EVERYTHING

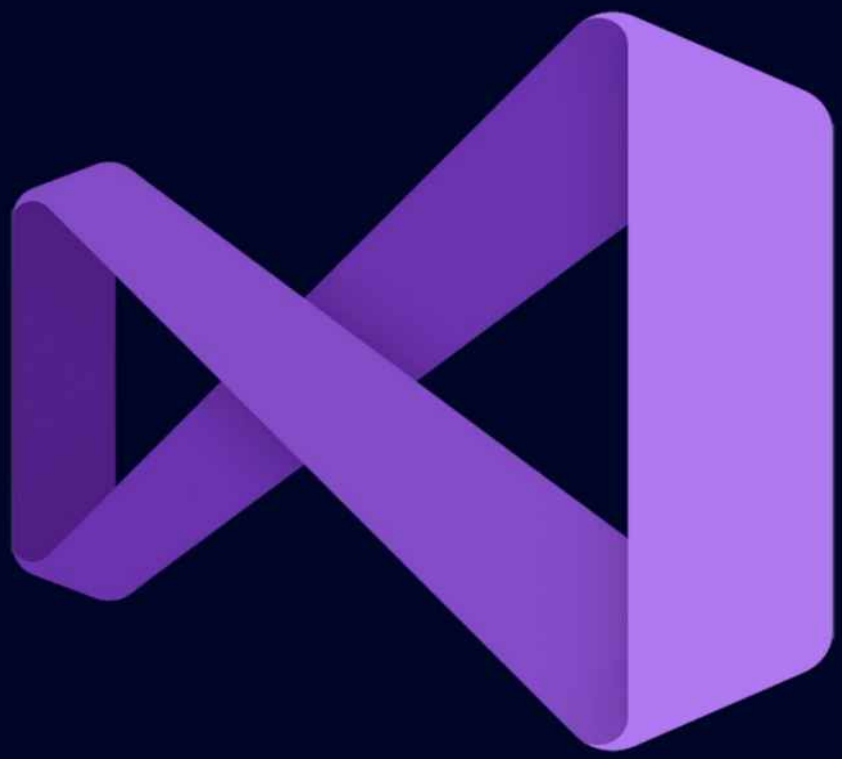
MICROSOFT

# VISUAL STUDIO

(Beginners, Experts & Seniors Guide)

2022

EVERYTHING YOU NEED TO KNOW ABOUT VISUAL STUDIO FOR CODING, PROGRAMMING & PROGRAMS DEVELOPMENT + PROFESSIONAL HACKS, TIPS & TRICKS FOR DEVELOPERS (BEGINNERS, EXPERTS & SENIORS GUIDE)



# EVERYTHING
# VISUAL STUDIO

Everything you Need to Know About Visual Studio for Coding, Programming & Programs Development + Professional Hacks, Tips & Tricks for Developers

(Beginners, Experts & Seniors Guide)

CARTY BINN

**Disclaimer and Terms of Use**

The author and publisher of this book and the accompanying materials have used their best efforts in preparing this book. The author and publisher make no representation or warranties with respect to the accuracy, applicability, fitness, or completeness of the contents of this book. The information contained in this book is strictly for informational purposes. Therefore, if you wish to apply the ideas contained in this book, you are taking full responsibility for your actions.

*Printed in the United States of America*

# CONTENTS

# CHAPTER 1

# GETTING STARTED WITH VISUAL STUDIO 2022

The Microsoft visual Studio is simply an integrated development environment. It is a popular software for .NET developers and is perfect to develop, debug, and deploy all the .NET apps and some other technology.

In this chapter, we are going to be taking you through the history, the installation process of the Visual studio, and the first setting before you start working with the IDE

## Technical requirements

The visual studio works on systems using Windows 10 or higher, a 64-bit processor; quad-core, or more. A minimum of 4GB. But, 8GB is recommended. You need to have at least 25 GB of space. Have administrator rights and there needs to be full internet access during the installation process.

## A brief history of Visual Studio

Version 13 of this Microsoft product is called Visual Studio 2022. Virtual Studio has gained a solid reputation among developers for its user-friendly interface, good support with timely updates, and potent tools for building scalable and clean code. Numerous platforms and technologies are supported by VS. VS is the ideal tool for different project types for many developers.

Before we go any further, we must go through the history of Visual Studio so that we can know how it evolved. The initial version of this product, Visual Studio 6.0, was launched in 1997. To function with Visual Basic 6.0, this version was developed. Then, in 2002, a new version was made available that supported.NET and C# (a new programming language at that time). It has since become the go-to tool for.NET developers.

Since VS 2005, Microsoft has adopted a new strategy with a free/premium version, which is a basic/free public version you could use for a personal

project, study, or midsize applications. Virtual Studio originally started as a premium application with a closed license.

Every two or three years, Microsoft publishes a new version of Visual Studio and offers updates for that version every two or three months, ensuring full support.

The development team introduced a fresh look and feel and other user experience enhancements in Virtual Studio 2012, which are also available in the 2022 version. Performance, the option to select between bright and dark schemes, and redesigned icons were among the most significant advancements in VS 2012 over earlier versions.

# Visual studio Flavors

**There have been three Visual Studio versions since 2012. Here they are:**

- Professional
- Community
- And Enterprise

**Here are how they work:**

## Visual studio community

This is the free version and it gives you the basic ability to create, debug and deploy .NET applications and also the collaboration instruments that have been integrated into the VS. You can only have five users in the Visual Studio Community and are limited only to the non-enterprise organizations.

**Here are the features you get:**

- You get the basic debugging tools for inspecting during coding.
- You get a performance and a diagnostic hub to analyze the performance of the application and the memory usage.
- You get the refractory tools that you can use to clean and style codes.
- You can test units.
- You can use the function to navigate to the definition of the method.
- You can collaborate in real-time.

Students, freelancers, and small businesses can use this edition.

While this edition comes with all the essential tools you'll need daily, there are specific situations involving unit testing, memory, or inspection when these tools are insufficient.

### Visual studio Professional

The visual Studio Professional is the licensed version of Visual studio and you can get it by subscribing. It is perfect for teams that have more than five developers. You get the same tools in the Visual Studio Community and you are also going to be getting CodeLens.

It is only $45 per month for each user and the professional subscription also gives you $50 in Azure credit, training support, and also the Azure Dev0ps.

### Visual studio Enterprise

**This is essentially like a premium subscription to the Visual Studio where you are going to be getting the Visual Studio community and professional features and also some other tools like:**

- Live unit test.
- The snapshot debugger lets you save snapshots when you are debugging and there is an error.
- You get a performance analysis tool for mobile applications.
- You also get Architectural layer diagrams.

This plan only costs $250 per month and an Azure credit of $150, Power Bi pro and Azure Dev0ps with test plans, and also all of the features of the Virtual studio.

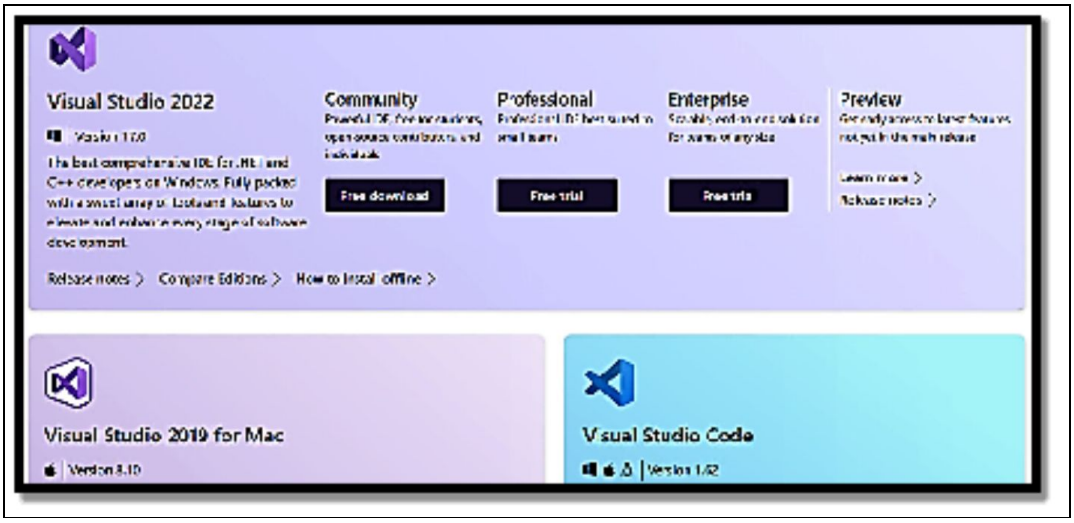
### Installing Visual Studio 2022

We are now going to be taking you through the process of installing Visual Studio in 2022.

## Getting the installer from the website

There is not a lot that is required before installing. what you just need to do is type VISUAL STUDIO COMMUNITY DOWNLOAD inside any browser. Then you are going to get the link to download it from the official page. Here is the link however:
https://visualstudio.microsoft.com/downloads/

When you are on the webpage, enter the visual studio and select the drop-down control that is marked **download visual studio** and choose **community.**

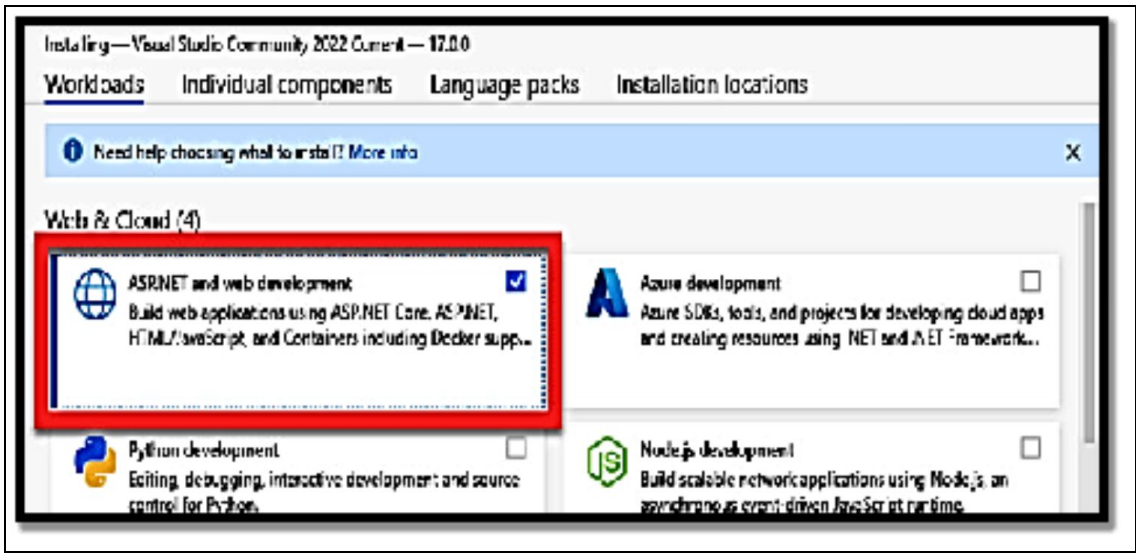



# The installation processes

After installing the downloader, you then need to run it so that the installer can update.

The development team of VS often updates the IDE; therefore, the installer would always search for the most recent update to carry out the relevant installation.

The opening screen of the installer consists of four main sections with the Workloads portion being the one that is displayed by default. It will be shown after the installer update is finished.




There are workloads available for a variety of project types, including Python-focused projects, desktop-focused projects, and mobile applications.

You must choose the workload ASP.NET and web development, as shown in the image below.  to install the templates and tools for web development.

Each job consists of a specific collection of tools and parts related to the chosen technology.

However, using the Individual components tab, it is also feasible to choose each of these components separately.

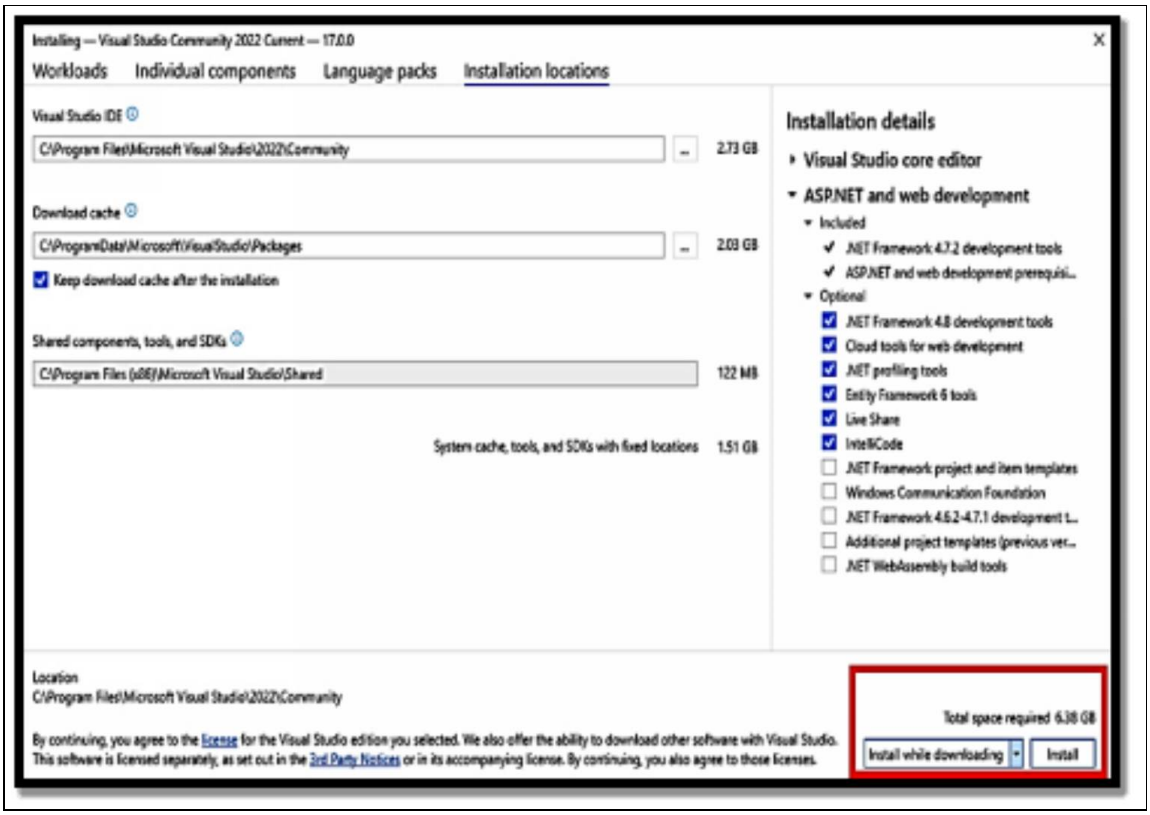

The parts are grouped in different categories like .NET, databases, clouds, and servers. You just have to choose a component and then add it to the installation.

In the **language packs** interface**,** you can pick a language.

Since the downloaded version of the VS installation generally matches the default language, this is quite helpful. From this point, you may choose a different language to swap between during your development process by deselecting the default language.

And finally, in the tab called **installation location,** you can then configure the paths for the system in both the visual Studio IDE and also the download cache.

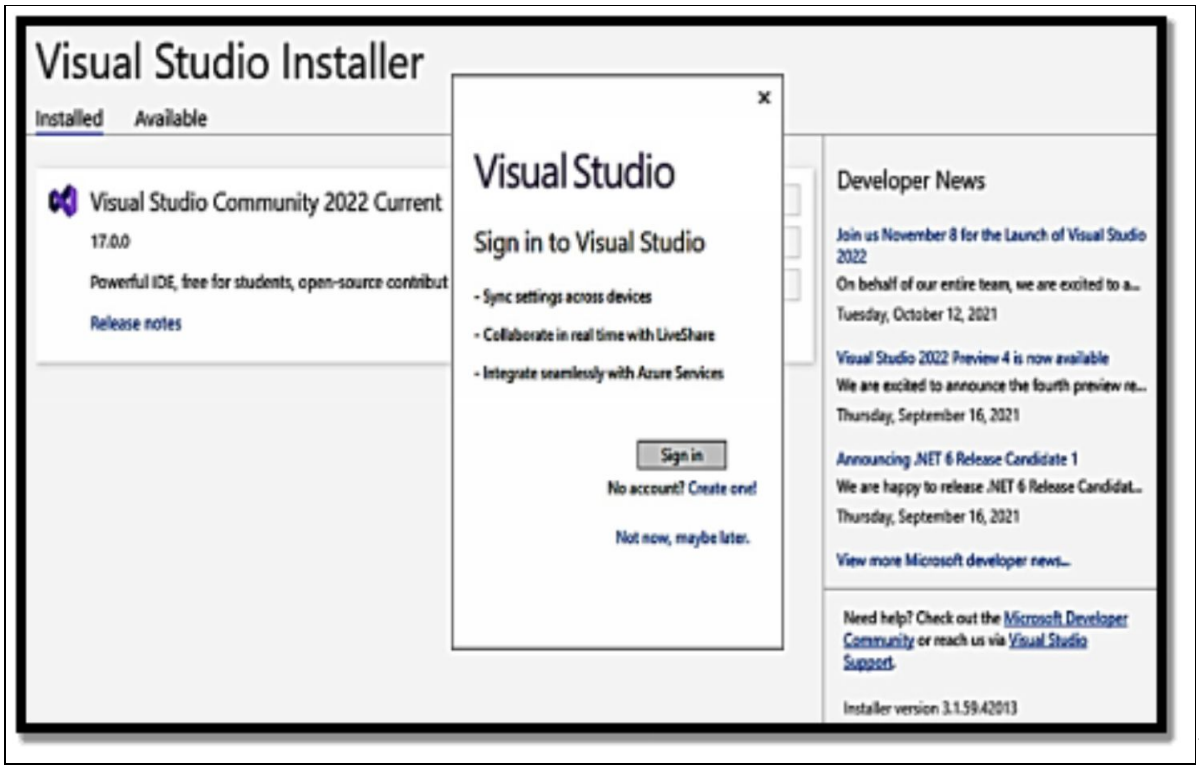
The next step is to pick the installation method from a list of alternatives. You may either download and install all the required components at once, or you can download them all first and install them later.




Then if you are happy with the installation size, you can then select **install** to begin the installation process.

Then you are going to see a window that gives you the information on the download progress.

And when the installation is done, you are going to see a window telling you to login into the Microsoft account so that you can get the license for the use of the tool. we will suggest that you log in here so that you do not lose access to the tool in the future.




Once you get a license for visual studio, you will see the startup screen to create, clone, and open the projects, so that you can see the installation is successful.

Once you install the visual studio and you see the first screen, open the visual studio with the **continue without code** option. You then will see the main screen inside the visual studio even though you did not open or create any project.

# Visual studio for Mac

Visual Studio also has a version for mac users. Developers can use this version just like they would in windows. Because of the incompatibility problems you are not going to get the .NET framework inside the Visual studio for Mac but only on windows.

You may customize your toolbar after launching a project, but by default, Visual Studio for Mac features the Solution Explorer, Properties, and Errors sections. Developers that require debugging and performance capabilities, as well as those who wish to create native apps for Android, iOS, and macOS utilizing the.NET MAUI (technology to create native apps using C#

and XAML or using Blazor) platform, should strongly consider adopting Visual Studio for Mac.

# Revision

1. What is a visual studio community?
2. How do you get the installer from the website?
3. What is a visual studio enterprise?

# Summary

You now know what Visual Studio is You also know the several IDE versions you may select to meet your project's demands, how to install Visual Studio 2022 for Windows, and how Visual Studio for Mac works in general.

# CHAPTER 2
# CONFIGURING THE IDE

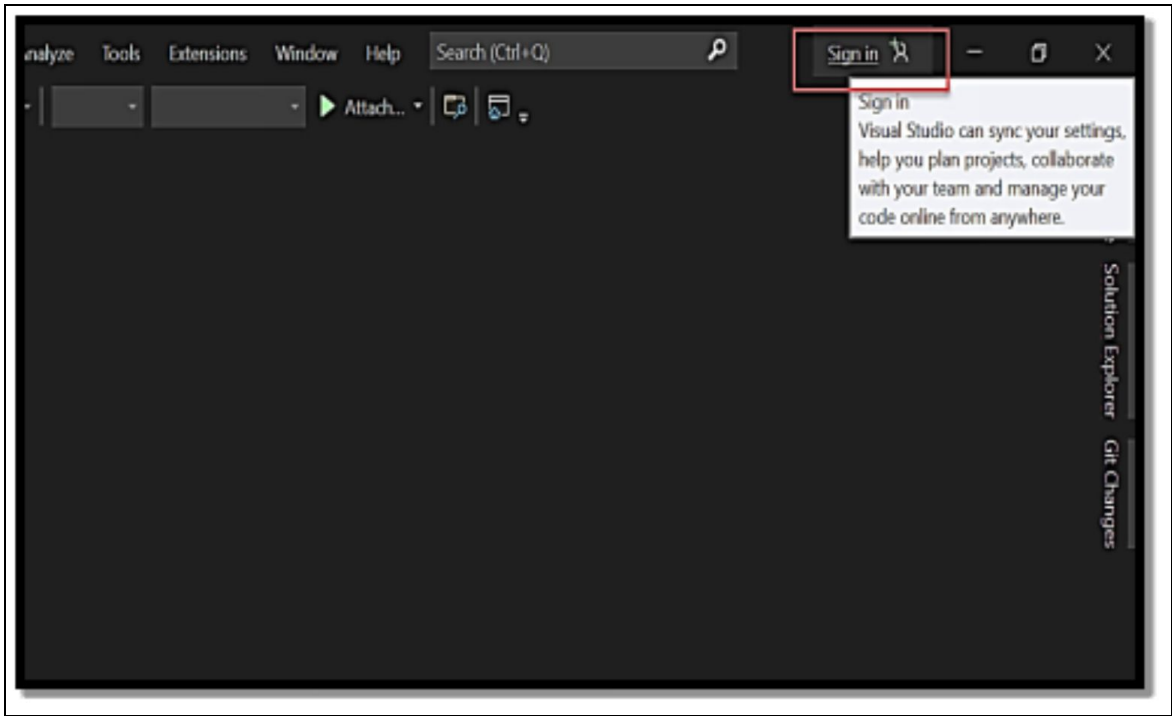
In this chapter, we are going to be going through the configuration settings in Visual Studio and improving productivity by giving you those exact things that you need both with colors and the window direction that you use.

## Technical Requirements

Once you install Visual Studio then you can begin configuring the IDE

### Synchronizing accounts and settings

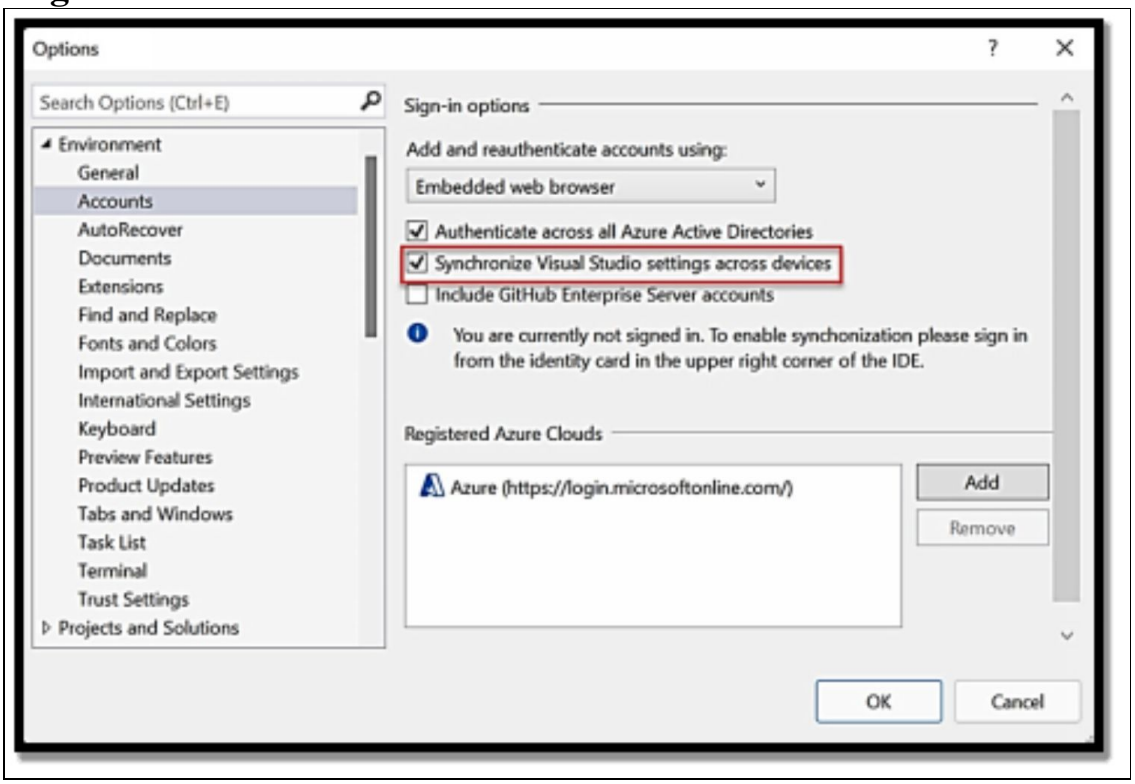
You can sync the configuration that you make across different computers. Once you have a Microsoft account you need to open Visual Studio. It is the first thing that is requested of you once you start visual studio or once you modify the account for use in Visual Studio direction, when you select the **sign-in** option on the top right side of the IDE




**Here are the configurations that you can synchronize with this process.**

- The themes and the menu settings.
- The keyboard shortcut.
- The fonts and the colors.
- The user-defined window layout configurations.
- Text editor settings.

However, if you do not want to synchronize the configuration on a computer then you need to go to **tools** then **option,** then **environment,** and finally **accounts.** There you can now disable **synchronize visual studio settings across devices.**




# Configuring the color scheme

One of the most significant methods to personalize Visual Studio 2022 is by changing the colors.

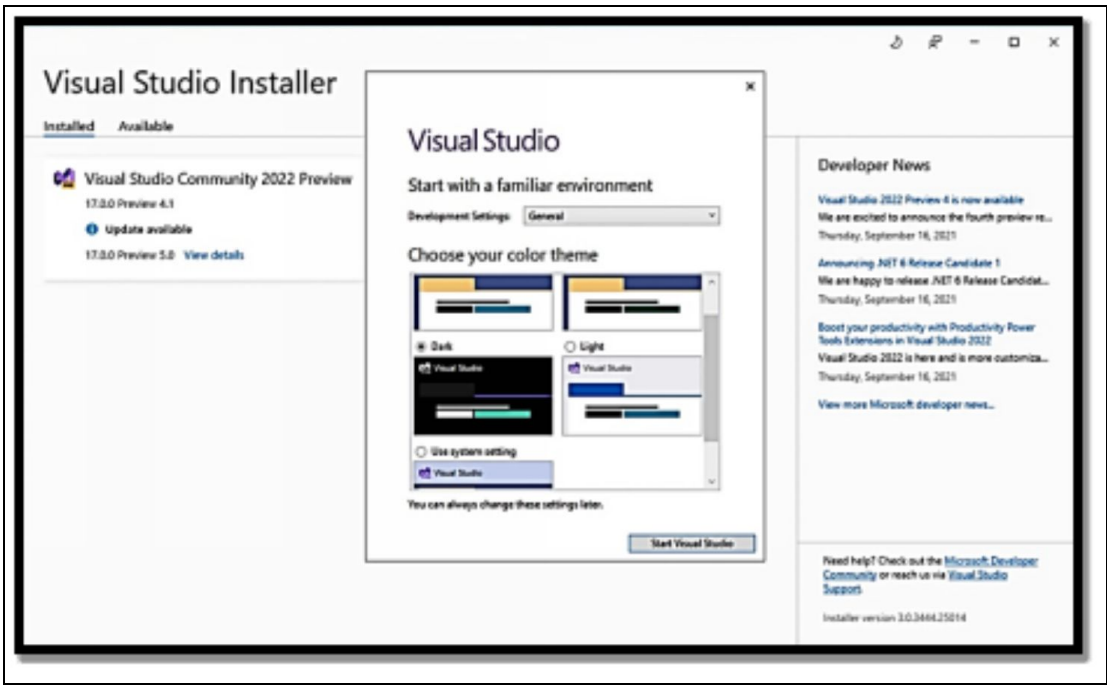
Regardless of whether you're in a well-lit area or are transitioning from another programming environment, you'll undoubtedly want to modify the colors to suit your tastes.

**There are two ways we may alter the color scheme in Visual Studio 2022 to our liking:**

- The visual studio default themes
- Visual studio color theme designer

# Visual studio default themes

Immediately after you open Visual Studio 2022 for the first time, you are going to get the following window telling you which development settings and color theme you prefer.



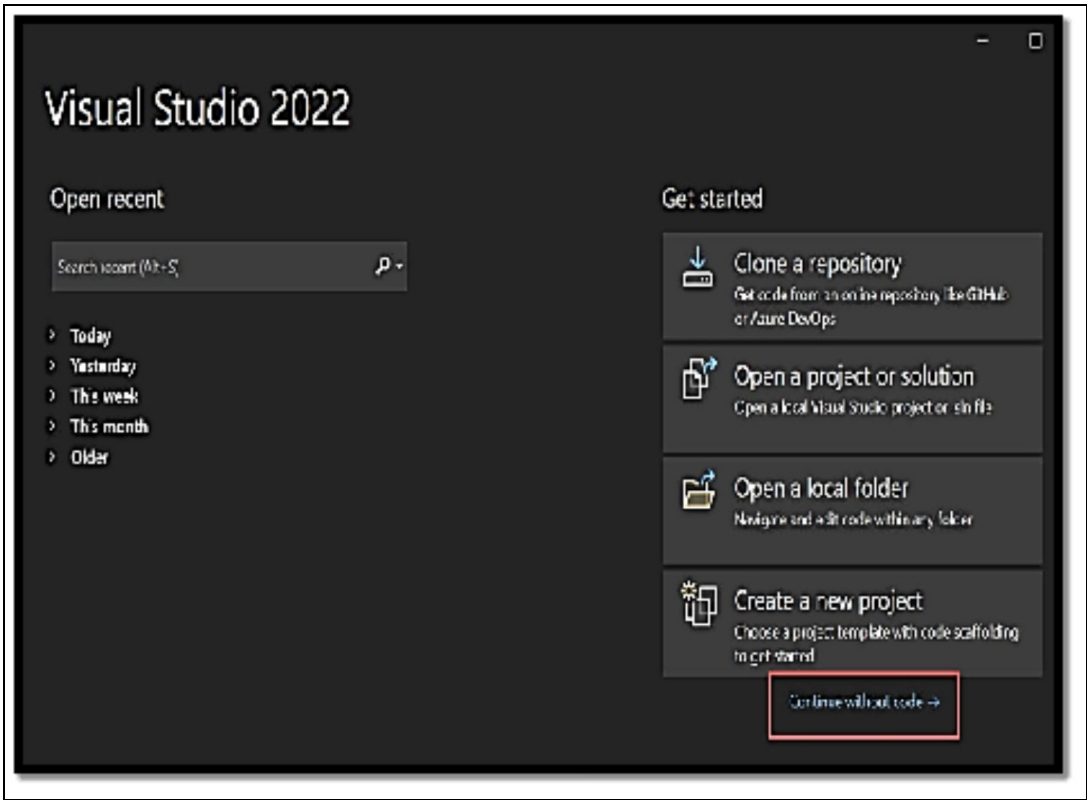

**The color themes are:**

- Dark,
- Light
- Blue,
- Blue with extra contrast
- Use system setting.

You can then preview the pre-established colors for each of these selections using the same window.

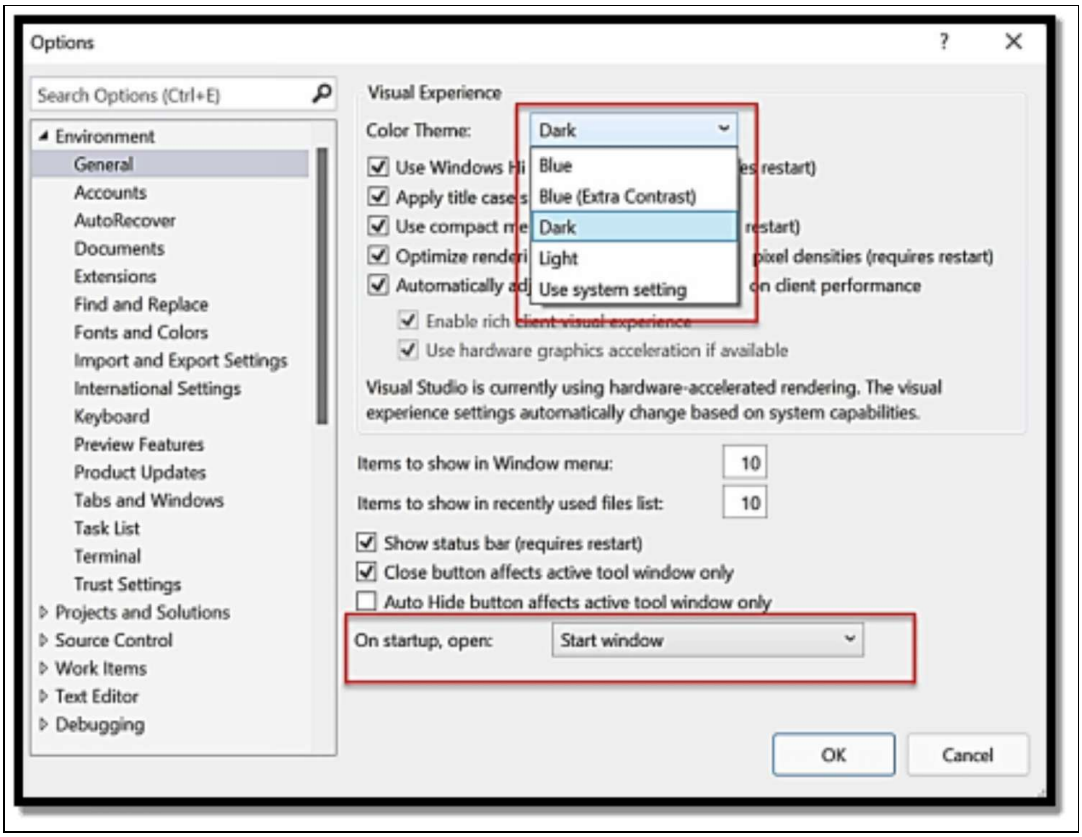
The theme of your choice will be applied and saved in the configuration connected to the Microsoft account you used to seek the license for the usage of the IDE as soon as you make your selection.

After choosing a theme at the beginning if you want to change it, go to Visual Studio and select **continue without code.**




Then go to the **tools** drop-down and enter **themes.** There you are going to see the default themes and the ones that you have installed. You then can select one.

You can also change the theme by going to **tools** then choosing **options** then **environment** then, select **general** and select the **color theme.**




Also from this menu, you can change the way you want the visual studio to start by choosing another option from the section **on startup>open.**

**These are the options that you can choose from:**

- **Start window—**this is the default window
- **The most recent solution—**means that when you open a visual studio it starts with the last opened project
- **Empty environment—**this opens the main visual studio window with no project or solution open.

# Visual studio market place

If you are not okay with the default themes then you can also go to the visual studio market place to find the community-made themes that are hosted in the visual studio market place. Once you get to the market place, select the filter to filter out just the themes created by other developers.

Clicking on the element will provide information on the theme and, nearly often, a picture of how it will seem in Visual Studio if it is applied.

If you wish to install the theme, simply click the **Download** button. This will download the relevant installer, which you can then run to begin the theme installation process.

To switch to the new theme after the theme has been installed, simply repeat the previous section's instructions.

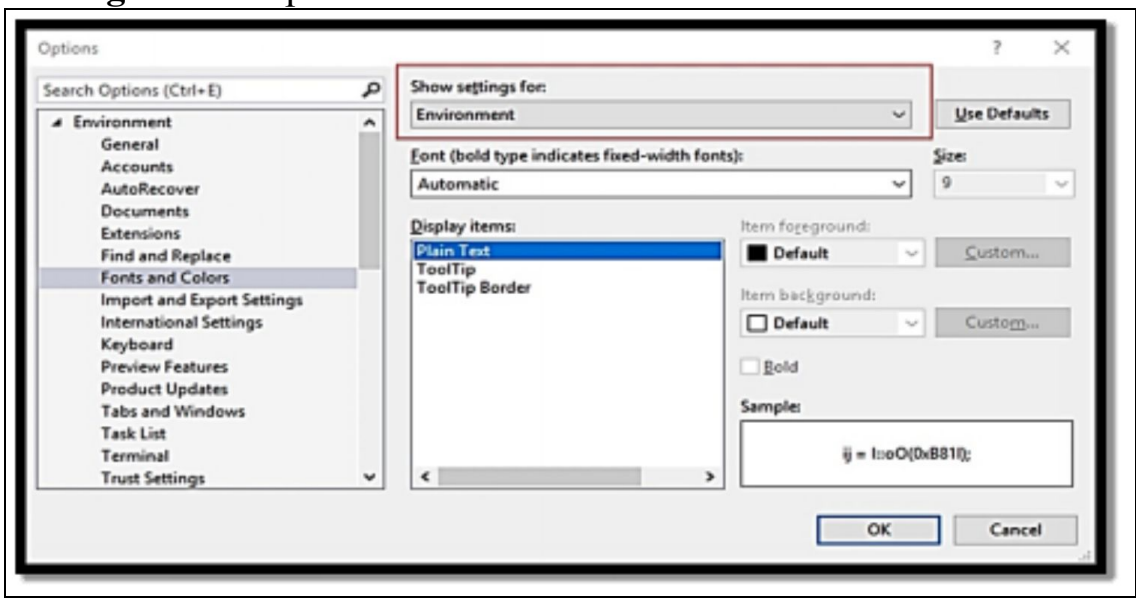
# Customizing fonts

You might want to change the font type in two places: the general environment and the source code editor.

## Changing fonts in the IDE

**Here are the ways that you can change the font inside the IDE.**

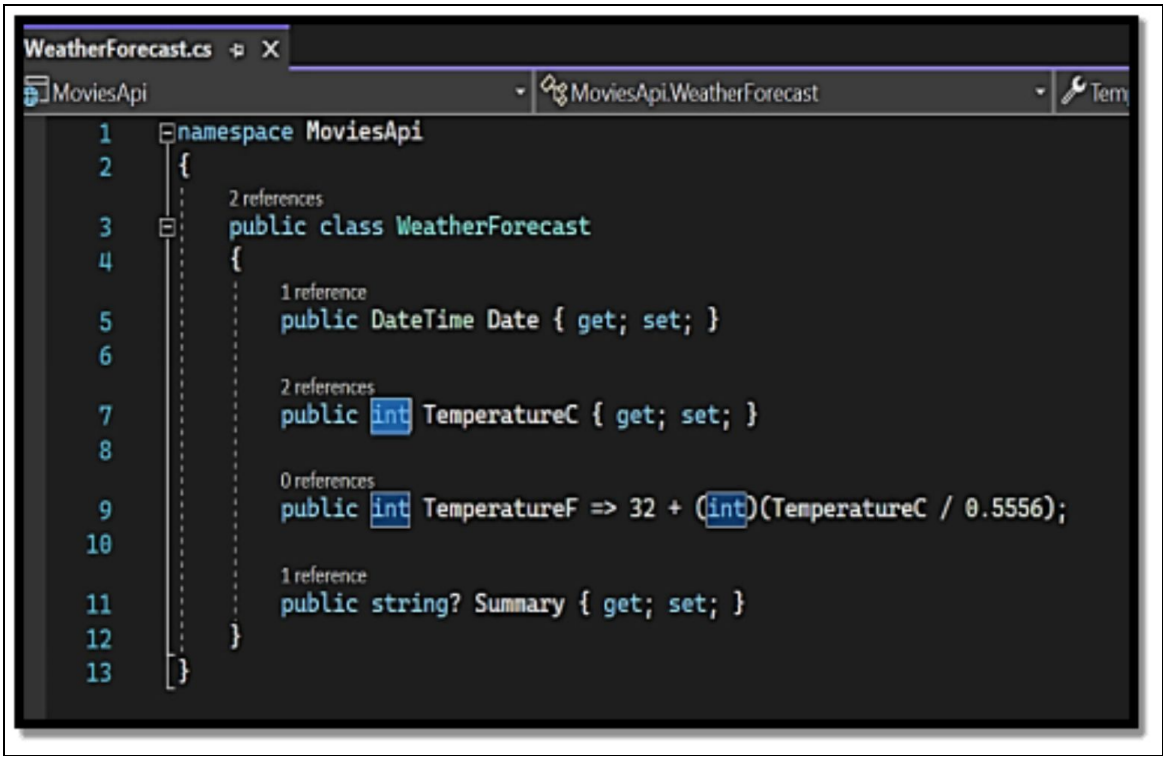
- Enter **tools** then **options.**
- Then inside the configuration window, enter **the environment** then **fonts and colors.**
- when you are in this menu choose **environments** from the "**show settings for"** drop down.




# Reference highlighting

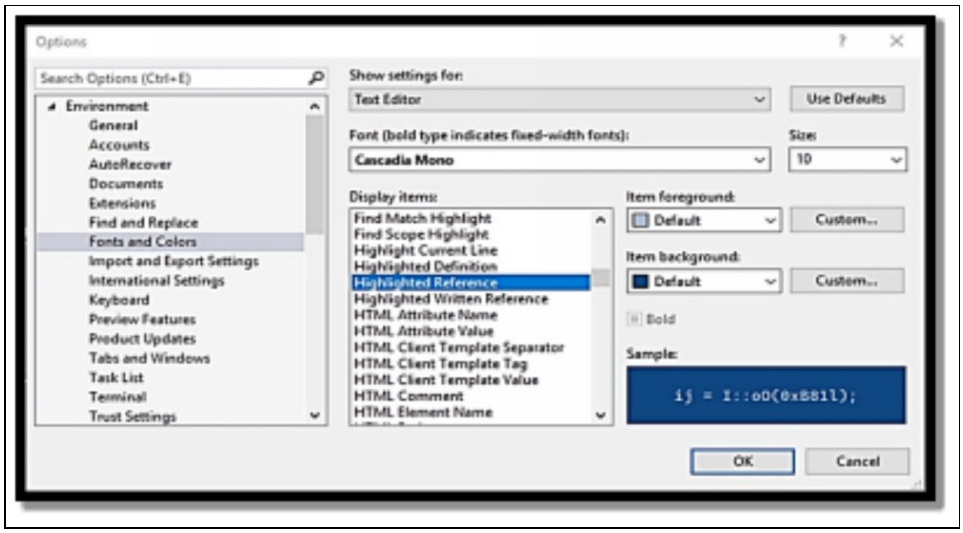
The highlighting of references may be changed, which is another frequent adjustment. This is the process of highlighting each instance of a chosen element, such as a variable or term.

The image below highlights all references to the “int keyword when placed above it best illustrates this:




**Here are the steps to change the color of the reference that you find.**

- enter **tools** and select the **options** menu.
- Go to the **fonts and color** section.
- Choose the **text editor** configuration.
- select the **display items** menu and enter the configuration named “**highlighted reference”** then you can change the color that corresponds.




The text color can be changed in the "**item foreground**" while the background color can be changed in the "**item background"** Then you can preview the color you choose with the **sample** menu.

# Changing fonts in the code editor

To then change the fonts inside the code editor, you need to follow the steps earlier mentioned.

You may alter the background color, font size, font color, and other features as part of the setup choices.

The ability to customize the configuration to your exact needs is one of the benefits of using this configuration option. This implies that in addition to many other options, you may change the typography for items like line numbers, bookmarks, selected text, and code snippets.

Let's look at how to modify the panels inside the IDE so that your workflow will be quicker following your projects now that you know how to change the font to suit your demands.

# Customizing the menu bar and the toolbars

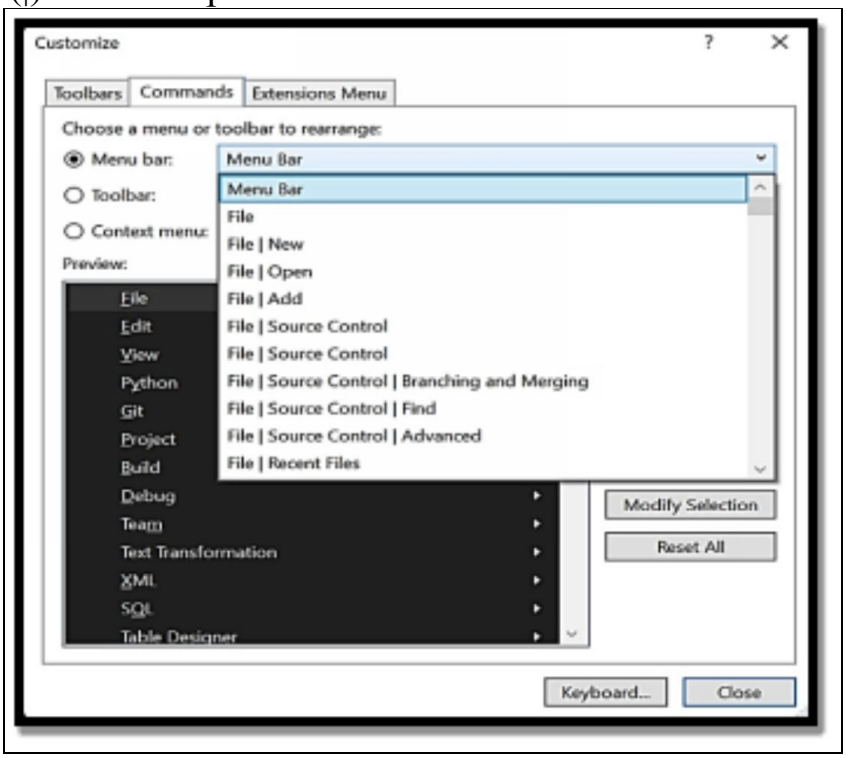
The menu bars and the tool bars are the best way to get tools or frequently used commands. It Is then important to learn how to customize these things so that you can develop the application. So, in this section, we are going to be going through the ways to customize the menu bar and the toolbar.

## Customizing the menu bar

The menu bar is the group of choices that appears at the top of the IDE and enables you to access a drop-down menu of choices (like the File menu) to carry out a particular activity, show tools, or make changes to a project:

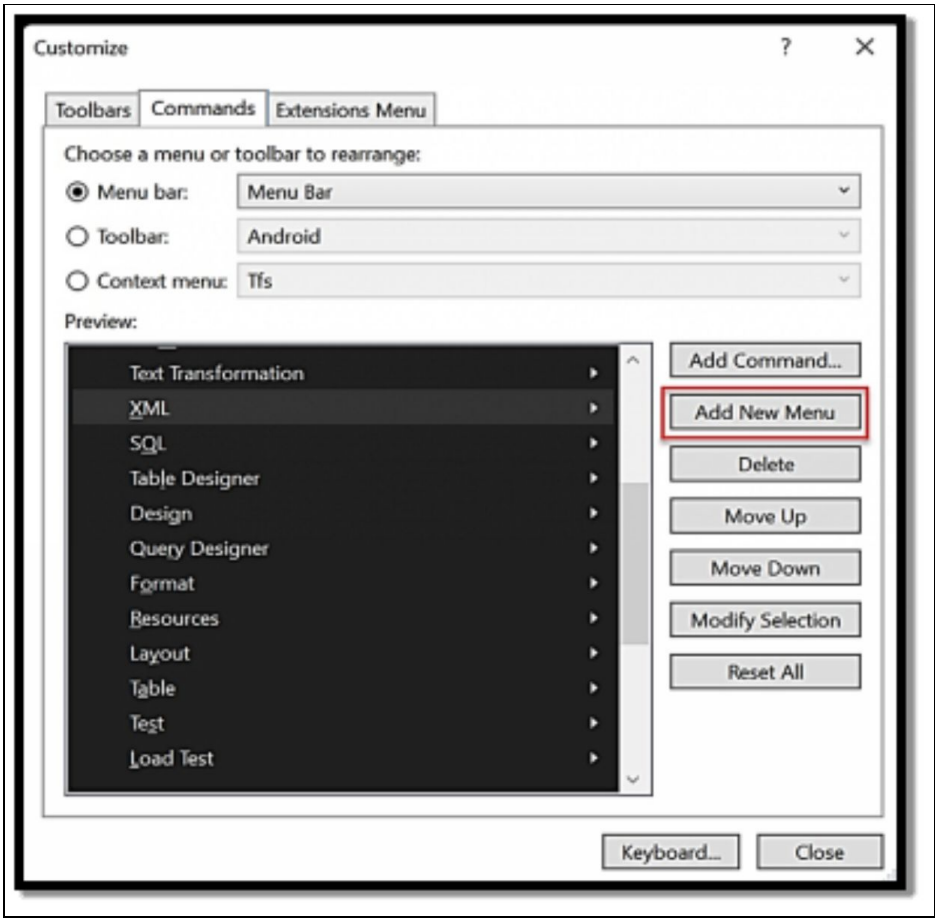
**The actions listed below must be followed if you wish to alter the tools that are a part of the default setup, either to add choices to a particular menu or to make your menus:**

- Enter **tools** and choose **customize**."
- Check out the Commands area.
- You should use the Menu bar option in this part to change the main menu bar and the secondary menu bar, which are distinguished by pipe symbols (|) in the dropdown.




- After choosing a menu that you want to configure you are going to see a preview to see how the menu is. Then you are going to see a new command inside the menu bar by selecting **add command.** this opens a new window that shows you the command grouped by category, that you can choose and add inside the chosen menu.

- To then remove the option that is added to the menu, select it then **delete it.**
- You can also **move up** and **move down** with the buttons on the menu.
- Then finally, you can also create submenus and menus. To do this, select **add new menu.** This adds a new menu based on the level of hierarchy you are in.

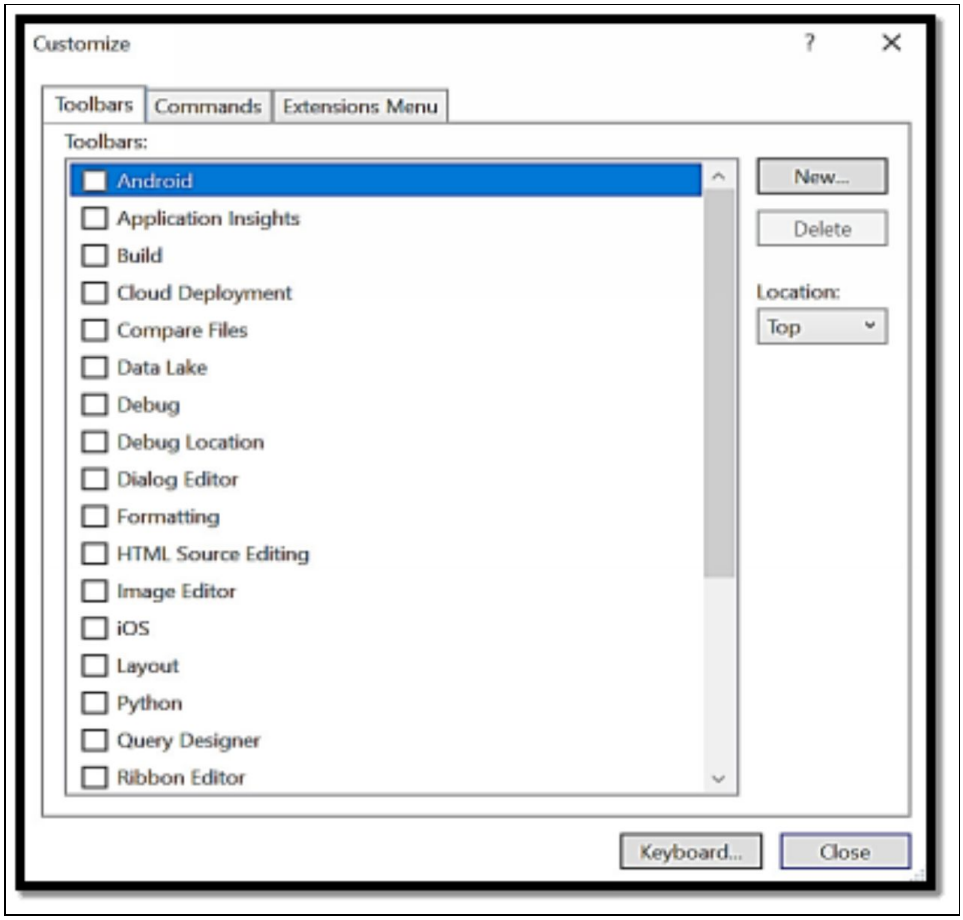



- Alternatively, to add a submenu in the **edit** menu you need to carry out the previous step when selecting the **edit** menu.

# Customizing the tool bar

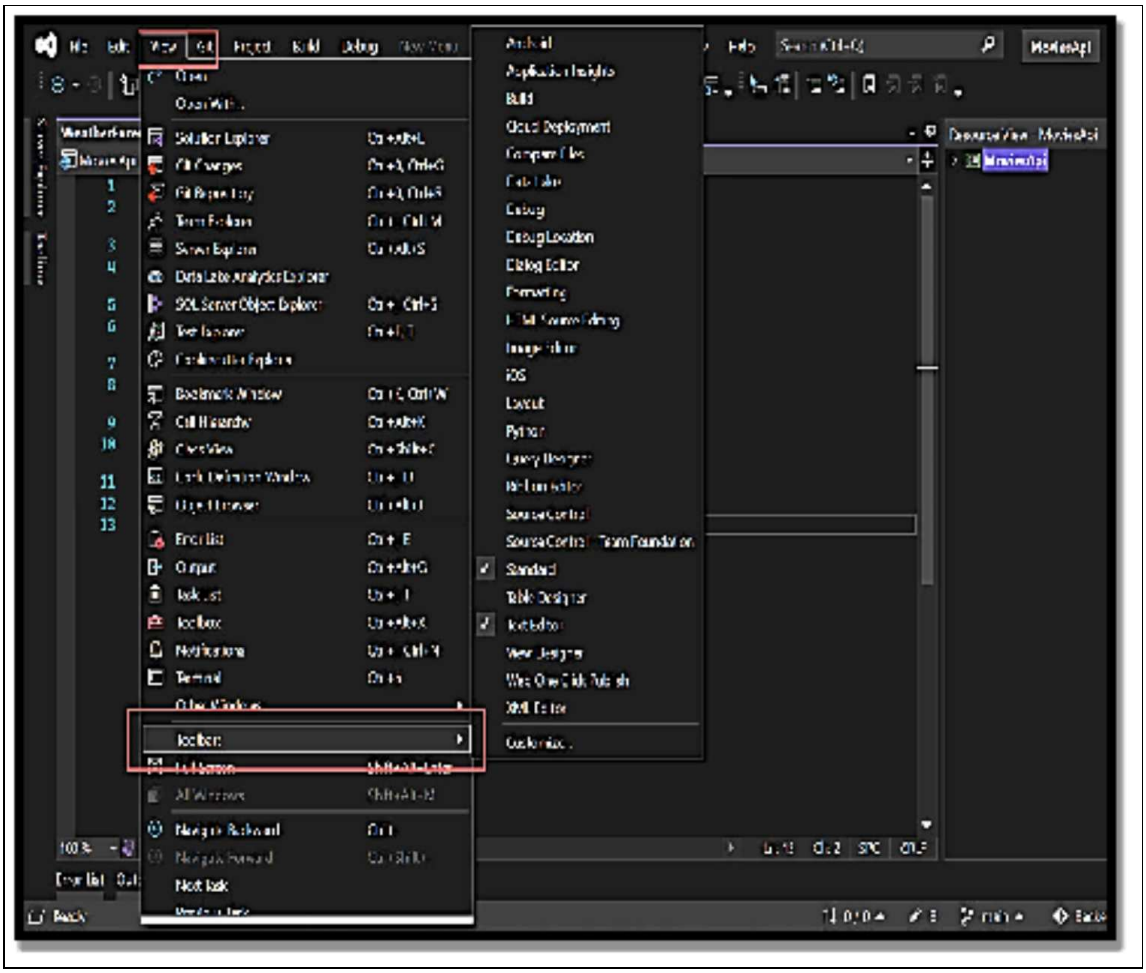
The toolbar is the set of commands that you can get to directly without having to first open the menu. By choosing **Tools** then **Customize**, you can also choose which commands will be shown in which tool groups. By default, the Toolbars tab in this window will display the many categories we might choose to display in the user interface. For API and web applications, the Standard option is by default chosen. Other toolbars may be included by

default depending on the projects, but we can manually add other toolbars by simply choosing them with a tick.




By selecting the Commands tab, you may add additional commands to a toolbar. We will choose the toolbar we want to change in this tab and follow the same instructions as in the section on customizing the menu bar.

The image below demonstrates how to enter the View menu and Toolbars, where you may pick and deselect the toolbars you are interested in, to rapidly add various toolbars to the IDE.



# Customizing panels

You may access particular tools in Visual Studio panels based on the kind of project you are working on.

These panels are made up of document editors and tools, some of which are frequently used, such as the solution explorer (to view the structure of your projects), the toolbox (which displays controls to drag and drop according to the current project), the properties panel (to change the properties of the selected element), and the code editor.

In this section, we are going to be taking you through crucial topics about customizing panels.

# Adding tools to panels

There are a few tools or windows that you can both show or hide and you can find them on the **view** menu.

The tools that are often used the most are presented right away when you open this menu. These tools include, among other potent ones, the server explorer, the class viewer, the output window, the terminal, and the error listing.

There is another group of instruments that are less popular but could be useful to you at some time. These may be found under “Other Windows.”

You may access tools from this page, including the Package Manager Console, Data Sources, C# interactive, and containers, among others.

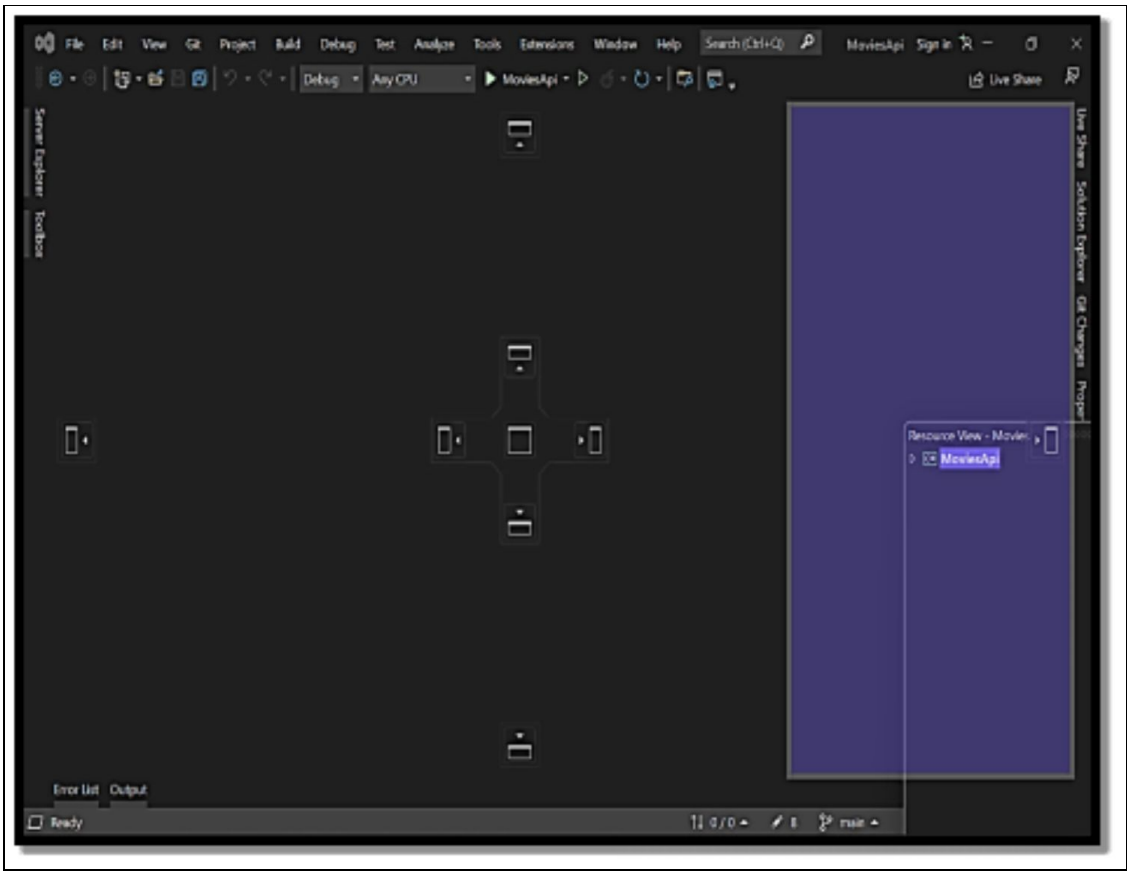
Simply choose one of these tools to add it to one of the panels, and it will be integrated into your present environment in a strategic panel. For instance, the Server Explorer tool will be added to the left panel if you add it. The Output tool, on the other hand, will be added to the bottom pane if you add it.

# Panel accommodation

The ability to organize the tool panels in any way you choose is one of the many benefits of Visual Studio. It is practical to be familiar with a panel's construction to get the best outcomes.

There are five areas on each panel where you may put tools. These portions, along with one in the middle, are situated on either side of the panel.

Simply locate the cursor at the top of the tool and move it to the desired panel to insert it into a panel. The IDE will automatically suggest potential locations for the tool, making the process quick and straightforward. You can even remove the panel from the main windows and utilize it independently.




# Working with document

There are a few options that you can add if you are working with a document editor like the code editor. To use this option, right-click on the tab of the document.

These choices are quite simple to use; for instance, selecting the Float option will make the editor into a floating window that we can move to a second monitor. With the Pin Tab option, we can position the tab at the top of all currently active windows, and the Set Tab Layout option lets us reposition the tab group to the left, top, or right.

It is crucial to emphasize that we will have more alternatives accessible if we have more documents open. With this choice, we may organize documents into groups to spread out the available space and use it for activities that will boost productivity, like comparing two documents.

# Managing window layouts

As a developer, you may encounter a variety of projects. In case you need to manage database-related tasks for a project, you should generally have the Server Explorer tool open.

You might need to interact with databases on another project you're working on concurrently, therefore you want to keep your toolbox open.

There are a few tasks that are related to the windows layout. Here are they

**You can save the layout.**

You must select the **Window** then **Save Window Layout** option after configuring your panels with the tools needed for a specific project.

A new window will open and ask you to give your workspace setup a name. As soon as you input a name, the setting is instantly stored.

Go to the **Window** and choose "**Manage Window Layouts"** option. this will display a window containing all of your previously stored workspaces or layouts, allowing you to confirm that the change has been made.

**To apply a window layout,**

Once you've stored at least one layout, you may use it to load the panels with the tools in your workspace in the same order as when you last saved the window layout.

Go to the **Window** then select **Apply Window Layout** option, which displays all of the previously stored layouts, then you need to apply a layout. Your workspace will load when you choose the one you wish to use.

**Here is how to reset the window layout**

Finally, you might wish to undo any modifications to a workspace so that you can restore Visual Studio to its default settings. Fortunately, you can access this option by selecting **Window** > **Reset Window Layout** from the menu.

When we click this button, a prompt will appear asking whether we consent to go back to the default setup. Simply selecting "**Yes"** will enable the environment with its default settings.

## Review question

- How do you customize the toolbar?
- What are the tools accommodated in the panel?
- How do you customize the menu bar?

## Summary

In this chapter, we went through the basics of customizing a visual studio, and how to choose themes, fonts, colors, and panels.

# CHAPTER 3
# IMPROVEMENTS IN VISUAL STUDIO 2022

A new update means that there are new features, new experiences, and functions. In this chapter, we are going to be taking you through some of these new features and function

## Technical Requirements

### 64-bit architecture

The new architecture in 64-bit is a straightforward innovation in Visual Studio 2022. it uses a 64-bit CPU internally to enhance speed and decrease execution times for several operations.

Only 64-bit systems are supported by Visual Studio 2022; these systems are typical of modern laptops and PCs.

In the past, there used to be a 4 GB memory access restriction with 32-bit architecture. Now that there is a 64-bit architecture, it is feasible to access more memory, which decreases time constraints and prevents IDE freezes or crashes.

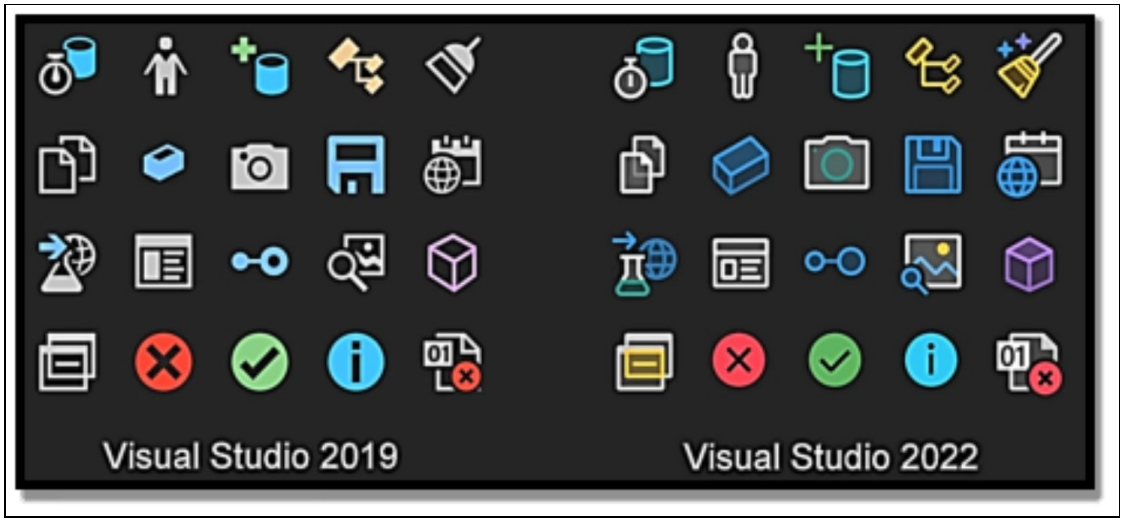
Overall, the performance for several cases was enhanced in version 2022 by the Visual Studio development team. When working on huge projects, you'll notice the difference, and future editions' performance will be enhanced.

Although 64-bit is a good speed gain, it doesn't enhance user involvement when coding.

## New Icons and styles

There are now new icons and styles in the Visual Studio. While this feature might look simply, it actually can help you to navigate through the applications with the visual elements and also identify the tool actions and tools.  Take the image below, for example, the broom icon has a different contrast from the yellow icon and a more modern-looking feel. This means

that you can use code cleanup. The broom icon can be used to execute code cleanup in other to fix the code format.



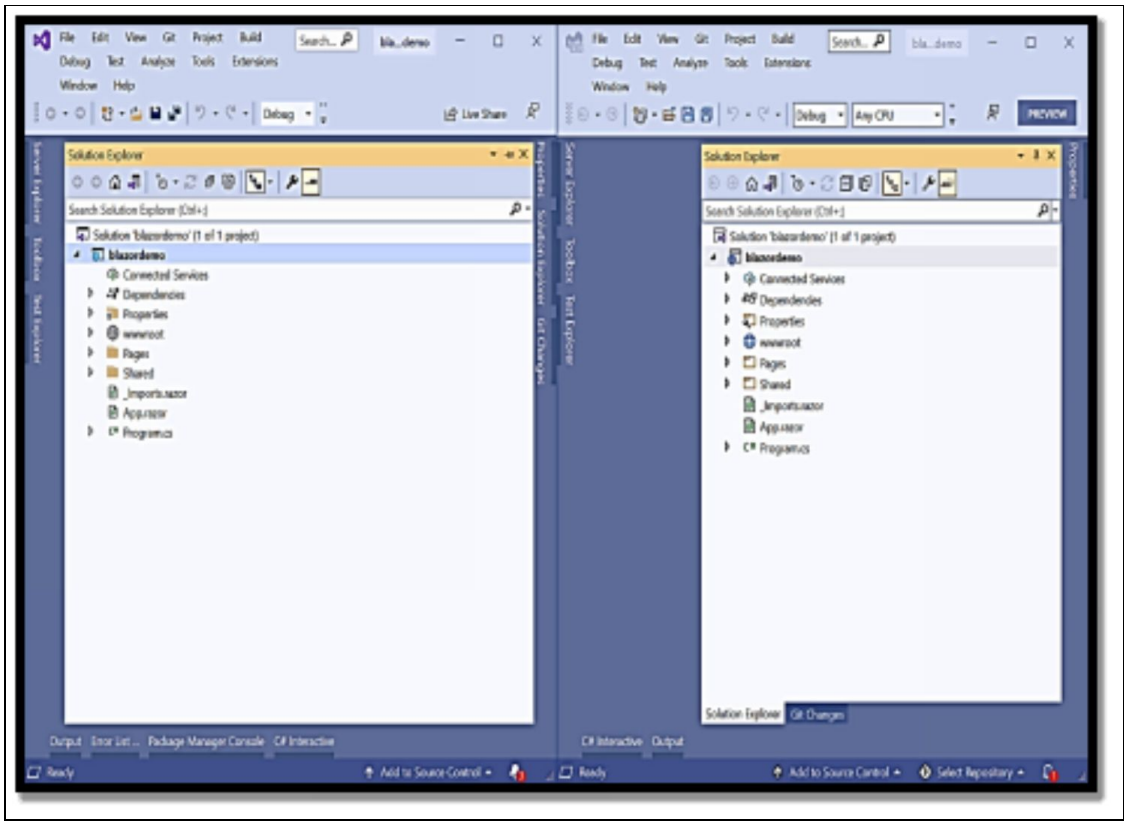

The new contrast between the letters, the icons, and the background means that working in the visual studio is now more pleasant and not as tiring.

The Visual Studio development team collaborated with the developer community to meet three goals for the new icons in Visual Studio 2022: uniformity, readability, and familiarity. As a consequence, a collection of icons with the same meaning and recognized silhouettes were created. They also had similar colors and high contrast.

Dark themes are increasingly being used in apps throughout the industry, in addition to being more popular with developers in general.

Another instance of applying the blue motif is in the image below.

**Even if the colors are relatively similar, there are minor differences and the new icons make the screen appear better:**



The other motifs have also had minor adjustments, although blue and dark themes were the most enhanced themes.

The use of Visual Studio is made easier with these new icons and styles. We'll examine Visual Studio's support for.NET 6 in the part after that.

# .NET 6 support

With the help of C#, F#, and Visual Basic (C# being the most popular),” .NET” is a free, open-source framework used to build online, desktop, mobile, and other types of applications.

Every year, a new version of .NET is released. A new long-term support version of .NET, .NET 6, delivers significant advancements including minimum APIs and C# 10 compatibilities.

The latest version of Visual Studio 2022 is prepared for project creation, compilation, and publication.

# Hot reload

Hot Reload was the most often requested feature in the .NET developer community for a very long time. By refreshing an application after each modification to the code, a developer's productivity is increased, which is already present in many technologies.

Hot Reload recompiles code once changes are made to it. This greatly boosts productivity since an application may display visual changes instantly without having to restart.

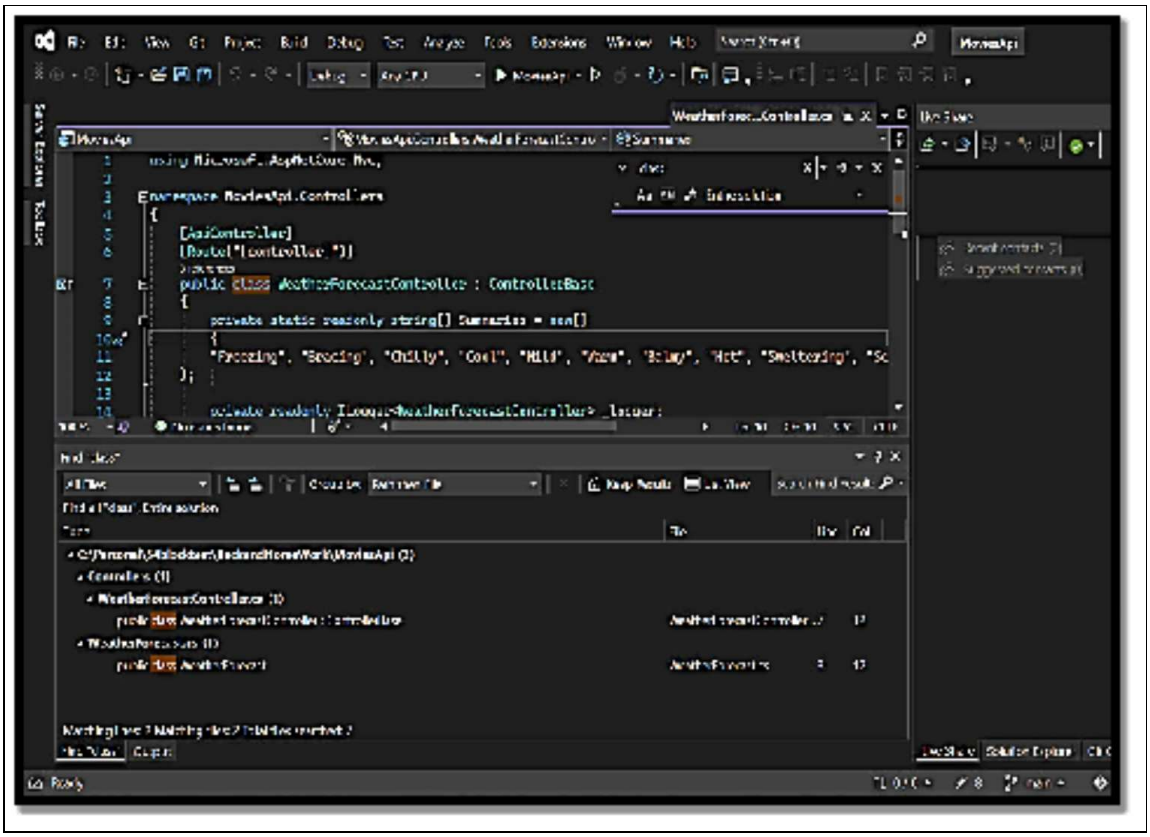
Even outside businesses are developing this function; the most well-known is LiveSharp, which charges a monthly membership.

For a variety of applications, including those combining ASP.NET and Blazor (by building web apps utilizing WebAssembly and Razor components), Visual Studio 2022 has this fantastic feature:

# Other Improvements

Performance is only one of the enhancements in Visual Studio 2022. In addition to 64-bit support, tools like the search file tool were enhanced in Visual Studio 2022 to make it easier to locate specific components in projects with lots of files.

**You can see a search example in the diagram below:**



You also have a brand-new breakpoint type (a breakpoint is a technique to stop application execution in a specific line of code). We may add a breakpoint that is based on a breakpoint that was made earlier.

We now have a new experimental capability for Git integration that enables us to manage several repositories within the same program. As a result, you may make changes to many projects without starting additional instances of Visual Studio, then commit the changes (store them in Git).

Last but not least, IntelliCode is unquestionably one of the most significant updates added to Visual Studio 2022. It enables developers to work with confidence, identify issues quickly, and conduct targeted code reviews by recommending actions that are most likely to be taken first on a to-do list.

# Review Question

- What are the new features of the visual studio?
- What is Hot reload?
- How do you access the new icons and styles

# Summary

Now that we have taken you through the main upgrades on the visual studio, we are then going to be going through some of these features.

# CHAPTER 4
# CREATING PROJECT AND TEMPLATES

Templates in Visual Studio are a set of files, references, project settings, and compilation choices for using a certain technology. Templates give us the foundational code we need to work with and serve as a model that we can follow and finish by adding our business needs and logic. Although certain templates, such as class libraries, are installed by default, other ones are installed according to the workloads that were chosen during the installation of Visual Studio. We will have a variety of template alternatives to pick from depending on the project or technology we wish to employ. The greatest course of action you can do to prevent technical debt and future troubles in your architecture is to choose the appropriate template that suits your objectives.

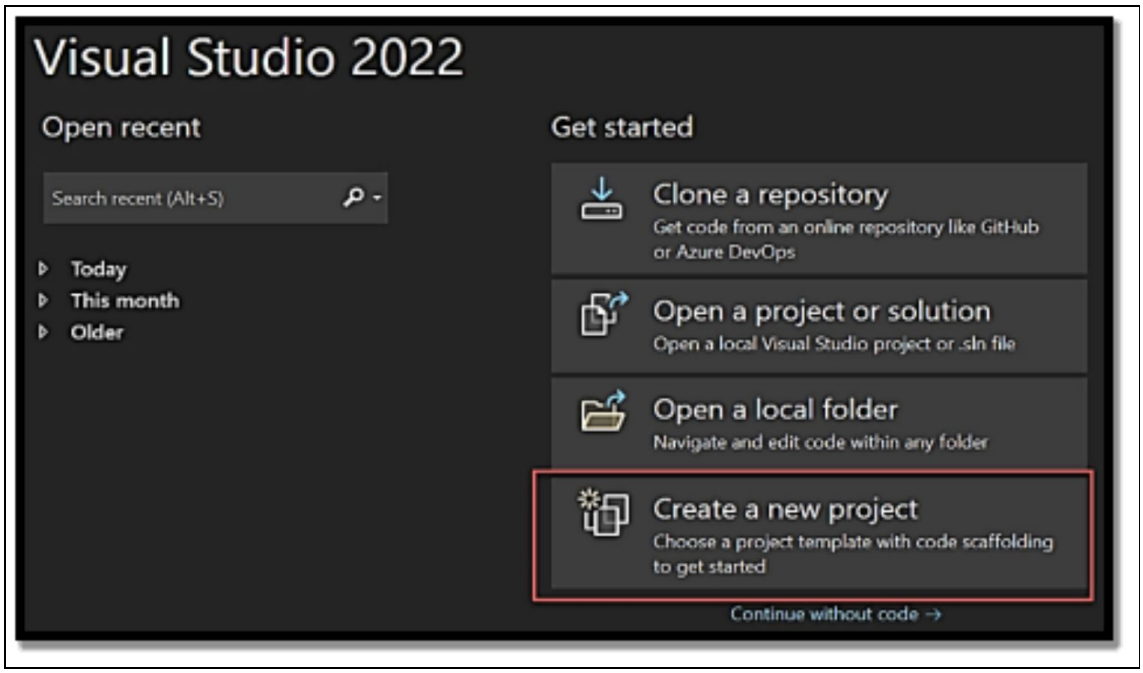
We are then going to be going through the templates that have been provided by Visual studio. Then we are going to be going through the ways to pick the best template for the project that you want to take.

## Technical requirements

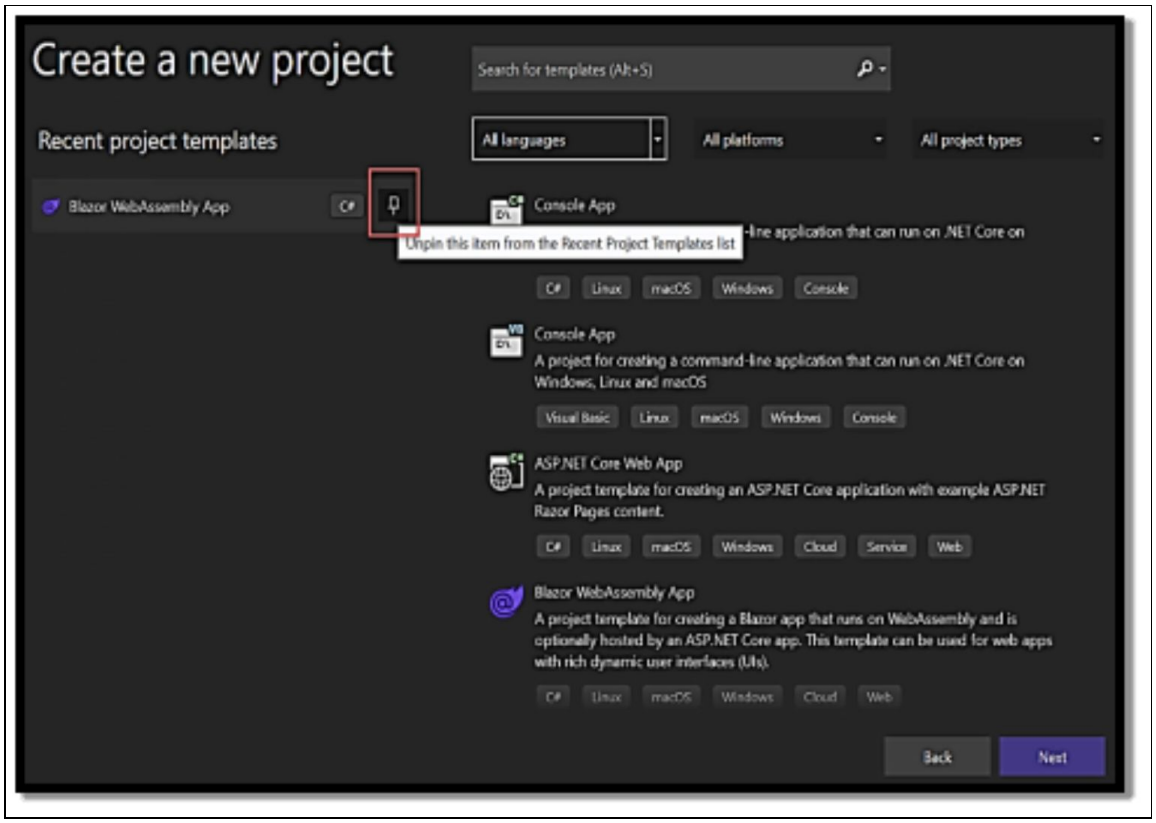
### Selecting and searching for templates

Depending on the kind of project you are working on, Visual Studio provides a variety of templates that we may use with .**NET** and other technologies.

Simply launch Visual Studio and use the option to create a new project to explore the templates in Visual Studio 2022.




Once you choose the option, you are going to see the templates available to create your projects. Also on the left, you are going to see the recently opened templates and you can pin and unpin the template so that you can have access to them quickly.

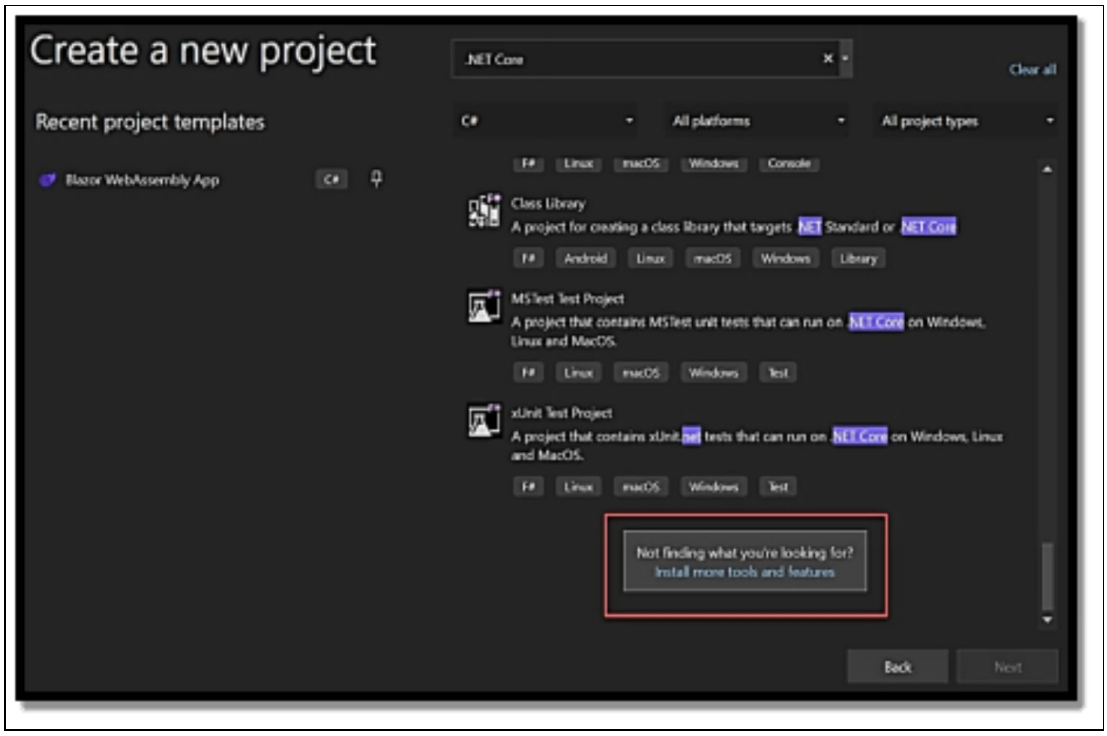



In this image, Additionally, a search box for sorting the various templates is shown at the top. You may start a template search by typing the phrase associated with the name or the technology you'll be using. All of the templates that include the phrase you provided will be shown by the search engine.

Three drop-down buttons that act as filters for searching for templates specific to the programming language, platforms, or project types we wish to develop are visible below the search bar. The following are some of the most significant choices for each of these filters: **all languages, all platforms, and all project types**

you can also combine the filter and the search bar. you can choose the type of .NET core and choose the C# option in the drop-down language and you are going to find all of the projects that are related to the .NET core and the c#.

at the end of the results, you can then install some other templates if you do not see the option that you are looking for.




It is crucial to remember that there is a description in each template that provides information about it, making it simple to determine whether it fits the structure and schema we want.

Last but not least, if the template description includes the phrase Empty, it indicates that the template doesn't come with any default components or modules but instead offers examples or demos for the developer. Only the project and the essential parts are included in this form of a template, which may be assembled and operated.

# Templates for .NET core

**There is a difference between the .NET core and the ASP.NET core. Here they are:**

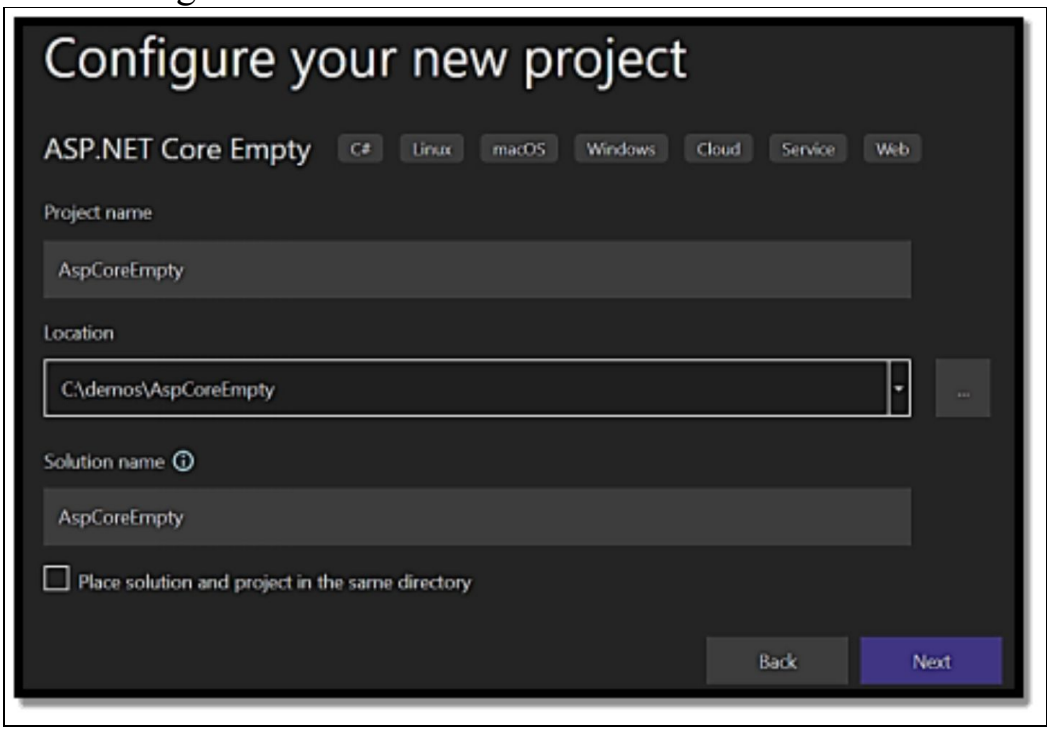
- The .NET core is a cross-platform, free, and open-source framework (now called .NET) that is used to develop online, desktop, mobile, and other types of applications utilizing a standard library and the programming languages C#, Visual Basic, or F#.
- On the other side, ASP.NET Core is a multi-platform web technology for building contemporary apps. NET. For this reason, even though

the.NET Core framework has been superseded by .NET 6, .NET Core-related terminology can still be found in the templates.

We may search for and choose the ASP.NET Core Empty template if we wish to start a fresh project of the ASP.NET Core type. The Hello World message will be displayed when this template's basic API project is executed. We must input a project name, such as AspCoreEmpty, to configure the new project.

Additionally, we need to specify where we want to store the project. You have the option of leaving the default path selected or selecting a different one, such as C:\demos\AspCoreEmpty.

Set the solution's name to finish. It will be the same as the project by default, but you may change it to something else. We have used the same name in the image below.




**Choose the next button and you are going to see the new window.**

- Framework: This dropdown will provide the range of frameworks that are compatible with the technology you have chosen. Although it is possible to use an earlier framework, it is usually recommended to choose the most recent version of the framework available when starting a new project.

- Set up for HTTPS: By selecting this checkbox, you may set up the project to utilize a self-signed SSL certificate. When you launch the program, you will be prompted to trust the certificate if you are working on such a project for the first time to ensure that everything functions properly. Although the HTTP protocol can be used, it is almost always advised to utilize HTTPS in practical situations.
- **Enable Docker:** Using this option, you may make your project compatible with Docker.

Then when you create the project, you can then analyze the structure.

You will see all the related files to the template in the **solution explorer.** There, you will see the files that are related to the template that was created for the project.

**There are just two files that deal with the project's structure, and they are as follows:**

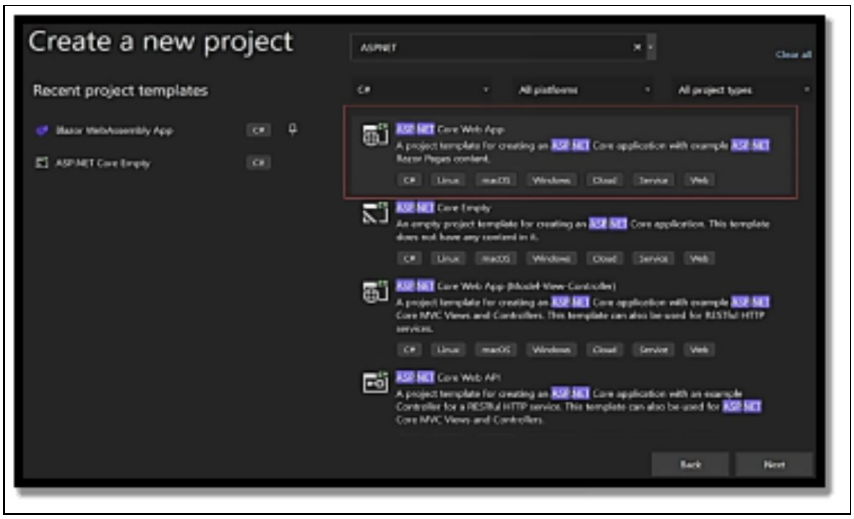
An ASP.NET Core project's default settings are stored in a JSON file called appsettings.json.

**Program.cs:** This is the starting point of an application that contains a sequence of commands to configure and launch the application.

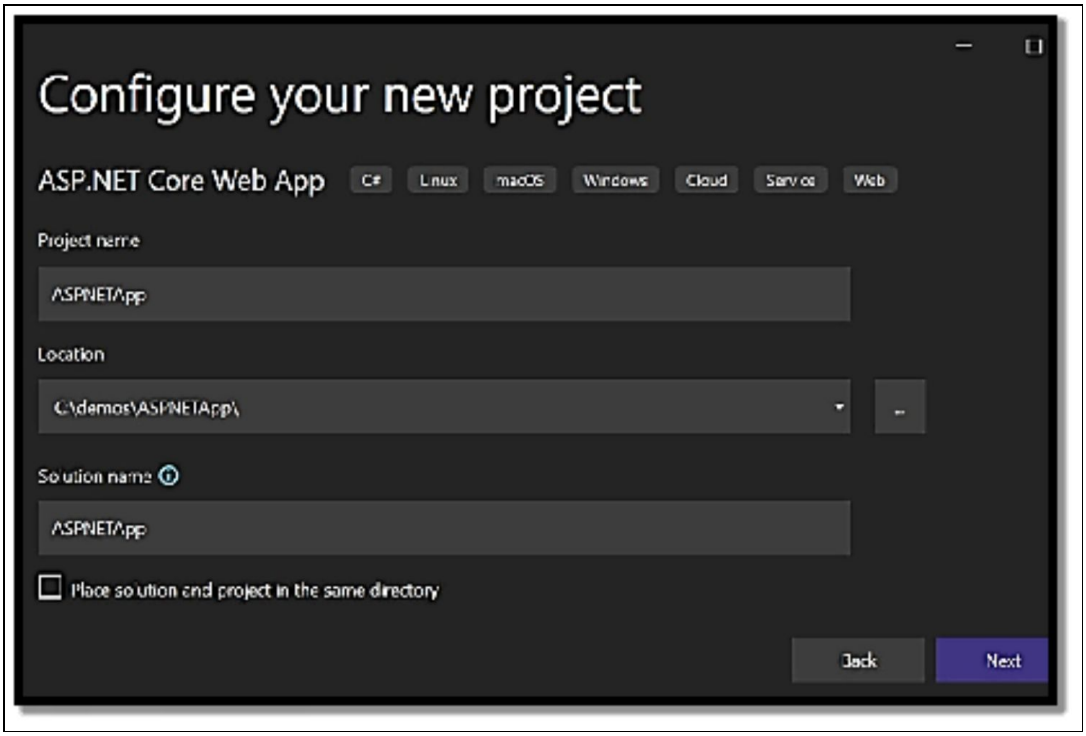
The template we choose makes use of a new idea from.NET 6 called Minimal APIs, which is a technique to leverage helper functions in Program cs to construct a basic endpoint with the least amount of code possible.

When an application is run, it will by default display a Hello world message that is returned in a GET method and mapped into the base URL, such as http://myapiurl/.

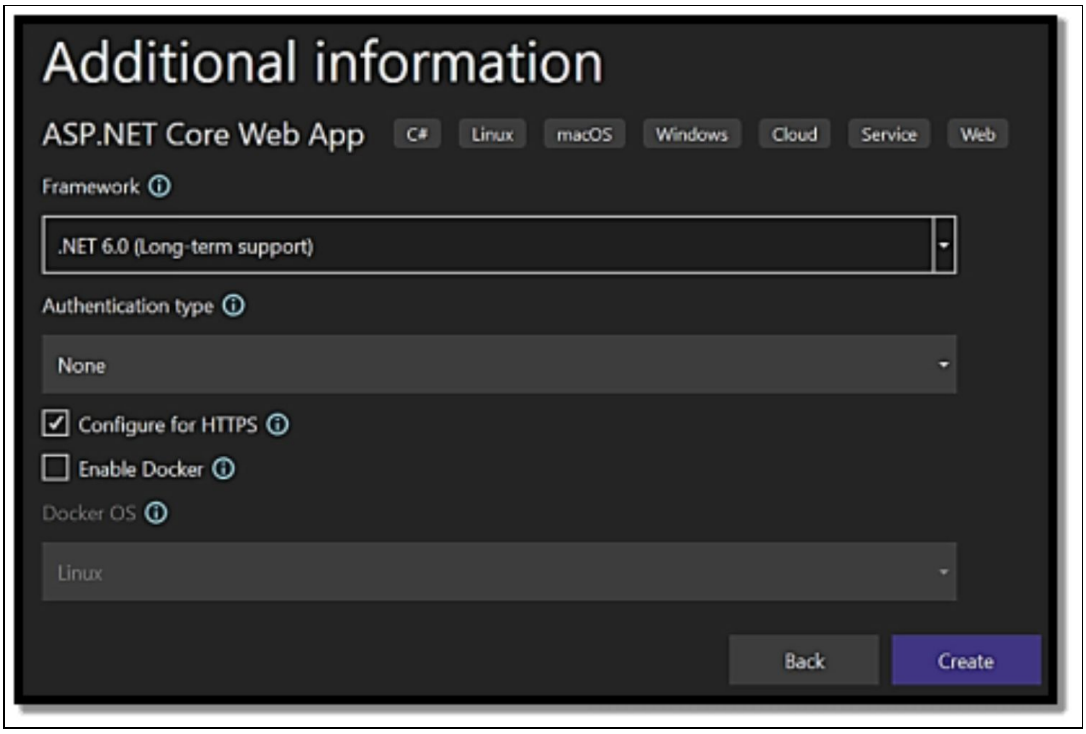
So here is how to create a project using the ASP.NET core Web app template so that you can analyze the structure. You need to open the visual studio like in the image below and choose "**create a new project."** Then you need to look for the item ASPNET and then choose **ASP.NET core web app.**




Select **next** then fill the information out that is required for the subsequent project.




Once you fill in the additional information for the project, you can then choose **create** so that you can generate the project using the ASP.NET core web app template.




If you want to build a web application that runs on the server and uses .NET and C#, this template is ideal. Additionally, it leverages razor pages (a syntax that combines C# with HTML in files with the cshtml extension) to break up a web application into manageable chunks.

Because they operate quicker and have better search engine optimization (SEO), modern applications typically run in a browser, which increases traffic to your website. However, dashboards, internal projects, administrative panels, and many other sorts of online applications continue to benefit greatly from server-side applications.

Static files like CSS, pictures, and JavaScript are related to the **wwwroot** folder. Additionally, all the UI components are located in a subdirectory named Pages.

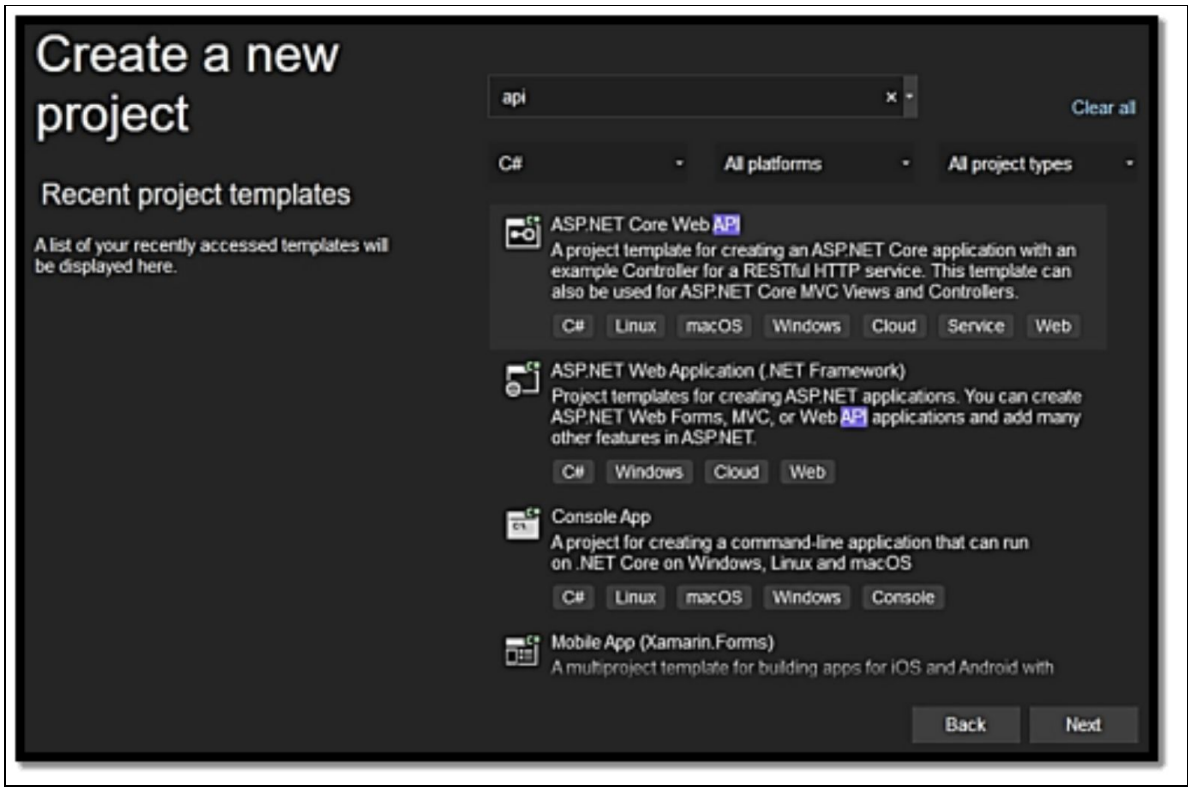
As an example, the **index.cshtml** file contains both HTML and C# code. You may utilize the C# code in the file thanks to the @ symbol. To map the values on the page, @model, for instance, sets the model type.

# Templates for APIs

It is now very common to use API-based endpoints in apps. This is because they can be scaled to meet demand and offer a high degree of security and interoperability without relying on a specific technology or operating

system. Additionally, we may use APIs to construct sophisticated architectures like microservices, in which all of our business logic is dispersed among several tiny, independent services.

Given how crucial it is to be able to construct API-based solutions, Visual Studio 2022 includes a specific template that enables us to build.NET 6-based APIs that we can search for using the API keyword filter.




After choosing the template, you are going to need to fill in some other information according to the needs in the image below.

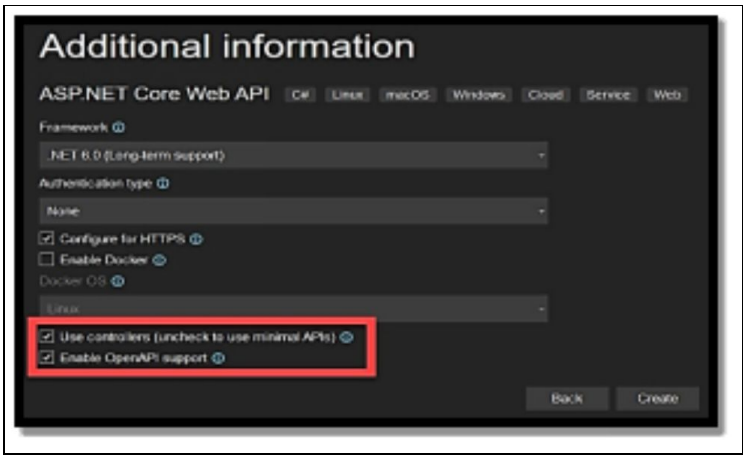
We will be prompted to choose the framework version and if we need any kind of authentication in the next box, along with other information that we have previously seen in the Templates for .NET Core section. **However, we have a few more choices that are not included in a typical ASP.NET Core project, and they are as follows:**

**User controllers**: If this choice is made, the setup will permit the usage of controller files. A feature known as minimal APIs will be utilized if it is

deselected, producing the very minimum of code required to have a usable API.

**Enable support for OpenAPI**: An open-source toolkit called Swagger, which is based on the OpenAPI specification, enables us to describe the APIs we develop in a clear, concise, and organized manner, giving API users good documentation.

**In this example, we have chosen the two options**

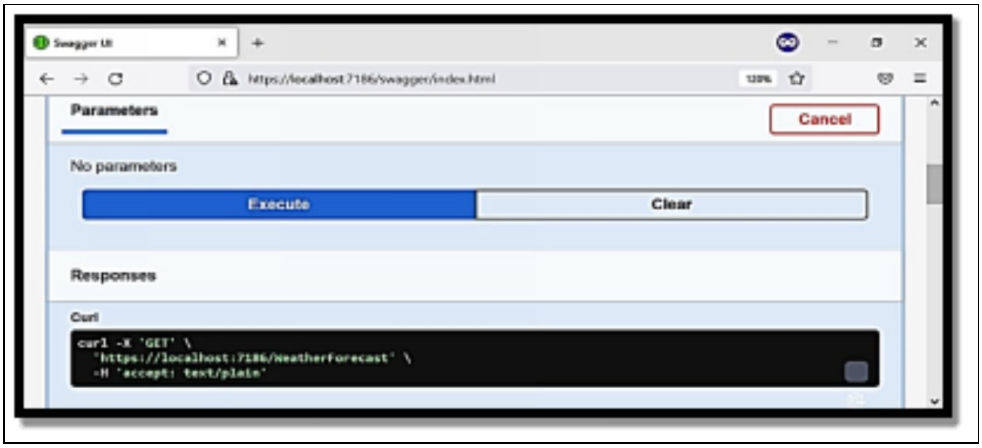



When you create the project, a folder called controller is also created.

Inside the controller folder, you are going to see the controller that is going to be created as a part of the API. Do not forget that you can see the controller inside this folder since we have chosen the **use controller** option

To then run it, select the green arrow or the play icon. You can start the project by debugging on or off. There are a few practical shortcuts for beginning a project. If you want to launch the project without debugging, press **Ctrl + F5** while holding down the F5 key on your computer.

**When you run the application, you are going to go to a window with the URL that ends with /swagger/index.html:**




All of the functionality that is available in our API and is described using Swagger can be found here, including all of the endpoints that we develop from the controllers in our project. A library called Swagger leverages the OpenAPI standard to display every endpoint, along with the arguments needed and the kinds of results that are returned.

We will be able to test each endpoint from this same page to validate them and perform any necessary troubleshooting when an endpoint does not function as planned. As you can see, this tool is quite beneficial for both developments and for giving consumers access to the developed API's documentation.

# Templates for .NET framework

The first .NET implementation was the .NET Framework, which Microsoft launched in 2002. Making this framework multi-platform was the long-term plan. Unfortunately, owing to API limitations, it always officially functioned on Windows-based devices. To address this issue, particular versions of the framework were developed to accommodate various systems, which fragmented the platform. The Microsoft team has eventually been successful in unifying the .NET platform thanks to the expertise gained from these initiatives that were introduced over time.

Numerous businesses worldwide utilize ASP.NET Web Forms, MVC, or Web API applications built on the.NET Framework, which served as the dominant framework for developing .NET applications for many years. This is the rationale behind the continued inclusion of .NET Framework

templates in Visual Studio. This does not imply that utilizing these templates to build apps is a good idea. You should always choose to employ contemporary technology with ongoing support.

Although the .NET Framework has several templates, we are more interested in one for web development. This is known as ASP.NET Web Application (.NET Framework), and you may find it by searching for "net framework,"

After choosing the template, several fields to fill out, including the project name and location among others, will appear in the following box. The crucial thing to remember is that we have a dropdown that displays a sizable number of .NET Framework versions that we may use.

The choice of these templates will be determined by the requirements of the application, but as was said at the beginning of this section, it is advised to use templates that are. NET-focused.

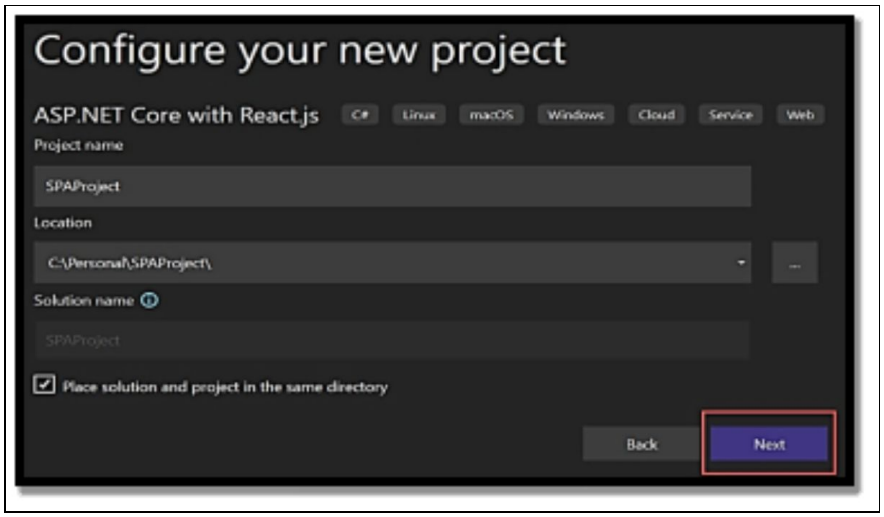
# Templates for SPAs

A single HyperText Markup Language (HTML) file is used to render every element in SPAs, a great design for online projects. Many libraries and frameworks, including Angular, React.js, and Blazor WebAssembly, make use of this idea.

Every component needed to build a monolithic application utilizing ASP.NET on the backend and a SPA library or framework on the frontend will be included in each project built using these templates.

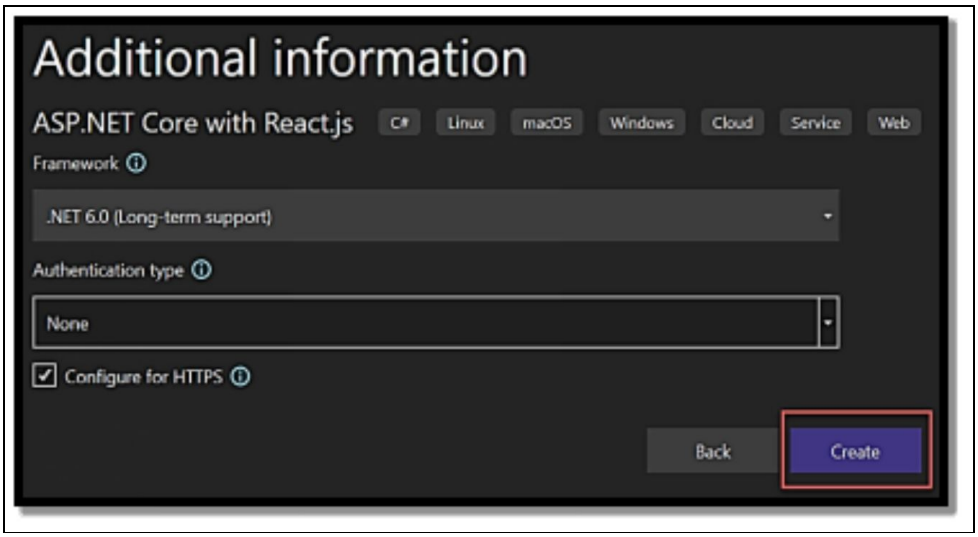
**Here is how to create a project with the SPA. But first, there are three SPA templates to work with.**

- ASP.NET Core with Angular
- ASP.NET Core with React.js
- ASP.NET Core with React.js and Redux




We are going to be using the ASP.NET Core with the react.js and choose **next** to then continue.

Then select .NET 6 as the framework and **none** as the authentication type and select **create.**




You are going to see the project that you created using the SPA template and analyze the architecture.

**There are three folders that you can highlight with this template.**

- The client app has the client application
- **Controllers** have the controller that is related to the business logic in the server
- **Pages** refer to the razor pages which have to do with the UI components that are rendered on the server.

  You can run the project with the play button. and if the project is running you are going to see the HTML page that explains how the

template functions.

The template is straightforward yet has all the information required to build a new web application with a monolithic architecture and adhere to best practices on both the backend and frontend. This template is a wonderful choice if you need a web application with high-performance security. You need more expertise in JavaScript and C# for the backend of this template because React.js employs JavaScript in its syntaxes. For developing our UI and business logic in the template that we examined in the Templates for.NET Core part, C# was more crucial than JavaScript.

# Review Question

- How do you access the templates for SPAs?
- How do you choose and search for templates?
- What are the templates for API?

# Summary

Depending on the workload introduced during the installation process, Visual Studio offers a variety of alternatives for generating projects using templates. To rapidly search the templates, we offer four filters. We have the option to utilize the search box to discover our template, using a keyword or a specific word, in addition to filtering by language, platform, and project type.

We must always input a project name and the folder where our project will be generated after choosing the template to utilize. The next step is to choose the target framework and some optional data. After choosing to Create on the screen, Visual Studio will launch the project, giving us the option to get started on our project right immediately.

# CHAPTER 5

# DEBUGGING AND COMPILING YOUR PROJECTS

The first thing to learn as a developer is how to debug programs including .NET projects.

You need to be familiar with Visual Studio 2022's many windows to utilize them to examine data and find flaws, as well as with as many of the debugging tools it provides, such as breakpoints. A breakpoint provides the ability to halt the execution of a program, allowing you to verify its behavior and observe the states of all the objects.

In this chapter, we are going to be taking you through the ways to debug and compile your project.

## Technical requirements

### Debugging projects in the visual studio

#### Understanding the technical aspect of debugging

To clarify the terms, here are the things to know.

First, checking for code problems is what is meant by the phrase "debugging." The usage of a tool like an IDE is not always included in this. Debugging may, for instance, include looking for mistakes in code that have been written on paper.

Typically, this is not possible, hence a tool called a debugger is frequently utilized. You may use this tool to examine your code while the program is running since it is tied to the process you will use to run the application.

### Differentiating between debug mode and run mode

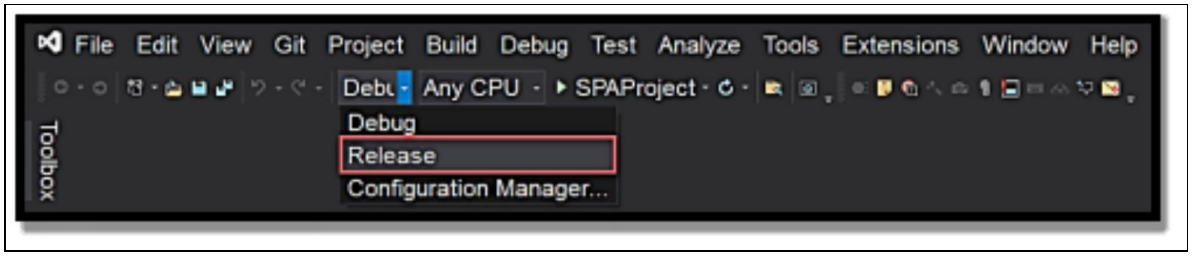
The term debug mode and run mode can be confusing. So, we must clarify them.

We are going to first work with the **debug mode.** In this instance, we can activate the option by first choosing the **debug** configuration by selecting

the green button that is in the same place where the project name is.

After choosing this option, the debugger is then going to be added to the application execution, which gives us the option to use functions like stopping at breakpoints within the application.

Alternatively, you can use another configuration from the drop-down.

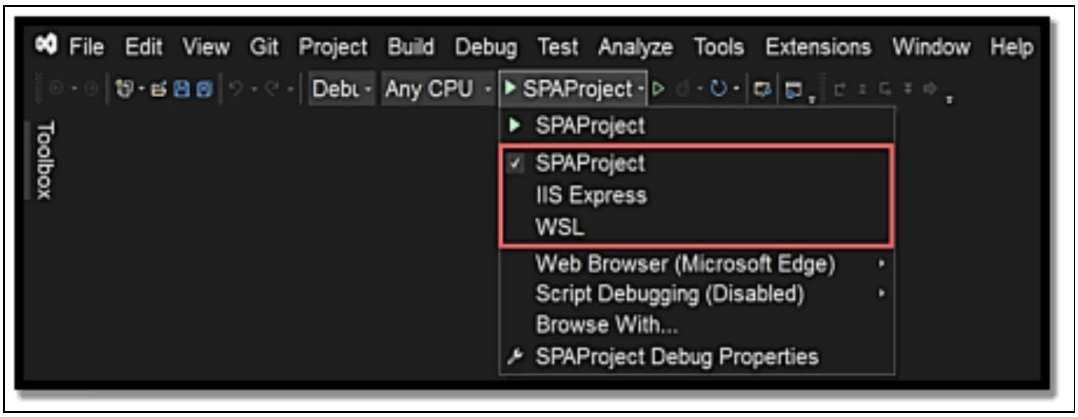



You will have a better notion of how your program will act toward the end user if this option is chosen and we move forward to begin the execution of the application without the debugger connected. This suggests that while efficiency will be improved, code debugging and exception location visibility will not be possible.

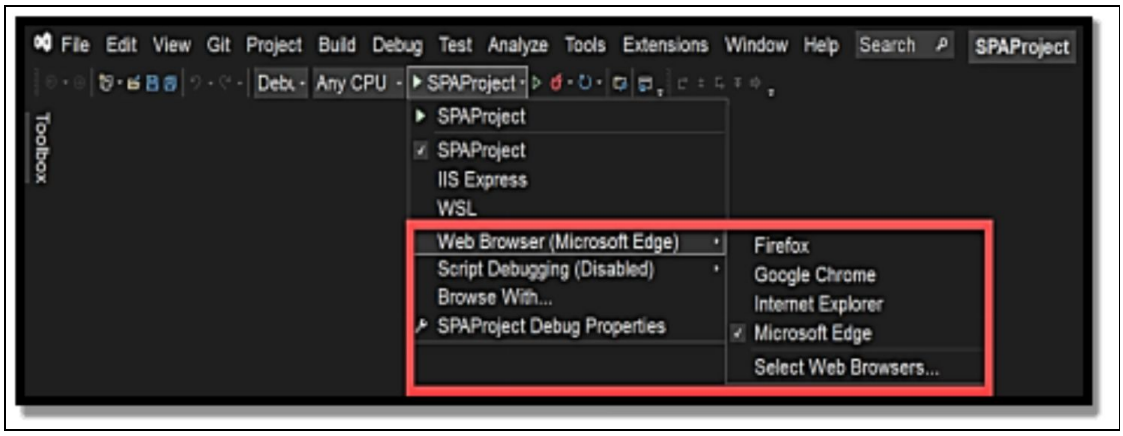
# Project debugging initialization options

We must be aware that we have a variety of alternatives for running our tests as a component of the debugging and execution settings of an application.

**As seen in the image below, you may see several choices for the deployment of your application if you drop down the options next to the green button with the name of your project:**



The first three options can be used to specify which of the servers you will like to use for testing.  The Kestrel server is used by default. However, you can use the IIS Express or for Linux-based environments, you can use the WSL




These options are efficient especially when you will like to use a specific browser for its tools. You will see the pause, stop and start buttons.

# Exploring breakpoints in visual studio

The breakpoints are important in the development of programs. You can stop the application flow at any time in other to look through the object's state. To then place the breakpoints inside the visual studio you need to first position them next to where the numbers are lined. Then you are going to see a gray circle that hides and unhides when you move the cursor above the numbering lines.

After finding the line to debug, left-click to turn the circle red. Then move the cursor away from the circle.

Then when you start the application with the breakpoint application flow stops once you start the application.

When the program has reached the breakpoint, we may use a variety of tools to see the status of the program. For instance, if we put the cursor over a variable before the debug line, we can check the variable's current status. If it is raw data, you can instantly see its value; if it is an object, you may access its attributes and look at each one individually.

We may use several tools to check the program's state after it has reached the breakpoint. For instance, we may examine the state of a variable by placing the mouse over it before the debug line. While dealing with raw data, you can immediately see its value; when dealing with objects, you may access their characteristics and examine each one separately.

Also, when you right-click on the breakpoint that you picked, you will then find the option to add some other functionalities.

Adding breakpoints this way can be easy. However, we have to acknowledge the fact, that you are going to need some breakpoints active in some circumstances.

## Navigation between breakpoints

Since we now know how to place the breakpoints inside the source code, we can then go further to execute the application differently through the button on the upper menu.

With "**step into"** This enables us to run every line of source code. If we call methods that are accessible through the source code as part of the flow, we will automatically move there after doing line-by-line debugging.

**Step over.** This enables us to avoid stepping into any methods that the current method invokes by simply navigating the lines of code within the current method.

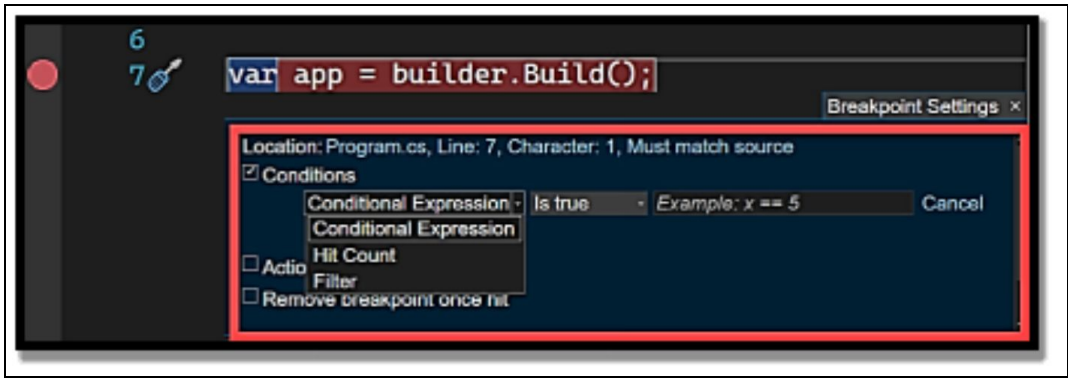
**Step out** you can use this button inside the method. This makes us **step out** and go back to the line after executing the method.

To then examine the breakpoints, you can then activate the breakpoints window. You can do this by going to **debug** then **windows** then **breakpoints.**

## Conditional breakpoints.

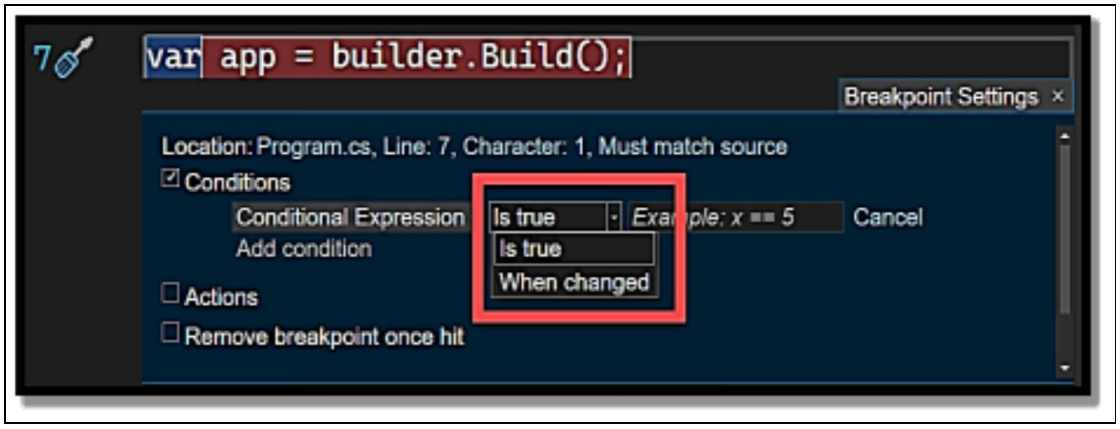
You may occasionally need your breakpoint to halt when specific requirements are satisfied. The best course of action in this situation is to use conditional breakpoints. Simply right-click on the sidebar to display the various breakpoint types that are accessible, as shown in the Navigating between breakpoints section. This will allow you to enter a breakpoint of

this kind. As demonstrated in this image, choose the **Conditional Breakpoint** type to bring up a dialog with pre-selected options.




Then inside the conditional breakpoints, you can then choose the **conditional expression** condition, the **filter** condition, or the **hit count** condition

**In the conditional expression, you can terminate the application when a previously defined true condition is met or when the value of an object changes:**




In the first scenario, we may define a Boolean expression that can compare anything from two values to complex objects.

Then it creates a for loop which prints the numbers from 1 to 10 so that there is a better appreciation for the examples. You can also place a pin on the “I” variable by hovering over the variable while running the application and selecting the **pin.**

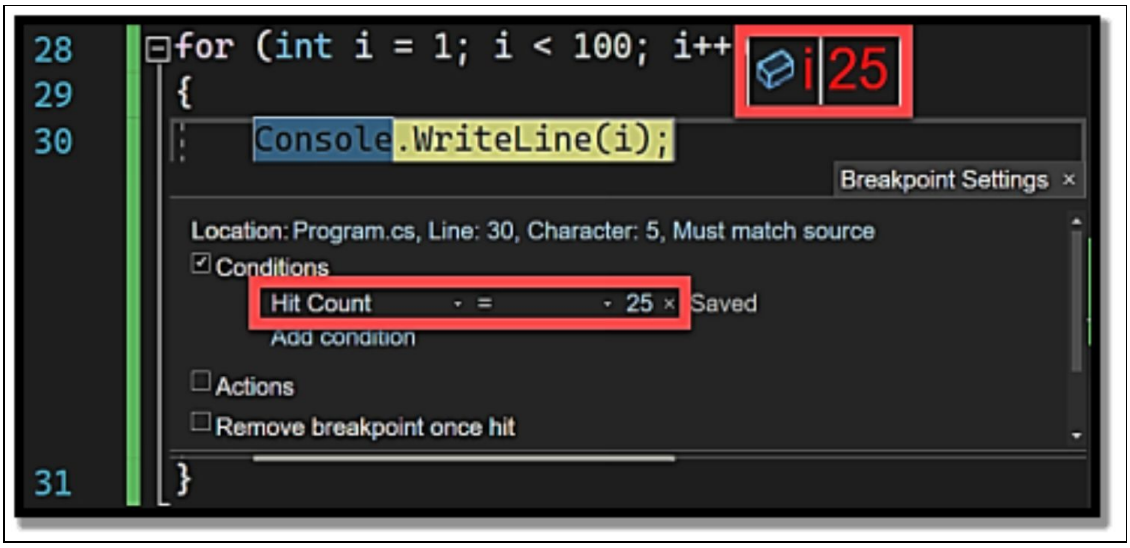
You can indicate if you want to stop the application if the “I” value is more than 5. Once you start the application, the breakpoint stops when the “I” value is 6.

Then you can monitor objects, fields, or property making the breakpoint stop when it changes. You can see this here when we changed the **conditional expression** type to “**when changed”** and you indicate you want to then monitor the variable of “I”


This means that when the breakpoint is executed the value of “I” is 1

# And with the hit count

The Hit Count condition type was created to address issues with loops. For instance, let's say you have a set of conditions that need you to print a particular number of reports, but you suddenly find that the calculations start to falter after report 25. Setting a breakpoint, starting the program, running the code 25 times to reach iteration 25, and then evaluating it could be your first thought. Although this is the right method to accomplish it, setting the Hit Count option and specifying the number of rounds you want to skip. in this case, 25, as we can see in this image is a better way to do it.



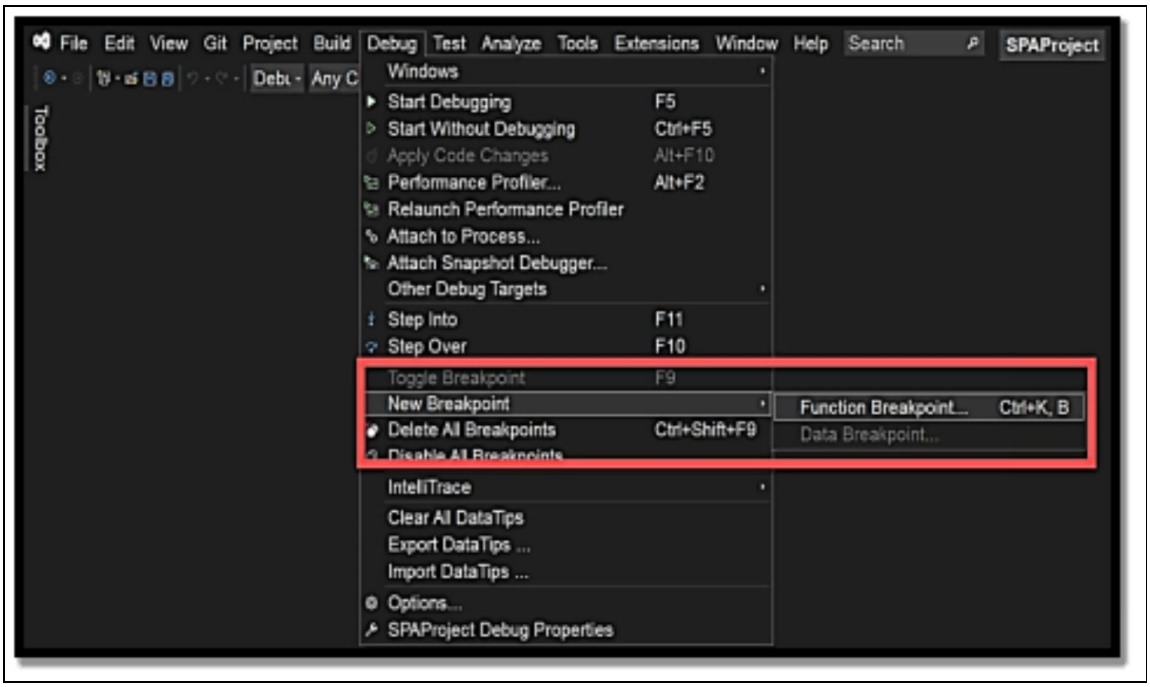

Then with the **filter** condition, as a result, we will be able to set up a breakpoint based on a set of predetermined phrases. These expressions, which vary from the machine name through processes, to thread attributes, may be found in the Filter box, as illustrated in this image:

## Function breakpoints

Even if we haven't established a breakpoint as done in the Conditional breakpoints section, the Function Breakpoint type will let us debug a method as it is being performed, as its name suggests. If you have hundreds of code lines and the name of the function you want to debug, this is helpful.

This breakpoint is positioned differently from the breakpoints we observed in the Conditional breakpoints section.

First, as shown in the Figure below, rather than inserting the red dot in the code, you may select **Debug** then go to New **Breakpoint** then Function **Breakpoint** from the menu, and type the name of the function **you wish to set the breakpoint in:**




Then you can add the **breakpoint function** by going to the **breakpoints** window.

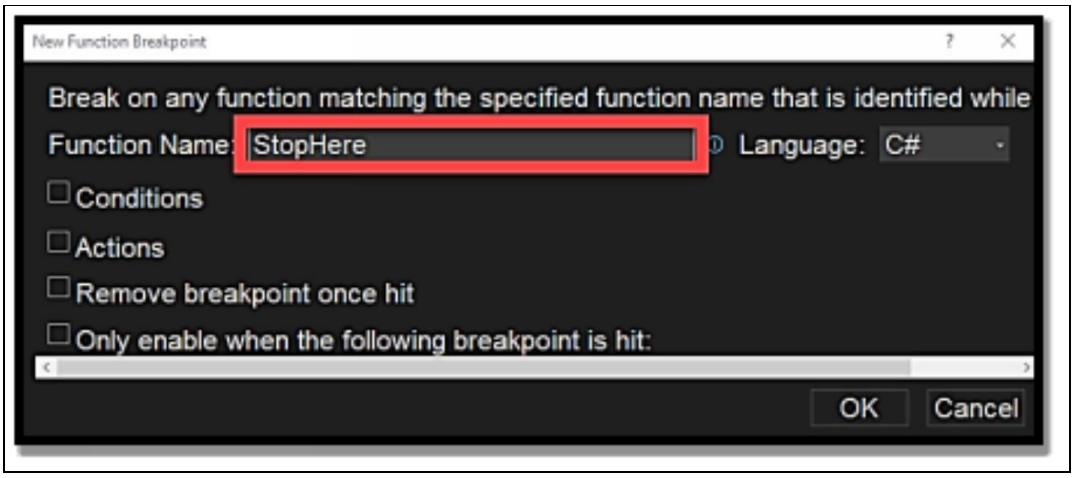
After choosing the **function breakpoint** button you are going to see another window, that inquiries about the method's name.

If you know the function name you want to evaluate, then you can type it inside the **function name** box.

**There are three ways to do this**

- Type the function name
- Specify the name of the function using a particular overload
- Specifying the d11 name if you already have a source code for it.

For this example, we are going to enter the name of the **stophere** function.

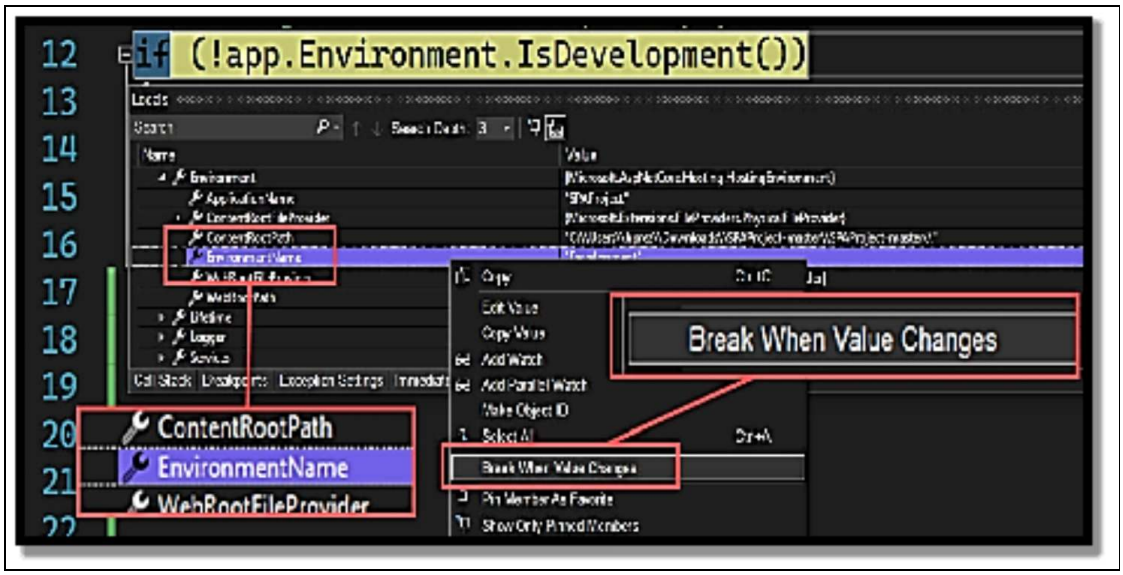



After establishing the date, you need to execute the application once you call this new method. This makes the application stop at the beginning of the method.

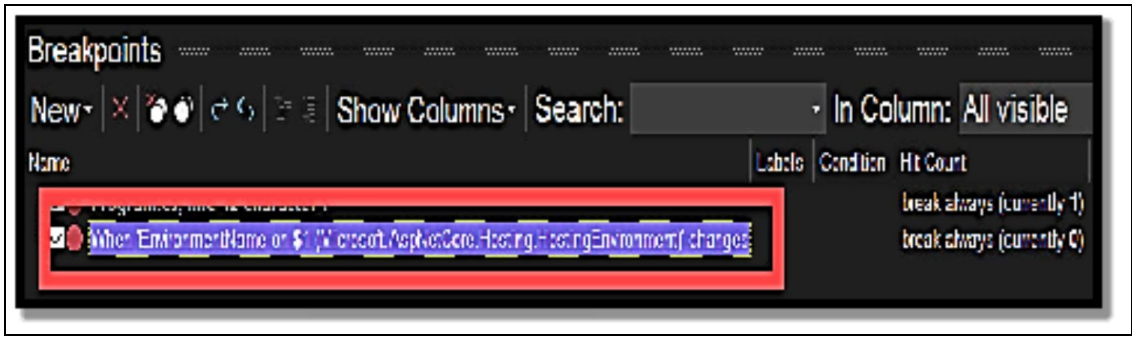
## Data breakpoints

Data breakpoints are your best choice if you want to be able to set breakpoints when an object's properties change. You'll notice that the option is disabled if you try to add one of these breakpoints from the Breakpoints box.

This is because before we can begin monitoring the property, we must first set a breakpoint at a certain location. Start the program after this is finished until the breakpoint is set, and then right-click in one of the **Autos, Watch**, or **Locals** windows where the instance of interest appears to see the option called Break When Value Changes, as shown in the image. We want to monitor the **EnvironmentName** property of the **app** object in this window.




After choosing the option, you're going to see the way the breakpoint is automatically created, which detects when the property that we have specified changes.

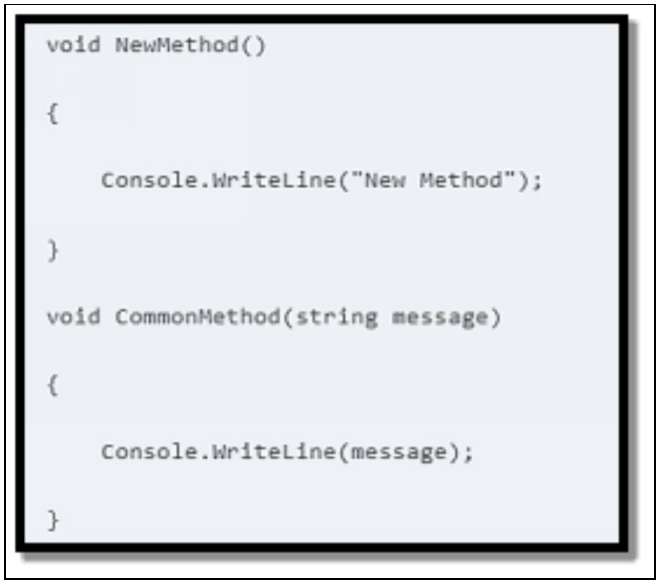



Dependent break points temporary breakpoints. Imagine that we then modify the property data, with this code,
app.Environment.EnvironmentName = "Testing Data Breakpoints";

Then after running the application, you are then going to see the way that the breakpoint that was there initially was reached.

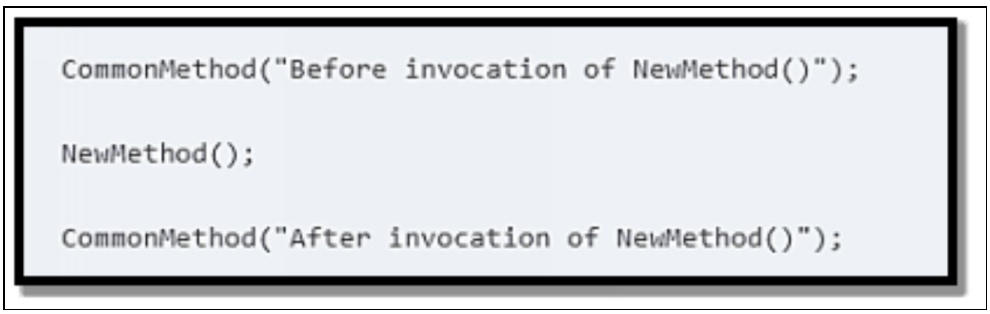
This particular breakpoint is a unique breakpoint that will only be carried out if another breakpoint has been reached. It might not be particularly helpful in a straightforward context, but in complicated situations with hundreds of features at play, it might be highly beneficial.

Imagine, for instance, that you are testing a new capability that calls a method that is used several times across your program. A standard breakpoint would cause the method to halt each time it was called. You can declare that you only want to halt execution if the breakpoint of your new feature is reached by using a dependent breakpoint. Then to show how it works, we added some methods to the program.cs file



```
void NewMethod()

{

    Console.WriteLine("New Method");

}

void CommonMethod(string message)

{

    Console.WriteLine(message);

}
```

The point of the **commonmethod** function is that it gives us the opportunity through the contents of the string that is passed as a parameter and to know where the breakpoint stops. Knowing this we can add some lines we can call **commonmethod, newmethod, and commonethod again.**



```
CommonMethod("Before invocation of NewMethod()");

NewMethod();

CommonMethod("After invocation of NewMethod()");
```

The line of code on which we wish to depend—the one that must be run first in order therefore for the dependent breakpoint to also be executed—will first have a regular breakpoint placed in it. Because it is the method we want to test in our example, we will put it in the NewMethod functionality.

You can then right-click the line to create the dependent breakpoint.

Then you are going to see the **breakpoint setting,** where you can know which of the breakpoints you will like to use to launch the dependent breakpoint.

Once the **breakpoint setting** closes, then you are going to see the way the special breakpoint is created.

Finally, you can run the program to see how, despite "**CommonMethod"** having been called first, it stops first on the **NewMethod** call. If you keep running the program, the dependent breakpoint would stop and display the warning after calling "**NewMethod ()**."

### Temporary Breakpoints

Temporary breakpoints are not meant to last. They are breakpoints that, as their name implies, are immediately destroyed after being executed. You must right-click on the breakpoint line and choose Insert Temporary Breakpoint to set one of these.

If you execute the program, you will observe that it will be immediately removed after reaching the breakpoint. This kind of breakpoint might be used, for instance, to assess the cycle's initial iteration.

## Inspection tool for debugging

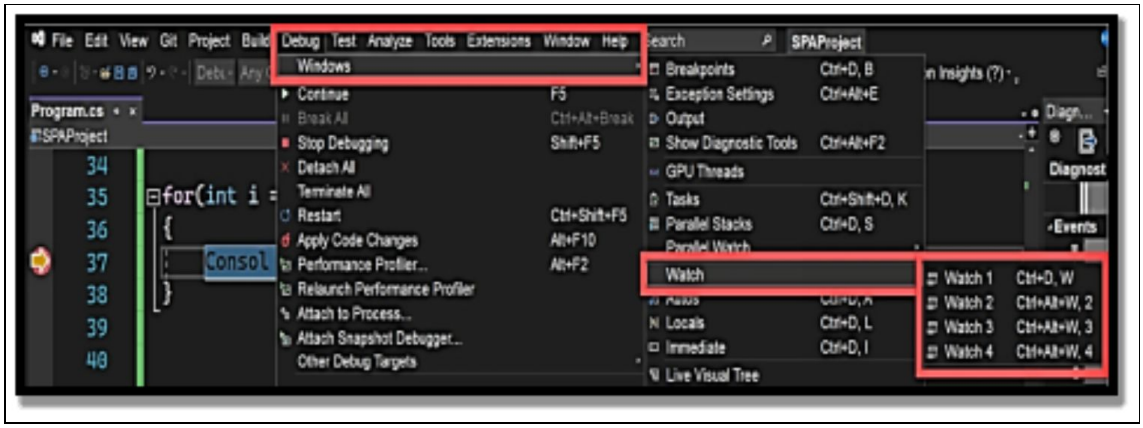
You can use the inspection tools in visual studios to find information to visualize. In this section, we are going to be taking you through them

### Watch window

As we step-by-step run our code, the Watch window will let us monitor the values of variables or properties. It is particularly helpful when we have code that is repeated frequently, such as cycles or popular procedures. We must first set a breakpoint in the code and run the application before we can access this window.

You can then run the **Debug** and **Windows** menu whenever the application reaches the breakpoint. This will display a new set of debugging windows

that are only accessible while the application is running. Let's choose a window by clicking the **Watch** option.



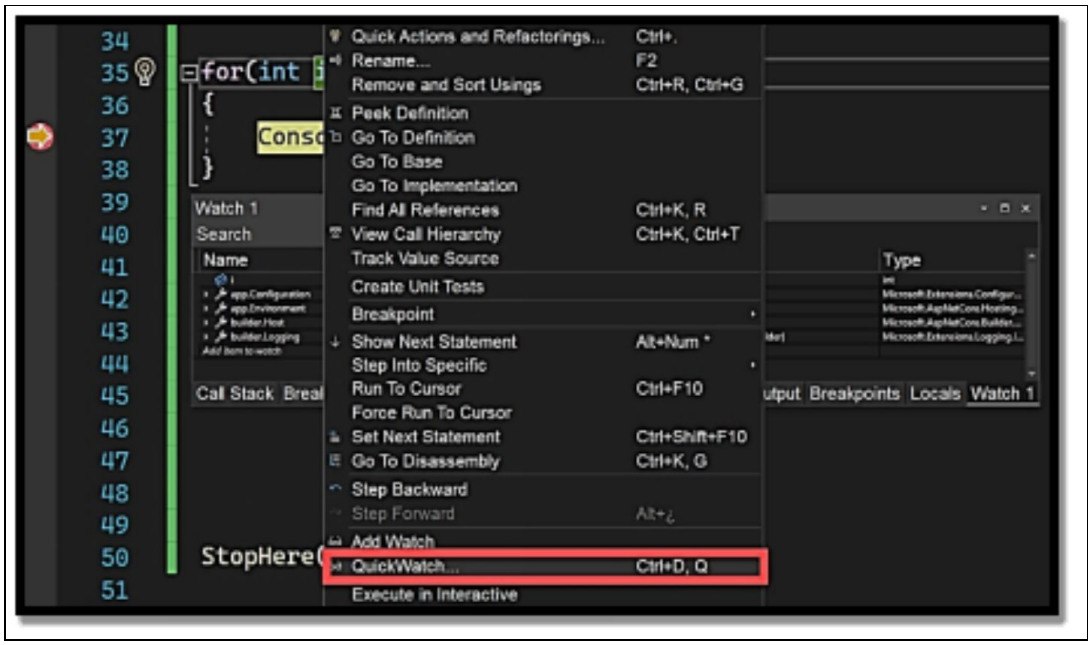

After you open the window, you can then add the names of the variables and properties to monitor. If you enter a valid variable name for the scope, then you can see the corresponding value inside the column for **value.**

You can also add a variable to the watch window by right-clicking on the variable you will like to monitor whilst running the application

## Quick watch

The Watch window is excellent for following values as they move through the program. However, we may opt to utilize the **QuickWatch** window, which can be accessed through the **Debug** then **QuickWatch** menu, if we wish to employ the same capability as the Watch Window to test particular expressions without retaining the same ones.

**For the illustration, we will utilize the following for loop example (found in the Hit Count section):**




This selection will open a modal window, preventing us from using the application further until we shut it. We will be able to view information like the expression's name, value, and the kind of data we are evaluating in this window.

We will be able to change the phrase in this window, which will be helpful. As a result, we may assess more variables and even carry out any actions we want to verify. This image illustrates this in action after entering a new expression to determine whether or not its outcome is true

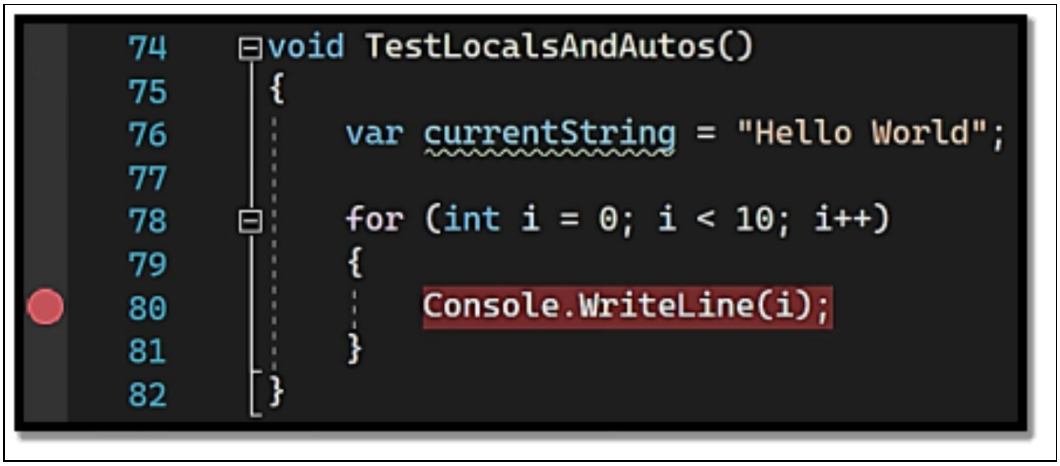
# The autos and locals' windows

You can find out information about variables and attributes in the Autos and Locals windows without having to put them elsewhere, unlike the Watch window.

These, however, have a limited range. The Locals window will only display values for the current scope, which is often the function or method in which the breakpoint is located, but the Autos window displays the value of the variables around the breakpoint we've put.

The fact that these windows will only be shown when the program is running and once a breakpoint has been set is another crucial aspect of them.

I added a new function and a breakpoint to our Program.cs file to demonstrate this window pair.



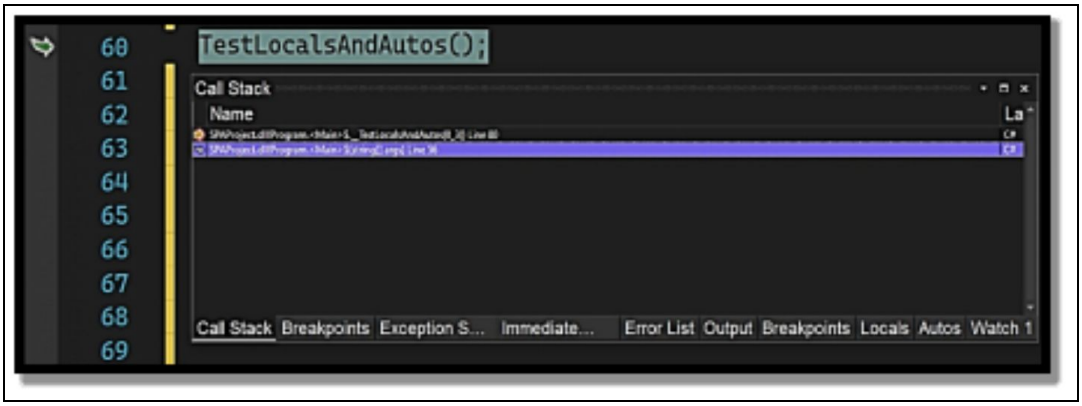

Looking at the **local's** windows, you are going to see all of the local variables and in the autos window, you are going to see the contents for the "I" variable as it is within the scope of the breakpoint which refers to them **for a** loop.

# Call stack

Although the most common form of error correction is the use of windows, as we saw in the Watch window, Quick Watch, and The Autos and Locals windows sections, we also have several additional options to monitor the order in which our methods are invoked.

One of these windows is called Call Stack, and when a breakpoint is set, it displays the stack of calls that have been placed up to that point. We must choose the **Debug** then **Windows Call Stack** option to see the window. The window may be docked using the thumbtack/pin icon if you wish it to appear for each debug session even though this menu item will only be accessible when the application is active.

In this image, the identical code from The Autos and Locals windows section was run and stopped at the same breakpoint. The Call Stack pane displays the line of code in which we are after the program has stopped because of the breakpoint as well as the collection of method calls that were performed to get to the breakpoint, as seen in the accompanying picture.




If we double-click on any of the calls, we can also examine the application's status in each one of them. This will enable us to view the values of the properties and variables at a certain moment in the application's flow.

To avoid confusion with the breakpoint's color, a separate color will be used to highlight the code area from where the method is called when we click this.

# Immediate window

Then there is the **immediate** window. You can open this window by going to **debug** then **windows** then **immediate,** to then analyze expressions, run statements, and also print the values of a variable and property.

# Review Question

- How do you debug projects in the visual studio?
- How do you navigate through break points
- What are breakpoints?

# Summary

There are several options in Visual Studio for debugging source code. We learned about breakpoints in this chapter, including what they are, the many kinds of breakpoints, and the accompanying windows we may open to tracking data in variables and attributes. This knowledge has been crucial since it will enable you to address issues with your code as they arise.

# CHAPTER 6
# ADDING CODE SNIPPETS

A code snippet is useful when reusing codes. There are some default code snippets that Visual Studio offers and there is also the possibility to create one for yourself. We are going to take you through the process of writing code faster with snippets and even creating your own.

## Technical requirements

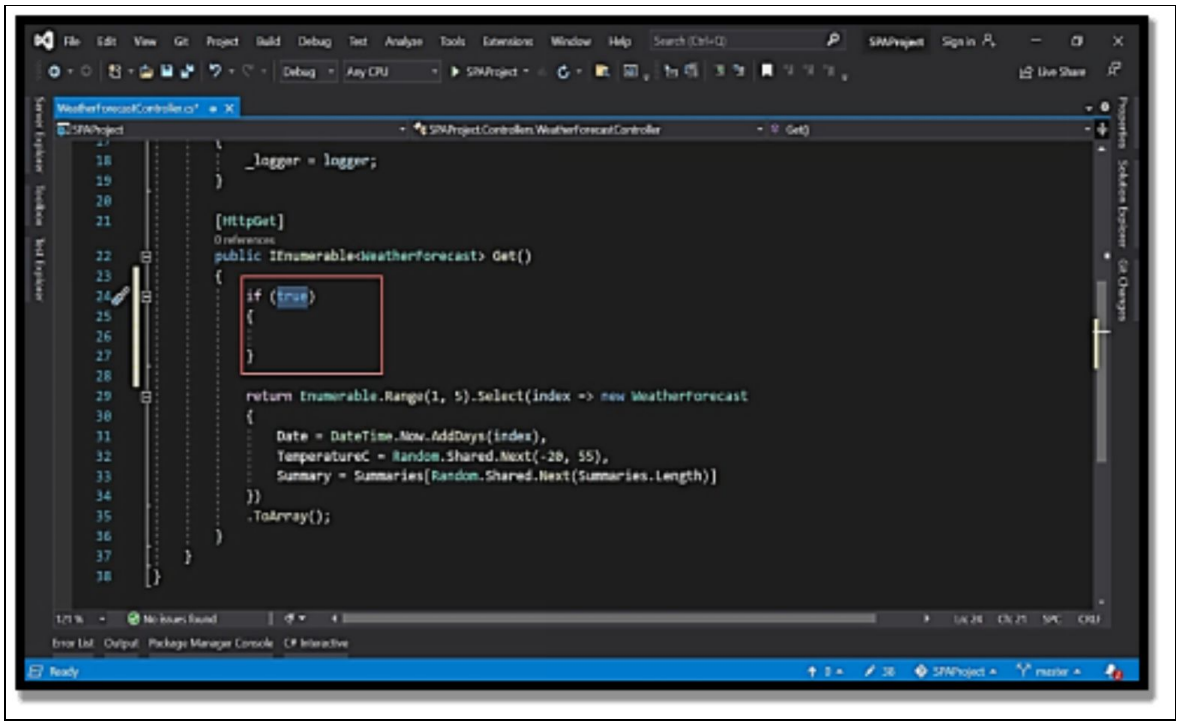
### What are code snippets?

By building templates that create frequent code, such as conditionals, loops, or comment structures, code snippets are a quick and easy method to reuse code.

For practically all of the available technologies and programming languages, Visual Studio comes with a ton of code snippets by default. Let's look at some of the different uses for code snippets in Visual Studio.

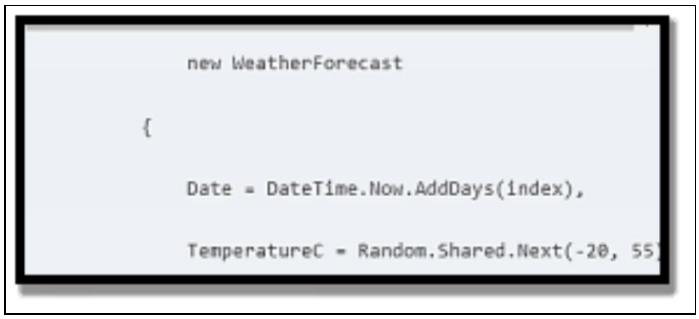
Given how frequently conditional statements are used for coding, Visual Studio provides you the opportunity to write this code fast. If you don't want to take any action, you can click it or keep writing. To automatically generate the if statement, you may alternatively hit the tab key twice.

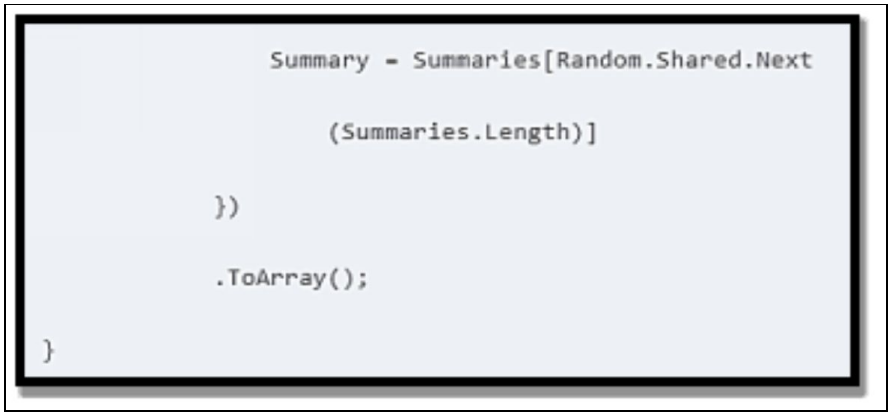
If you see the code snippet highlighted, then you can double-click on the tab key so that it generates a code for the IF statement with the brackets.




You are going to get the if statement with the braces and TRUE as the value by default. Then you are going to have to replace true with your condition. In this example, you can place the condition so that it gives you an empty collection if your operating system is Linux.

```
[HttpGet]

public IEnumerable<WeatherForecast> Get()

{

            if (OperatingSystem.IsLinux())

            {

                return new List<WeatherForecast>();

            }

          return Enumerable.Range(1, 5).Select(index =>
```



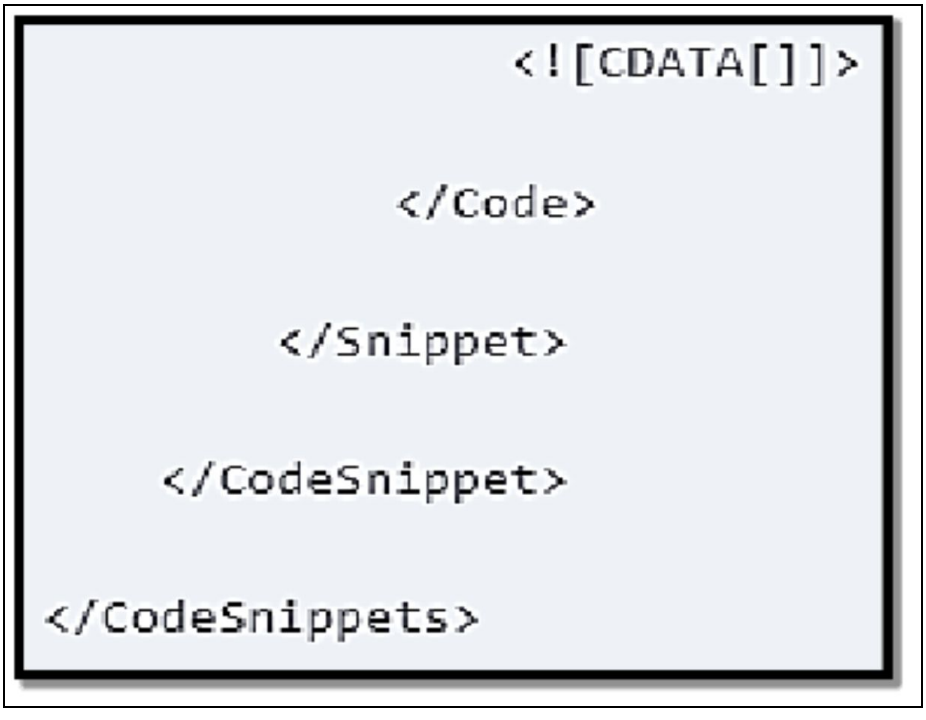

If you checked the previous code block, we placed a condition in the GET method before going to the default logic to see if the operating system where the app is running is Linux. Inside this condition, you can return an empty list.

You are going to find important default code snippets for c#.

# Creating code snippets

To then create a code snippet inside the visual studio you are going to first have to create the file using a snippet extension. The file has an XML format and it also has a refreshable base template. Here we have a template example.

```
<?xml version="1.0" encoding="utf-8"?>
<CodeSnippets xmlns="http://schemas.microsoft.com/
    VisualStudio/2005/CodeSnippet">
    <CodeSnippet Format="1.0.0">
        <Header>
            <Title></Title>
            <Author></Author>
            <Description></Description>
            <Shortcut></Shortcut>
        </Header>
        <Snippet>
            <Code Language="">
```




Here are the details of the property

The title refers to the name and the general information

Author refers to the author or the creator

The description refers to what your code snippet does

Shortcut refers to the shortcut to call the code snippet.

**Here are the properties of the snippet:**

Language refers to the programming language

[CDATA[] holds the code

Then you can create code snippets that test if the operating system that runs the code is Linux

```
<?xml version="1.0" encoding="utf-8"?>

<CodeSnippets xmlns="http://schemas.microsoft.com/Visual

    <CodeSnippet Format="1.0.0">

        <Header>

            <Title>If Linux condition</Title>

            <Author>Myself</Author>

            <Description>Conditional to now if the

                operating system is Linux</Description>

            <Shortcut>ifln</Shortcut>

        </Header>

        <Snippet>
```


Then you can create another folder inside the documents or any folder specific to the project then save the file with the SNIPPET extension.

Finally, add the code snippet section inside VISUAL STUDIO and go to tools then code snippets manager and enter the **language** dropdown and choose Csharp

The Code Snippets Manager may be opened by pressing Ctrl + K, then Ctrl + B.

Depending on the installed workload, the Language option covers any programming language and technology that Visual Studio supports.

You may choose the folder where your code snippet was produced by clicking Add...

Once the folder has been added, a new folder with the updated code snippet will appear in the list. If you choose it, the right panel will display further information.

Then you can use the code snippet in C# files

# Deleting code snippets

Human mistakes might cause us to include unnecessary code snippets or choose the incorrect code snippet. Visual Studio can remove code snippets in certain situations. Go to **Tools** then enter Code **Snippets Manage**r and choose the CodeSnippets folder to access this option. The code snippet you added to the section on creating code snippets is located in this folder. Choose the appropriate folder for you if you specified a different name. The placement of the Remove button is visible.

The entire folder and all the code snippets inside will be deleted if the Remove button is pressed. We must make a folder with a good name for our code snippets because it is not feasible to remove code snippets individually in Visual Studio 2022. After deleting the folder, Visual Studio will no longer recommend code snippets.

# Importing code snippets

To add code snippets into a folder that has been created already inside the **code snippets manager,** you can choose **import** and then choose the code snippet that you will like to import inside the chosen folder.

Choose the code snippet that was created inside **creating code snippets** section and choose **open.** And finally, you need to choose the location folder for the code snippet and choose **finish.** You are then going to see the code snippet.

# Review Questions

- What are code snippets
- How do you create one?
- How do you delete snippets

# Summary

You can now use Visual Studio code snippets to boost productivity. You may utilize patterns to develop your code snippets to satisfy your needs and discover the code components that are frequently used in your design. Additionally, you are aware of how to handle code snippets by importing and deleting them. You will understand the value of code snippets and why

Visual Studio is a potent IDE that enables developers to create code more quickly after finishing the examples in this chapter.

# CHAPTER 7

# CODING EFFICIENTLY WITH AI AND CODE VIEWS

In one way or another, artificial intelligence is a large and fascinating area, that can help us live better lives. If you don't believe it, consider how frequently you use the Google search engine during the day. We see, hear, and utilize it every day. Other places you may find it include photo-editing software, where you can, for instance, nearly perfectly erase the backdrop from an image. Another excellent application of artificial intelligence is in social networks, which are continually digesting the finest suggestions to keep you on them as long as possible.

Fortunately, predictive code integration, a feature of current software development tools, allows us to select the specific portions of code we require at any given time. This is accomplished using a robust feature in Visual Studio called Visual Studio IntelliCode.

Similar to this, we have a variety of visual tools and windows that may be used to quickly locate relationships in our code.

## Technical requirements

### Understanding code lens

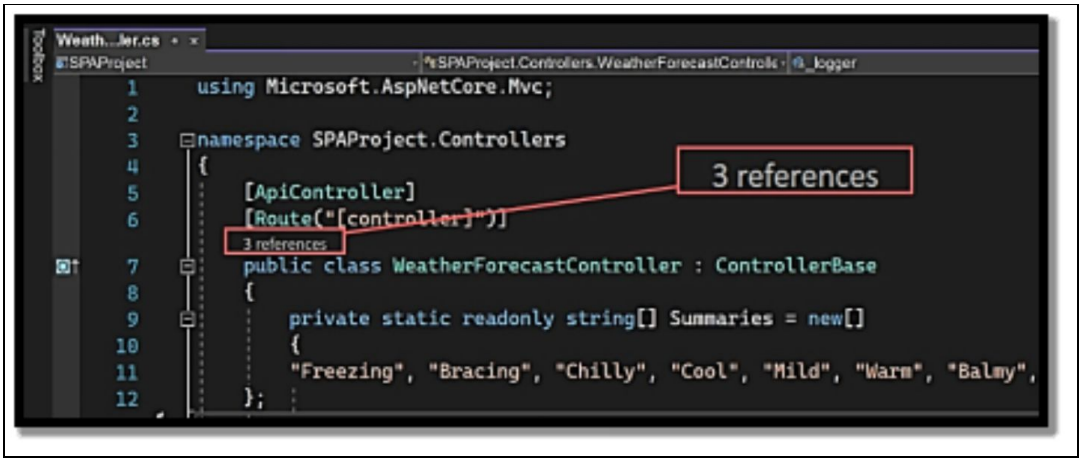
The codelens help to find the references within code, the links between the parts, go through the change history inside the code, the bugs linked, code review, unit testing, etc.

### Finding references in code

Since the very first time we used Visual Studio, CodeLens has been displayed in our code files.

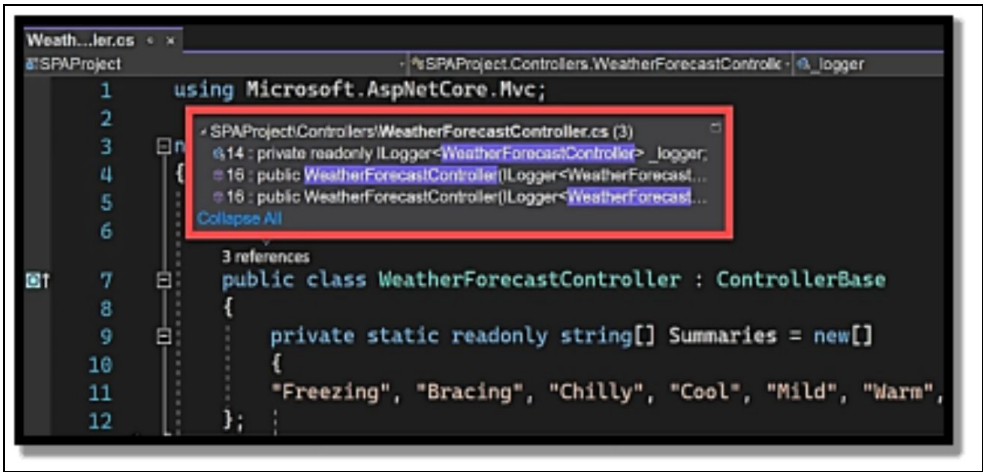
This is verifiable by visiting any class, method, or property and making sure that a sentence specifying the number of references to it in the project is there.

**In this image, we opened weatherforecastcontroller.cs file and it shows us the three references of the weatherforecastcontroller:**

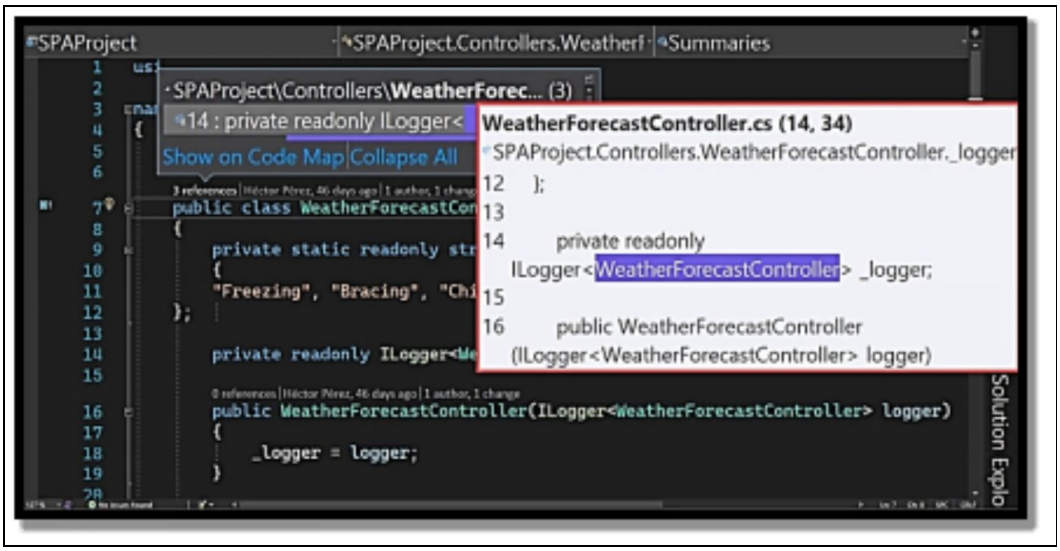



This indicates that our app uses the WeatherForecastController class three times. We can examine all references that make use of this class by moving on and clicking the legend labeled 3 references, which is seen in this image.

**As we continue with the example in this image, we can see that it is utilized in the WeatherForecastController class, notably in lines 14 and 16:**




In addition, we may move our cursor over any of the lines that were discovered, which will display a portion of the four lines of code that are closest to the reference. In this approach, we can better understand its intended use

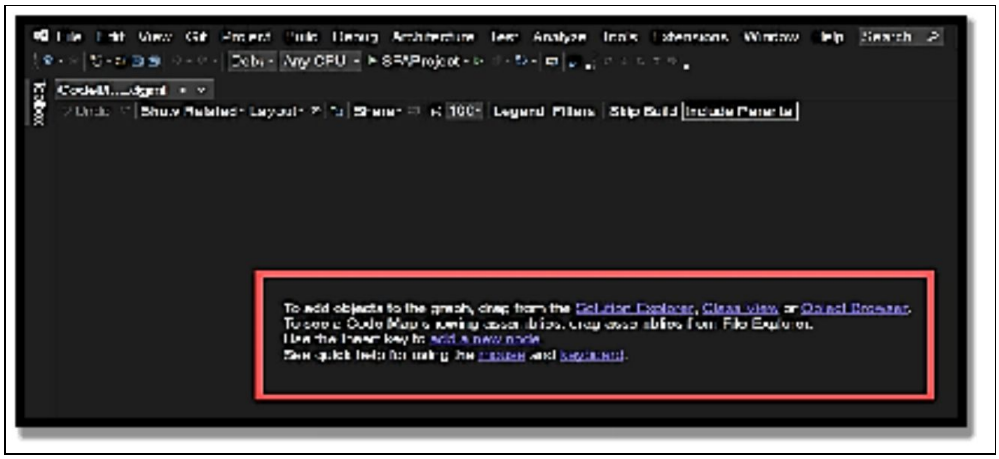



This is important when you are working on a new project and you want to know what the parts of codes do.

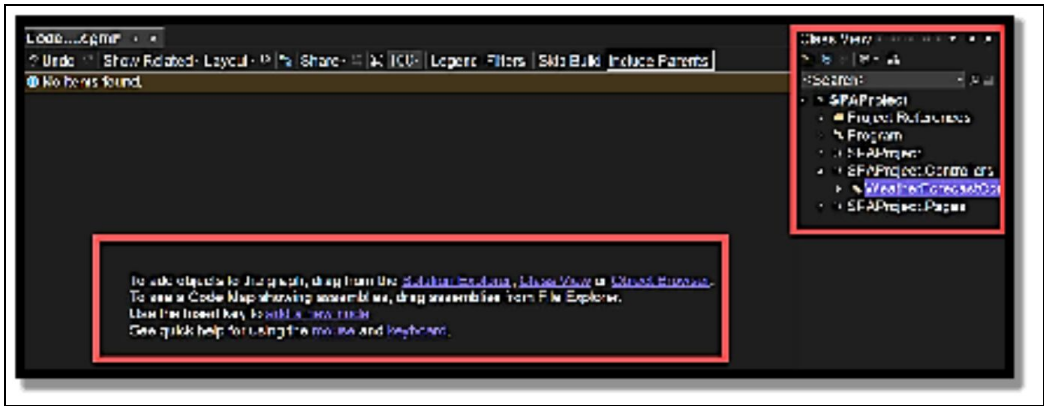
# Code maps

Code maps are a quick and effective approach to visualizing relationships in code. As its name suggests, this tool enables the production of visual maps from the code. With the help of this tool, we can observe the relationships, attributes, and organizational structure of the entities, which enables us to gauge the potential impact of changes we make.

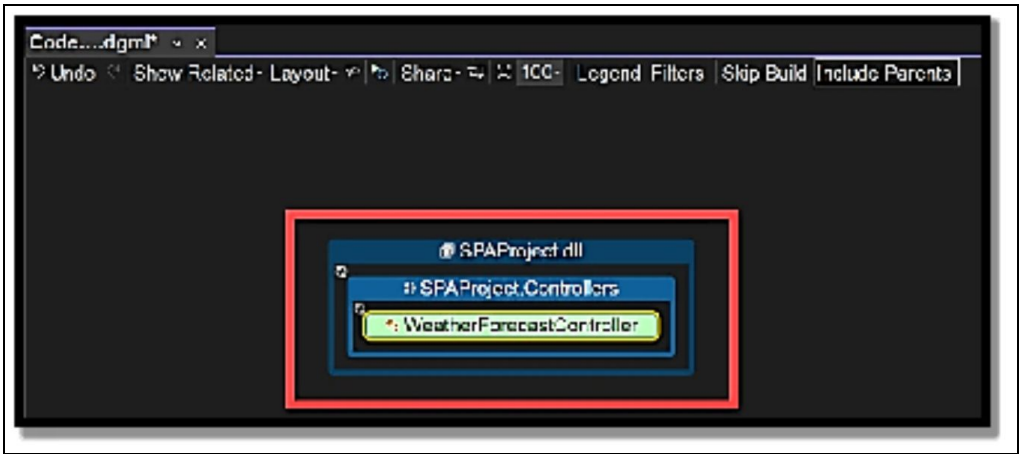
Code maps may be made in a variety of methods. First, choose **Architecture** then New **Code Map** from the menu. As demonstrated in this image this will launch a new document with a.dgml extension into which we will be asked to drag files from the solution explorer, class view, or object browser.



Then to do a test, select the **class view** link. This opens the class window and it expands the SPAProject.controllers namespace. This then shows you where the weatherforecastcontroller class is as seen in this image.



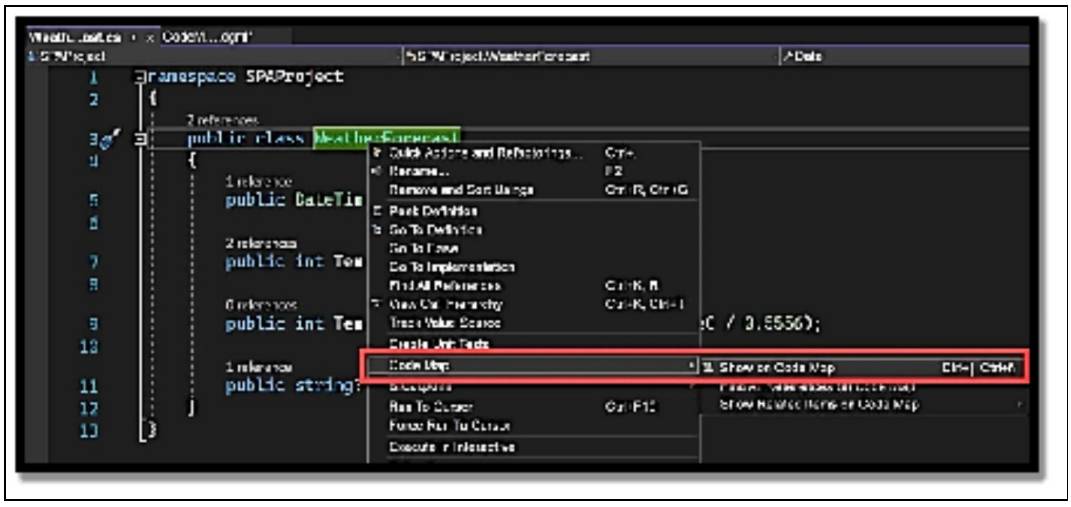
Then drag **weatherforecastcontroller** class inside the code map file. Then you are going to see a graph with the dragged class, namespace with it, and then .dll where it is hosted.



Furthermore, to expand the weatherforecastcontroller class inside this diagram, we are going to see the parts of the class like the attributes and the behaviors, and the relationships of the same class.

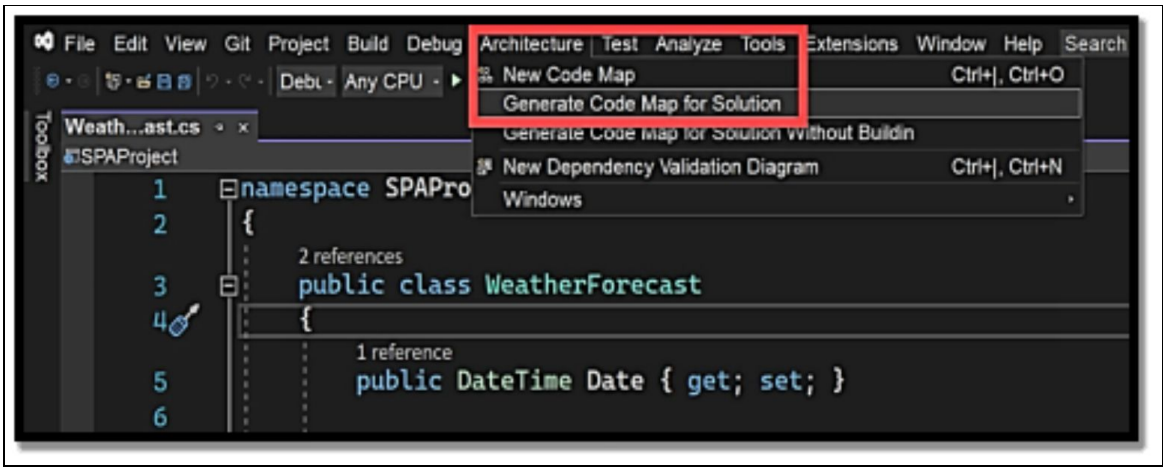

In this image, here is the code map of the weather forecast controller class working after all its members have been expanded. With this, the fields, the properties, and the methods are all related.

Going to the file in which the member we are interested in is located, for instance, the WeatherForecast.cs file, is another method for creating a map code from the source code. The Code **Map** Then **Show on Code Map** option will appear when we right-click on a class, method, property, or field once the file has been opened, as demonstrated in



This option then gives you a new .dgml file and if you have created one already like in this case, then add the references with the respective relations inside the file that was opened previously.

Then to see the relationships inside the solution and not add entity by entity from the menu **architecture,** you can choose to **generate a code map for the solution** option in this image.




This then generates the respective code map for the entire solution. It can even take a lot of time depending on the number of references within the code.

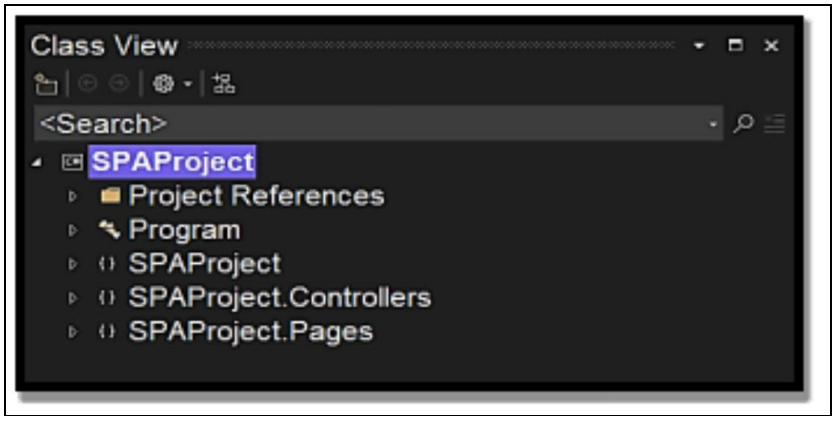
# Working with code Views

There are some other windows that you can use to assess the project class and the members quickly

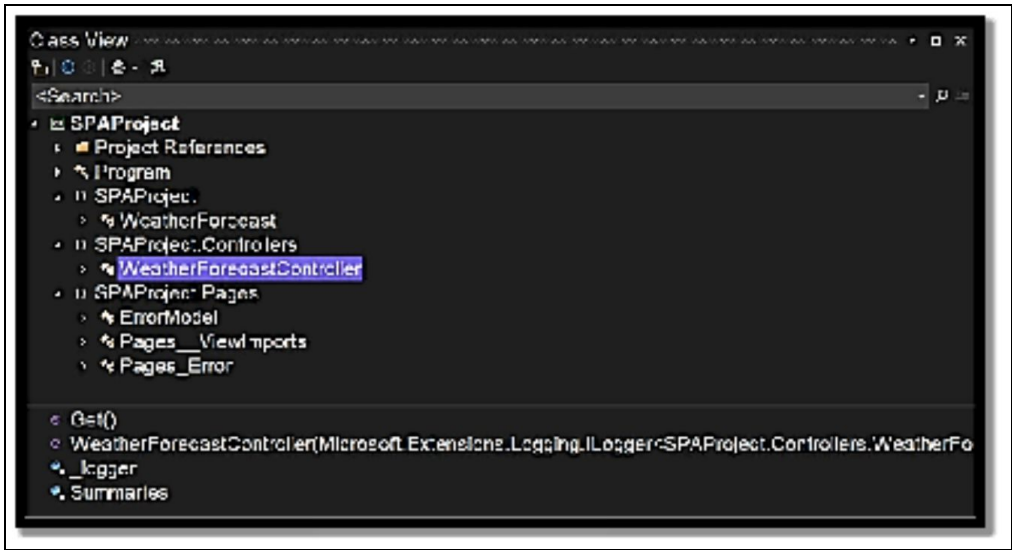
## Class views

The class view is a window that enables you to rapidly access all of the components of a Visual Studio project, including namespaces, types, interfaces, enumerations, and classes. You might not think it's very useful if you've just used Visual Studio for a few minor projects. However, it is a great choice to look at code if, like me, you deal with solutions that can contain up to 20 projects or more.

The Class View window, which displays all the components of the open solution, may be accessed by choosing the **View** then Class **View** option from the menu.




From the image above, we have a very basic understanding of the structure of the project by going through the namespaces the project has been divided into. After expanding the nodes of the namespace, you can then find the different classes that are part of the namespaces.




Also, the properties and methods that make up any selected project component, such as a class, will be shown in the lower portion of the window

Another fantastic benefit of this window is that since the changes will be done instantaneously and automatically, you won't even need to recompile

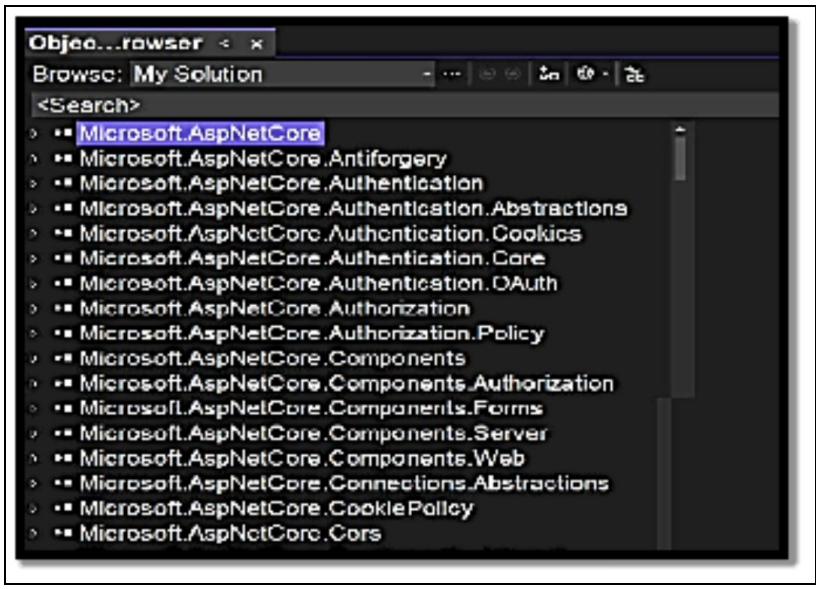
the project to see them.

The image above also shows a row of buttons at the top of the window that we may use to make new folders, move between the chosen items, alter how the window is shown, and perhaps even add a class to a code map file.

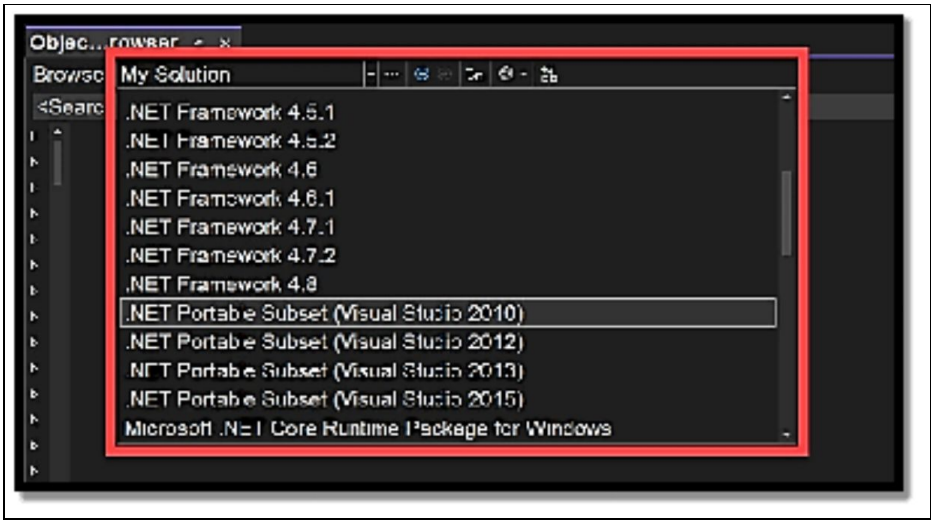
# The object browser

Since the inception of Visual Studio, the object browser has been a very helpful window. This window allows you to thoroughly analyze all of the assemblies that are utilized in your project and offers information about each one. We may do so via **the View** then **Object Browser** menu to get to this window.

The assemblies that are utilized in our solution will open and load once we choose the Object Browser option. We may analyze assemblies that are a part of the framework we are using as well as assemblies that we have written ourselves, as seen in this image, which explains why the list of assemblies is rather lengthy.




Looking at the image above you will see the filter, which is deployable to find which framework or the set of libraries to analyze.




The list looks different depending on the workloads that were picked during the installation of the visual studio

There is also a search engine to look for the search term and you are going to see all of the possible matches including the type of data found.

Finally, after choosing any item from this list, you are going to see another list on the right panel with all the chosen types like the methods, the properties, the structures, the enumerations, and some other types that are available and you are going to see a description of the chosen member.

# Using visual intellicode

With the use of artificial intelligence, IntelliCode, a technology included in Visual Studio 2022, enables you to create code more quickly. Although it was beginning to show some of its potential in Visual Studio 2019, all the capabilities have been included in this edition. It is a tool that has been trained using thousands of well-known open-source projects that are available on GitHub.

While you are writing code, IntelliCode may make pattern and style suggestions, providing you with precise advice based on the situation you are in. This allows you to finish lines of code. Additionally, IntelliCode supports completion in several programming languages and may display the methods and properties you are most likely to use, like C#, C++, XAML,TypeScript,JavaScript, and Visual Basic.

# Whole line completions

Intellicode helps to complete lines of codes and you are going to see the code predictions displayed based on the entries inside the code like the variable names, function names, libraries used within the project, and IntelliSense options used.

**You can see entire hints to complete whole lines in two ways:**

1. When you are writing the code
2. When you are using the intellicode by choosing one of the items in the IntelliSense suggestion list.  after creating a constructor for the weatherforecast class and typing the letter s, the list of the IntelliSense suggestions is going to appear. We can go through them and intellicode shows the suggestions
   When you press the tab key twice it ignores the suggestion.

# Intellicode suggestions

Similar code adjustments in our projects can be carried out with help from IntelliCode recommendations. IntelliCode monitors the code we write and notifies us through recommendations if it notices any code repetition that may be applied to our code.

Unexpectedly, IntelliCode can assist us in identifying modifications that we may have overlooked, including adjustments to formulae, because it is based on the semantic structure of code.

Consider the following scenario: Suppose we have certain methods that enable us to calculate some static values.

```
public float Calculate1()
{
    var minValue = 25;
    return (float)((minValue + 126) * (Math.PI / minValue));
}
public float Calculate2()
{
    var minValue = 88;
    return (float)((minValue + 126) * (Math.PI / minValue));
```

The calculation also has the same pattern and the thing that changes is the min value variable, so we have created a new method called **calculate,** which does the same thing by receiving the parameter.

```
public float Calculate(int value)
{
   return (float)((value + 126) * (Math.PI / value));
}
```

Then we replaced the calculate1 method code in other to invoke a newly created calculate method.

```
public float Calculate1()

{

    var minValue = 25;

    return Calculate(minValue);

}
```

Then after going to the calculate2 method and typing the names of the calculate method, you are going to see an intellicode hint.


The suggestions of the intellicode, make us know that we can also apply the invocation into the new method, that we can apply when we click the tab key, then you can ignore the suggestion and continue to write.

# Review Questions

- What is tellicode?
- What are code maps?
- How do you look for references in code?

# Summary

Some windows and tools in Visual Studio may be quite helpful to us while we work on our projects.

We have shown how CodeLens can assist us in visually and programmatically locating references and connections. In a similar vein, we have researched the various code windows that aid in project class member analysis. Finally, we have seen how IntelliCode, a recent addition to the IDE, facilitates speedy code writing by offering a variety of ideas.

# CHAPTER 8
# WEB TOOLS AND HOT RELOAD

There are a lot of tools that are compatible with the .NET applications and there are also tools for as many programming languages and technologies and they include web development technologies like JavaScript, HTML, and CSS.

In the latest update, there are a few upgrades that make coding faster in the front end. That is the design tool for web developers working with CS, HTML, and JavaScript. Using these tools, you do not require editors or IDEs to finish the activity when you are working with the technology.

So, we are going to be teaching you about some of the web tools and how to use them to simplify when making a common statement. With this tool, you can generate codes automatically, make installation and also specify web version libraries, inspect the JavaScript code, and also refresh the applications to then see the changes when in real-time

## Technical requirements

### Using scaffolding

One of Visual Studio's most useful tools for developers is scaffolding. By just selecting a few parameters, scaffolding allows us to generate code automatically, saving us time.

Scaffolding is a widely used idea in software development and is not specific to Visual Studio. Model view controller (MVC) component code development is typically related to scaffolding. MVC is a well-liked design paradigm for online applications. You must divide the tasks involved in making web apps into three distinct components using MVC:

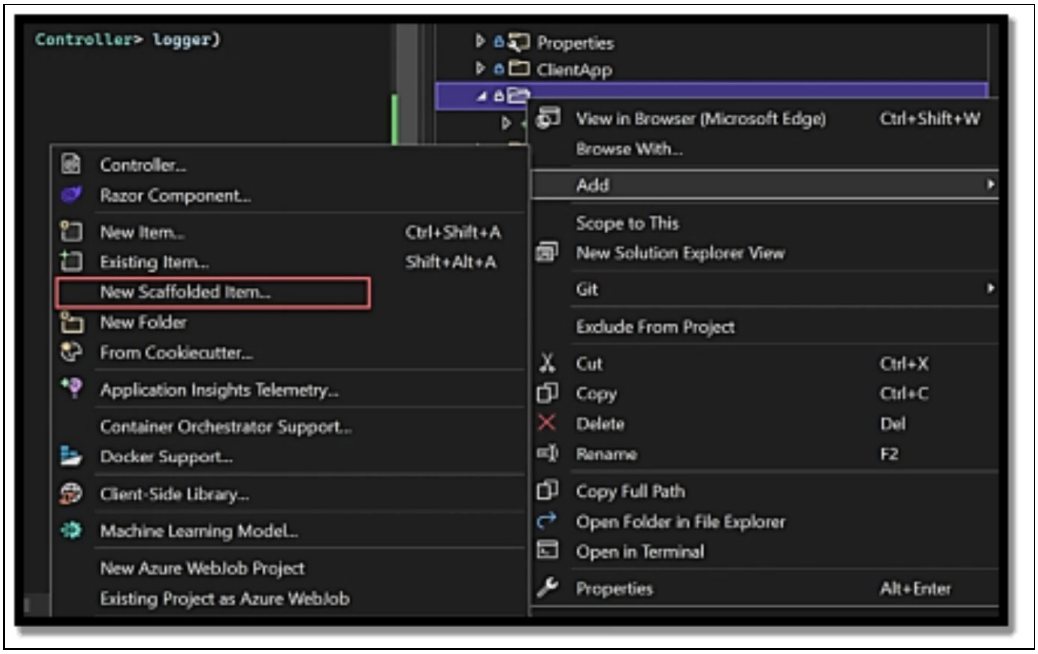
**They are the:**

**View-** this is the interface that the user interacts with

**Model-** this helps to save data

**Controller**- this handle all of the actions that the user performs.

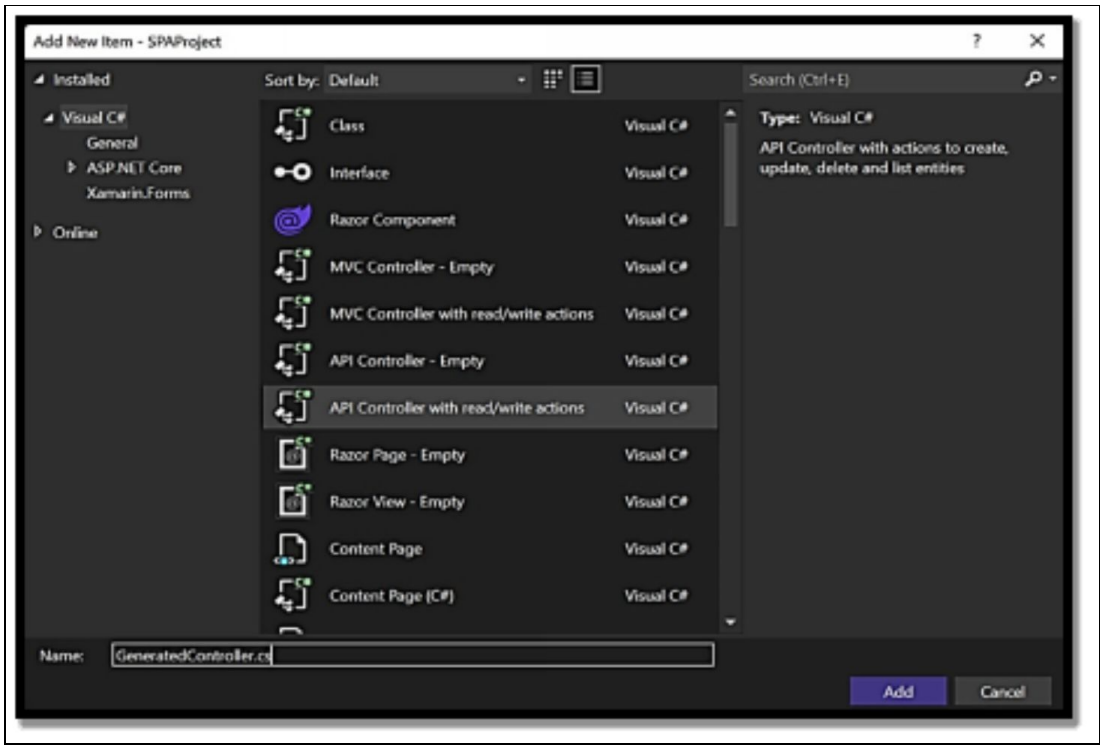
To then use the scaffolding, you need to choose the folder from the **solution explorer** and then right-click to give you the option of choosing a **new scaffolded item.**



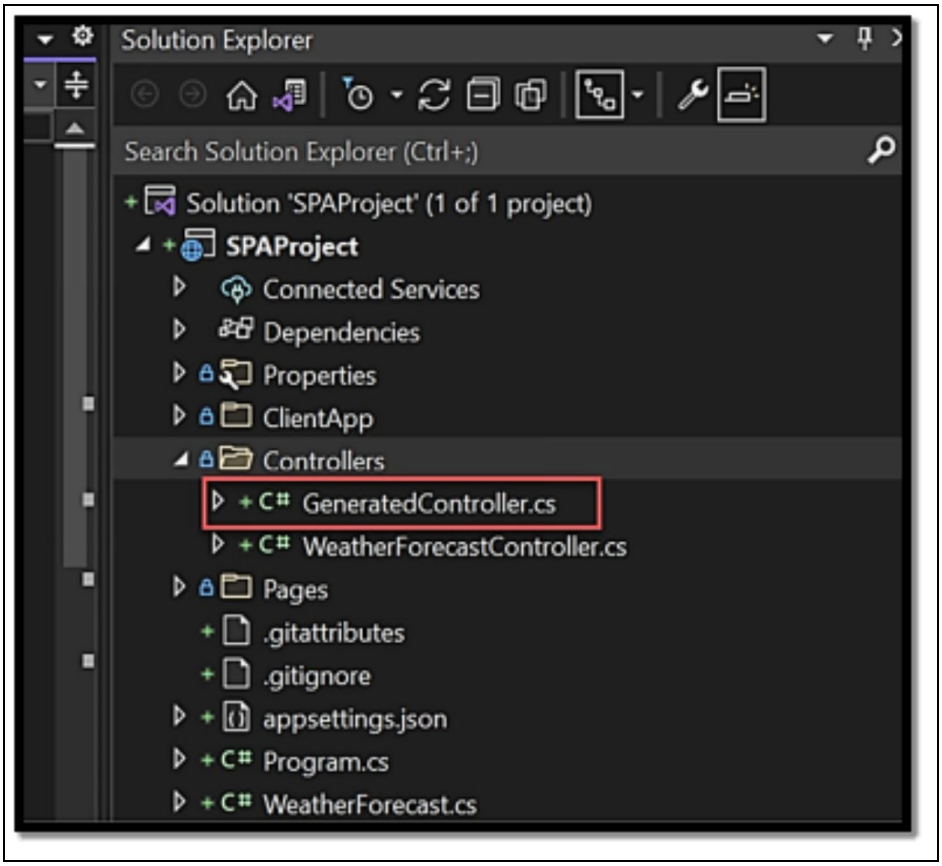

When you make use of this option, you can then create other elements inside the project that is related to the MVC structure

When you click on the **new scaffolded item,** you are going to see a list of elements that you can create with the scaffolding tool.

Then we choose an **API controller with read/to write actions** to generate the API controller using the actions for the GET, POST, and DELETE. then select **generatedcontroller.cs** and select **add**

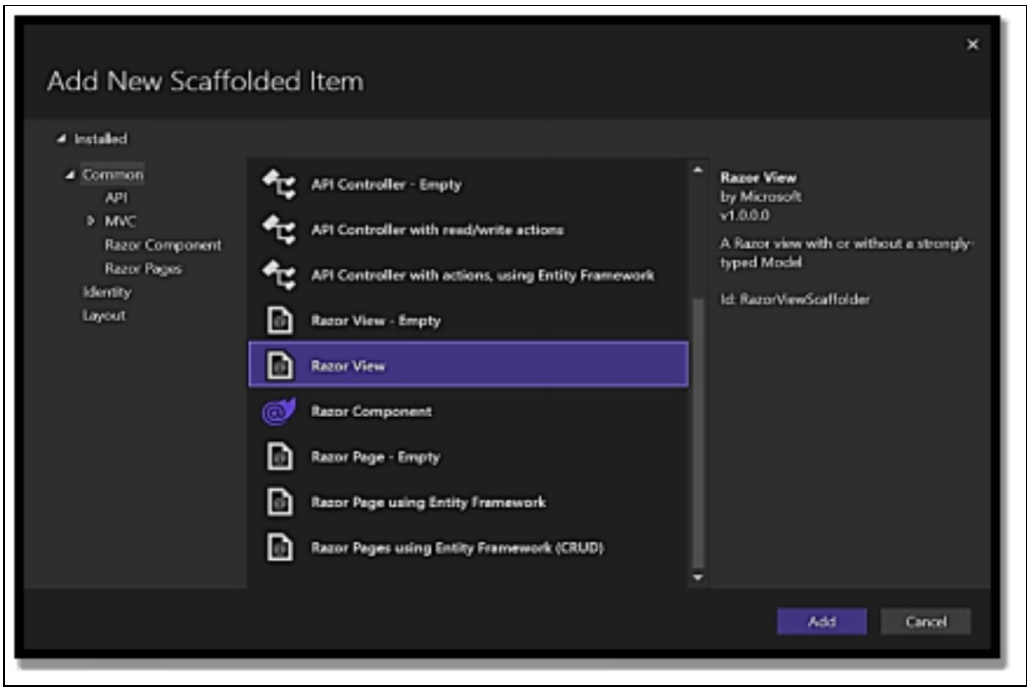



then a new controller is generated inside the controller's folder with endpoints that are by default.

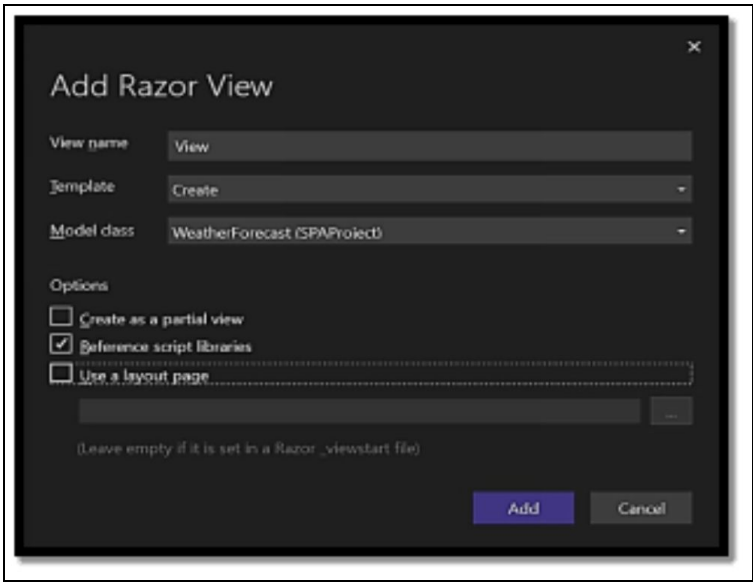



Once you create the base template, then you can now create the data type and a method name that matches the model

We can then create another view page with the scaffolding by going to **add a new scaffolded item** option again and choosing **razor view.**




You can also pick some other templates, and they include **empty.** And in this situation, choose **create,** then uncheck **use a layout page** option and choose **weatherforecast** model.




Then to finish the process, select **add** and you are going to see the new view page inside the **pages folder.** Then visual studio analyzes the model then creates another form for the property inside the model while considering the property type.

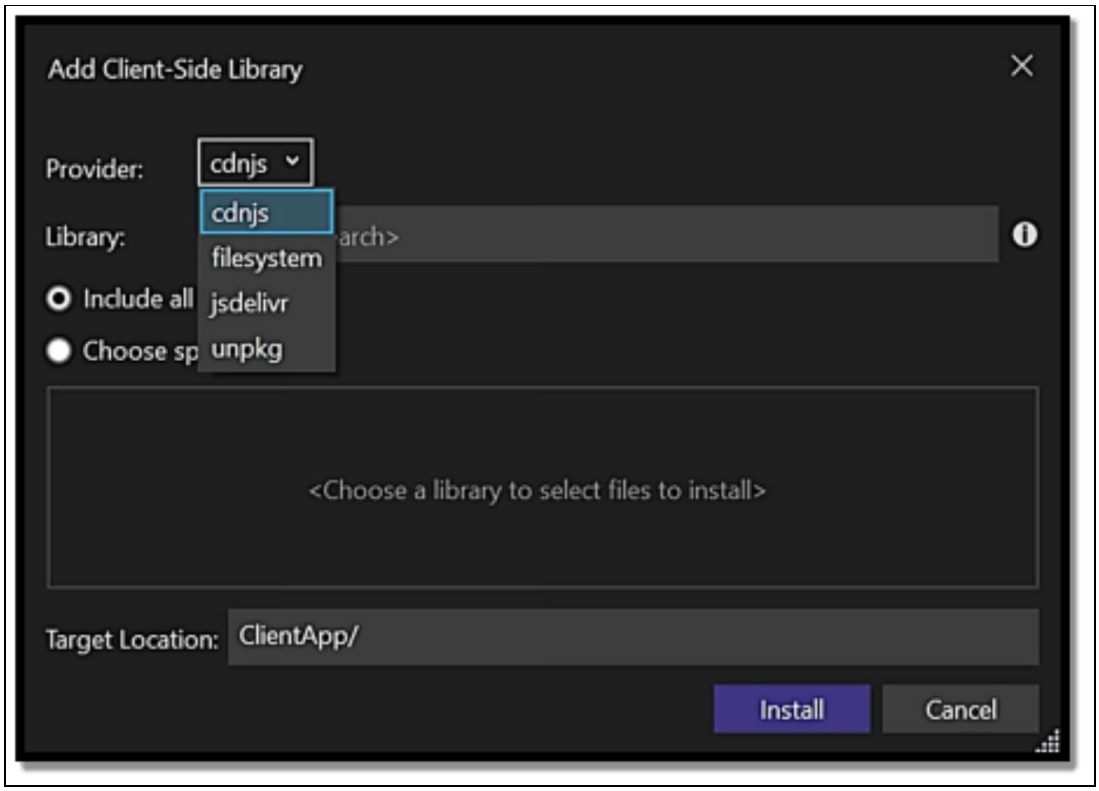
When you go to the view.cshtml file, you are going to see a template used to create a new item from the **weatherforecast** model. Furthermore, the scaffolding adds labels, input, and span to control errors.


# Installing JavaScript and CSS libraries

A proof of concept (POC), demo, or basic project may be quickly created using a template from Visual Studio, but there is a chance that the project will expand in terms of functionality and services. In this case, libraries will be necessary to extend the features built into the main template and optimize and potentialize our project.

You may launch **Solution Explorer** and right-click on the **ClientApp** folder to add a new JavaScript library to our SPAProject. You may choose **Client-Side Library** from the options.

Once you click on the option, you are going to have to get a model which permits you to add a web library from resources. “CDNJS” is the default, however, you can choose another source that is supported by the visual studio.




**There are a lot of different sources that you can choose from and they are**

**cdnjs:** Cloudflare-supported open-source library for quick and dependable content delivery

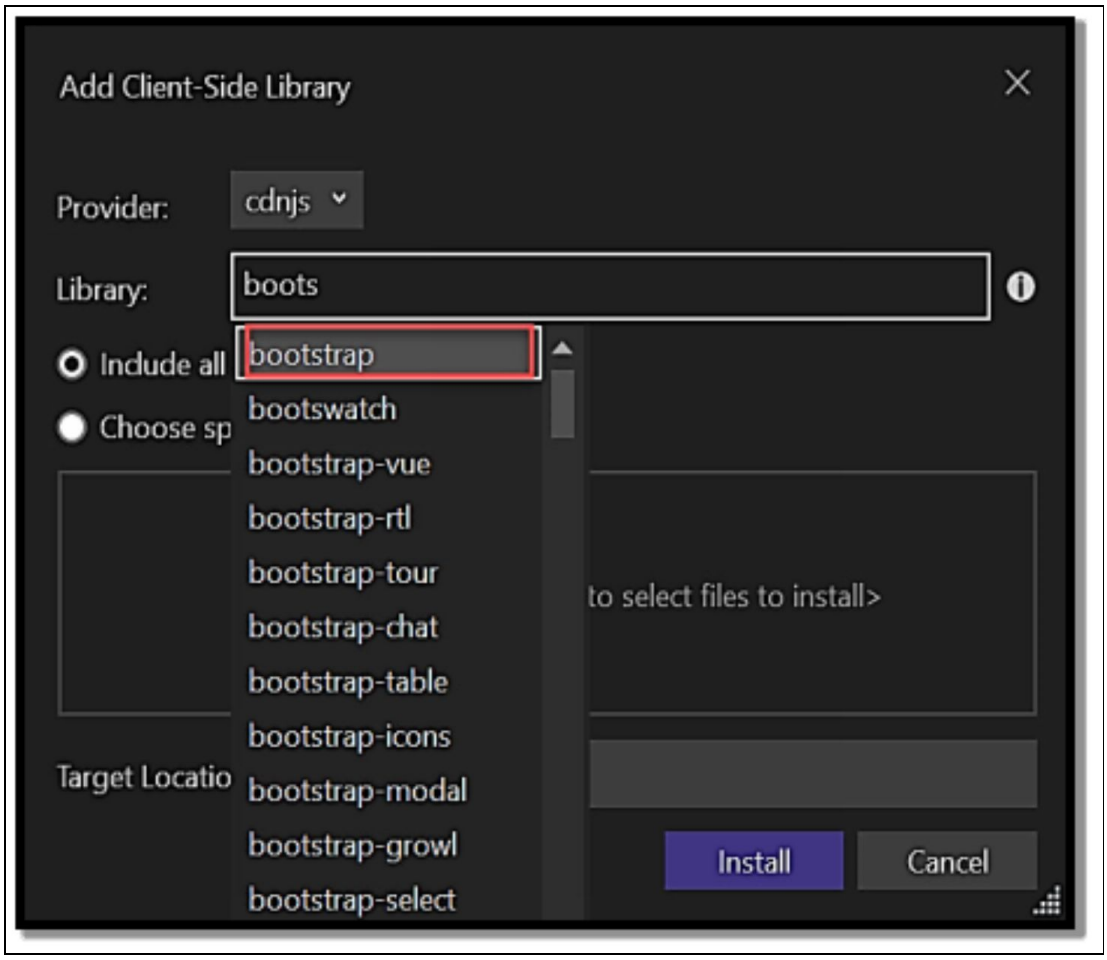
**filesystem:** Local packages created by us

Free content delivery network with GitHub and npm integration is called jsdelivr.

Michael Jackson maintains the unpkg global open-source content delivery system.

To then analyze the way libraries are added to the project, choose, cdnjs and look for bootstrap. Bootstrap creates web interfaces easily with CSS classes.

You are going to get a list of libraries once you begin typing.

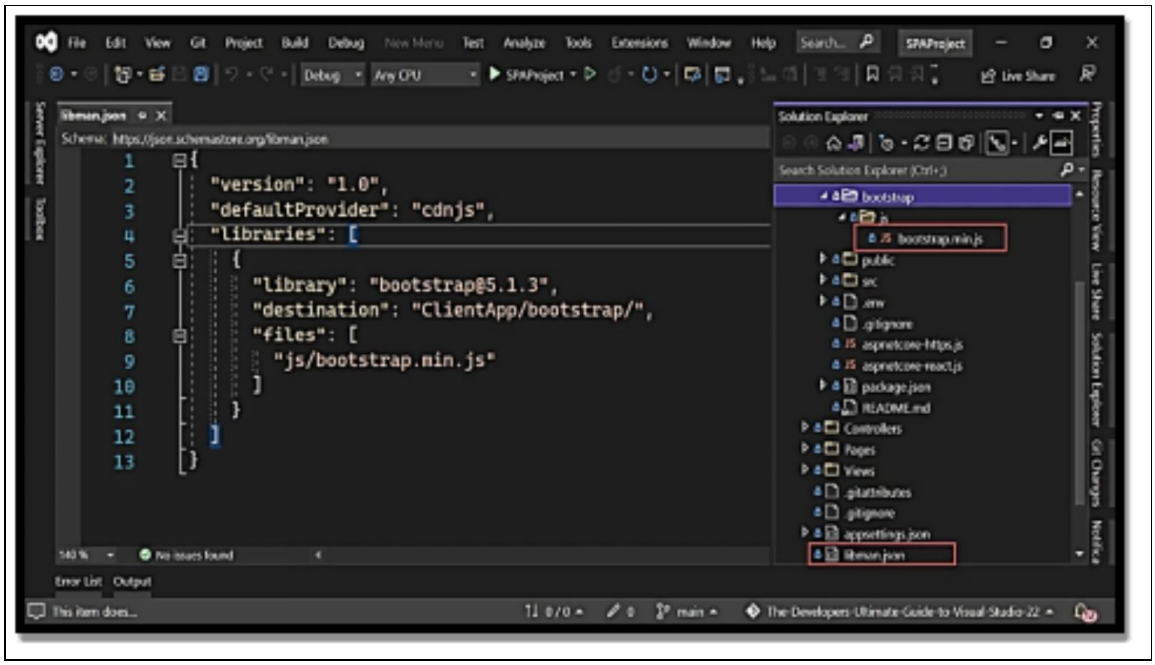



After choosing **bootstrap,** you are going to find the most recent version of the library chosen. And you can then choose all of the parts that are associated with the library. However, you generally use the minify version. Choose only the files you want, with **choose specific files.**

Click Install to add Bootstrap to our project at this time.

After installing this library, you will see a new folder in the ClientApp folder that includes all the Bootstrap-related files.

Additionally, you will notice a brand-new file named libman.json that lists the libraries that Visual Studio was used to install in the project:




When the project's repository doesn't contain the files for these libraries, this file enables Visual Studio to download them from the servers.

The operations will all be carried out by Visual Studio automatically, and it will also produce the libman.json file, which contains information about the version of each library and its project destination folder.

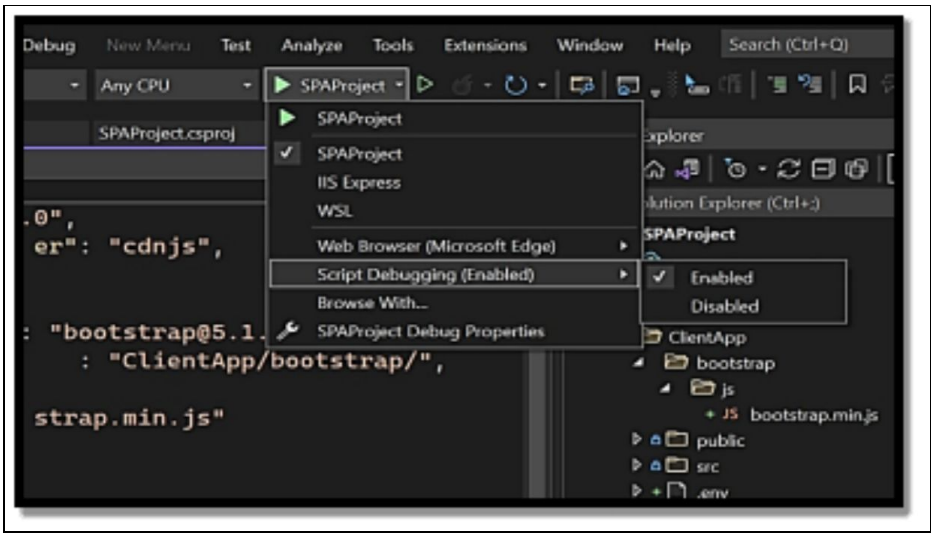
You now understand the various methods allowed by Visual Studio for including JavaScript and CSS libraries in your project. Let's examine how to rapidly identify and fix problems in JavaScript code using Visual Studio debugging.

# Debugging in JavaScript

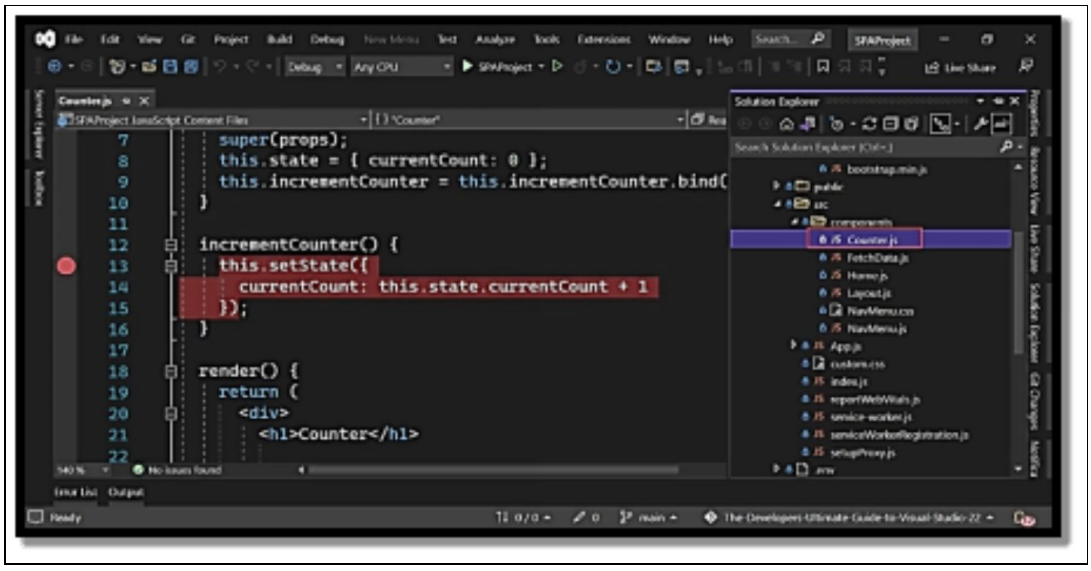
When our application exhibits odd behavior, a problem, or a blocker, we must debug the project. Debugging for several programming languages, including JavaScript, is supported by Visual Studio. Since we can debug the frontend side (for instance, with JavaScript) and the backend side (for instance, with C#) using the same IDE, this is a fantastic feature.

We must choose the Script Debugging (Enabled) option in Visual Studio to debug JavaScript and TypeScript code.

**On the project's execution menu, there is the following option:**




You can then run the project using the debug mode. However, you need to then add a breakpoint to then inspect the code. Go to the **client app** then **src** components and enter **counter.js** to then create a new breakpoint that is in line with the number 13




Then you can begin the project with the option that is in the banner or you can select f5 and look for the computer module and go to the counter module. After choosing **increment,** then visual studio ends the execution inside the JavaScript.

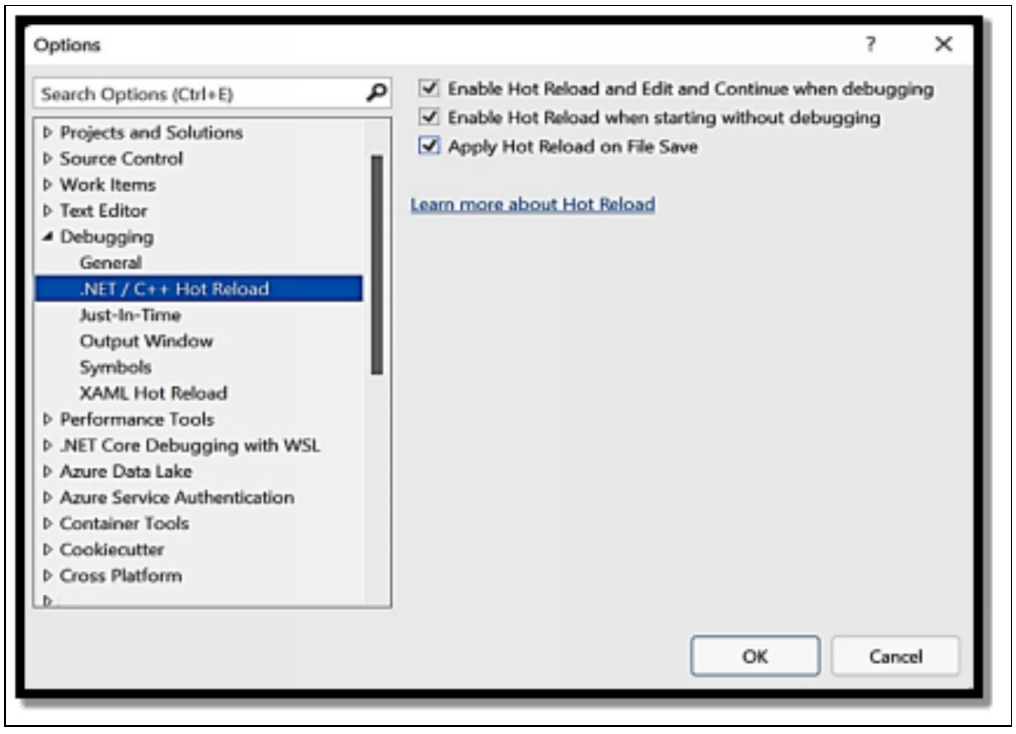
Then you need to inspect the variable in the file

Then finally, you can debug the JavaScript code with the visual studio.

# Hot reload

For a long time, C# programmers yearned for a capability that would let them monitor changes to web applications in real-time. Since C# is a compiled language, this presented a significant issue due to its naturalness as a programming language. To be used by an interpreter, a compiled language must be transformed into a low-level language, which uses up time and resources on the computer. There is a brand-new flame-shaped symbol in this image. You may use the Hot Reload on File Save option to have a web application instantly reload after changes have been saved, or you can click this icon to refresh the changes in the browser after doing so.

We can adjust a few settings for the Hot Reload function to suit our needs. When the Hot Reload button is active, you can access the features by selecting Settings, or you may go to Tools then Options and select Debugging then .NET / C++ Hot Reload.




### Here is how the options work

When in debug mode, this option makes **Hot Reload and Edit** and Continue available.

When this option is selected, a hot reload is made available when an application is launched without debugging.

Apply Hot Reload on File Save: The program will reload after you make changes to a file and save it.

You may launch a project in Visual Studio by hitting F5 or by selecting the **Debug** then **Start Debugging** option from the menu. After that, you may modify the user interface in any way you like.

**For instance, you can go to the NavMenu.js file and rename the NavLink from Counter-to-Counter Module:**

Once you save the file with **ctrl**+s you are going to see the changes.

To leverage this fantastic functionality, we don't need to add a library to the project or set up an extension in Visual Studio. All web applications, including ASP.NET MVC, SPA with React and Angular (like our SPAProject), and Blazor, allow hot reloading.

Now that Hot Reload is available, you may utilize it in your web project to increase productivity when you are writing and need to test the UI changes right away.

# Review Question

- What is Hot reload?
- What is scaffolding in Visual studio and how do you use them
- How do you debug JavaScript

# Summary

You are now prepared to use Visual Studio's web tools to develop more quickly and with higher quality. We can quickly develop MVC model components thanks to scaffolding. A template with straightforward sample code is used by Visual Studio to produce the code.

Using the Visual Studio tools, you also learned how to add JavaScript and CSS libraries. You are aware of how to choose the appropriate library version and upgrade dependencies going forward thanks to these tools.

JavaScript debugging allows you to execute the code step by step while analyzing the values of the variables and workflows if there is a problem or odd behavior in the code.

Finally, you discovered how to leverage Visual Studio 2022's hot reload feature to refresh the application while debugging and viewing changes made to the code in real-time.

# CHAPTER 9
# STYLING AND CLEANUP TOOLS

Code style settings can be specified for each project individually using an EditorConfig file or globally for all code you edit in Visual Studio on the Options page for the text editor. Using the Code Cleanup tools of the Visual Studio 2022 and Format Document Visual Studio 2017 commands, you can also set Visual Studio to apply these code style preferences to C# code.

## Technical requirements

Before you can work with the styling and cleanup tools, you must understand some of the technical requirements

### Working with CSS styling tools

It is great to have the tools that help you to work with the CSS files properly. With these tools editing the files are both quick and easy. There are a few tools available in the visual studio and we are going to be taking you through them.

**Here they are**

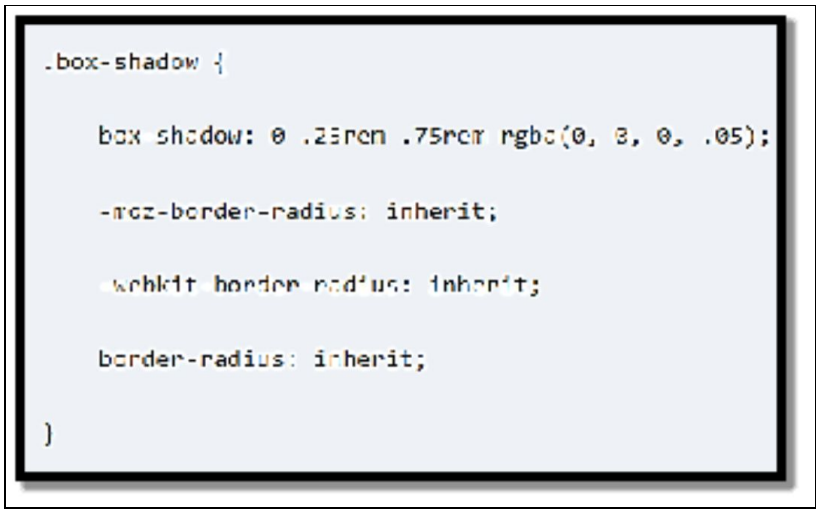
### CSS3 snippets

You can have issues if you are working with different browsers and most of it has to do with compatibility to display different styles. When working with the CSS it looks different. Depending on the browser.

In the visual studio, there are CSS3 snippets that can help you when integrating CSS properties in a different browser and not have to go through the task of writing it out.

We can identify the **.box-shadow style** by opening the SPAProject then entering ClientApp clicking on src then components and select **NavMenu**.css file and looking for it. When we begin entering the phrase "border-radius" within this style, a set of IntelliSense recommendations appear.

You'll notice that this collection has two different sorts of icons, some of which have a blue geometric form and others which have a square white figure. Since these are CSS3 snippets, the white-colored symbols among them are the ones we are most interested in. Once we've chosen the snippet we're interested in, we can scroll through the list using the keyboard's keys by pressing the tab key twice. This will implement the CSS3 snippet, which is cross-browser compatible. Take a look at this.



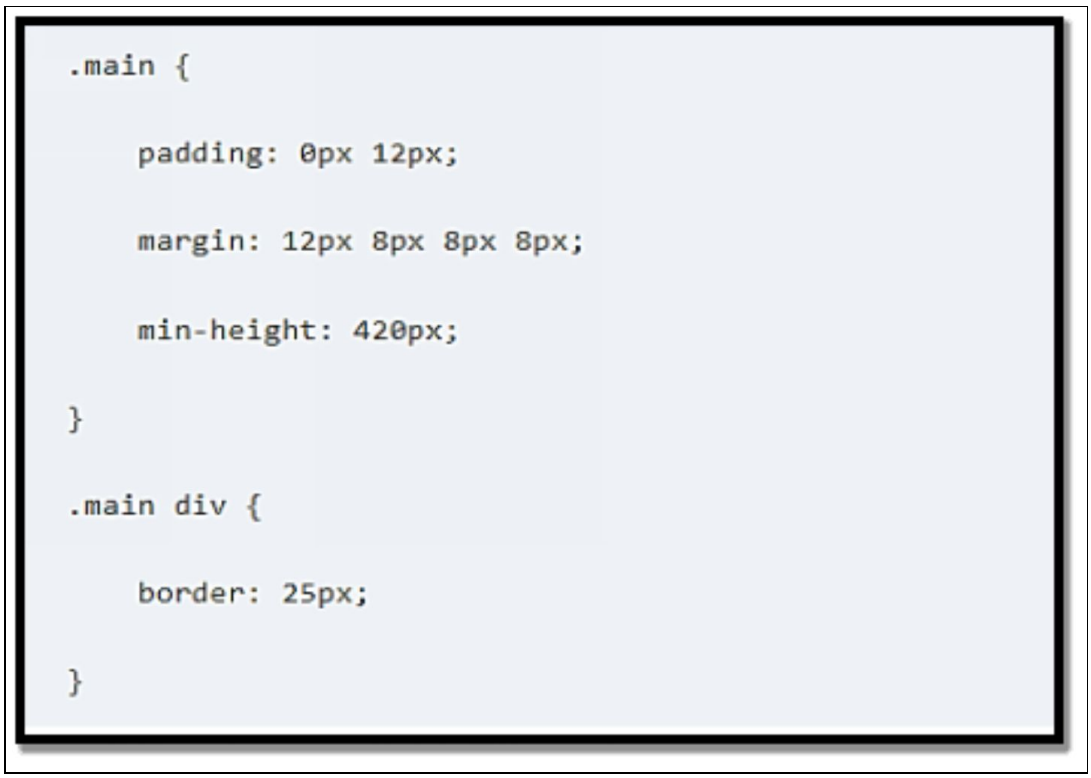

**These are some of the most common multi-browser CSS3 styles**

- Alignment styles
- Background styles
- Animation styles
- Border styles
- Box styles
- Column styles
- Mask styles
- Grid styles
- Transition styles mask styles

# Hierarchical CSS Indentation

The Hierarchical CSS indentation helps to increase productivity well and it shows the style content through the spaces at the beginning of the line and also the sub-styles that belong to the parent style.

With visual studio, you can also create indentations quickly from the style files. If you want to create a style that we call **.main** and also a sub-style that affects all the div elements inside the .**main** style, here is what to do



```
.main {
    padding: 0px 12px;
    margin: 12px 8px 8px 8px;
    min-height: 420px;
}
.main div {
    border: 25px;
}
```

Going by the rules, after writing the styles with the same indentation level you can then go ahead to apply the indentation by entering **edit** and choosing **advanced** then selecting **format document** in other to generate the hierarchical indentation of the whole document just like we have in this code block.

```
.main {

    padding: 0px 12px;

    margin: 12px 8px 8px 8px;

    min-height: 420px;

}

    .main div {

        border: 25px;

    }
```

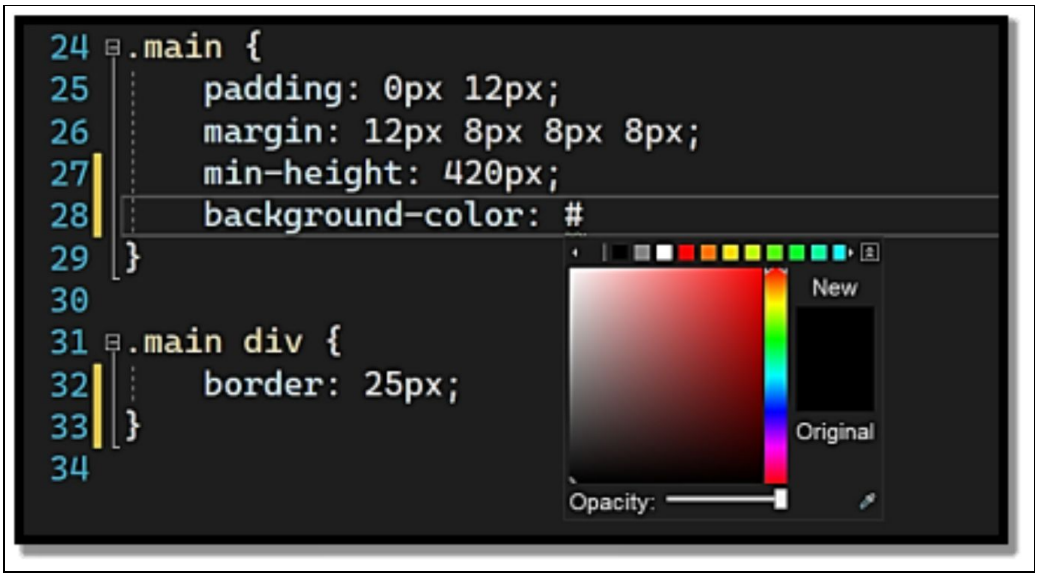
However, when you want to add the same indentation to a set of styles that you have chosen, you can go to **edit** and select **advanced** then choose **format selection**

# Color picker

The option to choose a color for an element is one feature that is quite helpful when dealing with styles. Fortunately, Visual Studio provides a built-in color picker and, although appearing to be quite basic, performs a great job.

Let's test it by making changes to the .**main style** we developed in the section on Hierarchical CSS indentation. When you enter the **background-color**: attribute, a vertical display of preset colors with given names will appear for you to choose from. Enter the # sign instead of this list. You'll notice a new horizontal list of predetermined colors right away.

Then to set the custom color, select the button at the bottom of the color list and it shows the color picker.




Then you can choose the color from the color selection. Then you can switch the color hue, the opacity, or the transparency of the colors that you choose. You can also use the eyedropper, to choose the colors from external sources like an image.

What makes this tool great is that you can store colors that custom chose internally to then reuse them through the CSS files.  if you go to the.Main div styles and you will like to add the custom color inside the color attribute, you can see it in the horizontal colors list the custom color that is listed.

The color picker is just perfect to assign colors.

# Intelligence in style files

You can also use IntelliSense to create a style file.

Here is how it works practically. You need to enter the **client app,** then choose **src** and select **components** then select **navmenu.css.** when you enter the file you have to then create another new style that is called the .intellisense.

The list of all the features that we may add to the newly formed style will appear if we position ourselves inside the style and hit the Ctrl + spacebar key combination. It will begin filtering the list with the matches of what we input once we begin typing the name of an attribute.

Additionally, by clicking the tab key, you may choose an item from the list and finish typing its name.

Let's assume for this demonstration that we must utilize the background property but are unsure of the potential values we may provide it. IntelliSense can assist us visually by providing examples of how to utilize each characteristic.

Also, IntelliSense customizes the results based on the context of the chosen characteristic. Consider the situation when we need to give the font-family property of a style a list of fonts. You are also going to see the list of values that corresponds with the font family.


To then assign values inside the font-weight attribute you are going to see the result based on this attribute.

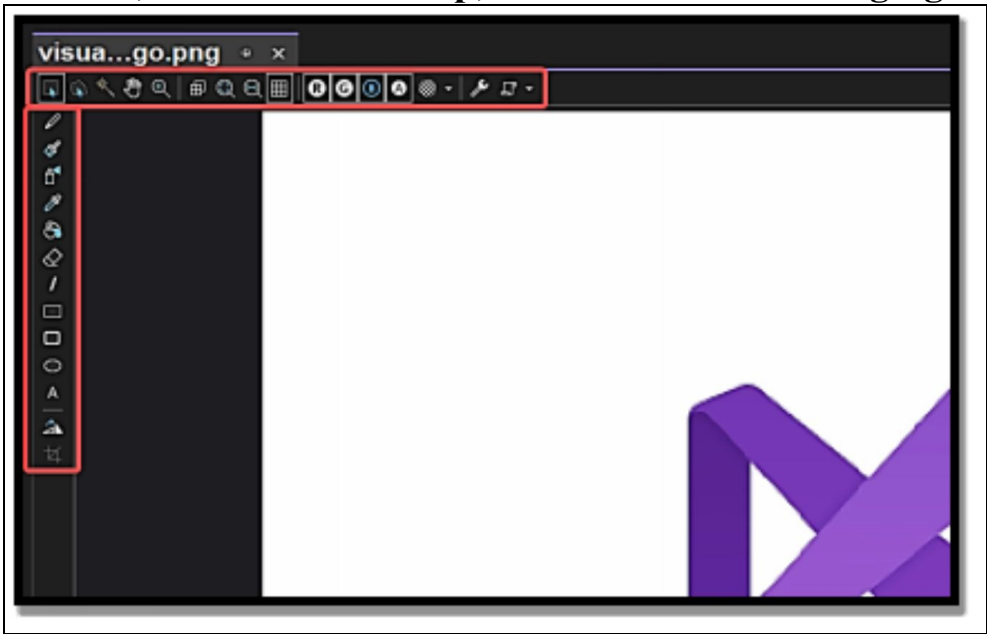
# Working with Images

**The image editor is not talked about enough. You have to install it as they are helpful to do some basic editing of images.**

- You can use them to rescale images
- Change the color
- To rotate images
- To add texts to images
- To add filters.

When we first open an image (in this example, visualstudiologo.png), you are going to observe two toolbars.

**The image editor toolbar, which is on the left, and the image editor mode toolbar, which is on the top, as seen in the following figure:**

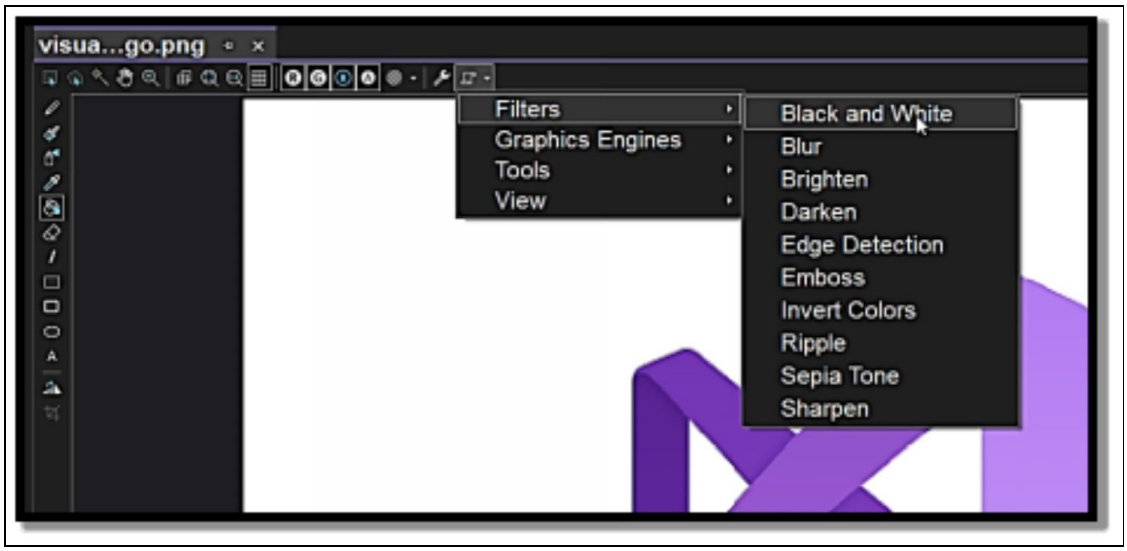



The image editor toolbar can be seen on the left side of the editor and you are going to see the tools that perform specialized actions on images like adding shapes and rotating the image.

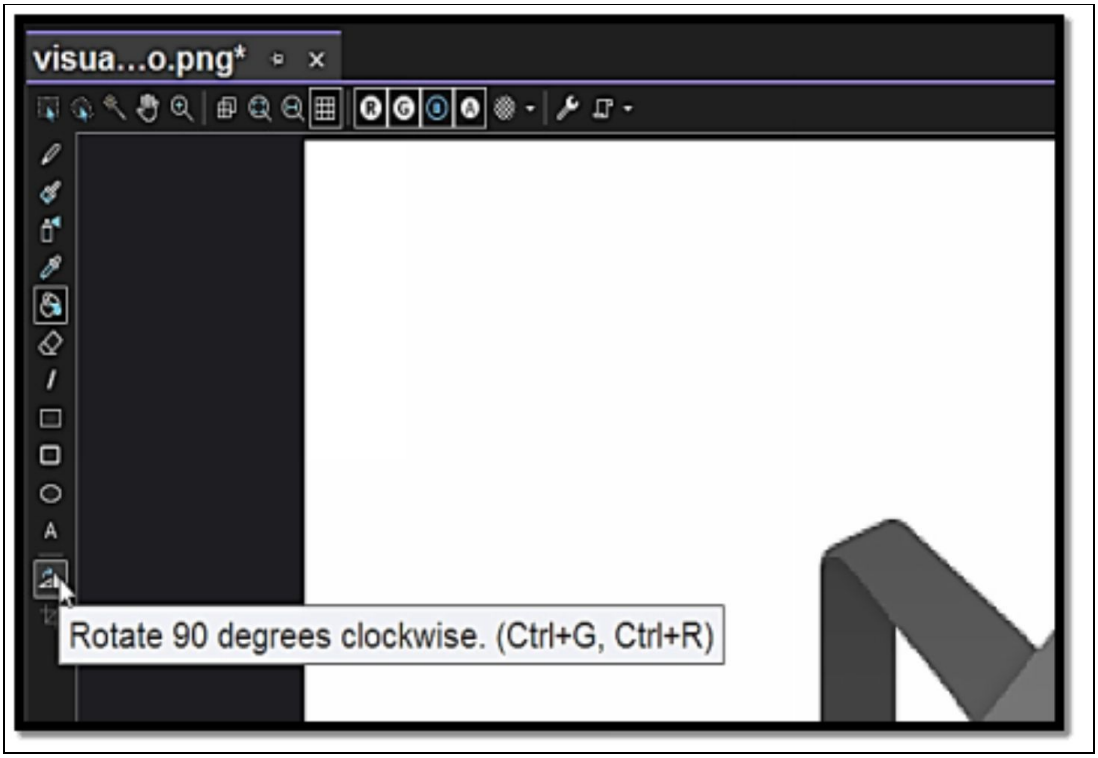
Looking at the top, you are going to see the image editor mode toolbar. In the toolbar, you are going to see buttons that perform the more advanced commands like running irregular selection, panning, zooming, wand selection, and the image properties.

If we want to, for example, convert an image to greyscale, then flip the image and write texts inside the image, here is what to do.

- Go to the image editor toolbar and choose **advanced** then select **filters** then choose **black and white.**




- Go to the image editor toolbar and double-click on the **rotate image** button




- Then finally, choose the text tool then add the text you want to add inside the **properties** window.

You can then save the image so that the changes you made are permanent

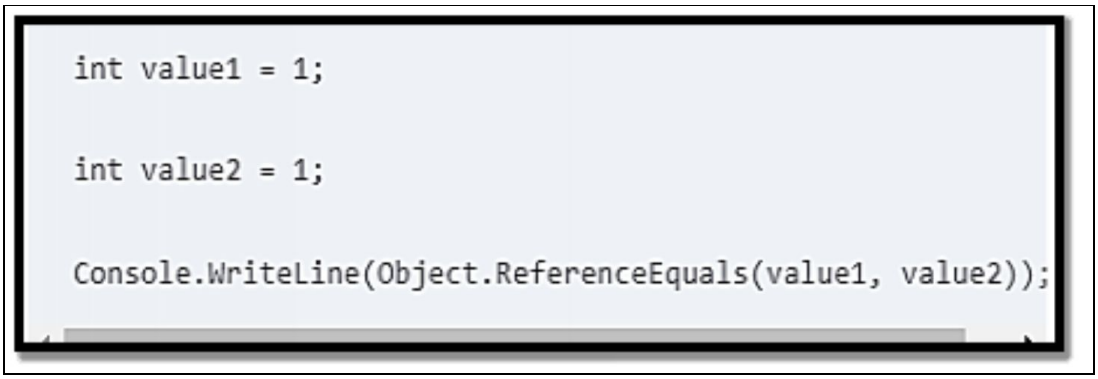
# Cleaning code with code analysis tools

The C# or Visual Basic code analyzers in Visual Studio 2022 enable us to maintain high standards of coding and a unified style in the source code. Projects must be set up with a.NET 5 or above framework version to use this capability by default. Analysis violations will be distinguished from compilation errors by the CA prefix for code quality analysis violations and the IDE prefix for style analysis violations.

## Code quality analysis

Code quality includes, among other things, having secure source code, code that performs as well as it can, and well-designed code. Fortunately, Visual Studio's built-in rules may assist us in maintaining high-quality code.

Now, let us illustrate a violation practically. Go to the program.cs files and enter the following



```
int value1 = 1;

int value2 = 1;

Console.WriteLine(Object.ReferenceEquals(value1, value2));
```

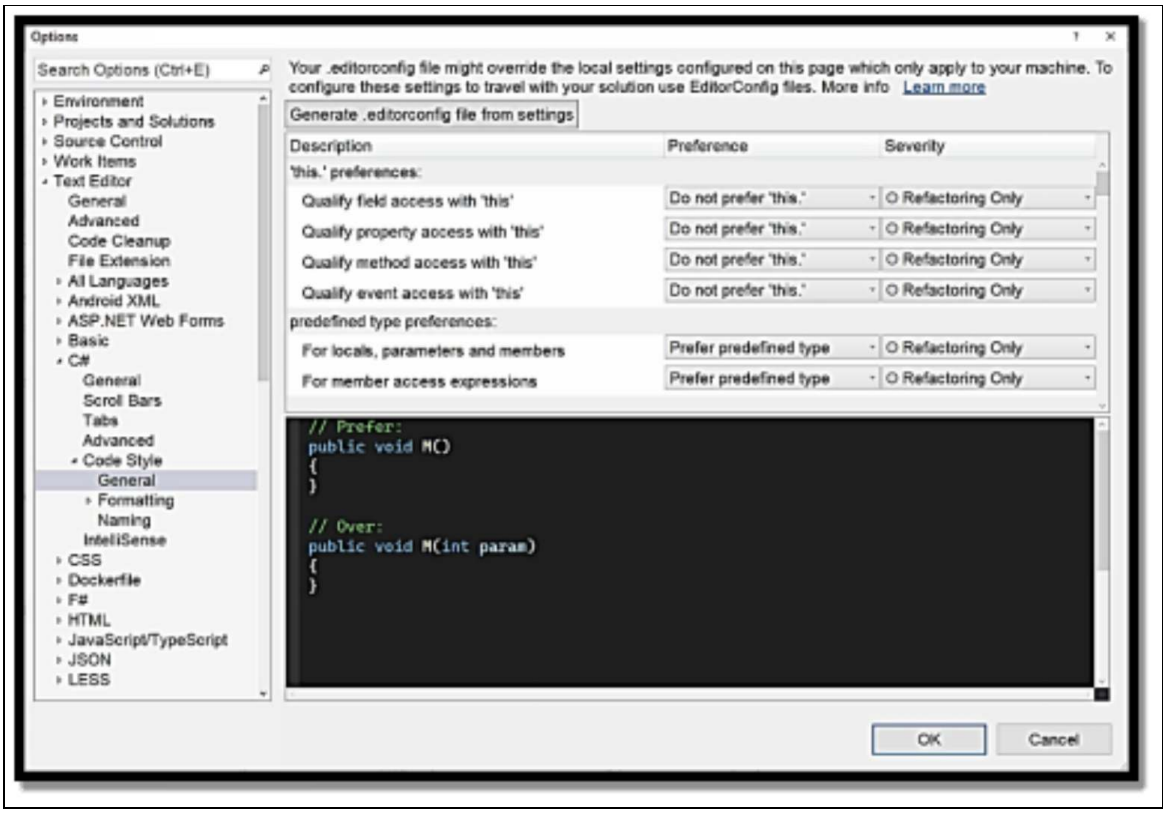
Then you can compile the project by right-clicking on the name of the project and selecting **build.**

We don't observe issues during compilation; but, if you visit the Error List page and look at the list of Warnings, you'll notice several designated with the CA prefix. In our particular case, we can see the CA2013 warning, which informs us not to give a value of type int to the ReferenceEquals function because it will always return a false result because the value **underwent a boxing operation (conversion from a value type to a reference type):**

# Working with code styles

The code styles can be a very useful setting for c# and the visual basic developer to have the right name especially when the project is shared with other members of your team.

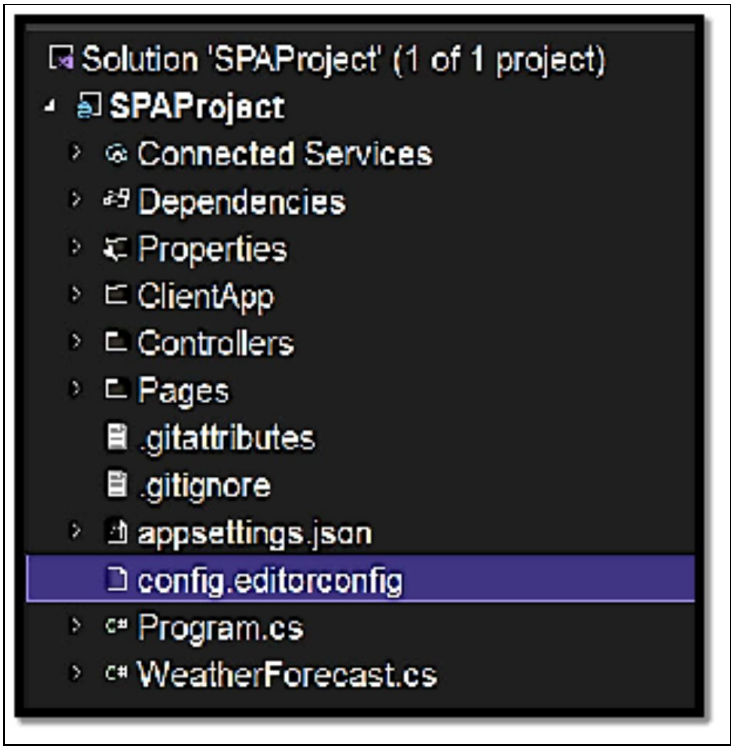
You can create code styles for certain projects. To use it, open **tools,** then select **options** then **text editor** and choose **C#** and select the **code style** and finally choose **general.**




We will be able to examine the code style settings for the current computer when we enter this window. Any of these choices can be modified to meet the code nomenclature we want.

We can change the configuration settings if we need to supply a configuration file that will be used as part of the solution, even if it is opened on a different system. When we have the configurations we want to use for the entire solution, we should click the Generate.editorconfig file from the settings button.

You are then going to see a dialog that has space for the name and the path you want the configuration to be saved. Here we saved it at **config.editorconfig**

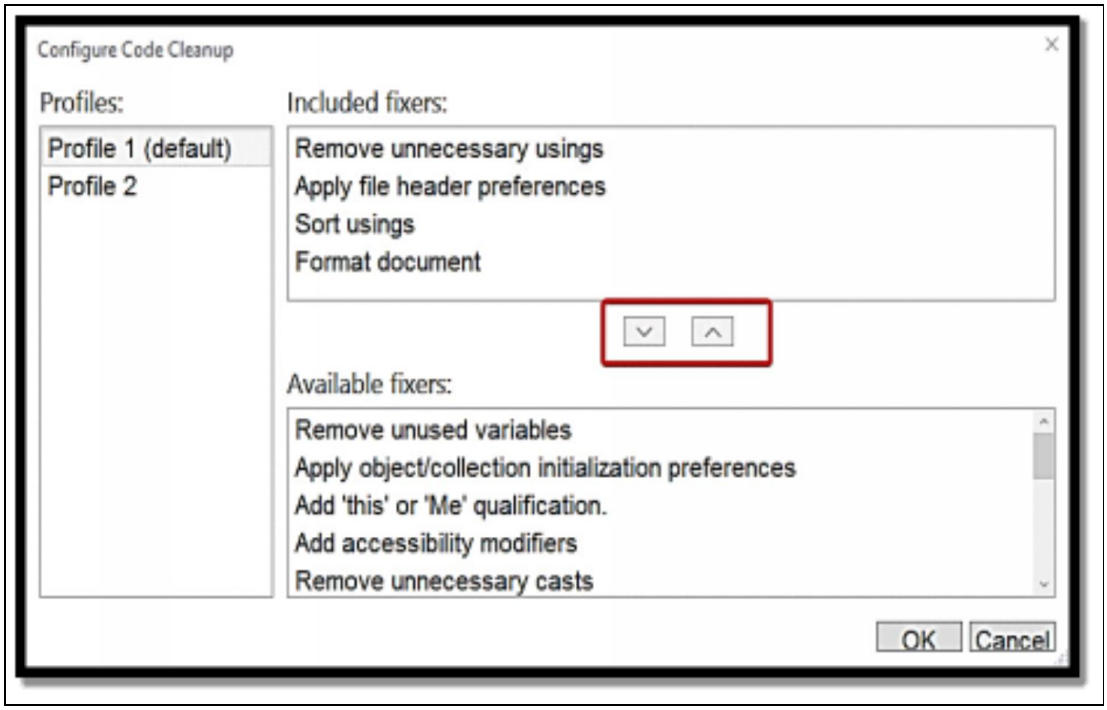



when you open the file, you create the text editor opens then you can see the setting you applied in the text format and you can also change the prechosen parameters. These changes also accompany solutions that are part of the project and have the same name.

# Configuring a code cleanup profile

These are configurations to indicate the type of code cleanup you will like to add. To access the code cleanup profile, here is what to do: enter **tools,**

choose **options** and select **text editor** then pick the **code cleanup** and select **configure code cleanup** menu.




When you notice the image above, you can set two cleaning profiles and profile 1 is executed automatically

# Executing code cleanup

After creating the code cleanup file and establishing fixers for the cleanup, here is how to then execute the cleanup.

Go to the bottom of the editor and select the **broom** icon

This will only apply the rules set up in the active cleaning profile, as explained in the Configuring a code cleanup profile part.

**Let's say, for instance, that the default configuration profile has the setup shown below:**

Then you can configure the **weather forecast.cs** file using this code with the indentation and with unused **using .system.data.common** name.

# Review Question

What are CSS3 snippets?

What are color pickers?

What is intelligence?

# Summary

The many tools that Visual Studio offers for frontend and backend developers were covered in this chapter.

We discovered how CSS3 snippets may facilitate the speedy creation of cross-browser-compatible styles. Similarly, hierarchical CSS indentation aids in maintaining the readability of styles. Additionally, the color picker may help you rapidly choose colors, IntelliSense is available when editing CSS files, and the image editor offers helpful tools if you need to make simple changes to your photos.

In the case of code analysis, we have discovered how to code style analysis aids in maintaining a nomenclature whether we are working solo or in a team. Code quality analysis aids in having secure and dependable code.

# CHAPTER 10
# PUBLISHING PROJECTS

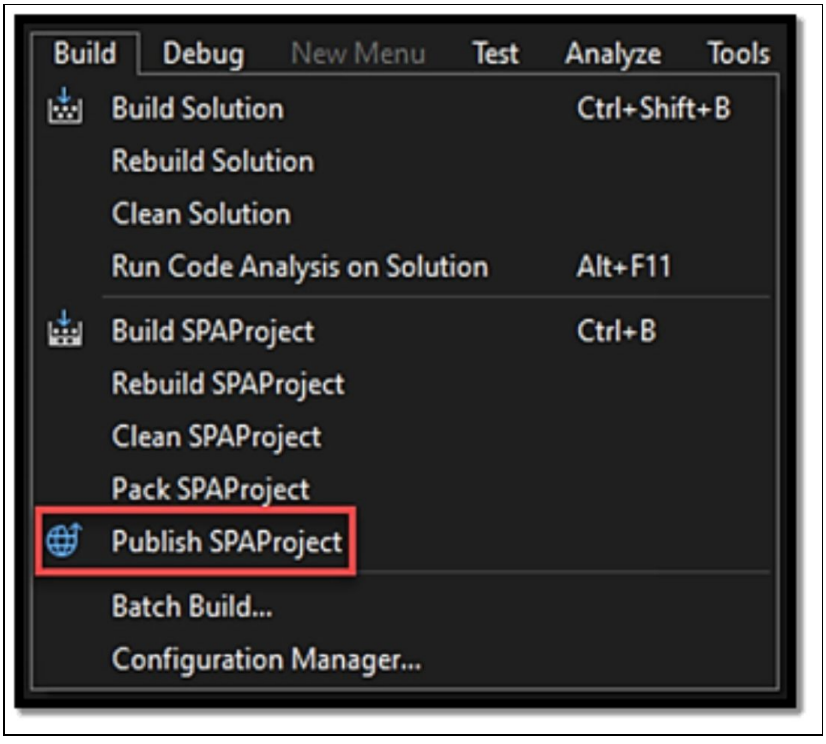
Once you are done with the proof of concept of the minimum viable product, you are going to have to launch changes to see the way the project works in a real scenario and then share the published projects with customers. There are a few tools that help you to publish these projects either from the environment or from the cloud.

Here we are going to be taking you through the ways to publish projects saving time and making publishing simple.

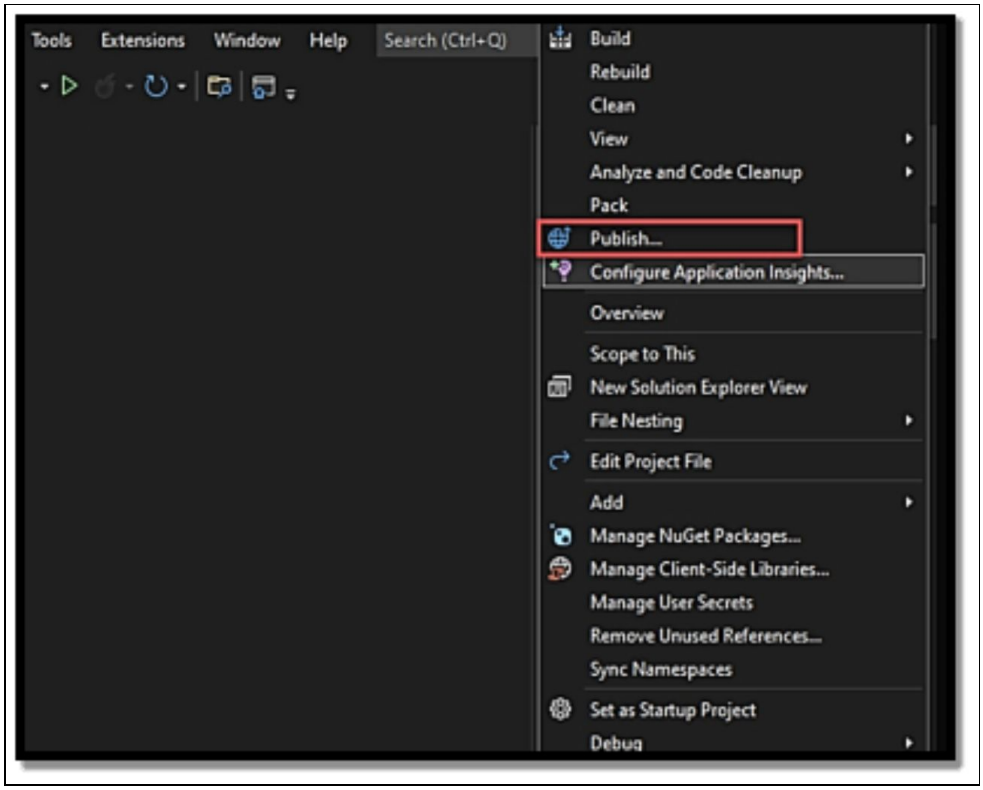
## Technical requirements

### The option to publish a project

There are two ways to publish projects. From the menu and the solution explorer. From the menu, go to **build** and select **publish.** And the main project inside the recent solution is deployed automatically.



Then you can right-click on the projects you will like to publish from the **solution explorer.**




Then after you publish, you are going to see a modal window that shows you the deployment types that are supported for the main project inside the solution or the project that you chose. The options are **azure, docker container registry, folder, FTP/FTPS server, webservers, and import profile.**

We are going to be taking you through some of the popular options for publishing.

# Publishing in a folder

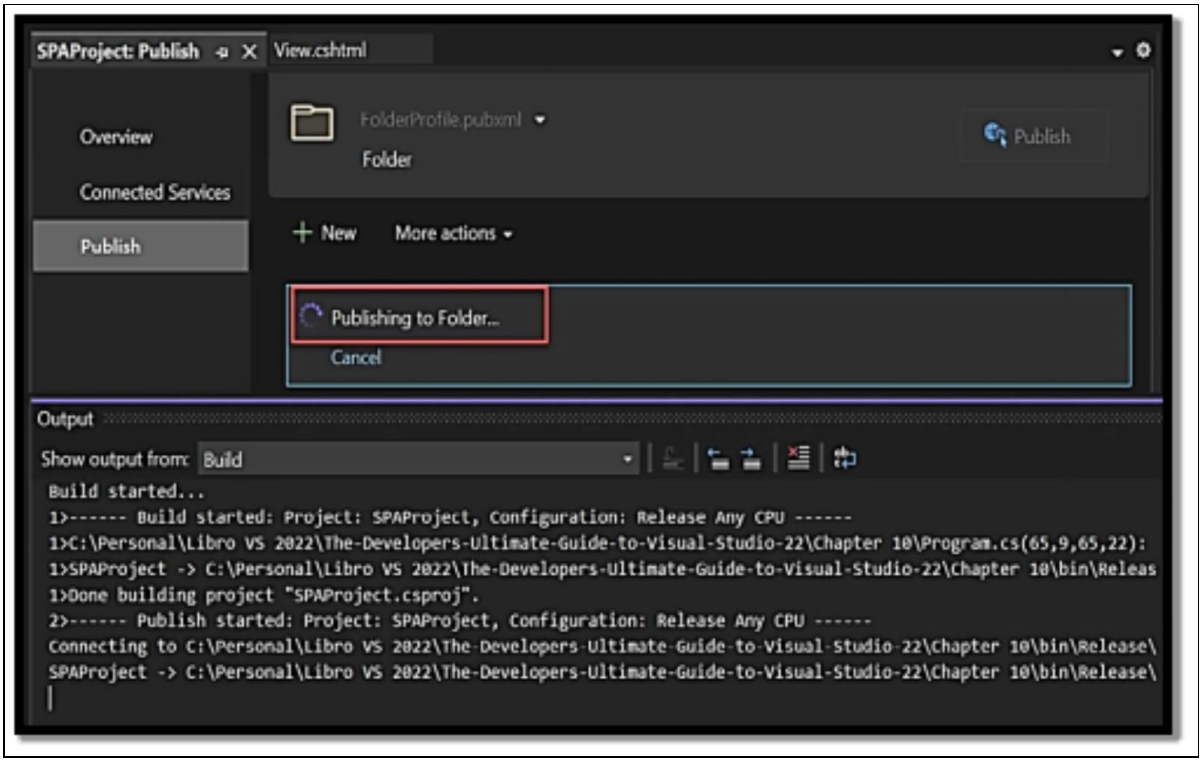
Publishing on the folder is a very common mode of publishing. You use the site system to save inside a folder and include all of the resources to use in local servers like IIS, NGINX, or Apache. It is easy to publish this way.

Go to the publishing function using the steps we enumerated above and choose the folder and choose **next**

In the following window, choose the folder you will like to save the project that you published using **browse.** There are absolutes and relative paths. Then to conclude, select **finish.**

The choices we previously selected are contained in a file with the. pubxml suffix that is created by Visual Studio when the publishing setup is complete. The project may now be published in the chosen folder by using the Publish button:

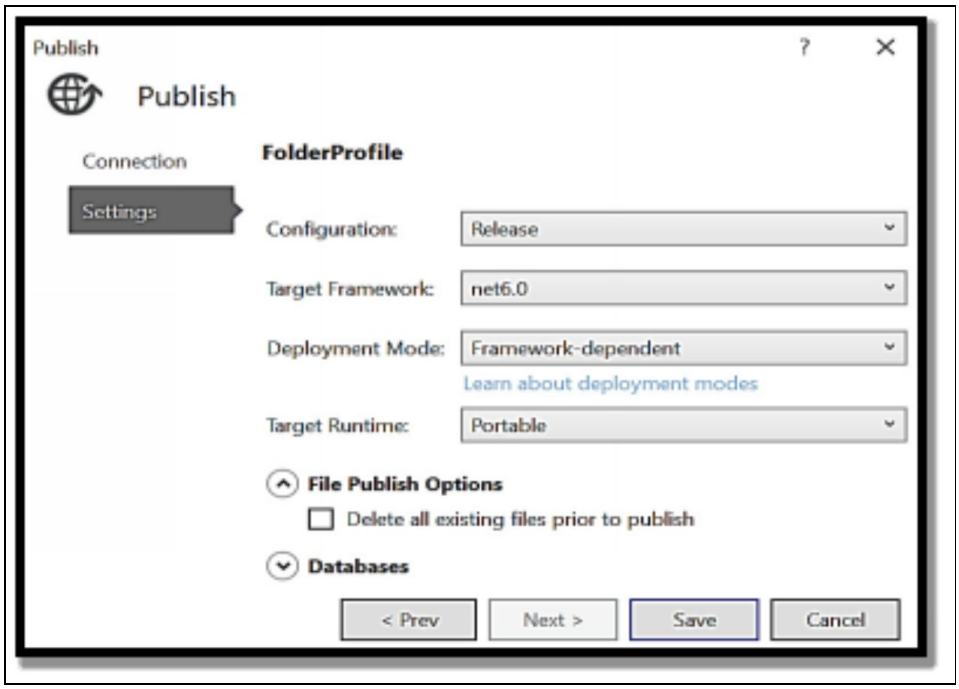
Visual Studio will display a Publishing to Folder... dialog and a console log once you click "**Publish**," where you can see all the steps that were taken throughout this procedure. In the image below, you are going to see it in **output**




then when you wait, you are going to see the confirmation message that tells you if the publication was successful or not.

If successful, in the image, below you are going to see the project publication on the right.


Then select **delete the existing file** and you are going to see some extra options we can change based on our needs.




**Confirmation:** there are two options:debug and release. Debug is better for production

**Target Framework**: Usually, the same version is used in the project, but we have the option to execute our project on a different version.

**Deployment Mode**: We have two options: Framework-dependent, which specifies that the framework must be installed on the server, or Self-

contained, which specifies that the framework and its project dependencies are included in the publication.

The system architecture and operating system where the program will be released are known as the **target runtime.**

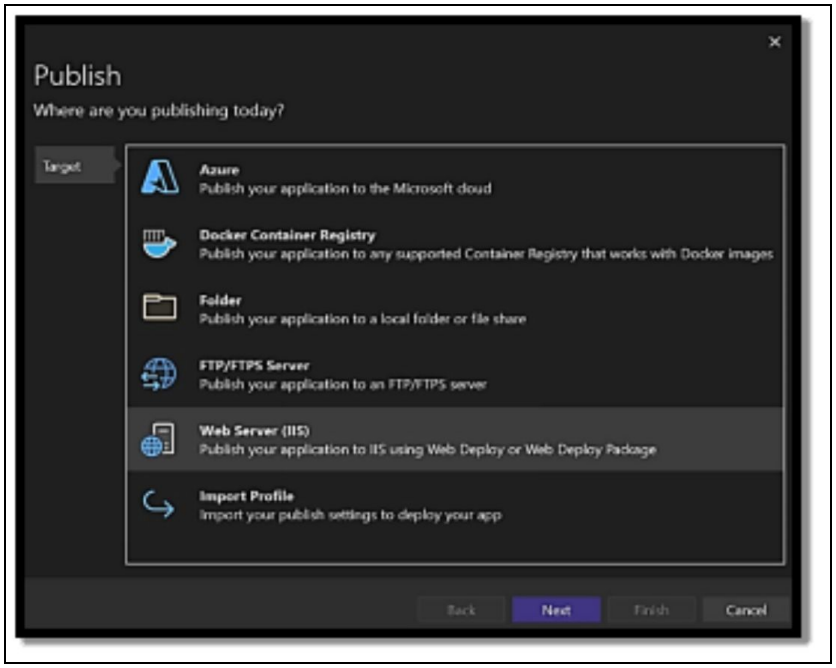
Remove all current files before publishing: By doing this, Visual Studio can remove all current files from the folder that is being used to publish the project.

# Publishing in IIS

This is another popular server when windows are concerned and you are going to see it in all of the Windows servers. You can also install it.

Here is how to publish web apps in IIS.

You need to use the **publish** steps we enumerated earlier then choose **Web server.**




Then choose **next** so that it takes you to the following window.

**There are then two options:**

**Web deploys:** this deploys folder and files in the IIS

**Web deploy package:** starts a zip file and includes all of the files inside the publishing folder.

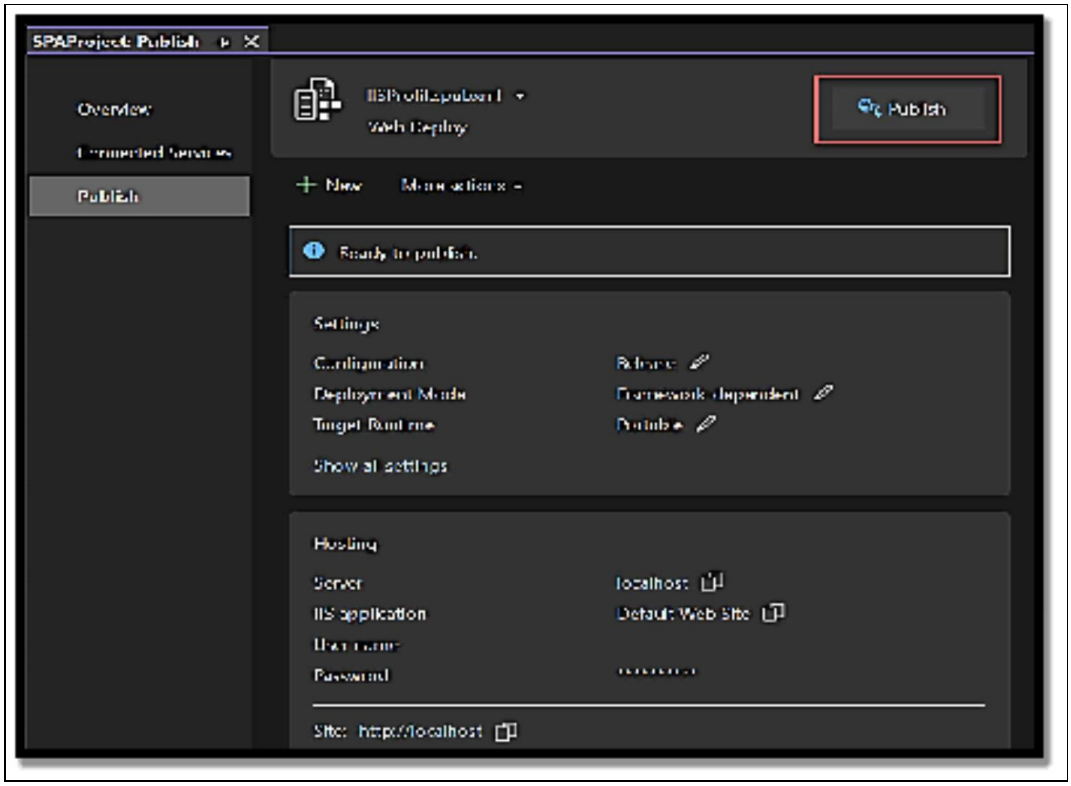
After choosing **web deploy,** you have to enter the name of the server, the site, and the **URL destination.**

We'll make use of the IIS default website. You can create a different site name in IIS if you currently use this site with another application.

You may determine whether Visual Studio can construct the site and complete the publishing using the configuration by using the **Validate Connection** button. The setup may be finished by pressing the **Finish** button.

To run this successfully, you have to execute the visual studio as the admin so that IIS can have access.

Once you are done with the steps above, you are going to see **a ready-to-publish** message and the option to edit the settings. Then you can begin to use the **publish** button.



By default, visual studio opens a new window inside the browser with the URL of the site setup in the first configuration inside the **destination URL** visual studio then opens the http://localhost

And you have deployed the web application to IIS with visual studio.

# Publishing In Microsoft Azure

You can also publish to Azure and it is popular amongst start-ups and .NET developers. To then deploy the project to Azure, first choose the publish option with the methods that we have reviewed above and choose **azure.** Then choose the **azure app service.** You have to remove the .pubxml file inside the project to create a new configuration

**There are a few options to deploy the azure**

Publish the project in a **platform as a service (PaaS)** container using Windows as the operating system using Azure App Service (Windows).

Publish the project in a PaaS container running Linux using **Azure App Service (Linux**).

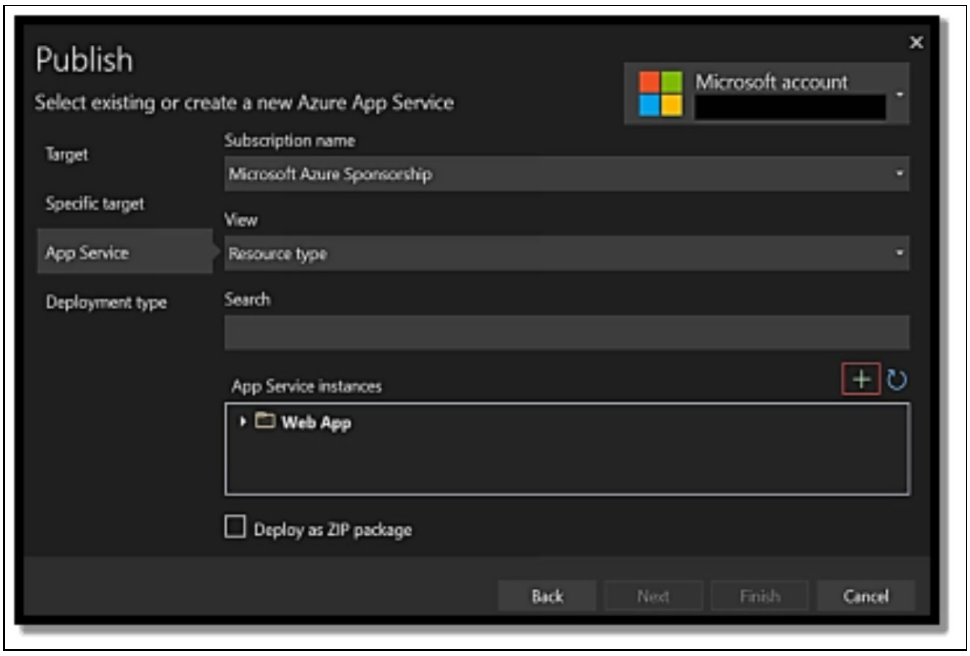
Run a container in an Azure App by utilizing an Azure App Service container.

Create a Docker image of the application and publish it to Azure Container Registry.

Azure Virtual Machine: Manually launch the application using IIS after publishing a folder in a virtual machine.

Then when you choose **next,** you are going to get a new window to perform authentication inside **azure** and then connect it using services and the resources inside Azure for the account.

To proceed with the setup and link Visual Studio with the resources in our Azure account, we notice a new window where we must authenticate with Azure using the Sign In... button. Using the green cross button (s), we can choose the subscription and launch a new web app instance after completing the authentication.

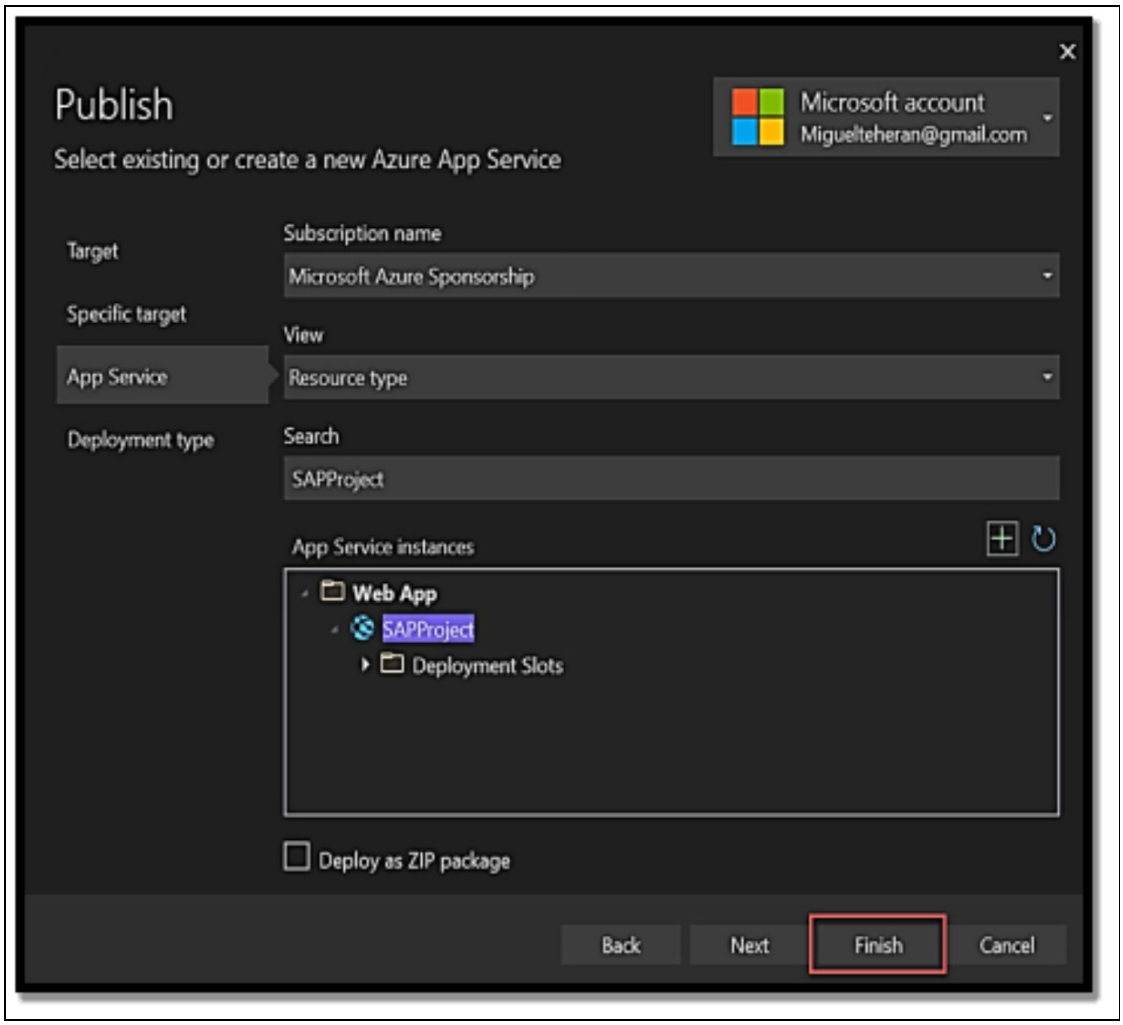



Then choose a hosting plan for the SPA project

If you have a free plan, you have a limited time to run the app

Once you create the hosting plan, you can choose a name and the resource group for the project.

Then you are going to see a preview of the created project. When you choose the **finish** button, you are done




To finish the integration with Azure there are two options.

Publish (generates pubxml file): Produce the configuration in a.pubxml file. Using the Publish button, we must manually publish the project.

leveraging **GitHub for CI/CD Workflow actions (creates a yml file):** This option generates a.yml file with continuous integration (CI) and continuous deployment (CD) options to deploy the project after each change in code if we have our project on GitHub.

In this example, choose to publish and select **finish.**

The .pubxml file is created and the **publish** button becomes visible. Select **Publish.** And the visual studio will open the site.

**Review Question**

- How do you publish a project?
- How do you publish in Azure?
- How do you publish in IIS?

# Summary

You are now familiar with how to use the build menu, the project menu, or the Publish option in Visual Studio. You can now deploy your web application on the filesystem using the Folder option, then host the program on a server. Additionally, you are familiar with utilizing Visual Studio to deploy a project in the IIS server. You learned how to publish a project in Azure using the pubxml file's publishing configuration and the Windows app service option.

# CHAPTER 11
# IMPLEMENTING GIT INTEGRATION

If you want to have control over your project, you need to have a platform to change the controls to develop software on visual studio.

There are a lot of version systems. However, Git is the one that is used more commonly and that is why IDES adds tools that can help one to manage repositories through the technology natively.

We are going to be learning how to manage the GIt repositories in this chapter using GitHub.

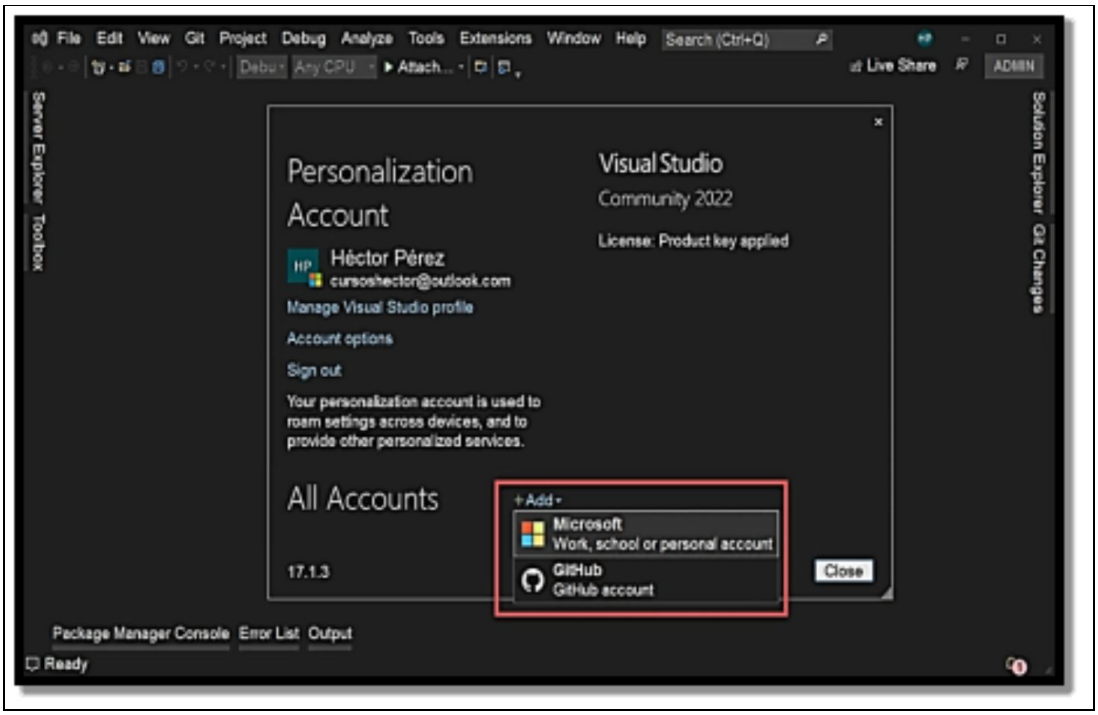
## Technical Requirements

### Getting started with GIt settings

It is easy to work with GIt tools when you are working with the visual studio since it comes in by default when you are installing. This means that you can work on the project immediately.

If you want to have access to the code projects that are hosted on GitHub, you have to first sign in to the Microsoft account, and when you log in choose your profile icon and choose the **account settings** option

You are going to then see the window to customize the account. It is from here that you can then add GitHub with the **add** button




When we click the button, the GitHub login gateway will be routed to us, where we may log in using an already-existing GitHub account or create a new account if we don't already have one. It is crucial to click the button that reads "Authorize GitHub:" because after successful login, we will be prompted to grant Visual Studio permission to communicate with GitHub services.


Next, the GitHub account is going to be added as part of the account that is associated using the main visual studio account so that we can begin to work with the repositories of GitHub

# Creating a Git repository

Creating repositories using GitHub is quite easy. Using a previously created project, choose **Git** and select **create Git repository** menu and enter the information for the repository.

**Local path**: This determines the location on the local computer of the source code (often the location where the project was created). gitignore template: This feature enables you to choose a template that defines a collection of files that will not be uploaded to the repository. For instance, the template you choose may pick files that are created following a compilation and can be generated again on a different machine.

**License template:** You may select a license for the repository code using this template, which specifies what users can and cannot do with the project code.

**Add a README**: By doing this, you may include a file that explains the goal of the repository.

**Account**: You may choose here the GitHub account you want the repository to be published to. If an account hasn't already been attached, you can do so from this page.

**Owner**: If a GitHub account is a member of multiple work teams, you may choose which account should control the repository using this setting.

**Repository name**: Although the project's name is often used, this feature allows you to change the repository's name. The final repository's URL will be impacted by this.

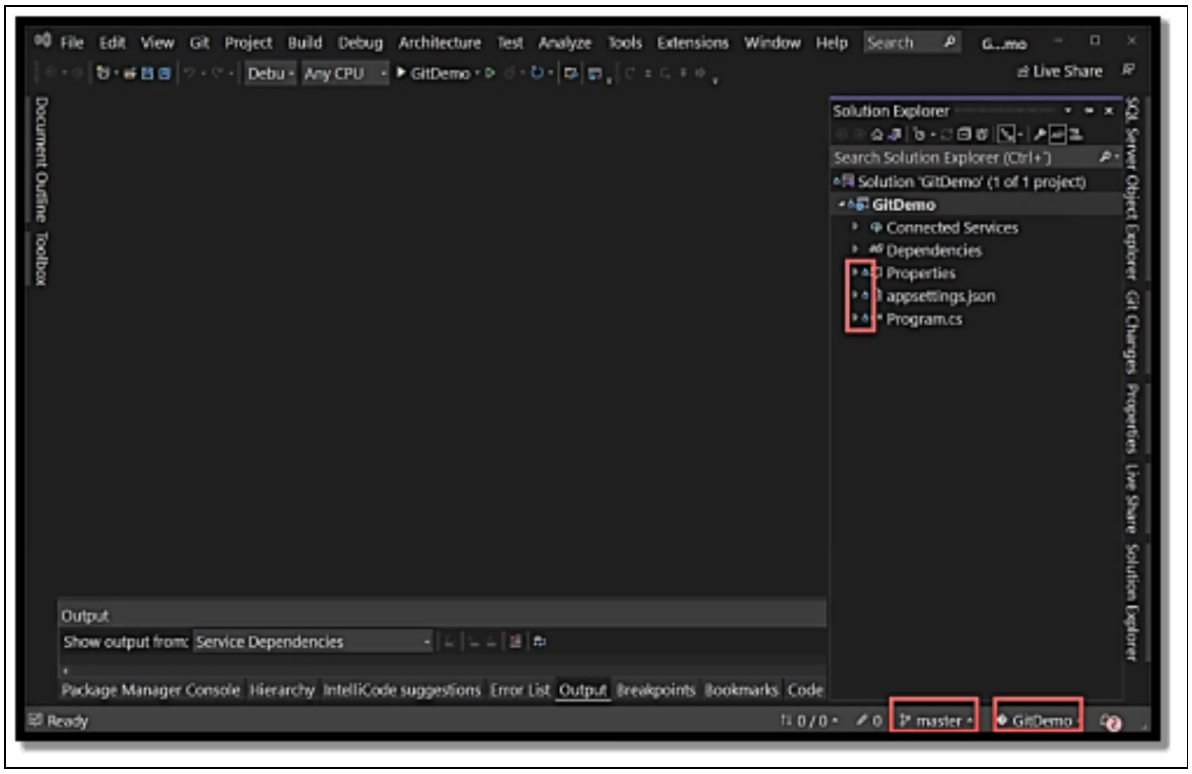
**Description:** You may describe the newly established repository using this.

**Private repository:** You may choose whether a repository is private or public with this feature.

To begin creating the repository when these settings have been configured, just click the Create and Push-button. You must complete this step in your GitHub account to complete the tasks in the subsequent sections.

In the image below, several indicators are indicating a project is a part of a Git repository.

**For instance, source files with lock icons on the left side of them are the files that make up the repository itself:**



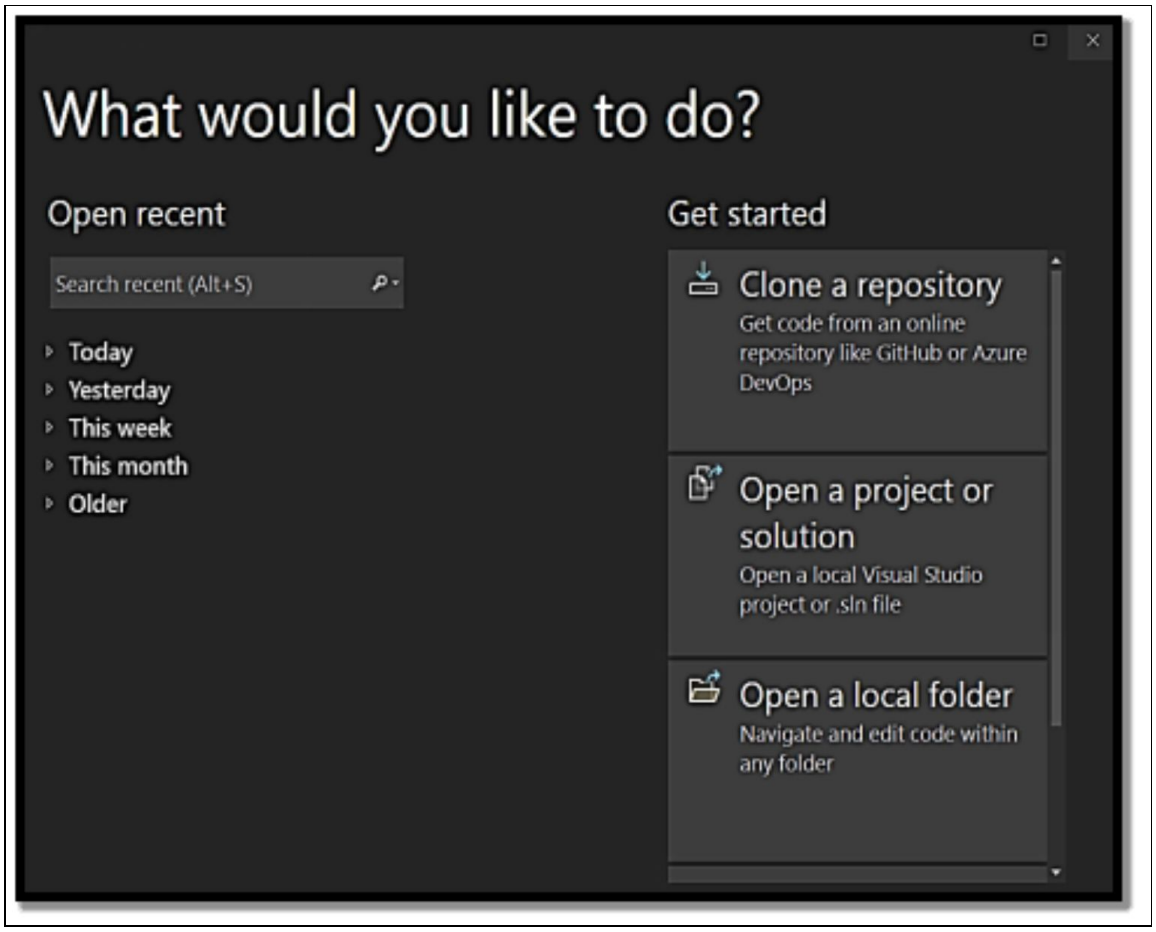

Then when you check the branch icon at the bottom, you can indicate which of the branch you want to work on.

# Cloning a git repository

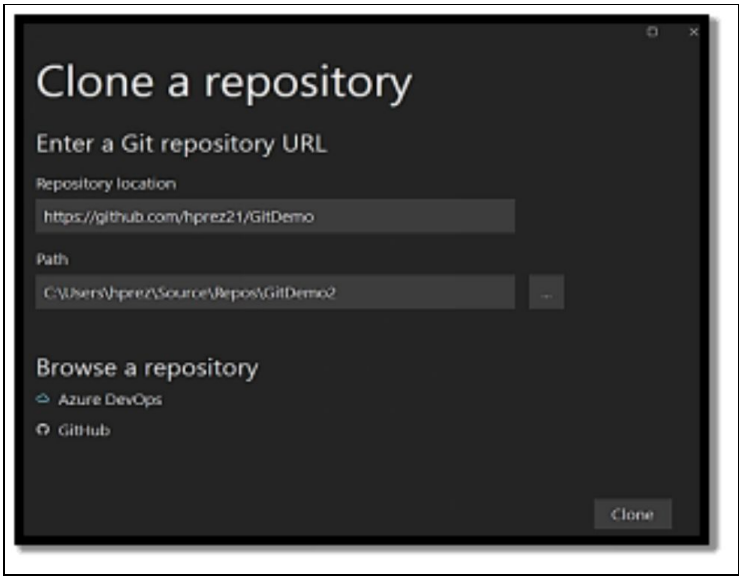
If you want to then clone an existing repository when you want to work with a team instead of starting from the beginning, here is what to do.

Closing an active project, opening a new instance of Visual Studio, or selecting **File** then the **Start** Window will all take you to the first window of Visual Studio, where it is simplest to clone a repository.

**The Clone a repository button is the first choice in this window:**




After pressing the button, you are going to see a new window that will tell you to indicate the URL of the remote repository and the local path where the source code is going to be stored.

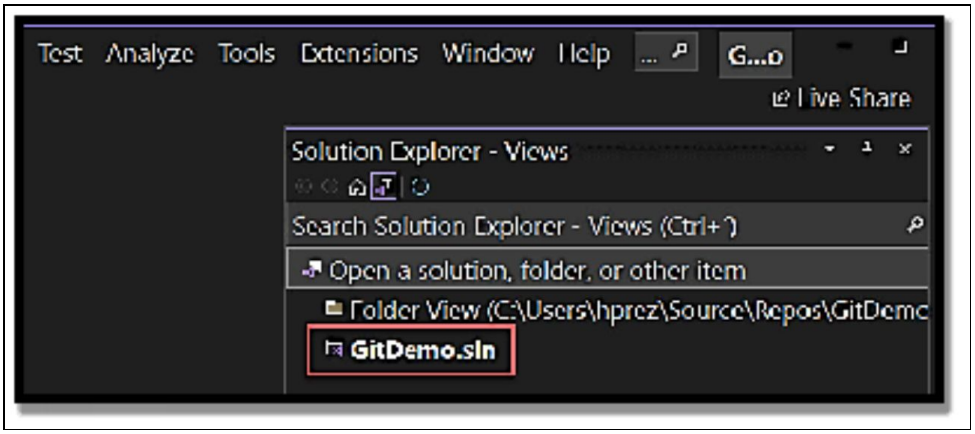


The URL of the repository generated in the Creating a Git repository section will be used for this example and should be like this: **https://github.com/{your-github-username-here}/GitDemo.**

We must choose a new path for the project because we already established one with the same name in the section on creating a Git repository. Then you need to rename the folder to GitDemo2 to make things easier.

Following the completion of the window's fields, we will click the Clone button to begin the process of locally cloning the repository.

**When the procedure is complete, a folder view will appear in the Solution Explorer window:**




When you switch from the folder view to the **solution** you have to select **gitdemo.sln** file. This then opens the solutions and all the files, which allows you to then work with the source file.

# Fetching pulling and pushing Git repositories

The most important tool when you are working with the Git repositories is learning how to fetch, pull and push operations.

**You can do this in two ways:**

By accessing through the Git menu

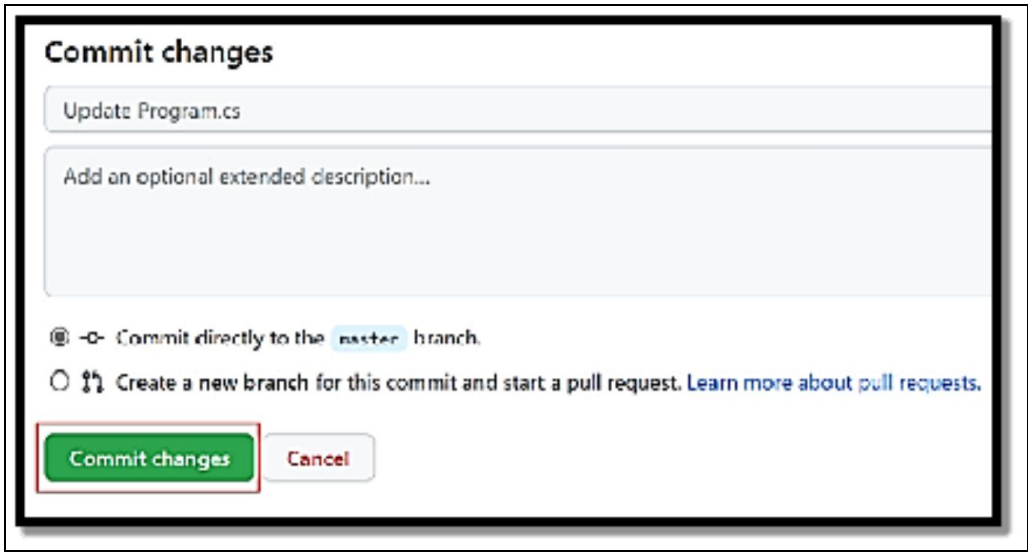
And the other way is to enable the **git changes** windows that you can open with the **view** and select b **git changes**

## Fetching repositories

When you are fetching repositories, you are checking if there are remote repositories you can incorporate into your local repository.

To work with this example, enter the **GitHub** portal and log in, then open the repository that you created, then enter the **program.cs** file and select the **pencil icon.** This way, you can edit the file.


You can make simple changes when you add exclamation marks at the end of the strings inline-four. And after you make that change, find your way to the bottom of the page and choose the green button.



Then when you choose the fetch button, you will see the outgoing and incoming messages which indicate that there are changes made to the repositories that you have not applied to the local project.

Then when you choose the messages, you are going to see the version history of the project which shows us changes that you have not applied locally.

It should be noted that, unlike the pull and push procedures, fetch does not make any modifications locally or in the remote repository. Next, we'll talk about these procedures.

# Pulling repositories

When we are talking about pulling repositories, we are talking about downloading the most recent changes from the repositories to the local project. When you go to the **fetching repositories** section you are going to see the pending changes. So, we are going to select the **pull** button.

After you download the changes, you are going to see messages that indicate which of the commit has been added to the current project.

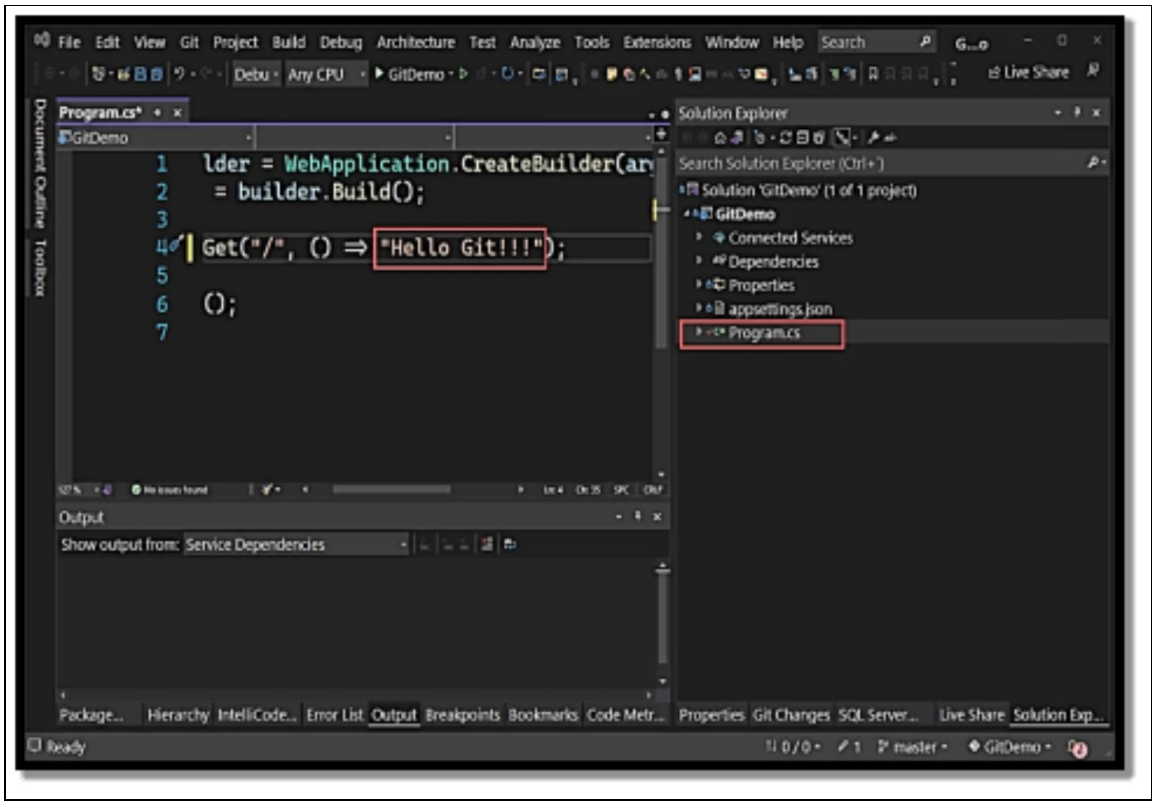
When you choose the name of the commit, you are going to see the window which shows the changes that you applied to the project.

## Pushing to repositories

Uploading updates to a repository is referred to as a push operation. Let's edit line four of the Program.cs code to read Hello Git Instead of Hello World to see how this process functions. app.MapGet("/", () => "Hello Git!!!");

A red checkmark appears next to the file icon in **Solution Explorer**, indicating that the local repository has changed.

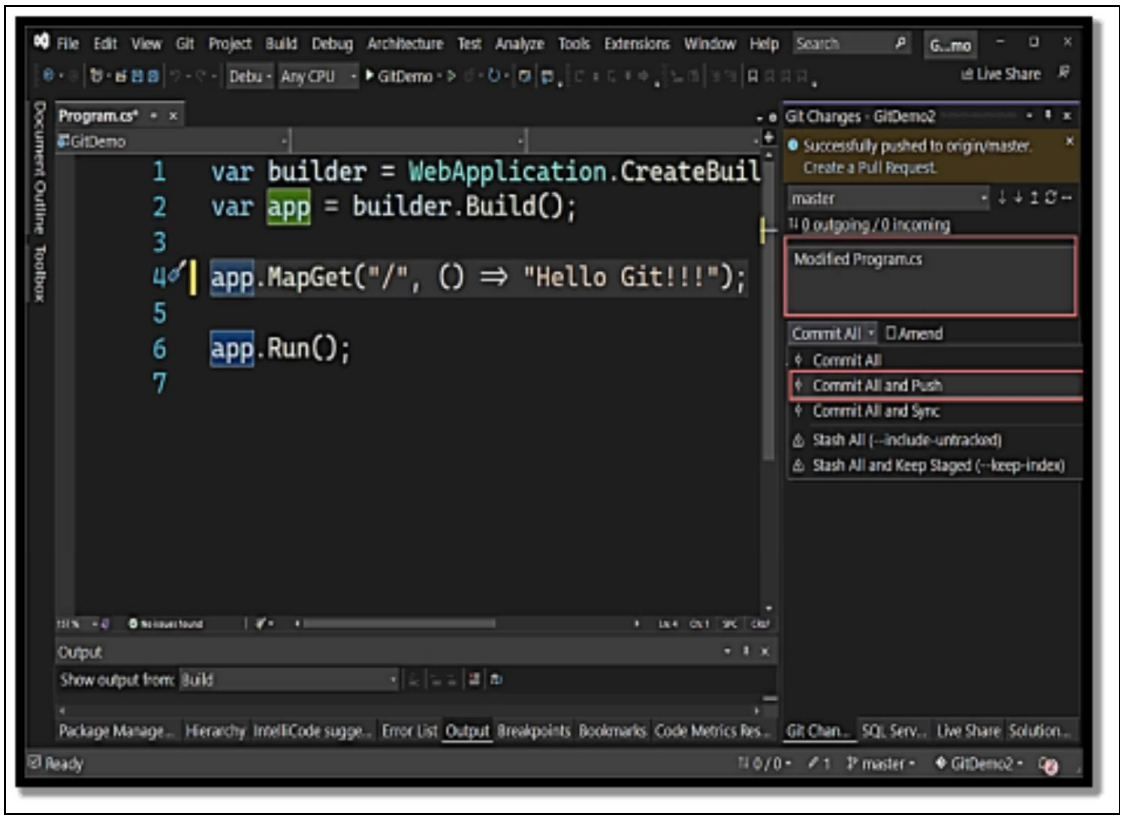
**We can now publish this update to the GitHub repository:**




Also, when you go to the **git changes** window, you are going to see a list of the files that there are changes that can be uploaded into the repository.

We can also notice a button with the Commit All legend in the previous illustration. By using this button, a local commit may be made without impacting the remote repository. There are several options available in this drop-down button, including the Commit All and Push options. The modifications will be pushed to the distant repository using this option.

**To make a point, we'll add the Modified Program.cs message, as seen in this image. and then choose the Commit All and Push option to make the server-side adjustments:**




When you then enter the online repository in the GitHub account, you are going to see the changes that have been successfully applied.

# Managing branches

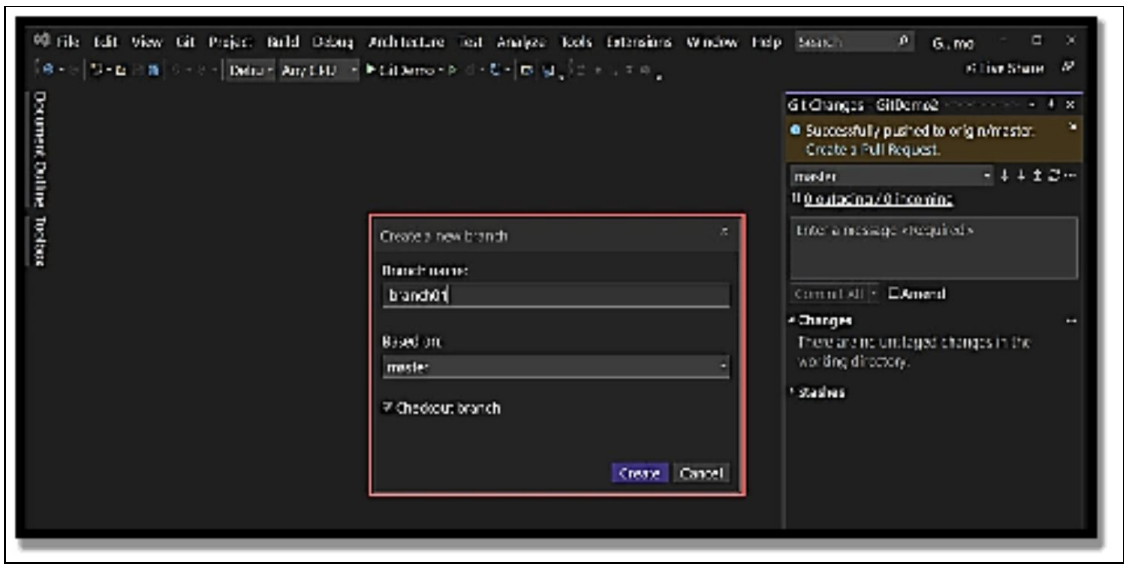
We have been working on the master branch, which is our project's primary branch, up to this point. Think of this branch as a timeline with commits acting as each event. When there is a problem and you need to revert to a prior version, reversing the modifications of a certain commit, is quite helpful.

But if you're part of a team, it's typical that you'll need to develop functionality in a sandbox before merging it into the master branch. In situations like these, Git branches will be useful since they let us build a new project branch from an existing repository branch and work on it separately from the main repository.

Simply select the **Git** then **New Branch** menu option to start a new branch. When you do this, a new window will appear, asking for the branch name, the branch that the new branch will be based on, and a checkbox labeled

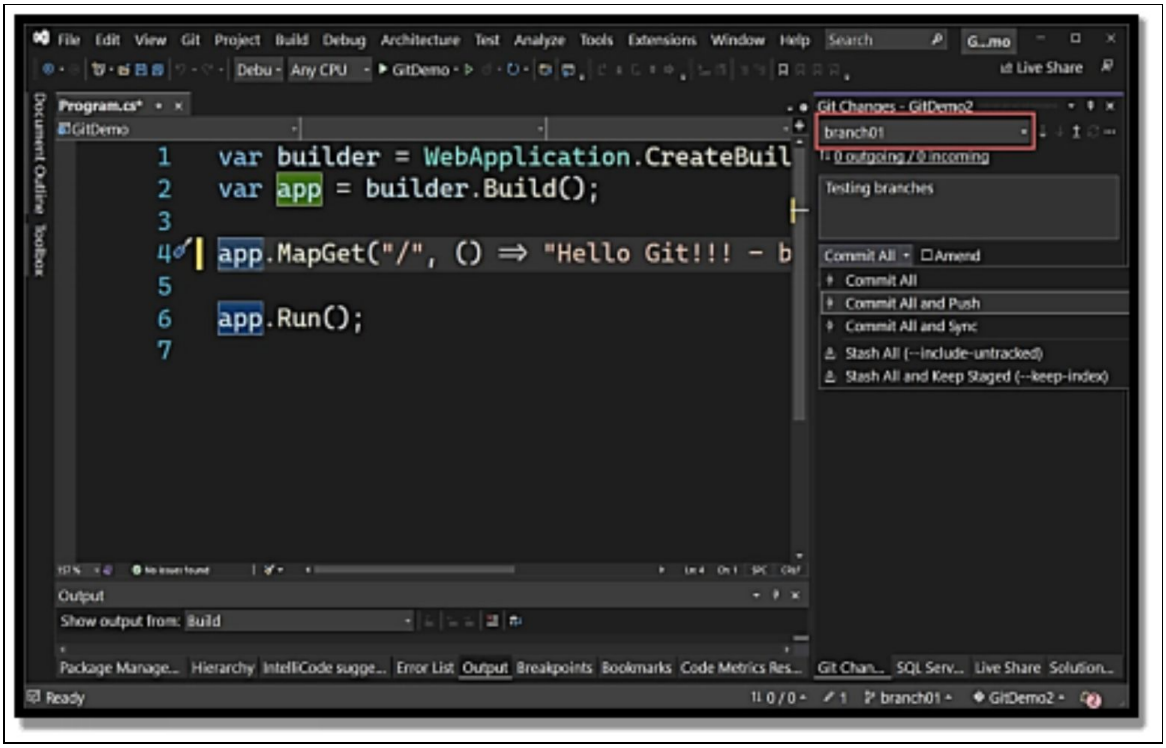
"Checkout branch," which, if selected, will switch to the new branch once it is formed.

For the sake of this demonstration, let's call the branch branch01 and assume that it will be based on the master branch.



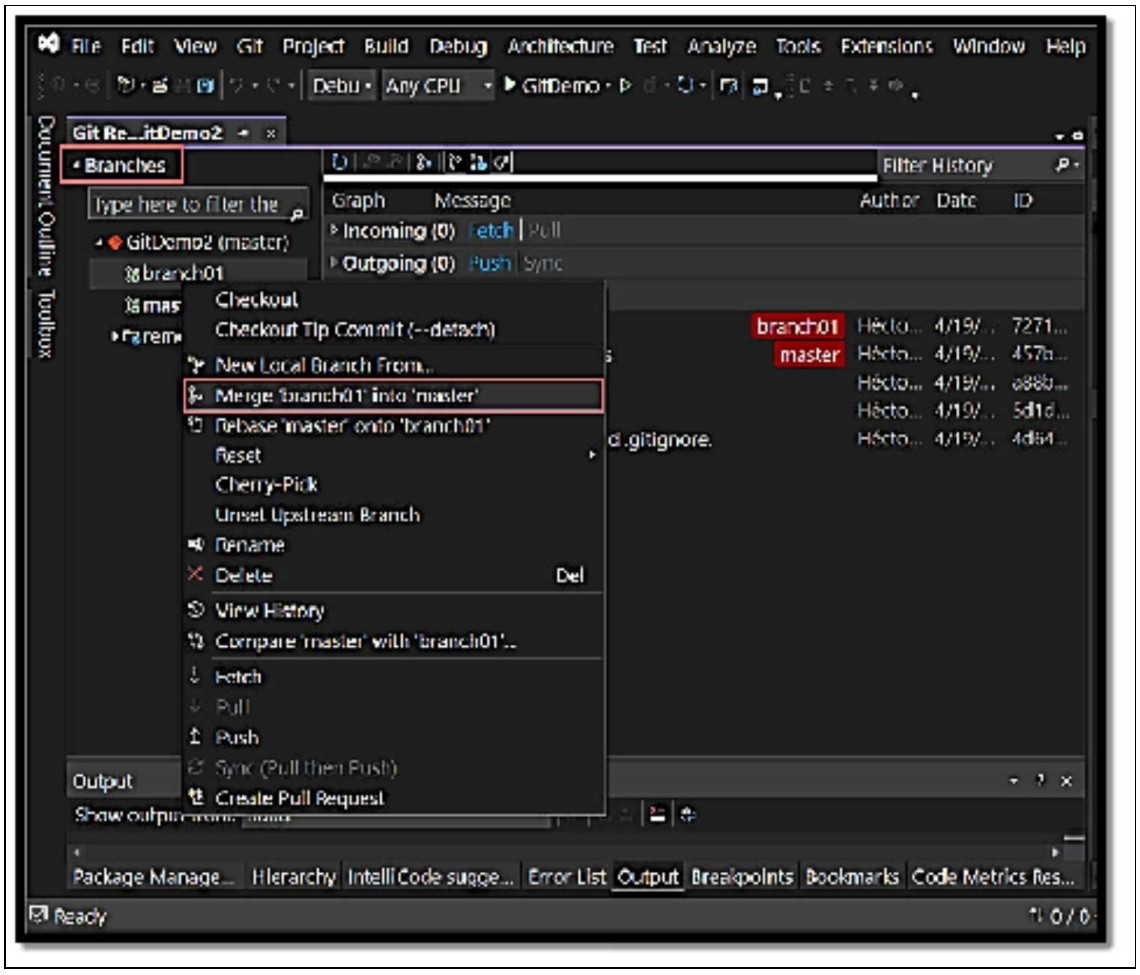

Then when you create the branch, you can then add the changes we want to add. For this, we are going to open a program.cs file and then modify line four again using this code. app.market("/", () => "Hello Git!!! - branch01");

After making these changes, you can then push the code ensuring that you apply the changes to the branch01.




As a result, we may add functionality to a separate remote repository without having an impact on how the master branch functions.

You should generally incorporate your work back into the master branch once it has been tested. From the **View** then Git **Repository** window, you may perform this. We'll see a list of all the different branches in our project in the Branches area of this window. To go to the master branch, where we will add the branch01 modifications, just right-click on it and select the Checkout option. Next, right-click the branch01 branch and choose the option to "Merge branch01 into master":




By doing this, the branch01 branch will be combined with the master branch, including the new functionality in the project.

After learning how to construct branches and merge them, it's time to examine Visual Studio's ability to show us the changes as we make them to the source code.

## Viewing changes in repositories

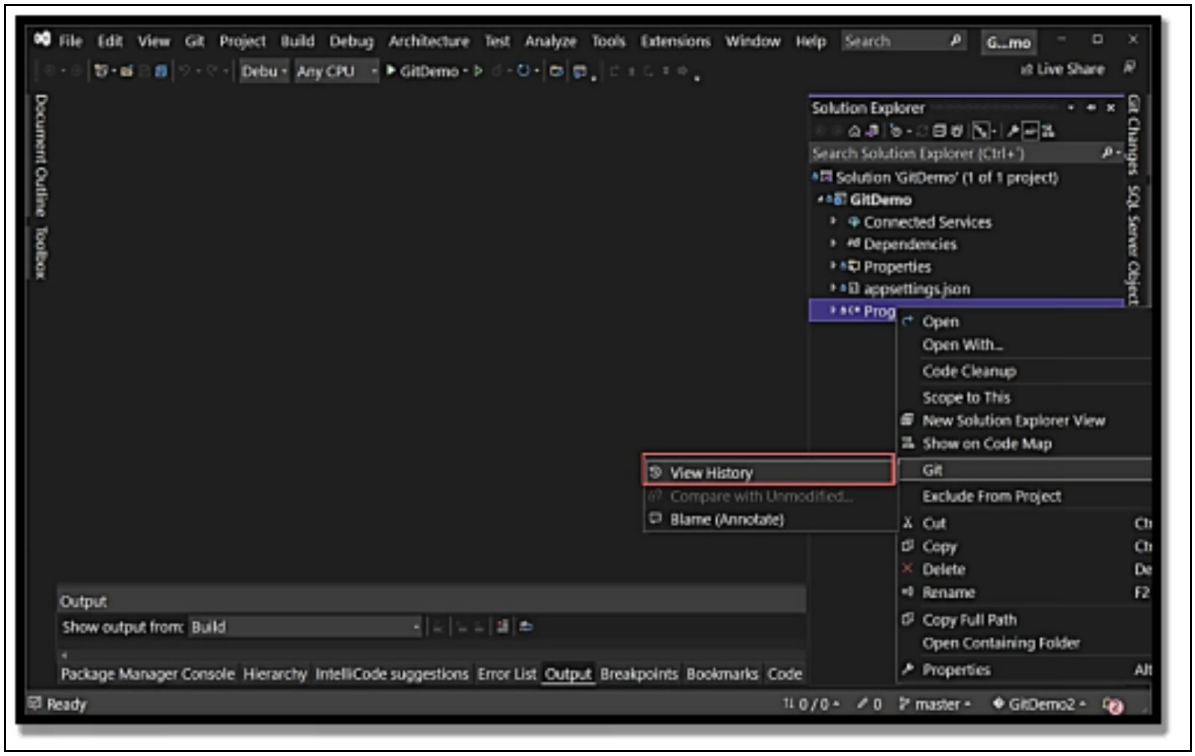
There are a few ways that you can use visual studio to visualize changes. The first is through the **GitHub repository window.**

Through **Incoming**, we can see which versions in our local repository have not yet been applied. Through Outgoing, we can see which commits have been made locally but have not yet been sent to the server.

**Simply right-click on two or more repositories and choose the Compare repositories option to see the changes that have been made between them:**

A new window will open up showing the differences between the various source files.

Right-clicking a file in the Solution Explorer and choosing **Git** then **view History** will also open the changes window for the selected file only, providing another method to view changes to a single file.




# Review Question

- How do you implement Git Integration?
- How do you manage branches?
- How do you look for repositories

# Summary

This chapter taught us how Visual Studio combines capabilities to make managing projects with Git and GitHub simple.

For all developers who wish to manage their projects in a more controlled manner to have an overview of a project structure in earlier versions, knowledge of how to deal with Git-based projects is essential.

Similarly, if you collaborate with other developers, you will be able to divide up project tasks, test them separately, and combine them once the code is solid, protecting the tested source code.

To do this, we have learned how to set up a GitHub account in Visual Studio, how to build and clone repositories, how to carry out get, pull, and push operations, how to manage branches in our projects, and lastly, how to see changes in repositories.

# CHAPTER 12
# SHARING CODE WITH LIVE SHARE

You can use the sharing tools to work remotely and work easily with teams. In visual studio, it’s incredible to see how we can collaborate on the same record, document, or share resources with other people in real-time. We need to be working on the same machine, in the same location, or using a technology that allows us to conduct a video chat to practice pair programming, a technique where two programmers work on the same code together.

Visual Studio Live Share is another tool that comes with Visual Studio and you can use it to share your codes with other visual studio programmers and codes.

## Technical requirements

### Understanding Visual Studio Live Share

Programmers who use Visual Studio and Visual Studio Code may collaborate in real-time with the help of VS Live Share. When Live Share was introduced as an add-on for Visual Studio 2017, it offered a few trial capabilities. It comes with all the features of VS 2022 and is included by default.

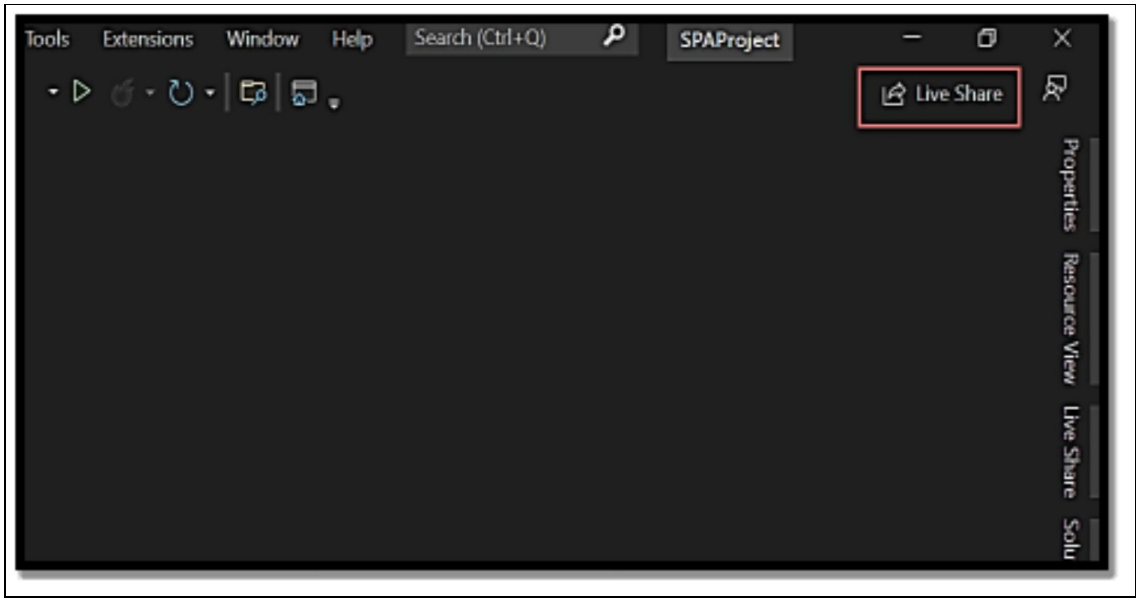
By installing the extension found at this URL: https://marketplace.visualstudio.com/items?itemName=MS-vsliveshare.vsliveshare, you may utilize Live Share in VS Code for free as well.

For remote developers, Live Share offers capabilities for editing, debugging, sharing terminals, and completing projects. To see and work with code, we don't need to clone the repository or apply for other extensions.

The Live share tool is not new. however, the other IDEs come with extensions and parts to share code and work in real-time.

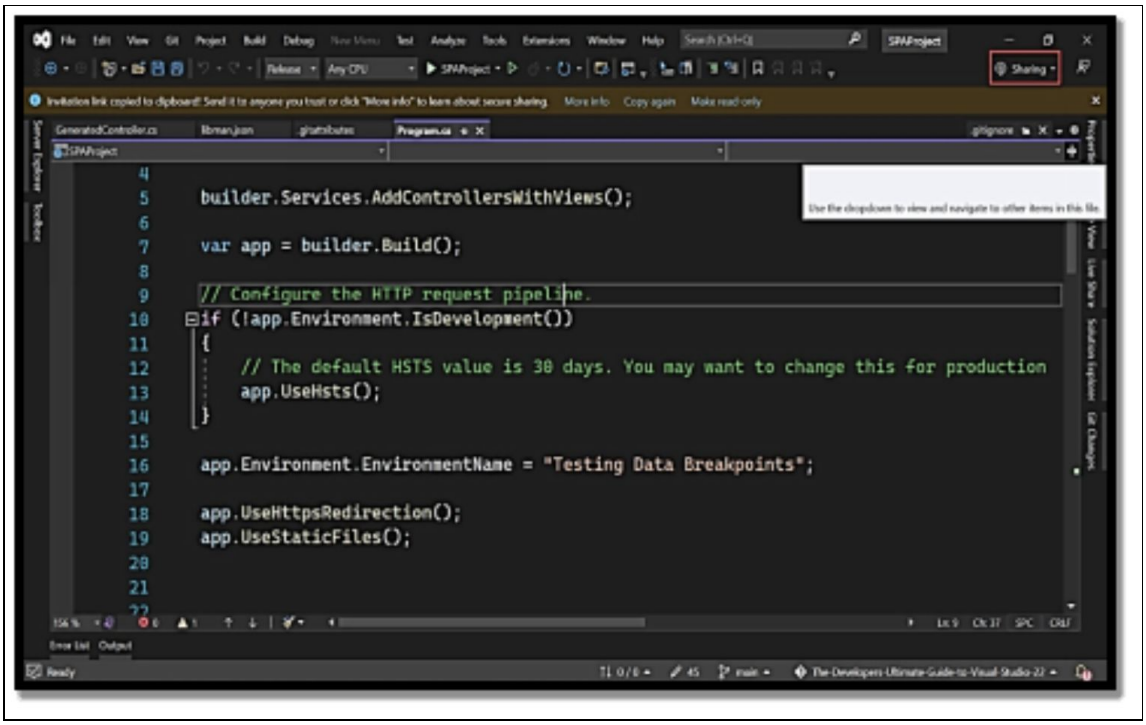
# Using live share

Go to the icon on top of the main window in the visual studio



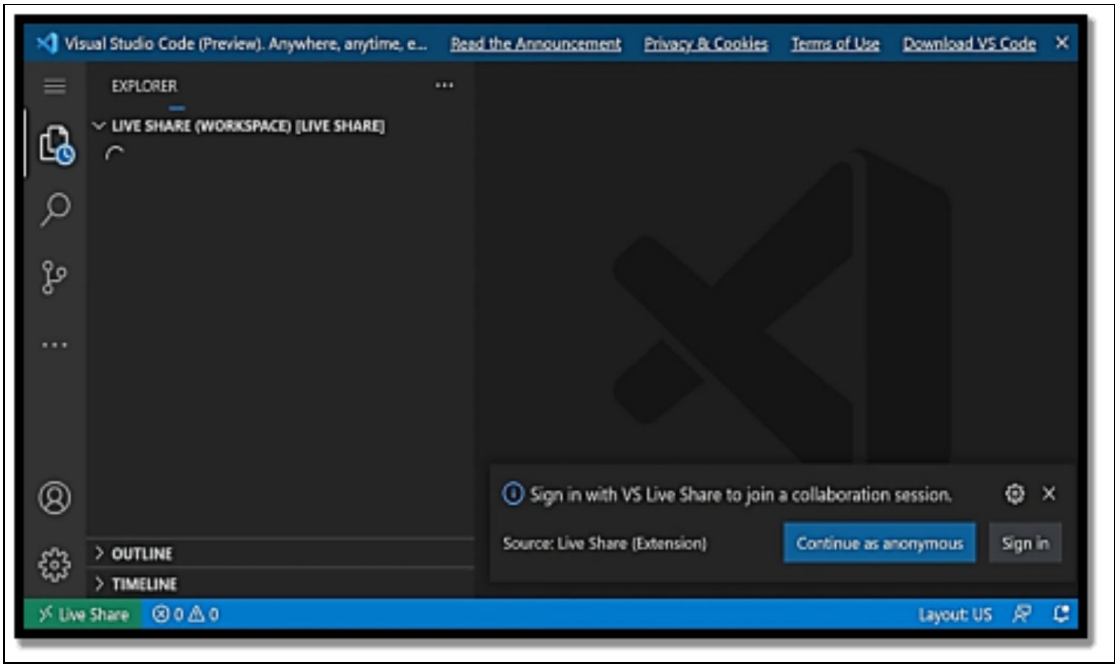

You may choose the account to use to start a new live sharing session by clicking on this icon, which will open a new window. This account is necessary, and by default, the account used in Visual Studio is chosen. Additionally, the area where the Live Share button is visible displays a Sharing... notice.

You may click **Select** after choosing the account you wish to use. The URL to share the session will then be copied to your clipboard automatically when you receive a confirmation message. The live session is shown by the live share icon, which says **Sharing**...




Then you can share the invitation link with the other developer.

**Visual Code for the web** will be shown when the link is opened in a browser. There are no system prerequisites or operating system dependencies for this version of the VS Code editor because it operates as a web page.



Then you can **continue as anonymous** to advance and **join the session.**

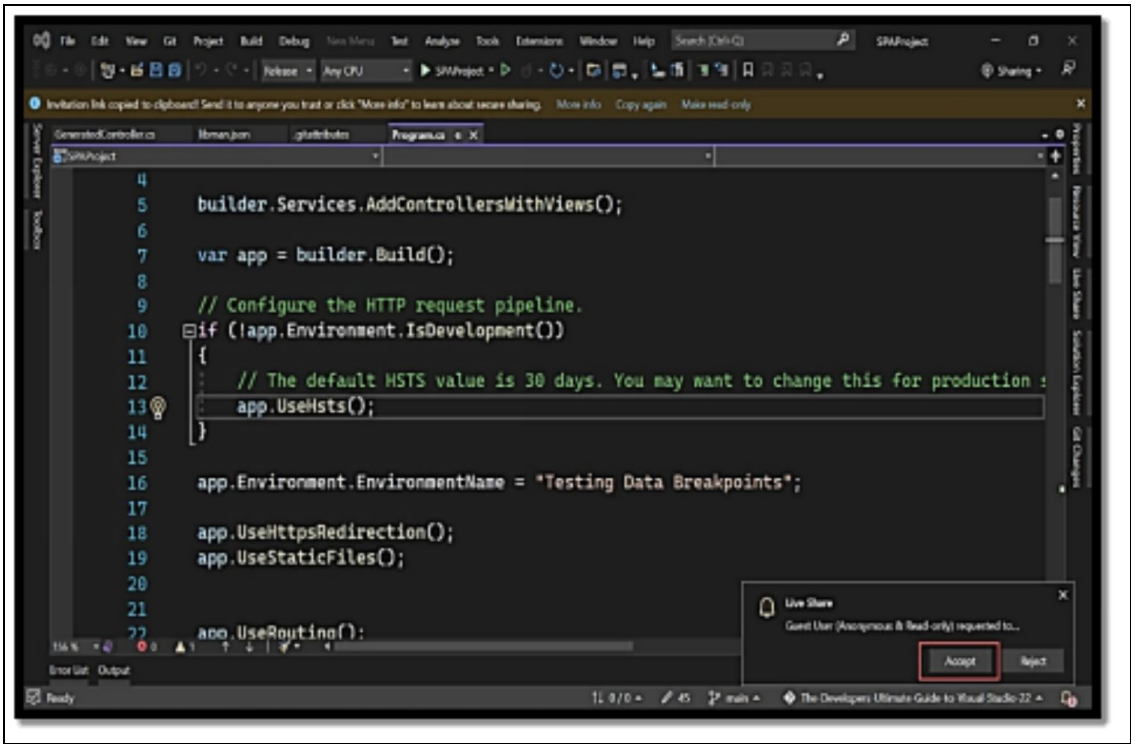
**There are three options.**

**Continue on the web:** Continue using VS Code on the web browser by selecting this option.

**Open in Visual Studio Code**: This choice will launch Visual Studio Code on your computer, allowing you to edit the code there.

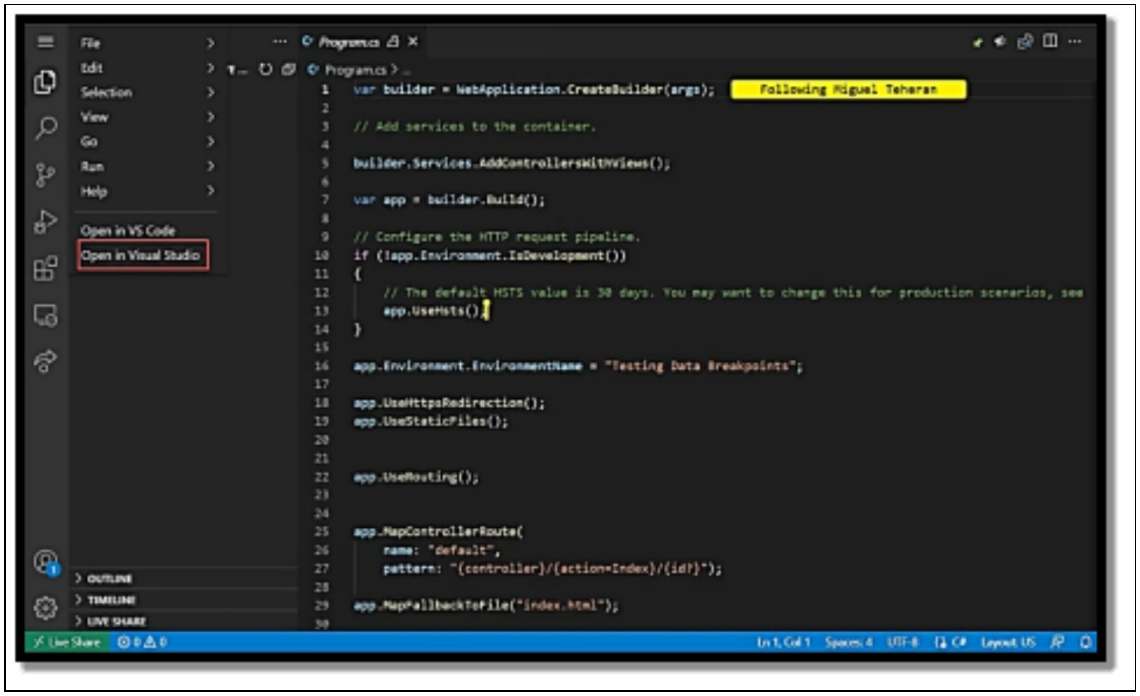
**Open in Visual Studio**: Selecting this option will launch VS on your computer so that you may edit the code there.

To join a session using the browser, choose to Continue in Web. A new notification message with information about the collaborators who are attempting to join the session will be displayed to the session creator. You can accept the new participant and carry on with the meeting.




Live Share will share the code and display the changes in real-time after approving the guest user. In this instance, the session's creator Miguel Teheran's code actions are visible to the guest user. See this image

**Following Miguel Teheran's message, highlighted in yellow, and also displays the choice to begin the session in both VS Code and VS:**




You will see the menu that has an option you can pick from.

**Here are the options.**

**Let's go through these choices and features in more detail:**

**View Live Share Window:** View the status of the session, along with its information and participants.

**Copy Link**: To share the link with others, copy it to the clipboard.

Open a new window to share our local server with other people participating in the session by selecting **Manage Shared Servers**.

**Share terminal(read-only)** To share the console log and the outcomes with others, open the terminal in read-only mode and share it with them. No commands can be entered into the terminal by collaborators.

**Share Terminal (Read/Write)**: This feature allows users to share their terminals during a session and allows remote command execution.

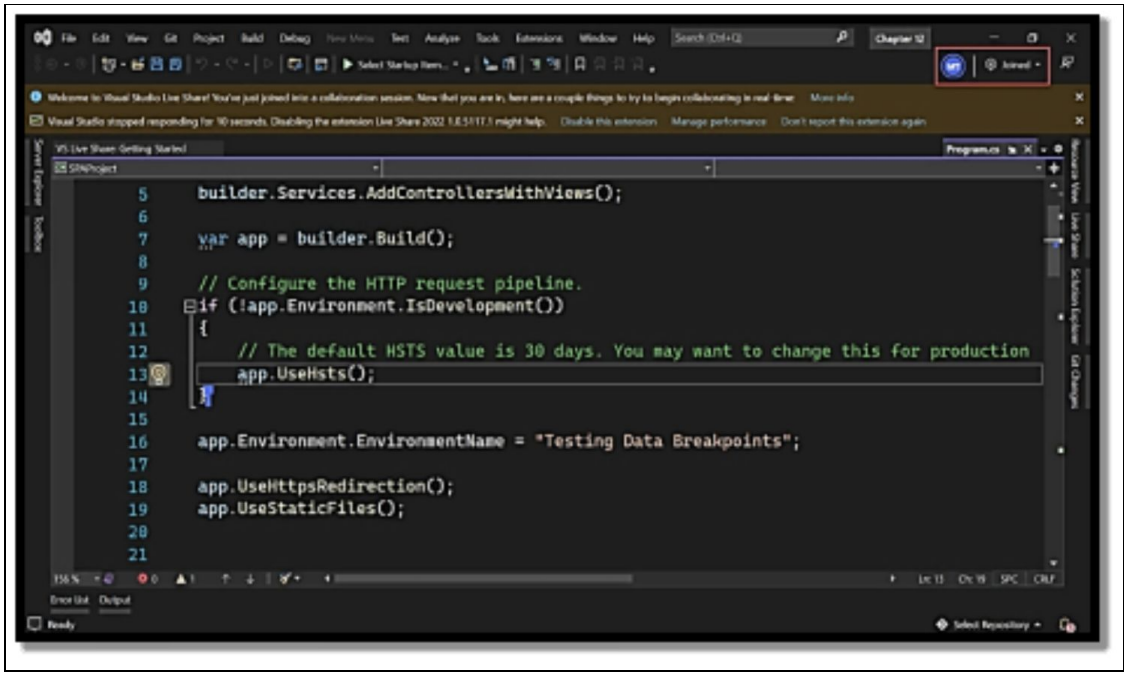
**End Live Share Session**Stop the Live Share session to end it for all connected users.

We can see the visitor user who is looking at the code in Program.cs in the Participants area. The shared terminals and servers are also visible. We can stop the session for all participants at any moment using the red square.

# Performing live editing

We can invite other users to join a Live Share session, but they can only do so in read-only or edit mode. Since all guest users are by default in read-only mode, let's utilize VS to join the session as a logged-in user.

As seen in this image, after selecting Open in Visual Studio, VS will load the project and display a loading notice before showing the code for the currently active session. To control the session as a collaborator, we will have a menu where the live share button is situated.




You are also going to see the icons on the left pane that represents the active collaborators inside the session. You can choose **joined** so that you can see the option.

**Here are the options:**

**View Live Share Window**: Using this option, you can see the current state of the session as well as its participants and its specifics.

**Session chat-**Using the session chat feature, a new window will appear, allowing you to text and communicate with other team members.

**View shared servers.** Shared servers in the current session may be seen by selecting View Shared Servers from the menu.

**Show Shared Terminals**: Selecting this option will bring up a new window displaying all of the shared terminals that are active.

**Focus Participants**: By choosing this option, you will maintain your attention on the work that your coworkers are producing or modifying.

**Leave Live Share Session**: By choosing this option, you will end the current session while the other participants can carry on.

After editing parts of the code, you can add a comment for the build method.

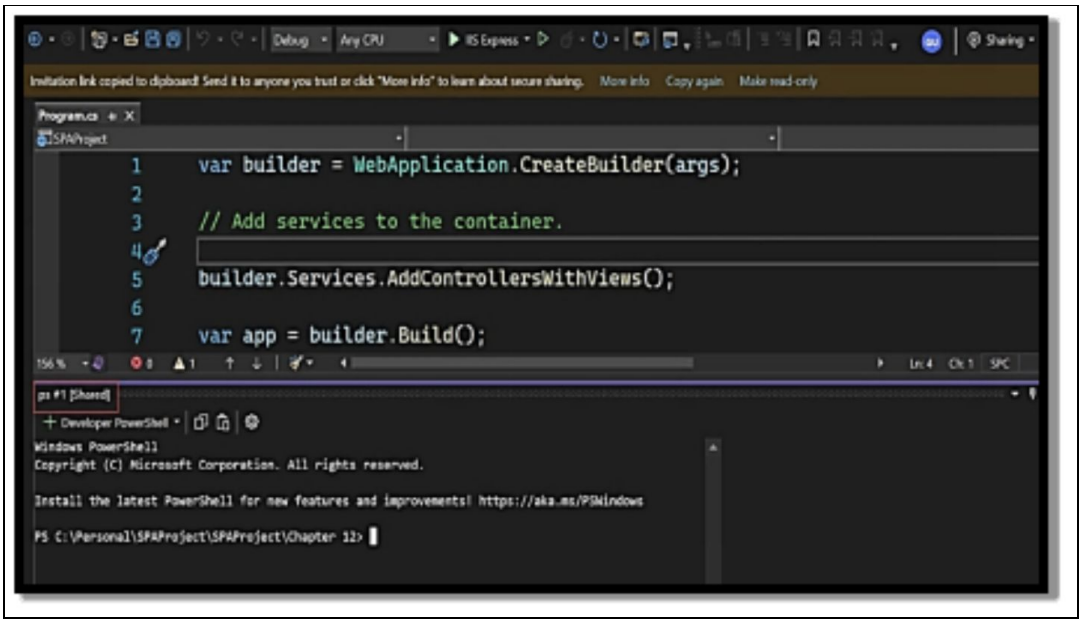
The host and other participants in the session can immediately view changes made to the file as the collaborator begins to type. The name of the collaborator will appear on that line, highlighted in a random color.

# Sharing a terminal with other collaborators

You can also share terminals with other people. When you are sharing terminals, you are allowing others have more details of the project with the command line. When you create a new session, you can make use of the share terminal option so that other users can use the terminal we created.

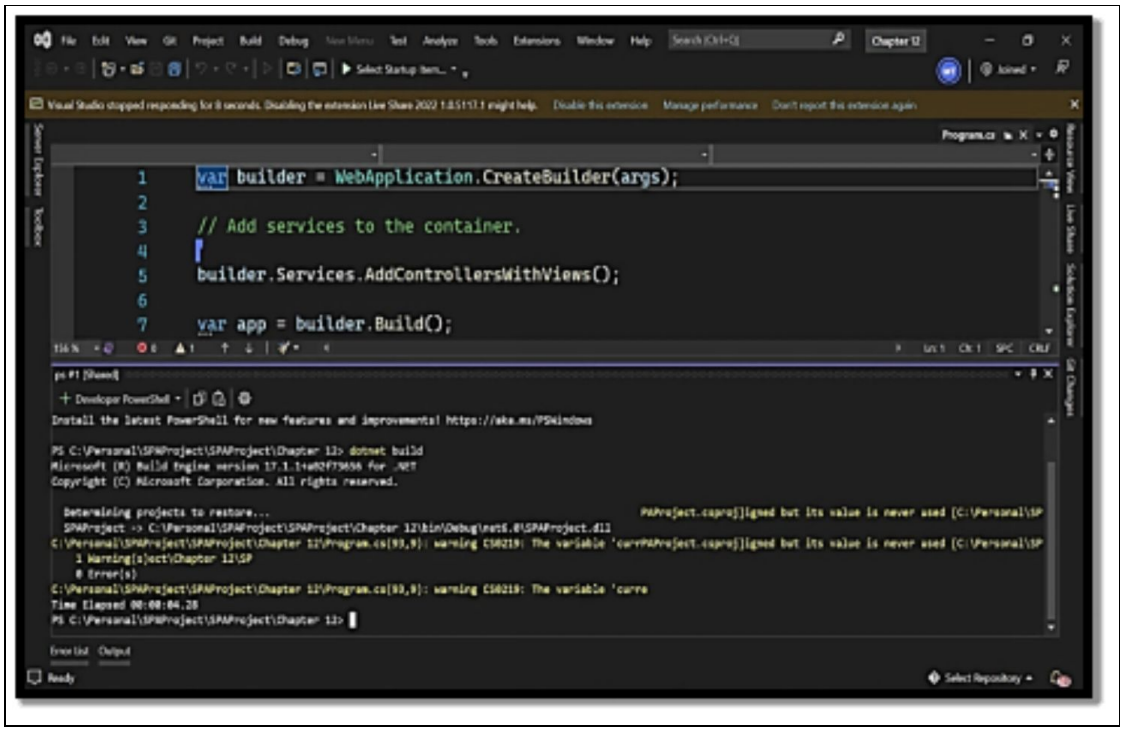
We will notice a new terminal in a window with a mark indicating that the terminal is shared after selecting **Share Terminal (Read/Write**).

**This image shows the new terminal that VS has inserted in the bottom panel:**




The shared terminal is going to be shared automatically.

Any command in the project may be run by the session's collaborator to contribute information or receive more details using the windows commands. In this situation, we may make use of the**.NET command-line interface (CLI**), a tool that is part of.NET and allows us to build, restore, run, and deploy the .NET-based applications using straightforward terminal commands. You may compile the project using the dotnet build command to interact with it. you are going to see a warning and error during compilation in the terminal.




The hosts and the other collaborators in the session will see the command executed and the results inside the terminal. The command will be executed using the host environment.

You may visit the Live Share window to view the status of the current session, including the shared terminals and their permissions, if the terminal is not automatically shown or if you wish to know which additional terminals are shared in the session.

You may invite people to work with you on your project and share a terminal, when necessary, now that you understand how to utilize Live Share in Visual Studio 2022 as a host and collaborator.

# Review Question

- How do you share terminals with others?
- How do you integrate visual studio with live share?
- How do you do Live editing?

# Summary

A live Share is a fantastic tool for collaborative real-time work. This tool has become increasingly crucial as a result of globalization, taking into account that a worldwide development team may consist of developers working in various time zones and using various technologies all over the world.

You now know how to utilize Live Share to collaborate with other developers on a project as a team. You are familiar with how to open a Live Share session in Visual Studio, check who is participating, share a terminal, and stop the session whenever you choose. You also looked at various editors and IDEs that are Live Share alternatives.

# CHAPTER 13

# WORKING WITH EXTENSIONS IN A VISUAL STUDIO

You can work with extensions inside the visual studio. like you can use the extensions to apply new themes, add custom colors, or some complex functionality like code refactoring tools.

In this chapter, we are going to be taking you through how to work with the different extension tools available

## Technical requirements

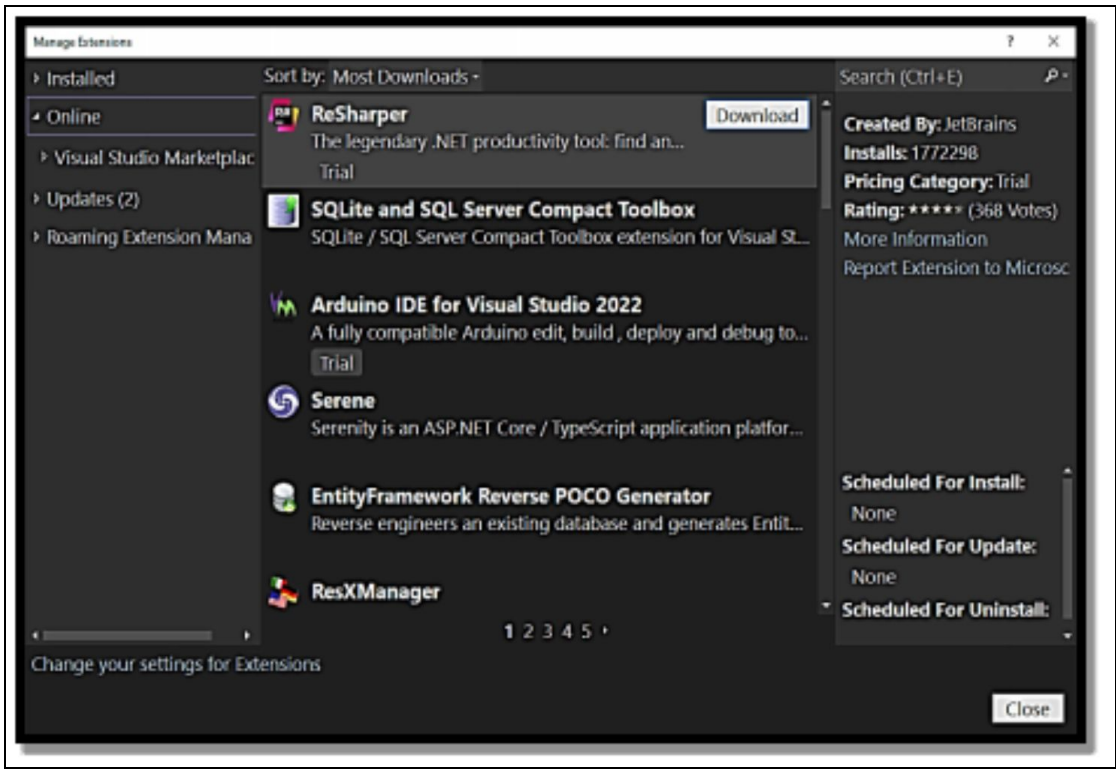
You must install the Visual Studio extension development package to build a theme as an extension, as demonstrated in the Creating a new theme as an extension section.

You may either install it while building the extension, as demonstrated in the Creating a new theme as an extension section or before you begin reading that part.

### Working with the extensions tool

The main goal of extensions in Visual Studio is to increase your daily productivity by adding features that might be somewhat specialized for particular tasks, like making recommendations for best practices in code, performing code cleanup, highlighting messages in the output window, adding visual features in code files, or interacting with SQLite databases. The Microsoft team made a wise decision by enabling developers to add new features to Visual Studio and share them with the rest of the world.

So, to access the extension, you need to go through the extension tool by entering **extensions** and then **managing extensions.**




You are then going to see the following options to filter the extensions.

**Installed**: These refer to the extensions that we've already set up in this instance of Visual Studio.

**Online**: The extensions that can be added to Visual Studio are included in this category. We have a filter with the Sort by a legend in the upper section that lets us sort the extensions in several ways.

**Updates**: This category displays extensions that need updates that will either repair bugs or provide enhancements to the extension.

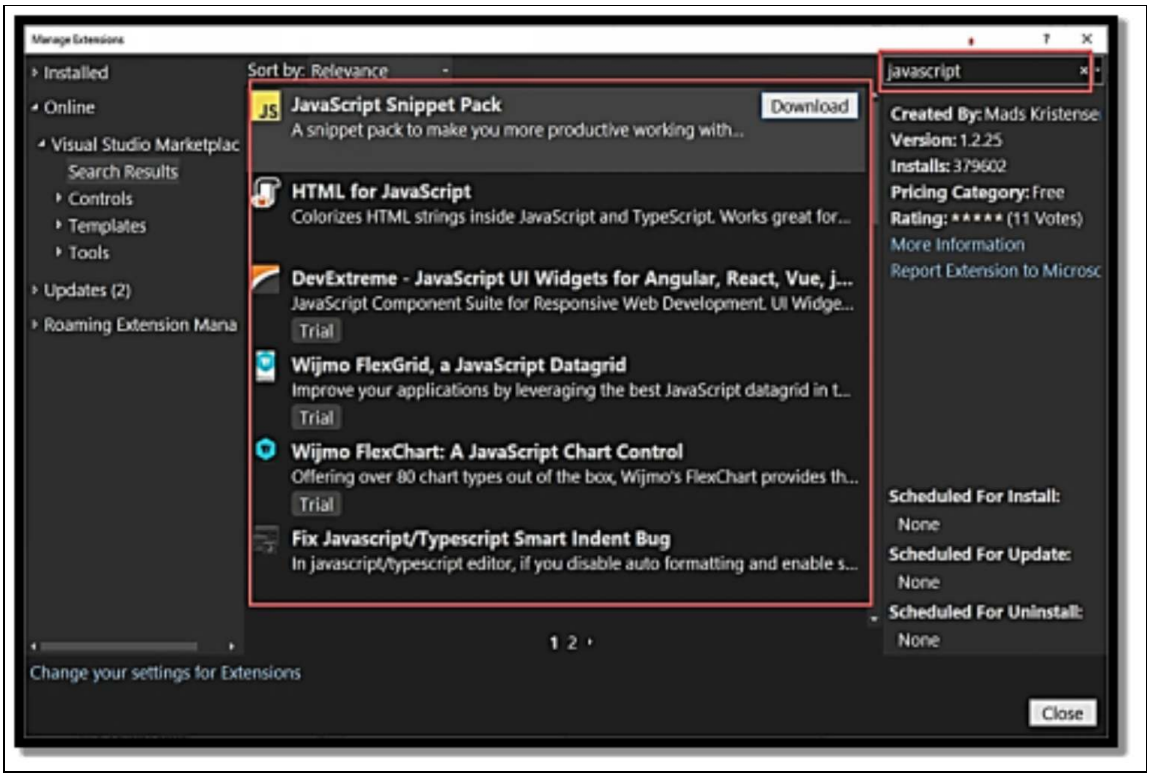
The **Roaming Extensions Manager** category enables you to examine extensions that have already been set up and connected to a Microsoft account to install them in the current instance of Visual Studio. This implies that if you install an extension while signing in on a separate computer, you will be able to see it in this list and can simply install it without having to remember its name.

# Searching for and installing extensions

From the extension tool in the image below, you are going to notice a search box to look for some tools. The search is typically performed in the category that you choose.

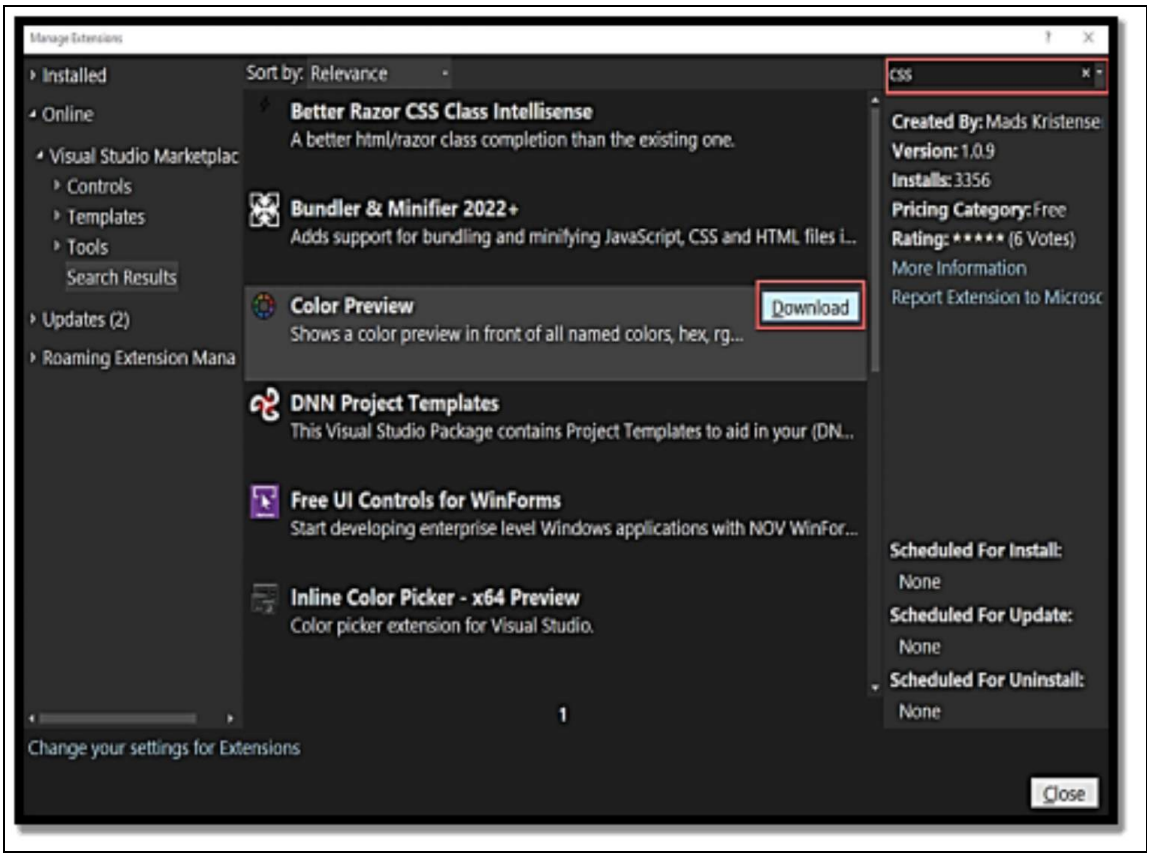
The easiest thing to do if you want to search all the extensions is to click on the Online category and do the search there.

**Let's look for extensions associated with the term "JavaScript" in our example:**



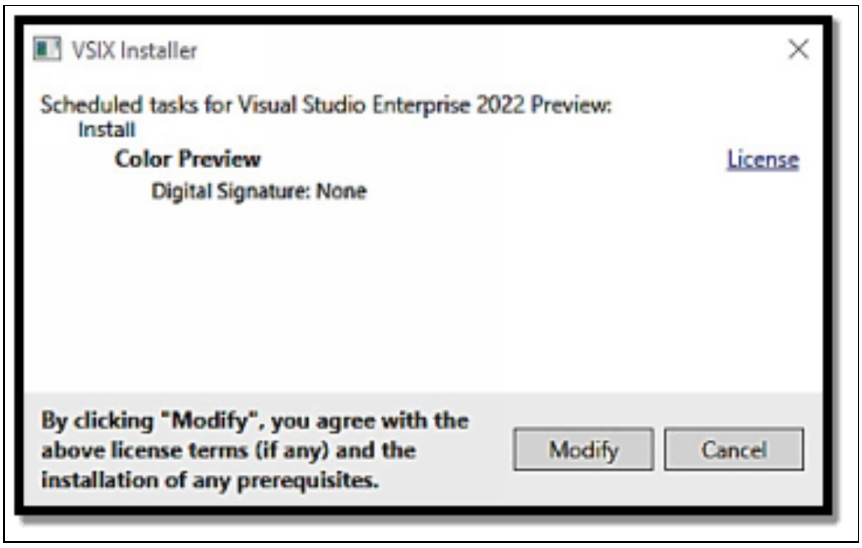

The drop-down menu at the top, which by default displays as sorting by Relevance, will determine the order of the list of extensions. However, we may also pick other alternatives, such as showing by Most Downloads, Trending, Most Recent, or Highest Rating. Let's run one more test, but this time we'll add an extension. In the list of CSSS-related extensions, you will find an extension with the name Color Preview.

**Simply click the Download button to begin the download process and install an extension:**



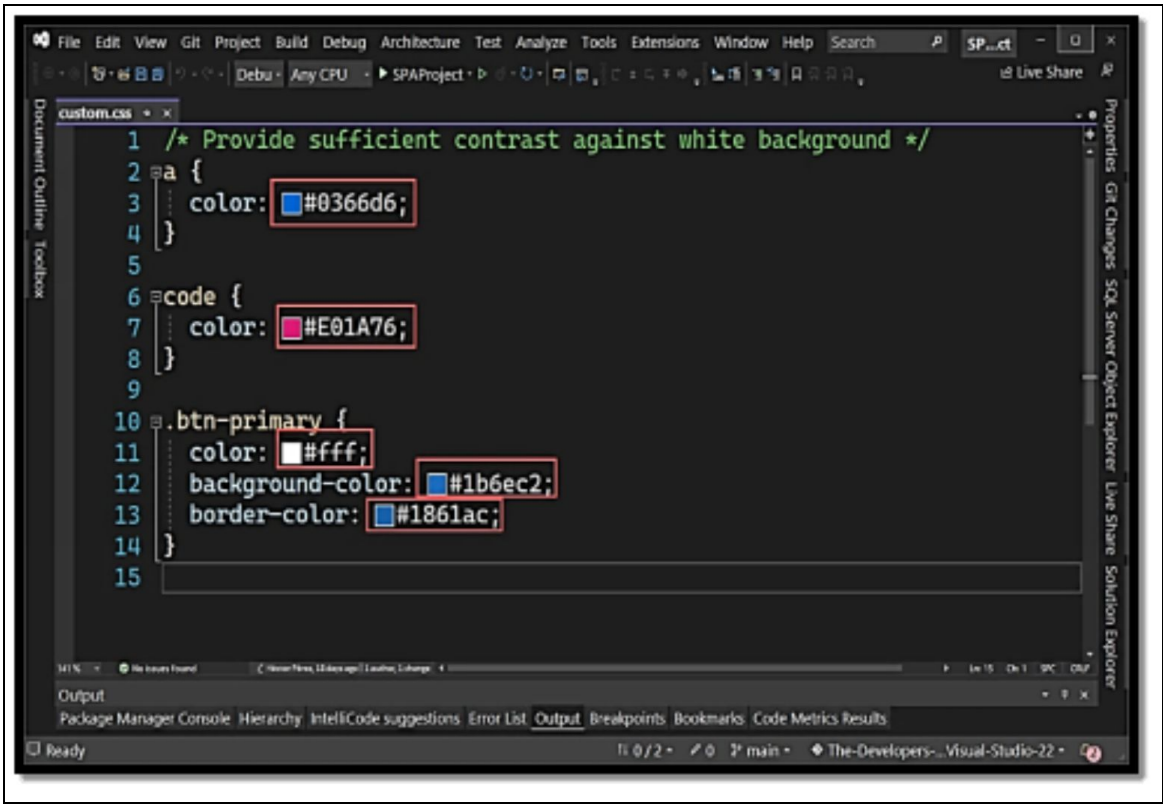

Your modifications will be scheduled after the extension download is finished, according to a notification that will show up at the bottom of the window. When all Microsoft Visual Studio windows are closed, the changes will start. Since the notice is fairly detailed, we'll relaunch Visual Studio to carry out the installation.

When Visual Studio is shut down, the installation process will begin, which will display as seen in this image both the extension that will be installed on the Visual Studio instance and other details like the types of license and digital signatures.




the information in this image is presented, and if we agree with it, we must click the Modify button to begin the installation of the IDE.

You must restart a version of Visual Studio when the installation is finished. Even though nothing appears to have changed at first look, this addition has allowed us to see the chosen color applied in the properties of a CSS file within the IDE, as seen in this image.



Then you can compare this with the ones on the CSS styling tools

To uninstall Extension, simply launch the extensions tool once more and seek the extension in the Installed section if you want to remove a particular extension. following that, click the **Uninstall** option.

You may temporarily deactivate an extension by clicking the Disable button in the screenshot above until you decide to enable it again.

# Reviewing visual studio Marketplace

The online extension store for Visual Studio 2022 is called Visual Studio Marketplace. Additionally, you may discover extensions for other members of the Visual Studio family in this market, including Azure DevOps and Visual Studio Code. The following URL will take you to the marketplace: https://marketplace.visualstudio.com/vs.

The user interface for the extensions tool is quite different after you access the site, but the basic functionality is the same. The many extensions are listed on the main gateway according to Featured, **Trending, Most Popular, and Highest Rated.**

you can search for what you are looking for. You are going to notice soon enough that you cannot use all of the extensions on Visual Studio 22. So, when you are looking for anything apply the filter **visual studio 2022.**


then enter what you are looking for then select it so that it goes to the extension page.

You may discover data on the extension page, including the number of installs, reviews, ratings, the change log, and the project page, among other things, that will give you a basic impression of the extension. Click the green **Download** button if the extension's description says it can assist you in solving an issue, as illustrated in the example below.

A file with a vsix extension will begin to download; in this example, the file's name is VSIconizer. six. Making sure that all instances of the IDE are closed before running the downloaded file will start the same installation procedure that we saw in the Finding and installing extensions section. This will allow you to install the extension in Visual Studio.

After the installation is complete and Visual Studio 2022 is launched, we can see how the IDE tabs' design has changed, as shown in this image, with the text being substituted by icons.

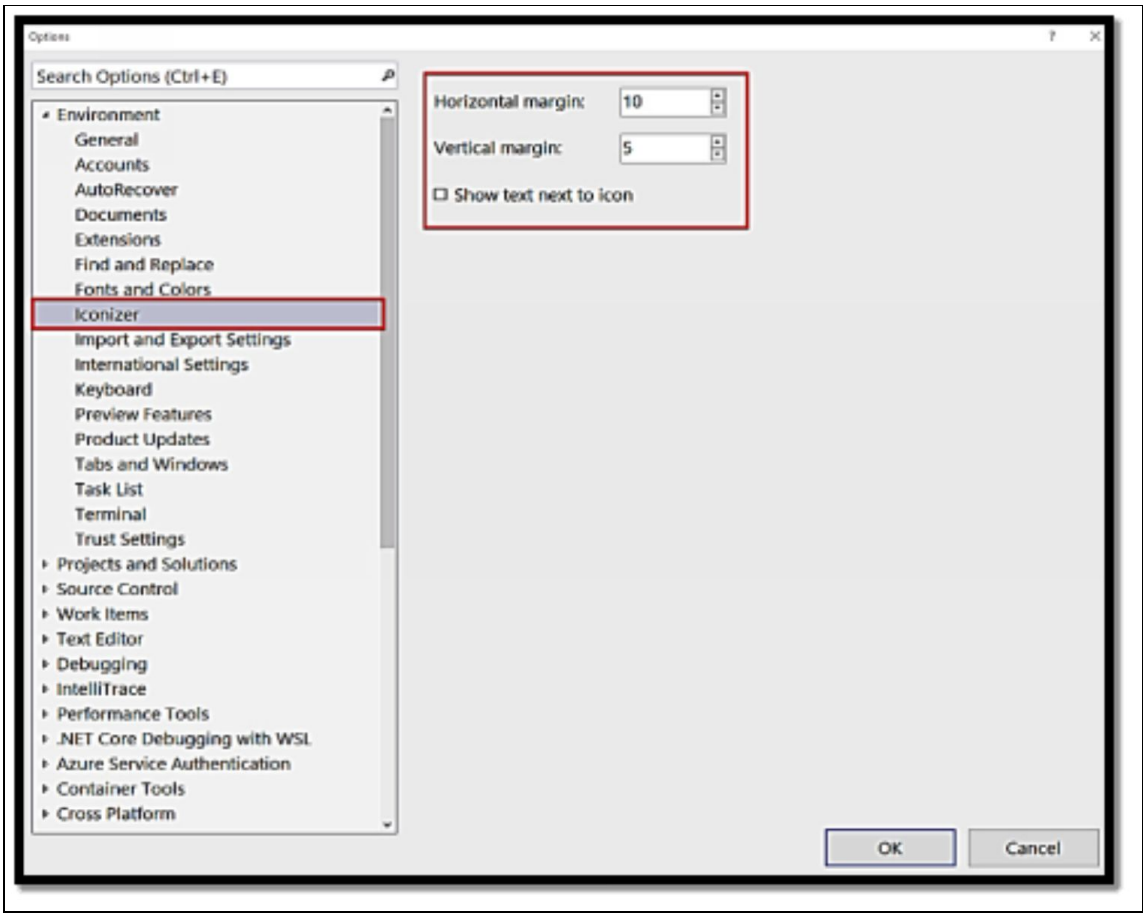
Installing an extension from the marketplace is that simple.

# Setting up extensions

Since every Visual Studio extension is different and has a specialized function, there isn't a common approach to set up them. The majority of extensions will indeed have configuration choices, either through a specific window or through the configuration options.

The extension page itself is the best resource for learning about these settings options. For instance, if you installed the visual studio ionizer, you are going to notice that the default behavior of the Visual Studio Iconizer extension, is to display just the icons on the tabs. According to the extension website, this behavior may be adjusted such that the tab's text appears next to the extension-added icon.

You can do this by going through the options added to the **tools** and selecting **options** and choose **the environment** than going to the **iconizer.**

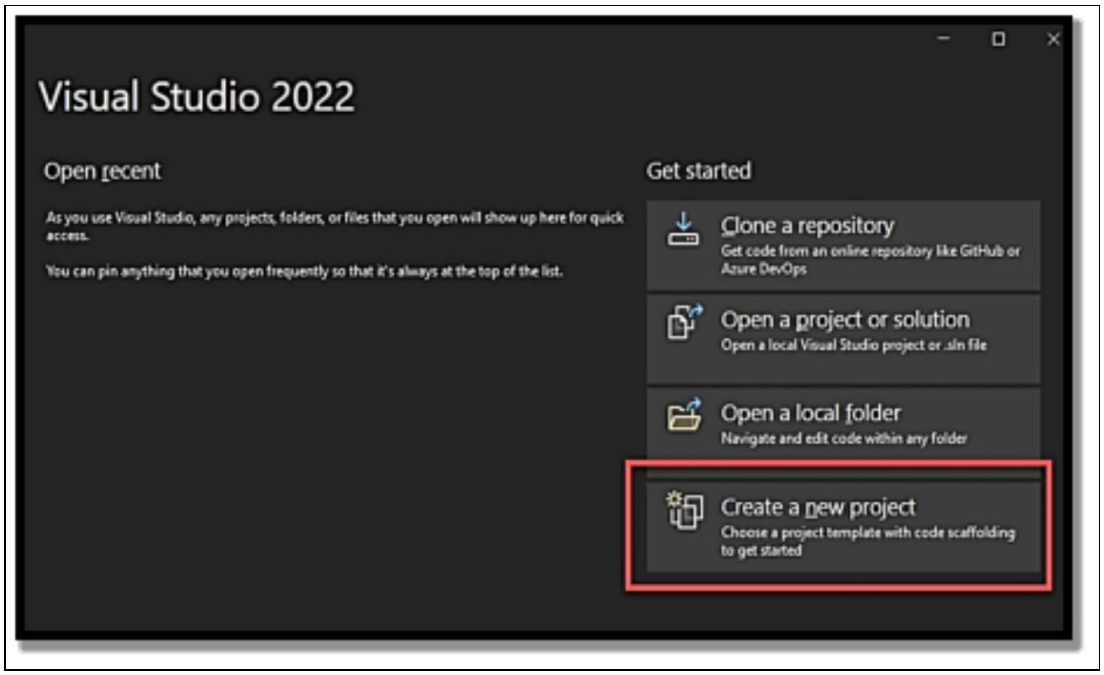



You are going to see a section like this after creating an extension and the more complex ones are going to create a special menu.

# Creating a new theme as an extension

With the help of the addon known as Visual Studio Color Theme Designer, you may carry out an even more specialized color setup for your IDE. As we saw in the part on reviewing the Visual Studio Marketplace, you can search for and download this extension there.

**Once the extension is installed, you must launch Visual Studio and choose the Create a new project option, as demonstrated here:**

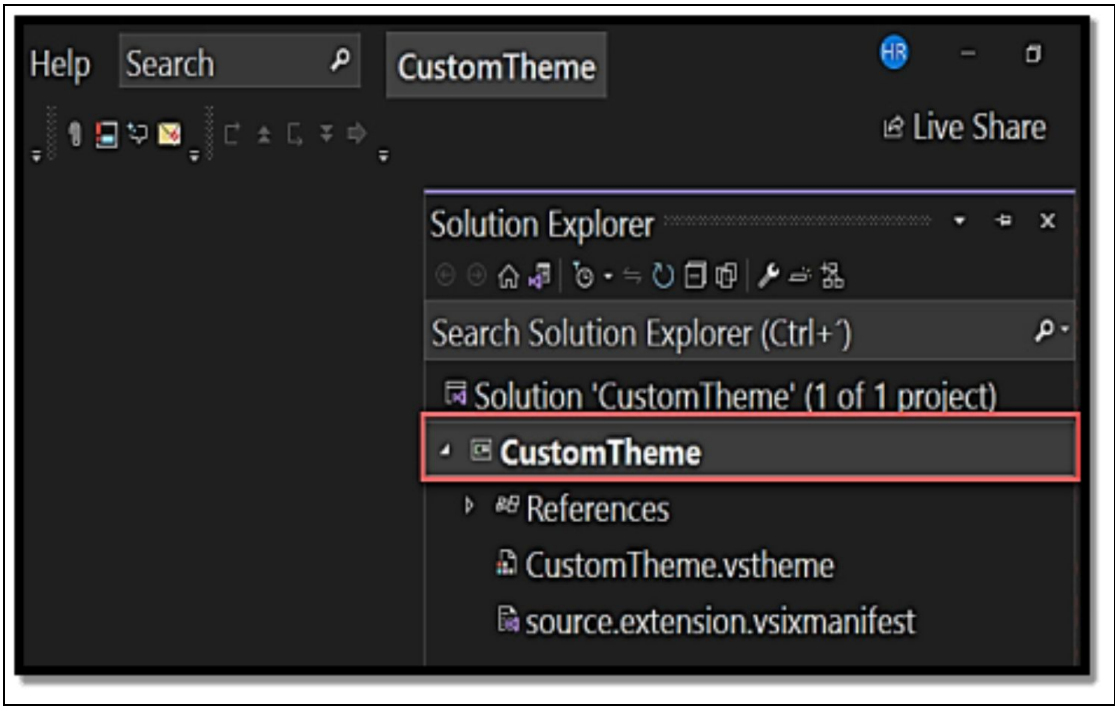



In the next window, you need to choose the project from the list of VStheme projects, installed because of the extension. And if you do not find it first, you need to look for it inside the search box on top.

Then in the **new project** configuration window, you can enter the name of the project referring to the theme you created. After naming it and putting in all the credentials then you are shown the window that indicates that you have to install the **visual studio extension development** and select **install.**

The environment becomes enabled to develop the extension.

After the IDE installs the extension, then the new project is then created with elements. In other to configure a new theme, you need to double click on the file Vstheme extension when you are in the solution explorer

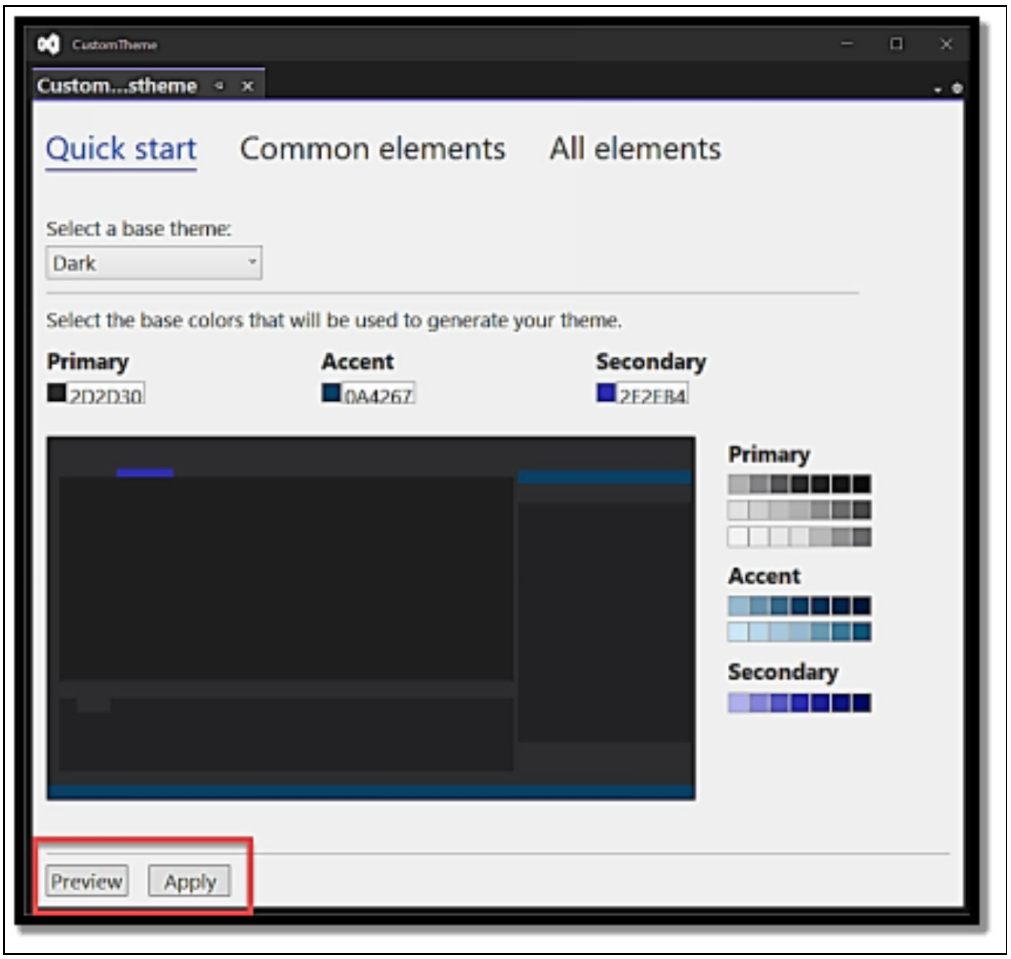



Next, you can start a wizard that helps you to create a new one. you are going to start with a tab **quick start,** to choose a base theme from the installed ones to modify the new theme.

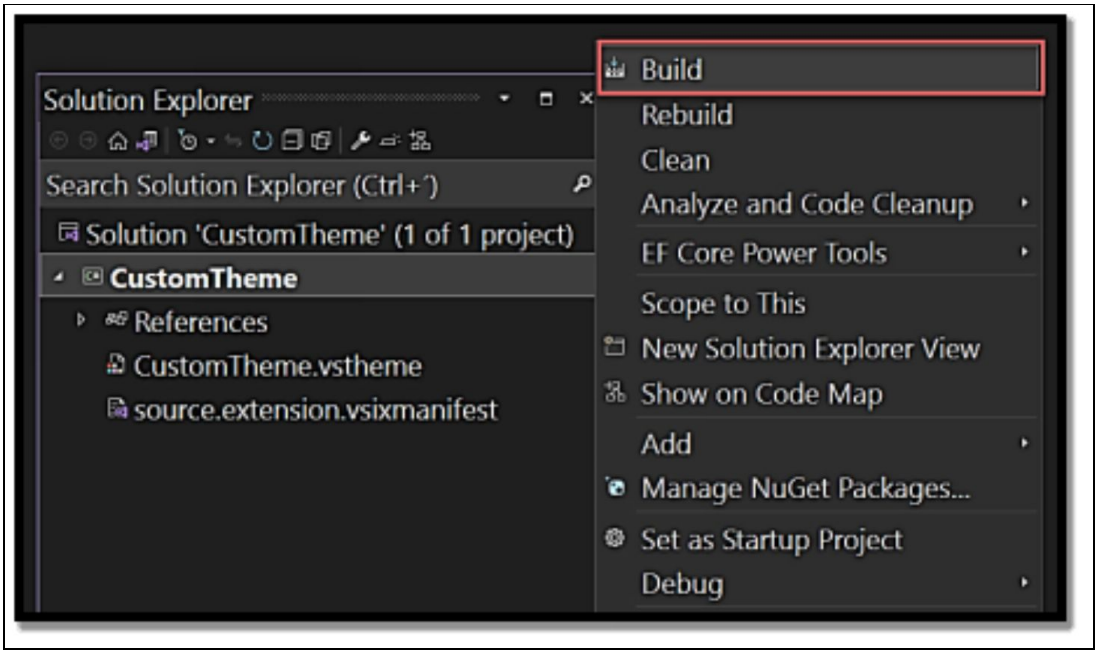
If you choose the dark theme, you can also choose the three primary colors, **primary accent, and secondary** so that you can get a new color and apply it.

The same process may be used to customize your website in more depth under both the Common elements and All elements' tabs, as illustrated in this image.

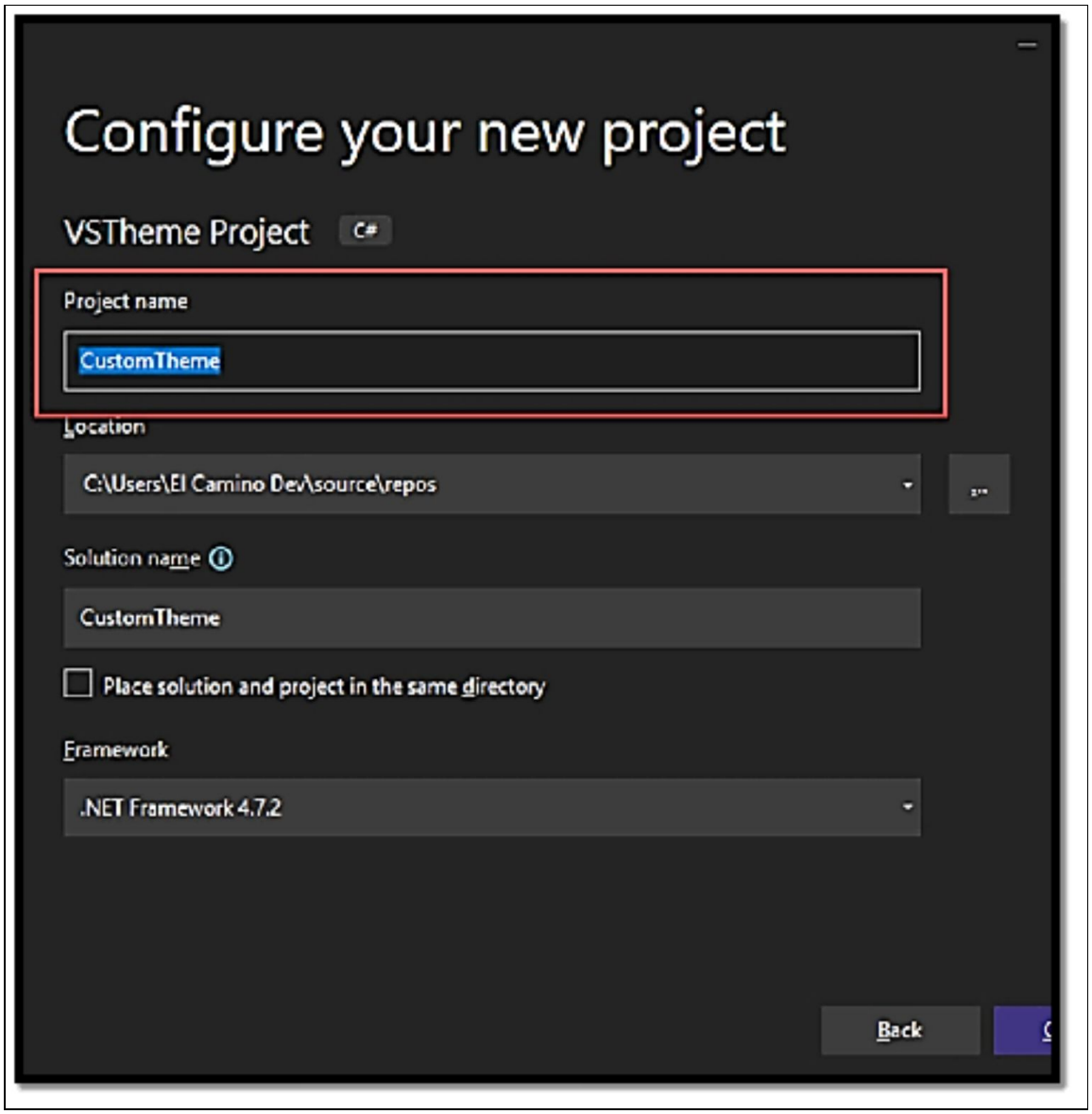
Finally, to utilize the theme, there is an Apply button and a Preview button at the bottom of the window. The Apply button allows you to see how the changes will appear with your new setup.




To then install the extension, you need to compile the project. Here is how to compile them.




After compiling the project, you have to enter the directory on the project and choose the **open folder** inside the file explore. The project location's Windows file explorer will be opened as a result. We will now search this path for the file that is contained in the bin>Debug > CustomTheme.vsix directory. The file's name will be derived from the project's name. Since the project in our example has the name CustomTheme, as seen in this image, you will discover the file if you created the project under a different name. You may post this file to Visual Studio Marketplace or distribute it to others to share the theme.




# Review Question

- How do you work with extensions?
- How do you set an extension up?
- How do you create a new theme?

# Summary

This chapter has shown us how extensions are a means to increase Visual Studio's capability, always enhancing user experience and reducing development time. We evaluated a few add-ons that fundamentally altered

the functionality of the IDE, including the ability to preview colors in CSS files, convert tabs to icons, and produce themes for sharing.

Additionally, we went through how to use the extensions tool and Visual Studio Marketplace to find and install extensions. The usage of an extension allowed us to design a new custom theme that can be shared with others after carefully examining how extensions are often configured.

# CHAPTER 14
# USING POPULAR EXTENSIONS

Now in the previous chapter, we took you through how to use extensions from the visual studio and its third-party vendors. However, in this chapter, we'll examine various open-source add-ons that might boost productivity and enhance our Visual Studio experience. The following sections cover the installation and evaluation of the extensions:

## Technical Requirements

### Adding HTML snippet Pack

Earlier in this book, we went through the ways to use code snippets and how they can be helpful when we want to import, create and remove snippets with the snippet manager.

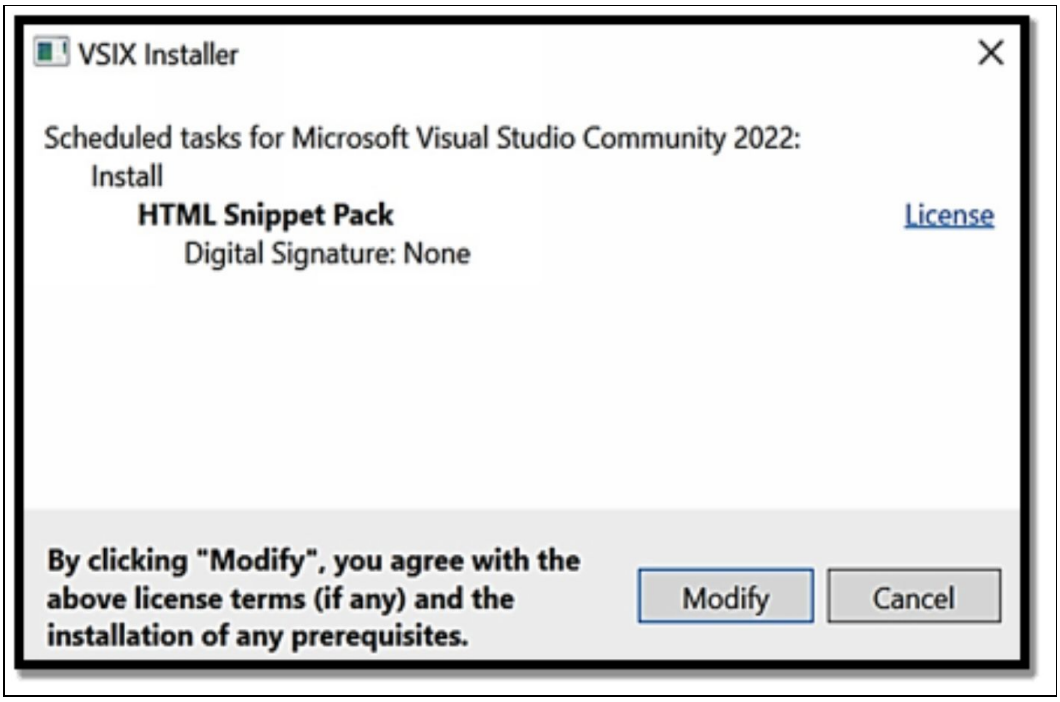
There are a few extensions we can use from the extension marketplace. Simply go to https://marketplace.visualstudio.com/ and enter a snippet inside the search bar.

**HTML Snippet Pack** is one of the most well-liked snippet collections for web developers. With the help of this addon, you may code HTML more quickly by producing code fragments and HTML components after entering a few characters in the editor. Installing this extension will allow us to simply generate HTML components using snippets.

## Downloading and installing HTML snippet pack

Go to the visual studio and enter **extensions** then choose to **manage extensions** then enter the HTMlsnippet inside the search bar. choose **HTML snippet pack** and select **download.**


When the installation is finished, we will get a message telling us to restart Visual Studio by closing and reopening it. Once Visual Studio has been reopened, the extension installation request will be presented. Select **Modify**.




You are going to then see the progress bar showing the installation progress. When It is done it says, **modification complete.**

# Using HTML snippet pack

To use the HTML Snippet Pack extension, we must browse to an HTML file, so let's go to SPAProject then **ClientApp** next, enter **public** then choose index.html.

The term "article" can be entered in this document's body element.

Then click on the tab keys to then generate the HTML element for the article in the code part.

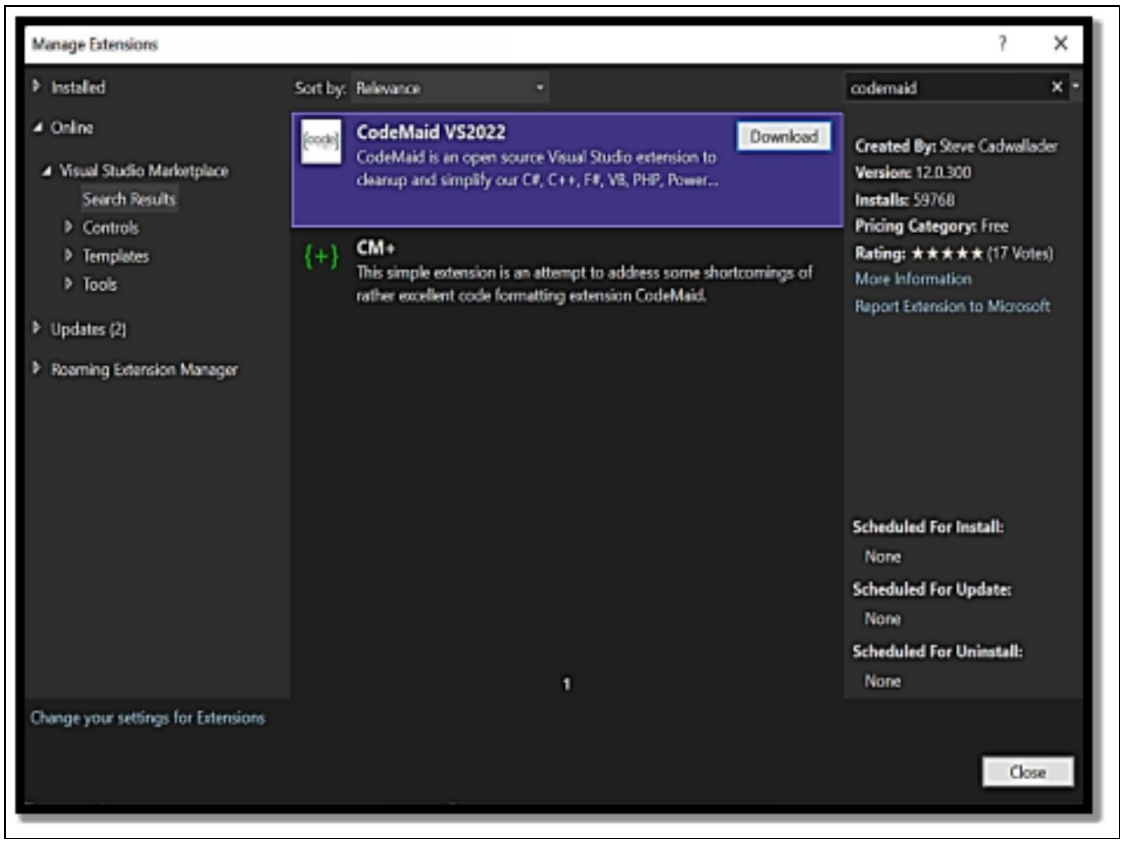
We can easily produce elements for li, ul, image, input, and nearly all of the other HTML standard elements, just as we did with the article element.

# Cleaning up code with the code maid.

We can streamline and organize our code with the help of the fantastic add-on CodeMaid. It supports C#, C++, F#, PHP, VB, PowerShell, R, XAML, JSON, XML, ASP, HTML, LESS, CSS, SCSS, JavaScript, and TypeScript and is free to use.

# Installing code maid in visual studio 2022

To install the code maid, enter the **extensions** then choose to **manage extensions** and enter code maid inside the search bar and choose **code maid VS2022**




Search for code maid, then click Download to download the extension. When we start Visual Studio the following time, the extension will already be installed, according to a notification we receive. Restart Visual Studio with SPAProject after closing it. You can choose Modify, then wait for the installation to finish before doing so

Then you are going to see another option inside the **extension's** menu with the functionalities and the settings that are related to the code maid.

# Using CodeMaid

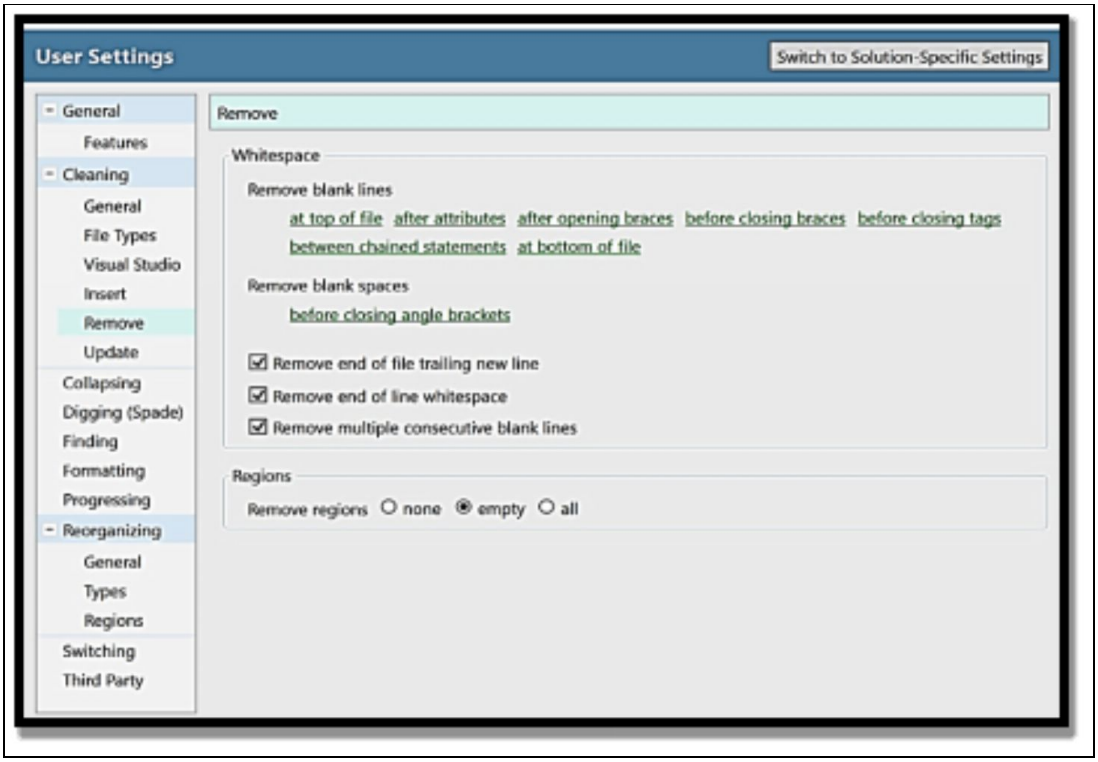
If all of the options are disabled, we can still use the **cleanup all code** menu to clean up the project with the default settings. You then are going to get a confirmation message before the process. You should select **yes** and

continue. Once you confirm it, code maid then analyzes the file and looks for the white spaces and the empty lines, sot lines, etc. the progress bar tells you all about it.

All of the files will be saved after the cleaning is finished. While we may open a file to view changes, the Git integration in Visual Studio makes it simple to identify discrepancies. These only functions if the project is already linked to a Git repository. To view the changes, open Program.cs and choose the Git Changes tab.

Inside the program.cs, you are going to see two lines with space in the if conditional statement code maid has been removed.

Depending on our needs, we may enable or disable specific options in CodeMaid. To specify the circumstances in which CodeMaid can delete code, go to Extensions and choose CodeMaid then select Options and click the Remove section.




Additionally, there are other choices in CodeMaid for eliminating blank lines and gaps. All of the settings are active by default, but you may click

on the elements you wish to turn off to configure CodeMaid following the requirements of your project.

# Compiling web libraries with web compiler

To acquire the production version of your web project if TypeScript, LESS, or Sass are being used, you must pre-compile the code so that the browser can understand every line of code. Keep in mind that the browser can only interpret JavaScript, HTML, and CSS. You can accomplish this quickly and view the precompiled code right away by using the Web Compiler in Visual Studio.

## Installing web compiler

Enter **extension** then choose **manage extension** and enter **web compiler** into the search bar. then from here, it is the same step with the adding HTML snippet pack part to finish the installation. The new extension is then completed after you open and close the visual studio.

## Using web compiler

Any JavaScript file in the project may be browsed to and selected to utilize Web Compiler; after right-clicking, a new option will appear in the menu. Let's choose **ClientApp** and then aspnetcore-react.js as an illustration.

We may compile the file and create a new version using ECMAScript 2009 (ES5), a JavaScript specification that enables us to support older browser versions, using the **Web Compiler** then select **Compile** file option.

The aspnetcore-react.es5.js file has a new syntax but still contains the same functionality as aspnetcore-react.js.

You are going to see some codes that represent the first 20 lines.

The code for how the backend asp.net application and the React application communicate is included in the aspnetcore-react.js file. Because this file makes use of the most recent JavaScript capabilities, Web Compiler must convert the code to an older version to support older browsers and improve library compatibility.

Then you are going to see some codes that represent the first 20 lines aspnetcore-react.es5/js

A file named compilerconfig.json will include each file that is added to Web Compiler. This file is connected to Visual Studio's compilation process. This implies that the project may be built and published normally, and the related files will be created automatically.

Each configuration is a JSON object that contains two properties – **inputFile** is the location of the source file to compile, and the **output file** has the location of the file generated by the Web Compiler.

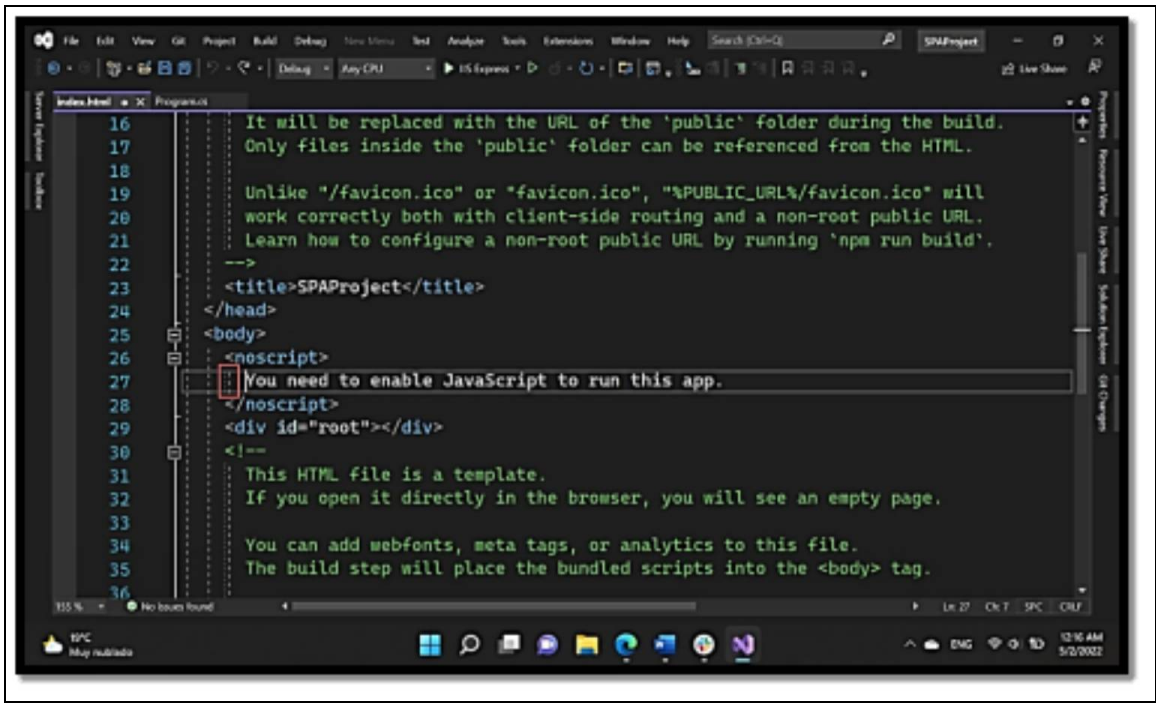
# Identifying white spaces with indent guides

The indent guides are useful and help us to look for the extra white spaces and indentations inside the structure of the code.

## Installing instant guides

To install indent guides, enter **extensions** and choose **manage extensions** and enter **indent guides.**

To get this extension, click Download. To install it, shut and reopen Visual Studio. To finish the installation, proceed with the instructions from the Adding HTML Snippet Pack section. After the installation is finished and Visual Studio is running once more, visit the index.html page to view updated guidelines that demonstrate how to use tabs and white space to separate text from text editor objects.




The formatting of our files may be much improved with the help of this tool, which works with all of the programming languages that Visual Studio supports. We can change a few other variables to suit our interests. Select Tools then Options and go to Indent Guides from the menu. Numerous choices to alter the look, behavior, and highlights are available there, along with options to create a quick start or default setup.

# Review Question

- How do you add an HTML snippet?
- How do you install code maid?
- How do you look for empty spaces?

# Summary

Using the Visual Studio Manage Extensions option, we can access the marketplace for Visual Studio, which contains a wide variety of extensions. To increase our efficiency when coding on HTML files, we may utilize HTML Snippet Pack, which we can obtain in the Studio Marketplace.

You now understand how to use CodeMaid to debug and clean up the code in your projects. For all the files in your project, you may effortlessly delete

blank lines and white spaces and design your validation format. Additionally, you learned how to set up Web Compiler, which compiles and converts web files like JavaScript scripts or libraries like LESS and Sass into general code that a browser can easily understand.

# CHAPTER 15
# LEARNING KEYBOARD SHORTCUTS

There are a few shortcuts to boost your productivity and perform some common actions inside the IDE and the source code. There are some default shortcuts. In this chapter, we are going to be taking you through some of the most important default shortcuts on Visual studio.

The keyboard shortcuts are crucial while working with Visual Studio so that we can complete tasks fast. This will save you time on monotonous operations like formatting a whole source code file or renaming a class member.

Even if you use them slowly at first, you must put these keyboard shortcuts into practice and start using them gradually. You'll notice that as time passes, you'll carry them out instinctively without the need for any kind of visual guidance.

## Technical requirements

It is significant to remember that various keyboard mapping techniques might entirely alter the shortcut keys. Additionally, keyboard shortcuts may be changed by extensions like ReSharper. You need to have a Default keyboard mapping scheme, which can be chosen from the Tools then choose **Options** and select **Environment** then **Keyboard** menu, to correctly use all of the shortcuts demonstrated in this chapter.

## Shortcuts for use in source code

Even the most seasoned developer may struggle when dealing with thousands of lines of code while working with source code. Because of this, using Visual Studio's robust search features to search and move between lines of code is a great idea.

The same is true for jobs like editing and revising code, not to mention testing and debugging in complex projects.

Working with shortcuts that provide you quick access to these tools should thus be a top goal in your development career. Check out these keyboard

shortcuts.

# Shortcuts for searching and navigating through source code

When working on a project with only one class, finding members of that class fast could seem like a straightforward process. However, if you work on projects with hundreds of classes or even many projects inside the same solution, you might not think so.

**The following keyboard shortcuts are quite useful in situations like this.**

- **Ctrl + Q:** Shows the **Visual Studio Search** window
- Pressing Ctrl and T brings up the **Go to All** tool.
- To navigate backward across open documents in the current session, press the keys Ctrl and -.
- To advance through opened documents in the current session, use Ctrl + Shift + -.
- **F12:** Opens the **class definition** page.
- You may see and change a class's code from the code file you are creating by pressing Alt + F12, which opens a pop-up window.
- Ctrl + F12 navigates to a class member's implementation.
- When the error list window is displayed and there are many errors reported, pressing Ctrl + Shift + F12 advances to the next error line.
- **F8:** Moves ahead in the current window's results list.
- Shift + F8 navigates backward via the current window's results list.
- With these shortcuts, you can navigate through files, members, and the result and perform tasks without taking your hands away from the keyboard.
- Shortcuts for editing and refactoring
- Deletes selected lines using Ctrl + Shift + L.
- **Ctrl + Shift + V:** Displays the contents of the buffer ring and enables pasting (it refers to the history of elements that have been previously copied)
- Using the keyboard shortcuts Ctrl + K and Ctrl + D, the entire page is styled.

- Ctrl + K and Ctrl + F only apply style rules to the document's chosen lines.
- Use the keyboard shortcuts Ctrl + K and Ctrl + S to enclose code between typical phrases like cycles (while, for, and so forth), control statements (if, switch, and so forth), or code sections (#region).
- Pressing the keys R and R renames a member.
- Pressing the keys R and E together create a property for a class field.
- **Ctrl + F:** Searches the code for a given text.
- Selects all of the lines in the current file with Ctrl + A.
- **Ctrl + S:** Saves the current file's pending modifications.
- All open files' pending modifications are saved by pressing Ctrl + Shift + S.
- Here are some of the commands that help you apply changes to the source code.
- Enter with Alt to see fast actions.
- **Ctrl + K and Ctrl + I:** Retrieves information about a class member.
- Pressing the keys Ctrl and K and C will comment numerous lines in the source code.
- **Ctrl + K and Ctrl + U:** Uncomment numerous lines in the source code that you have selected.
- Ctrl + Shift +.: Enlarges the currently open file.
- The current file is zoomed out by pressing Ctrl + Shift +,
- Ctrl + Up Advances a code file's chosen lines.
- **Ctrl + Down:** Scroll down a code file's chosen lines.

# Shortcuts for debugging and testing

**This is another popular task that we do in the visual studio. Here are some of the shortcuts to perform the tasks**

- F5 launches the program in debug mode
- Ctrl + F5 launches the program without using debug mode
- When an application is running, Shift + 5 stops it. When you press Ctrl + Shift + F5, the project is rebuilt and a new debugging session is started.
- **F9:** Adds or takes away a breakpoint

- When debugging, F10 skips the execution of the code; F11 debugs the source code line by line.
- **Shift + F11:** Stops the method from being executed.
- Starts unit test execution in debug mode with Ctrl + R and Ctrl + A.
- Starts unit test execution without using debug mode with Ctrl + R, A.

# The most common shortcuts for use in the IDE

When you are working with a visual studio, it can be necessary to know how to move around.

**Here are some of the shortcuts that help you to optimize task performance.**

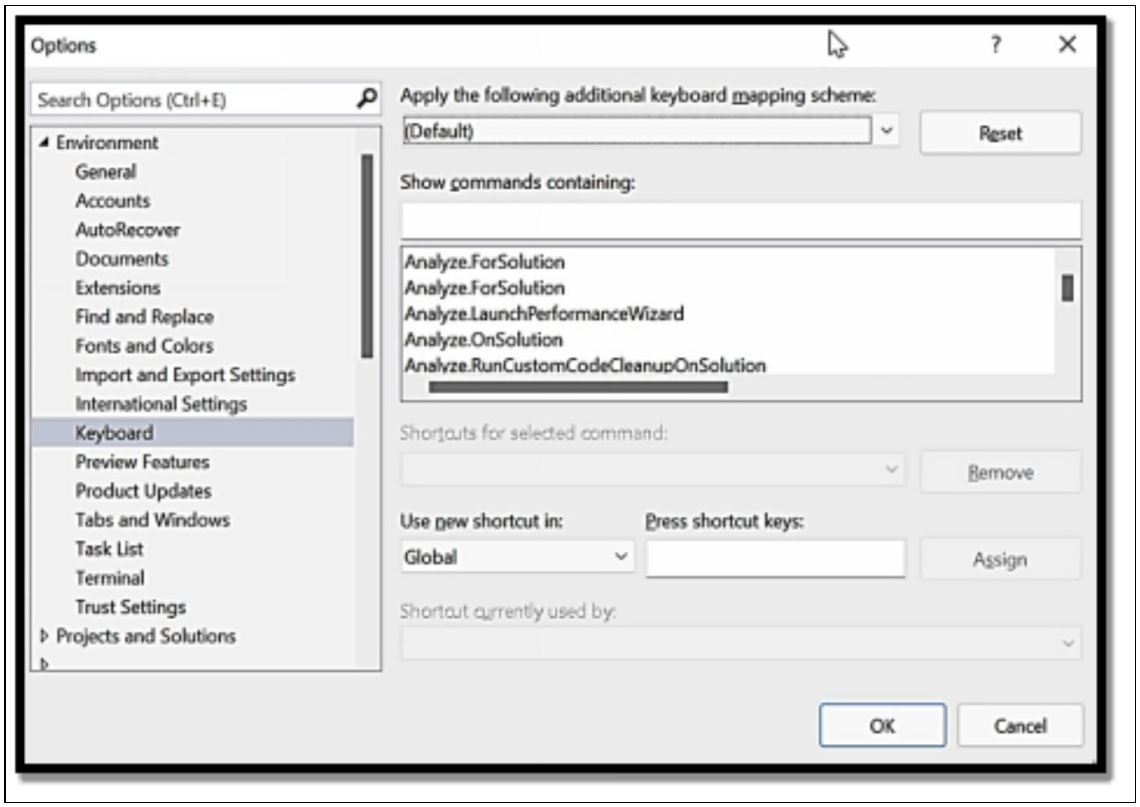
- If a file is open, we can easily pick it in the Solution Explorer box by pressing Ctrl + [+ S.
- Solution Explorer is opened by pressing Ctrl, Alt, and L.
- **Ctrl + Alt + O:** Launches the Output window.
- Error List window is opened by pressing Ctrl +, E.
- You may scroll back and forth between open panels' windows by using the keys Alt and F6.
- You may advance between windows on open panels by pressing Shift + Alt + F6.
- The current tool window may be closed by pressing Shift + Esc.
- **Ctrl + Shift + Tab:** Opens a separate window with the open documents, where you may choose the one that was last modified.
- Even when it's not the same session, scrolling up between open documents is possible by pressing Ctrl + Alt + Pg Up.
- Even when it's not the same session, scrolling down between open documents is possible by pressing Ctrl + Alt + Pg Dn.
- **Ctrl + and Ctrl + M:** Launches Team Explorer.
- The breakpoints window opens when Ctrl + Alt + B is pressed.
- Pressing F4 brings up the Properties window
- Ctrl + Tab: Opens a window with all the open documents, where you may choose the most recent one.
- **Shift + Alt + Enter:** This enables you to maximize the Visual Studio environment so you can focus on the active document. The same

shortcut may also be used to exit full-screen mode.

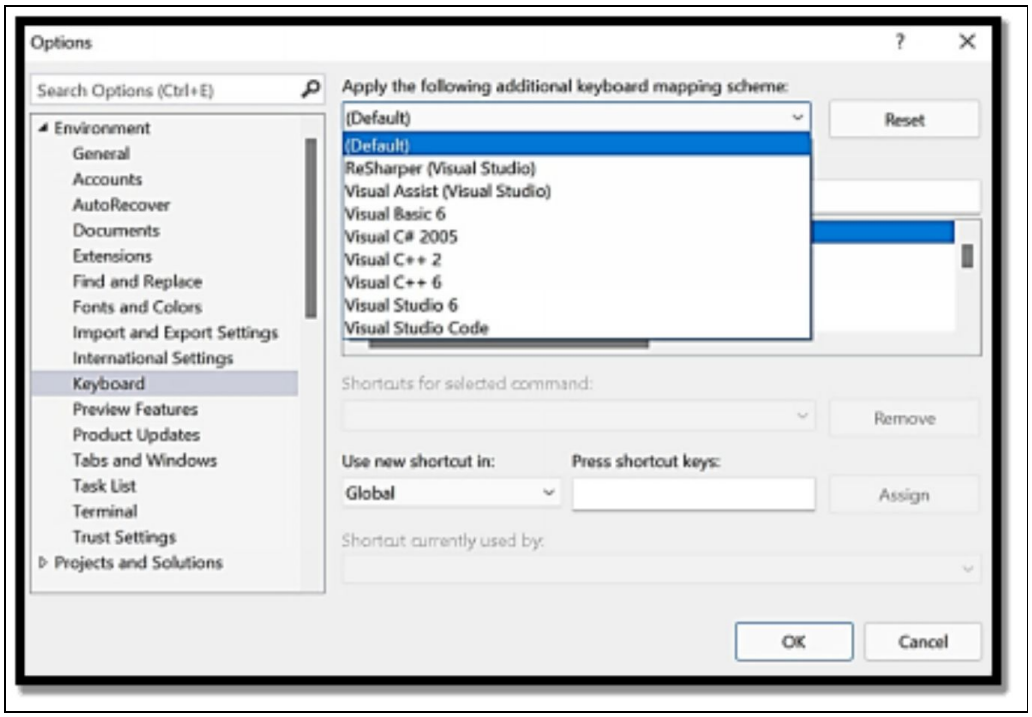
- You may advance between the many bookmarks that are a part of a project by pressing Ctrl + K + N.
- You may browse backward through the many bookmarks that are a part of a project by pressing Ctrl + K + P.

# Creating custom shortcuts

There are a few options to create your shortcut. First enter **tools** then choose **options** and select **environment then** keyboard and you are going to see all of the available shortcuts.




Visual Studio has a scheme for shortcuts where you may build them up according to the scenario as well as all the present shortcuts for all the features. By default, Visual Studio comes with a variety of keyboard layouts and keyboard shortcut settings.




Select the **Apply the following extra keyboard mapping scheme option**, then choose to Analyze to create a new shortcut. the command **RunDefaultCodeCleanUpOnSolution** This command carries out a procedure to clean up the code to enhance the format and get rid of extraneous code. The **Ctrl + Alt + Y** combination can be added to the Press shortcut keys option to set the shortcut for this command.

Then select **assign** and then choose **ok** to finish.

# Review Question

- What are shortcuts?
- What are the shortcuts to look for source codes?
- How do you create a custom shortcut?

# Summary.

In this chapter, we covered all the helpful shortcuts that we may utilize while writing or carrying out an operation with tools or features in Visual Studio.

The ability to use shortcuts in your everyday work will enable you to become a more productive developer by reducing your reliance on the

keyboard for Visual Studio operations.

We also learned how to use a key combination to construct our shortcuts and automate routine chores in our projects.

# INDEX

**B**

## C

**D**

## E

## F



## G

## H



## I

## J



## K



## L

**M**

## N



## O



## P

## Q



## R



## S

## T

## U



## V

## W

**X**