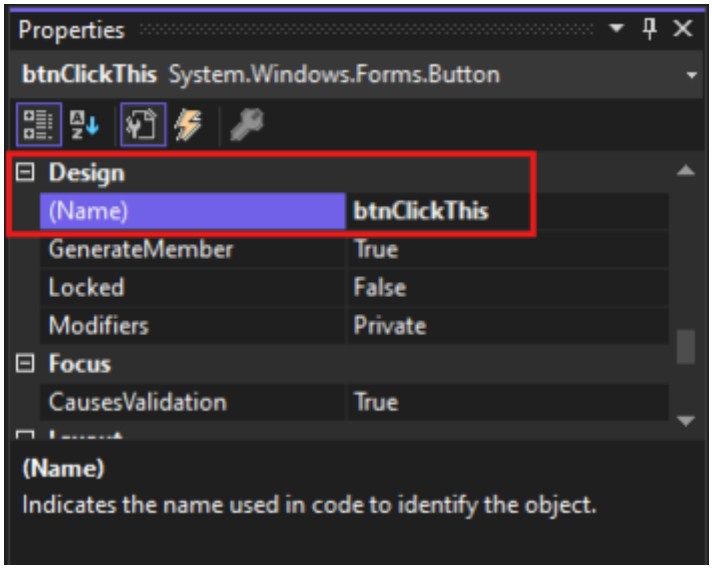


٤. في نافذة الخصائص Properties حدد النص Text قم بتغيير الاسم من button1 إلى click this ثم أضغط Enter



إذا لم تتمكن من رؤية نافذة الخصائص Properties فيمكنك فتحها من شريط القائمة. حدد View ثم اختر Properties أو اضغط على F4

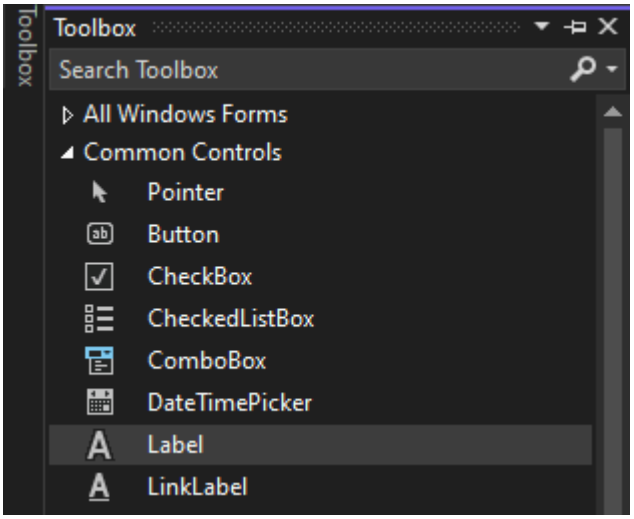
٥. في نافذة الخصائص Properties حدد خاصية الاسم Name قم بتغيير الاسم من button1 إلى btnClickThis ثم اضغط Enter



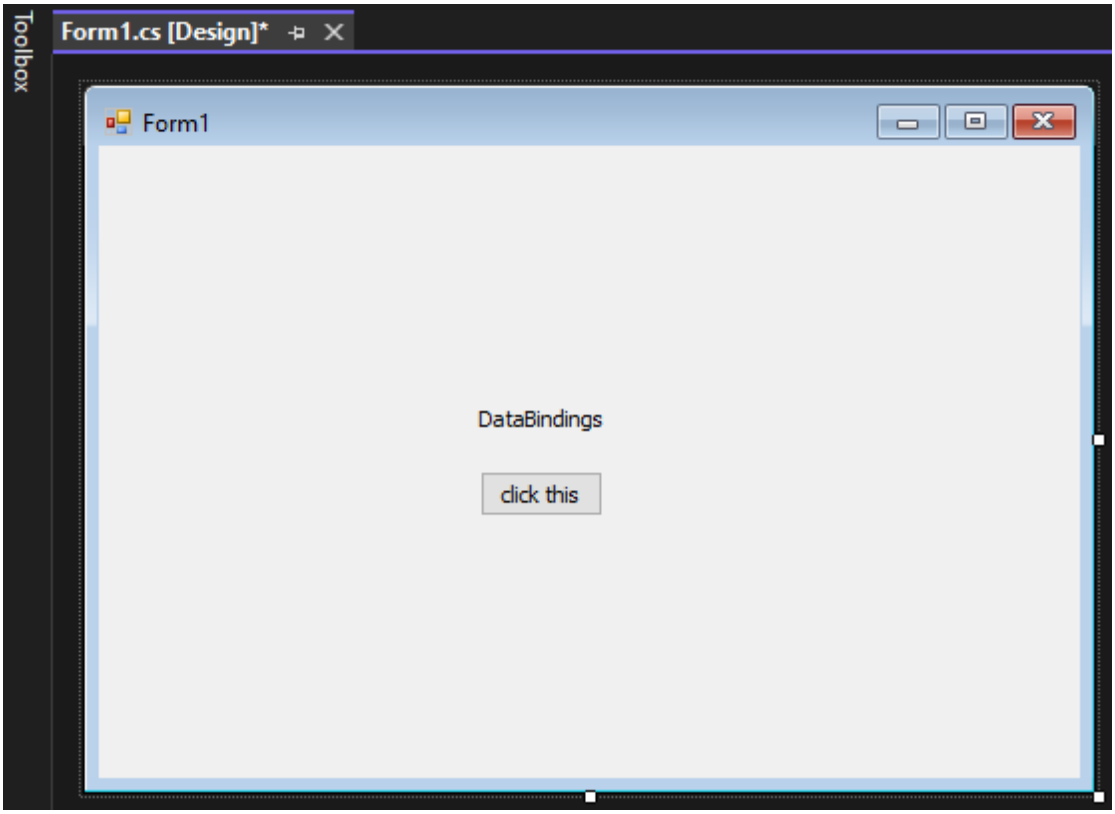
إضافة عنصر تسمية Label إلى النموذج Form

بعد إضافة عنصر تحكم زر لإنشاء إجراء، أضف عنصر تحكم تسمية **Label** لإظهار نص محدد، يسهل فهم البرنامج على المستخدم.

١. حدد عنصر التحكم **Label** من صندوق الأدوات، ثم اسحبه إلى النموذج وأسقطه أسفل الزر.

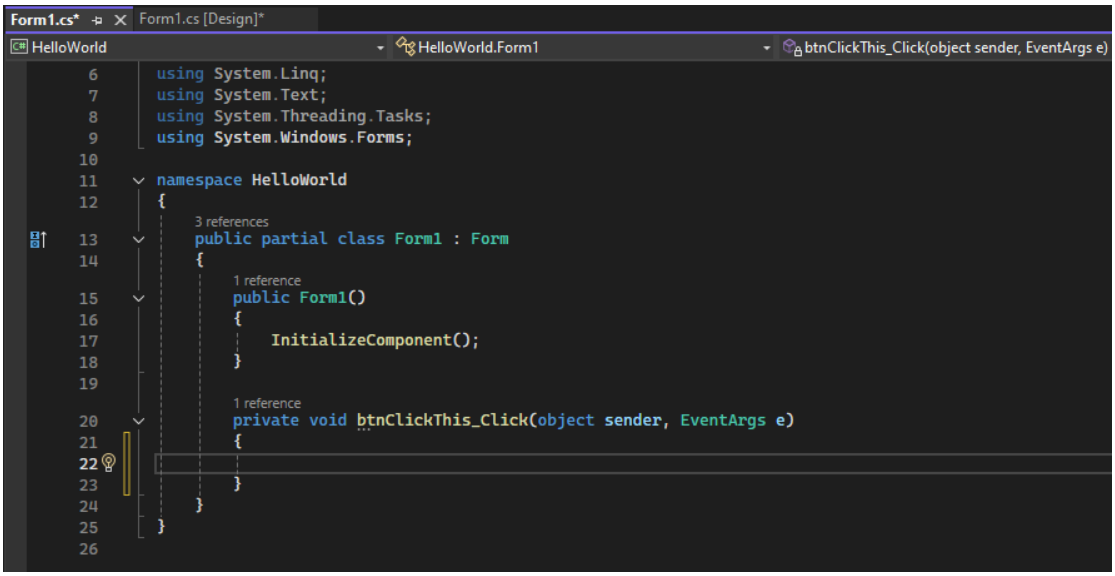


٢. من نافذة الخصائص Properties قم بتغيير Text نص label1 إلى DataBindings والاسم Name إلى lblHelloWorld



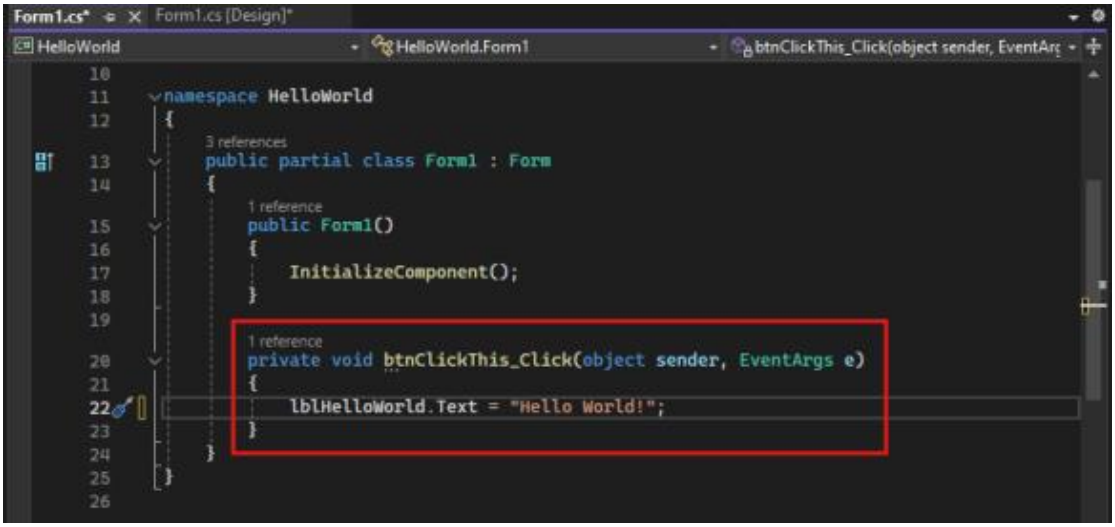
إضافة التعليمات البرمجية إلى النموذج Form

١. من قائمة تصميم Form انقر نقرًا مزدوجًا فوق الزر click this لفتح نافذة Form1.cs



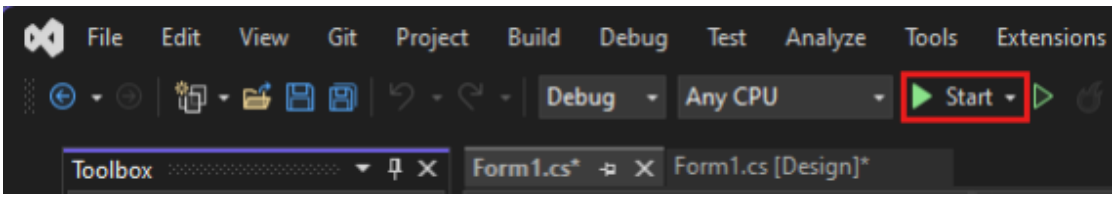
٢. اكتب التعليمات البرمجية داخل حدث الزر، بعد السطر `private void`

```
lblHelloWorld.Text = "Hello World!";
```

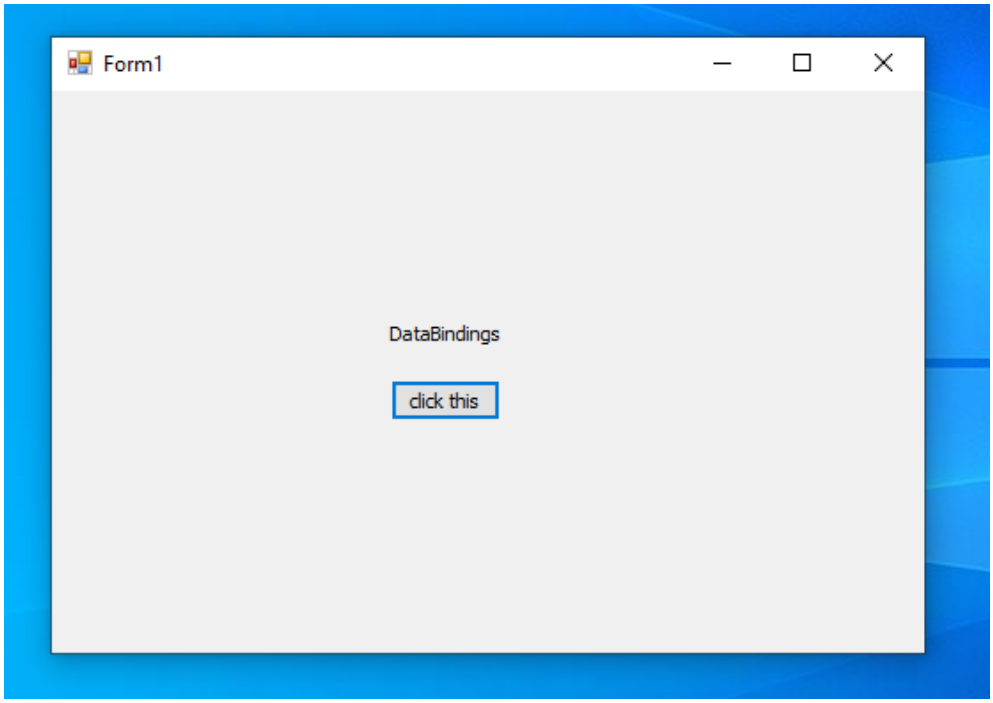


تشغيل التطبيق

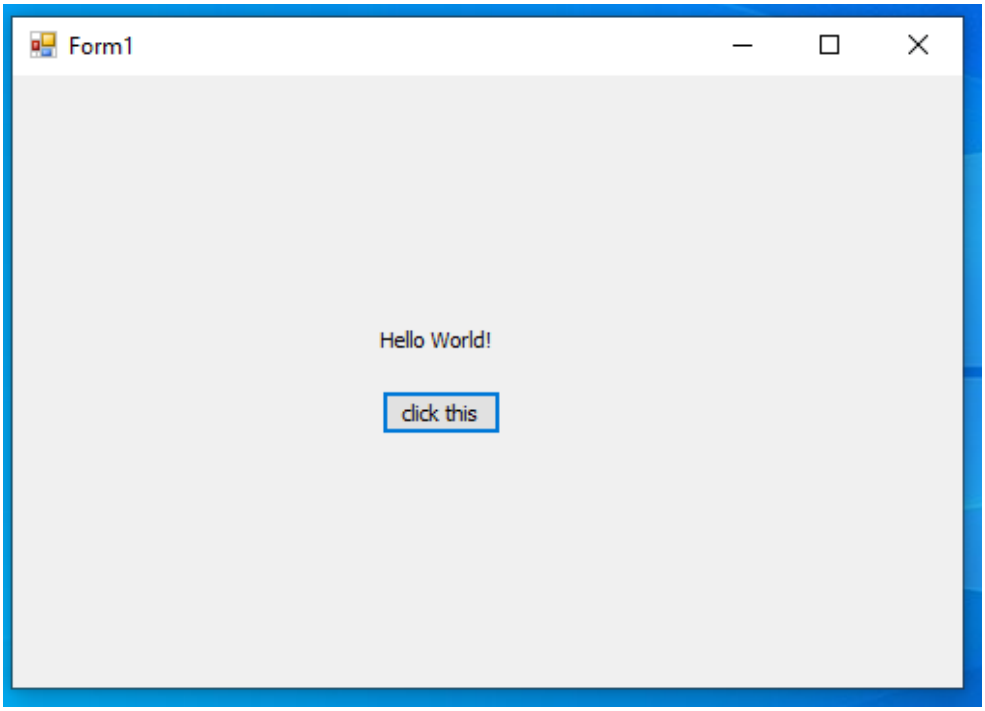
١. حدد زر البدء Start لتشغيل التطبيق



سوف يعمل التطبيق كما هو موضح في الصورة التالية:



٢. اضغط على الزر click this لاحظ أن نص Label تغير إلى Hello World!



٣. لإيقاف تشغيل التطبيق أغلق مربع الحوار Form1

تهانينا على إكمال هذا البرنامج التعليمي. لمعرفة المزيد، تابع البرنامج التعليمي التالي.

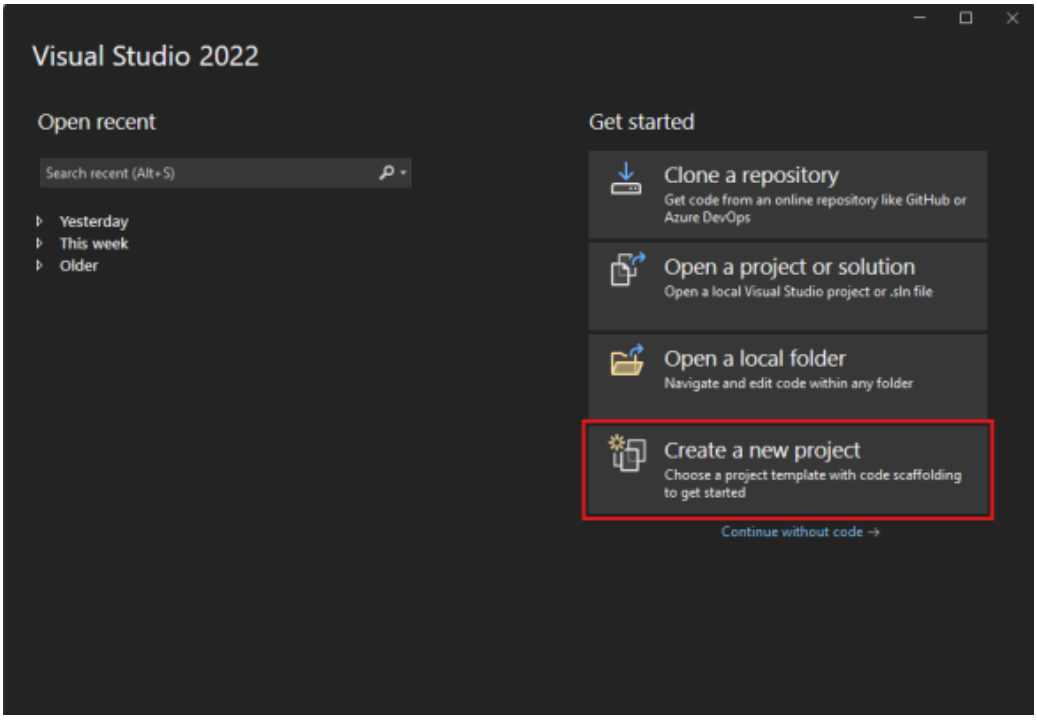
٤ إنشاء تطبيق عارض الصور

إنشاء المشروع

عند إنشاء تطبيق عارض صور، فإن الخطوة الأولى هي إنشاء مشروع Windows Forms App

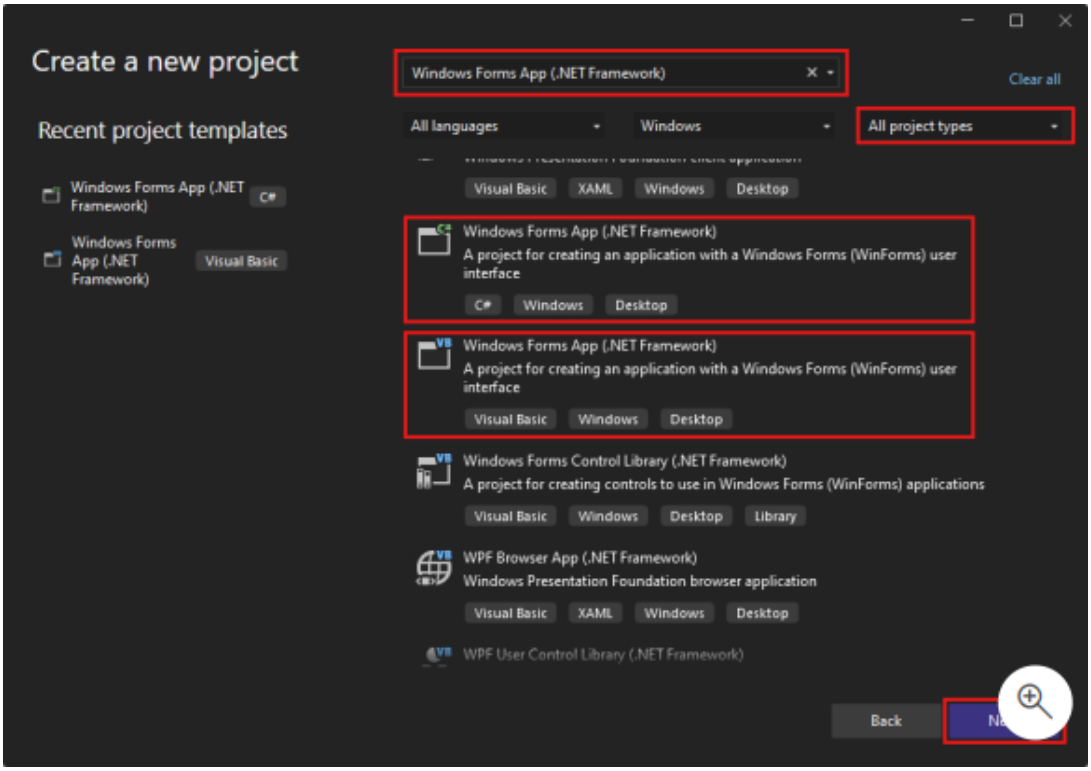
١. افتح Visual Studio

٢. في نافذة البدء، حدد إنشاء مشروع جديد Create a new project

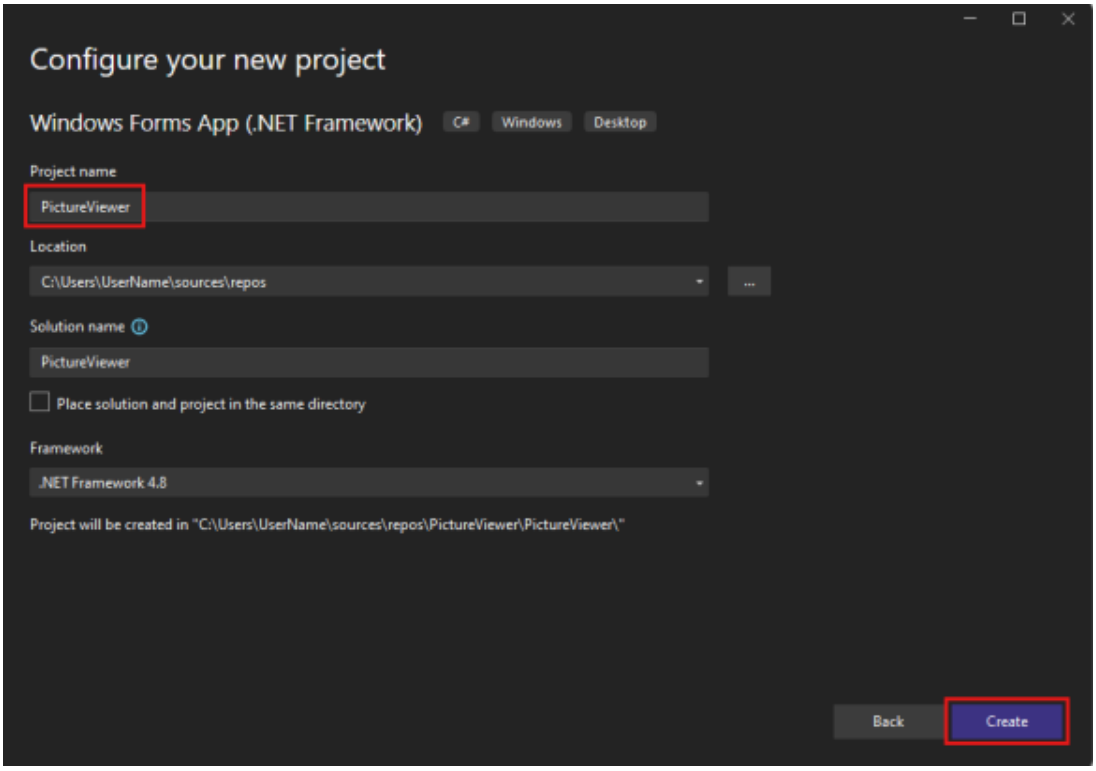


٣. في نافذة Create a new project حدد من قائمة النوع Desktop ثم اختر مشروع Windows Forms App

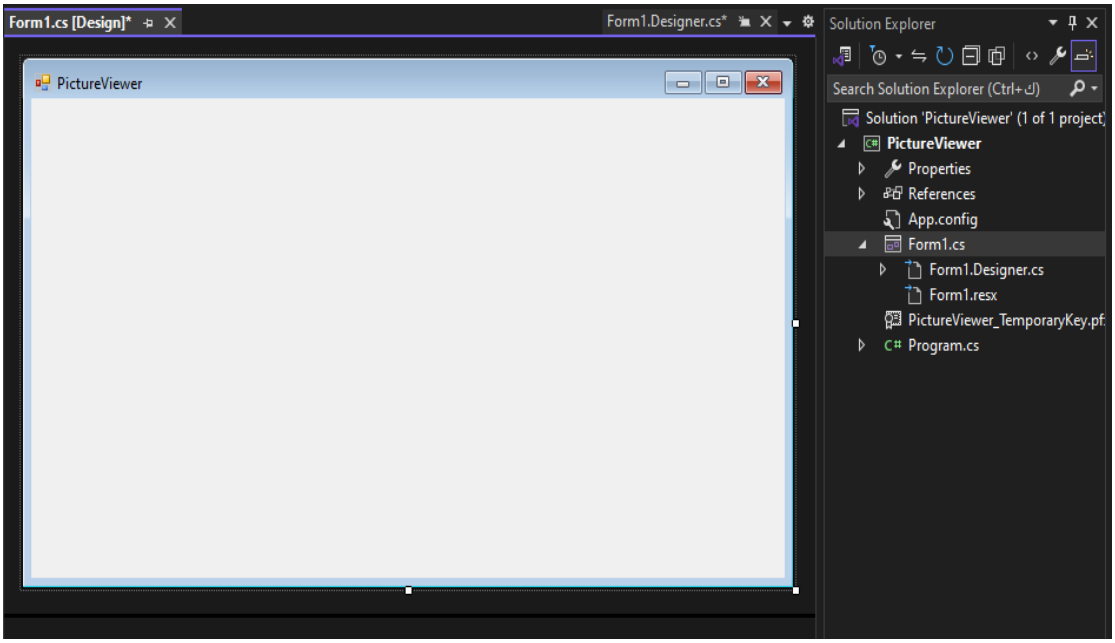
٤. حدد Windows Forms App(.NET Framework) لـ C# اضغط Next



٥. في نافذة تكوين مشروعك الجديد Configure your new project قم بتسمية مشروعك PictureViewer ثم اضغط إنشاء Create



ينشئ Visual Studio قائمة Solution وهي عبارة عن حاوية لجميع المشروعات والملفات التي يحتاجها تطبيقك.



في هذه المرحلة، يعرض Visual Studio نموذجًا فارغًا Windows Form Designer

إضافة عنصر تخطيط TableLayoutPanel

يحتوي تطبيق عارض الصور على مربع صورة ومربع اختيار وأربعة أزرار، يتحكم عنصر التخطيط TableLayoutPanel في موقعها داخل النموذج Form بشكل آلي، يوضح لك هذا القسم كيفية تغيير عنوان النموذج Form وتغيير حجم النموذج وإضافة عنصر تخطيط TableLayoutPanel

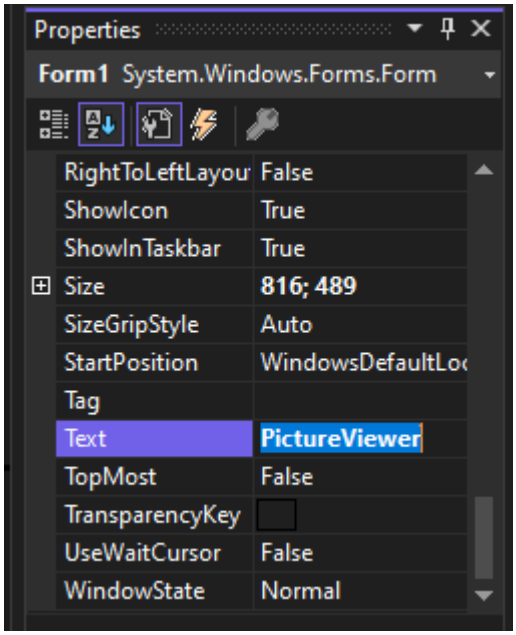
١. في مشروعك، حدد Form1.cs تظهر علامة التبويب Form1.cs[Design]

٢. حدد أي مكان في Form1

٣. تعرض نافذة الخصائص Properties الآن خصائص النموذج، توجد نافذة Properties عادةً في الجزء السفلي الأيمن من Visual Studio يتحكم هذا القسم في خصائص مختلفة، مثل لون المقدمة والخلفية، ونص العنوان الذي يظهر في الجزء العلوي من النموذج، وحجم النموذج.

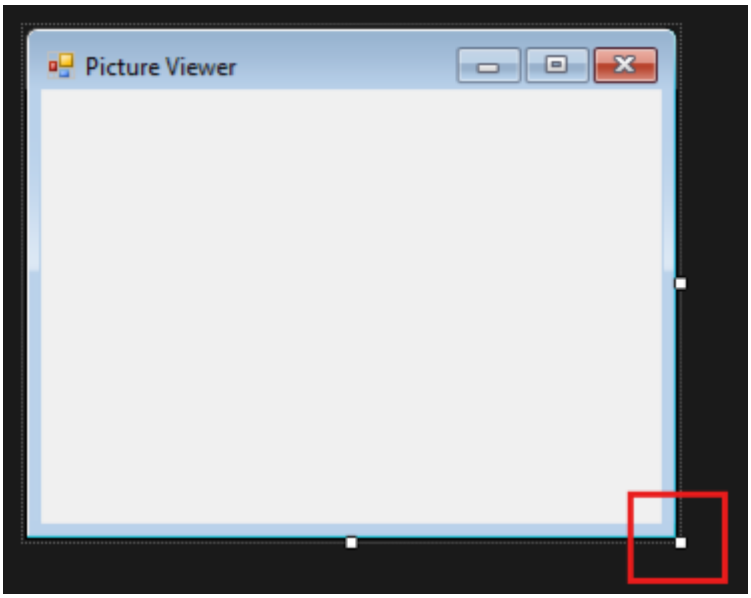
إذا لم تتمكن من رؤية Properties حدد عرض View ثم نافذة Properties أو حدد العنصر أو النموذج ثم اضغط الزر الأيمن ثم اختر Properties

٤. ابحث عن خاصية النص Text ثم أكتب PictureViewer واضغط Enter



يتضمن نموذجك Form الآن نص PictureViewer في شريط العنوان.

٥. حدد Form1 مرة أخرى، حدد مقبض السحب الموجود في أسفل يمين Form المقبض عبارة عن مربع أبيض صغير في الزاوية اليمنى السفلية من Form اسحب المقبض لتغيير حجم النموذج Form بحيث يصبح أعرض وأطول قليلًا.

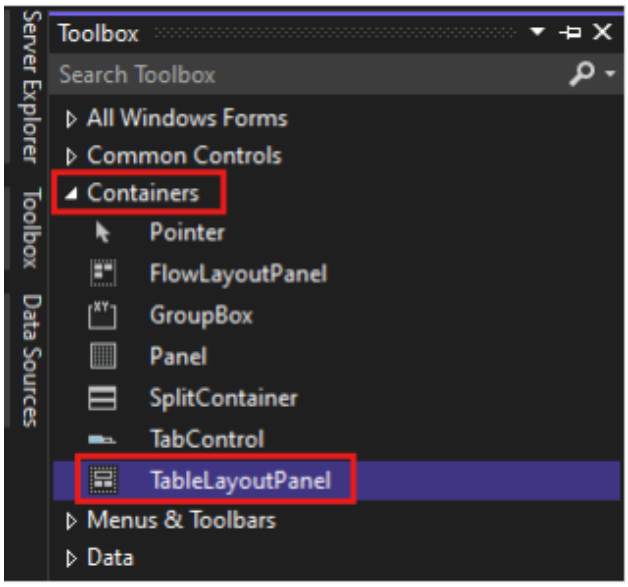


إذا نظرت إلى نافذة الخصائص فستجد أن خاصية الحجم Size مختلفة. يمكنك أيضًا تغيير حجم النموذج عن طريق تغيير خاصية Size

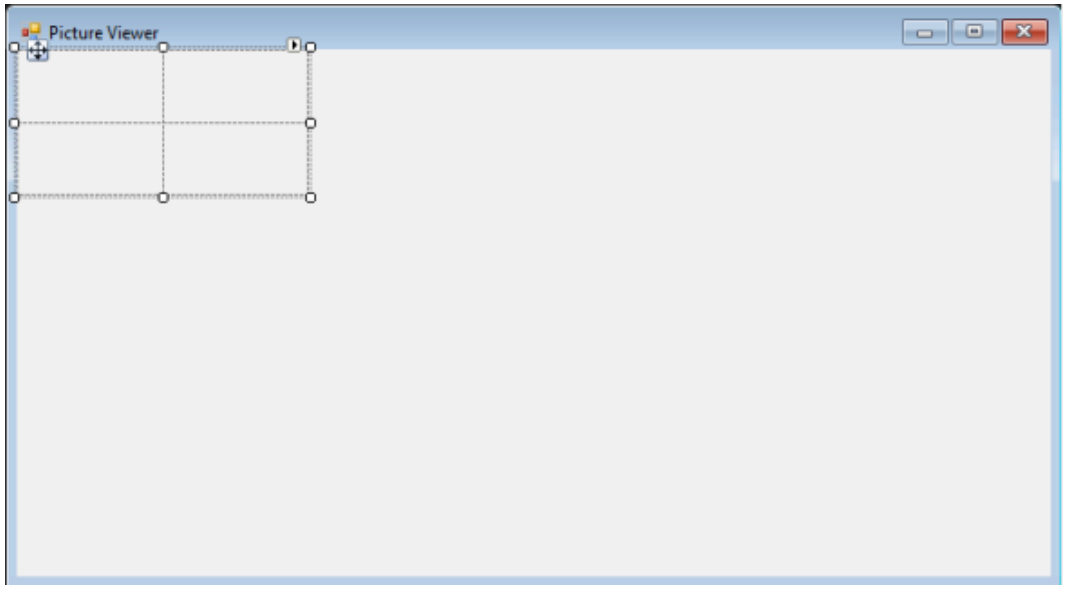
٦. حدد علامة التبويب Toolbox من الجانب الأيسر من Visual Studio

إذا لم تتمكن من رؤيتها حدد View ثم Toolbox أو اضغط **Ctrl + Alt + X**

٧. حدد رمز المثلث الصغير جوار Containers ثم حدد TableLayoutPanel



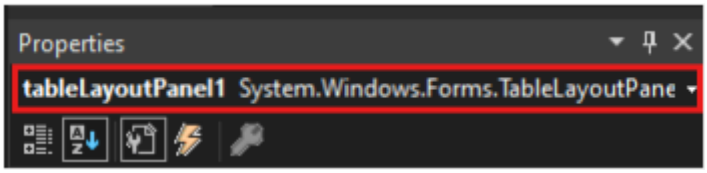
٨. انقر نقرًا مزدوجًا فوق TableLayoutPanel يمكنك أيضًا سحبه إلى Form يظهر عنصر التحكم.



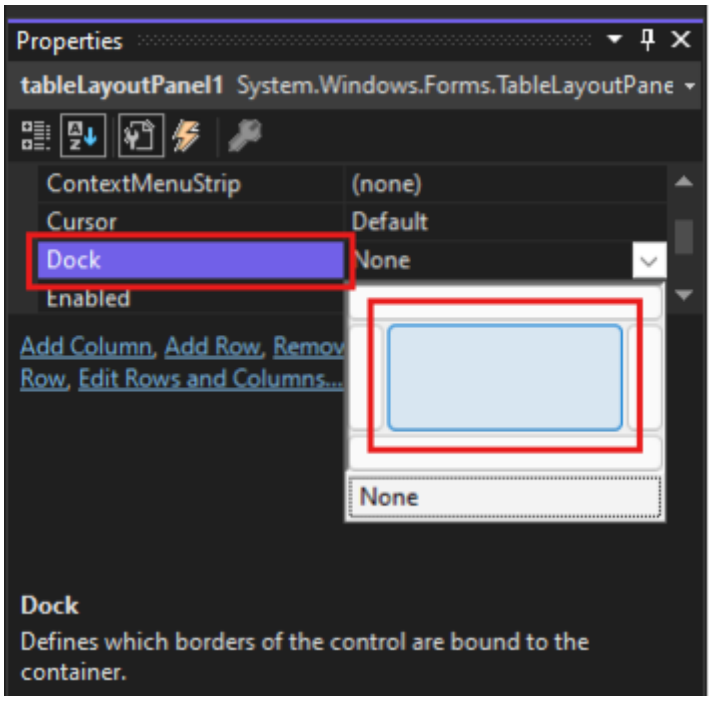
ملحوظة:

إذا ظهرت نافذة داخل النموذج بعنوان TableLayoutPanel Tasks بعد إضافة TableLayoutPanel فحدد أي مكان داخل النموذج لإغلاقها.

٩. حدد TableLayoutPanel1 يمكنك التحقق من عنصر التحكم المحدد من خلال النظر إلى نافذة الخصائص Properties

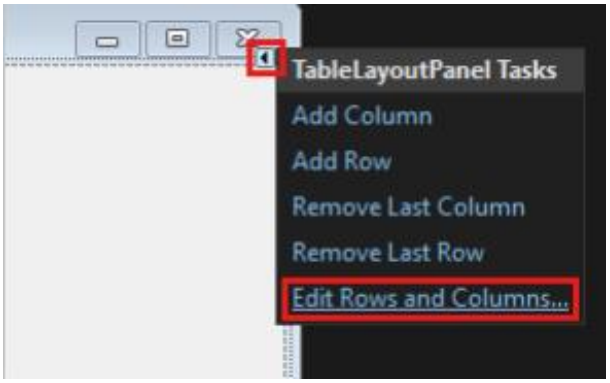


١٠. مع تحديد TableLayoutPanel1 ابحث عن خاصية Dock التي لها القيمة None ثم حدد Fill وهو الزر الكبير الموجود في منتصف القائمة المنسدلة.



الآن تغير حجم TableLayoutPanel ليناسب حجم Form بالكامل، إذا قمت بتغيير حجم Form مرة أخرى يتغير حجم TableLayoutPanel أيضًا ليناسب الحجم الجديد ل Form

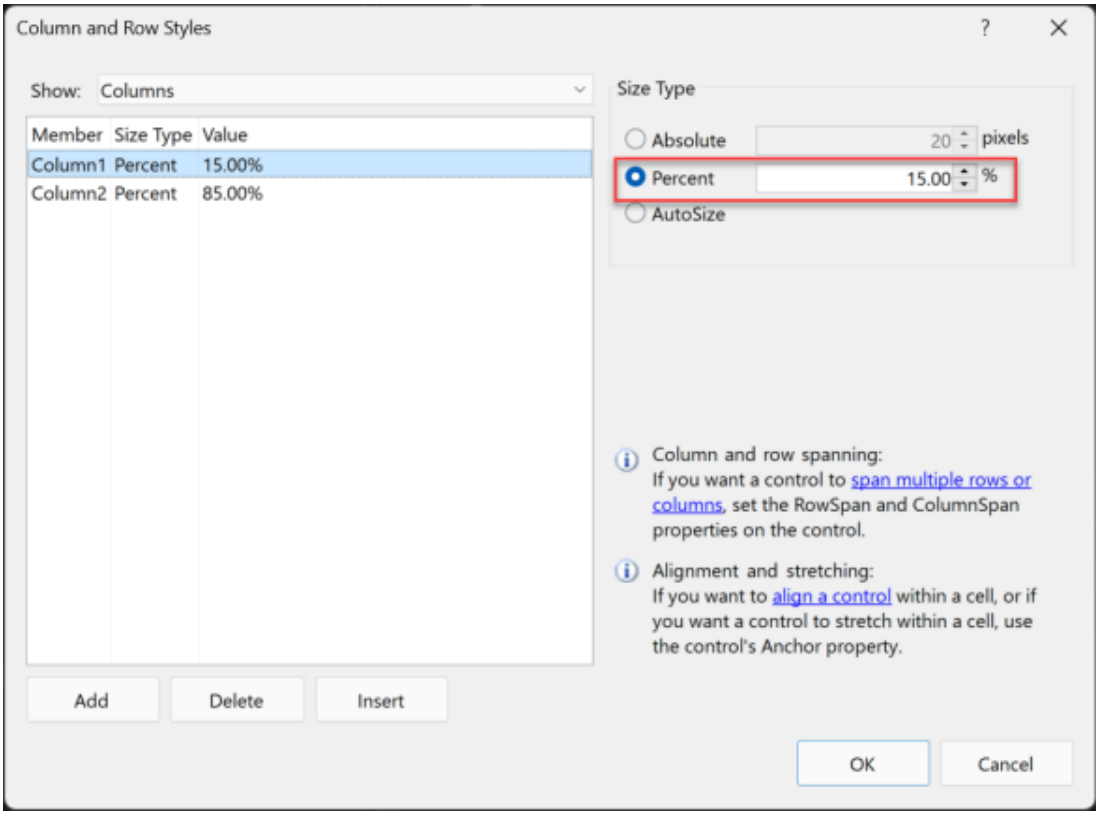
١١. في الزاوية اليمنى العليا ل TableLayoutPanel يوجد زر مثلث أسود صغير، حدد المثلث لعرض قائمة مهام التحكم.



١٢. لعرض مربع الحوار أنماط الأعمدة والصفوف Column and Row Styles حدد تحرير الصفوف والأعمدة Edit Rows and Columns

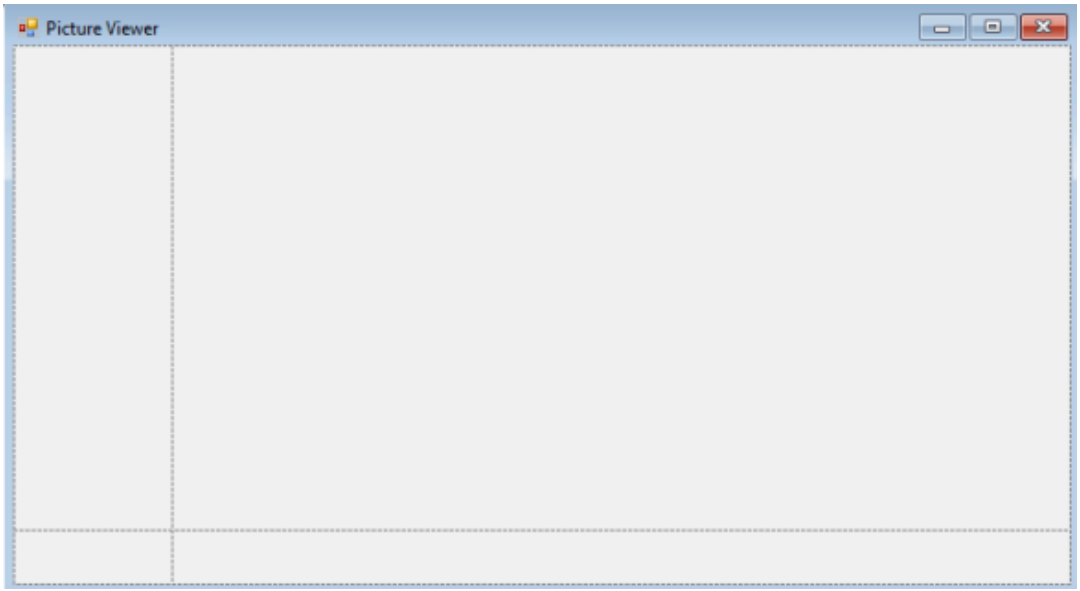
١٣. تأكد من تحديد خيار النسبة المئوية، حدد العمود١ Column1 واضبط حجمه على 15 بالمائة.

١٤. حدد العمود٢ Column2 وقم بضبطه على 85 بالمائة.



١٥. في أعلى مربع الحوار أنماط الأعمدة والصفوف Column and Row Styles حدد الصفوف Rows اضبط الصف١ Row1 على 90 بالمائة والصف٢ Row2 على 10 بالمائة، اضغط موافق OK لحفظ التغييرات.

تحتوي لوحة TableLayoutPanel الخاصة بك الآن على صف علوي كبير، وصف سفلي صغير، وعمود أيسر صغير، وعمود أيمن كبير.



تم إكمال تخطيطك.

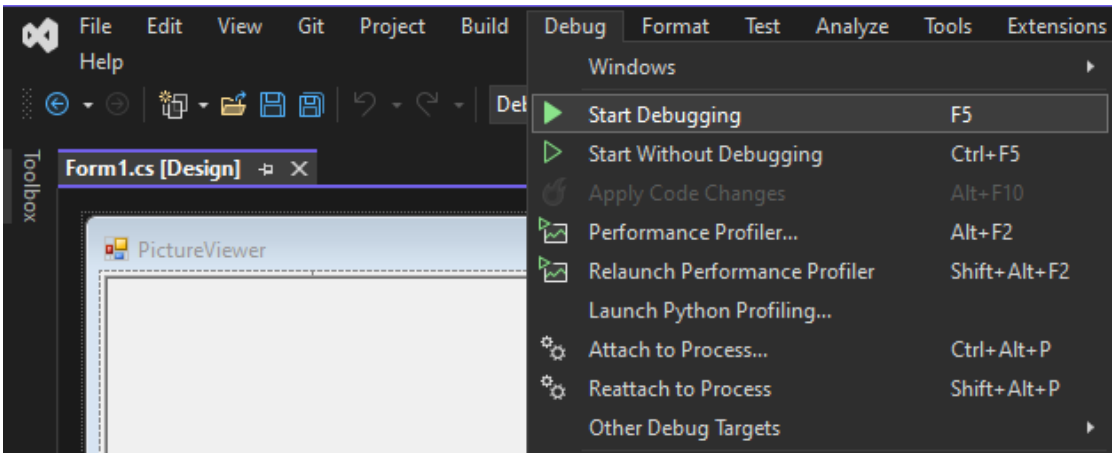
قم بتشغيل تطبيقك

عند إنشاء مشروع Windows Forms App فإنك تقوم بإنشاء برنامج يتم تشغيله، في هذه المرحلة، لا يقوم تطبيق PictureViewer بالكثير، يعرض نافذة فارغة تعرض في شريط العنوان "PictureViewer"

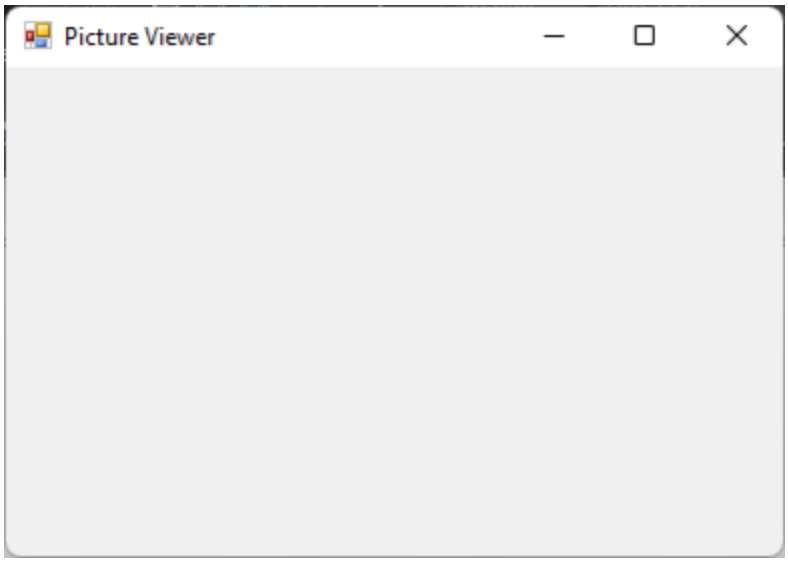
لتشغيل التطبيق، اتبع الخطوات التالية

١. استخدم إحدى الطرق التالية:

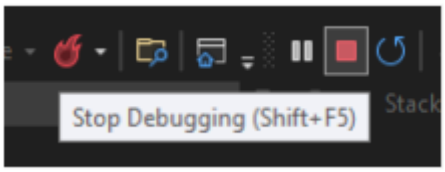
- حدد مفتاح F5
- في شريط القائمة، حدد تصحيح الأخطاء Debug بدء التصحيح Start Debugging
- على شريط الأدوات، حدد زر البدء Start



قم بتشغيل التطبيق الخاص بك، تظهر نافذة بعنوان PictureViewer



ألق نظرة على شريط أدوات Visual Studio IDE تظهر أزرار إضافية على شريط الأدوات عند تشغيل التطبيق، تتيح لك هذه الأزرار القيام بأشياء مثل إيقاف تشغيل التطبيق وبدء تشغيله، وتساعدك على تعقب أي أخطاء.



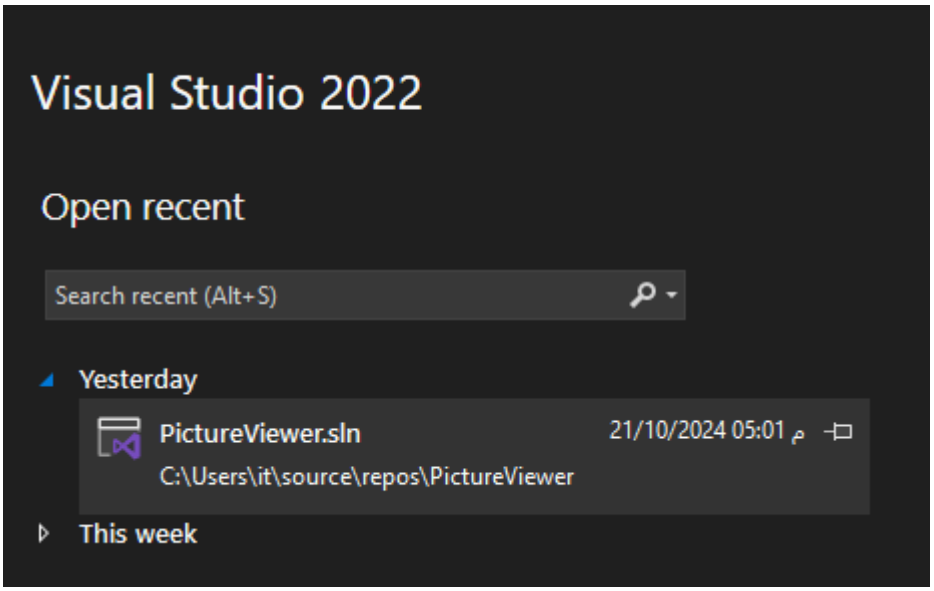
٢. استخدم إحدى الطرق التالية لإيقاف تطبيقك:

- على شريط الأدوات، حدد زر إيقاف التصحيح Stop Debugging
- في شريط القائمة، حدد تصحيح الأخطاء Debug إيقاف التصحيح Stop Debugging
- من لوحة المفاتيح، أدخل **Shift + F5**
- في الزاوية العلوية من نافذة Picture Viewer حدد **X**

إضافة عناصر التحكم

يستخدم تطبيق Picture Viewer عنصر تحكم PictureBox لعرض صورة، ويستخدم العديد من الأزرار لإدارة الصورة والخلفية وإغلاق التطبيق. يمكنك إضافة عنصر PictureBox من Toolbox

١. افتح Visual Studio يظهر مشروع Picture Viewer الخاص بك ضمن فتح الأخير



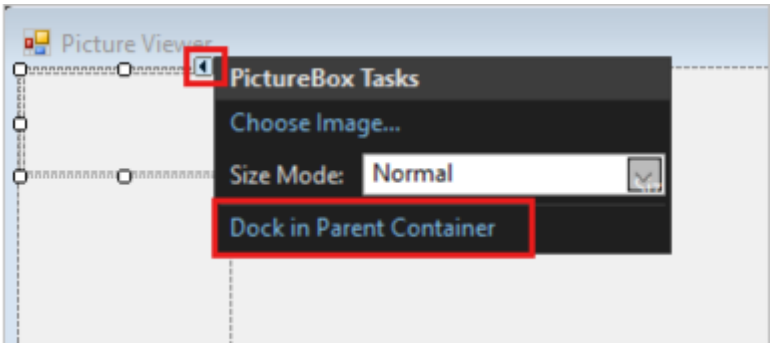
أو اضغط الزر الأيمن للماوس، فوق ايقونة اختصار Visual Studio

٢. في Windows Form Designer اختر TableLayoutPanel الذي أضفته في الدرس السابق، حدد نافذة الخصائص Properties للعنصر TableLayoutPanel1

٣. على الجانب الأيسر من Visual Studio IDE حدد علامة التبويب Toolbox قم بتوسيع القائمة Common Controls او من عناصر التحكم الشائعة.

٤. انقر نقرًا مزدوجًا فوق العنصر PictureBox لإضافة العنصر إلى الخلية الفارغة الأولى من TableLayoutPanel أو اسحب عنصر PictureBox إليها.

٥. اختر عنصر التحكم PictureBox لتحديده، ثم حدد المثلث الأسود، لعرض قائمة المهام الخاصة به.



٦. حدد Dock in Parent Container الذي يقوم بتعيين خاصية Dock إلى Fill يمكنك أيضًا رؤية هذه القيمة في نافذة Properties

٧. في نافذة الخصائص لـ PictureBox اضبط خاصية ColumnSpan إلى 2 تملأ PictureBox الآن كلا العمودين.

٨. قم بتعيين خاصية BorderStyle إلى Fixed3D

٩. في Windows Form Designer اختر TableLayoutPanel ثم من Toolbox انقر نقرًا مزدوجًا فوق العنصر CheckBox لإضافة العنصر إلى الخلية اليسرى السفلية من TableLayoutPanel

١٠. اختر خاصية النص Text وأكتب Stretch

إضافة أزرار إلى لوحة التخطيط flowLayoutPanel

حتى الآن، قمت بإضافة عناصر تحكم إلى TableLayoutPanel توضح لك هذه الخطوات التالية، كيفية إضافة أربعة أزرار إلى لوحة تخطيط جديدة، في TableLayoutPanel

FlowLayoutPanel عبارة عن حاوية تنظيم، تقوم بترتيب عناصر التحكم المختلفة الأخرى، في صف واحد بالتتابع، واحًدا تلو الآخر.

١. في Windows Form Designer اختر TableLayoutPanel ثم من Toolbox > Containers انقر نقرًا مزدوجًا فوق العنصر flowLayoutPanel لإضافة العنصر الجديد إلى الخلية الأخيرة في TableLayoutPanel

٢. اختر FlowLayoutPanel قم بتعيين خاصية Dock إلى Fill يمكنك ضبط هذه الخاصية عن طريق تحديد المثلث الأسود، ثم تحديد Dock

٣. اختر FlowLayoutPanel ثم افتح Toolbox > Common Controls انقر نقرًا مزدوجًا فوق العنصر Button لإضافة عنصر تحكم زر button1

٤. انقر نقرًا مزدوجًا فوق العنصر Button لإضافة عنصر زر آخر يسمي button2

٥. أضف زرين آخرين بهذه الطريقة button4 – button3 يمكنك استخدام النسخ واللصق.

٦. اختر الزر الأول button1 قم بضبط خاصية النص Text على Show a picture

٧. اختر الزر button2 قم بضبط خاصية النص Text على Clear the picture

الزر button3 قم بضبط خاصية النص Text على Set the background color

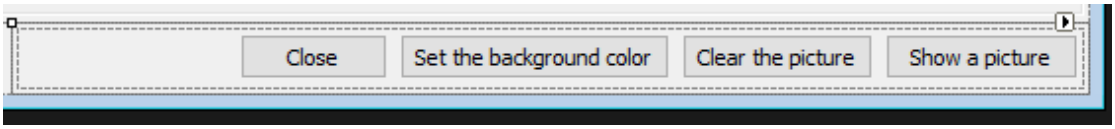
الزر button4 قم بضبط خاصية النص Text على Close

٨. لتغيير حجم الأزرار وترتيبها، حدد FlowLayoutPanel اضبط خاصية FlowDirection على RightToLeft

يجب أن تتم محاذاة الأزرار على الجانب الأيمن من الخلية، ويجب عكس ترتيبها بحيث يكون الزر Show a picture على يمين الخلية، يمكنك سحب الأزرار حول FlowLayoutPanel بأي ترتيب.

٩. اختر زر الإغلاق Close لتحديده، ثم لتحديد بقية الأزرار في نفس الوقت، اضغط مع الاستمرار على Ctrl واختر بقية الأزرار.

١٠. افتح نافذة الخصائص واضبط خاصية AutoSize على True يتم تغيير حجم الأزرار لتناسب النص الموجود بها.



يمكنك تشغيل البرنامج لمعرفة شكل عناصر التحكم، اضغط المفتاح F5 أو على شريط الأدوات، حدد زر البدء Start الأزرار التي أضفتها لا تفعل أي شيء حتى الآن.

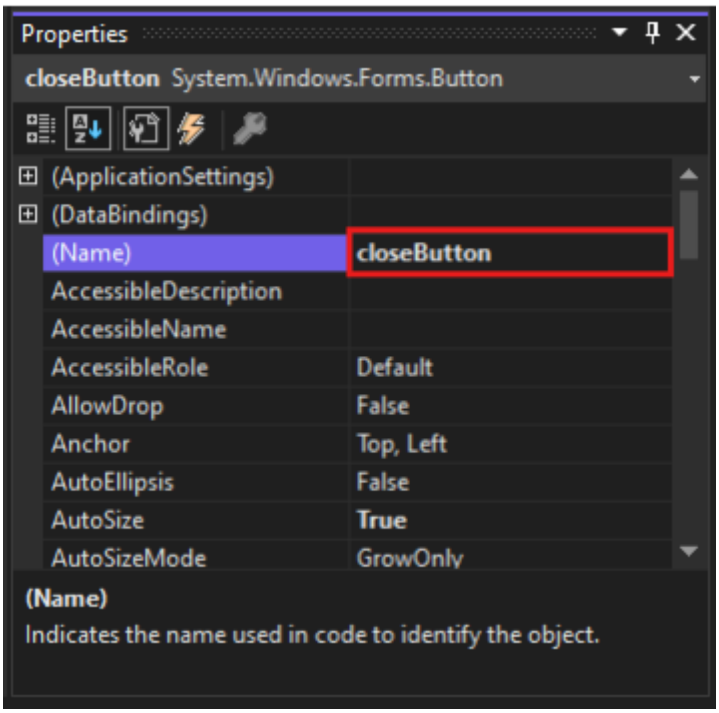
تغيير أسماء عناصر التحكم

يوجد أربعة أزرار في النموذج الخاص بك، تسمى button1 button2 button3 button4 استخدم خطوات اختيار الأسماء في C# لمنحها أسماء أكثر إفادة.

إذا كنت لا تزال تحدد كل الأزرار، اضغط Esc لإلغاء تحديد الكل.

١. حدد زر الإغلاق Close

٢. في نافذة الخصائص Properties ابحث عن خاصية الاسم Name قم بتغيير الاسم إلى closeButton



لا يقبل Visual Studio IDE الأسماء التي تحتوي على مسافات.

٣. قم بإعادة تسمية الأزرار الثلاثة الأخرى إلى:

زر Show a picture إلى اسم showButton

زر Clear the picture إلى اسم clearButton

زر Set the background color إلى اسم backGroundButton

يمكنك تغيير اسم أي عنصر تحكم، مثل TableLayoutPanel أو CheckBox

إضافة مكونات الحوار

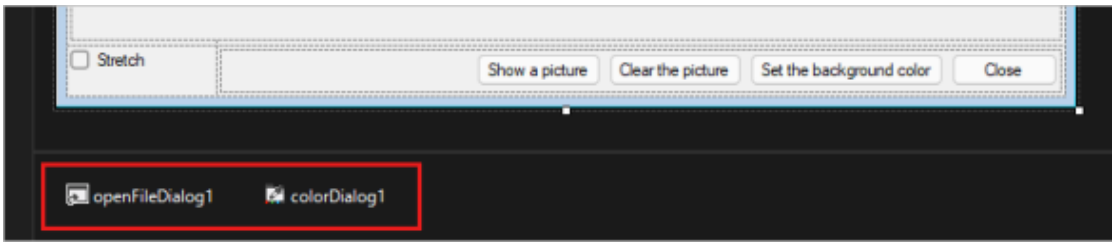
يمكن لتطبيقك فتح ملفات الصور، واختيار لون الخلفية باستخدام المكونات، المكون يشبه عنصر التحكم، يمكنك استخدام صندوق الأدوات Toolbox لإضافة مكون إلى النموذج Form وتعيين خصائصه، باستخدام نافذة الخصائص Properties

على عكس عنصر التحكم، فإن إضافة مكون إلى النموذج لا يضيف عنصرًا مرئيًا، بل يوفر سلوكيات معينة، يمكنك تشغيلها باستخدام التعليمات البرمجية، على سبيل المثال OpenFileDialog إنه مكون يفتح مربع حوار.

في هذه القسم يمكنك إضافة مكوني OpenFileDialog و ColorDialog إلى النموذج Form

١. في Windows Form Designer من Toolbox > Dialogs انقر نقرًا مزدوجًا فوق المكون OpenFileDialog والمكون ColorDialog

٢. تظهر المكونات كأيقونات أسفل Form Designer



٣. اختر أيقونة openFileDialog1 وقم بتعيين خاصيتين:

- خاصية التصفية Filter اضبطها إلى القيمة:

JPEG Files (*.jpg)|*.jpg|PNG Files (*.png)|*.png|BMP Files (*.bmp)|*.bmp| all (*.*)|

تحدد إعدادات خاصية التصفية الأنواع التي يعرضها مربع الحوار، تحديد صورة.

- خاصية العنوان Text إلى القيمة Open picture

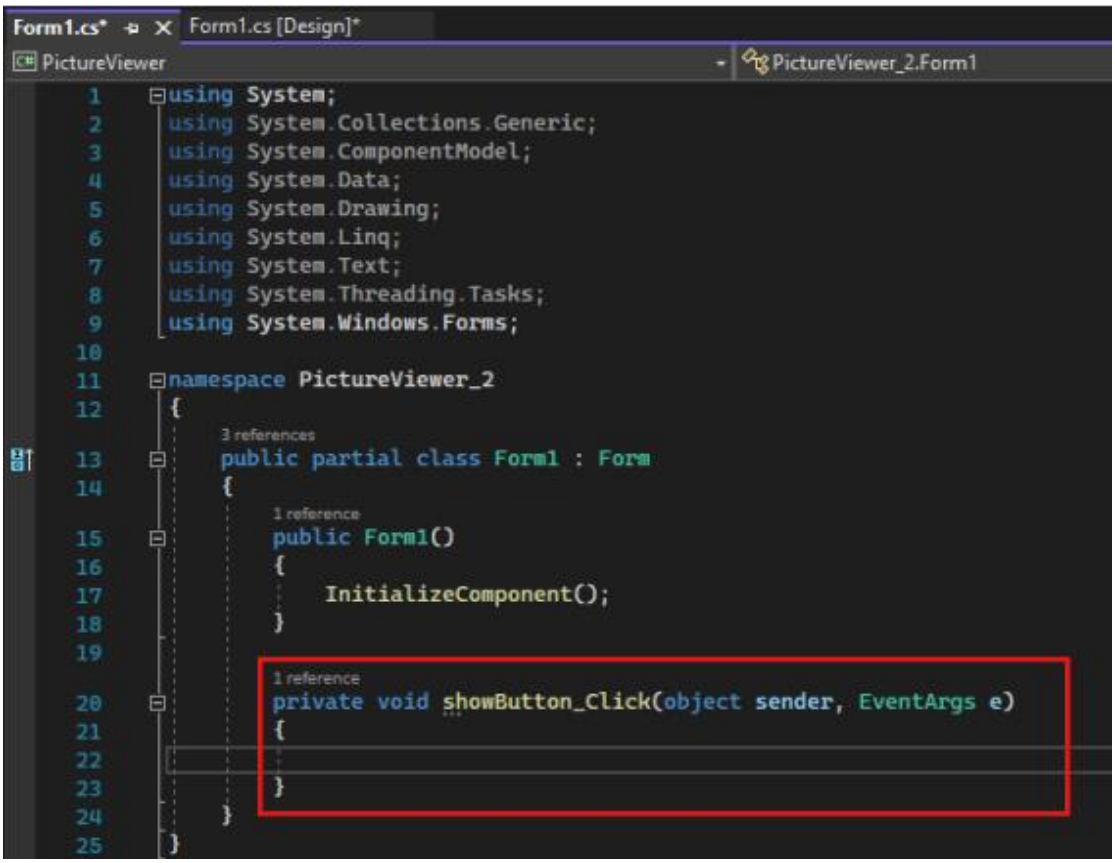
أضف معالجات الأحداث إلى عناصر التحكم

في هذا القسم، أضف معالجات الأحداث لعناصر التحكم التي أضفتها، يستدعي تطبيقك معالج حدث، عند حدوث إجراء، مثل الضغط فوق زر.

١. في Form Designer انقر نقرًا مزدوجًا فوق زر Show a picture يمكنك بدلًا من ذلك، تحديد الزر Show a picture في Form ثم الضغط على Enter

٢. يفتح برنامج Visual Studio IDE بالنسبة ل C# علامة تبويب في النافذة الرئيسية، تسمي Form1.cs

تعرض هذه العلامة ملف التعليمات البرمجية خلف النموذج Form



٣. ركز على هذا الجزء من التعليمات البرمجية.

سي شارب

```
private void showButton_Click(object sender, EventArgs e)
{
}
```

٤. اختر علامة التبويب Form1.cs[Design] ثم انقر نقرًا مزدوجًا فوق الزر Clear the picture لفتح معالج التعليمات البرمجية الخاصة بالزر، كرر ذلك للزرين المتبقيين، في كل مرة يضيف Visual Studio IDE أسلوب جديد Method إلى ملف التعليمات البرمجية ل Form

٥. انقر نقرًا مزدوجًا فوق CheckBox لفتح معالج التعليمات الخاص بالعنصر، عند تحديد أو إلغاء تحديد، مربع الاختيار CheckBox يتم استدعاء الأسلوب checkBox1_CheckedChanged()

يعرض المقطع التالي التعليمات الجديدة التي تراها في المحرر.

سي شارب

```
private void clearButton_Click(object sender, EventArgs e)
{
}

private void backgroundButton_Click(object sender, EventArgs e)
{
}

private void closeButton_Click(object sender, EventArgs e)
{
}

private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
}
```

يمكن أن يكون للأساليب، بما في ذلك معالجات الأحداث، أي اسم تر يده، عندما تضيف معالج حدث، فإنه ينشئ اسمًا بناءً على اسم عنصر التحكم والحدث الذي تتم معالجته، باستخدام IDE

على سبيل المثال، يتم استدعاء الحدث Click الخاص بالزر المسمى showButton ينفذ عن الضغط فوق الزر showButton_Click()

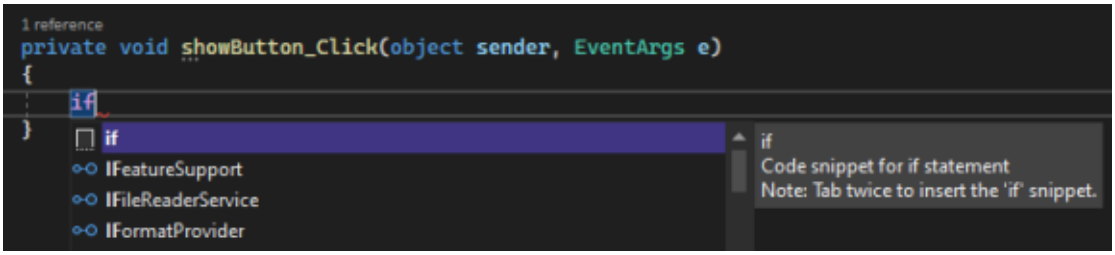
اكتب التعليمات البرمجية لفتح مربع الحوار OpenFileDialog

يستخدم زر Show a picture مكون OpenFileDialog يضيف هذا الإجراء التعليمات لاستدعاء المكون، وعرض ملف صورة.

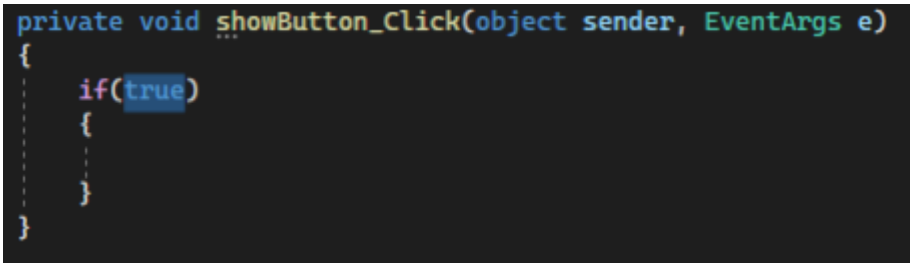
توفر بيئة التطوير المتكاملة Visual Studio أداة قوية تسمى IntelliSense تقترح أداة IntelliSense أثناء الكتابة، تعليمات برمجية محتملة.

١. في Form Designer انقر نقرًا مزدوجًا فوق الزر Show a picture لفتح أسلوب showButton_Click() تأكد أن المؤشر داخل الأسلوب.

٢. اكتب حرف i في السطر الفارغ بين القوسين {} لفتح نافذة IntelliSense



٣. يجب أن تبرز نافذة IntelliSense وتظهر الكلمة الأساسية if اضغط Tab مرتين لإكمال جملة if تلقائيًا.



٤. حدد true وأكتب op لاستبدالها، ستعرض نافذة IntelliSense المكون openFileDialog1

٥. اختر openFileDialog1

٦. بعد openFileDialog1 اكتب نقطة (.) يوفر لك IntelliSense مباشرًة جميع خصائص والأساليب ل OpenFileDialog ابدأ في كتابة ShowDialog()

٧. تأكد من إضافة الأقواس () يجب أن تكون التعليمة مثل openFileDialog1.ShowDialog()

٨. أضغط مسافة، ثم بالنسبة إلى لغة C# أضف علامتي يساوي == بالنسبة إلى لغة Visual Basic أضف علامة يساوي = واحدة.

٩. أضف مسافة أخرى، استخدم IntelliSense لاختيار DialogResult

١٠. اكتب نقطة . لفتح قيم DialogResult ثم أكتب OK أو اختار من نافذة IntelliSense

يجب أن يكون السطر الأول من التعليمات البرمجية كاملًا، مثل:

```
if (openFileDialog1.ShowDialog() == DialogResult.OK)
```

١١. أضف السطر التالي من التعليمات البرمجية:

```
pictureBox1.Load(openFileDialog1.FileName);
```

يمكنك نسخ ولصق أو استخدام IntelliSense يجب أن يكون الأسلوب showButton_Click() مشابه للتعليمات التالية:

```
private void showButton_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        pictureBox1.Load(openFileDialog1.FileName);
    }
}
```

١٢. أضف التعليق التالي إلى تعليماتك:

```
// Show the Open File dialog. If the user clicks OK, load the
// picture that the user chose.
```

ملحوظة:

من الأفضل دائمًا التعليق على تعليماتك البرمجية، بما يتناسب مع قواعد التعليق التي تعلمناها في الجزء الأول، حيث تسهل التعليقات فهم التعليمات وصيانتها.

كتابة التعليمات لعناصر التحكم الأخرى

يفتح Picture Viewer مربع الحوار `OpenFileDialog` إذا قمت بتشغيل التطبيق الآن، فيمكنك تحديد زر Show a picture حيث يمكنك تحديد صورة لعرضها.

في هذا القسم، إضافة التعليمات الخاصة بمعالجات الأحداث الأخرى.

١. في Form Designer انقر نقرًا مزدوجًا فوق الزر Clear the picture لفتح أسلوب clearButton_Click() وأضف التعليمات التالية:

```
private void clearButton_Click(object sender, EventArgs e)
{
// Clear the picture.
pictureBox1.Image = null;
}
```

٢. انقر نقرًا مزدوجًا فوق الزر Set the background color لفتح أسلوب backGroundButton_Click() وأضف التعليمات التالية:

```
private void backgroundButton_Click(object sender,
EventArgs e)
{
// Show the color dialog box. If the user clicks OK, change the
// PictureBox control's background to the color the user chose.
 if (colorDialog1.ShowDialog() == DialogResult.OK)
        pictureBox1.BackColor = colorDialog1.Color;
}
```

٣. انقر نقرًا مزدوجًا فوق الزر Close وأضف التعليمات التالية:

```
private void closeButton_Click(object sender, EventArgs e)
{
    this.Close();
}
```

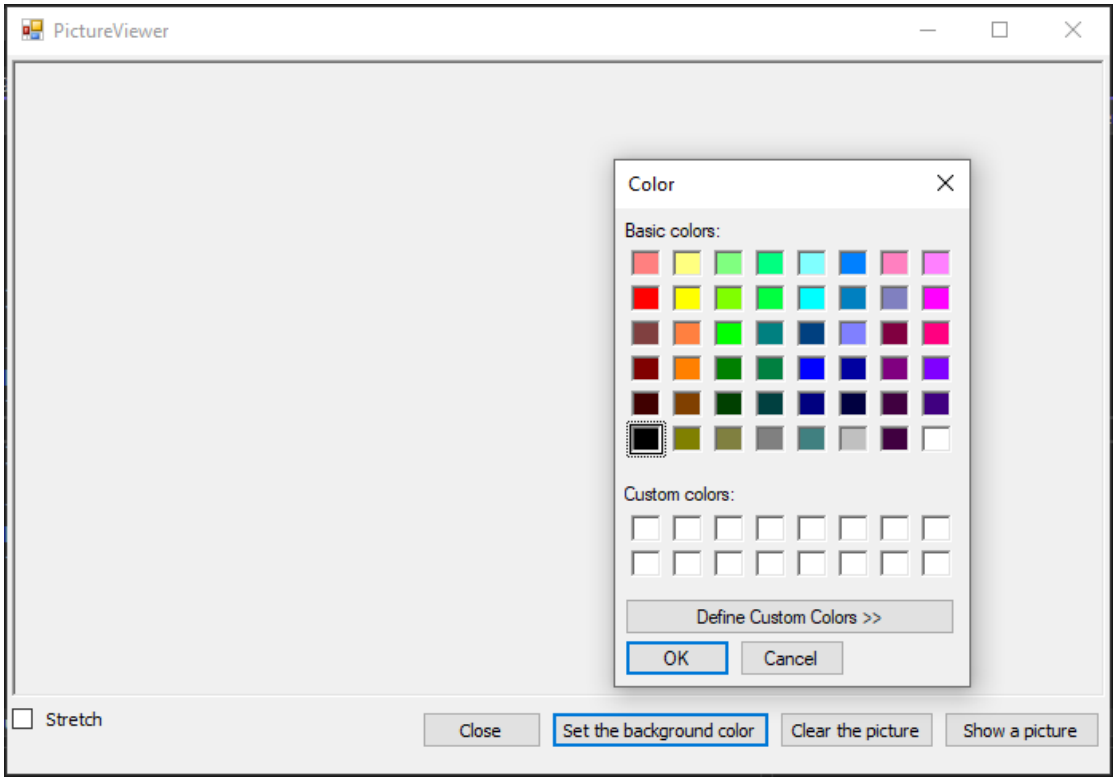
٤. انقر نقرًا مزدوجًا فوق مربع الاختيار Stretch وأضف التعليمات التالية:

```
  // If the user selects the Stretch check box,
  // change the PictureBox's
  // SizeMode property to "Stretch". If the user clears
  // the check box, change it to "Normal".
 if (checkBox1.Checked)
   pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;
   else
     pictureBox1.SizeMode = PictureBoxSizeMode.Normal;
```

تشغيل التطبيق

يمكنك تشغيل تطبيقك في أي وقت أثناء كتابته، بعد إضافة التعليمات السابقة، يكتمل Picture Viewer استخدم إحدى الطرق التي تعلمتها لتشغيل تطبيقك، واختبر كافة عناصر التحكم.

١. تظهر نافذة Picture Viewer



٢. اضغط الزر Set the background color واختر لوًنا لتعيين لون الخلفية.

٣. اضغط الزر Show a picture لعرض صورة.

٤. حدد مربع الاختيار Stretch ثم قم بإلغاء التحديد.

٥. اضغط الزر Clear the picture للتأكد من مسح الشاشة.

٦. اضغط الزر Close للخروج من التطبيق.

٥ إنشاء تطبيق اختبار الرياضيات

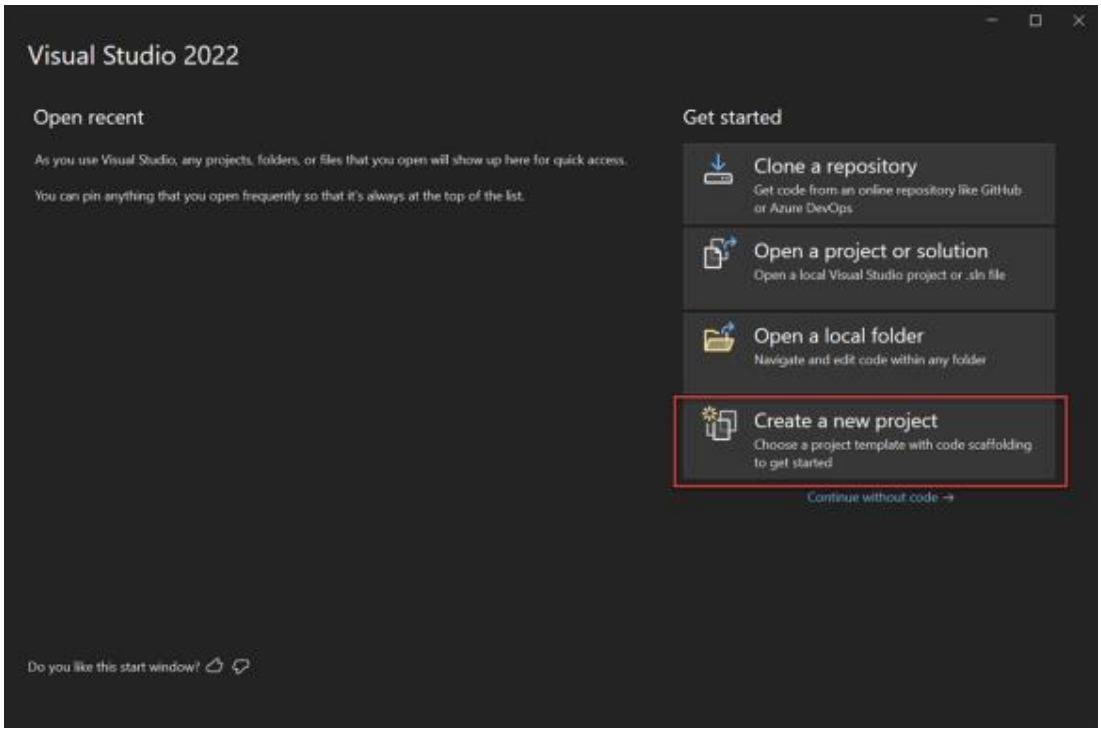
في هذه الوحدة ستقوم بإنشاء اختبار رياضيات، يحتوي الاختبار على أربع مسائل رياضية عشوائية، يحاول المشارك في الاختبار الإجابة عليها خلال فترة زمنية محددة.

إنشاء مشروع Windows Forms App

عند إنشاء اختبار رياضيات، فإن الخطوة الأولى هي إنشاء مشروع Windows Forms App

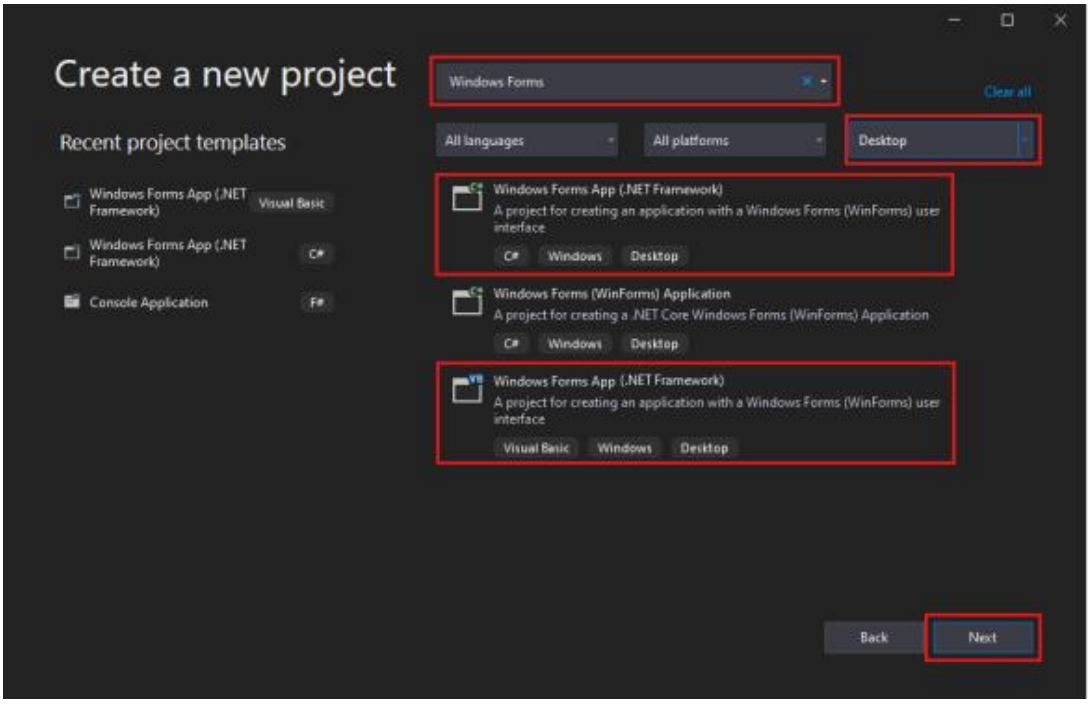
١. فتح Visual Studio

٢. في نافذة البدء، حدد إنشاء مشروع جديد Create a new project

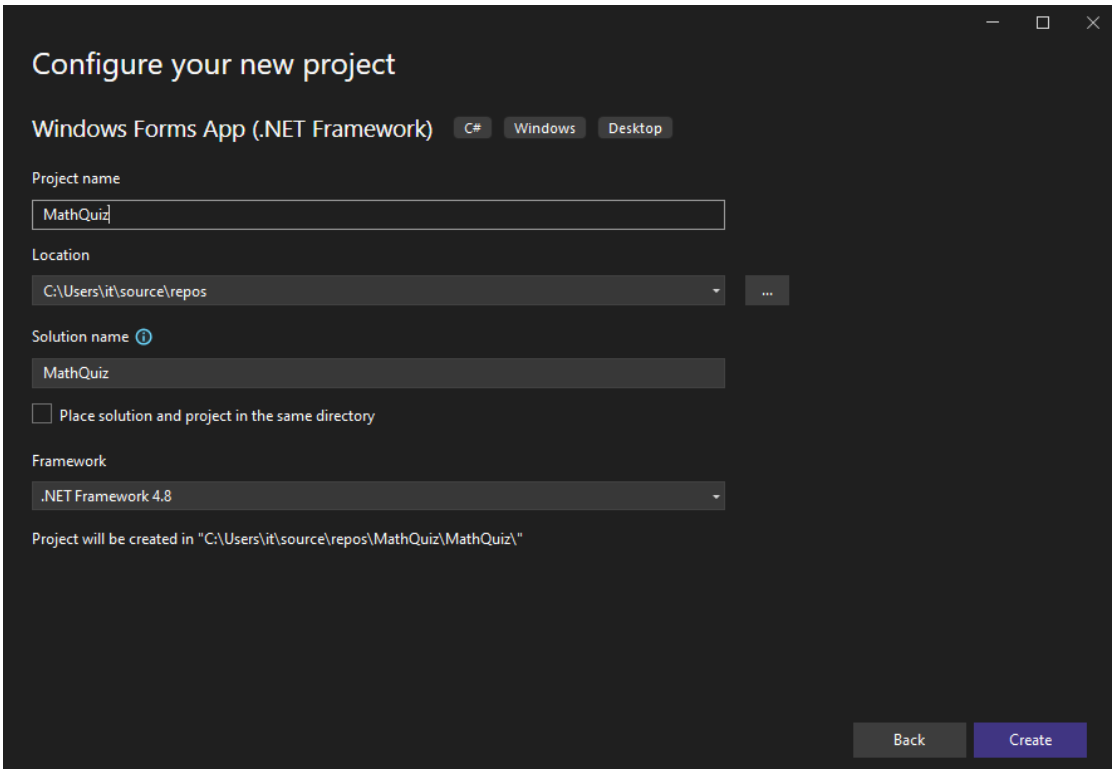


٣. في نافذة Create a new project ابحث عن Windows Forms واختر من قائمة النوع Desktop

٤. حدد قالب تطبيق Windows Forms App(.NET Framework) ل C# ثم اضغط Next



٥. في نافذة تكوين مشروعك الجديد Configure your new project قم بتسمية مشروعك MathQuiz ثم حدد إنشاء Create



ينشئ Visual Studio قائمة Solution وهي عبارة عن حاوية لجميع المشروعات والملفات والحلول التي يحتاجها تطبيقك.

تعيين خصائص النموذج Form

بعد تحديد القالب وتسمية الملف، يفتح لك Visual Studio نموذجًا Form يوضح لك هذا القسم كيفية تغيير بعض خصائص النموذج.

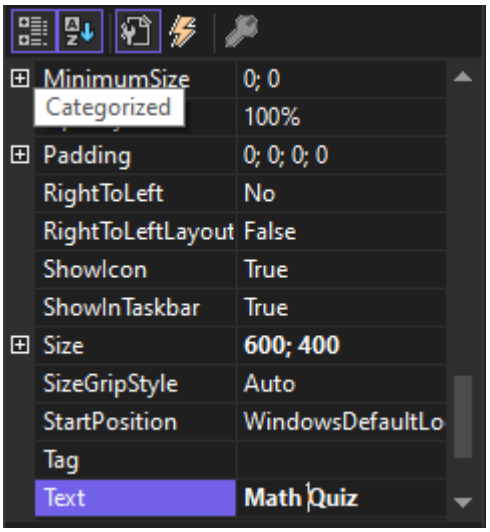
١. في مشروعك، حدد علامة التبويب Form1.cs[Design]

٢. حدد النموذج Form1 تعرض نافذة Properties الآن خصائص Form1

٣. ابحث عن خاصية Text في نافذة Properties قد تحتاج إلى التمرير لأسفل، اعتمادًا على كيفية فرز القائمة، أدخل كلمة Math Quiz لقيمة Text ثم اضغط Enter

ملحوظة

يمكنك عرض الخصائص حسب الفئة أو أبجديًا، استخدم الأزرار الموجودة في نافذة Properties للتبديل بين الفئتين.



يتضمن نموذجك Form الآن نص Math Quiz في شريط العنوان.

٤. عدل حجم Form إلى 500 بكسل عرضًا و400 بكسل طولًا

يمكنك تغيير حجم Form عن طريق سحب الحواف أو سحب مقبض السحب، حتى يظهر الحجم الصحيح كقيمة Size في نافذة الخصائص، مقبض السحب عبارة عن مربع أبيض صغير في الزاوية اليمنى السفلية من Form يمكنك أيضًا تغيير حجم Form عن طريق تغيير قيم خاصية Size

٥. قم بتغيير قيمة الخاصية FormBorderStyle إلى Fixed3D وقم بتعيين الخاصية MaximizeBox إلى False تمنع هذه القيم المشاركين في الاختبار من تغيير حجم Form

إنشاء مربع، الوقت المتبقي

يحتوي تطبيق Math Quiz على مربع في الزاوية اليمنى العليا، يظهر هذا المربع عدد الثواني المتبقية في الاختبار، يوضح لك هذا القسم كيفية استخدام تسمية لهذا المربع.

١. حدد Toolbox "صندوق الأدوات" إذا لم تتمكن من رؤية خيار Toolbox فيمكنك فتحه من شريط القائمة، حدد View اختر Toolbox أو اضغط **Ctrl + Alt + X**

٢. قم بتحديد عنصر التحكم Label "ملصق تسمية" ثم اسحبه إلى النموذج.

٣. في نافذة الخصائص Properties قم بتعيين الخصائص التالية ل Label

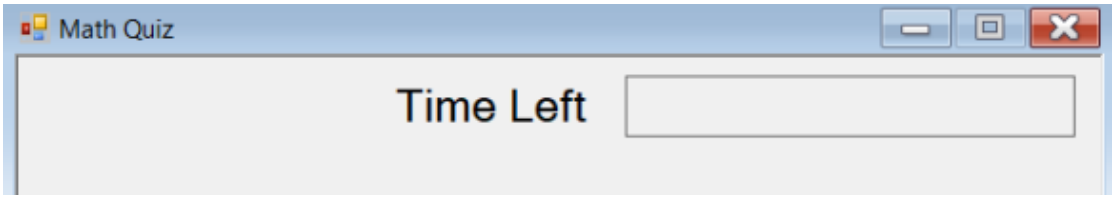
- قم بضبط الاسم Name على timeLabel
- قم بتغيير AutoSize إلى False حتى تتمكن من تغيير حجم العنصر
- قم بتغيير BorderStyle إلى FixedSingle لرسم خط حول العنصر
- اضبط الحجم Size على 200, 30
- حدد خاصية النص Text ثم أمسح قيمة النص
- ثم قم بضبط حجم خاصية الخط على 15.75

٤. انقل Label إلى الزاوية اليمنى العليا من النموذج، عندما تظهر خطوط فاصلة زرقاء، استخدمها لتحديد موضع عنصر التحكم في النموذج.

٥. أضف عنصر تحكم Label آخر، ثم اضبط حجم الخط الخاص به على 15.75

٦. قم بضبط خاصية Text الخاصة Label إلى Time Left "الوقت المتبقي"

٧. قم بتحريك Label بحيث يصبح على يسار timeLabel



إضافة عناصر تحكم لمشكلة الجمع

الجزء الأول من الاختبار عبارة عن مسألة جمع، يوضح لك هذا القسم كيفية استخدام العلامات لعرض هذه المسألة.

١. أضف عنصر التحكم Label

٢. من نافذة الخصائص Properties قم بتعيين الخصائص التالية ل Label

- اضبط Text إلى "؟" علامة الاستفهام
- اضبط AutoSize إلى False
- اضبط الحجم Size على 50, 60
- ضبط حجم الخط Font إلى 18
- تعيين خاصية TextAlign إلى MiddleCenter
- ضبط الموقع على 50 ,70 لضبط موقع عنصر التحكم داخل النموذج
- تعيين خاصية الاسم Name إلى plusLeftLabel

٣. في Form حدد العنصر plusLeftLabel الذي أنشأته، وانسخه.

٤. قم بلصق العنصر plusLeftLabel في Form ثلاث مرات.

٥. قم بترتيب العناصر الثلاث الجديدة بحيث تكون في صف واحد على يمين العنصر plusLeftLabel

٦. ضبط خاصية Text الخاصة بالعنصر الثاني إلى "+" علامة الجمع.

٧. تعيين خاصية Name الخاصة بالعنصر الثالث إلى `plusRightLabel`

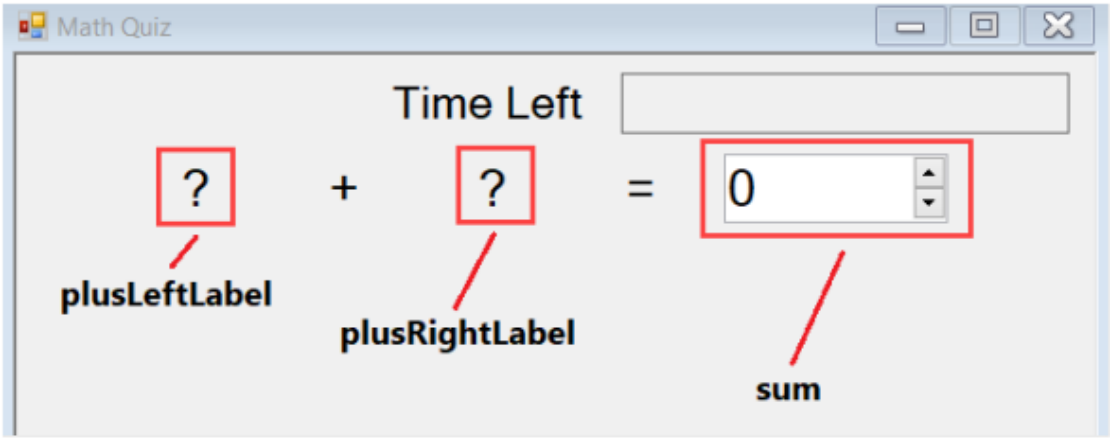
٨. ضبط خاصية Text الخاصة بالعنصر الرابع إلى "=" علامة يساوي.

٩. حدد Form أضف من Toolbox عنصر تحكم NumericUpDown ستتعلم المزيد عن هذا النوع من عناصر التحكم لاحقًا.

١٠. من نافذة الخصائص Properties قم بتعيين الخصائص التالية لعنصر NumericUpDown

- ضبط حجم الخط Font إلى 18
- اضبط خاصية MaximumSize > Width عرض 100
- تعيين خاصية الاسم Name إلى sum

١١. قم بمحاذاة عنصر التحكم NumericUpDown مع عناصر التحكم Label الأخرى.

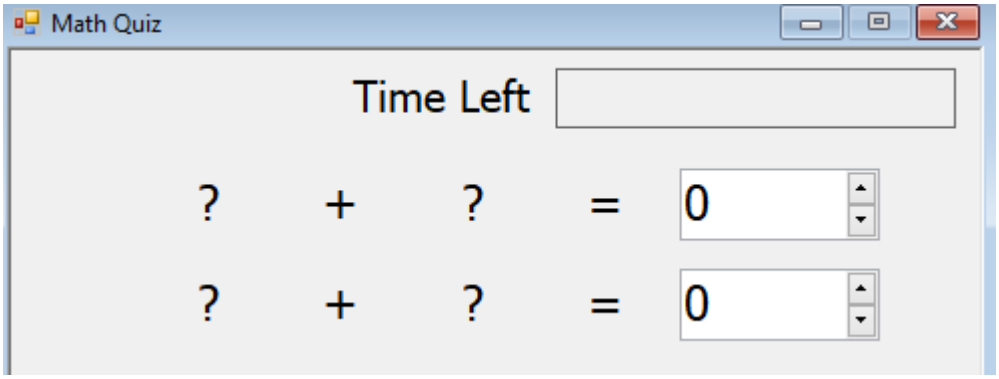


إضافة عناصر تحكم لمسائل الطرح والضرب والقسمة

قم بإضافة Label إلى Form لمسائل الرياضيات المتبقية.

١. انسخ عناصر التحكم Label الأربعة وعنصر NumericUpDown التي أنشأتها لمشكلة الجمع، وألصقها في Form

٢. قم بضبط موقع عناصر التحكم الجديدة، لتتماشى مع عناصر الجمع +



٣. من نافذة الخصائص، قم بتعيين الخصائص التالية، لعناصر التحكم الجديدة:

- تعيين خاصية الاسم Name الخاص بعلامة الاستفهام الأولى إلى minusLeftLabel
- ضبط خاصية Text الخاصة بعنصر Label الثاني إلى "-" علامة الطرح.
- تعيين خاصية الاسم Name الخاص بعلامة الاستفهام الثانية إلى minusRightLabel
- تعيين خاصية الاسم Name الخاص بالعنصر NumericUpDown إلى difference

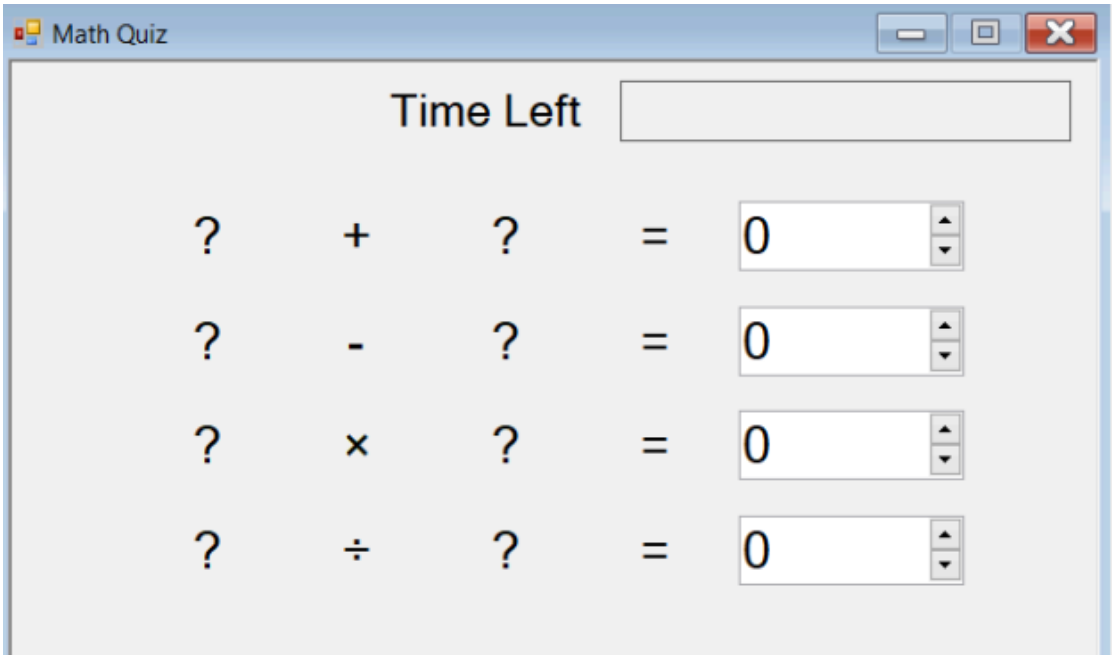
٤. انسخ صف عناصر تحكم الجمع مرة أخرى، ثم ألصقها مرتين أخريين في النموذج.

٥. من نافذة الخصائص، قم بتعيين الخصائص التالية، لعناصر الصف الثالث:

- تعيين خاصية الاسم Name الخاص بعلامة الاستفهام الأولى إلى timesLeftLabel
- ضبط خاصية Text الخاصة بعنصر Label الثاني إلى "*" علامة الضرب.
- تعيين خاصية الاسم Name الخاص بعلامة الاستفهام الثانية إلى timesRightLabel
- تعيين خاصية الاسم Name الخاص بالعنصر NumericUpDown إلى product

٦. قم بتعيين الخصائص التالية، لعناصر الصف الرابع:

- تعيين خاصية الاسم Name الخاص بعلامة الاستفهام الأولى إلى dividedLeftLabel
- ضبط خاصية Text الخاصة بعنصر Label الثاني إلى "/" علامة القسمة.
- تعيين خاصية الاسم Name الخاص بعلامة الاستفهام الثانية إلى dividedRightLabel
- تعيين خاصية الاسم Name الخاص بالعنصر NumericUpDown إلى quotient



إضافة زر للبدء، وتعيين ترتيب فهرس علامات التبويب عناصر التحكم

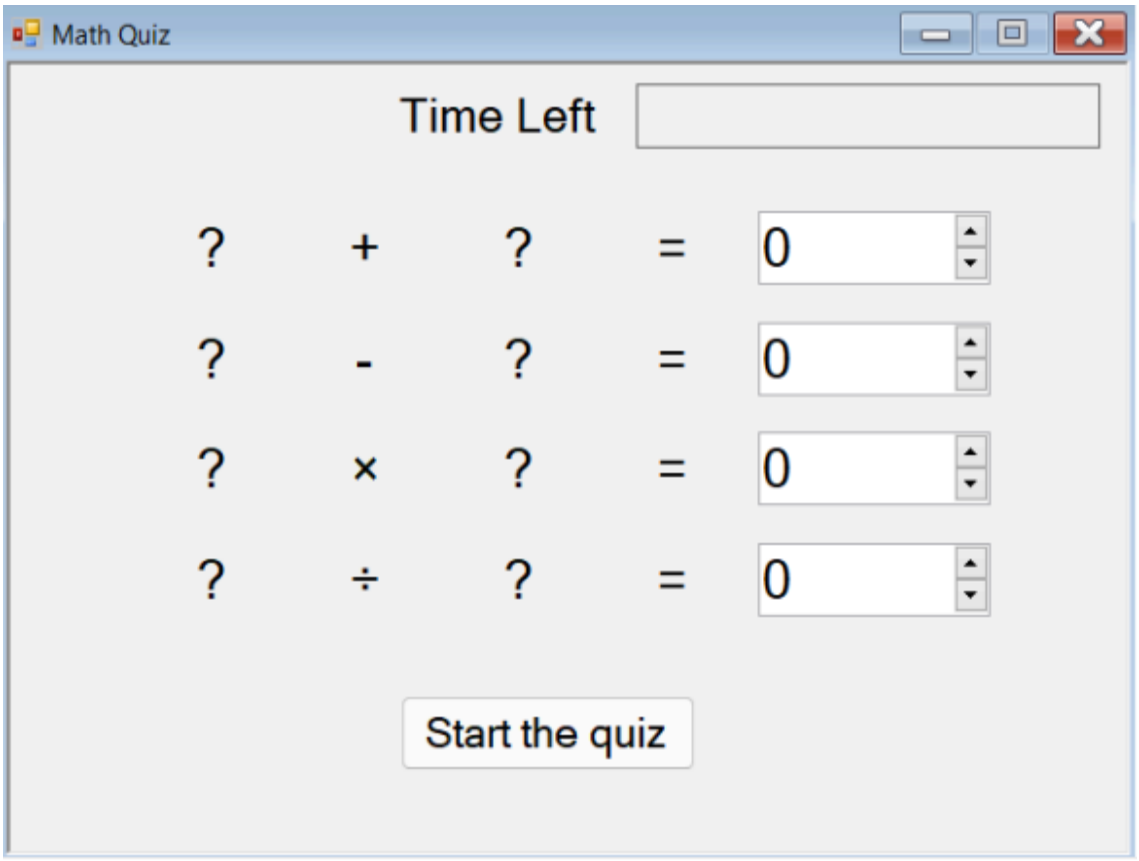
يوضح لك هذا القسم كيفية إضافة زر بدء، وكيفية تحديد ترتيب علامات التبويب لكافة عناصر التحكم، يحدد هذا الترتيب كيفية انتقال المشارك في الاختبار من عنصر تحكم إلى آخر باستخدام مفتاح Tab

١. إضافة عنصر التحكم زر Button من صندوق الأدوات إلى النموذج.

٢. من نافذة الخصائص، قم بتعيين خصائص Button

- ضبط خاصية Text إلى Start the quiz
- تعيين خاصية الاسم Name إلى startButton
- ضبط حجم الخط Font إلى 14
- اضبط خاصية AutoSize إلى True مما يتسبب في تغيير حجم الزر Button تلقائيًا، كي يتناسب مع النص
- اضبط خاصية TabIndex على 0 هذه القيمة تجعل زر البدء هو عنصر التحكم الأول الذي يتلقى التركيز

٣. قم بوضع Start the quiz في المنتصف بالقرب من أسفل النموذج.



٤. من نافذة الخصائص، قم بتعيين خاصية TabIndex لكافة عناصر تحكم NumericUpDown

- قم بتعيين TabIndex 1 لعنصر التحكم الجمع
- قم بتعيين TabIndex 2 لعنصر التحكم الطرح
- قم بتعيين TabIndex 3 لعنصر التحكم الضرب
- قم بتعيين TabIndex 4 لعنصر التحكم القسمة

تشغيل التطبيق

لم تكتب التعليمات بعد، ولكن لا يزال بإمكانك تشغيل التطبيق، للتحقق مما إذا كانت قيم TabIndex تعمل بالشكل المتوقع.

١. استخدم إحدى الطرق التالية لحفظ التطبيق:

- اضغط **Ctrl + Shift + S**
- في شريط القائمة، حدد ملف > حفظ الكل File > Save All
- على شريط الأدوات، حدد زر حفظ الكل Save All

٢. استخدم إحدى الطرق لتشغيل التطبيق.

٣. قم بتحديد مفتاح Tab عدة مرات لرؤية كيفية انتقال التركيز من عنصر تحكم إلى آخر.

إضافة مسائل رياضية إلى تطبيق Math Quiz

في هذا القسم سوف تستخدم عناصر التحكم، من خلال إضافة تعليمات برمجية، لإنشاء مسائل رياضية تعتمد على أرقام عشوائية، وإنشاء أسلوب يحمل اسم StartTheQuiz() لملء المسائل.

سنتعلم كيفية:

- كتابة تعليمات لإنشاء كائنات عشوائية لاستخدامها في مسائل الرياضيات.
- إضافة معالج حدث لزر البدء startButton
- كتابة تعليمات لبدء الاختبار Start the quiz

إنشاء مشكلة جمع عشوائية

١. حدد النموذج Form

٢. في شريط القائمة، حدد عرض ثم اختر التعليمات View > Code أو اضغط مفتاح F7 حتى تتمكن من عرض صفحة ملف التعليمات البرمجية خلف النموذج.

٣. قم بإنشاء كائن عشوائي عن طريق إضافة التعليمات التالية بالقرب من أعلى التعليمات الموجودة.

```
Random randomizer = new Random();
```

```
namespace MathQuiz
{
    3 references
    public partial class Form1 : Form
    {
        // Create a Random object called randomizer
        // to generate random numbers.
        Random randomizer = new Random();
```

يمكنك استخدام عبارات مثل هذه، new لإنشاء عناصر تحكم مثل أزرار Button و Label و OpenFileDialogs و ColorDialogs وحتى النماذج Forms تسمى هذه العناصر كائنات Randoms

عند تشغيل البرنامج، يتم تشغيل Form تقوم التعليمات الموجود خلفه، بإنشاء كائن عشوائي ويسمي `randomizer`

يحتاج اختبارك إلى متغيرات لتخزين الأرقام العشوائية التي ينشئها لكل مشكلة، قبل استخدام المتغيرات، عليك إعلانها، مما يعني سرد أسمائها، وأنواع بياناتها.

أضف متغيرين صحيحين إلى النموذج، وأطلق عليهما اسم addend1, addend2

كما تعرف، تستخدم بناء جملة لإضافة متغير عدد صحيح مشابهًا لإضافة الكائن العشوائي كما توضح التعليمات التالية:

```
// Create a Random object called randomizer
// to generate random numbers.
Random randomizer = new Random();

// These integer variables store the numbers
// for the addition problem.
int addend1;
int addend2;
```

٣. أضف أسلوب يسمى StartTheQuiz() يستخدم هذا الأسلوب أسلوب Next() المساعد، الخاصة بالكائن العشوائي randomizer لتوليد أرقام عشوائية للعلامات، ستقوم في النهاية بملء جميع المسائل الرياضية ثم بدء تشغيل المؤقت، لذا أضف هذه المعلومات إلى التعليق الموجز.

يجب أن يبدو الأسلوب StartTheQuiz() مثل التعليمات التالية:

```
/// <summary>
/// Start the quiz by filling in all of the problems
/// and starting the timer.
/// </summary>
public void StartTheQuiz()
{
    // خاص بمسألة الجمع
    addend1 = randomizer.Next(51);
    addend2 = randomizer.Next(51);

    // Convert the two randomly generated numbers
    // into strings so that they can be displayed
    // in the label controls.
    plusLeftLabel.Text = addend1.ToString();
    plusRightLabel.Text = addend2.ToString();

    // 'sum' is the name of the NumericUpDown control.
    // This step makes sure its value is zero before
    // adding any values to it.
    sum.Value = 0;
}
```

عند استخدام `Next()` مع كائن عشوائي، كما هو الحال عند استدعاء `randomizer.Next(51);` ستحصل على رقم عشوائي أقل من 50 أو بين 0 ,50 تستدعي هذه التعليمة `randomizer.Next(51);` الثانية بحيث يصل مجموع الرقمين العشوائيين إلى إجابة بين 0 و 100

٤. انظر عن كثب إلى هذه التصريحات:

```
plusLeftLabel.Text = addend1.ToString();
plusRightLabel.Text = addend2.ToString();
```

تحدد هذه العبارات خصائص النص `Text` لـ `plusLeftLabel, plusRightLabel` بحيث تعرض الرقمين العشوائيين، تعرض عناصر التحكم في `Label` القيم بتنسيق نصي، وفي البرمجة، يقوم أسلوب كل عدد صحيح `ToString()` بتحويل العدد الصحيح إلى نص يمكن لعنصر Label عرضه.

إنشاء مسائل عشوائية للطرح والضرب والقسمة

الخطوة التالية هي إعلان المتغيرات، وتوفير قيم عشوائية لمسائل الرياضيات الأخرى.

١. أضف متغيرات الأعداد الصحيحة لمسائل الرياضيات المتبقية إلى النموذج، بعد متغيرات مسألة الجمع، يجب أن تبدو التعليمات كاملة مثل العينة التالية:

```
// Create a Random object called randomizer
// to generate random numbers.
Random randomizer = new Random();
// These integer variables store the numbers
// for the addition problem.
int addend1;
int addend2;
// These integer variables store the numbers
// for the subtraction problem.
int minuend;
int subtrahend;
// These integer variables store the numbers
// for the multiplication problem.
int multiplicand;
int multiplier;
// These integer variables store the numbers
// for the division problem.
int dividend;
int divisor;
```

٢. قم بتعديل الأسلوب `StartTheQuiz()` عن طريق إضافة التعليمات التالية:

```
public void StartTheQuiz()
 {
// Fill in the addition problem.
// Generate two random numbers to add.
// Store the values in the variables 'addend1' and
'addend2'.
addend1 = randomizer.Next(51);
addend2 = randomizer.Next(51);
// Convert the two randomly generated numbers
// into strings so that they can be displayed
// in the label controls.
plusLeftLabel.Text = addend1.ToString();
```

```
plusRightLabel.Text = addend2.ToString();
// 'sum' is the name of the NumericUpDown control.
// This step makes sure its value is zero before
// adding any values to it.
sum.Value = 0;
// Fill in the subtraction problem.
minuend = randomizer.Next(1, 101);
subtrahend = randomizer.Next(1, minuend);
minusLeftLabel.Text = minuend.ToString();
minusRightLabel.Text = subtrahend.ToString();
difference.Value = 0;
// Fill in the multiplication problem.
multiplicand = randomizer.Next(2, 11);
multiplier = randomizer.Next(2, 11);
timesLeftLabel.Text = multiplicand.ToString();
timesRightLabel.Text = multiplier.ToString();
product.Value = 0;
// Fill in the division problem.
divisor = randomizer.Next(2, 11);
int temporaryQuotient = randomizer.Next(2, 11);
dividend = divisor * temporaryQuotient;
dividedLeftLabel.Text = dividend.ToString();
dividedRightLabel.Text = divisor.ToString();
quotient.Value = 0;

}
```

تستخدم هذه التعليمات أسلوب `Next()` من فئة `Random` بطريقة مختلفة قليلًا عن طريقة استخدام `Next()` في مشكلة الجمع، فعندما يعطي الأسلوب قيمتين، فإنه يختار رقمًا عشوائيًا أكبر من أو يساوي القيمة الأولى وأقل من القيمة الثانية.

من خلال استخدام `Next()` ذات الوسيطتين، يمكنك التأكد من أن مسألة الطرح لها إجابة موجبة، وأن إجابة الضرب هي 100 على الأكثر، وأن إجابة القسمة ليست كسرًا.

إضافة معالج حدث إلى زر البدء startButton

في هذا القسم، يمكنك إضافة تعليمات لبدء الاختبار، عند النقر فوق زر startButton التعليمات التي يتم تشغيلها كرد فعل لحدث مثل النقر فوق زر Button_Click() يسمى معالج الحدث.

١. في Form انقر نقرًا مزدوجًا فوق الزر Start the quiz تظهر صفحة التعليمات، ويظهر أسلوب جديد، ينفذ فقط عند النقر فوق الزر، وقت التشغيل.

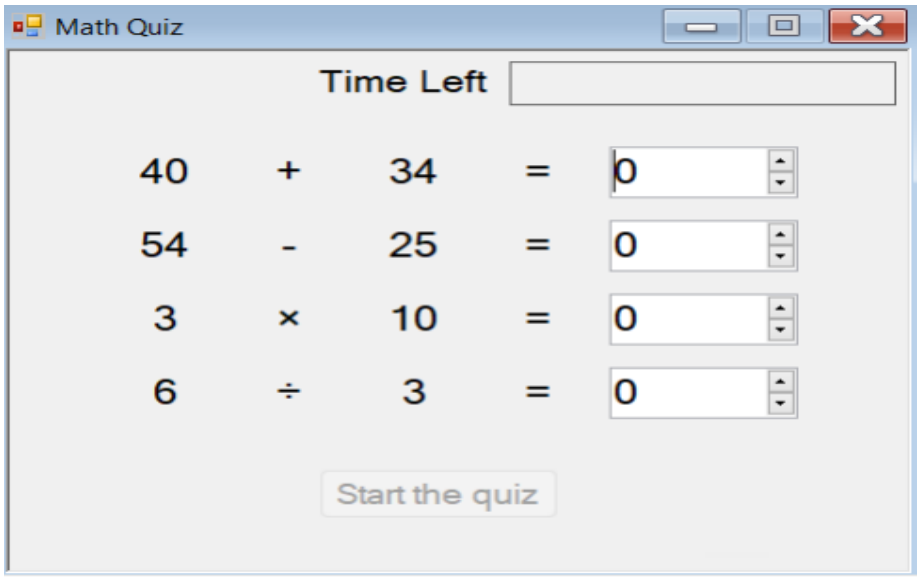
تضيف هذه الإجراءات معالج حدث Button_Click() النقر فوق زر startButton عندما يختار المشارك في الاختبار هذا الزر، يقوم التطبيق بتشغيل التعليمات التي سوف تضيفها إلى هذا الأسلوب الجديد.

٢. أضف العبارتين التاليتين حتى يبدأ معالج الحدث الاختبار:

```
private void startButton_Click(object sender, EventArgs e)
{
    StartTheQuiz();
    startButton.Enabled = false;
```

العبارة الأولى تقوم باستدعاء الأسلوب `StartTheQuiz()` الذي كتبناه في القسم السابق. العبارة الثانية تقوم بتعيين الخاصية `Enabled` إلى `false` الخاصة بالعنصر `startButton` بحيث لا يتمكن المشارك في الاختبار من نقر الزر أثناء الاختبار، وتغيير القيم.

٣. احفظ التعليمات البرمجية، وقم بتشغيل التطبيق، ثم انقر زر Start the quiz تظهر مسائل رياضية عشوائية، كما هو موضح في لقطة الشاشة التالية.



إضافة مؤقت إلى تطبيق Math Quiz

يستخدم الاختبار مكون عنصر تحكم Timer تتبع التعليمات الموجودة خلف هذا العنصر، الوقت المنقضي، ويتحقق من إجابات المشارك في الاختبار.

في هذا القسم، ستتعلم كيفية:

- إضافة عنصر التحكم Timer المؤقت
- أضف معالج حدث Timer
- اكتب تعليمات للتحقق من إجابات المستخدم، وعرض الرسائل، وملء الإجابات الصحيحة

إضافة عداد تنازلي countdown timer

لتتبع الوقت أثناء الاختبار، يمكنك استخدام مكون المؤقت Timer كما تحتاج إلى متغير لتخزين مقدار الوقت المتبقي.

١. أضف متغيرًا صحيحًا يسمي timeLeft بنفس الطريقة التي أعلنت بها المتغيرات سابقًا، ضع إعلان timeLeft بعد الإعلانات الأخرى مباشرًة، يجب أن تبدو التعليمات مثل العينة التالية:

```
public partial class Form1 : Form
{
    // Create a Random object called randomizer
    // to generate random numbers.
    Random randomizer = new Random();

    // These integer variables store the numbers
    // for the addition problem.
    int addend1;
    int addend2;

    // These integer variables store the numbers
    // for the subtraction problem.
    int minuend;
    int subtrahend;

    // These integer variables store the numbers
    // for the multiplication problem.
    int multiplicand;
    int multiplier;
```

```
    // These integer variables store the numbers
    // for the division problem.
    int dividend;
    int divisor;

    // This integer variable keeps track of the
    // remaining time.
    int timeLeft;
```

٢. من النموذج حدد صندوق الأدوات Toolbox ثم Components وأسحب عنصر Timer يظهر عنصر التحكم Timer في المنطقة الرمادية أسفل نافذة التصميم.

٣. في النموذج، حدد أيقونة timer1 الذي أضفته للتو، واضبط خاصية Interval على 1000 نظرًا لأن هذه الفترة الزمنية بالمللي ثانية، فإن القيمة 1000 تتسبب في قيام المؤقت بإثارة حدث نقرة Tick كل ثانية.

التحقق من الإجابات

نظرًا لأن المؤقت يثير حدث نقرة Tick كل ثانية، فمن المنطقي التحقق من الوقت المنقضي في معالج حدث Tick ومن العملي أيضًا التحقق من الإجابات في معالج الحدث هذا، إذا نفد الوقت، أو إذا كانت الإجابات صحيحة، فيجب أن ينتهي الاختبار.

قبل كتابة معالج الحدث هذا، أضف أسلوب `CheckTheAnswer()` يتم استدعاؤه لتحديد ما إذا كانت إجابات مسائل الرياضيات صحيحة أم لا، يجب أن يكون هذا الأسلوب متوافق مع الأساليب الأخرى `StartTheQuiz()` يجب ان تبدو التعليمات مثل المثال التالي:

```
private bool CheckTheAnswer()
{
  if ((addend1 + addend2 == sum.Value)
     && (minuend - subtrahend == difference.Value)
     && (multiplicand * multiplier == product.Value)
     && (dividend / divisor == quotient.Value))
      return true;
  else
      return false;
}
```

يحدد هذا الأسلوب `CheckTheAnswer()` إجابات مسائل الرياضيات، وتقارن النتائج بالقيم الموجودة في عناصر التحكم NumericUpDown تتضمن هذه التعليمات:

- لا يمكنك إدخال علامة الضرب (×) وعلامة القسمة (÷) بسهولة باستخدام لوحة المفاتيح، لذا تقبل لغة C# علامة (*) للضرب وعلامة (/) للقسمة.
- في C# تستخدم علامتي `&&` بدلاً من المعامل `and` يمكنك استخدام `logical and` للتحقق مما إذا كان أكثر من شرط صحيحًا، في هذه الحالة، إذا كانت القيم كلها صحيحة، سيرجع الأسلوب قيمة `true` والا سيرجع الأسلوب قيمة `false`
- تستخدم الكلمة الأساسية `if` خاصية `Value` لعنصر NumericUpDown للوصول إلى القيمة الحالية للعنصر، لاحقًا يمكنك استخدام نفس الخاصية لعرض الإجابة الصحيحة في كل عنصر تحكم.

إضافة معالج حدث event handler إلى المؤقت timer

الآن بعد أن أصبح لديك أسلوب للتحقق من الإجابات، يمكنك كتابة التعليمات الخاصة بمعالج حدث Tick يتم تشغيل هذه التعليمات كل ثانية، بعد أن يثير المؤقت حدث Tick يتحقق معالج الحدث هذا من إجابات المشارك في الاختبار عن طريق استدعاء أسلوب `CheckTheAnswer()` كما يتحقق من الوقت الذي انقضى اثناء الاختبار.

١ في النموذج انقر نقرًا مزدوجًا فوق العنصر Timer أو حدده ثم اضغط Enter تضيف هذه الإجراءات معالج حدث Tick event handler يظهر محرر التعليمات البرمجية، ويعرض أسلوب معالج حدث Tick handler's method

بالنسبة لـ C# فإنها تضيف سطرًا من التعليمات البرمجية في ملف Form1.Designer.cs الذي يقوم بربط معالج الحدث:

```
timer1.Tick += new EventHandler(timer1_Tick);
```

٢. أضف العبارات التالية إلى أسلوب معالج الحدث الجديد:

```
private void timer1_Tick(object sender, EventArgs e)
{
    if (CheckTheAnswer())
     {
    // If CheckTheAnswer() returns true, then the user
    // got the answer right. Stop the timer
    // and show a MessageBox.
     timer1.Stop();
     MessageBox.Show("You got all the answers right!",
                     "Congratulations!");
```

```
            startButton.Enabled = true;
        }
        else if (timeLeft > 0)
        {
        // If CheckTheAnswer() returns false, keep counting
        // down. Decrease the time left by one second and
        // display the new time left by updating the
        // Time Left label.
           timeLeft = timeLeft - 1;
            timeLabel.Text = timeLeft + " seconds";
       }
        else
        {
         // If the user ran out of time, stop the timer, show
         // a MessageBox, and fill in the answers.
         timer1.Stop();
         timeLabel.Text = "Time's up!";
         MessageBox.Show("You didn't finish in
time.", "Sorry!");
         sum.Value = addend1 + addend2;
         difference.Value = minuend - subtrahend;
         product.Value = multiplicand * multiplier;
         quotient.Value = dividend / divisor;
         startButton.Enabled = true;
       }
}
```

يتم تشغيل هذا الأسلوب في كل ثانية من الاختبار، تتحقق التعليمات أولًا من القيمة التي يرجعها `CheckTheAnswer()`

- إذا كانت جميع الإجابات صحيحة، فإن هذه القيمة هي `true` وينتهي الاختبار:
  - يتوقف المؤقت timer
  - تظهر رسالة تهنئة congratulatory message
  - تم تعيين الخاصية Enabled لعنصر الزر startButton إلى true حتى يتمكن المشارك من بدء اختبار آخر.
- إذا ارجع `CheckTheAnswer()` قيمة `false` تتحقق التعليمات من قيمة timeLeft
  - إذا كان هذا المتغير أكبر من 0 يطرح المؤقت 1 من timeLeft كما يقوم أيضًا بتحديث خاصية Text لعنصر timeLabel لإظهار عدد الثواني المتبقية للمشارك في الاختبار.

- إذا لم يتبق وقت، يتوقف المؤقت، ويغير نص timeLabel إلى Time's up! وتظهر رسالة تعلن انتهاء الاختبار، وتظهر الإجابات، ويصبح زر startButton متاحًا مرة أخرى.

بدء تشغيل المؤقت Start the timer

١. لبدء تشغيل المؤقت عند بدء الاختبار، أضف ثلاثة أسطر إلى نهاية الأسلوب StartTheQuiz() كما يوضح المثال التالي:

```
public void StartTheQuiz()
{
    // Fill in the addition problem.
    // Generate two random numbers to add.
    // Store the values in the variables 'addend1'
and 'addend2'
    addend1 = randomizer.Next(51);
    addend2 = randomizer.Next(51);

   // Convert the two randomly generated numbers
   // into strings so that they can be displayed
   // in the label controls.
    plusLeftLabel.Text = addend1.ToString();
    plusRightLabel.Text = addend2.ToString();

    // 'sum' is the name of the NumericUpDown
control.
   // This step makes sure its value is zero before
    // adding any values to it.
    sum.Value = 0;

    // Fill in the subtraction problem.
    minuend = randomizer.Next(1, 101);
    subtrahend = randomizer.Next(1, minuend);
    minusLeftLabel.Text = minuend.ToString();
   minusRightLabel.Text = subtrahend.ToString();
    difference.Value = 0;

    // Fill in the multiplication problem.
    multiplicand = randomizer.Next(2, 11);
    multiplier = randomizer.Next(2, 11);
   timesLeftLabel.Text = multiplicand.ToString();
    timesRightLabel.Text = multiplier.ToString();
```

```
        product.Value = 0;

        // Fill in the division problem.
        divisor = randomizer.Next(2, 11);
        int temporaryQuotient = randomizer.Next(2, 11);
        dividend = divisor * temporaryQuotient;
        dividedLeftLabel.Text = dividend.ToString();
        dividedRightLabel.Text = divisor.ToString();
        quotient.Value = 0;

        // Start the timer.
        timeLeft = 30;
        timeLabel.Text = "30 seconds";
        timer1.Start();
}
```

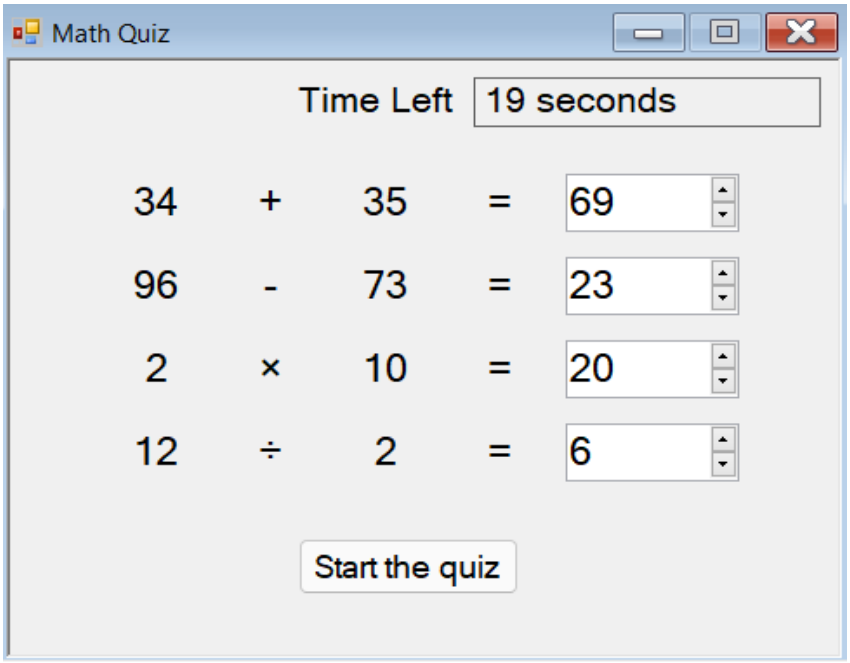
عند بدء الاختبار، تقوم هذه التعليمات بتعيين المتغير timeLeft إلى 30 وخاصية Text لعنصر التحكم **timeLabel** إلى "30 seconds" ثم يبدأ أسلوب **Start()** لعنصر Timer في العد التنازلي.

تشغيل التطبيق

١. احفظ برنامجك ثم قم بتشغيله.

٢. انقر زر Start the quiz يبدأ العد التنازلي للمؤقت، عندما ينفد الوقت، ينتهي الاختبار، وتظهر الإجابات.

٣. ابدأ اختبار آخر، وقدم إجابات صحيحة للمسائل الرياضية، عندما تجيب بشكل صحيح في غضون المهلة الزمنية، يتم فتح مربع رسالة، ويصبح زر البدء متاحًا، ويتوقف المؤقت.



تخصيص مظهر تطبيق Math Quiz

يوضح هذا القسم كيفية تحسين الاختبار الخاص بك، عن طريق مسح القيم الافتراضية، وتخصيص مظهر عناصر التحكم.

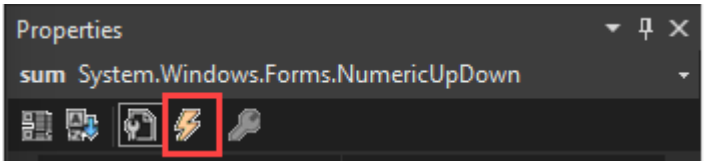
في هذا القسم النهائي ستتعلم كيفية:

- أضف معالجات الأحداث لمسح القيم الافتراضية لعنصر التحكم NumericUpDown
- تخصيص الاختبار Customize the quiz

إضافة معالجات الأحداث لعناصر التحكم NumericUpDown

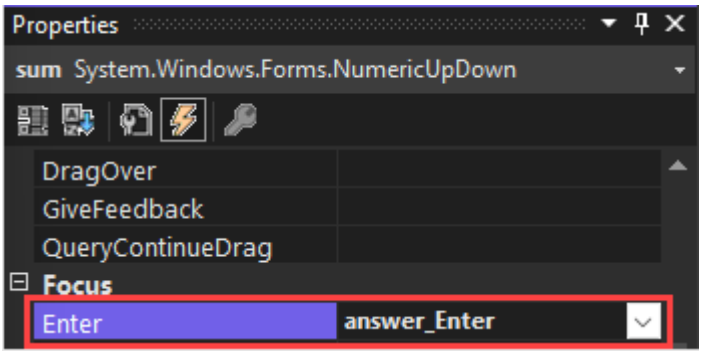
يحتوي الاختبار على عناصر تحكم NumericUpDown يستخدمها المشاركون في الاختبار لإدخال الأرقام، عند إدخال إجابة، يتعين عليك إما تحديد القيمة الافتراضية أولًا أو حذف تلك القيمة يدويًا، من خلال إضافة معالج حدث Enter يمكنك تسهيل إدخال الإجابات، تحدد هذه التعليمات القيمة الحالية في كل عنصر تحكم NumericUpDown ويمسحها بمجرد تحديد المشارك لها، وبدء إدخال قيمة مختلفة.

١. حدد عنصر التحكم NumericUpDown الأول في Form من نافذة الخصائص Properties حدد أيقونة الأحداث Events على شريط الأدوات toolbar



تعرض علامة التبويب الأحداث Events في الخصائص جميع الأحداث التي يمكنك الاستجابة لها للعنصر الذي حددته في النموذج، في هذه الحالة، تتعلق جميع الأحداث المدرجة بعنصر التحكم NumericUpDown

٢. حدد الحدث Enter وأدخل answer_Enter وانقر Enter



يظهر محرر التعليمات البرمجية ويعرض معالج الحدث Enter الذي قمت بإنشائه لعنصر التحكم NumericUpDown

٣. في أسلوب معالج الحدث answer_Enter أضف التعليمات التالية:

```
private void answer_Enter(object sender, EventArgs e)
{
    // Select the whole answer in the
NumericUpDown control.
  NumericUpDown answerBox = sender as NumericUpDown;

    if (answerBox != null)
    {
     int lengthOfAnswer =
answerBox.Value.ToString().Length;
        answerBox.Select(0, lengthOfAnswer);
    }
}
```

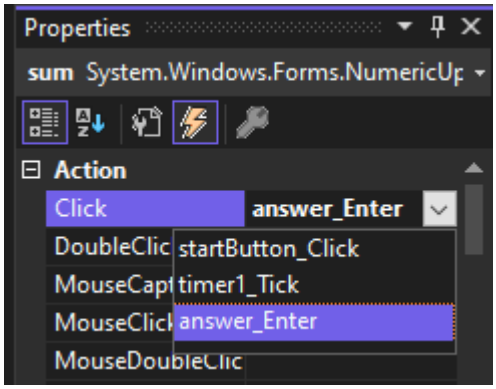
في هذه التعليمات:

- يعلن السطر الأول عن الأسلوب، ويتضمن معلمة تسمى `sender` في C# تكون المعلمة هي `object sender` تشير هذه المعلمة إلى الكائن الذي يتم إطلاق حدثه، والذي يُعرف باسم المرسل sender في هذه الحالة، يكون كائن المرسل هو عنصر التحكم NumericUpDown
- يقوم السطر الأول داخل الأسلوب بتحويل أو تحويل المرسل من كائن عام إلى عنصر تحكم NumericUpDown يقوم هذا السطر أيضًا بتعيين اسم answerBox لعنصر التحكم NumericUpDown
  - ستستخدم كافة عناصر التحكم NumericUpDown في النموذج هذا الأسلوب، وليس فقط عنصر التحكم في مسألة الجمع.
- يتحقق السطر التالي مما إذا كان تم تحويل answerBox بنجاح إلى عنصر تحكم NumericUpDown
- يحدد السطر الأول داخل العبارة if طول الإجابة الموجودة حاليًا في عنصر NumericUpDown
- يستخدم السطر الثاني داخل العبارة if طول الإجابة لتحديد القيمة الحالية في عنصر التحكم.

عندما يختار المشارك في الاختبار عنصر التحكم، يقوم Visual Studio بتشغيل هذا الحدث، تحدد هذه التعليمات الإجابة الحالية، بمجرد أن يبدأ المشارك في الاختبار إدخال إجابة مختلفة، يتم مسح الإجابة الحالية واستبدالها بالإجابة الجديدة.

١. في Windows Forms Designer حدد عنصر التحكم NumericUpDown الخاص بمسألة الجمع the addition problem's

٢. في صفحة الأحداث Events في نافذة الخصائص، أبحث عن حدث النقر **Click** event ثم حدد answer_Enter من القائمة المنسدلة، معالج الحدث هذا، هو الذي أضفته للتو.



٣. في Windows Forms Designer حدد عنصر التحكم NumericUpDown الخاص بمسألة الطرح the subtraction problem's

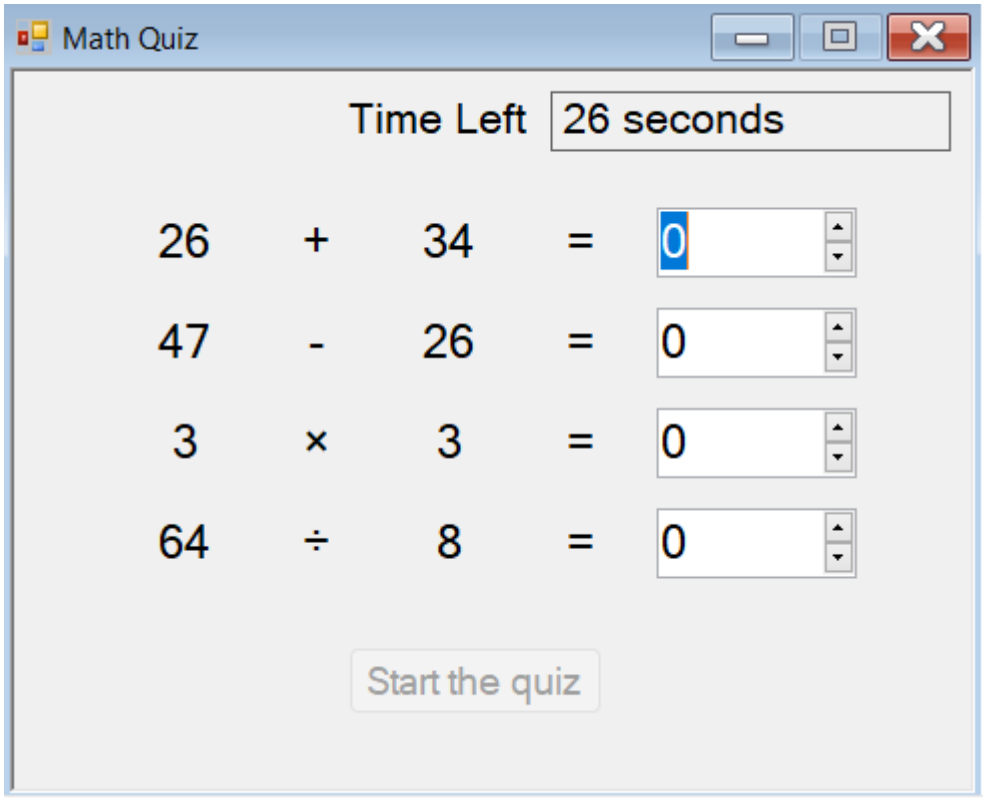
٤. في صفحة الأحداث Events في نافذة الخصائص، أبحث عن حدث الإدخال **Enter** event ثم حدد answer_Enter من القائمة المنسدلة، معالج الحدث هذا، هو الذي أضفته للتو، كرر هذه الخطوة لحدث النقر **Click** event

٥. كرر الخطوتين السابقتين لعناصر التحكم NumericUpDown في الضرب والقسمة the multiplication and division

تشغيل التطبيق

١. احفظ برنامجك ثم قم بتشغيله

٢. ابدأ اختبارًا ثم حدد عنصر NumericUpDown يتم تحديد القيمة الموجودة تلقائيًا، ثم مسحها عند البدء في إدخال قيمة مختلفة.



قم بتخصيص الاختبار الخاص بك

في هذا الجزء الأخير من البرنامج التعليمي، سوف تستكشف بعض الأساليب، لتخصيص الاختبار وتوسيع ما تعلمته.

تغيير لون الملصق Label

استخدم خاصية BackColor في عنصر التحكم timeLabel لتحويل هذا الملصق إلى اللون الأحمر، عندما يتبقى خمس ثوانٍ فقط في الاختبار.

```
timeLabel.BackColor = Color.Red;
```

قم بإعادة تعيين اللون عند انتهاء الاختبار.

قم بتشغيل صوت للحصول على إجابة صحيحة

امنح المشارك في الاختبار تلميحًا من خلال تشغيل صوت عند إدخال الإجابة الصحيحة في عنصر NumericUpDown لتنفيذ هذه الوظيفة، اكتب معالج حدث event handler لحدث ValueChanged الخاص بكل عنصر تحكم، يتم تشغيل هذا النوع من الأحداث كلما غيّر أحد المشاركين في الاختبار قيمة عنصر التحكم.

تهانينا! لقد انتهيت من هذه السلسلة من الدروس التعليمية، لقد أكملت مهام البرمجة والتصميم، في بيئة التطوير المتكاملة لبرنامج Visual Studio

٦ إنشاء لعبة Matching Game

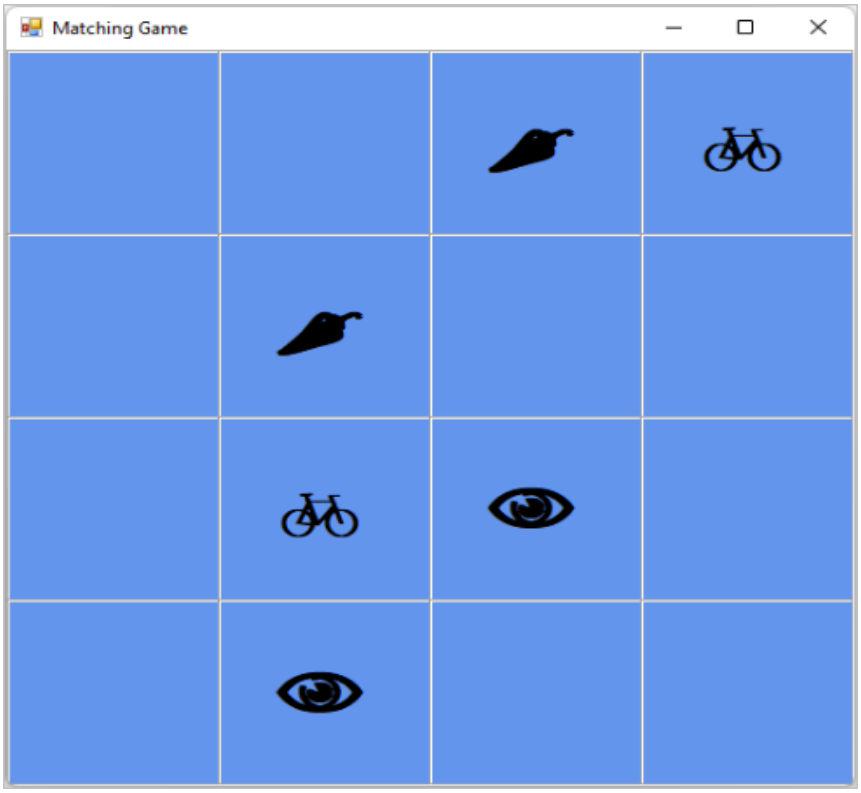
في هذه الوحدة المكونة من أربعة أقسام تعليمية، يمكنك إنشاء لعبة مطابقة، Windows Forms

في لعبة المطابقة، يختار اللاعب مربعًا لرؤية أيقونة، ثم يختار مربعًا آخر، إذا تطابقت الأيقونات، فستظل مرئية، وإذا لم تتطابق، تخفي اللعبة كلا الرمزين. في هذا الدرس التعليمي، يمكنك تعيين الأيقونات للملصقات Labels بشكل عشوائي، يمكنك ضبطها بحيث تكون مخفية ثم يتم عرضها عند تحديدها.

استخدم هذه البرامج التعليمية للتعرف على المهام التالية في بيئة التطوير المتكاملة الخاصة بـ Visual Studio

- تخزين الكائنات objects مثل الأيقونات icons في كائن List<T>
- استخدم حلقة `foreach` للتكرار عبر العناصر الموجودة في القائمة list
- تتبع حالة النموذج باستخدام المتغيرات المرجعية reference variables
- إنشاء معالج حدث للاستجابة للأحداث events التي يمكنك استخدامها مع كائنات متعددة multiple objects
- إنشاء مؤقت timer يقوم بالعد التنازلي ثم يقوم بإطلاق حدث ما مرة واحدة، بالضبط بعد بدئه.

عندما تنتهي، سيكون لديك لعبة كاملة.



في هذا القسم التعليمي الأول، ستتعلم كيفية:

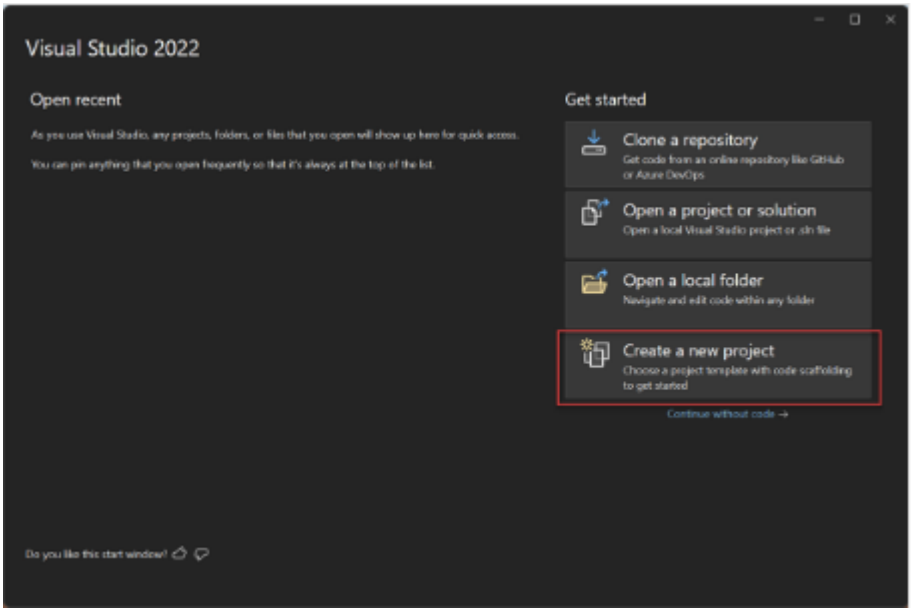
- إنشاء مشروع Visual Studio project يستخدم Windows Forms
- إضافة وتنسيق عنصر تخطيط layout
- إضافة وتنسيق ملصقات labels

إنشاء المشروع MatchingGame

الخطوة الأولى لإنشاء لعبتك هي إنشاء مشروع تطبيق Windows Forms

١. افتح Visual Studio

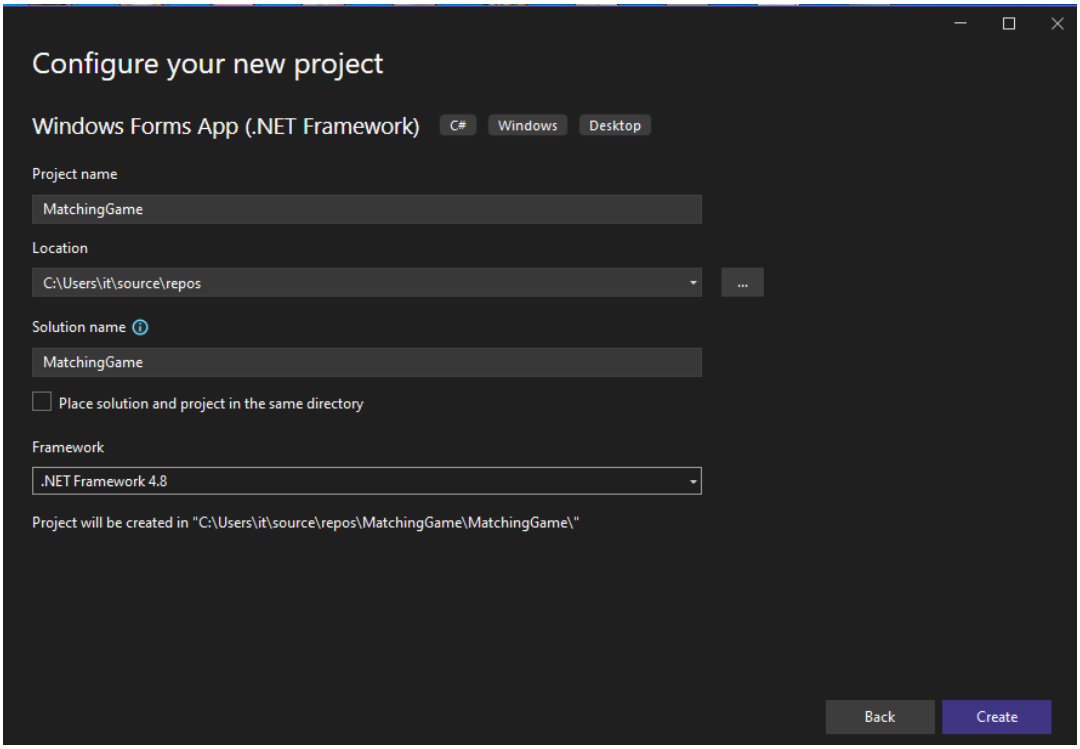
٢. من نافذة البدء حدد إنشاء مشروع جديد Create a new project



٣. في نافذة إنشاء مشروع جديد Create a new project ابحث عن Windows Forms وحدد Desktop من قائمة أنواع المشاريع All project types

٤. حدد قالب تطبيق Windows Forms App (.NET Framework) ثم اضغط Next

٥. في نافذة تكوين مشروعك الجديد Configure your new project قم بتسمية المشروع MatchingGame ثم اضغط إنشاء Create



ينشئ Visual Studio حلاً solution لتطبيقك، والحل solution عبارة عن حاوية للمشروعات والملفات التي يحتاجها تطبيقك.

في هذه المرحلة، يعرض Visual Studio نموذجًا فارغًا empty form في Windows Forms Designer

إنشاء تخطيط layout للعبة

سنقوم بإنشاء الشبكة ذات الأربعة أبعاد للعبة.

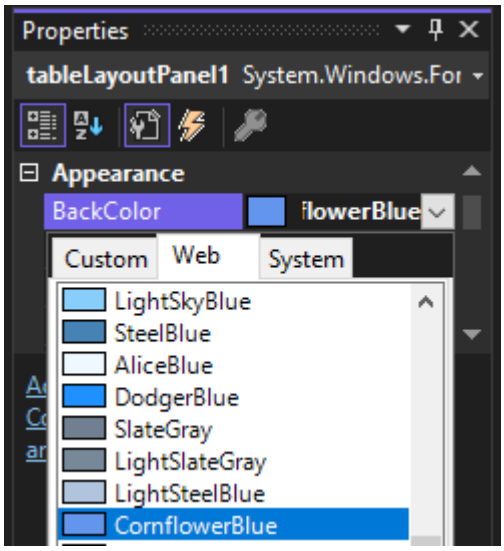
١. حدد النموذج لتحديد مصمم نماذج الويندوز Windows Forms Designer تظهر علامة التبويب [Design] Form1.cs في نافذة الخصائص، اضبط خصائص النموذج التالية:

- قم بتغيير خاصية النص Text من Form1 إلى Matching Game الآن يظهر هذا النص في أعلى نافذة اللعبة.
- قم بتعيين حجم النموذج، يمكنك تغييره إما عن طريق تعيين خاصية Size إلى 550, 550 أو عن طريق سحب زاوية النموذج حتى ترى الحجم المطلوب في أسفل بيئة التطوير المتكاملة Visual Studio IDE

٢. افتح صندوق الأدوات Toolbox واسحب عنصر التحكم TableLayoutPanel من فئة Containers أو انقر فوقه نقرًا مزدوجًا.

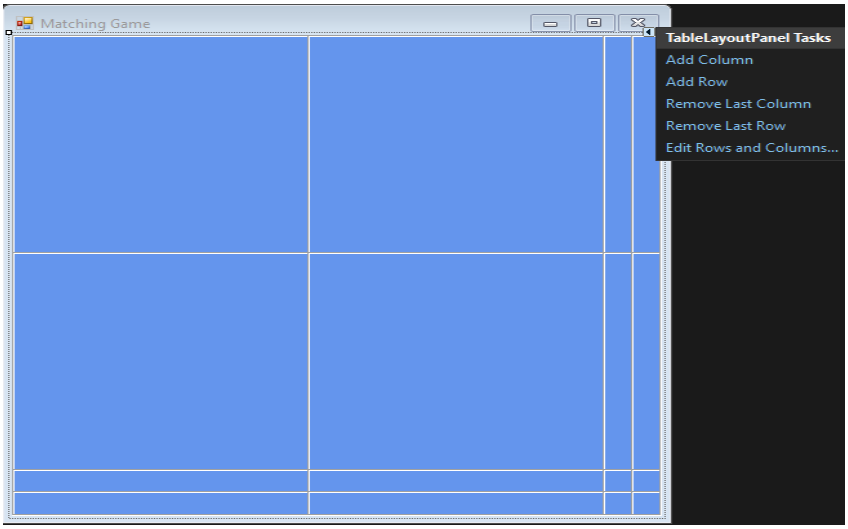
اضبط الخصائص التالية TableLayoutPanel من نافذة الخصائص:

- اضبط خاصية BackColor على CornflowerBlue لتعيين هذه الخاصية، حدد السهم لأسفل بجوار خاصية BackColor تظهر قائمة منسدلة حدد Web اختر من الأسماء CornflowerBlue أو يمكنك كتابة الاسم في الخانة.

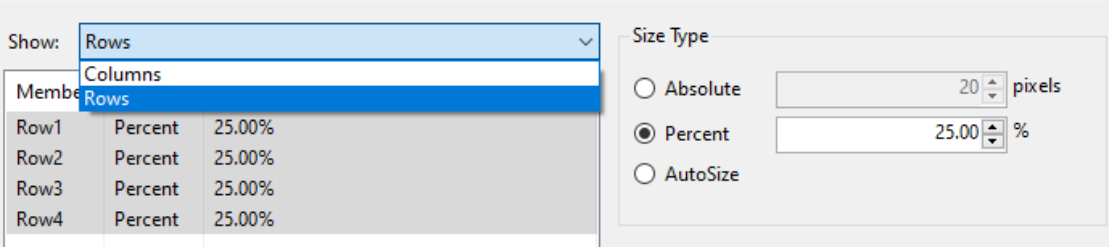


ملحوظة: الألوان ليست مرتبة أبجديًا، اللون CornflowerBlue يقع في أسفل القائمة.

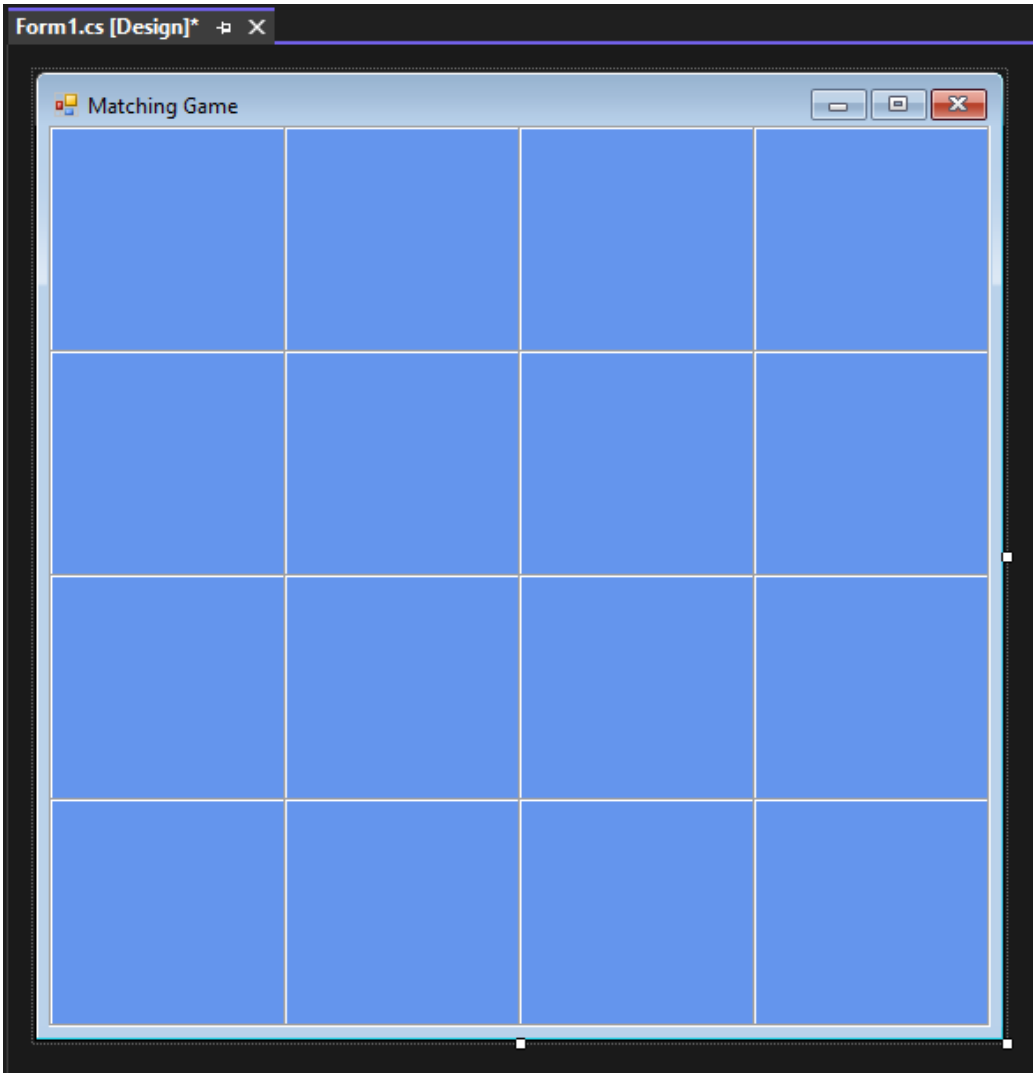
- اضبط خاصية Dock على Fill من خلال تحديد السهم لأسفل ثم تحديد الزر الأوسط الكبير، يعمل هذا الخيار على نشر الجدول بحيث يغطي كامل مساحة النموذج.
- اضبط خاصية CellBorderStyle على Inset توفر هذه القيمة حدودًا مرئية بين كل خلية على اللوحة.
- حدد زر المثلث في الزاوية اليمنى العليا من TableLayoutPanel "لوحة تخطيط الجدول" لعرض قائمة المهام الخاصة بها، في قائمة المهام، اضغط إضافة صف Add Row مرتين، لإضافة صفين آخرين، ثم اضغط إضافة عمود Add Column مرتين، لإضافة عمودين آخرين.



- في قائمة المهام، حدد تحرير الصفوف والأعمدة Edit Rows and Columns لفتح نافذة أنماط الأعمدة والصفوف Column and Row Styles لكل عمود، حدد خيار النسبة المئوية Percent ثم اضبط عرض كل عمود على 25 بالمائة.
- قم بتحديد الصفوف Rows من القائمة الموجودة في أعلى النافذة، ثم قم بضبط ارتفاع كل صف إلى 25 بالمائة.



- عندما تنتهي، حدد موافق OK لحفظ التغييرات.



أصبحت TableLayoutPanel الآن عبارة عن شبكة بأربعة مربعات تحتوي على ١٦ خلية مربعة متساوية الحجم، وتظهر الرموز لاحقًا في هذه الصفوف والأعمدة.

إضافة وتنسيق الملصقات labels

في هذا القسم، يمكنك إنشاء وتنسيق الملصقات التي تظهر أثناء اللعبة.

١. تأكد من تحديد TableLayoutPanel في محرر النماذج. يجب أن ترى tableLayoutPanel1 أعلى نافذة الخصائص Properties إذا لم يكن محددًا، فحدد tableLayoutPanel1 في النموذج، أو حدده من القائمة الموجودة في أعلى نافذة الخصائص.

٢. افتح صندوق الأدوات، وافتح فئة عناصر التحكم الشائعة Common Controls أضف عنصر تحكم ملصق تسمية Label إلى الخلية العلوية اليسرى من TableLayoutPanel تم الآن، تحديد عنصر التحكم label واضبط الخصائص التالية له:

- تعيين خاصية BackColor إلى CornflowerBlue
- تعيين خاصية AutoSize إلى False
- تعيين خاصية Dock إلى Fill
- تعيين خاصية TextAlign إلى MiddleCenter من خلال تحديد زر السهم لأسفل بجوار الخاصية ثم تحديد الزر الأوسط، تضمن هذه القيمة ظهور الرمز في منتصف الخلية.
- حدد خاصية الخط Font يظهر زر النقاط الثلاث (...) حدد النقاط الثلاث واضبط قيمة الخط على Webdings ونمط الخط Font Style على Bold والحجم Size على 48
- اضبط خاصية النص Text لعنصر label إلى الحرف **c**

تحتوي الخلية العلوية اليسرى من TableLayoutPanel الآن، على مربع أسود يقع في وسط خلفية زرقاء.

ملحوظة:

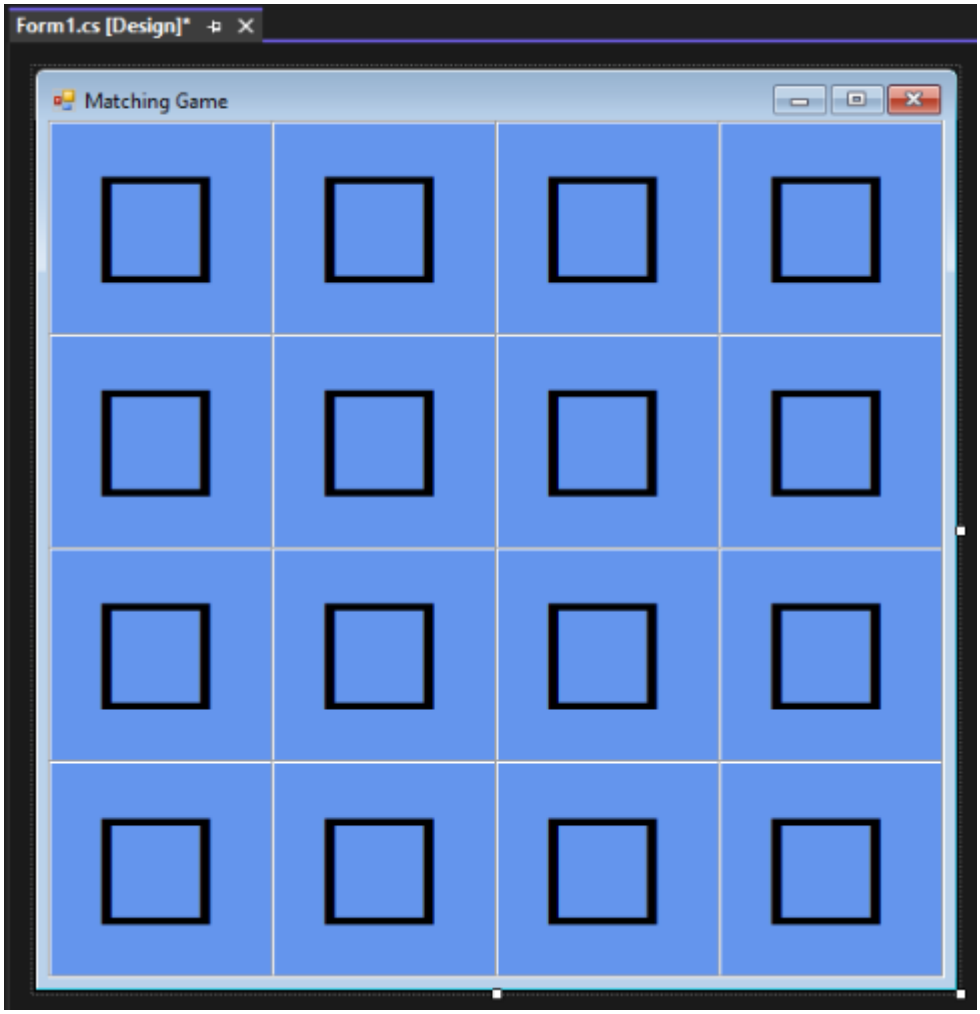
Webdings هو خط أيقوني يأتي مع نظام التشغيل Windows في لعبة المطابقة الخاصة بك، يقوم اللاعب بمطابقة أزواج من الرموز، يعرض هذا الخط Webdings الرموز المطابقة، إذا لم يتم عرض رموز بشكل صحيح على النموذج، قم بتعيين خاصية UseCompatibleTextRendering للملصقات الموجودة في النموذج إلى True

بدلاً من حرف **c** جرّب أحرفًا مختلفة في خاصية النص Text فعلامة التعجب هي عنكبوت، والحرف الكبير N هو عين، والفاصلة هي فلفل حار.

٣. حدد عنصر التحكم Label الخاص بك وانسخه إلى الخلية التالية في TableLayoutPanel اضغط مفتاحي **Ctrl+C** أو من شريط القائمة اختر **Edit** > **Copy** ثم ألصقها باستخدام **Ctrl+V** or **Edit** > **Paste**

تظهر نسخة من الملصق الأول Label في الخلية الثانية في TableLayoutPanel قم بلصقها مرة خرى، وسيظهر ملصق Label أخر في الخلية الثالثة، استمر في لصق عناصر التحكم حتى يتم ملء جميع الخلايا.

يكتمل تخطيط النموذج الخاص بك بهذه الخطوة.



إضافة أيقونات إلى لعبة المطابقة Matching Game

في لعبة المطابقة، يختار اللاعب مربعًا لرؤية أيقونة، ثم يختار مربعًا آخر، إذا تطابقت الأيقونات، فستظل مرئية، وإذا لم تتطابق، تخفي اللعبة كلا الرمزين. في هذا الدرس التعليمي، يمكنك تعيين الأيقونات للملصقات Labels بشكل عشوائي، يمكنك ضبطها بحيث تكون مخفية ثم يتم عرضها عند تحديدها.

في هذا القسم الثاني، ستتعلم كيفية:

- إضافة كائنًا عشوائيًا Random object وقائمة الأيقونات list of icons
- تعيين أيقونة عشوائية لكل ملصق label
- إضافة معالجات الأحداث التي تعرض الأيقونات على الملصقات labels

إضافة كائنًا عشوائيًا وقائمة الأيقونات

في هذا القسم، يمكنك إنشاء مجموعة من الأيقونات أو الرموز المتطابقة للعبة، تتم إضافة كل رمز إلى خليتين TableLayoutPanel والكائن الثاني هو كائن List<T> ويخزن الأيقونات المختارة عشوائيًا.

١. افتح Visual Studio يظهر مشروع MatchingGame الخاص بك ضمن افتح الأخير Open recent

٢. اختر Form1.cs ثم حدد **View > Code** كبديل حدد مفتاح F7 أو انقر نقرًا مزدوجًا فوق Form1 يعرض برنامج Visual Studio IDE وحدة الكود الخاصة بـ Form1

٣. أضف التعليمات التالية، في التعليمات الموجودة:

```
public partial class Form1 : Form
{
   // Use this Random object to choose random icons for the
squares
    Random random = new Random();

    // Each of these letters is an interesting icon
    // in the Webdings font,
    // and each icon appears twice in this list
    List<string> icons = new List<string>()
{
    "!", "!", "N", "N", ",", ",", "k", "k",
    "b", "b", "v", "v", "w", "w", "z", "z"
};
```

تأكد من وضع التعليمات بعد إعلان الفئة مباشرةً وبعد القوس المتعرج الافتتاحي

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

يمكنك استخدام كائنات القائمة لتتبع أنواع مختلفة من العناصر، يمكن أن تحتوي القائمة على أرقام أو قيم صحيحة/خاطئة true/false أو نص أو كائنات أخرى، في لعبة المطابقة الخاصة بك، يحتوي كائن القائمة على ١٦ سلسلة strings سلسلة لكل خلية في TableLayoutPanel كل سلسلة عبارة عن حرف واحد، يتوافق مع الأيقونات/الرموز الموجودة في الملصقات labels تظهر هذه الأحرف في خط Webdings كحافلة، دراجة، وغير ذلك.

ملحوظة:

يمكن للقوائم أن تتقلص وتنمو حسب الحاجة، وهو أمر مهم في هذا البرنامج.

معرفة المزيد عن القوائم، راجع List<T> لرؤية مثال في C# راجع مثال لقائمة أساسية A basic list example

تعيين أيقونة عشوائية لكل ملصق Label

في كل مرة تقوم فيها بتشغيل البرنامج، يقوم بتعيين الأيقونات بشكل عشوائي لعناصر labels في النموذج باستخدام الأسلوب AssignIconsToSquares() تستخدم هذه التعليمات الكلمة الأساسية foreach

١. أضف أسلوب AssignIconsToSquares() إلى ملف التعليمات:

```
private void AssignIconsToSquares()
{
  foreach (Control control in
tableLayoutPanel1.Controls)
  {
      Label iconLabel = control as Label;
      if (iconLabel != null)
      {
          int randomNumber = random.Next(icons.Count);
          iconLabel.Text = icons[randomNumber];
          // iconLabel.ForeColor = iconLabel.BackColor;
          icons.RemoveAt(randomNumber);
      }
  }
}
```

بإمكانك إدخال هذا الأسلوب أسفل التعليمات التي أضفتها في القسم السابق.

ملحوظة:

تم حذف أحد الأسطر عمدًا، يمكنك إضافته لاحقًا في هذا الإجراء.

يتكرر الأسلوب AssignIconsToSquares() خلال كل عنصر Label في TableLayoutPanel وتنفذ نفس العبارات لكل منها، وتسحب Labels أيقونة عشوائية من القائمة.

- يقوم السطر الأول بتحويل متغير التحكم إلى ملصق يسمى iconLabel
- السطر الثاني عبارة عن جملة if للتحقق من نجاح التحويل، إذا نجح التحويل فسيتم تنفيذ البيانات الموجودة في بيان if
- السطر الأول في if ينشئ متغير باسم randomNumber يحتوي على رقم عشوائي يتوافق مع أحد العناصر الموجودة في قائمة الأيقونات، ويستخدم هذا السطر الأسلوب المساعد Next() الخاص بالكائن Random ويقوم هذا الأسلوب Next() بإرجاع الرقم العشوائي، ويستخدم هذا السطر أيضًا خاصية Count الخاصة بقائمة الأيقونات لتحديد النطاق الذي سيتم اختيار الرقم العشوائي منه.
- يقوم السطر التالي بتعيين أحد عناصر قائمة الأيقونات إلى خاصية النص Text الخاصة بملصق التسمية Label
- السطر التالي يخفي الأيقونات. تم التعليق على السطر هنا حتى تتمكن من التحقق من بقية الكود قبل المتابعة.
- يؤدي السطر الأخير في جملة if إلى إزالة الأيقونة التي تمت إضافتها إلى النموذج من القائمة.

١. أضف استدعاءً الأسلوب `AssignIconsToSquares()` إلى منشئ `Form1()` يملأ هذه الأسلوب لوحة اللعبة بالأيقونات، يتم استدعاء المنشئين عند إنشاء كائن Form1.cs

```
public Form1()
 {
     InitializeComponent();

     AssignIconsToSquares();
 }
```

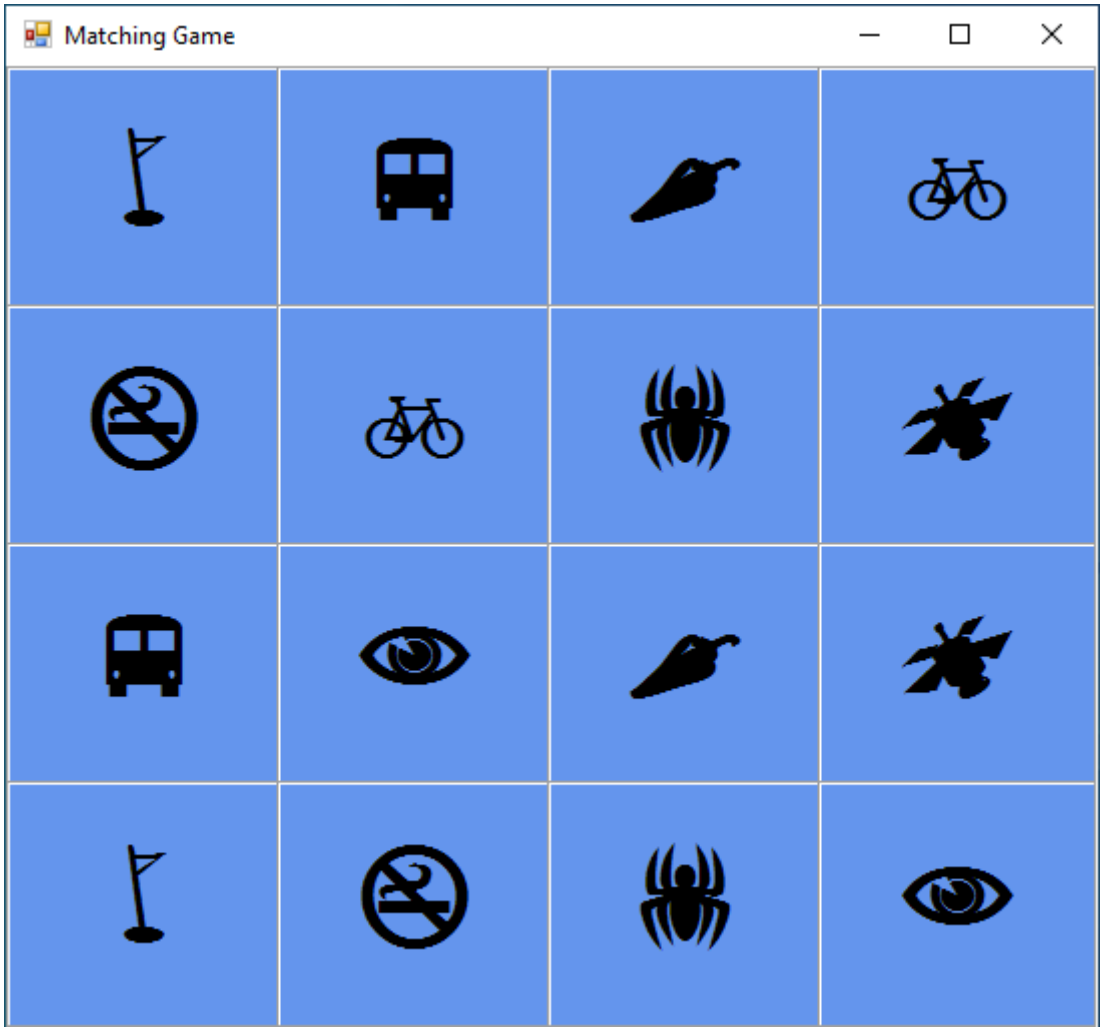
لمزيد من المعلومات، راجع المنشئين أو استخدام المنشئين والمدمرين دليل برمجة C#

٢. احفظ البرنامج وشغله، يجب أن يظهر نموذج به أيقونات عشوائية مخصصة لكل ملصق.

ملحوظة:

إذا لم يتم عرض أيقونات Webdings بشكل صحيح في النموذج، فقم بتعيين خاصية UseCompatibleTextRendering لملصقات Labels الموجودة في النموذج إلى True

٣. أغلق البرنامج، ثم قم بتشغيله مرة أخرى. يتم تعيين أيقونات مختلفة لكل ملصق.



أصبحت الأيقونات مرئية الآن لأنك لم تقم بإخفائها، لإخفائها عن المستخدم، يمكنك تعيين خاصية ForeColor لكل علامة بنفس لون خاصية BackColor الخاصة بها.

٤. أوقف البرنامج، قم بإزالة علامات التعليق لسطر التعليمات البرمجية المُعلّق عليه داخل عبارة if داخل الأسلوب AssignIconsToSquares()

```
iconLabel.ForeColor = iconLabel.BackColor;
```

إذا قمت بتشغيل البرنامج مرة أخرى، يبدو أن الأيقونات قد اختفت، تظهر خلفية زرقاء فقط، يتم تعيين الأيقونات بشكل عشوائي ولا تزال موجودة.

إضافة معالجات الأحداث إلى الملصقات Labels

في لعبة المطابقة هذه، يكشف اللاعب عن أيقونة مخفية، ثم أيقونة ثانية، إذا تطابقت الأيقونتان، فستظلان مرئيتين، وإلا، فسيتم إخفاء الأيقونتين مرة أخرى.

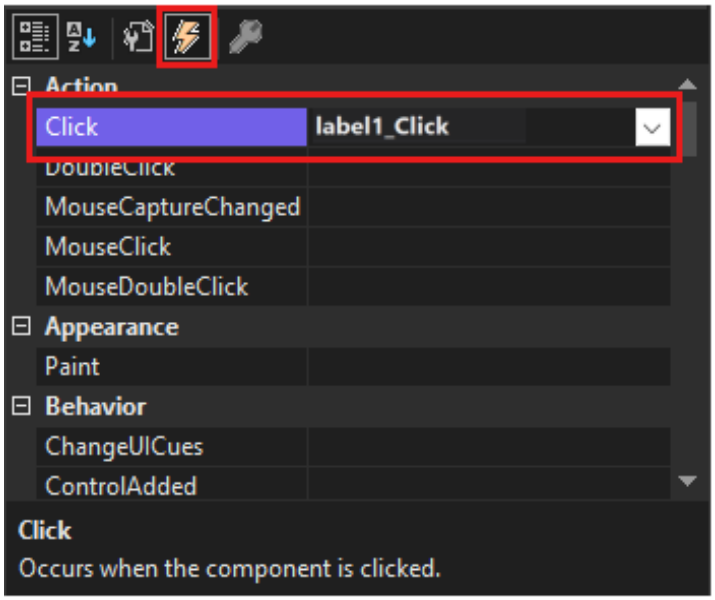
لجعل لعبتك تعمل بهذه الطريقة، أضف معالج حدث النقر Click الذي يغير لون الملصق المحدد ليتناسب مع الخلفية.

١. افتح النموذج في مصمم نماذج الويندوز Windows Forms Designer حدد Form1.cs

٢. اختر عنصر Label الأول لتحديده، ثم انقر فوقه نقرًا مزدوجًا لإضافة معالج حدث Click يسمي label1_Click() إلى ملف التعليمات البرمجية.

٣. ثم اضغط باستمرار على مفتاح Ctrl أثناء تحديد كل ملصق Label من Labels الأخرى، تأكد من تحديد كافة Labels

٤. في نافذة الخصائص، حدد زر الأحداث Events وهو عبارة عن علامة صاعقة، بالنسبة لحدث النقر Click event حدد label1_Click() في المربع.



٥. انقر مفتاح الإدخال Enter سيضيف IDE حدث Click إلى معالج الحدث المسمى label1_Click() في التعليمات الموجودة في ملف Form1.cs نظرًا لأنك حددت جميع العلامات، فإن المعالج مرتبط بكل علامة موجودة.

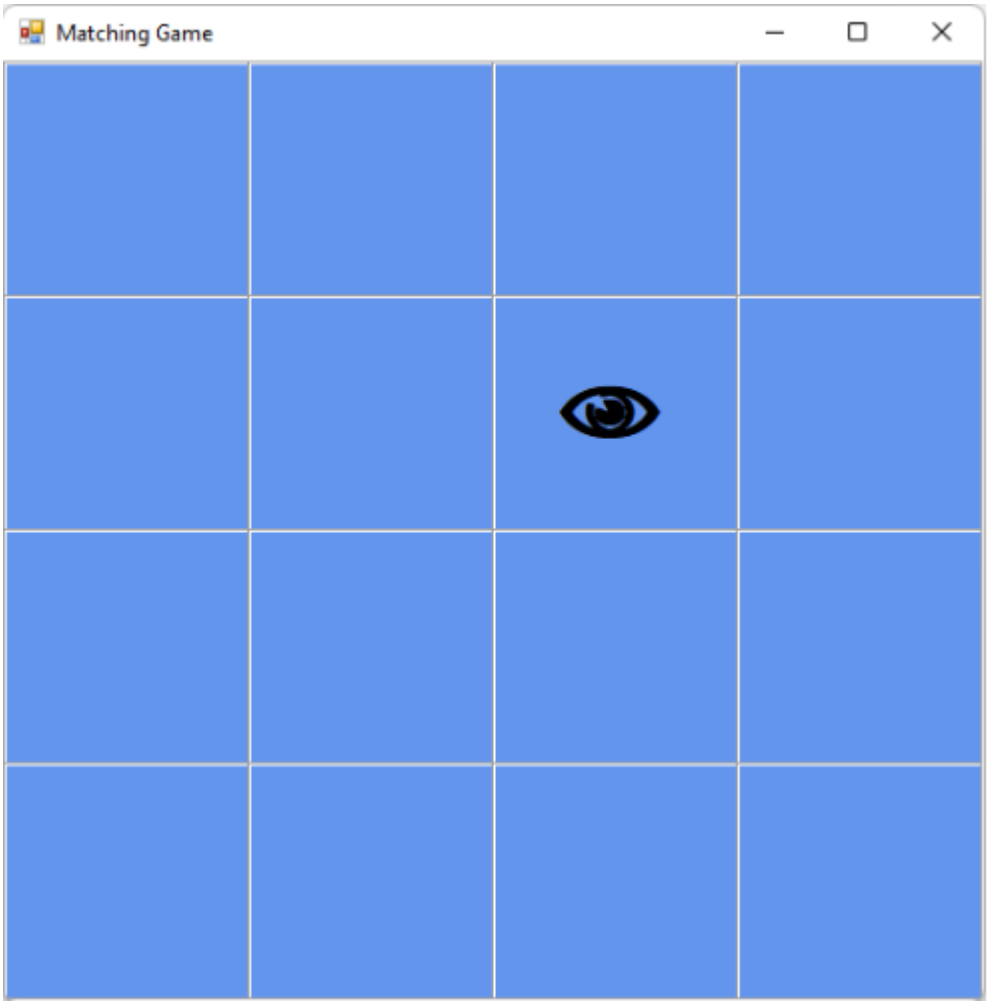
٦. املأ بقية التعليمات:

```
/// <summary>
/// Every label's Click event is handled by this event handler
/// </summary>
/// <param name="sender">The label that was clicked</param>
/// <param name="e"></param>
private void label1_Click(object sender, EventArgs e)
{
  Label clickedLabel = sender as Label;

  if (clickedLabel != null)
  {
      // If the clicked label is black, the player clicked
      // an icon that's already been revealed --
      // ignore the click
    if (clickedLabel.ForeColor == Color.Black)
        return;

      clickedLabel.ForeColor = Color.Black;
  }
}
```

ملحوظة:

إذا قمت بنسخ ولصق كتلة التعليمات label1_Click() بدلًا من إدخال التعليمات يدويًا، فتأكد من استبدال كتلة label1_Click() الموجودة، والا فستحصل في النهاية على كتلة تعليمات برمجية مكررة.

اختر Debug > Start Debugging لتشغيل البرنامج، يجب أن ترى نموذجًا فارغًا بخلفية زرقاء، اختر أي من الخلايا في النموذج، ستظهر أحد الأيقونات، استمر في اختيار أماكن مختلفة في النموذج، يجب أن تظهر الأيقونات أثناء اختيارك لها.



إضافة متغيرات مرجعية reference variables ومؤقت timer إلى اللعبة

يحتاج برنامج لعبة المطابقة الخاص بك إلى تتبع عناصر التحكم في الملصقات Labels التي يختارها اللاعب، بعد أن يختار اللاعب الملصق الأول، يجب أن يعرض البرنامج الأيقونة، بعد اختيار الملصق الثاني، يجب أن يعرض البرنامج كلا الأيقونتين لفترة قصيرة، ثم يقوم بإخفاء كلا الأيقونتين.

يتتبع برنامجك الملصق الذي تختاره أولًا، وثانيًا باستخدام متغيرات مرجعية reference variables ويخفي المؤقت الأيقونات، ويتحكم في المدة التي تظهر فيها الأيقونات.

سنقوم بإنشاء:

- إضافة مراجع ملصقات label references
- إضافة مؤقت timer

إضافة مراجع ملصقات label references

في هذا القسم، ستضيف متغيرين مرجعيين إلى التعليمات الخاصة بك، وهما يتتبعان كائنات labels أو يشيران إليها.

١. قم بإضافة مراجع الملصق إلى النموذج، باستخدام التعليمات التالية في Form1.cs

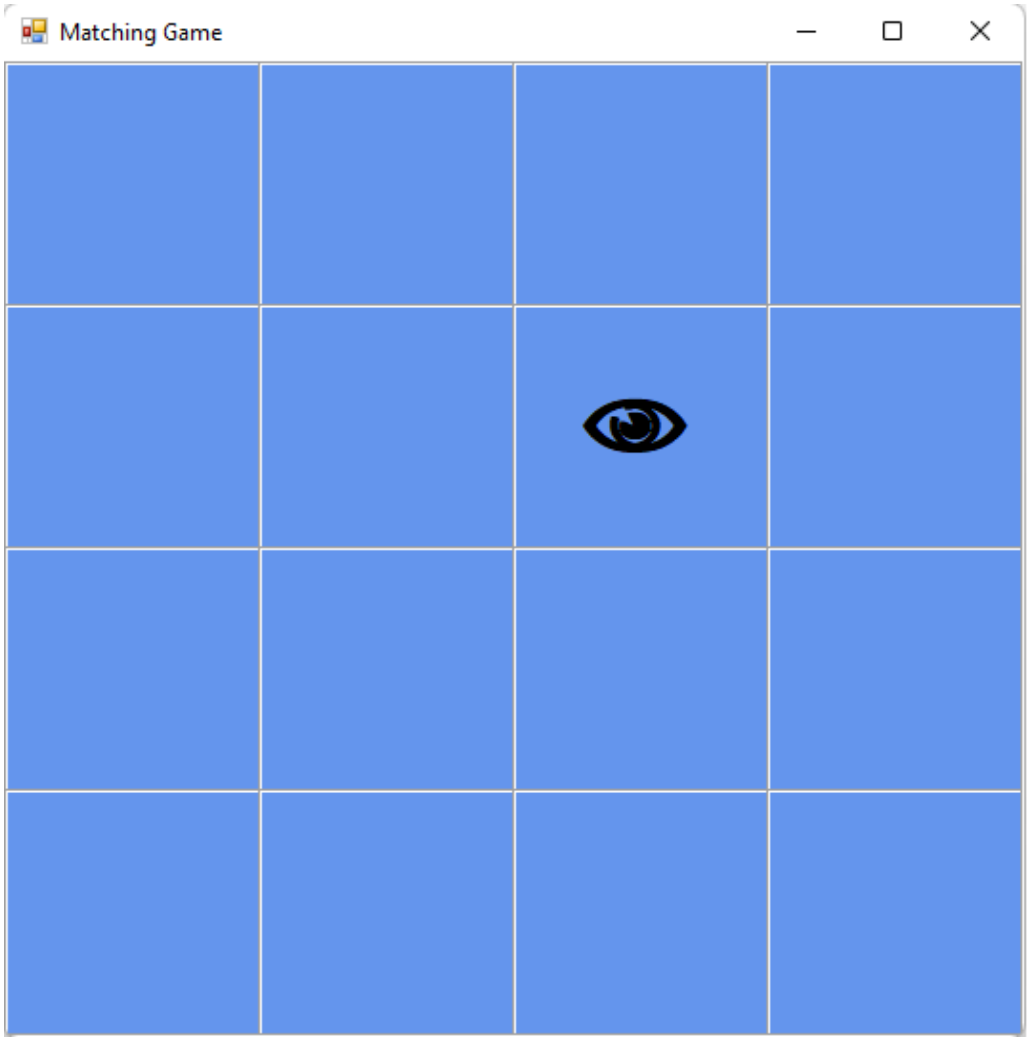
```
public partial class Form1 : Form
{
    // firstClicked points to the first Label control
    // that the player clicks, but it will be null
    // if the player hasn't clicked a label yet
     Label firstClicked = null;

    // secondClicked points to the second Label control
     // that the player clicks
     Label secondClicked = null;
```

لا تتسبب هذه العبارات في ظهور عناصر التحكم Label في النموذج لأنه لا توجد كلمة `new` الأساسية، عند بدء تشغيل البرنامج، يتم تعيين كل من `firstClicked` and `secondClicked` على `null`

٢. قم بتعديل معالج حدث النقر Click event handler لاستخدام متغير المرجع `firstClicked` الجديد، أزل الجملة الأخيرة من أسلوب معالج الحدث `label1_Click()` وهي (clickedLabel.ForeColor = Color.Black;) واستبدلها بعبارة if التالية:

```
/// <summary>
/// Every label's Click event is handled by this event handler
/// </summary>
/// <param name="sender">The label that was clicked</param>
/// <param name="e"></param>
private void label1_Click(object sender, EventArgs e)
{
  Label clickedLabel = sender as Label;

  if (clickedLabel != null)
  {
  // If the clicked label is black, the player clicked
  // an icon that's already been revealed --
  // ignore the click
  if (clickedLabel.ForeColor == Color.Black)
      return;

  // If firstClicked is null, this is the first icon
  // in the pair that the player clicked,
  // so set firstClicked to the label that the player
  // clicked, change its color to black, and return
   if (firstClicked == null)
   {
      firstClicked = clickedLabel;
      firstClicked.ForeColor = Color.Black;

      return;
   }
  }
}
```

٣. احفظ البرنامج وشغله، اختر أحد عناصر الملصقات، وستظهر الأيقونة الخاصة به، اختر عنصر آخر، ولاحظ أنه لم يحدث شيء.



تظهر فقط الأيقونة الأولى التي تم اختيارها، أما الأيقونات الأخرى فهي غير مرئية.

يقوم البرنامج بالفعل بتتبع الملصق الأول الذي اختاره اللاعب، المرجع reference `firstClicked` قيمته ليست `null`

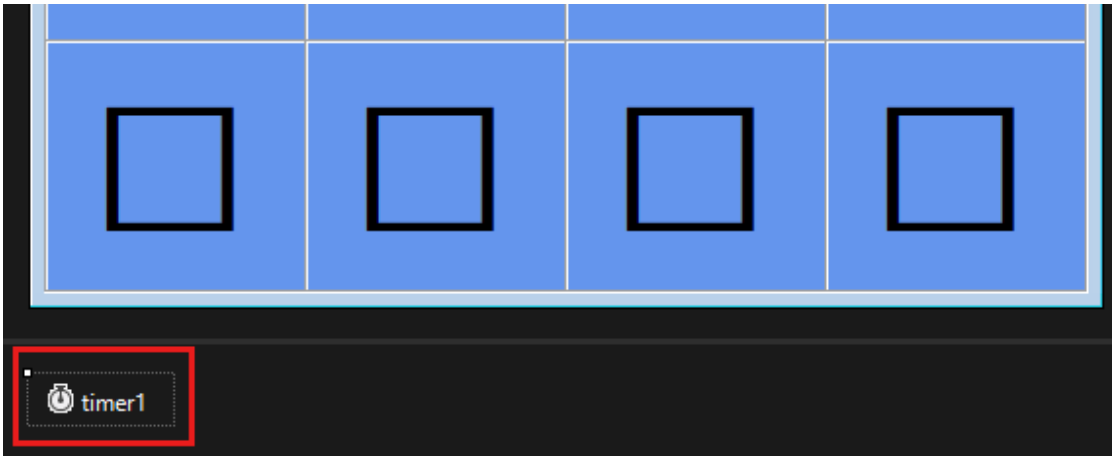
عندما تجد عبارة if أن `firstClicked` لا يساوي `null` فسيتم تنفيذ تعليمات العبارة the statements

إضافة مؤقت timer

يستخدم تطبيق Matching Game عنصر Timer ينتظر المؤقت، ثم يطلق حدثًا، يشار إليه باسم النقرة يمكن للمؤقت بدء إجراء أو تكرار إجراء بانتظام.

في برنامجك، يتيح المؤقت للاعب اختيار أيقونتين، إذا لم تتطابق الأيقونتان، فسيتم إخفاءهم مرة أخرى، بعد وقت قصير.

١. حدد علامة التبويب صندوق الأدوات Toolbox في فئة المكونات Components انقر نقرًا مزدوجًا فوق مكون Timer أو اسحبه إلى النموذج، تظهر أيقونة المؤقت، المسمى timer1 في المساحة أسفل النموذج.



٢. انقر فوق أيقونة timer1 لتحديده، من نافذة الخصائص Properties حدد زر Properties لعرض الخصائص.

٣. اضبط خاصية الفاصل الزمني Interval على 750 وهو ما يعادل 750 ميلي ثانية milliseconds

تخبر خاصية الفاصل الزمني Interval المؤقت بمدة الانتظار بين الملصقات، عندما يتم تشغيل حدث Tick الخاص به، يستدعي برنامجك أسلوب Start() لبدء تشغيل المؤقت بعد أن يختار اللاعب الملصق الثاني.

٤. حدد أيقونة المؤقت timer1 ثم اضغط على Enter أو انقر نقرًا مزدوجًا فوق الأيقونة، يضيف ل معالج حدث فارغ empty **Tick** event handler إلى Form1.cs قم باستبداله بالتعليمات التالية:

```
/// <summary>
/// This timer is started when the player clicks
/// two icons that don't match,
/// so it counts three quarters of a second
/// and then turns itself off and hides both icons
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void timer1_Tick(object sender, EventArgs e)
{
    // Stop the timer
    timer1.Stop();
```

```
    // Hide both icons
  firstClicked.ForeColor = firstClicked.BackColor;
 secondClicked.ForeColor = secondClicked.BackColor;

    // Reset firstClicked and secondClicked
    // so the next time a label is
   // clicked, the program knows it's the first click
    firstClicked = null;
    secondClicked = null;
}
```

يقوم معالج حدث Tick بثلاثة أشياء:

- يتأكد من عدم تشغيل المؤقت عن طريق استدعاء أسلوب Stop()
- ويستخدم متغيرين مرجعيين firstClicked and secondClicked لجعل أيقونات الملصقين اللذين اختارهما اللاعب غير مرئية مرة أخرى.
- يقوم بإعادة تعيين المتغيرات المرجعية firstClicked and secondClicked إلى null

٥. انتقل إلى محرر التعليمات البرمجية وأضف التعليمات إلى أعلى وأسفل أسلوب معالج الأحداث label1_Click() ستقوم هذه التعليمات بالتحقق مما إذا كان المؤقت ممكّنًا، وتعيين المتغير المرجعي secondClicked وبدء تشغيل المؤقت، يجب ان يبدو الآن أسلوب معالج الحدث label1_Click() كالتالي:

```
/// <summary>
/// Every label's Click event is handled by this event handler
/// </summary>
/// <param name="sender">The label that was clicked</param>
/// <param name="e"></param>
private void label1_Click(object sender, EventArgs e)
{
  // The timer is only on after two non-matching
  // icons have been shown to the player,
  // so ignore any clicks if the timer is running
  if (timer1.Enabled == true)
      return;

  Label clickedLabel = sender as Label;

  if (clickedLabel != null)
  {
    // If the clicked label is black, the player clicked
```

```
        // an icon that's already been revealed --
        // ignore the click
        if (clickedLabel.ForeColor == Color.Black)
            return;

        // If firstClicked is null, this is the first icon
        // in the pair that the player clicked,
        // so set firstClicked to the label that the player
        // clicked, change its color to black, and return
        if (firstClicked == null)
        {
            firstClicked = clickedLabel;
            firstClicked.ForeColor = Color.Black;
            return;
        }

        // If the player gets this far, the timer isn't
        // running and firstClicked isn't null,
        // so this must be the second icon the player clicked
        // Set its color to black
        secondClicked = clickedLabel;
        secondClicked.ForeColor = Color.Black;

        // If the player gets this far, the player
        // clicked two different icons, so start the
        // timer (which will wait three quarters of
        // a second, and then hide the icons)
        timer1.Start();
    }
}
```

- تتحقق التعليمات الموجود أعلى الأسلوب من بدء تشغيل المؤقت، من خلال التحقق من قيمة الخاصية Enabled إذا اختار اللاعب ملصقي `Label` الأول والثاني وبدأ تشغيل المؤقت، فلن يؤدي اختيار `Label` ثالث إلى أي شيء.
- تحدد التعليمات الموجودة في أسفل الأسلوب متغير المرجع secondClicked لتتبع عنصر `Label` الثاني، ثم يقوم بضبط لون أيقونة الملصق إلى الأسود، لجعله مرئيًا، ثم يبدأ تشغيل المؤقت في وضع اللقطة الواحدة، بحيث ينتظر لمدة 750 ميلي ثانية ثم يطلق نبضة واحدة، يقوم معالج حدث Tick الخاص بالمؤقت بإخفاء الأيقونتين، وإعادة تعيين متغيري المرجع firstClicked and secondClicked ليصبح النموذج جاهزًا للاعب لاختيار زوج آخر من الأيقونات.

٦. احفظ البرنامج وشغله، حدد مربعًا وستظهر الأيقونة، اختر مربعًا آخر، تظهر الأيقونة لفترة قصيرة ثم تختفي الأيقونتان.

يتتبع برنامجك الآن الأيقونتين الأولى والثانية اللذين تختارهما، ويستخدم المؤقت للتوقف مؤقتًا قبل اختفاء الأيقونتين.

عرض رسالة تهنئة عند فوز اللاعب

ستقوم بمراجعة لعبة المطابقة الخاصة بك، لإبقاء أزواج الأيقونات المتطابقة مرئية، ولعرض رسالة تهنئة عندما يفوز لاعب ما.

في هذا القسم التعليمي، سوف تتعلم كيفية:

- إبقاء أزواج الأيقونات مرئية
- التحقق من فوز اللاعب
- تجربة ميزات أخرى

إبقاء أزواج الأيقونات مرئية

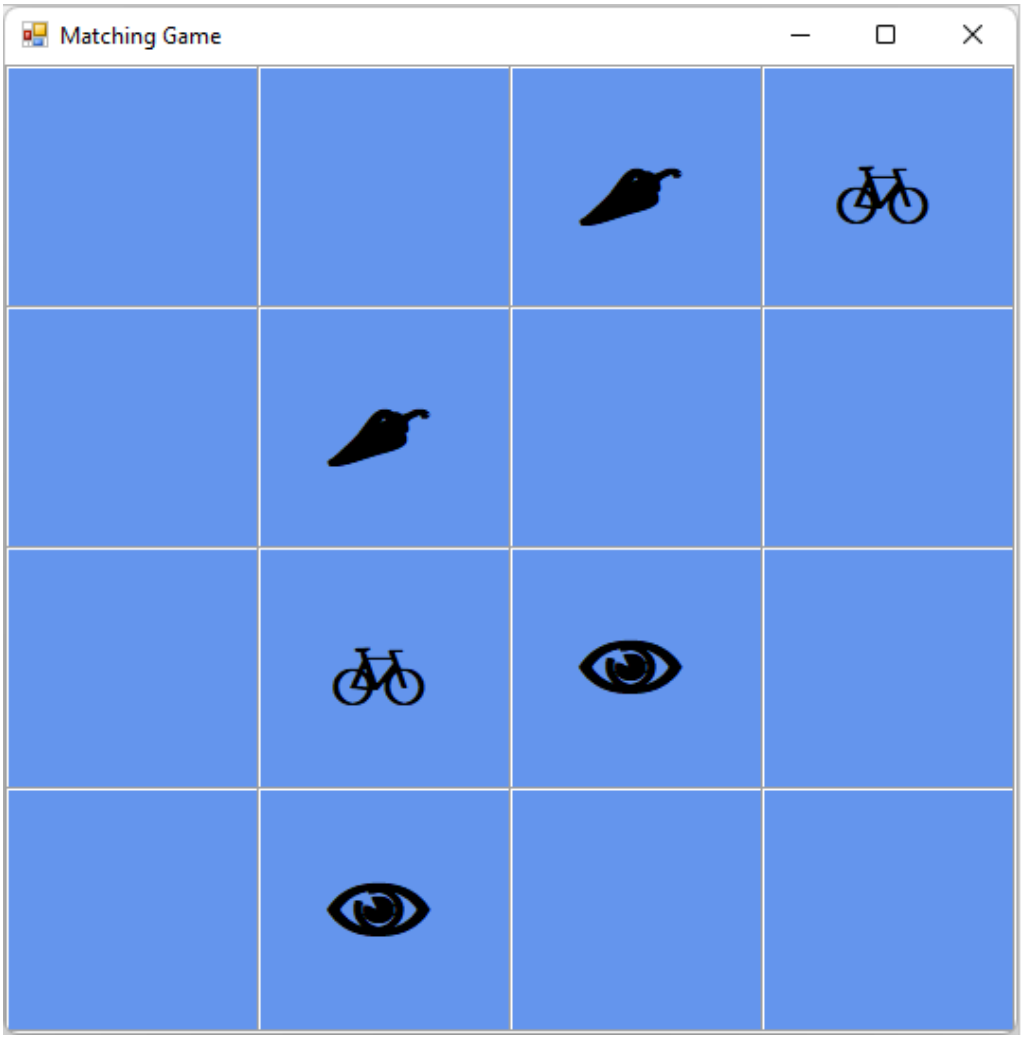
عندما يطابق أحد اللاعبين زوجًا، يجب أن تعيد اللعبة ضبط نفسها بحيث لا تحتفظ بعدها بأي labels تستخدم متغيرات المرجع firstClicked and secondClicked لا ينبغي لها إعادة ضبط ألوان الملصقين اللتين تمت مطابقتهما، يستمر عرض هذه الملصقات.

١. أضف عبارة if التالية إلى أسلوب معالج الحدث `label1_Click()` ضعها في نهاية التعليمات أعلى العبارة التي تبدأ فيها تشغيل المؤقت timer مباشرةً.

```
// If the player gets this far, the timer isn't
// running and firstClicked isn't null,
// so this must be the second icon the player clicked
// Set its color to black
secondClicked = clickedLabel;
secondClicked.ForeColor = Color.Black;

// If the player clicked two matching icons, keep them
// black and reset firstClicked and secondClicked
// so the player can click another icon
if (firstClicked.Text == secondClicked.Text)
{
    firstClicked = null;
    secondClicked = null;
    return;
}

// If the player gets this far, the player
// clicked two different icons, so start the
// timer (which will wait three quarters of
// a second, and then hide the icons)
timer1.Start();
```

تتحقق عبارة if مما إذا كانت الأيقونة الموجودة في الملصق الأول التي يختارها اللاعب هي نفس الأيقونة الموجودة في الملصق الثاني، إذا كانت الأيقونات متماثلة، يقوم البرنامج بتشغيل عباراته الثلاثة، تقوم العبارتان الأوليتان بإعادة تعيين متغيرات المرجع firstClicked and secondClicked لم يعد بإمكانهم تتبع أي من العلامات، العبارة الثالثة هي إرجاع return والتي تتخطى بقية العبارات في الأسلوب دون تشغيلها.

٢. قم بتشغيل البرنامج، ثم ابدأ باختيار المربعات الموجودة في النموذج.



إذا اخترت زوجًا لا يتطابقان، فسيتم تشغيل حدث Tick الخاص بالمؤقت، تختفي كلا الأيقونتين.

إذا قمت باختيار زوج متطابق، فسيتم تشغيل عبارة if الجديدة، تتسبب عبارة الإرجاع return statement في تخطي الأسلوب للتعليمات التي تشغل المؤقت، تظل الأيقونات مرئية.

التحقق من فوز اللاعب

لقد أنشأت لعبة ممتعة، يجب أن تنتهي اللعبة بعد فوز لاعب ما، يضيف هذا القسم أسلوب للتحقق مما إذا كان اللاعب قد فاز.

١. أضف أسلوب `CheckForWinner()` إلى أسفل التعليمات، أسفل معالج حدث timer1_Tick()

```
/// <summary>
/// Check every icon to see if it is matched, by
/// comparing its foreground color to its
background color.
/// If all of the icons are matched, the player wins
/// </summary>
private void CheckForWinner()
{
   // Go through all of the labels in the TableLayoutPanel,
  // checking each one to see if its icon is matched
 foreach (Control control in tableLayoutPanel1.Controls)
  {
      Label iconLabel = control as Label;

     if (iconLabel != null)
     {
      if (iconLabel.ForeColor == iconLabel.BackColor)
             return;
      }
  }

  // If the loop didn’t return, it didn't find
  // any unmatched icons
  // That means the user won. Show a message and close the form
 MessageBox.Show("You matched all the icons!",
"Congratulations");
    Close();
}
```

يستخدم الأسلوب حلقة `foreach` للانتقال عبر كل ملصق في TableLayoutPanel وتتحقق من لون أيقونة كل ملصق للتأكد من تطابقها مع الخلفية، إذا تطابقت الألوان، تظل الأيقونة غير مرئية، ولم يطابق اللاعب كل الأيقونات المتبقية.

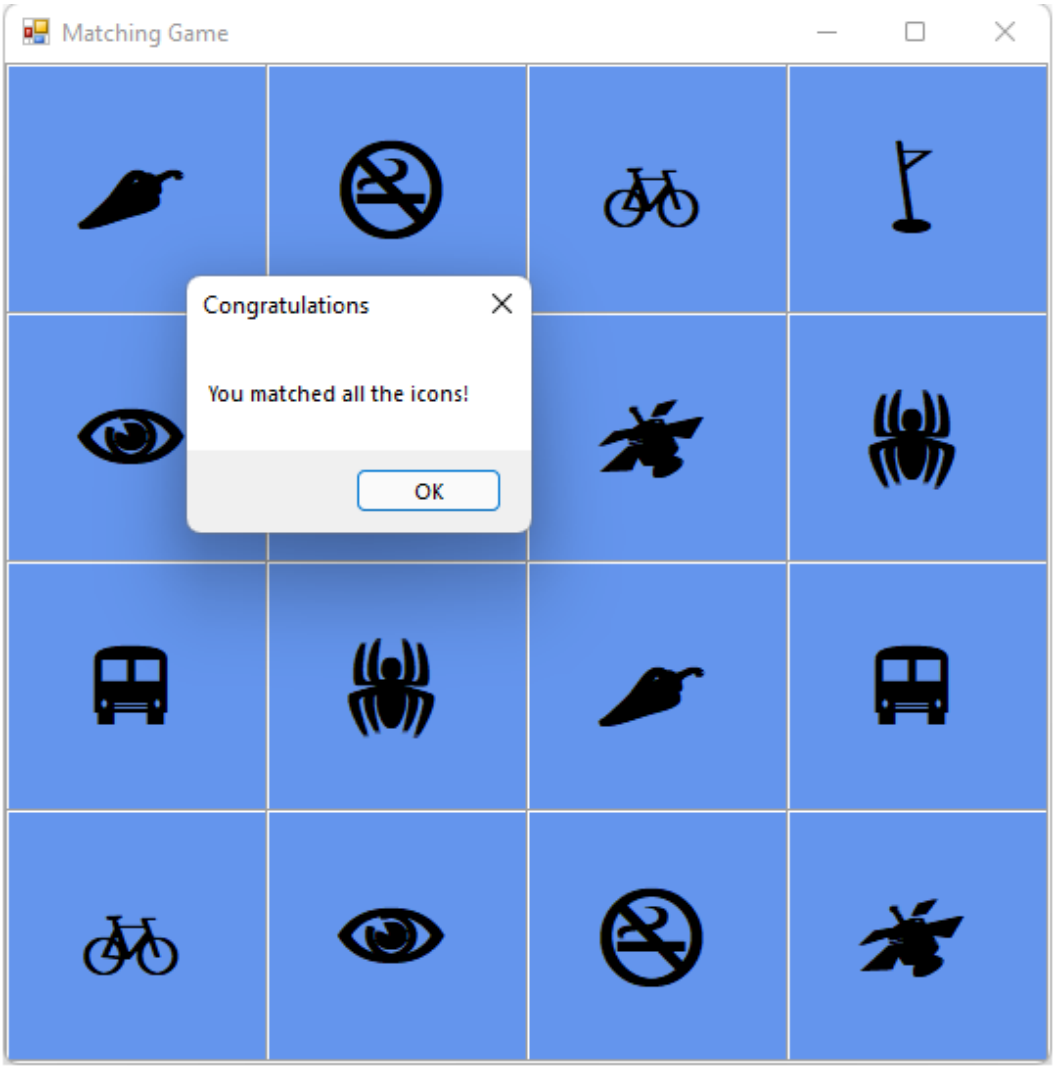
في هذه الحالة، يستخدم البرنامج `return;` لتخطي بقية الأسلوب إذا مرت الحلقة عبر جميع الملصقات دون تنفيذ return فهذا يعني أن جميع الأيقونات الموجودة في النموذج قد تمت مطابقتها، يعرض البرنامج مربع رسالة MessageBox لتهنئة اللاعب على الفوز، ثم يستدعي الأسلوب لإنهاء اللعبة.

٢. اطلب من معالج حدث النقر Click event الخاص بالملصق label استدعاء الأسلوب الجديد `CheckForWinner()`

```
secondClicked = clickedLabel;
secondClicked.ForeColor = Color.Black;

// Check to see if the player won
CheckForWinner();

// If the player clicked two matching icons, keep them
// black and reset firstClicked and secondClicked
// so the player can click another icon
if (firstClicked.Text == secondClicked.Text)
{
    firstClicked = null;
    secondClicked = null;
    return;
}
```

تأكد من أن البرنامج يتحقق من الفائز فورًا، بعد إظهار الأيقونة الثانية التي يختارها اللاعب، ابحث عن السطر الذي يحدد فيه لون الأيقونة الثانية المختارة، ثم استدعي الأسلوب `CheckForWinner()` بعد هذا السطر مباشرةً.

٣. احفظ البرنامج وشغله، العب اللعبة وطابق كل الأيقونات، عندما تفوز، يعرض البرنامج رسالة التهنئة.



بعد تحديد موافق OK سيتم إغلاق لعبة المطابقة.

تجربة ميزات أخرى

لقد اكتملت لعبة المطابقة الخاصة بك. يمكنك إضافة المزيد من الميزات لجعل هذه اللعبة أكثر تحديًا وإثارة للاهتمام. إليك بعض الخيارات.

- استبدل الأيقونات والألوان بتلك التي تختارها.

حاول إلقاء نظرة على خاصية ForeColor للعلامات Labels

- أضف مؤقتًا للعبة يحتسب المدة التي يستغرقها اللاعب للفوز.

يمكنك إضافة ملصق لعرض الوقت المنقضي، ضعها في TableLayoutPanel أضف مؤقتًا آخر إلى النموذج لتتبع الوقت، استخدم التعليمات لبدء تشغيل المؤقت عندما يبدأ اللاعب اللعبة، وأوقف المؤقت بعد مطابقة الأيقونتين الأخيرين.

- أضف صوتًا عندما يجد اللاعب تطابقًا، وصوتًا آخر عندما يكتشف اللاعب أيقونتين غير متطابقين، وصوتًا ثالثًا عندما يقوم البرنامج بإخفاء الأيقونات مرة أخرى.

لتشغيل الأصوات، يمكنك استخدام مساحة اسم System.Media لمزيد من المعلومات، راجع تشغيل الأصوات في تطبيقات C#

- اجعل اللعبة أكثر صعوبة عن طريق جعل اللوحة أكبر.

سوف تحتاج إلى القيام بأكثر من مجرد إضافة صفوف وأعمدة إلى TableLayoutPanel كما تحتاج إلى مراعاة عدد الأيقونات التي تقوم بإنشائها.

- اجعل اللعبة أكثر تحديًا عن طريق إخفاء الأيقونة الأولى إذا كان اللاعب بطيئًا جدًا في الاستجابة للعبة.

تهانينا! لقد أكملت هذه السلسلة من الدروس التعليمية، لقد قمت بتنفيذ مهام البرمجة والتصميم التالية في بيئة التطوير المتكاملة لبرنامج Visual Studio IDE

- الكائنات المخزنة، مثل الأيقونات، في قائمة Stored objects, such as icons, in a list
- تم استخدام حلقة للتكرار خلال قائمة Used a loop in C# to iterate through a list
- تم تتبع الحالة باستخدام المتغيرات المرجعية Kept track of state by using reference variables
- تم إنشاء معالج حدث للاستجابة للأحداث الخاصة بالعديد من الكائنات Built an event handler to respond to events for multiple objects
- تمت إضافة مؤقت يقوم بالعد التنازلي وإطلاق حدث Added a timer that counts down and fires an event

٧ الاتصال بقاعدة بيانات Access في تطبيقات .NET Framework.

ملاحظة

مجموعات البيانات والفئات ذات الصلة Datasets and related classes are legacy .NET Framework technologies هي تقنيات القديمة من أوائل عام 2000 والتي تمكن التطبيقات من العمل مع البيانات الموجودة في الذاكرة أثناء فصل التطبيقات عن قاعدة البيانات. التقنيات مفيدة بشكل خاص للتطبيقات التي تمكن المستخدمين من تعديل البيانات واستمرار التغييرات مرة أخرى في قاعدة البيانات. على الرغم من أن مجموعات البيانات أثبتت أنها تقنية ناجحة للغاية، إلا أننا نوصي بأن تستخدم تطبيقات .NET الجديدة Entity Framework Core يوفر Entity Framework طريقة أكثر طبيعية للعمل مع البيانات الجدولية كنماذج كائنات، وله واجهة برمجة أبسط.

يمكنك الاتصال بقاعدة بيانات Access (either an .mdb file or an .accdb file) باستخدام Visual Studio بعد تعريف الاتصال، تظهر البيانات في النافذة مصادر البيانات Data Sources من هناك، يمكنك سحب الجداول أو طرق العرض إلى سطح التصميم الخاص بك.

ملاحظة

إذا كنت تستخدم Visual Studio للاتصال بقواعد بيانات Access فستحتاج إلى أن تدرك أن إصدارات Visual Studio السابقة ل Visual Studio 2022 كلها عمليات 32Bit وهذا يعني أن بعض أدوات البيانات في Visual Studio 2019 والإصدارات السابقة ستتمكن فقط من الاتصال بقواعد بيانات Access باستخدام موفري بيانات 32Bit

إذا كنت تستخدم Visual Studio 2022 للاتصال بقواعد بيانات Access فستحتاج إلى إدراك أن Visual Studio 2022 هو الآن عملية 64Bit وهذا يعني أن بعض أدوات البيانات في Visual Studio لن تتمكن من الاتصال بقواعد بيانات Access باستخدام موفري البيانات 32Bit

إذا كنت بحاجة إلى صيانة تطبيقات 32Bit التي تتصل بقواعد بيانات Access فستظل قادرًا على إنشاء التطبيق وتشغيله باستخدام Visual Studio 2022 ومع ذلك، إذا كنت بحاجة إلى استخدام أي من أدوات بيانات Visual Studio مثل Server Explorer, Data Source Wizard, or the DataSet Designer فستحتاج إلى استخدام إصدار سابق من Visual Studio لا يزال عملية 32Bit كان الإصدار الأخير من Visual Studio الذي كان عملية 32Bit هو Visual Studio 2019

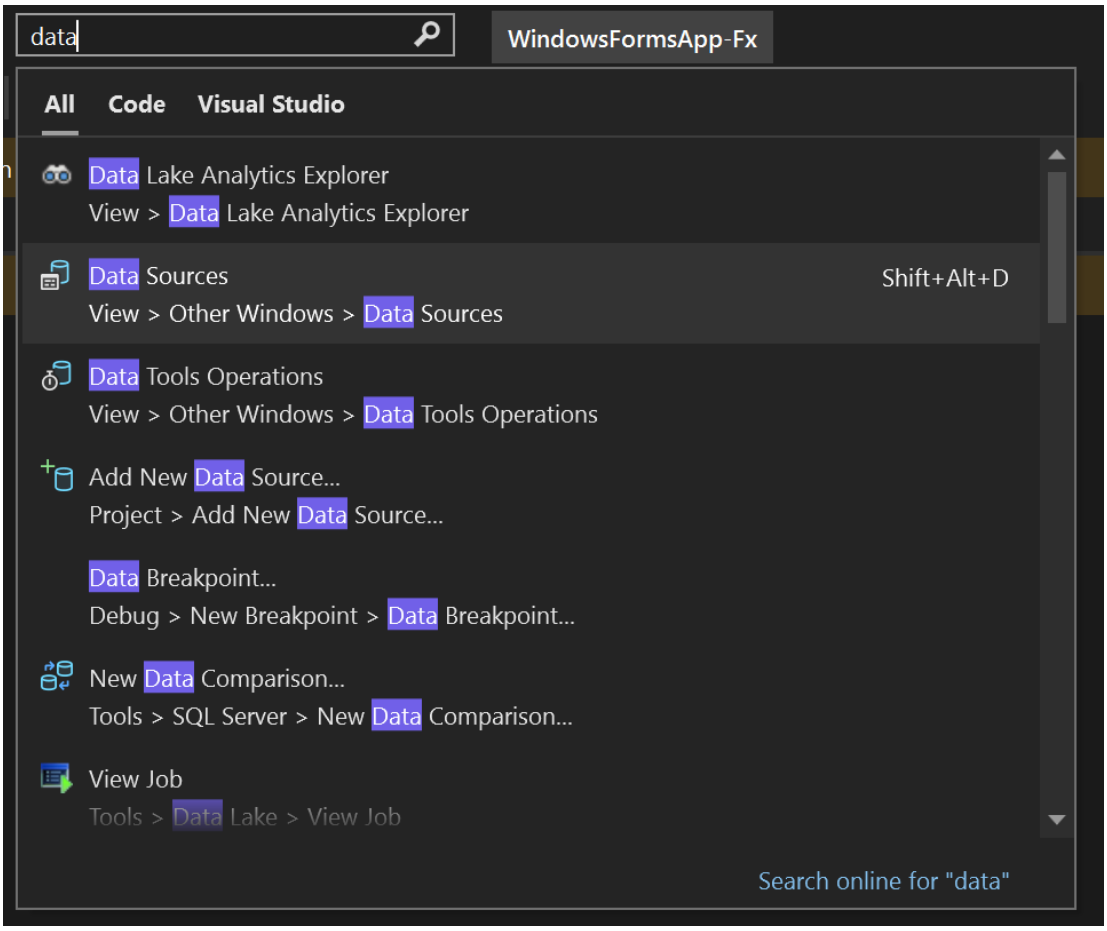
إذا كنت تخطط لتحويل المشروع إلى عملية 64Bit فمن المستحسن استخدام مشغل قاعدة بيانات Microsoft Access 64 Bit ويسمى أيضا مشغل اتصال الوصول (ACE) الرجاء مراجعة OLE DB Provider for Jet and ODBC driver are 32-bit versions only لمزيد من المعلومات.

إنشاء مجموعة بيانات dataset for an .accdb

اتصل بقواعد البيانات التي تم إنشاؤها باستخدام Microsoft 365 أو Access 2016 أو Access 2013 أو Access 2010 أو Access 2007 باستخدام الإجراء التالي:

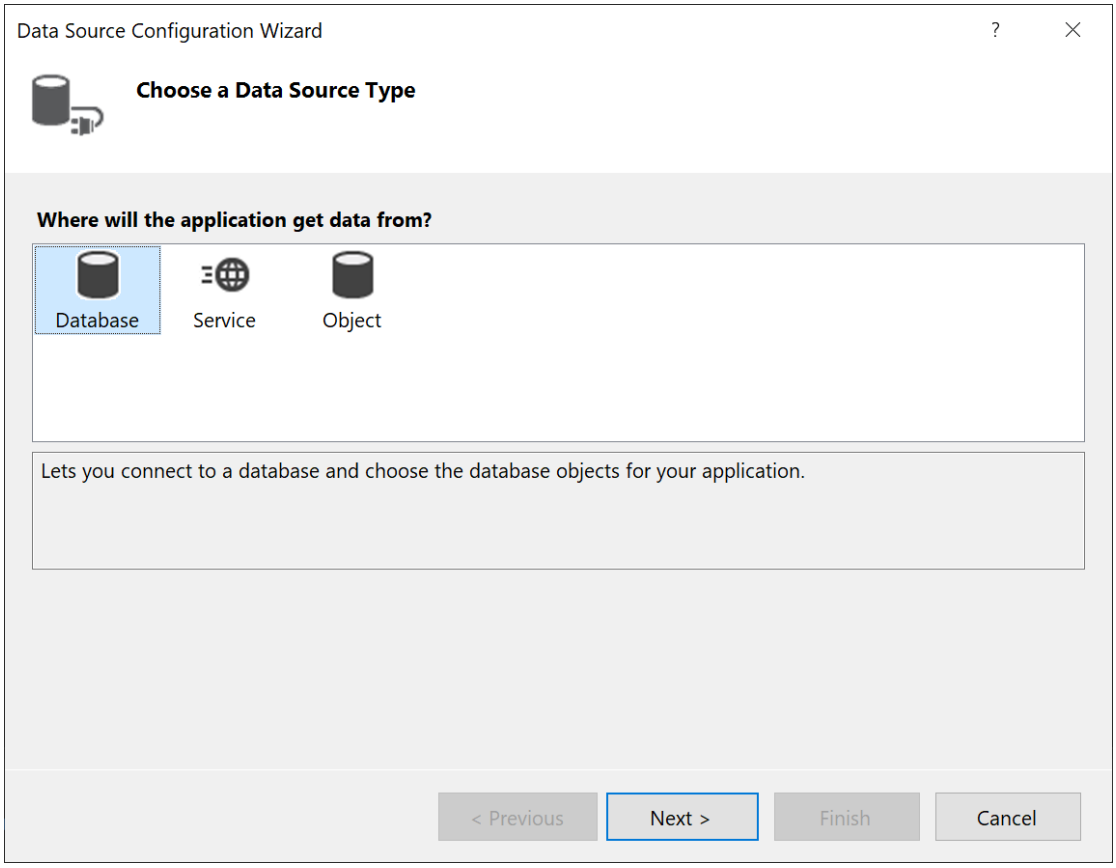
١. افتح مشروع تطبيق نماذج Windows أو WPF في Visual Studio

٢. لفتح نافذة مصادر البيانات **Data Sources** اضغط على **Ctrl+Q** وأدخل "data" في مربع البحث، واختر نافذة مصادر البيانات **Data Sources** أو من القائمة عرض **View** حدد **Other Windows > Data Sources** أو على لوحة المفاتيح، اضغط على **Shift+Alt+D**



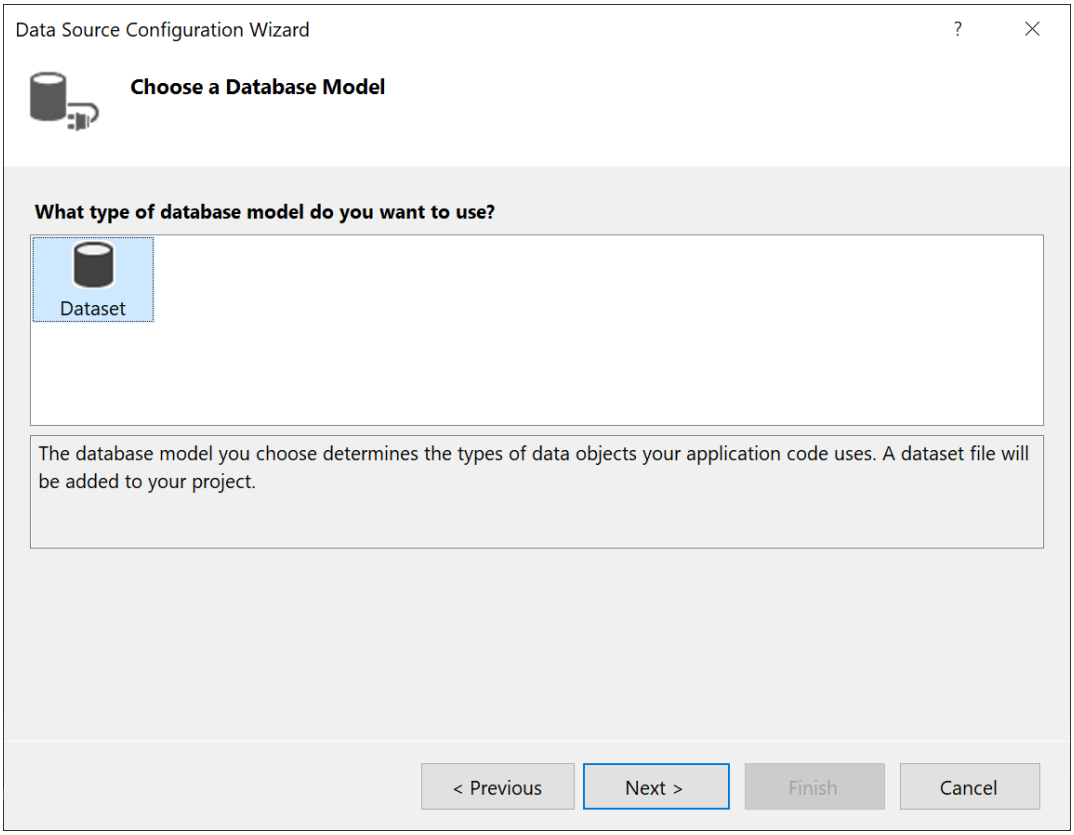
٣. في نافذة مصادر البيانات Data Sources انقر فوق إضافة مصدر بيانات جديد Add New Data Source

يفتح معالج تكوين مصدر البيانات Data Source Configuration Wizard

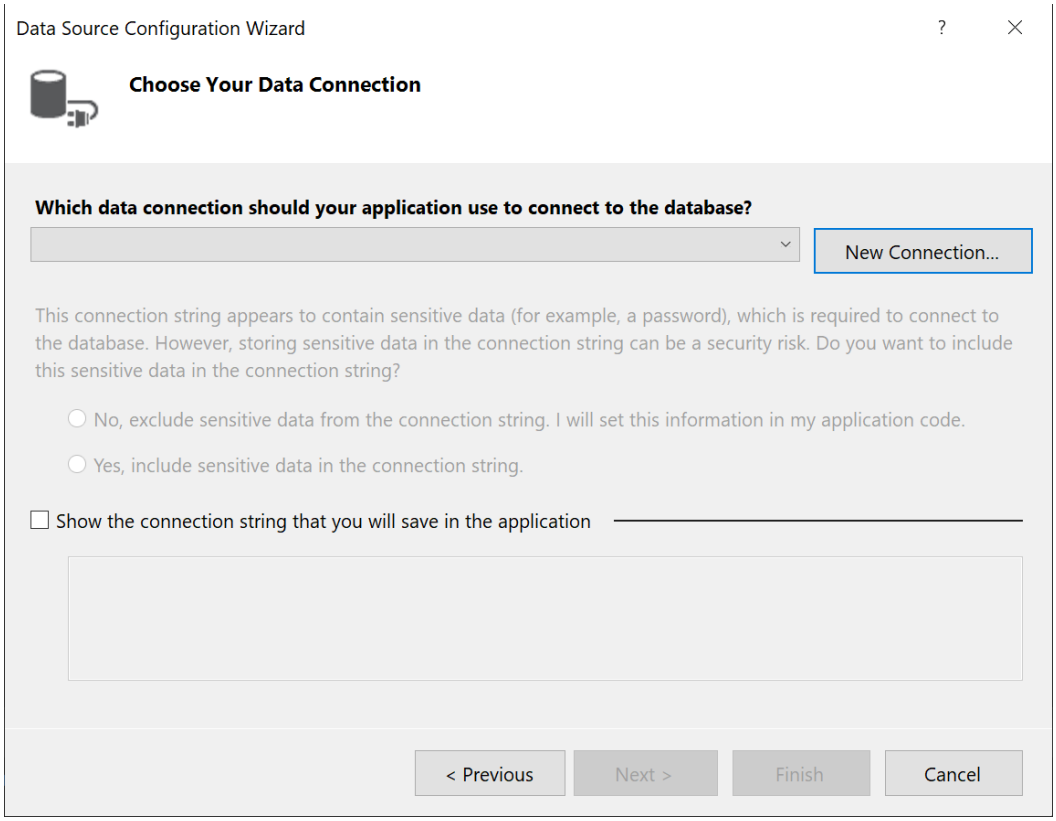


٤. حدد قاعدة البيانات Database في الصفحة اختيار نوع مصدر بيانات Choose a Data Source Type ثم حدد التالي Next

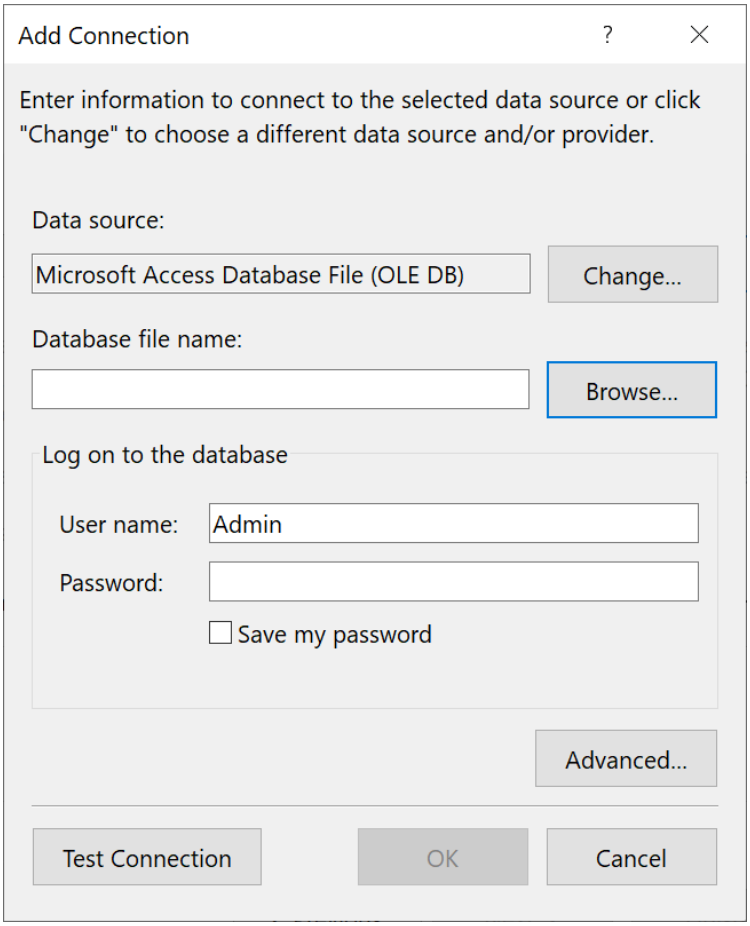
٥. حدد مجموعة البيانات Dataset في الصفحة اختيار نموذج قاعدة بيانات Choose a Database Model ثم حدد التالي Next



٦. في الصفحة اختيار اتصال البيانات Choose your Data Connection
حدد اتصال جديد New Connection لتكوين اتصال بيانات جديد

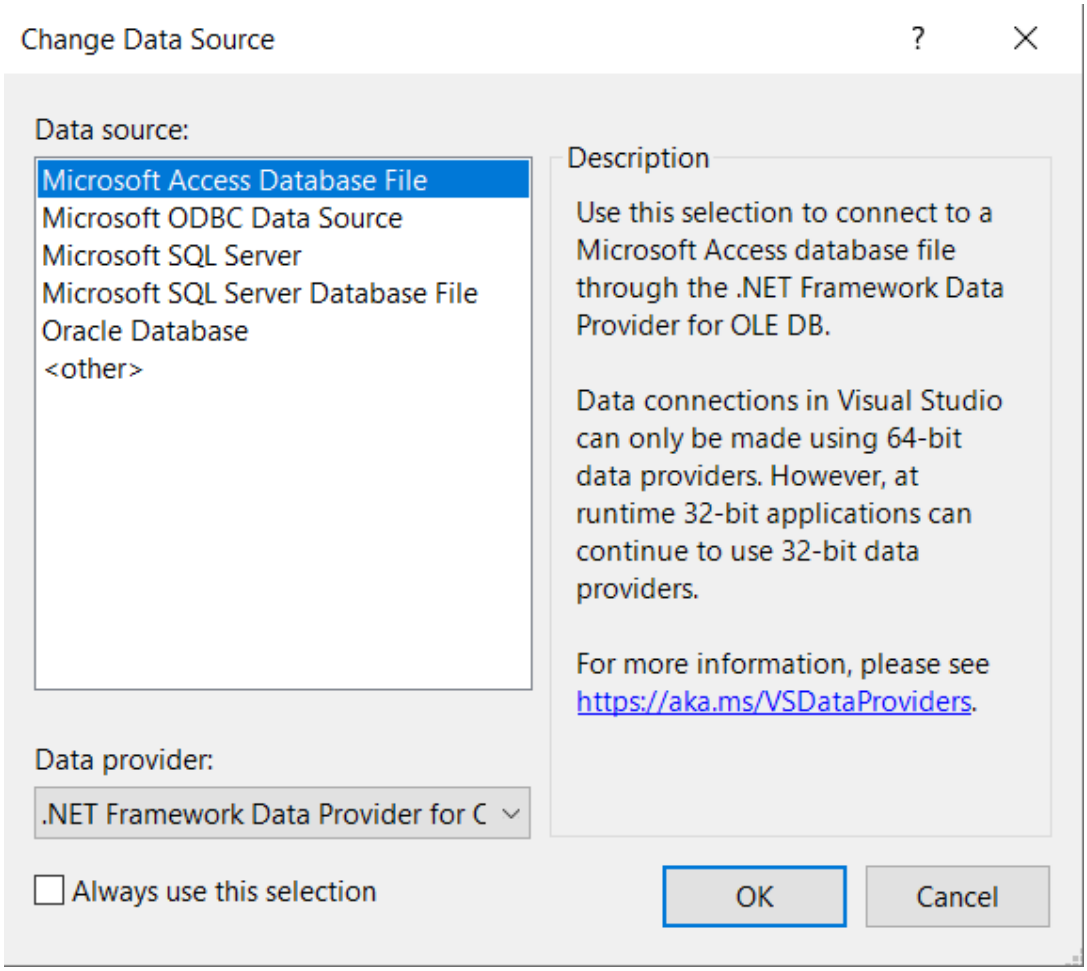


يفتح مربع الحوار إضافة اتصال Add Connection



٧. إذا لم يتم تعيين مصدر البيانات Data source إلى ملف قاعدة بيانات Microsoft Access Database File فحدد الزر تغيير Change

يفتح مربع الحوار تغيير مصدر البيانات Change Data Source في قائمة مصادر البيانات، اختر ملف قاعدة بيانات Microsoft Access Database File تم بالفعل تحديد الخيار موفر بيانات .NET Framework Data Provider for OLE DB.
اختر موافق OK



٨. اختر استعراض Browse بجوار اسم ملف قاعدة البيانات Database file name ثم انتقل إلى الملف `.accdb` واختر فتح Open

ملاحظة

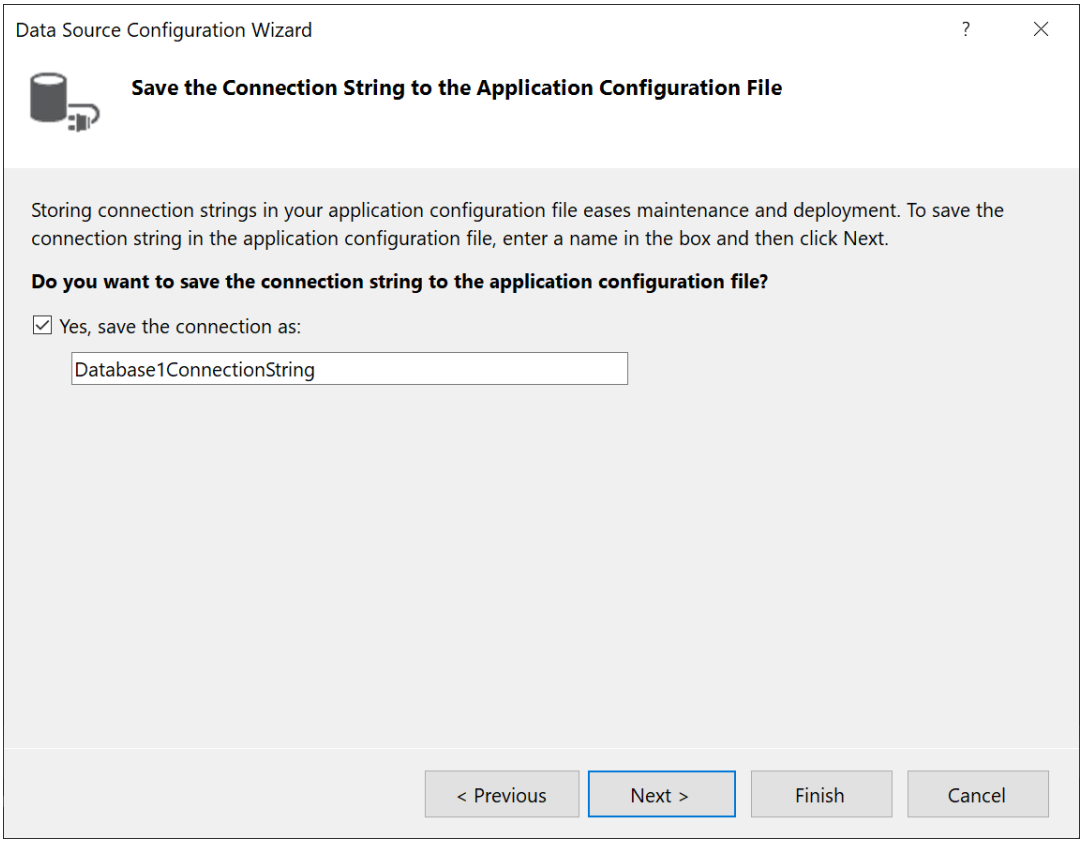
إذا لم تتطابق البت (32-bit or 64-bit) لـ Microsoft Office and Visual Studio فسترى خطأ أثناء الاتصال بقاعدة بيانات Access في Visual Studio 2019 ستحصل على خطأ مفاده أن موفر قاعدة البيانات غير مسجل. في Visual Studio 2022 سترى خطأ مفاده أنه لا يمكنك الاتصال بموفر بيانات 32-bit لحل هذا الخطأ، تأكد من أنه إذا كنت تستخدم إصدار 32-bit من Office فيجب استخدام Visual Studio 2019 أو إصدار سابق؛ للحصول على إصدار 64-bit من Office تحتاج إلى Visual Studio 2022 أو إصدار أحدث.

٩. أدخل اسم مستخدم وكلمة مرور (إذا لزم الأمر) ثم اختر موافق

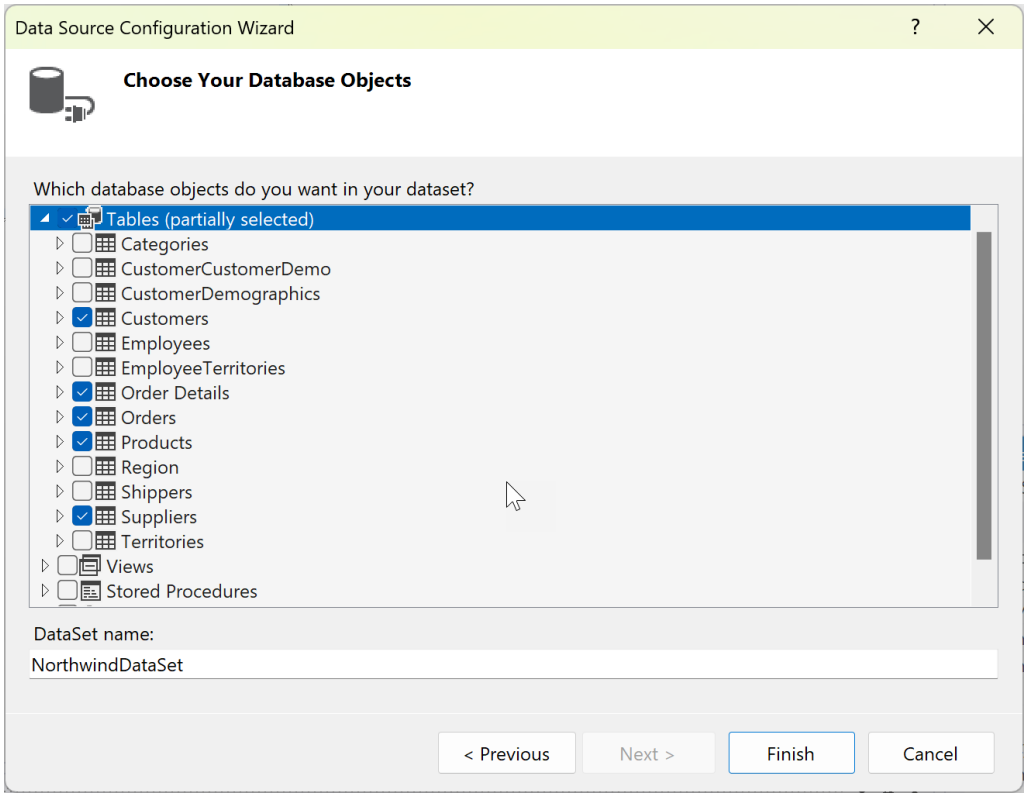
١٠. حدد التالي في صفحة اختيار اتصال البيانات الخاص بك Choose your Data Connection

قد تحصل على مربع حوار يخبرك أن ملف البيانات ليس في مشروعك الحالي. حدد نعم أو لا.

١١. حدد التالي في الصفحة حفظ سلسلة الاتصال في ملف تكوين التطبيق Save connection string to the Application Configuration file

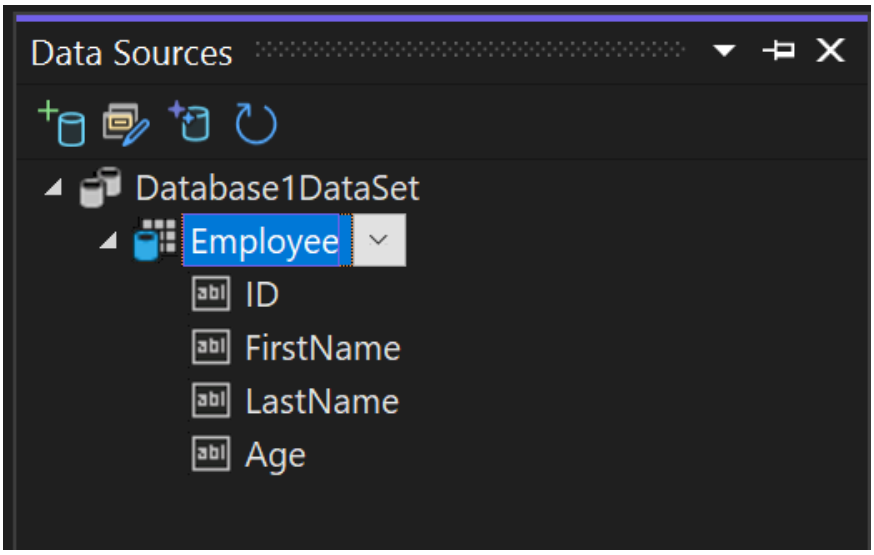


١٢. قم بتوسيع عقدة الجداول في الصفحة اختيار كائنات قاعدة البيانات Choose your Database Objects



١٣. حدد الجداول أو طرق العرض tables and views التي تريد تضمينها في مجموعة البيانات dataset ثم حدد إنهاء Finish

تتم إضافة مجموعة البيانات dataset إلى مشروعك، وتظهر الجداول وطرق العرض tables and views في النافذة مصادر البيانات Data Sources



١٤. على أجهزة 64-bit مع مشغل قاعدة بيانات 64-bit Access database تحتاج إلى التأكد من تشغيل التطبيق كتطبيق 64-bit افتح خصائص المشروع (اضغط على Alt+Enter أو انقر بزر الماوس الأيمن فوق عقدة المشروع project node

وحدد خصائص Properties) في علامة التبويب إنشاء Build قم بإلغاء تحديد خانة الاختيار Prefer 32-bit

إنشاء مجموعة بيانات لملف dataset for an .mdb

للاتصال بقواعد البيانات التي تم إنشاؤها باستخدام Access 2000-2003 استخدم الإجراء التالي:

١. افتح مشروع تطبيق Windows Forms or WPF في Visual Studio

٢. من القائمة عرض View حدد Other Windows > Data Sources

٣. في نافذة مصادر البيانات Data Sources انقر فوق إضافة مصدر بيانات جديد Add New Data Source

يفتح معالج تكوين مصدر البيانات Data Source Configuration Wizard

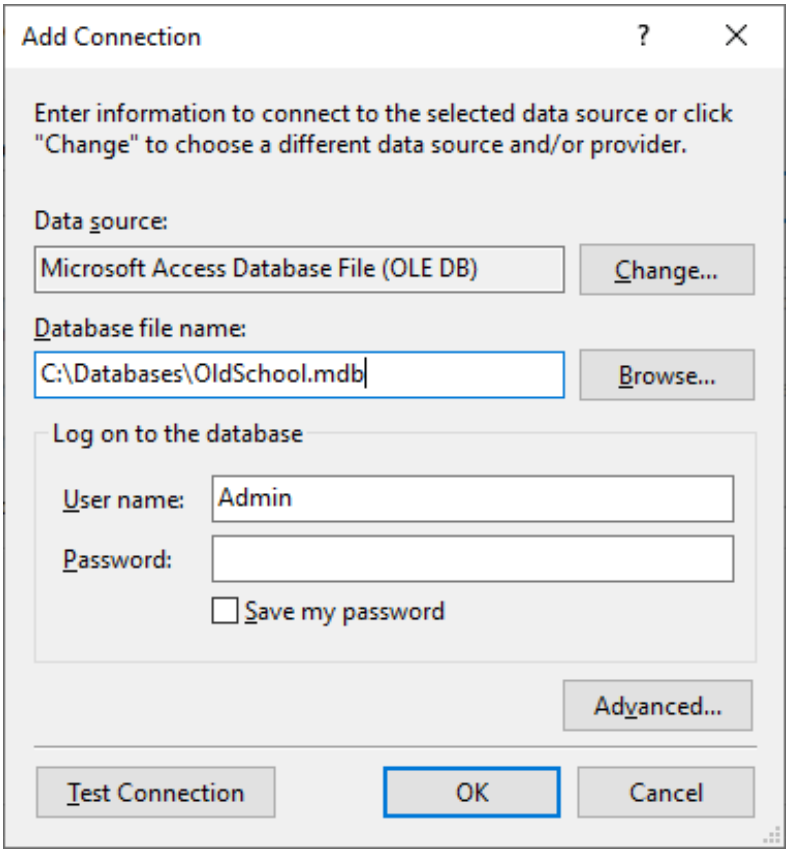
٤. حدد قاعدة البيانات Database في الصفحة اختيار نوع مصدر بيانات Choose a Data Source Type ثم حدد التالي.

٥. حدد مجموعة البيانات Dataset في الصفحة اختيار نموذج قاعدة بيانات Choose a Database Model ثم حدد التالي.

٦. في الصفحة اختيار اتصال البيانات Choose your Data Connection حدد اتصال جديد New Connection لتكوين اتصال بيانات جديد.

٧. إذا لم يكن مصدر البيانات هو ملف قاعدة بيانات Microsoft Access Database File (OLE DB) فحدد تغيير Change لفتح مربع الحوار تغيير مصدر البيانات Change Data Source وحدد Microsoft Access Database File ثم حدد موافق.

٨. في اسم ملف قاعدة البيانات Database file name حدد مسار الملف `.mdb` الذي تريد الاتصال به واسمه، ثم حدد موافق



٩. حدد التالي في صفحة اختيار اتصال البيانات الخاص بك Choose your Data Connection

١٠. حدد التالي في الصفحة حفظ سلسلة الاتصال في ملف تكوين التطبيق Save connection string to the Application Configuration file

١١. قم بتوسيع عقدة الجداول Tables node في الصفحة اختيار كائنات قاعدة البيانات Choose your Database Objects

١٢. حدد الجداول أو طرق العرض tables or views التي تريدها في مجموعة البيانات، ثم حدد إنهاء.

تتم إضافة مجموعة البيانات إلى مشروعك، وتظهر الجداول وطرق العرض في النافذة مصادر البيانات Data Sources

عرض التعليمات البرمجية التي تم إنشاؤها View the generated code

يتم تكوين أدوات البيانات لإنشاء الكثير من التعليمات البرمجية تلقائيًا عند إجراء عمليات معينة في مصمم النماذج. على سبيل المثال، عند سحب جدول وإفلاته في النموذج، تتم إضافة `DataGridView` ويتم إنشاء تعليمات برمجية لتوصيل البيانات بعنصر التحكم. يمكنك عرض هذا الرمز في الملف `*.Designer.cs` يضيف Visual Studio عددًا من العناصر الخاصين.

إذا قمت بتوسيع المنطقة المخفية، يمكنك أن ترى أن Visual Studio يضيف أيضًا كمية كبيرة من التعليمات البرمجية لإعداد عنصر التحكم مع ربط البيانات إلى الجدول الذي قمت بسحبه إلى النموذج DataGridView

أيضًا، في ملف التعليمات البرمجية الخلفية للنموذج الرئيسي، يضيف Visual Studio تعليمات برمجية تعالج إجراء الحفظ لحفظ التغييرات التفاعلية على البيانات، والتعليمة البرمجية التي تقوم بتحميل الجدول في محول الجدول loads the table into the table adapter

```
private void
ordersBindingNavigatorSaveItem_Click(object
sender, EventArgs e)
{
    this.Validate();
    this.ordersBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.database11
DataSet);

}

private void Form1_Load(object sender, EventArgs e)
{
   // TODO: This line of code loads data into the
'database11DataSet.Orders' table. You can move, or
remove it, as needed.

this.ordersTableAdapter.Fill(this.database11DataSe
t.Orders);
}
```

مبروك! باستخدام القليل من المساعدة من Visual Studio قمت بإنشاء تجربة تحرير مستندة إلى نموذج لجدول بيانات Access

الخطوات التالية

تتوفر مجموعة البيانات التي أنشأتها للتو في نافذة مصادر البيانات Data Sources يمكنك الآن تنفيذ أي من المهام التالية:

- حدد العناصر في النافذة مصادر البيانات Data Sources واسحبها إلى سطح النموذج أو التصميم (راجع ربط عناصر تحكم Windows Forms بالبيانات نظرة عامة على ربط بيانات Visual Studio أو WPF)
- افتح مصدر البيانات في مصمم مجموعة البيانات Dataset Designer لإضافة أو تحرير الكائنات objects التي تشكل مجموعة البيانات dataset
- أضف منطق التحقق من الصحة إلى الحدث RowChanging or ColumnChanging لجداول البيانات في مجموعة البيانات (راجع التحقق من صحة البيانات في مجموعات البيانات)

تمرين ربط تطبيق Student Grade بقواعد بيانات Access

في هذا القسم سننشئ تطبيق Student Grade الذي قمنا بإنشائه في الجزء الأول من الكتاب، باستخدام Visual Studio والاتصال بقاعدة بيانات Access لحفظ بيانات الطلاب.

إنشاء المشروع

تعلمنا سويًا خطوات إنشاء المشاريع في Visual Studio لذلك سوف ننتقل إلى الخطوة التالية. تأكد من اختيار StudentGrade لاسم التطبيق.

ملحوظة:

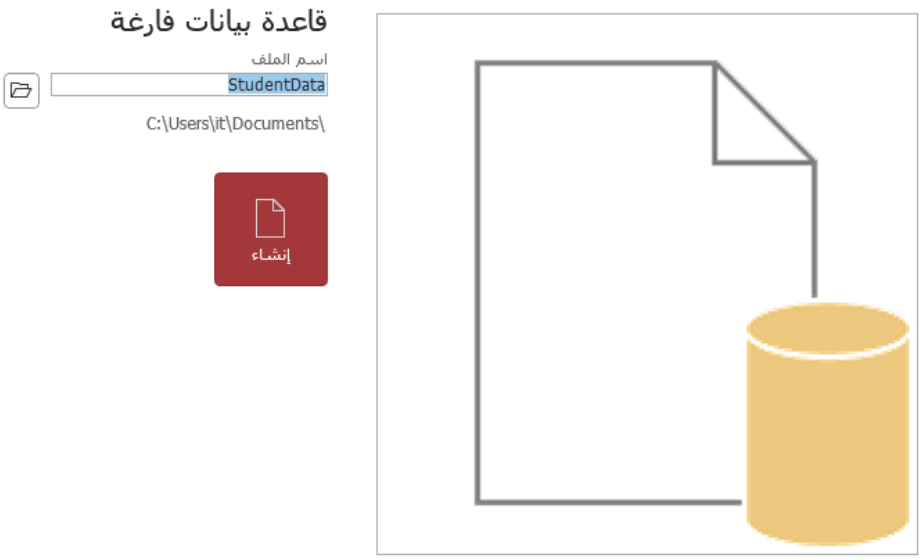
تأكد أنك قمت بإنشاء قاعدة بيانات، وتصميم الجداول بما يتناسب مع بيانات الطلاب.

تصميم قاعدة بيانات أكسس Access

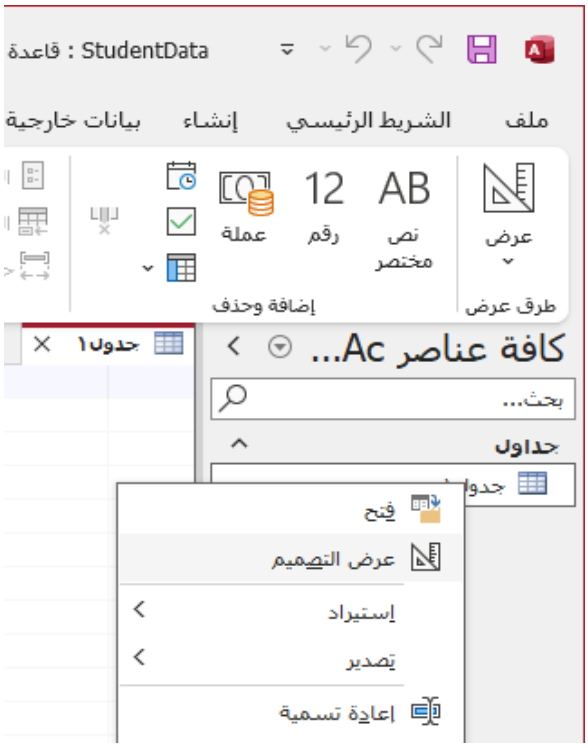
فتح برنامج Microsoft Access وأتبع الخطوات التالية:

١. من الصفحة الرئيسية اختر قاعدة بيانات فارغة.

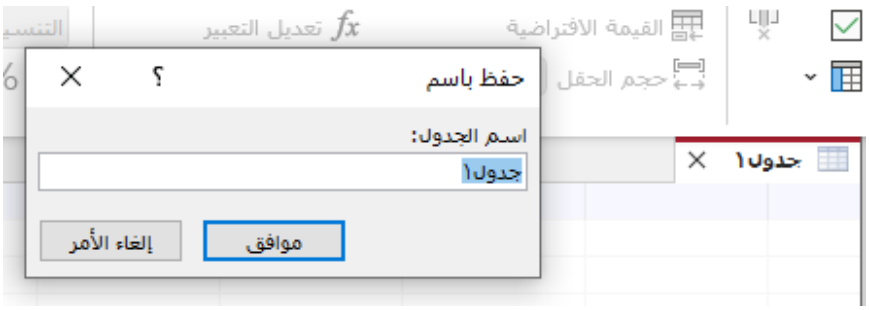
٢. في صفحة قاعدة بيانات فارغة اكتب اسم قاعدة البيانات StudentData ثم اضغط إنشاء.



٣. حدد جدول١ الموجود مسبقًا، واضغط مفتاح الماوس الأيمن ثم اختر عرض التصميم. يمكن إنشاء أكثر من جدول جديد.

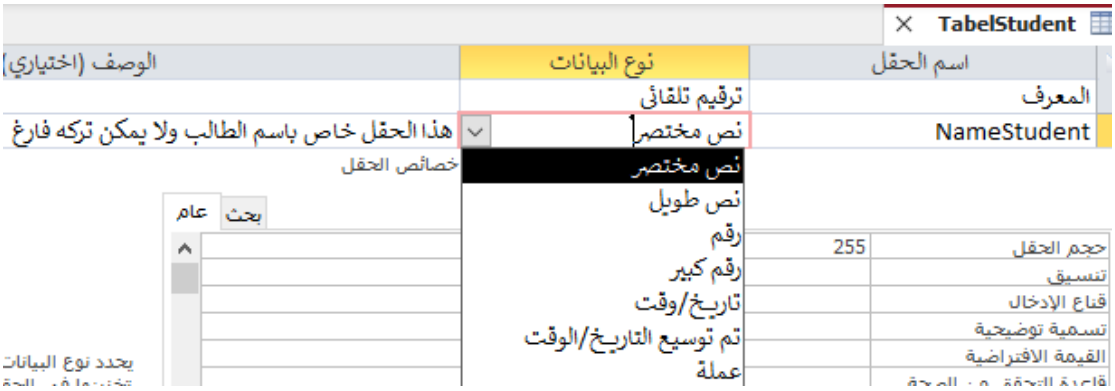


ستظهر هذه النافذة لاختيار اسم جديد للجدول.

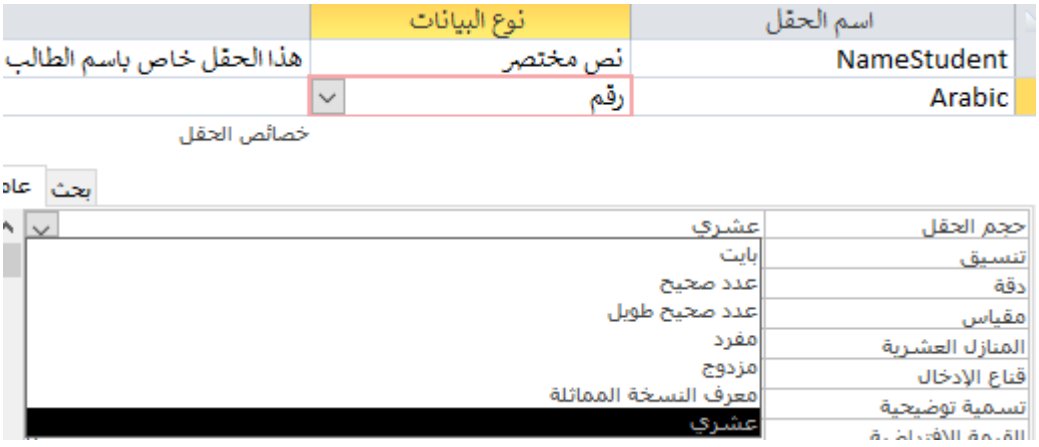


اكتب اسم الجدول TabelStudent واضغط موافق سوف تفتح هذه النافذة الجديدة التي تعرض تصميم الجدول.

٤. انتقل إلى صف جديد، في مربع اسم الحقل أدخل اسم "NameStudent" وفي مربع نوع البيانات اختر من القائمة المنسدلة "نص مختصر"

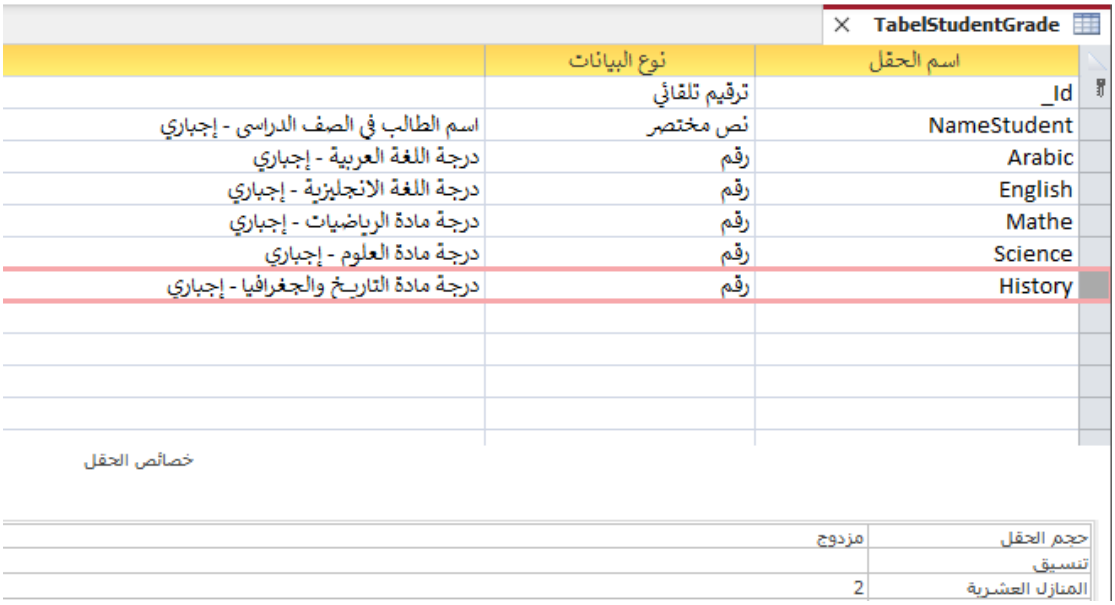


٥. انتقل إلى صف جديد، في مربع اسم الحقل أدخل اسم المادة "Arabic" وفي مربع نوع البيانات اختر من القائمة المنسدلة "رقم" ثم حدد حجم الحقل من القائمة في الأسفل واختر من القائمة المنسدلة عشري او مزدوج.



٦. اتبع هذه الخطوات مع باقي المواد الدراسية للطلاب.

في النهاية يجب ان يكون الجدول كالتالي:



هام:

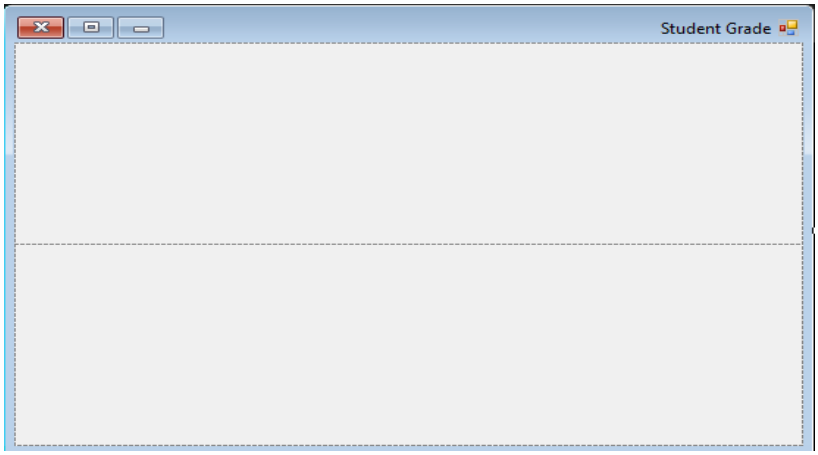
تأكد أنك تستخدم Microsoft Access 64bit وليس إصدار 32bit لان 32bit لا يعمل الا على إصدارات أقدم من Visual Studio 2022

تصميم الواجهة

١. افتح برنامج Visual Studio 2022 واختر مشروع StudentGrade

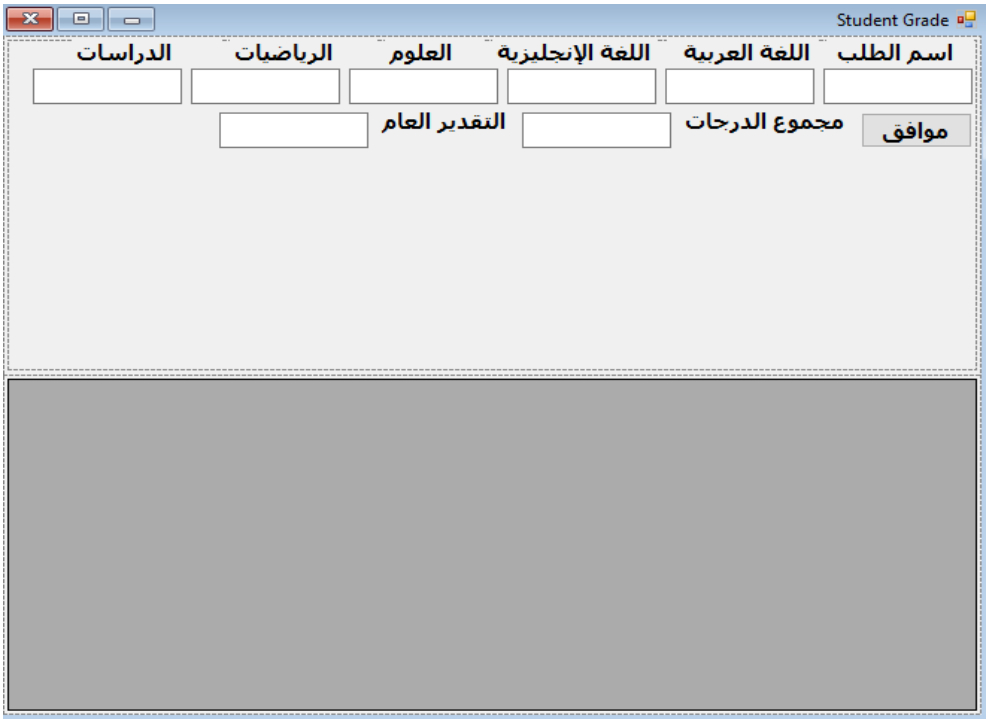
٢. في النموذج Form أضف العناصر التالية:

- عنصر TableLayoutPanel لتنسيق حجم عناصر التحكم تلقائيًا مع حجم النموذج Form
  - ضبط خاصية Dock على Fill
  - حذف الأعمدة Column وإبقاء صفين Rows فقط



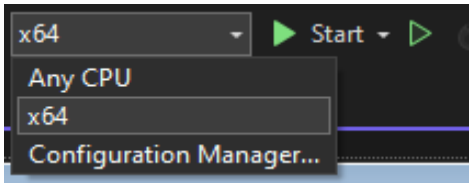
- عنصر FlowLayoutPanel في الصف العلوي من TableLayoutPanel لترتيب عناصر التحكم داخلها.
- عنصر DataGridView في الصف السفلي من TableLayoutPanel لعرض بيانات الطلاب من قاعدة البيانات.
- ثماني مربعات TextBox لعرض بيانات الطلاب.
  - اضبط خاصية الاسم Name لكل عنصر باسم المادة التي سيعرض درجاتها.
- ثماني ملصقات Label للتوضيح.
- زر Button واضبط خاصية Name إلى buttonGrade والنص إلى "موافق"

يجب ان يبدو التطبيق مشابهًا للتالي:



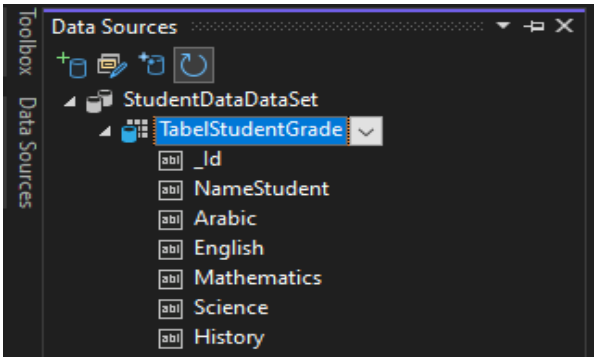
هام

تأكد من ضبط معمارية التطبيق على 64bit قبل تشغيل التطبيق كي لا يحدث استثناء.

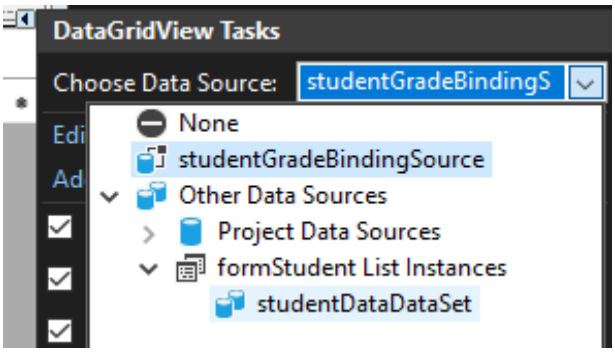


ربط قاعدة البيانات بالمشروع

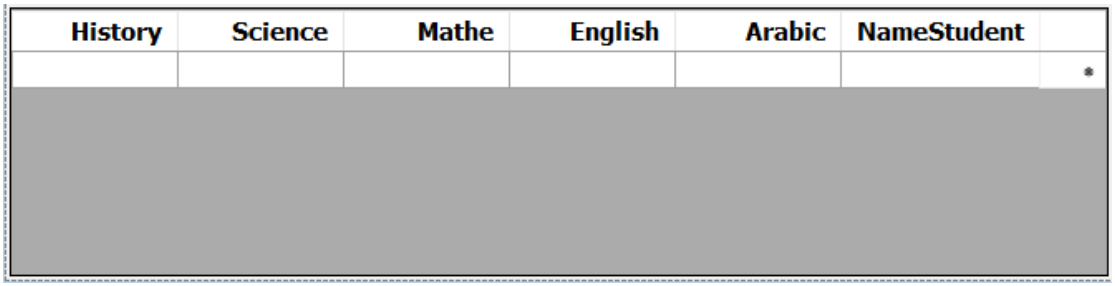
١. اربط قاعدة البيانات بالمشروع كما تعلمت في الدروس السابقة.



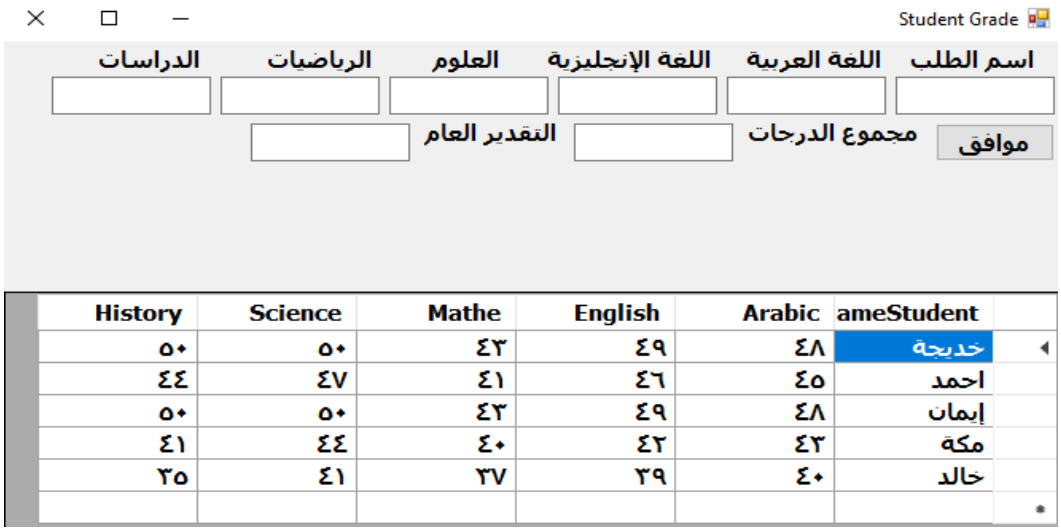
٢. حدد dataGridView1 ثم اضغط السهم الصغير في اقصى اليمن، حدد القائمة المنسدلة Choose Data Source واختر اسم الجدول الذي إنشاته StudentGrade



سيتم إضافة جميع حقول الجدول في dataGridView1 وتعرض بهذا الشكل.

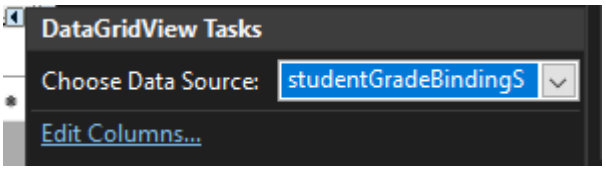


٣. شغل التطبيق سوف تعرض البيانات المخزنة في الجدول.

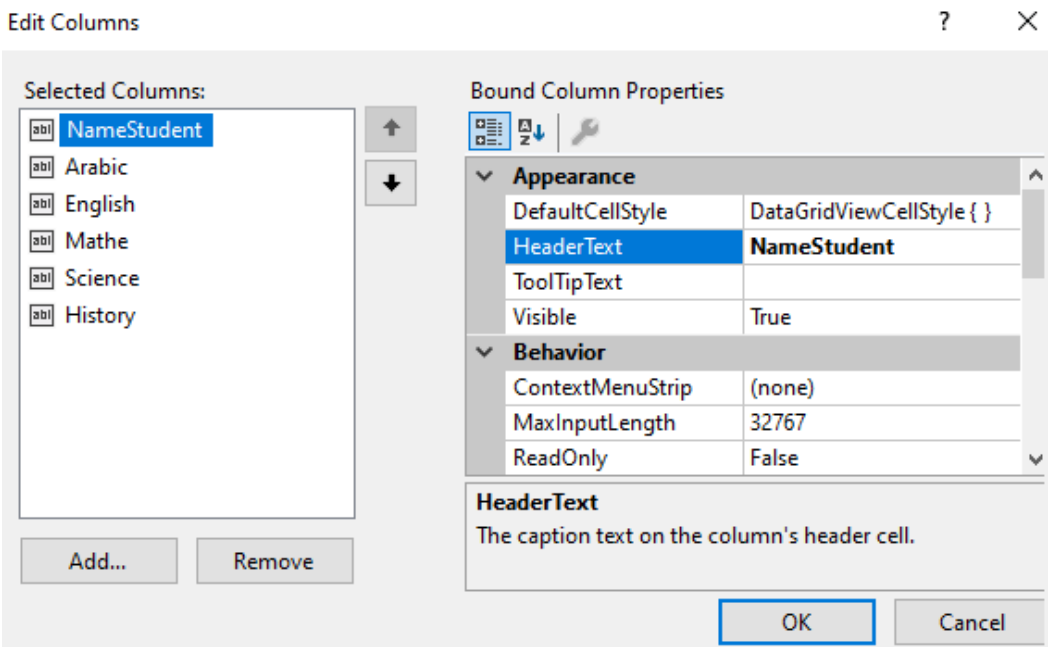


لاحظ ان صندوق النص TextBox لا فارغ، لا يعرض أي بيانات.

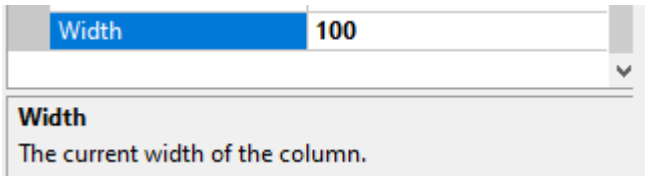
٤. تغيير أسماء الاعمدة في dataGridView1 للعرض للمستخدم فقط. قم بتحديد السهم الصغير واختيار من القائمة المنسدلة تعديل الاعمدة Edit Columns



في نافذة Edit Columns عدل خاصية HeaderText من NameStudent إلى "اسم الطالب"



عدل خاصية العرض Width إلى الحجم المناسب. اضبطها على 130

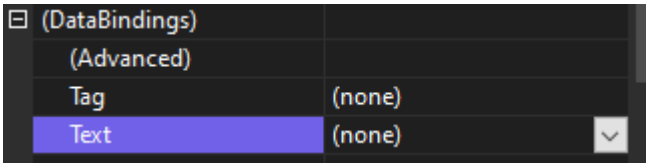


5. عدل جميع أسماء الاعمدة المتبقية في عنصر dataGridView1

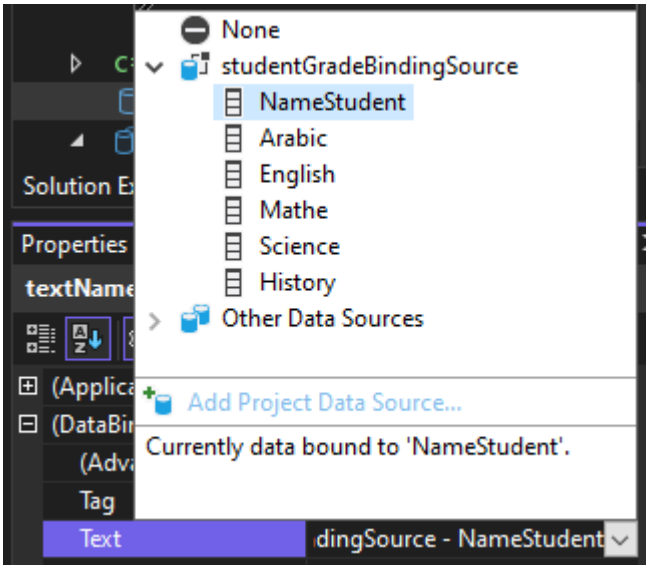
اسم الطالب	العربية	الانجليزية	الرياضيات	العلوم	التاريخ

٦. اربط عناصر TextBox بجدول البيانات.

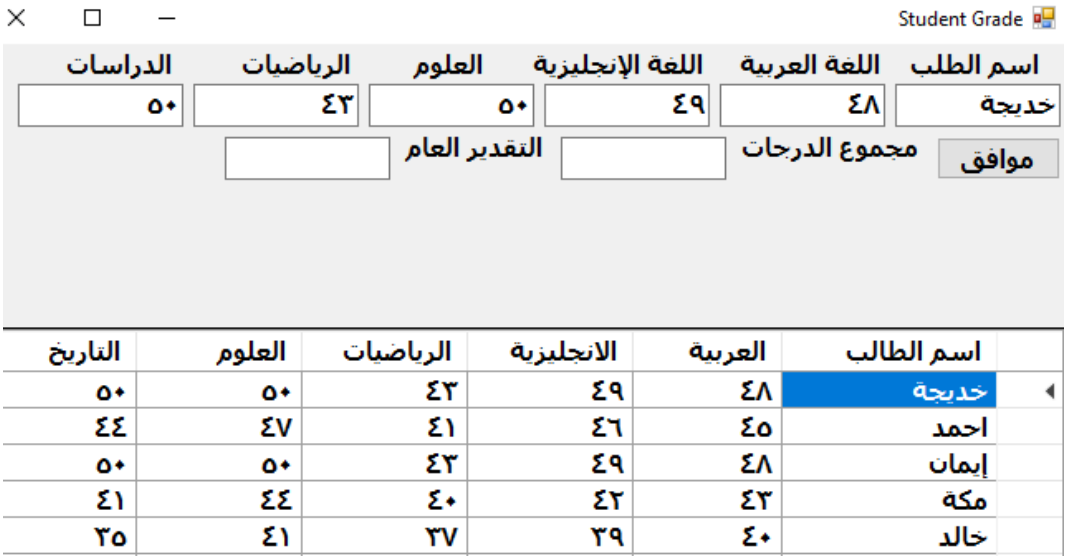
حدد عنصر textNameStudent من نافذة الخصائص، اختر خاصية DataBindings اضغط على علامة + لتوسيعها، ثم اختر خاصية Text



من القائمة المنسدلة اختر جدول البيانات StudentGrade ثم اسم الحقل NameStudent



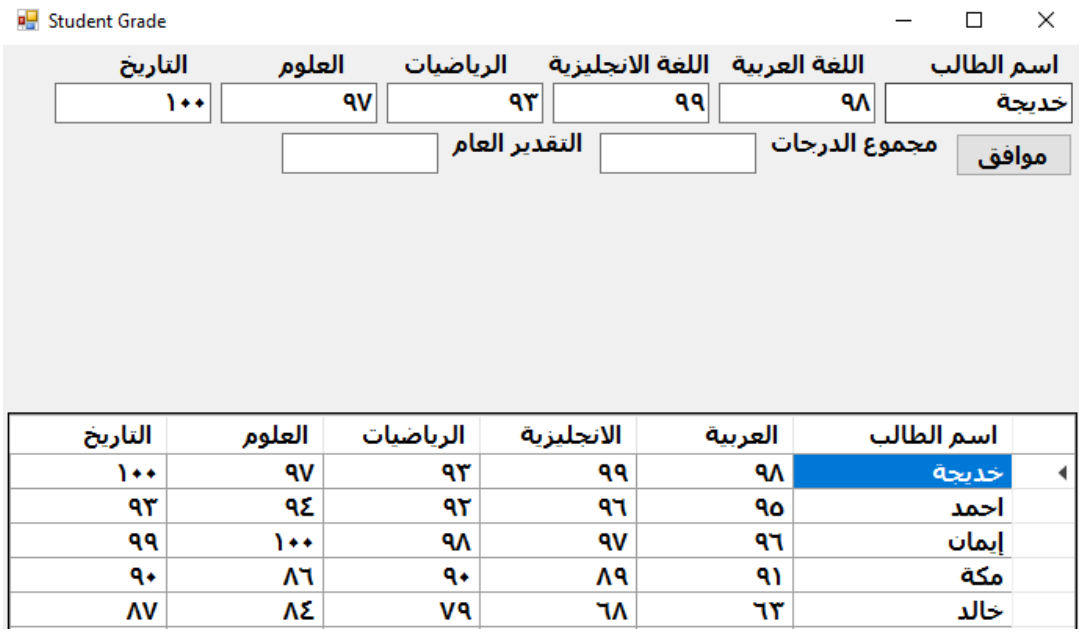
افعل ذلك مع جميع عناصر TextBox المتبقية.



٧. لحساب مجموع الدرجات والتقدير العام أضف معالج حدث للزر buttonGrade بالضغط مرتين بمفتاح الماوس الايسر فوق الزر، سينشئ Visual Studio معالج حدث باسم buttonGrade_Click() اضف داخله التعليمات التالية:

```
private void buttonGrade_Click(object sender, EventArgs e)
{
 // إنشاء متغيرات لالتقاط الدرجات
 float arabic = float.Parse(textArabic.Text);
 float eglish = float.Parse(textEnglish.Text);
 float science = float.Parse(textScience.Text);
 float mathe = float.Parse(textMathe.Text);
 float history = float.Parse(textHistory.Text);
 //  حساب مجموع الدرجات وعرضها
 float all = arabic + eglish + science + mathe +
history;
 textAll.Text = all.ToString();
 // حساب التقدير العام وعرضه
 float grade = all / 5;
 textGrade.Text = grade.ToString();
}
```

٨. شغل التطبيق. تنقل بين صفوف DataGridView واحسب المجموع والتقدير.



يمكنك إضافة مميزات للتطبيق.

٨. البدء في استخدام ASP.NET Core لإنشاء أول تطبيق ويب

في هذه الوحدة لتطوير C# باستخدام ASP.NET Core يمكنك إنشاء تطبيق ASP.NET Core web في Visual Studio

توضح لك هذه الوحدة التعليمية كيفية:

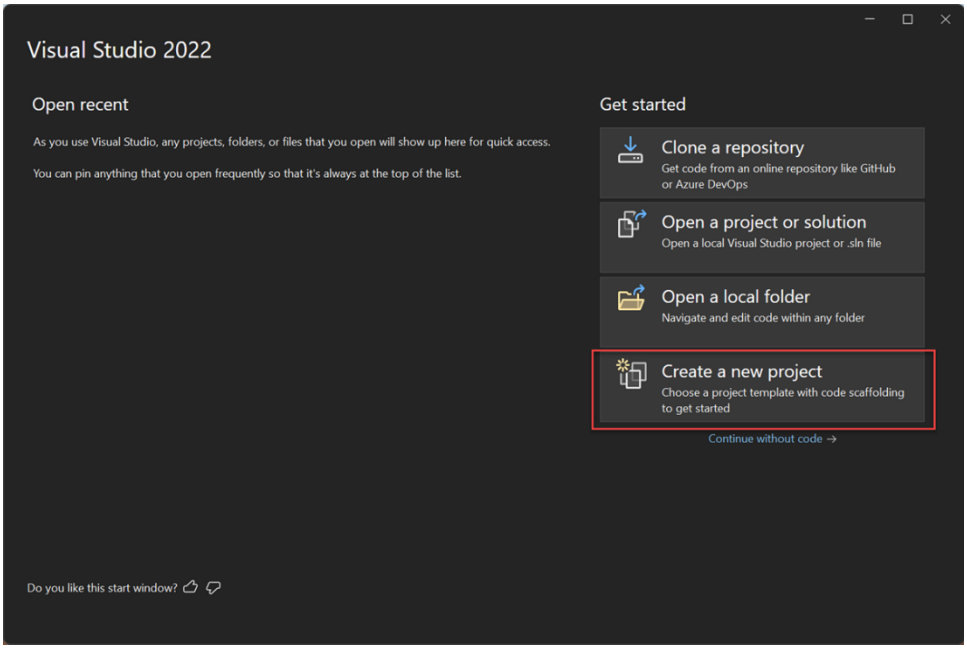
- إنشاء مشروع Visual Studio
- إنشاء تطبيق ويب ASP.NET Core web
- إجراء تغييرات على تطبيق الويب web
- استكشاف ميزات IDE
- تشغيل تطبيق الويب web

تحتاج لتثبيت ASP.NET and web development للتحقق من عبء العمل هذا أو تثبيته في Visual Studio حدد **أدوات** ثم **الحصول على الأدوات والميزات** Tools > Get Tools and Features لمزيد من المعلومات، راجع تغيير أحمال العمل أو المكونات الفردية.

إنشاء مشروع

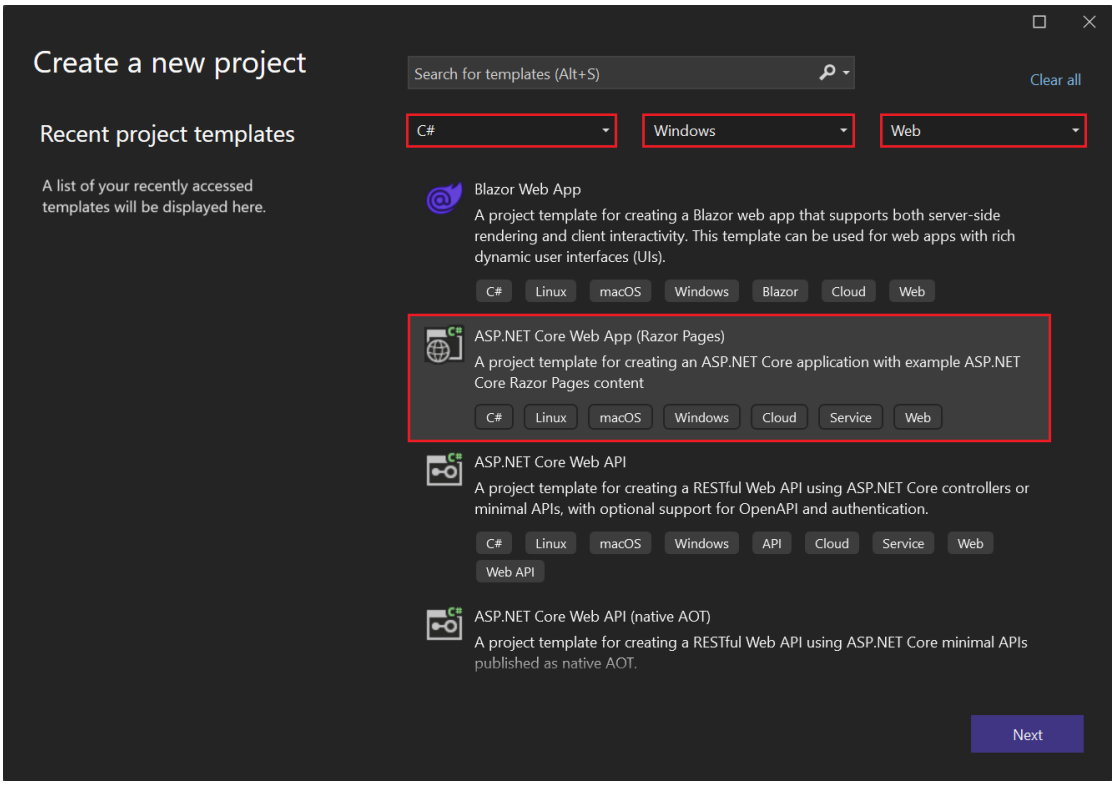
أولاً، قم بإنشاء مشروع ASP.NET Core يأتي نوع المشروع مع جميع ملفات القالب التي تحتاجها لبناء موقع ويب يعمل بكامل طاقته.

١. في نافذة البدء، حدد إنشاء مشروع جديد Create a new project

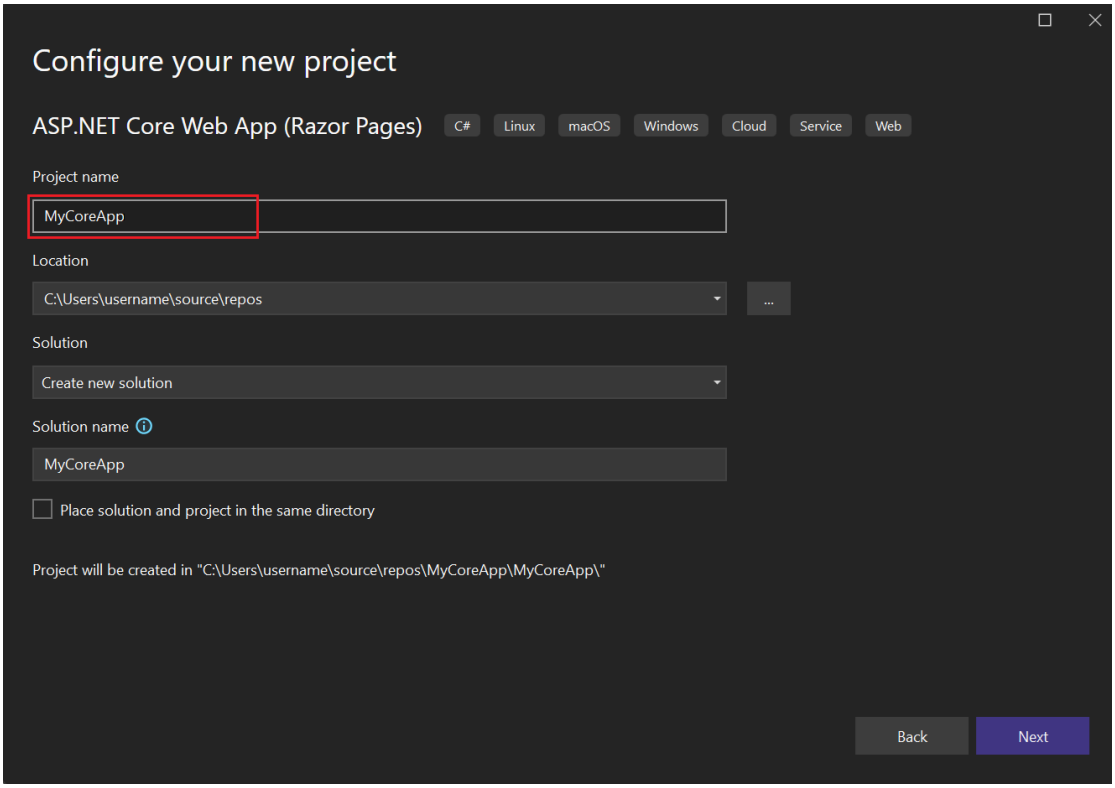


٢. من نافذة إنشاء مشروع جديد Create a new project حدد ة من قائمة اللغة، ثم حدد Windows من قائمة جميع الأنظمة الأساسية All platforms ومن قائمة جميع أنواع المشاريع All project types حدد **Web**

بعد تطبيق مرشحات اللغة والمنصة ونوع المشروع، حدد قالب **ASP.NET Core Web App (Razor Pages)** وحدد التالي Next



٣. في نافذة تكوين مشروعك الجديد Configure your new project في حقل اسم المشروع اكتب MyCoreApp وحدد التالي Next

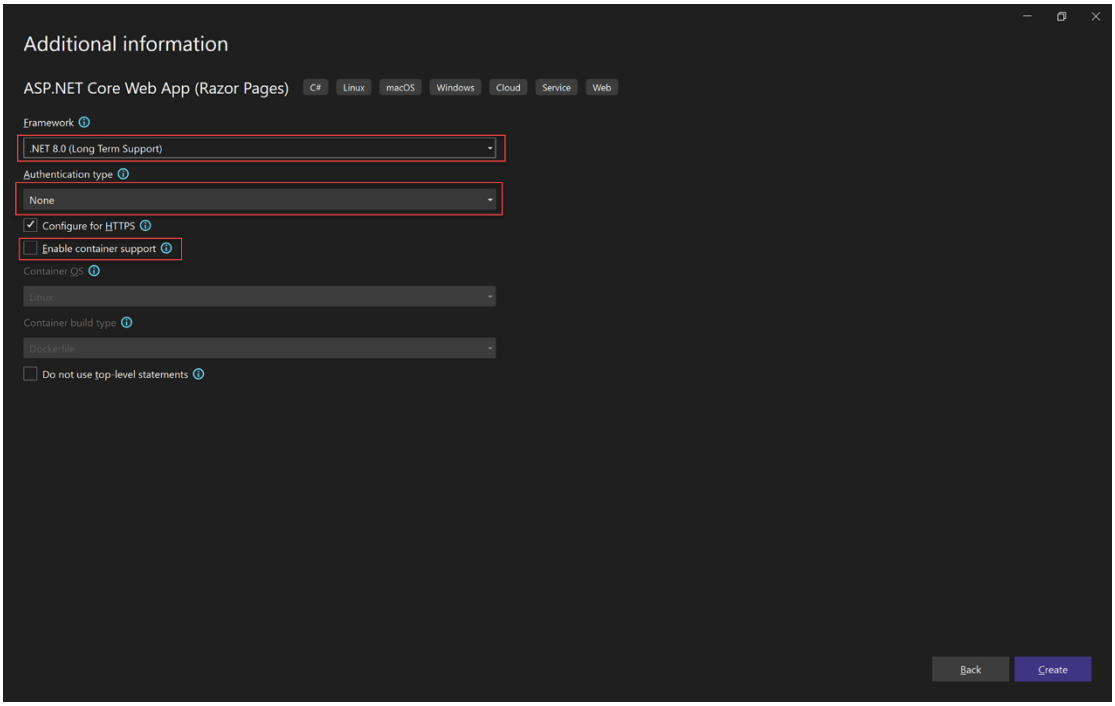


٤. في نافذة المعلومات الإضافية Additional information تأكد من ظهور **.NET 8.0** في حقل الإطار المستهدف Target Framework

من هذه النافذة، يمكنك تمكين دعم الحاوية وإضافة دعم المصادقة enable container support and add authentication support تحتوي القائمة المنسدلة لنوع المصادقة Authentication Type على الخيارات الأربعة التالية:

- لا شيء **None** لا يوجد مصادقة.
- الحسابات الفردية **Individual accounts** يتم تخزين هذه المصادقات في قاعدة بيانات محلية أو تعتمد على Azure
- منصة هوية **Microsoft identity platform** يستخدم هذا الخيار Microsoft Entra ID or Microsoft 365 للمصادقة
- **Windows** مناسب لتطبيقات الإنترنت

اترك مربع **تمكين دعم الحاوية Enable container support** بدون تحديد، ثم حدد **None** لنوع المصادقة.



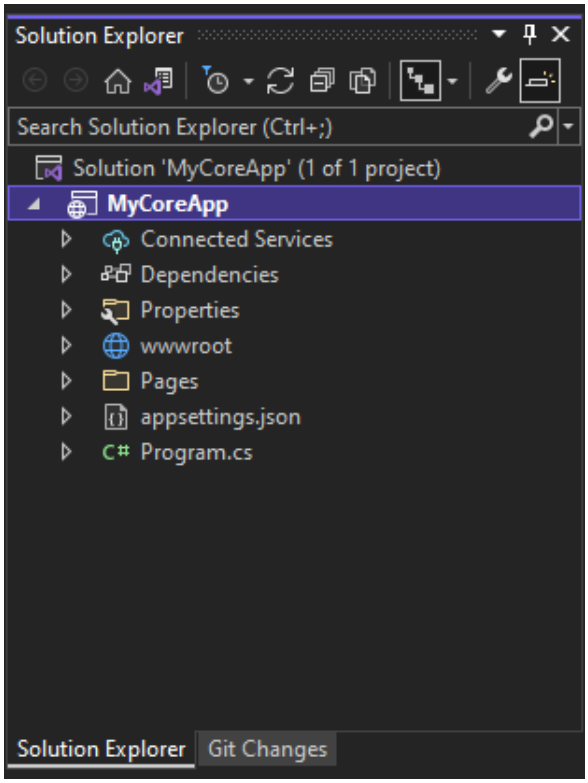
٥. اضغط إنشاء Create

حول الحل About solution

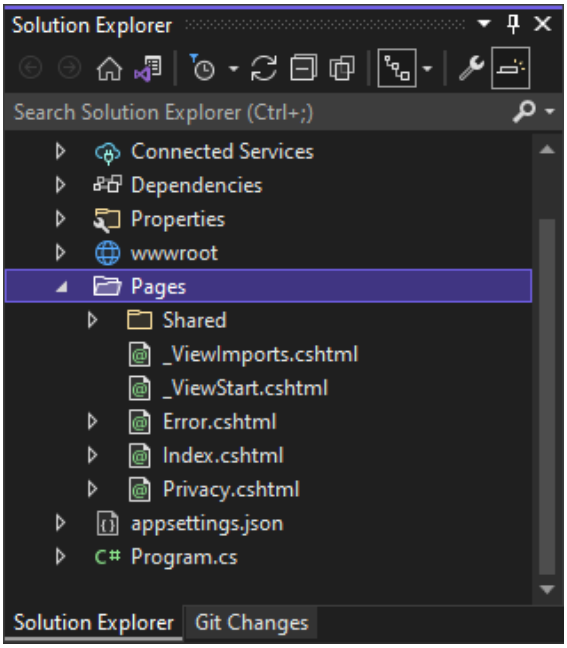
يتبع هذا الحل نمط تصميم **صفحة Razor** وهو مختلف عن نمط تصميم Model-View-Controller (MVC) من حيث أنه مبسط ليشمل تعليمات النموذج ووحدة التحكم داخل صفحة Razor نفسها.

جولة حول الحل solution

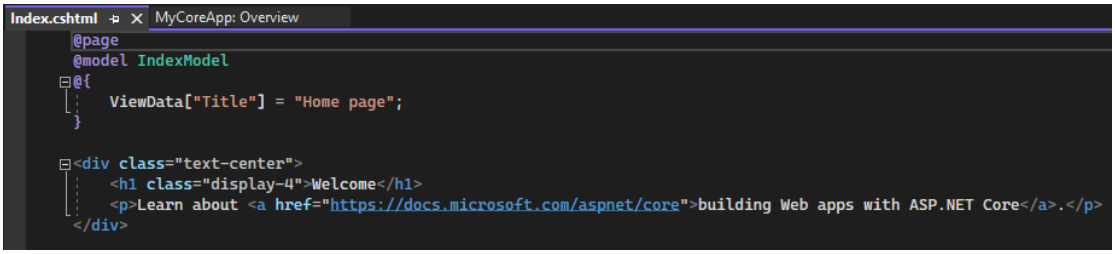
يقوم قالب المشروع بإنشاء solution واحد باستخدام مشروع ASP.NET Core يسمى MyCoreApp حدد علامة التبويب مستكشف الحلول Solution Explorer لعرض محتوياته.



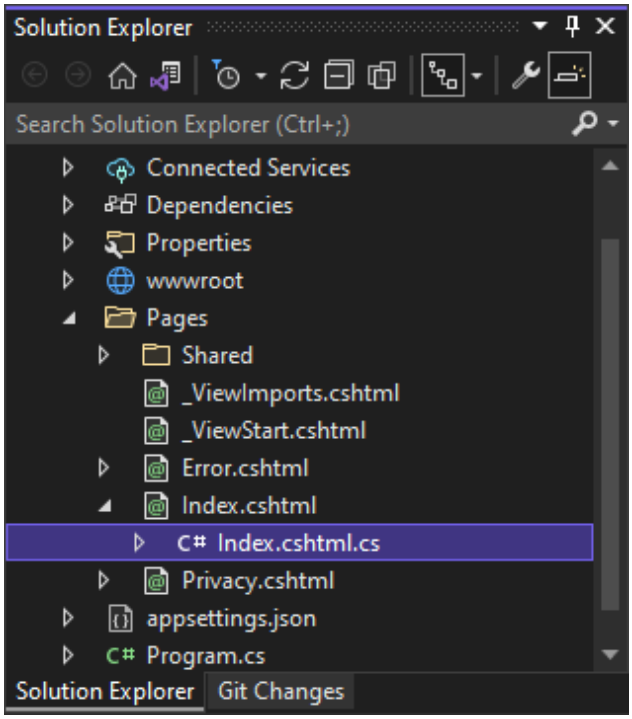
٢. توسيع مجلد الصفحات Pages folder



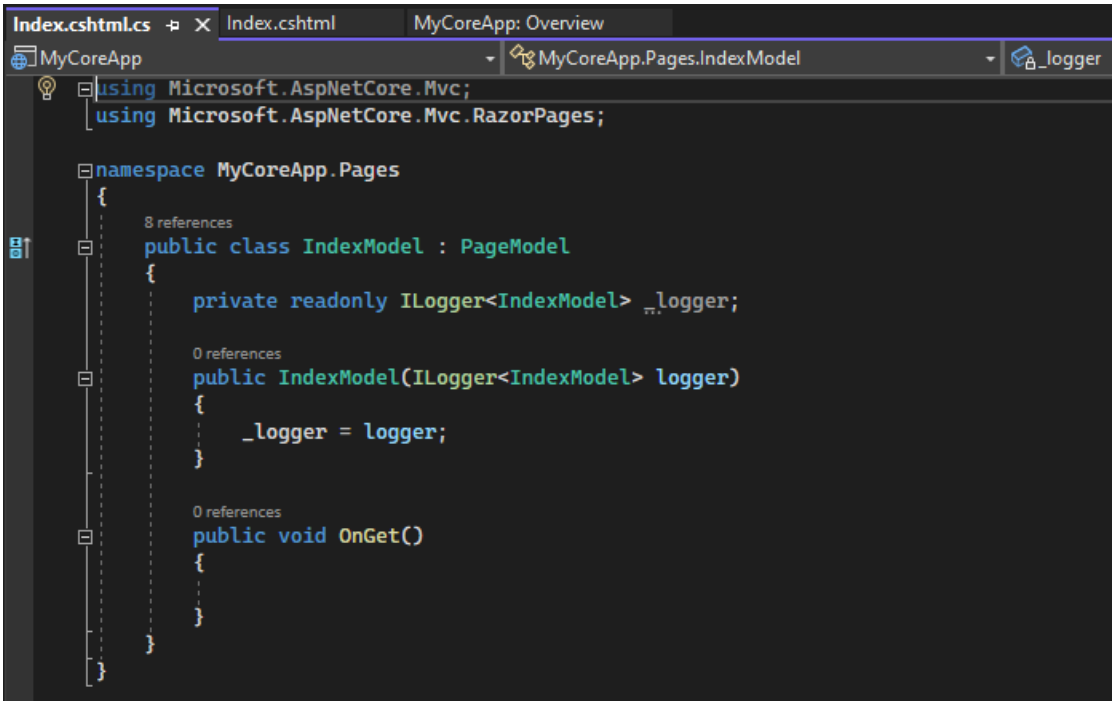
٣. حدد الملف **Index.cshtml** ثم اعرضه في محرر التعليمات.



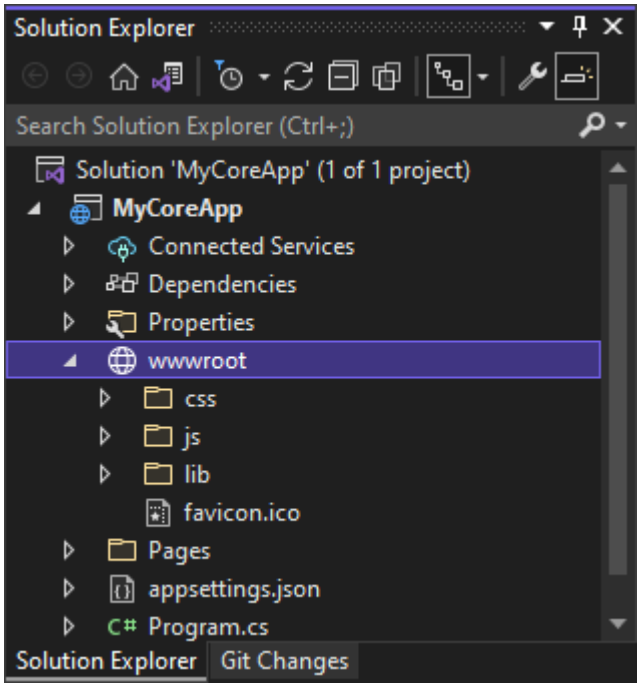
٤. يحتوي كل ملف cshtml. على ملف تعليمات مرتبط به. لفتح ملف التعليمات في المحرر، قم بتوسيع عقدة **Index.cshtml** في مستكشف الحلول، ثم حدد الملف **Index.cshtml.cs**



٥. عرض ملف Index.cshtml.cs في محرر التعليمات.

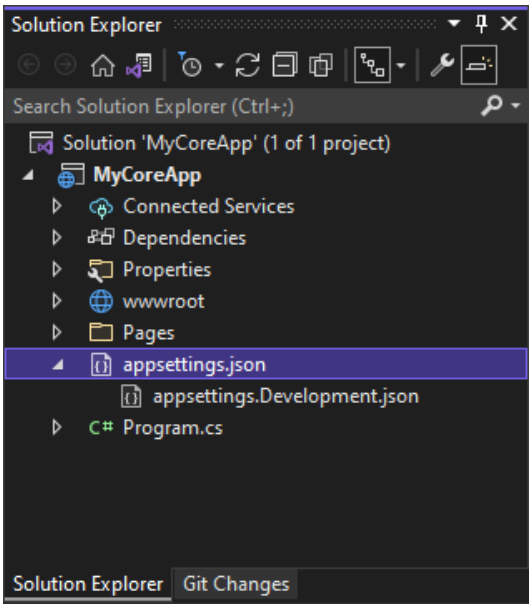


٦. يحتوي المشروع على مجلد **wwwroot** وهو الجذر لموقع الويب الخاص بك. قم بتوسيع المجلد لعرض محتوياته.



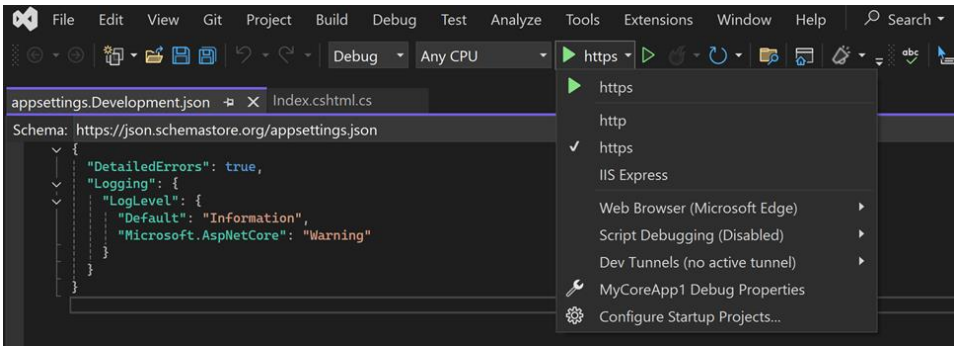
يمكنك وضع محتوى الموقع الثابت مثل CSS, images, and JavaScript libraries مباشرة في المسارات التي تريدها.

٧. يحتوي المشروع أيضًا على ملفات تكوين تدير تطبيق الويب في وقت التشغيل. يتم تخزين تكوين التطبيق الافتراضي في appsettings.json ومع ذلك، يمكنك تجاوز هذه الإعدادات باستخدام appsettings.Development.json قم بتوسيع ملف appsettings.json لعرض ملف appsettings.Development.json



تشغيل وتصحيح الأخطاء وإجراء التغييرات Run, debug, and make changes

١. في شريط الأدوات، حدد زر https لبناء التطبيق وتشغيله في وضع التصحيح. أو اضغط على F5 أو انتقل إلى تصحيح، بدء التصحيح Debug > Start Debugging من شريط القائمة.

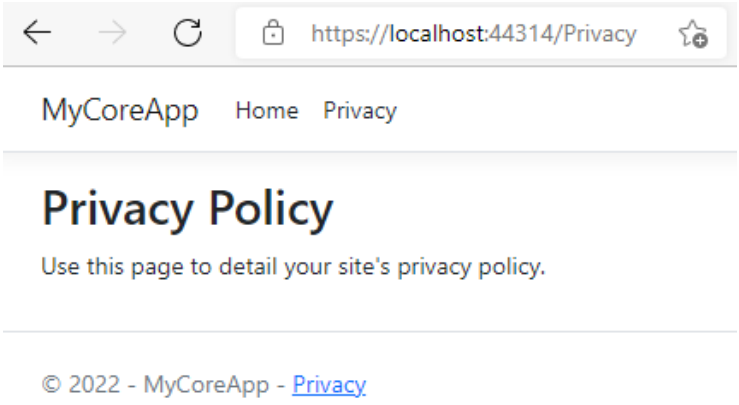


ملحوظة:

قد تتلقى أيضًا رسالة تسألك عما إذا كنت تريد قبول شهادة ASP.NET Core SSL لعرض التعليمات البرمجية في متصفح الويب، حدد **Yes** إذا تلقيت رسالة تحذير أمان متابعة حدد **Yes** تعرف على المزيد حول enforcing SSL in ASP.NET Core

٢. يقوم Visual Studio بتشغيل نافذة متصفح. يجب أن ترى بعد ذلك الصفحات الرئيسية والخصوصية Home and Privacy في شريط القائمة.

٣. حدد الخصوصية Privacy من شريط القائمة. تعرض صفحة الخصوصية في المتصفح النص الموجود في ملف Privacy.cshtml



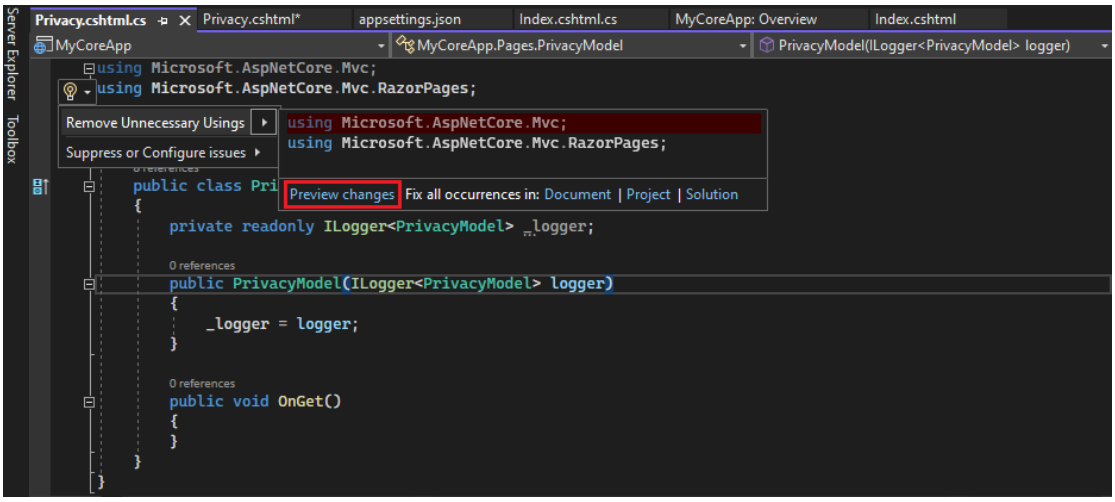
٤. ارجع إلى Visual Studio ثم اضغط على **Shift+F5** لإيقاف التصحيح. يؤدي هذا الإجراء إلى إغلاق المشروع في نافذة المتصفح.

٥. في Visual Studio افتح Privacy.cshtml للتحرير. بعد ذلك، احذف الجملة، استخدم هذه الصفحة لتفصيل سياسة الخصوصية الخاصة بموقعك، واستبدلها بـ هذه الصفحة قيد الإنشاء اعتبارًا من @ViewData ["TimeStamp"]

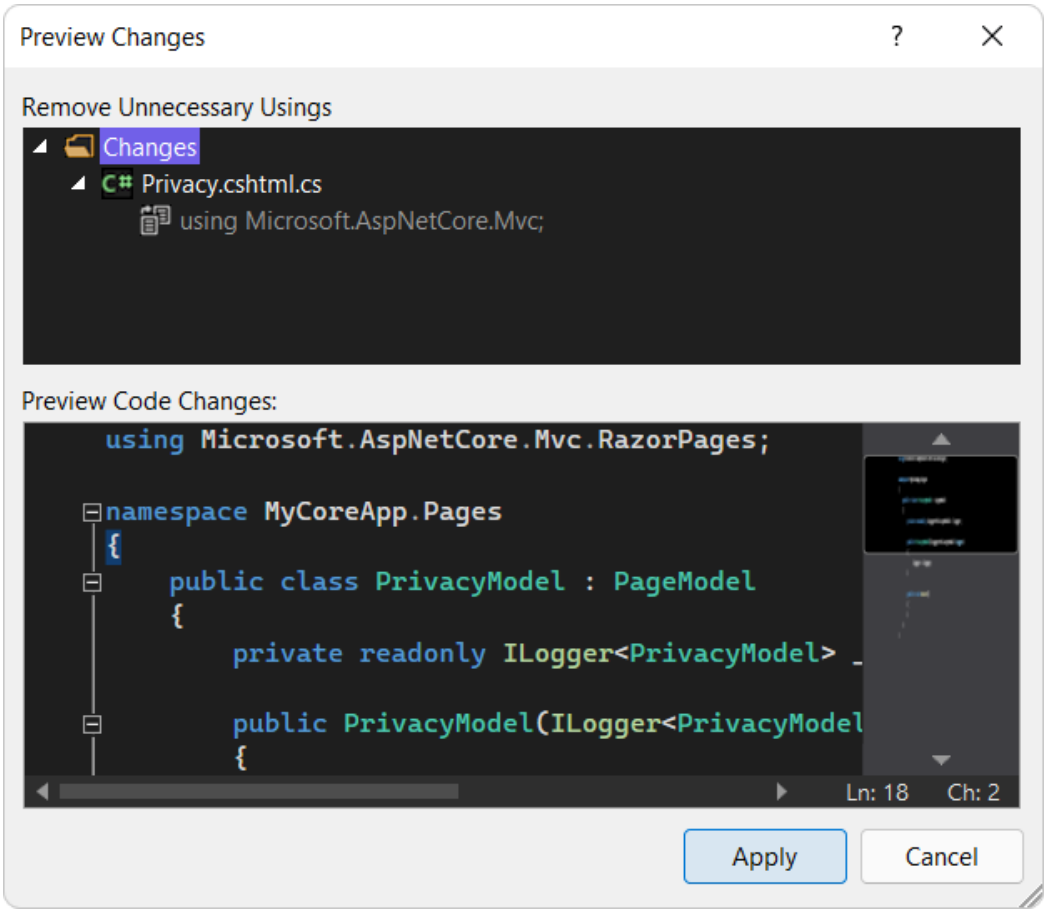
```
@page
@model PrivacyModel
@{
    ViewData["Title"] = "Privacy Policy";
}
<h1>@ViewData["Title"]</h1>

<p>This page is under construction as of @ViewData["Timestamp"]</p>
```

٦. الآن، لنقم بإجراء تغيير على التعليمات حدد Privacy.cshtml.cs ثم قم بتنظيف `using` directives التوجيهات الموجودة في أعلى الملف من خلال تحديد الاختصار التالي:

مرر الماوس فوق أو حدد توجيهًا `using` directives رمادي اللون، يظهر مصباح ضوئي للإجراءات السريعة أسفل علامة الإدراج أو في الهامش الأيسر. حدد المصباح الضوئي، ثم حدد سهم التوسيع بجوار إزالة الاستخدامات غير الضرورية Remove unnecessary usings



الآن حدد معاينة التغييرات Preview changes لرؤية التغييرات



حدد تطبيق Apply يقوم Visual Studio بحذف التوجيهات `using` directives غير الضرورية من الملف.

٧. بعد ذلك، قم بإنشاء سلسلة للتاريخ الحالي بتنسيق مناسب لثقافتك أو منطقتك باستخدام أسلوب DateTime.ToString

- تحدد المعلمة argument الأولى للأسلوب كيفية عرض التاريخ. يستخدم هذا المثال محدد التنسيق (d) الذي يشير إلى تنسيق التاريخ القصير.
- المعلمة argument الثانية هي كائن object CultureInfo الذي يحدد الثقافة أو المنطقة للتاريخ. تحدد المعلمة الثانية من بين أمور أخرى، لغة أي كلمات في التاريخ، ونوع الفواصل المستخدمة.

قم بتغيير نص الأسلوب `OnGet()` في **Privacy.cshtml.cs** إلى التعليمات التالية:

```
public void OnGet()
{
```

```
  string dateTime = DateTime.Now.ToString("d",
new CultureInfo("en-US"));

  ViewData["TimeStamp"] = dateTime;

}
```

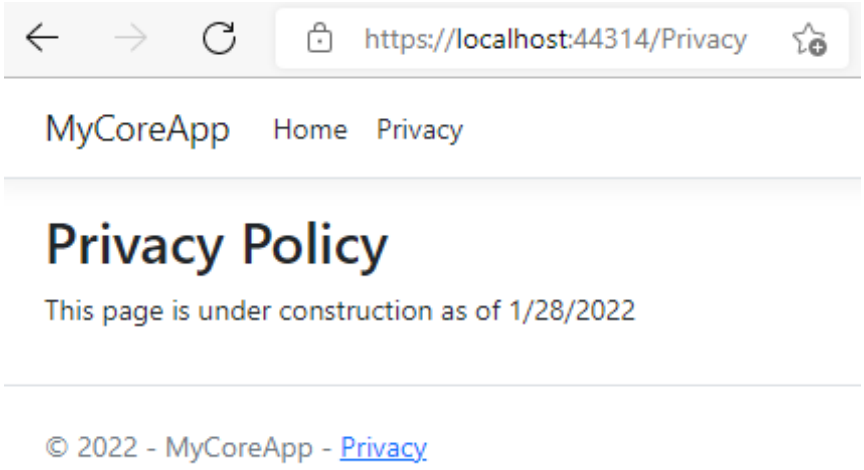
٨. لاحظ أن التوجيه using directive التالي يتم إضافته تلقائيًا إلى أعلى الملف:

```
using System.Globalization;
```

يحتوي `System.Globalization` على فئة class CultureInfo

٩. اضغط على F5 لفتح مشروعك في متصفح الويب

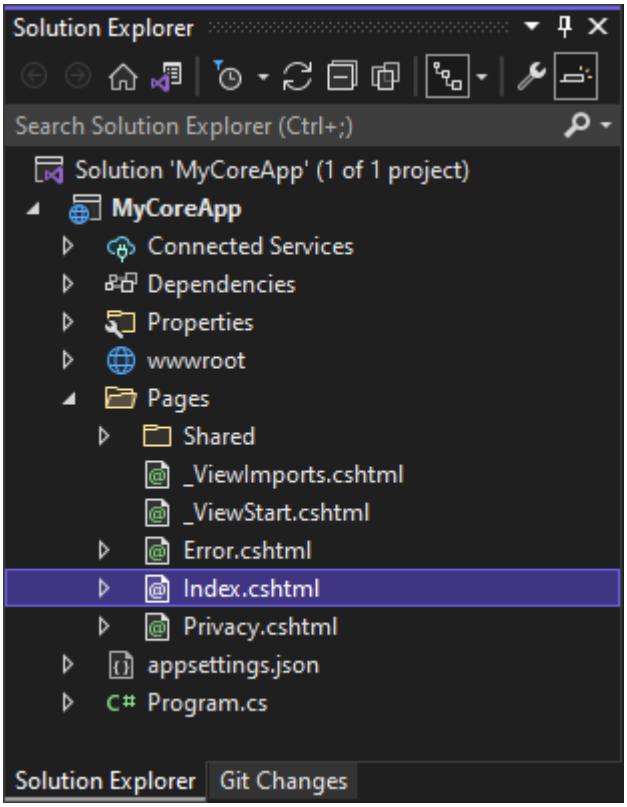
١٠. في أعلى موقع الويب، حدد الخصوصية Privacy لعرض التغييرات.



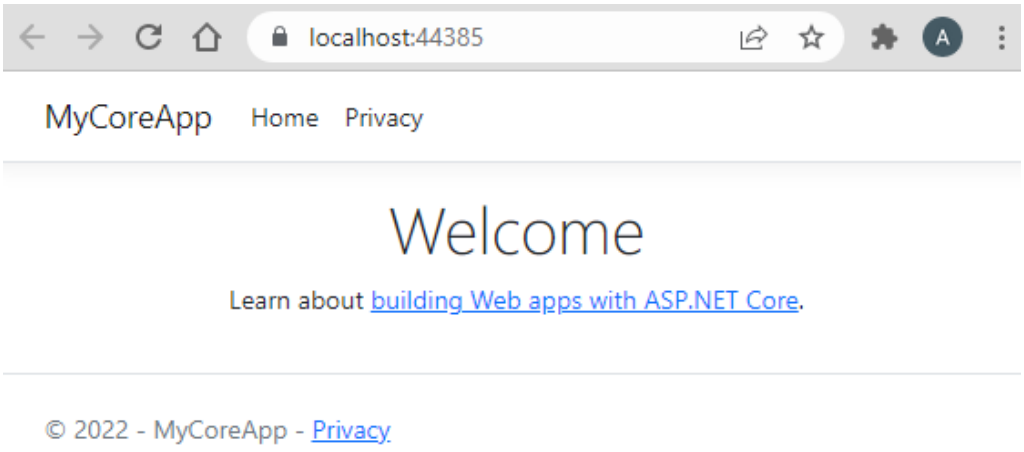
١١. أغلق متصفح الويب، ثم اضغط على **Shift + F5** لإيقاف عملية التصحيح.

تغيير الصفحة الرئيسية الخاصة بك

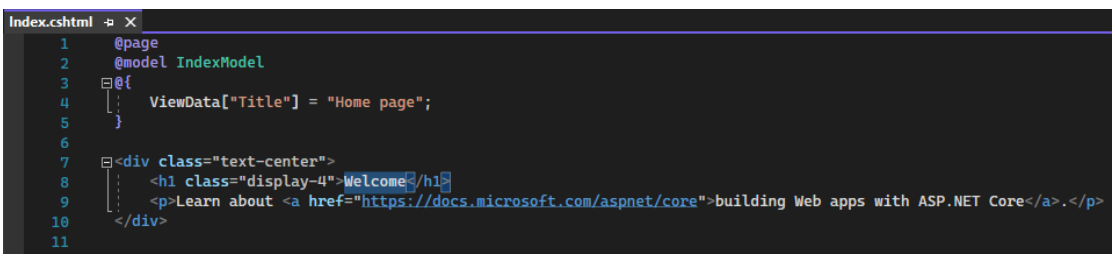
١. في مستكشف الحلول Solution Explorer قم بتوسيع مجلد الصفحات Pages ثم حدد Index.cshtml



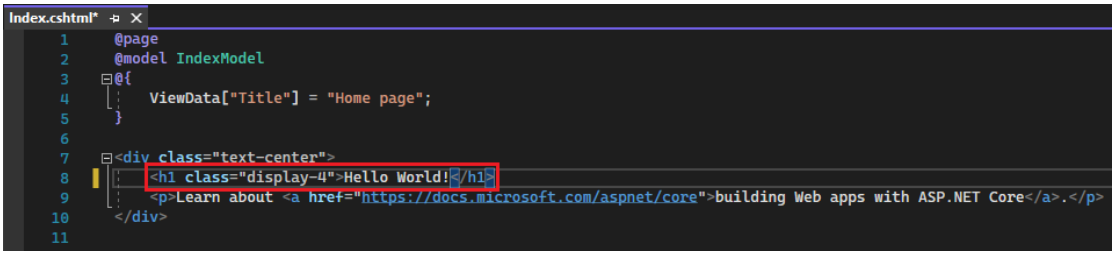
يتوافق ملف Index.cshtml مع صفحتك الرئيسية في تطبيق الويب، والذي يعمل في متصفح الويب.



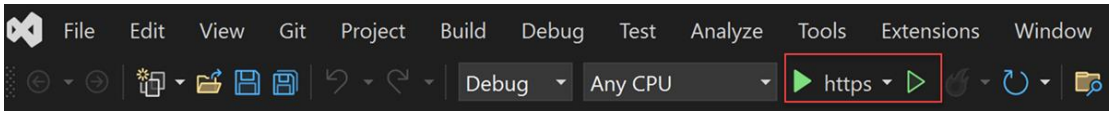
في محرر التعليمات، سترى تعليمات HTML للنص الذي يظهر في الصفحة **الرئيسية**.



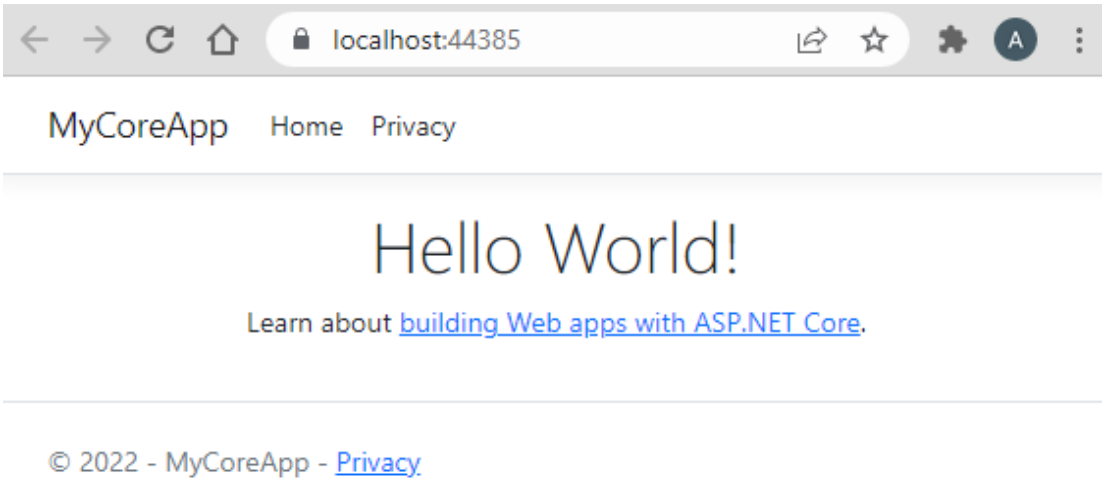
٢. استبدل نص الترحيب بنص مرحباً بالعالم Hello World!



٣. حدد زر **https** أو اضغط على **Ctrl + F5** لتشغيل التطبيق وفتحه في متصفح الويب.



٤. في متصفح الويب، يمكنك رؤية التغييرات الجديدة على الصفحة الرئيسية.



٥. أغلق متصفح الويب، واضغط على **Shift + F5** لإيقاف تصحيح الأخطاء، ثم احفظ مشروعك. يمكنك الآن إغلاق Visual Studio

تهانينا على إكمال هذا الجزء التعليمي! نأمل أن تكون قد استمتعت بتعلم C#, ASP.NET Core وبيئة التطوير المتكاملة Visual Studio IDE لمعرفة المزيد حول إنشاء تطبيق ويب أو موقع ويب باستخدام C# and ASP.NET تابع البرنامج التعليمي التالي Create a Razor Pages web app with ASP.NET Core

أو تعرف على كيفية وضع تطبيق الويب في حاويات باستخدام Docker Container Tools in Visual Studio

٩ مميزات برنامج Visual Studio

تصف هذه الوحدة ميزات للمطورين ذوي الخبرة أو المطورين الذين لديهم بالفعل معرفة بـ Visual Studio للحصول على مقدمة أساسية عن Visual Studio راجع الوحدة السابقة او راجع نظرة عامة على Visual Studio IDE overview

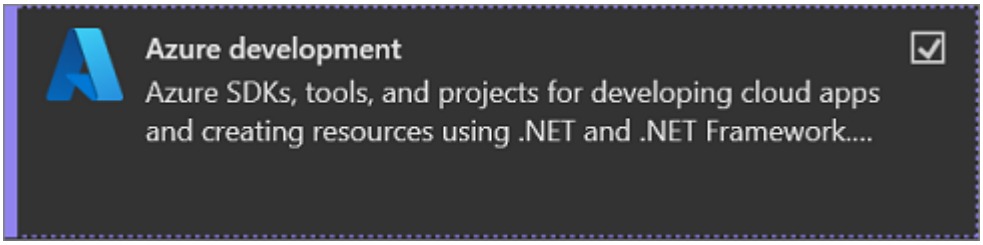
التثبيت المعياري Modular installation

في التثبيت المعياري modular installer لبرنامج Visual Studio يمكنك اختيار أحمال العمل workloads التي تريدها وتثبيتها، أحمال العمل عبارة عن مجموعات من الميزات التي تحتاجها لغات البرمجة أو المنصات للعمل، تساعد هذه الاستراتيجية المعيارية في الحفاظ على حجم تثبيت Visual Studio أصغر، بحيث يتم تثبيته وتحديثه بشكل أسرع.

لمعرفة المزيد حول إعداد Visual Studio على نظامك، راجع تثبيت Visual Studio

إنشاء تطبيقات Azure الممكّنة بالسحابة cloud-enabled Azure

يحتوي Visual Studio على مجموعة من الأدوات لإنشاء تطبيقات تدعم السحابة من Microsoft Azure بسهولة، يمكنك تكوين تطبيقات وخدمات Azure وإنشاؤها وتصحيح أخطائها وتعبئتها ونشرها مباشرةً من بيئة التطوير المتكاملة (IDE) الخاصة بـ Visual Studio للحصول على أدوات Azure وقوالب المشروع، حدد عبء عمل **Azure development** عند تثبيت Visual Studio



مهم:

تم إيقاف تشغيل نافذة Cloud Explorer في Visual Studio 2022 لمزيد من المعلومات، راجع إدارة الموارد المرتبطة بحسابات Azure

Manage the resources associated with your Azure accounts in Visual Studio Cloud Explorer

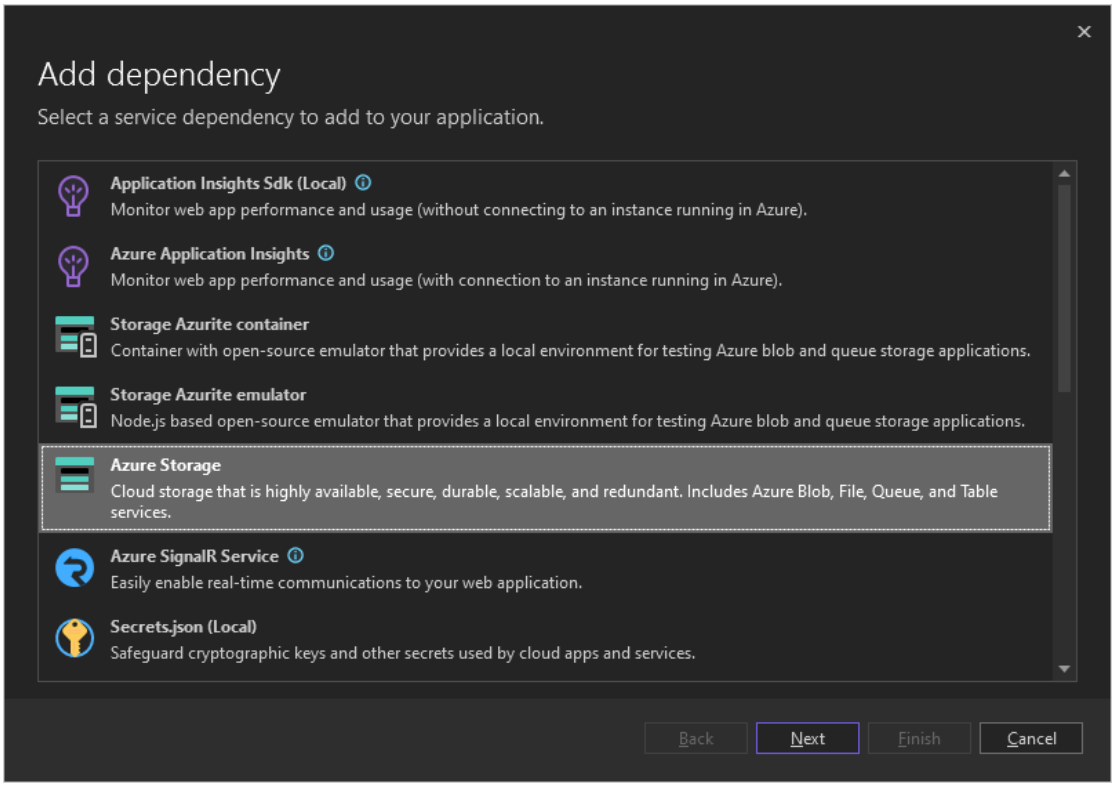
استخدم بوابة Azure للوصول إلى موارد Azure حسب الحاجة، يمكنك الاستمرار في استخدام عقدة Azure node الخاصة بـ Server Explorer في الإصدارات السابقة منVisual Studio

يمكنك استخدام خدمات Azure لتطبيقاتك عن طريق إضافة الخدمات المتصلة Connected Services مثل:

- Active Directory connected service لاستخدام Microsoft Entra ID (Microsoft Entra ID) للاتصال بتطبيقات الويب.
- Azure Storage connected service لتخزين الكائنات والطوابير والجداول blob storage, queues, and tables
- Key Vault connected service لإدارة الأسرار لتطبيقات الويب manage secrets for web apps

تعتمد الخدمات المتصلة Connected Services المتاحة على نوع مشروعك، أضف خدمة بالنقر بزر الماوس الأيمن على المشروع في مستكشف الحلول Solution Explorer واختيار إضافة > الخدمة المتصلة Add > Connected Service

في شاشة الخدمات المتصلة Connected Services حدد الرابط أو علامة الجمع لإضافة تبعية خدمة Add a service dependency في شاشة إضافة تبعية Add dependency حدد الخدمة التي تريد إضافتها، واتبع الشاشات للاتصال باشتراكك وخدمتك في Azure



لمزيد من المعلومات، راجع الانتقال إلى السحابة Move to the cloud With Visual Studio and Azure

إنشاء تطبيقات الويب web apps

يمكن أن يساعدك Visual Studio في كتابة تطبيقات للويب. يمكنك إنشاء تطبيقات ويب باستخدام ASP.NET, Node.js, Python, JavaScript, TypeScript يدعم Visual Studio العديد من أطر الويب، مثل web frameworks, such as Angular, jQuery, Express

يعمل ASP.NET Core and .NET Core على أنظمة التشغيل Windows, Mac, and Linux يعد ASP.NET Core تحديثًا رئيسيًا لـ MVC, WebAPI, SignalR تم تصميم ASP.NET Core من البداية لتوفير حزمة .NET بسيطة وقابلة للتكوين، لبناء تطبيقات وخدمات ويب حديثة قائمة على السحابة.

لمزيد من المعلومات راجع أدوات الويب الحديثة Modern web tooling

إنشاء تطبيقات وألعاب متعددة الأنظمة Build cross-platform apps and games

يمكن لـ Visual Studio إنشاء تطبيقات وألعاب لأنظمة التشغيل Windows, Mac, and Linux ولأجهزة Android, iOS, and other mobile devices باستخدام Visual Studio يمكنك إنشاء:

- تطبيقات .NET Core التي تعمل على أنظمة Windows, Mac, and Linux
- تطبيقات الهاتف المحمول لأنظمة التشغيل iOS, Android, and Windows باستخدام C# and F# باستخدام Xamarin
- ألعاب ثنائية وثلاثية الأبعاد في C# باستخدام أدوات Visual Studio Tools for Unity
- تطبيقات C++ الأصلية لأجهزة iOS, Android, and Windows شارك التعليمات البرمجية المشتركة في مكتبات iOS, Android, and Windows باستخدام C++ للتطوير عبر الأنظمة الأساسية C++ for cross-platform development

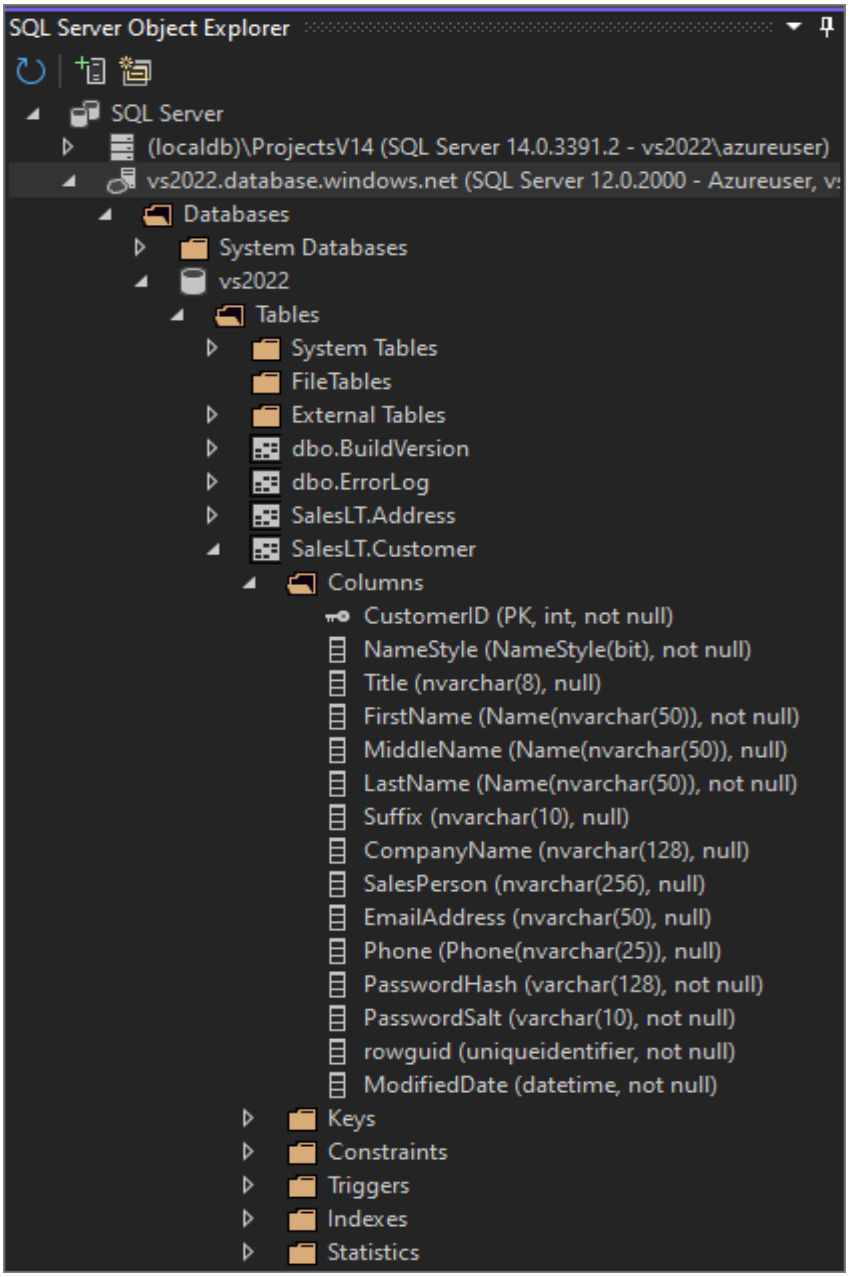
الاتصال بقواعد البيانات Connect to databases

يساعدك Server Explorer على استعراض وإدارة مثيلات الخادم والأصول محليًا وعن بُعد Azure, Microsoft 365, Salesforce.com, and websites

لفتح Server Explorer اختر View > Server Explorer لمزيد من المعلومات حول استخدام Server Explorer راجع إضافة اتصالات جديدة Add new connections

يوفر لك SQL Server Object Explorer عرضًا لكائنات قاعدة البيانات الخاصة بك، على غرار SQL Server Management Studio باستخدام SQL Server Object Explorer يمكنك القيام بأعمال إدارة وتصميم قواعد البيانات البسيطة. تتضمن الأمثلة تحرير بيانات الجدول ومقارنة المخططات وتنفيذ الاستعلامات باستخدام القوائم السياقية editing table data, comparing schemas, and executing queries by using contextual menus

لفتح SQL Server Object Explorer حدد الرمز الخاص به في الجزء العلوي من نافذة Server Explorer أو حدد View > SQL Server Object Explorer من القائمة العلوية في Visual Studio



SQL Server Data Tools (SSDT) هي بيئة تطوير قوية لـ SQL Server وAzure SQL Database وAzure SQL Data Warehouse باستخدام SSDT يمكنك إنشاء قواعد بيانات وتصحيح أخطائها وصيانتها وإعادة صياغتها. يمكنك العمل مع مشروع قاعدة بيانات أو مباشرة مع مثيل قاعدة بيانات متصلة محليًا أو خارجه. للحصول على SSDT استخدم Visual Studio Installer لتثبيت عبء عمل تخزين البيانات ومعالجتها Data storage and processing

تصحيح واختبار وتحسين أخطاء التعليمات Debug, test, and improve

عند كتابة التعليمات البرمجية، يجب عليك تشغيلها واختبارها بحثًا عن الأخطاء والأداء. باستخدام نظام تصحيح الأخطاء في Visual Studio يمكنك تصحيح أخطاء التعليمات البرمجية التي يتم تشغيلها في مشروعك المحلي، أو على جهاز بعيد، أو على محاكي جهاز انتقل عبر التعليمات البرمجية عبارة تلو الأخرى، وافحص المتغيرات أثناء التنقل. أو قم بتعيين نقاط توقف لا يتم الوصول إليها إلا عندما يكون شرط معين صحيحًا. يمكنك إدارة خيارات التصحيح في محرر التعليمات البرمجية نفسه، حتى لا تضطر إلى ترك تعليماتك البرمجية.

لمزيد من المعلومات حول تصحيح الأخطاء في Visual Studio راجع النظرة الأولى على مصحح الأخطاء

لتحسين أداء التطبيق، تحقق من ميزة إنشاء ملف تعريف في Visual Studio

يوفر Visual Studio خيارات اختبار مثل اختبار الوحدة، واختبار الوحدة المباشرة واختبار التحميل والأداء unit testing, Live Unit Testing, IntelliTest, and load and performance testing كما يتمتع Visual Studio بإمكانيات تحليل التعليمات البرمجية المتقدمة للعثور على عيوب التصميم والأمان وغيرها.

نشر تطبيقك النهائي Deploy your finished application

يحتوي Visual Studio على أدوات لنشر تطبيقك للمستخدمين أو العملاء من خلال Microsoft Store أو موقع SharePoint أو تقنيات InstallShield أو Windows Installer technologies يمكنك الوصول إلى كل هذه الخيارات من خلال Visual Studio IDE لمزيد من المعلومات، راجع نشر التطبيقات والخدمات والمكونات Deploy applications, services, and components

إدارة تعليمات المصدر والتعاون مع الآخرين Manage source code and collaborate with others

في Visual Studio يمكنك إدارة الكود المصدر في مستودعات Git التي يستضيفها أي موفر، بما في ذلك GitHub يمكنك أيضًا البحث عن Azure DevOps Server للاتصال به.

للحصول على التفاصيل الكاملة، راجع صفحة Git experience in Visual Studio وللحصول على برنامج تعليمي خطوة بخطوة حول كيفية الاتصال بمستودع Gitأو Azure DevOps باستخدامVisual Studio راجع صفحة فتح مشروع من مستودع Open a project from a repo

نصيحة:

نستمر في بناء مجموعة ميزات Git وتكرارها بناءً على تعليقاتك. لمزيد من المعلومات حول تحديث الميزة الأخير بالإضافة إلى رابط للاستطلاع حيث يمكنك مشاركة ملاحظاتك عليه، راجع منشور مدونة دعم المستودعات المتعددة Multi-repo support in Visual Studio

تحسين تجربة Visual Studio على Microsoft Dev Box

ملحوظة:

تتوفر هذه الميزة حاليًا في المعاينة العامة. تتعلق هذه المعلومات بميزة قد يتم تعديلها بشكل كبير قبل إصدارها. لا تقدم Microsoft أي ضمانات، صريحة أو ضمنية، فيما يتعلق بالمعلومات المقدمة هنا.

باستخدام Visual Studio 17.7 Preview 3 يمكنك إنشاء ذاكرة تخزين مؤقتة لبرنامج Visual Studio مسبقًا وتضمينها في صورة صندوق التطوير الخاص بك ونتيجة لذلك، سيقوم Visual Studio بتحميل الحل الخاص بك وتمكين ميزات IDE الرئيسية بشكل أسرع على صندوق التطوير الخاص بك. يمكنك أيضًا تحسين

أداء Git على المستودعات الكبيرة من خلال تمكين تحسينات رسم بياني لالتزامات Git في صور صندوق التطوير.

توسيع برنامج Visual Studio

إذا لم يكن Visual Studio يحتوي على الوظيفة الدقيقة التي تحتاجها، فيمكنك إضافتها. قم بتخصيص IDE بناءً على سير العمل والأسلوب الخاصين بك، وأضف الدعم للأدوات الخارجية غير المتكاملة مع Visual Studio وقم بتعديل الوظائف الحالية لزيادة إنتاجيتك. للحصول على أحدث إصدار من أدوات التوسعة الخاصة بـ Visual Studio (VSSDK) راجع Visual Studio SDK

يمكنك استخدام منصة Roslyn .NET Compiler Platform لكتابة محللي التعليمات البرمجية ومولدات التعليمات البرمجية الخاصة بك. يمكنك العثور على كل ما تحتاجه في Roslyn

ابحث عن الامتدادات الموجودة existing extensions لبرنامج Visual Studio والتي أنشأها مطوروMicrosoft ومجتمع تطوير Visual Studio

لمعرفة المزيد حول توسيع Visual Studio راجع توسيع Visual Studio IDE

مقدمة عن المشاريع والحلول projects and solutions

يستكشف هذا الدرس التمهيدي معنى إنشاء حل ومشروع solution and a project الحل solution عبارة عن حاوية لتنظيم مشروع أو أكثر من مشاريع التعليمات البرمجية ذات الصلة، مثل مشروع مكتبة فئة class library ومشروع اختبار مماثل corresponding test project

كتمرين تعليمي لفهم مفهوم المشروع، ستقوم بإنشاء حل ومشروع من الصفر، عادةً، ستستخدم قوالب مشروع Visual Studio لإنشاء مشاريع جديدة، ستنظر أيضًا إلى خصائص المشروع وبعض الملفات التي يمكن أن يحتويها، وإنشاء مرجع من مشروع إلى آخر.

ملحوظة:

لا يتطلب تطوير التطبيقات في Visual Studio حلولاً ومشاريع، يمكنك فقط فتح مجلد يحتوي على التعليمات البرمجية والبدء في الترميز، والبناء وتصحيح الأخطاء، على سبيل المثال، قد لا يحتوي مستودع **GitHub** على مشاريع وحلول Visual Studio لمزيد من المعلومات، راجع تطوير التعليمات البرمجية في Visual Studio بدون مشاريع أو حلول **Develop code in Visual Studio without projects or solutions**

إذا لم تقم بتثبيت Visual Studio بالفعل، فانتقل إلى صفحة تنزيلات Visual Studio لتثبيته مجانًا.

الحلول والمشاريع Solutions and projects

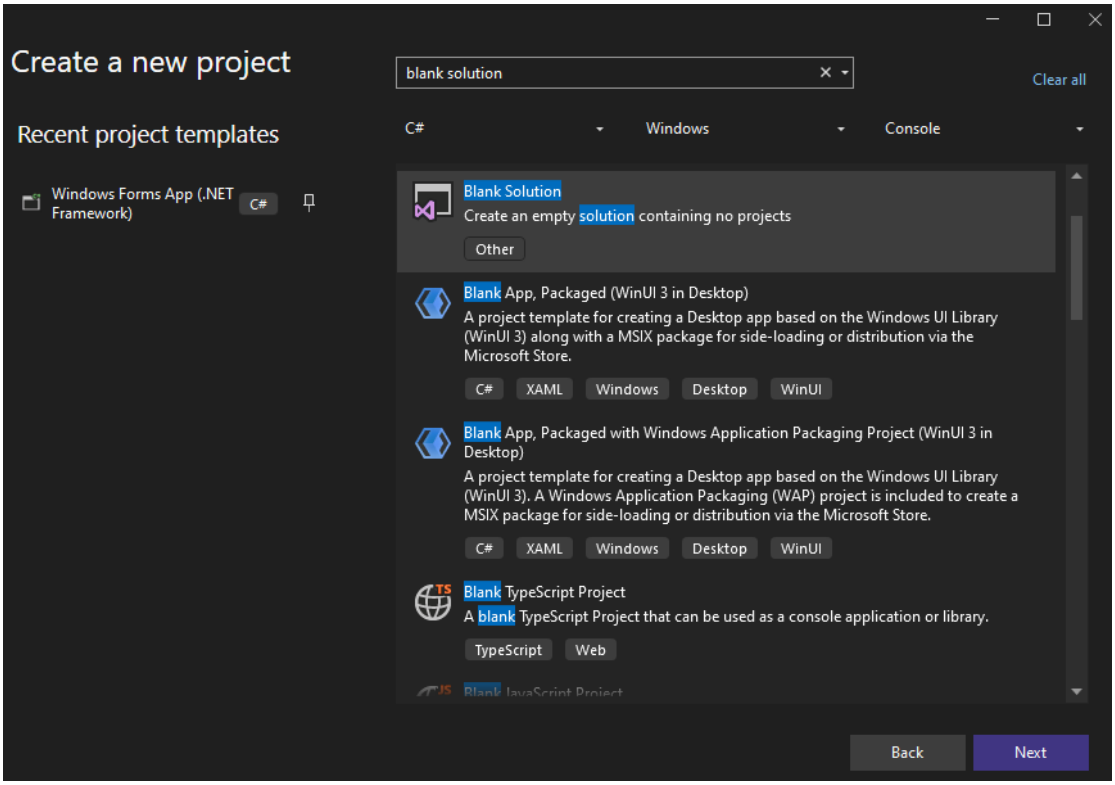
في Visual Studio لا يعد الحل "إجابة" الحل ببساطة عبارة عن حاوية يستخدمها Visual Studio لتنظيم مشروع واحد أو أكثر من المشاريع ذات الصلة. عند فتح حل، يقوم Visual Studio تلقائيًا بتحميل جميع المشاريع التي يحتوي عليها الحل.

إنشاء حل Create a solution

ابدأ استكشافك بإنشاء حل فارغ، بعد التعرف على Visual Studio ربما لن تقوم بإنشاء حلول فارغة في كثير من الأحيان، عند إنشاء مشروع جديد، يقوم Visual Studio تلقائيًا بإنشاء حل للمشروع ما لم يكن الحل مفتوحًا بالفعل.

١. افتح Visual Studio في نافذة البدء اختر إنشاء مشروع جديد Create a new project

٢. في صفحة إنشاء مشروع جديد Create a new project اكتب الحل الفارغ blank solution في مربع البحث، ثم حدد قالب الحل الفارغ blank solution ثم حدد التالي Next



نصيحة:

إذا كان لديك عدة أحمال عمل workloads مثبتة، فقد لا يظهر قالب الحل الفارغ blank solution في أعلى قائمة نتائج البحث، حاول التمرير عبر نتائج أخرى استنادًا إلى بحثك للعثور على القالب.

٣. في صفحة تكوين مشروعك الجديد Configure your new project قم بتسمية الحل QuickSolution ثم حدد إنشاء Create

يظهر حل QuickSolution في مستكشف الحلول Solution Explorer على الجانب الأيمن من Visual Studio ستستخدم مستكشف الحلول كثيرًا لاستعراض محتويات مشاريعك.

أضف مشروعًا

الآن أضف مشروعك الأول إلى الحل، ابدأ بمشروع فارغ وأضف العناصر التي تحتاجها.

١. انقر بزر الماوس الأيمن فوق الحل 'QuickSolution' في مستكشف الحلول Solution Explorer ثم حدد إضافة، مشروع جديد Add > New Project من قائمة السياق.

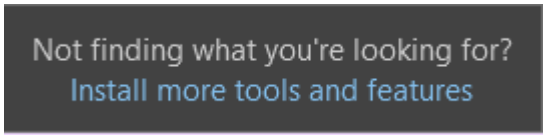
٢. في صفحة إضافة مشروع جديد Add a new project اكتب empty في مربع البحث في الأعلى، وحدد C# ضمن كل اللغات All languages

٣. حدد قالب مشروع C# فارغ Empty Project (.NET Framework) ثم حدد التالي Next

ملحوظة:

يستخدم Visual Studio المثبت المُستند إلى عبء العمل workload لتثبيت المكونات التي تحتاجها فقط لنوع التطوير الذي تقوم به، إذا لم تتمكن من رؤية قالب Empty Project (.NET Framework) فأنت بحاجة إلى تثبيت عبء العمل تطوير برامج سطح المكتب .NET desktop development في Visual Studio

تتمثل إحدى الطرق السهلة لتثبيت عبء عمل جديد، عند إنشاء مشروع جديد في تحديد رابط تثبيت المزيد من الأدوات والميزات Install more tools and features أسفل النص الذي يقول **Not finding what you're looking for?** "لم تجد ما تبحث عنه؟"



في برنامج تثبيت the Visual Studio Installer حدد عبء عمل تطوير سطح مكتب .NET desktop development. ثم حدد تعديل Modify

٤. في نافذة تكوين مشروعك الجديد Configure your new project قم بتسمية المشروع QuickDate ثم حدد إنشاء Create

يظهر مشروع QuickDate أسفل الحل في مستكشف الحلول، يحتوي المشروع على عقدة مراجع References node وملف واحد يسمى App.config

إضافة عنصر item إلى المشروع

أضف ملف التعليمات إلى مشروعك الفارغ.

١. من النقر بزر الماوس الأيمن أو قائمة السياق لمشروع QuickDate في مستكشف الحلول Solution Explorer حدد إضافة، عنصر جديد Add > New Item

سيتم فتح مربع الحوار إضافة عنصر جديد Add New Item حدد إظهار كافة القوالب Show All Templates إذا تم فتح مربع الحوار في عرض مضغوط

٢. قم بتوسيع عناصر Visual C# Items ثم حدد Code في الجزء الأوسط، حدد قالب عنصر الفئة Class ضمن الاسم Name اكتب Calendar ثم حدد Add

يضيف Visual Studio ملفًا باسم Calendar.cs إلى المشروع، يمثل **cs.** في النهاية، امتداد الملف، لملفات التعليمات البرمجية بلغة C# يظهر ملف Calendar.cs في التسلسل الهرمي للمشروع المرئي في Solution Explorer ويفتح الملف في المحرر.

٣. استبدل محتويات ملف Calendar.cs بالتعليمات التالية:

```
using System;

namespace QuickDate
{
    internal class Calendar
    {
       static void Main(string[] args)
       {
        DateTime now = GetCurrentDate();
     Console.WriteLine($"Today's date is {now}");
            Console.ReadLine();
       }

      internal static DateTime GetCurrentDate()
        {
            return DateTime.Now.Date;
        }
    }
}
```

لا تحتاج إلى فهم كل ما تفعله التعليمات حتى الآن، قم بتشغيل التطبيق بالضغط على Ctrl + F5 وتأكد من أن التطبيق يطبع تاريخ اليوم في نافذة وحدة التحكم أو الإخراج القياسي، ثم أغلق نافذة وحدة التحكم.

أضف مشروعًا ثانيًا

تحتوي الحلول عادةً على أكثر من مشروع واحد، وغالبًا ما تشير هذه المشاريع إلى بعضها البعض، قد تكون بعض المشاريع في الحل عبارة عن مكتبات فئات، وقد تكون بعضها الآخر عبارة عن تطبيقات قابلة للتنفيذ، وقد تكون بعضها الآخر عبارة عن مشاريع اختبار وحدات أو مواقع ويب.

لإضافة مشروع اختبار الوحدة unit test project إلى الحل solution ابدأ من قالب المشروع حتى لا تضطر إلى إضافة ملف تعليمات آخري إلى المشروع.

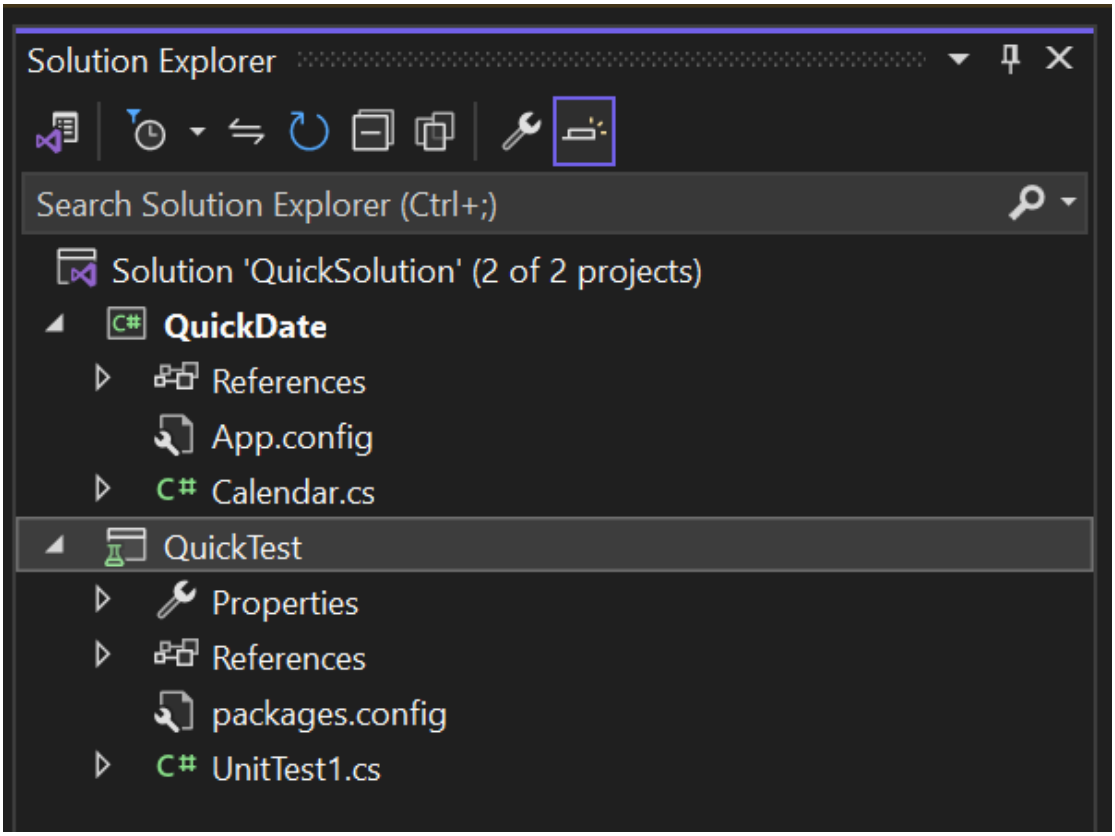
١. من النقر بزر الماوس الأيمن أو قائمة السياق للحل 'Solution 'QuickSolution في مستكشف الحلول، حدد إضافة، مشروع جديد Add > New Project

٢. في مربع الحوار إضافة مشروع جديد Add a new project اكتب اختبار الوحدة unit test في مربع البحث في الأعلى، ثم حدد C# ضمن كل اللغات All languages

٣. حدد قالب مشروع Unit Test Project (.NET Framework) ثم حدد التالي Next

٤. في صفحة تكوين مشروعك الجديد Configure your new project قم بتسمية المشروع QuickTest ثم حدد إنشاء Create

يضيف Visual Studio مشروع QuickTest إلى Solution Explorer ويفتح ملف UnitTest1.cs في المحرر.



إضافة مرجع المشروع Add a project reference

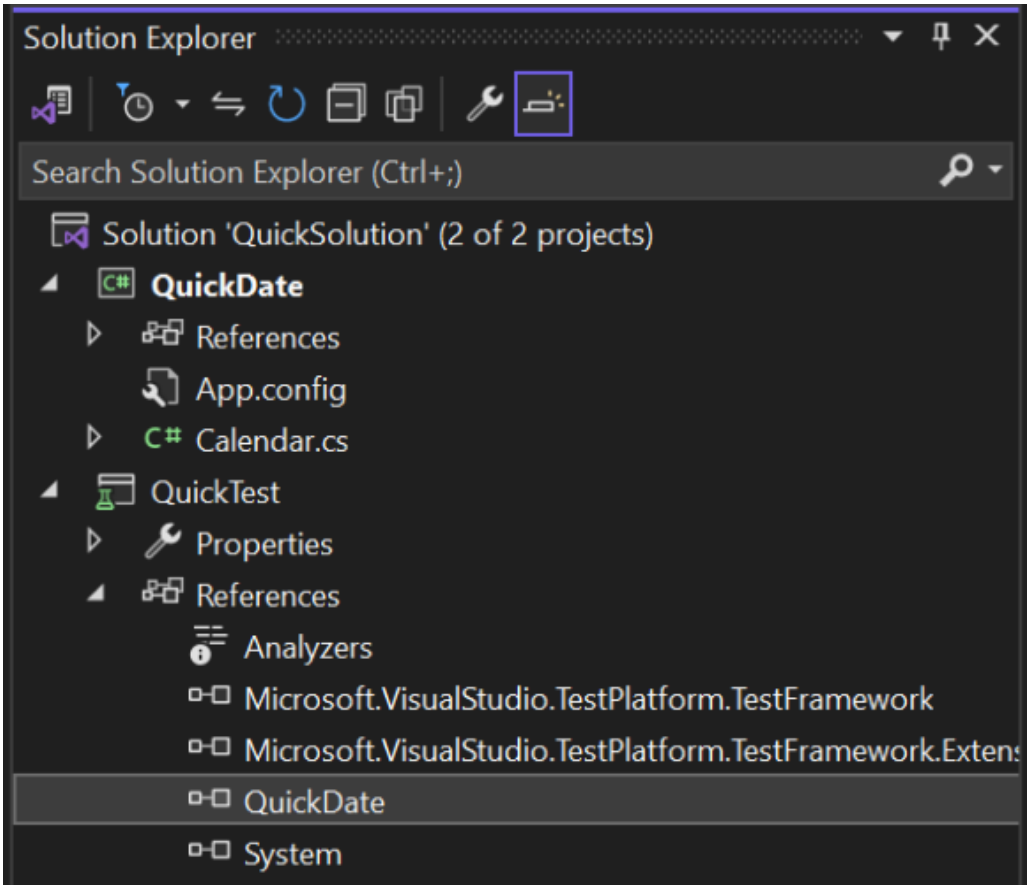
سوف تستخدم مشروع اختبار الوحدة unit test الجديد لاختبار أسلوبك في مشروع QuickDate لذا تحتاج إلى إضافة مرجع مشروع QuickDate إلى مشروع QuickTest

يؤدي إضافة المرجع إلى إنشاء تبعية أو اعتماد بناء build dependency بين المشروعين، مما يعني أنه عند بناء الحل، يقوم QuickDate بالبناء قبل QuickTest

١. في مستكشف الحلول، انقر بزر الماوس الأيمن فوق عقدة المراجع References node في مشروع QuickTest وحدد إضافة مرجع Add Reference من قائمة السياق.

٢. في مربع الحوار مدير المراجع Reference Manager حدد المشاريع Projects في الجزء الأوسط، حدد مربع الاختيار بجوار QuickDate ثم حدد موافق OK

يظهر مرجع لمشروع QuickDate ضمن مشروع QuickTest في مستكشف الحلول.



أضف تعليمات الاختبار test code

١. الآن قم بإضافة تعليمات الاختبار إلى ملف تعليمات الاختبار C# test code استبدل محتويات UnitTest1.cs بالتعليمات التالية:

```
using System;
using
Microsoft.VisualStudio.TestTools.UnitTesting;

namespace QuickTest
{
    [TestClass]
    public class UnitTest1
    {
        [TestMethod]
        public void TestGetCurrentDate()
        {
            Assert.AreEqual(DateTime.Now.Date,
QuickDate.Calendar.GetCurrentDate());
        }
    }
}
```

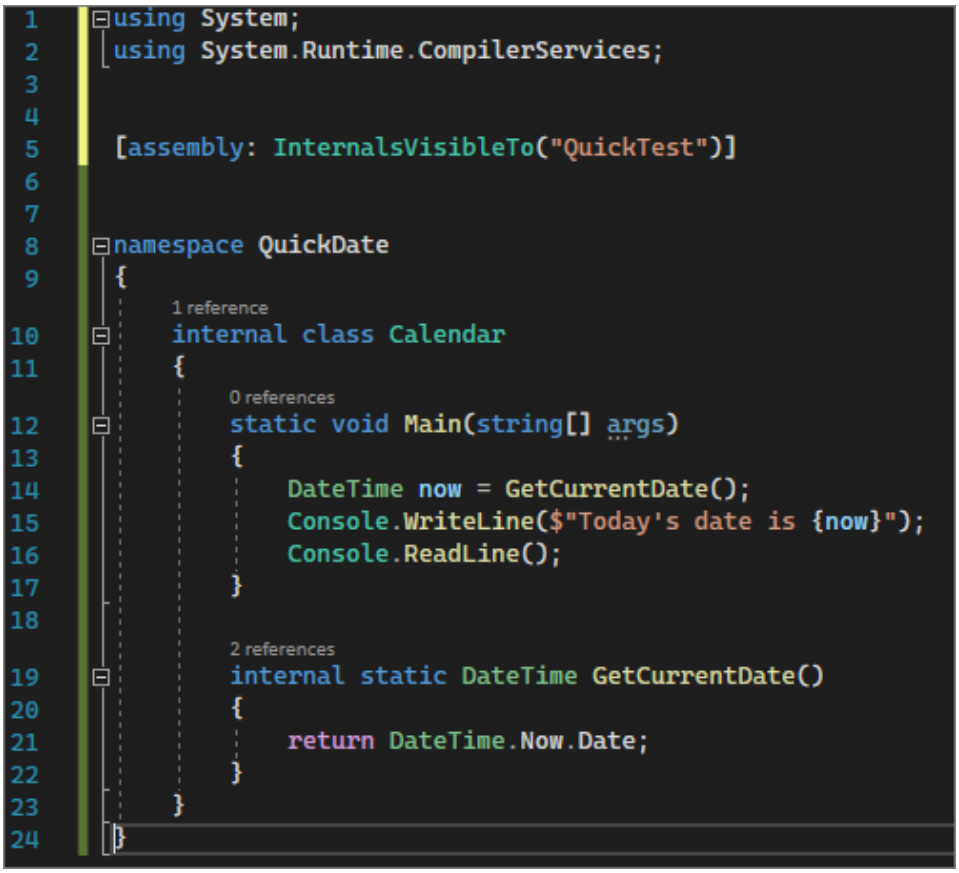
يظهر خط متعرج باللون الأحمر أسفل بعض التعليمات البرمجية، يمكنك إصلاح هذا الخطأ عن طريق جعل مشروع الاختبار تجميعًا صديقًا friend assembly لمشروع QuickDate

٢. في ملف Calendar.cs أضف العبارة التالية:

using statement and InternalsVisibleToAttribute attribute إلى أعلى الملف لحل الخطأ في مشروع الاختبار.

```
using System.Runtime.CompilerServices;

[assembly: InternalsVisibleTo("QuickTest")]
```

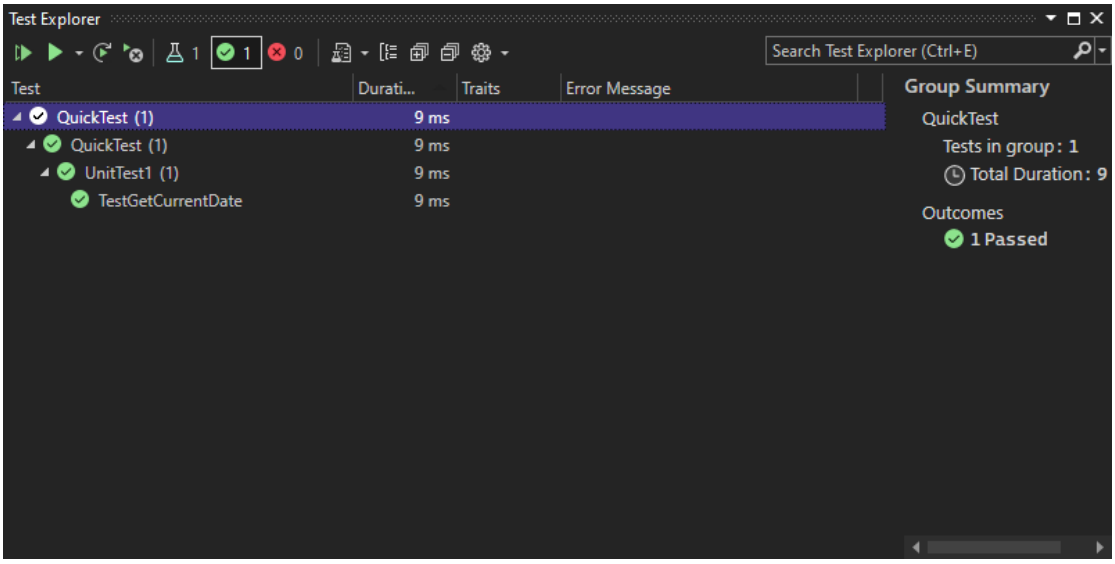
يجب أن تبدو تعليمات Calendar.cs مثل لقطة الشاشة التالية:



```
1  using System;
2  using System.Runtime.CompilerServices;
3
4
5   [assembly: InternalsVisibleTo("QuickTest")]
6
7
8  namespace QuickDate
9  {
       1 reference
10     internal class Calendar
11     {
           0 references
12         static void Main(string[] args)
13         {
14             DateTime now = GetCurrentDate();
15             Console.WriteLine($"Today's date is {now}");
16             Console.ReadLine();
17         }
18
           2 references
19         internal static DateTime GetCurrentDate()
20         {
21             return DateTime.Now.Date;
22         }
23     }
24 }
```

تشغيل اختبار الوحدة Run the unit test

للتأكد من أن اختبار الوحدة الخاص بك يعمل، اختر اختبار، تشغيل كافة الاختبارات Test > Run All Tests من شريط القائمة، تفتح نافذة مستكشف الاختبار Test Explorer ويجب أن ترى أن اختبار TestGetCurrentDate نجح.



يمكنك أيضًا فتح Test Explorer عن طريق اختيار Test > Test Explorer من شريط القائمة.

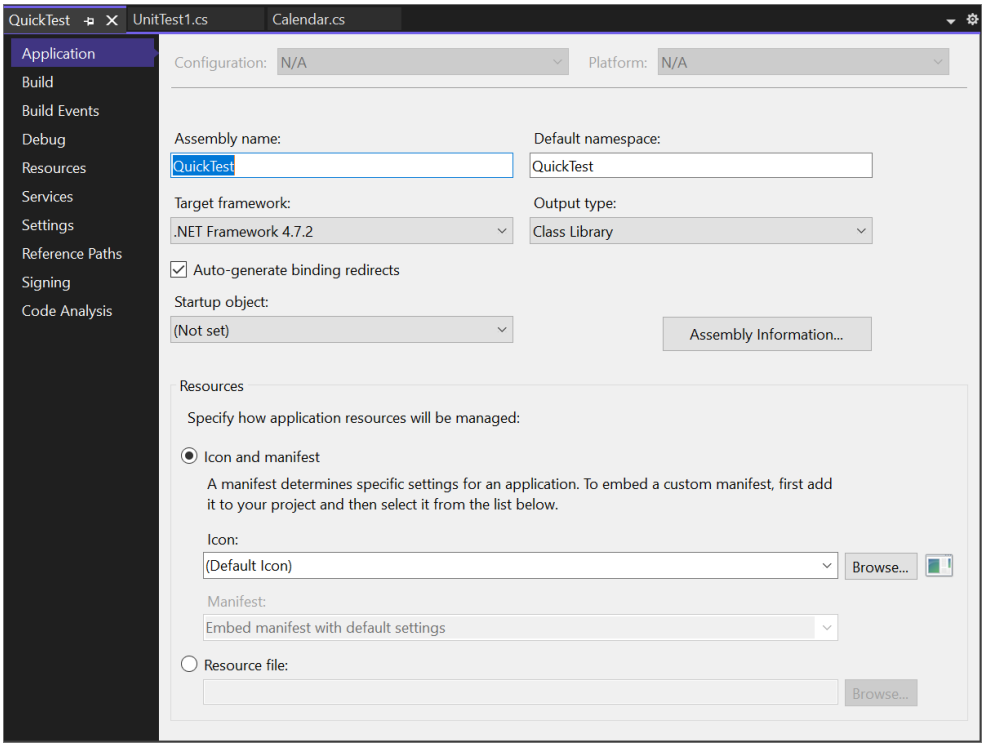
خصائص المشروع Project properties

يشير السطر الموجود في ملف Calendar.cs الذي يحتوي على السمة InternalsVisibleToAttribute إلى اسم التجميع assembly name أو اسم الملف لمشروع QuickTest قد لا يكون اسم التجميع هو نفسه اسم المشروع دائمًا. للعثور على اسم التجميع للمشروع، استخدم خصائص المشروع، تحتوي صفحات الخصائص على إعدادات مختلفة للمشروع.

١. في مستكشف الحلول Solution Explorer انقر بزر الماوس الأيمن فوق مشروع QuickTest وحدد خصائص Properties أو حدد المشروع واضغط على Alt + Enter

يتم فتح صفحات خصائص المشروع على علامة التبويب التطبيق Application اسم التجميع Assembly name لمشروع QuickTest هو QuickTest بالفعل.

إذا أردت، يمكنك تغيير الاسم هنا، عند إنشاء مشروع الاختبار، يتغير اسم الملف الثنائي الناتج من QuickTest.dll إلى <NewName>.dll



استكشف بعض علامات التبويب الأخرى في صفحات خصائص المشروع، مثل Build and Debug تختلف علامات التبويب هذه باختلاف أنواع الإنشاء والتصحيح المشاريع.