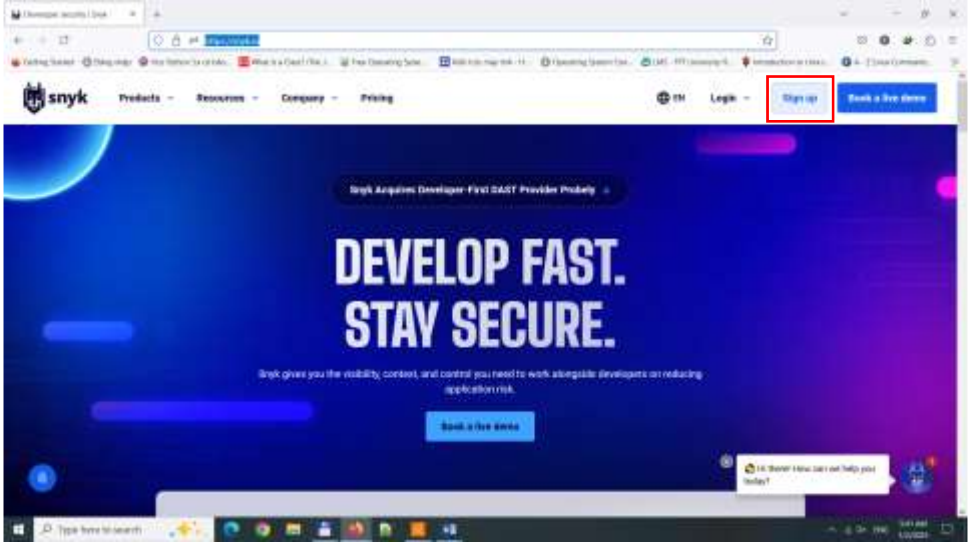
The Snyk quick user guide (C/C++)

Snyk is a developer security platform that enables application and cloud developers to secure their whole application — finding and fixing vulnerabilities from their first lines of code to their running cloud.

I. Sign up

1.1. Go to the Snyk website: https://snyk.io/ → **Sign up**



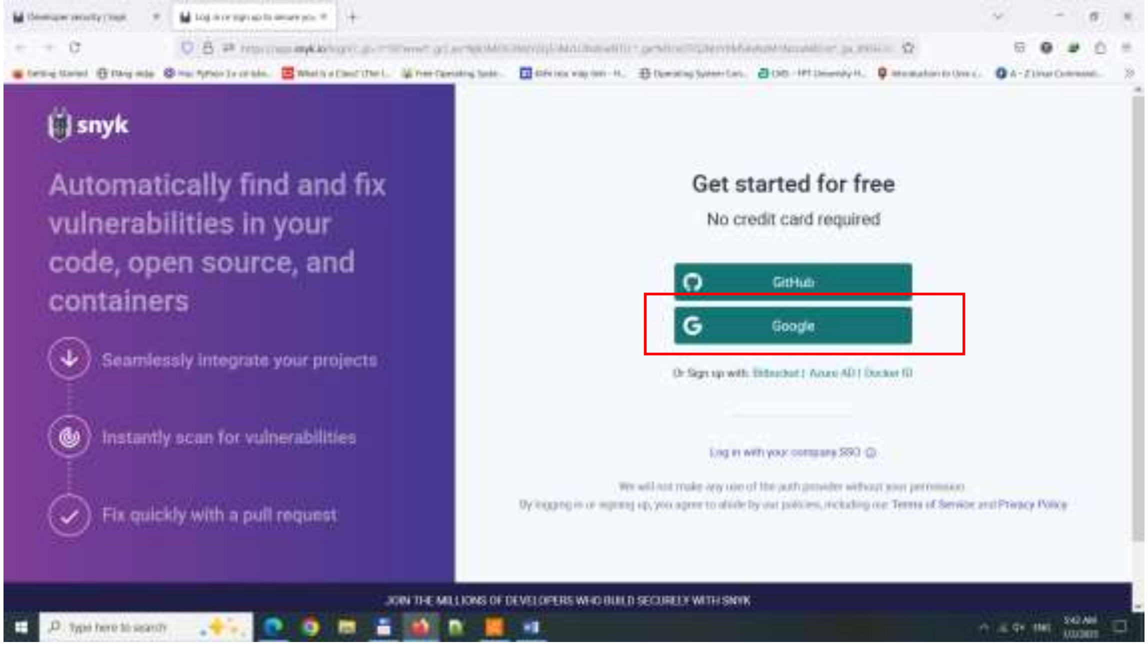
Snyk Acquires Developer-First DAST Provider Probely

DEVELOP FAST. STAY SECURE.

Snyk gives you the visibility, context, and control you need to work alongside developers on reducing application risk.

Hi there! How can we help you today?

1.2. Select sign up for an account. For example, select to sign up with a Google account.



Automatically find and fix vulnerabilities in your code, open source, and containers

Seamlessly integrate your projects

Instantly scan for vulnerabilities

Fix quickly with a pull request

Get started for free

No credit card required

Or Sign up with: Bitbucket | Azure AD | Docker ID

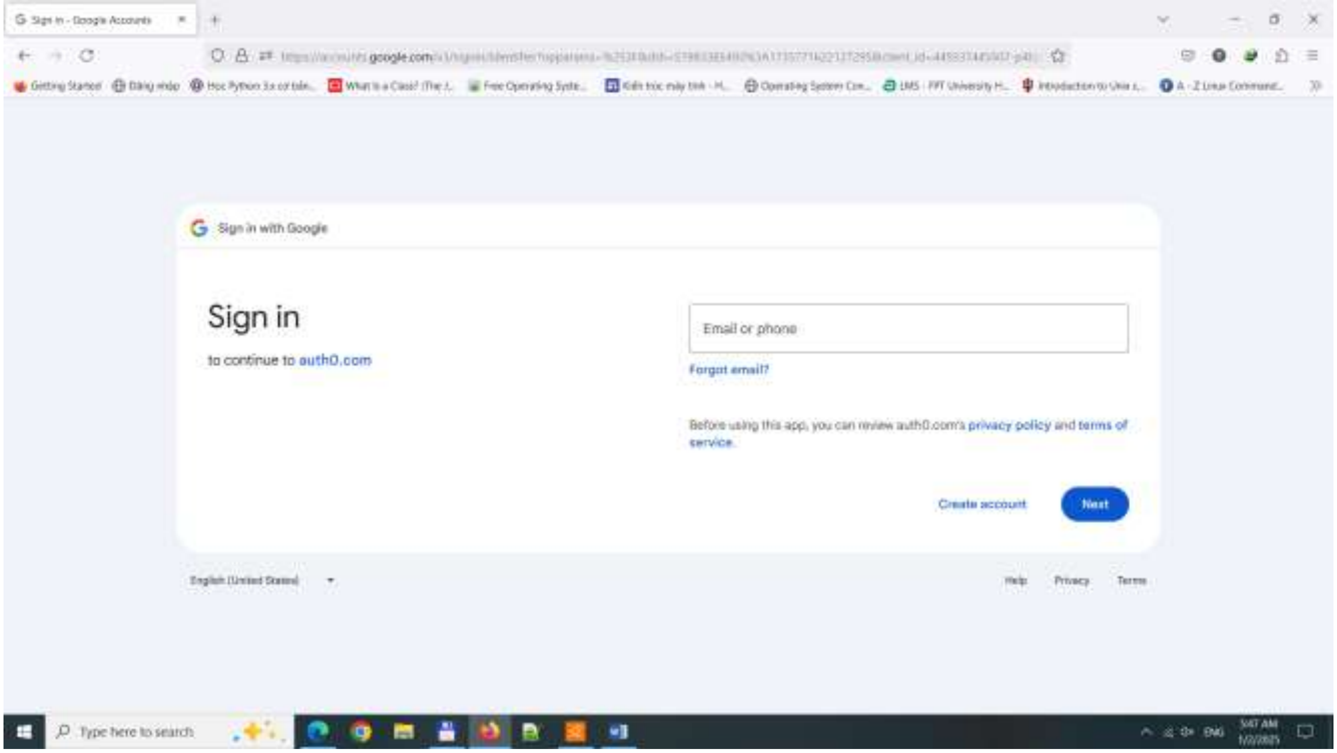
Log in with your company SSO

We will not make any use of the auth provider without your permission.

By logging in or signing up, you agree to abide by our policies, including our Terms of Service and Privacy Policy

JOIN THE MILLIONS OF DEVELOPERS WHO BUILD SECURELY WITH SNYK

1.3. Sign in the Google account. Choose **Continue** to sign up



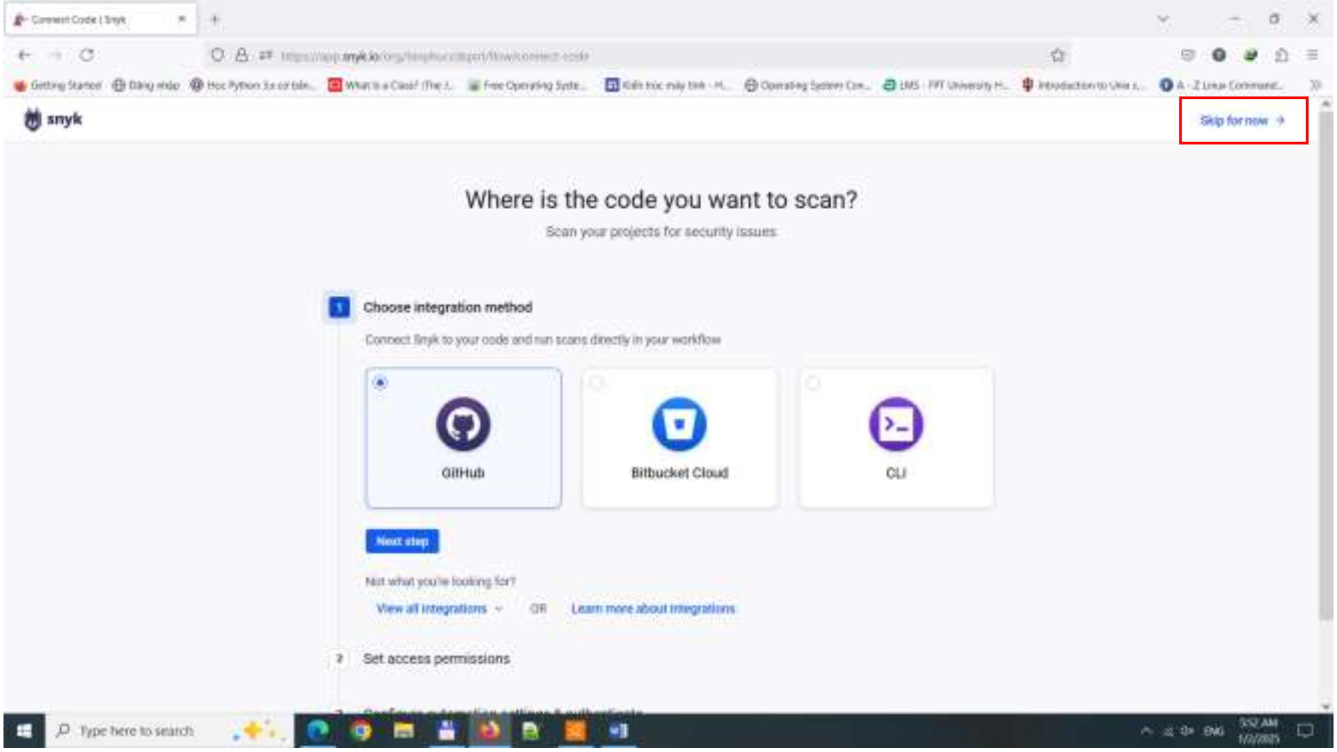
Sign in

to continue to auth0.com

Email or phone

Before using this app, you can review auth0.com's privacy policy and terms of service.

1.4. After Sign in the Google account. Choose **Skip for now**



Where is the code you want to scan?

Scan your projects for security issues

1 Choose integration method

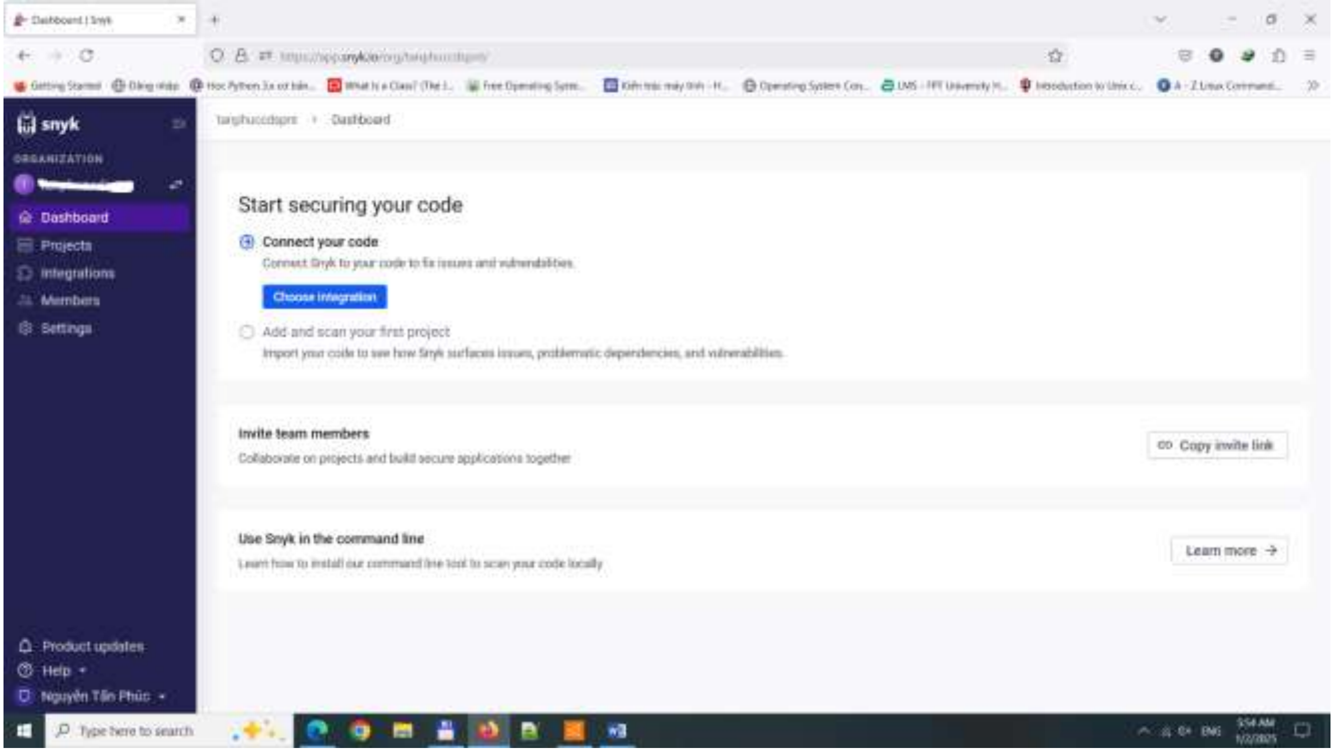
Connect Snyk to your code and run scans directly in your workflow

● GitHub ○ Bitbucket Cloud ○ CLI

Not what you're looking for?

2 Set access permissions

1.5. Dashboard of your account on Snyk



Start securing your code

Connect your code

Connect Snyk to your code to fix issues and vulnerabilities.

Add and scan your first project

Import your code to see how Snyk surfaces issues, problematic dependencies, and vulnerabilities.

Invite team members

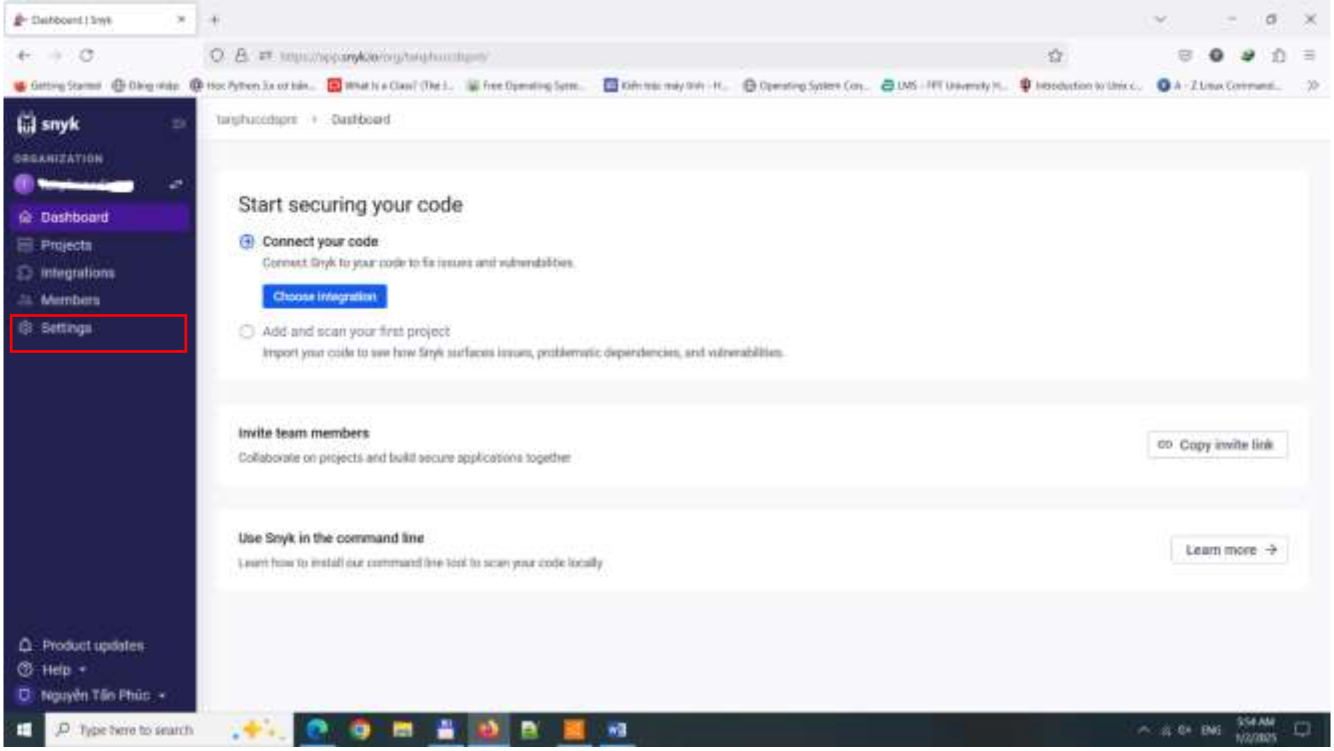
Collaborate on projects and build secure applications together

Use Snyk in the command line

Learn how to install our command line tool to scan your code locally

II. Configure

2.1. On your Dashboard → Choose **Setting**



Start securing your code

Connect your code

Connect Snyk to your code to fix issues and vulnerabilities.

Add and scan your first project

Import your code to see how Snyk surfaces issues, problematic dependencies, and vulnerabilities.

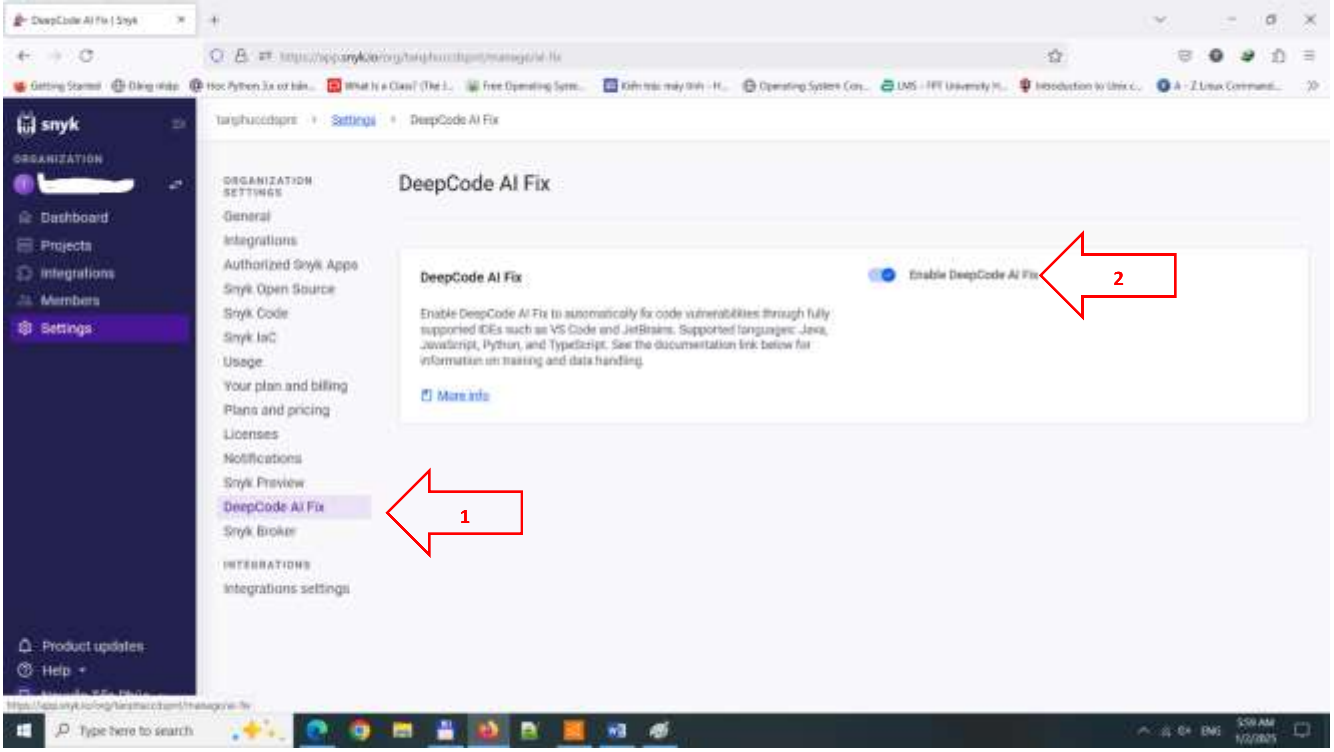
Invite team members

Collaborate on projects and build secure applications together

Use Snyk in the command line

Learn how to install our command line tool to scan your code locally

2.2. On **Setting → DeepCode AI Fix → Enable DeepCode AI Fix**



DeepCode AI Fix

DeepCode AI Fix

Enable DeepCode AI Fix

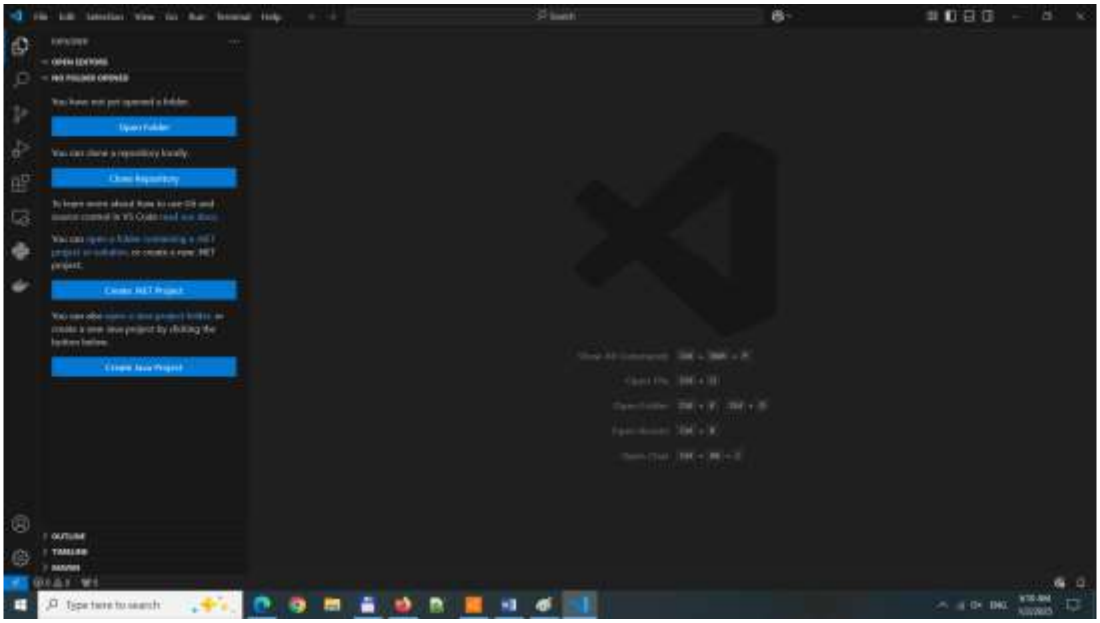
Enable DeepCode AI Fix to automatically fix code vulnerabilities through fully supported IDEs such as VS Code and JetBrains. Supported languages: Java, JavaScript, Python, and TypeScript. See the documentation link below for information on training and data handling.

More info

III. Using Snyk with Visual Studio Code IDE (VS Code)

After installing VS Code and opening VS Code

See installation and configuration instructions in part **IV. Visual Studio Code IDE (VS Code)** below



You have not yet opened a folder.

You can clone a repository locally.

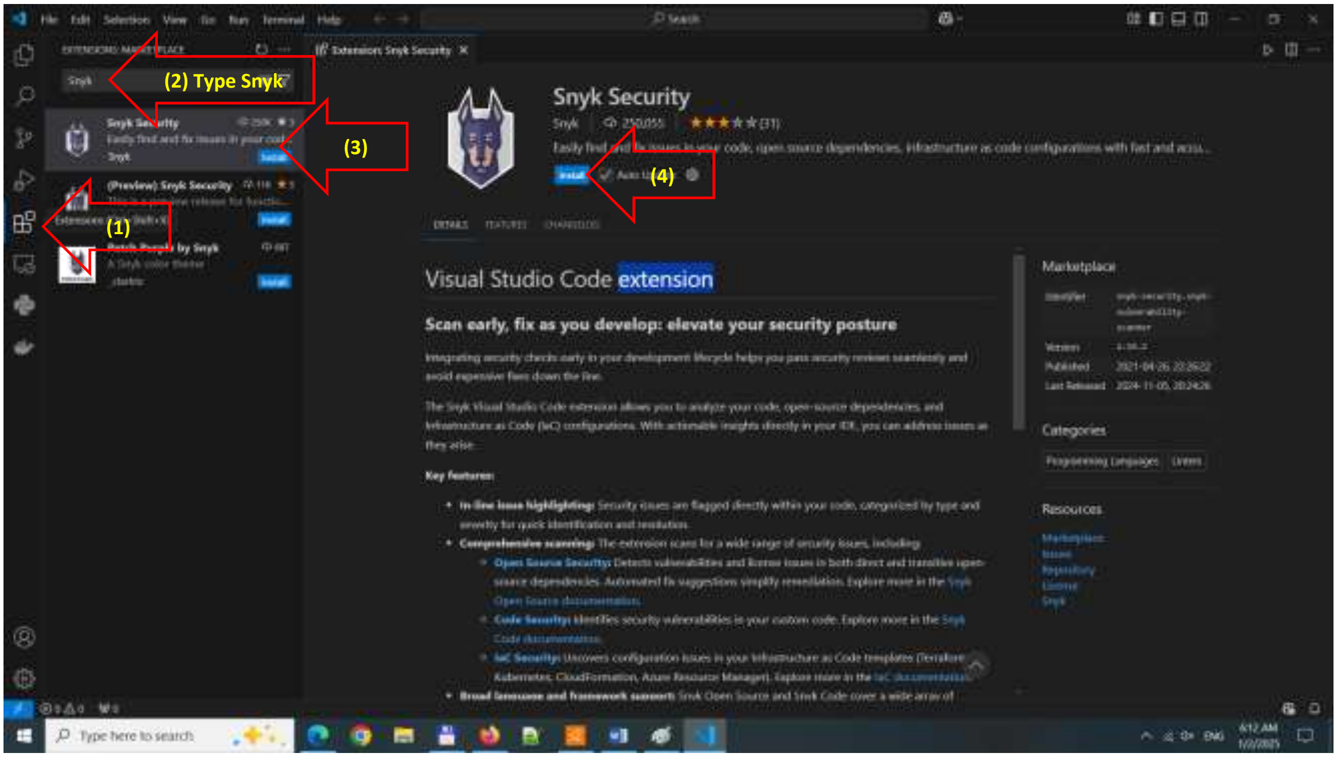
To learn more about how to use Git and source control in VS Code read our docs.

You can open a folder containing a .NET project or solution, or create a new .NET project.

You can also open a Java project folder, or create a new Java project by clicking the button below.

3.1. Install Snyk extension

- Choose **Extension** or press keys: **Ctrl + Shift + X**
- Type **Snyk** to filter
- Choose **Snyk Security**
- Choose **Install**



Snyk Security

Snyk | 250,055 | ★★★★★ (31)

Easily find and fix issues in your code, open source dependencies, infrastructure as code configurations with fast and accu...

Visual Studio Code extension

Scan early, fix as you develop: elevate your security posture

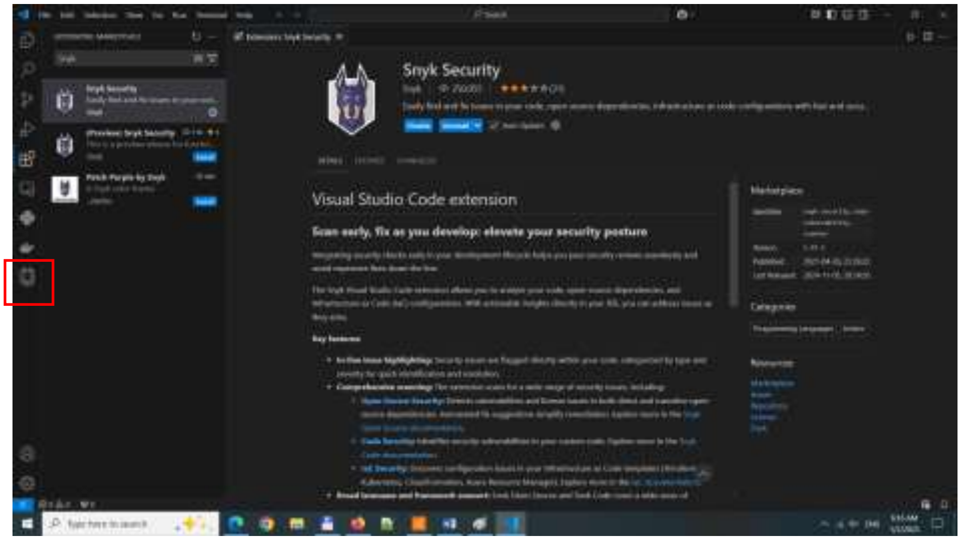
Integrating security checks early in your development lifecycle helps you pass security reviews seamlessly and avoid expensive fixes down the line.

The Snyk Visual Studio Code extension allows you to analyze your code, open-source dependencies, and Infrastructure as Code (IaC) configurations. With actionable insights directly in your IDE, you can address issues as they arise.

Key features:

- In-line issue highlighting: Security issues are flagged directly within your code, categorized by type and severity for quick identification and resolution.
- Comprehensive scanning: The extension scans for a wide range of security issues, including:
  - Open Source Security: Detects vulnerabilities and license issues in both direct and transitive open-source dependencies. Automated fix suggestions simplify remediation. Explore more in the Snyk Open Source documentation.
  - Code Security: Identifies security vulnerabilities in your custom code. Explore more in the Snyk Code documentation.
  - IaC Security: Uncovers configuration issues in your Infrastructure as Code templates (Terraform, Kubernetes, CloudFormation, Azure Resource Manager). Explore more in the IaC documentation.
- Broad language and framework support: Snyk Open Source and Snyk Code cover a wide array of

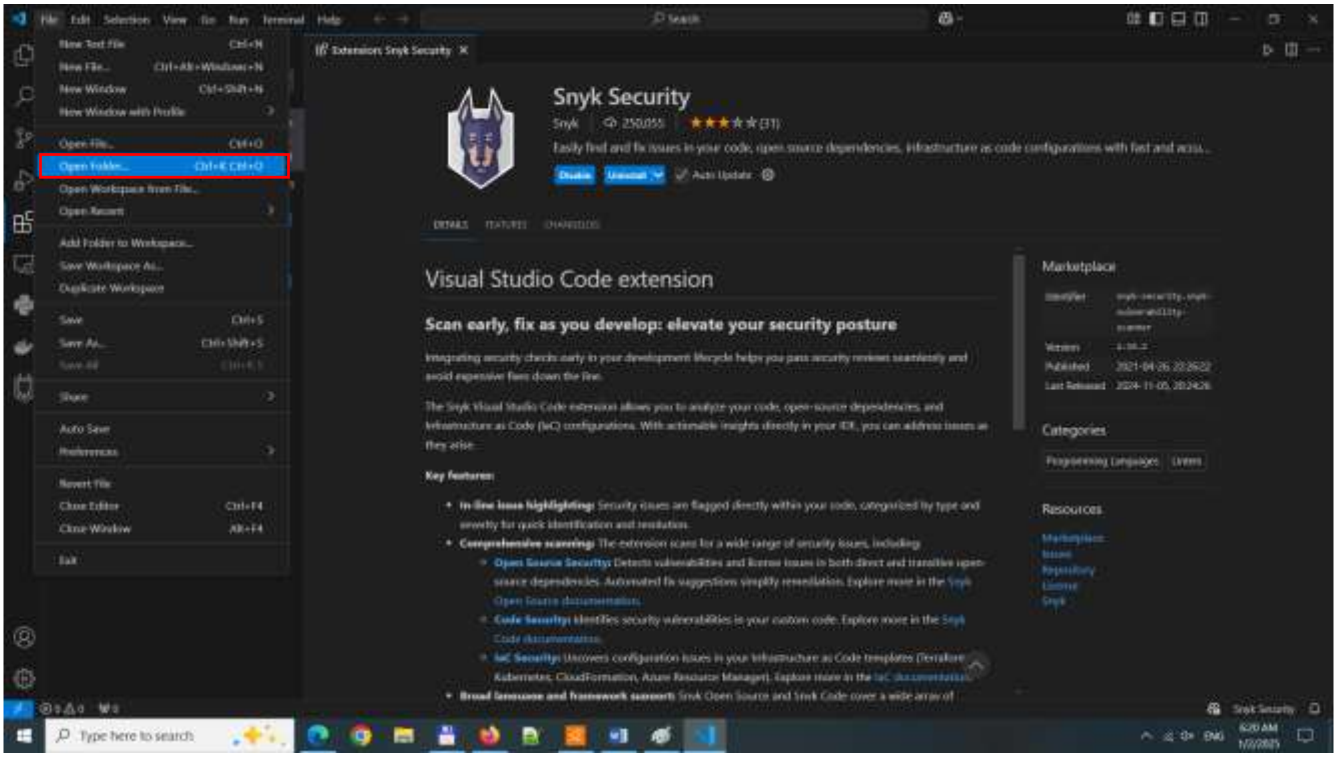
- After Install, the **Snyk icon** on the Activity Bar



Snyk Security
Visual Studio Code extension
Scan early, fix as you develop: elevate your security posture

3.2. Synk with C/C++

3.2.1. Open C/C++ project folder: Choose File → Open Folder or Ctrl +K



Easily find and fix issues in your code, open source dependencies, infrastructure as code configurations with fast and accu...

Visual Studio Code extension

Scan early, fix as you develop: elevate your security posture

Integrating security checks early in your development lifecycle helps you pass security reviews seamlessly and avoid expensive fixes down the line.

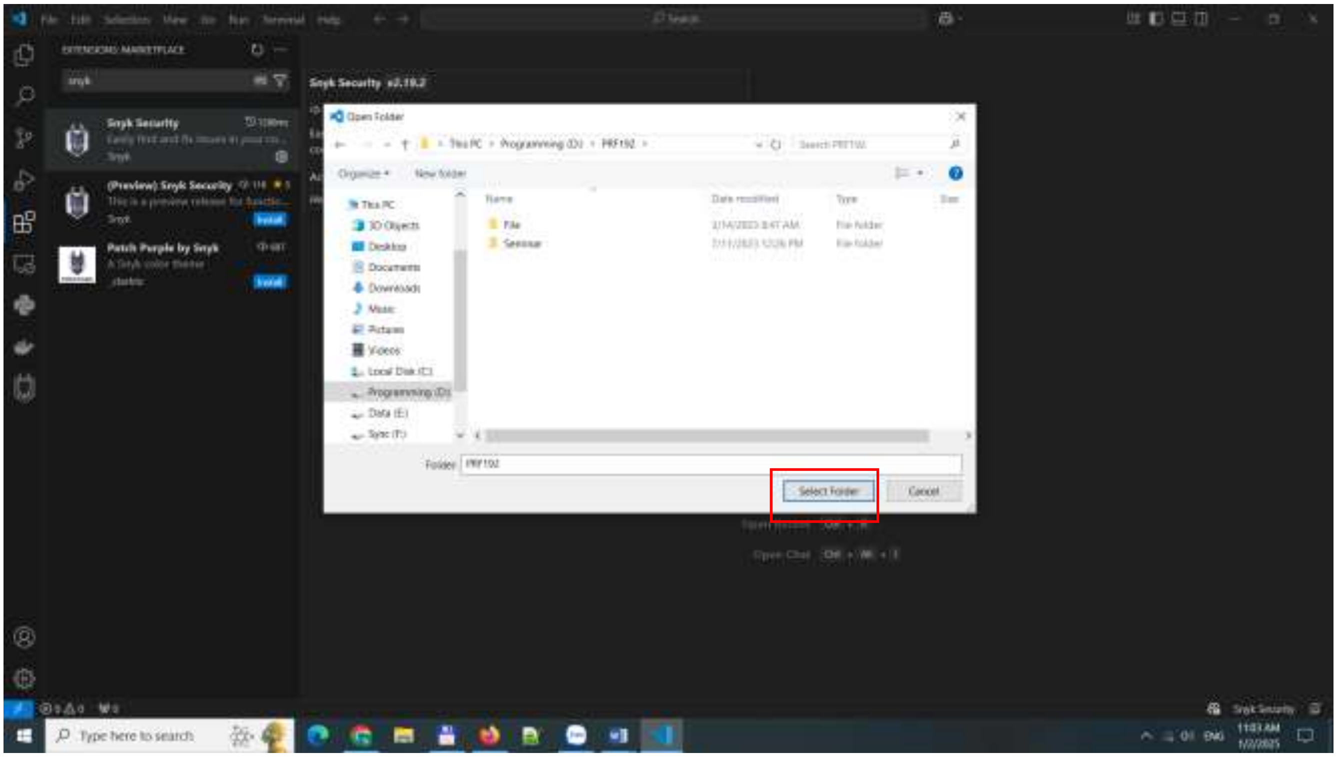
The Snyk Visual Studio Code extension allows you to analyze your code, open-source dependencies, and Infrastructure as Code (IaC) configurations. With actionable insights directly in your IDE, you can address issues as they arise.

Key features:

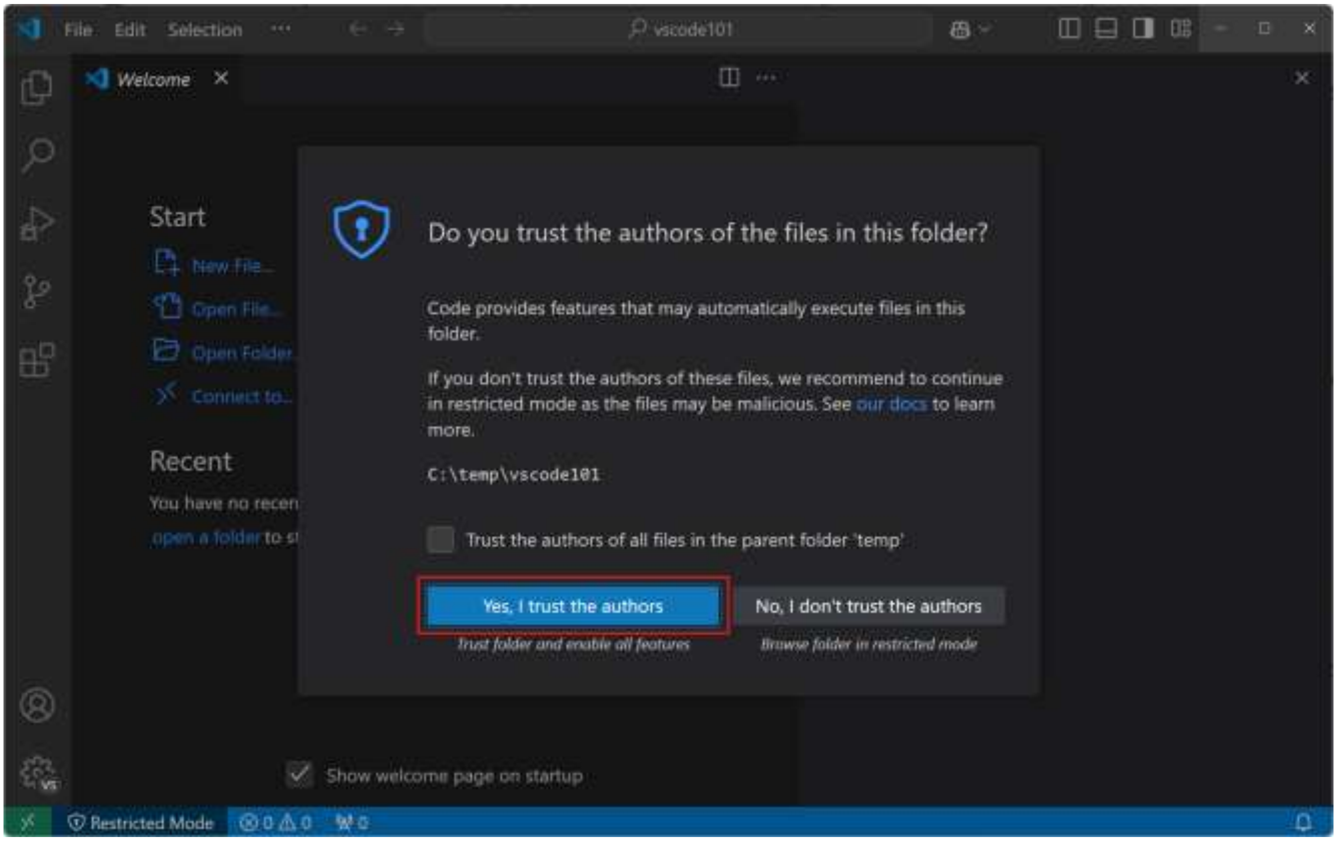
- In-line issue highlighting: Security issues are flagged directly within your code, categorized by type and severity for quick identification and resolution.
- Comprehensive scanning: The extension scans for a wide range of security issues, including:
  - Open Source Security: Detects vulnerabilities and license issues in both direct and transitive open-source dependencies. Automated fix suggestions simplify remediation. Explore more in the Snyk Open Source documentation.
  - Code Security: Identifies security vulnerabilities in your custom code. Explore more in the Snyk Code documentation.
  - IaC Security: Uncovers configuration issues in your Infrastructure as Code templates (Terraform, Kubernetes, CloudFormation, Azure Resource Manager). Explore more in the IaC documentation.
- Broad language and framework support: Snyk Open Source and Snyk Code cover a wide array of

Identifier	snyk-security.snyk-vulnerability-scanner
Version	2.19.2
Published	2021-04-26, 22:26:22
Last Released	2024-11-05, 20:24:26

After browser to C/C++ project folder: Choose **Select Folder**



If this folder opened the first time by VS Code, VS Code will display the Workspace Trust dialog. Then, select **Yes, I trust the authors** to enable all features in the workspace. Example, we opened folder c:\temp\vscode101



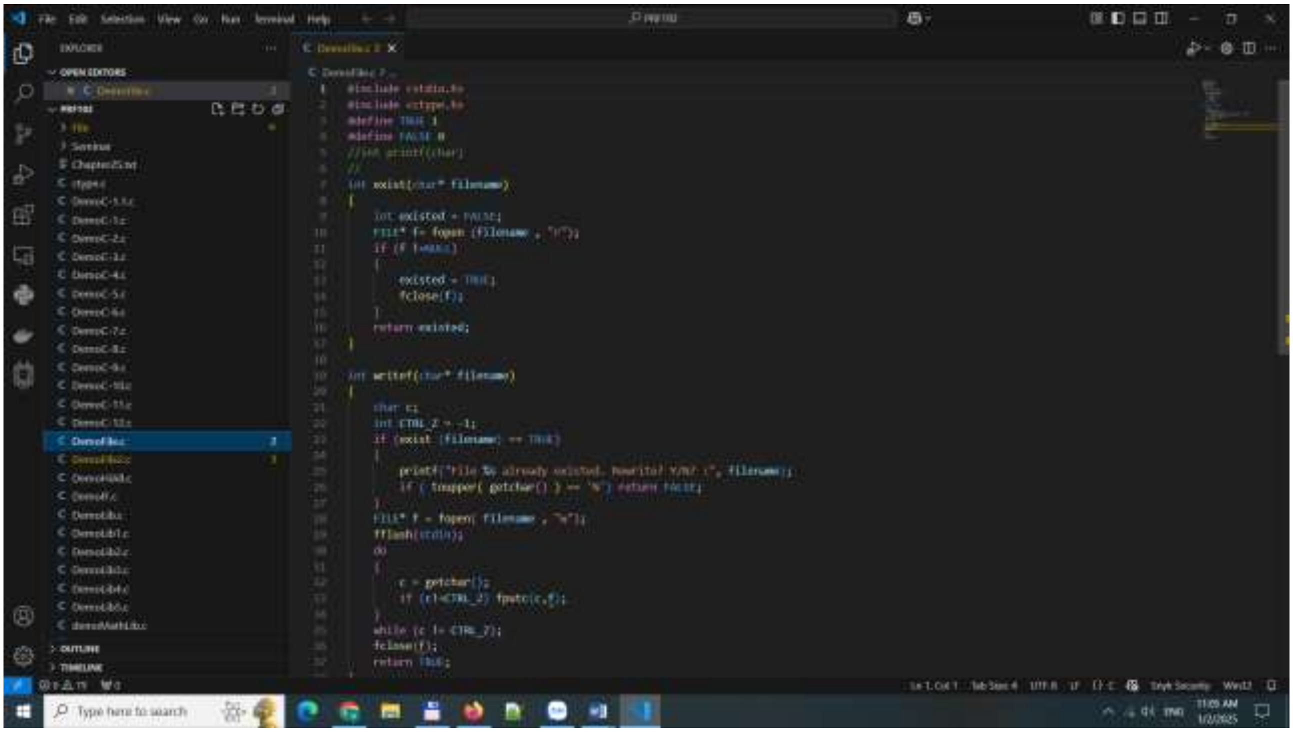
Do you trust the authors of the files in this folder?

Code provides features that may automatically execute files in this folder.

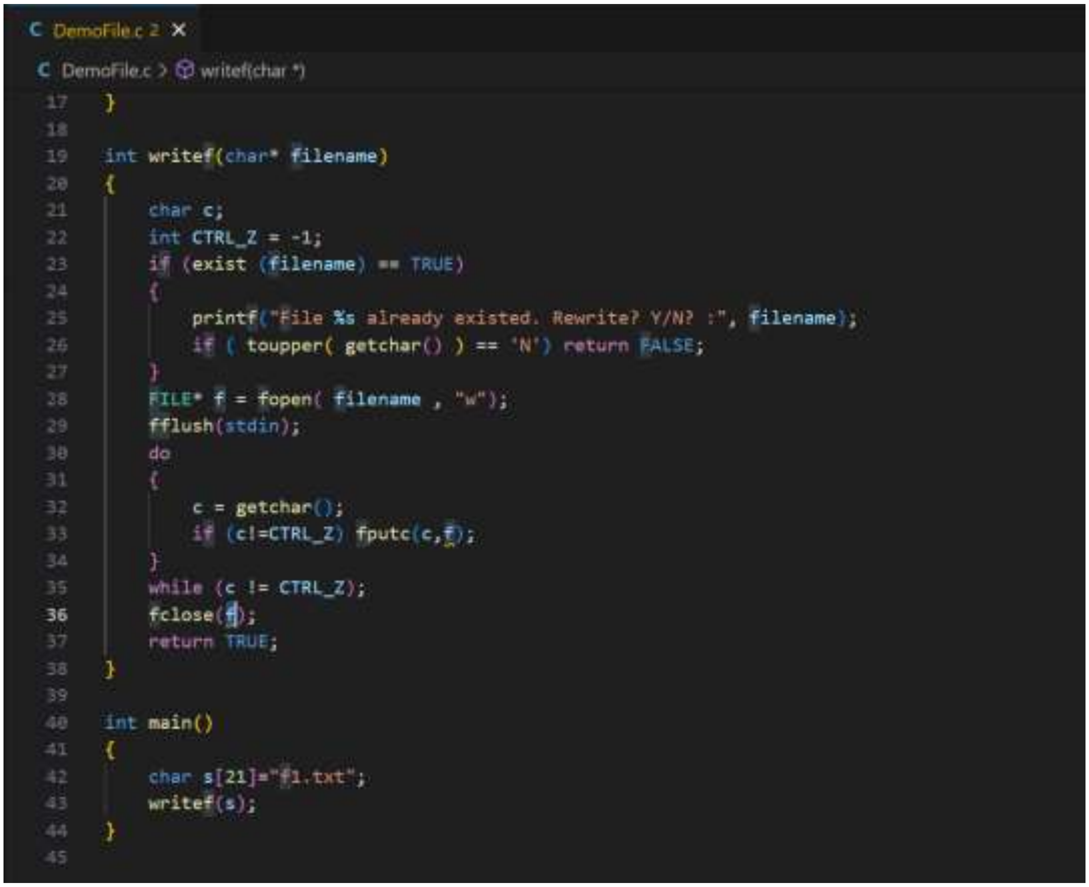
If you don't trust the authors of these files, we recommend to continue in restricted mode as the files may be malicious. See our docs to learn more.

C:\temp\vscode101

□ Trust the authors of all files in the parent folder 'temp'

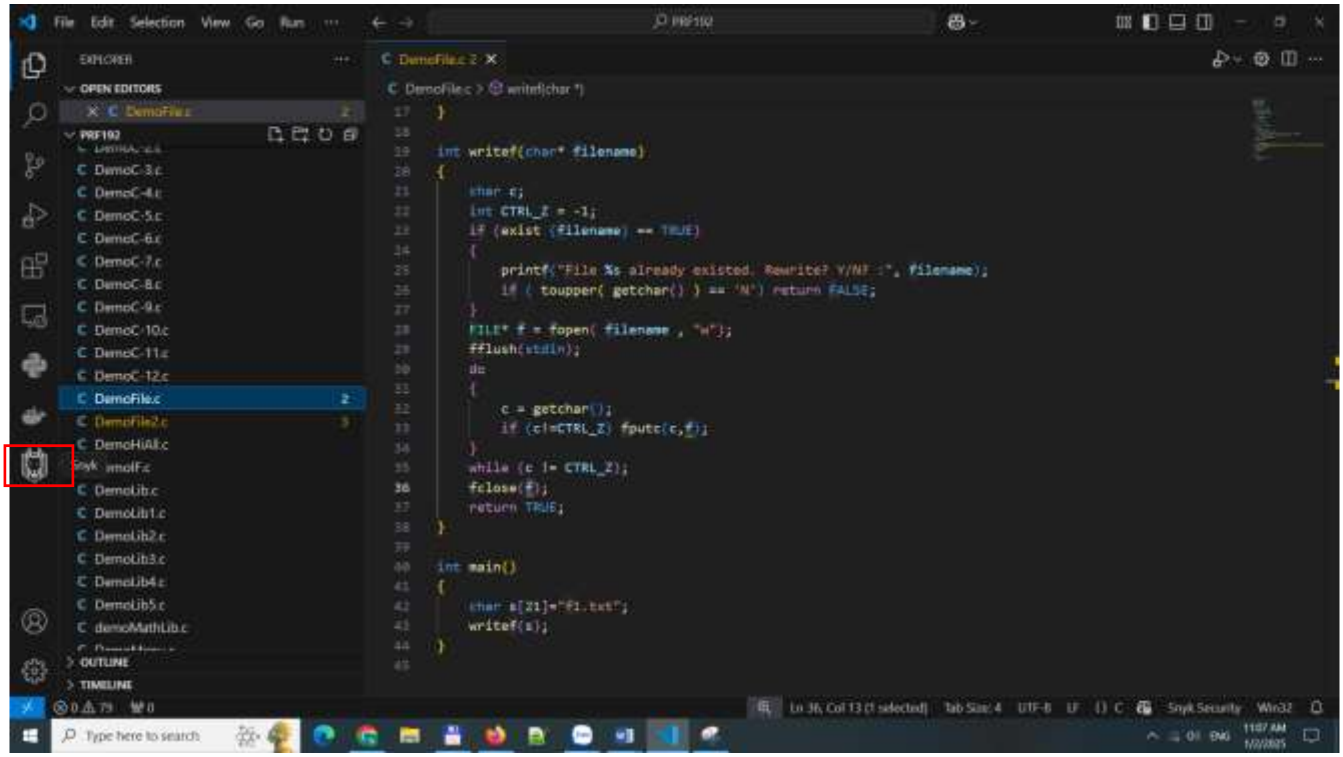


```
#include <stdio.h>
#include <ctype.h>
#define TRUE 1
#define FALSE 0
//int printf(char)
//
int exist(char* filename)
{
    int existed = FALSE;
    FILE* f= fopen (filename , "r");
    if (f !=NULL)
    {
        existed = TRUE;
        fclose(f);
    }
    return existed;
}

int writef(char* filename)
{
    char c;
    int CTRL_Z = -1;
    if (exist (filename) == TRUE)
    {
        printf("File %s already existed. Rewrite? Y/N? :", filename);
        if ( toupper( getchar() ) == 'N') return FALSE;
    }
    FILE* f = fopen( filename , "w");
    fflush(stdin);
    do
    {
        c = getchar();
        if (c!=CTRL_Z) fputc(c,f);
    }
    while (c != CTRL_Z);
    fclose(f);
    return TRUE;
```



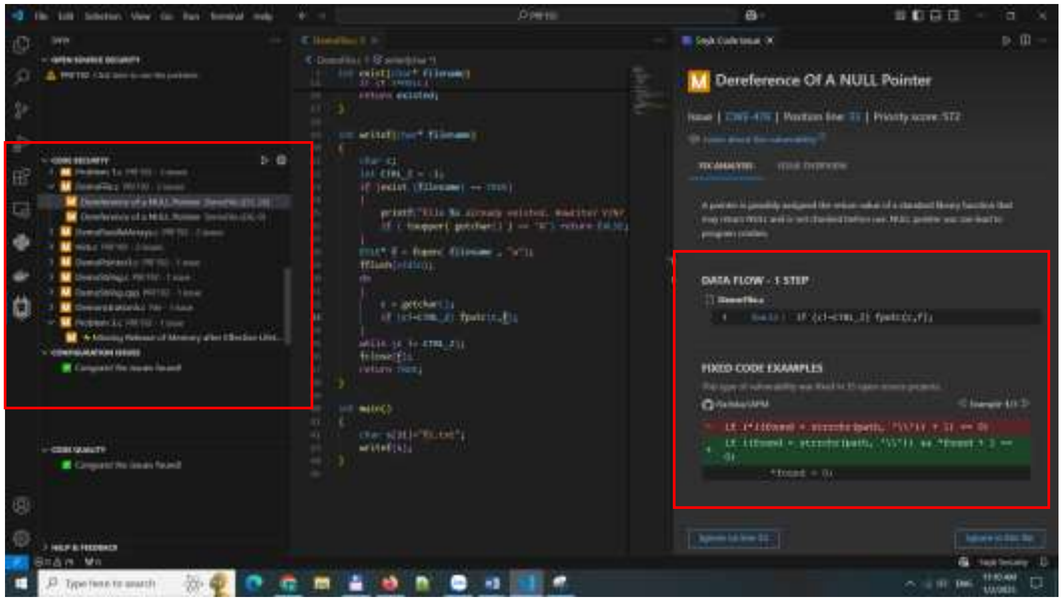
```c
}

int writef(char* filename)
{
    char c;
    int CTRL_Z = -1;
    if (exist (filename) == TRUE)
    {
        printf("File %s already existed. Rewrite? Y/N? :", filename);
        if ( toupper( getchar() ) == 'N') return FALSE;
    }
    FILE* f = fopen( filename , "w");
    fflush(stdin);
    do
    {
        c = getchar();
        if (c!=CTRL_Z) fputc(c,f);
    }
    while (c != CTRL_Z);
    fclose(f);
    return TRUE;
}

int main()
{
    char s[21]="f1.txt";
    writef(s);
}
```

3.2.2. Using Snyk to analyze code: Click **the Snyk icon** on the Activity Bar



```c
}

int writef(char* filename)
{
    char c;
    int CTRL_Z = -1;
    if (exist (filename) == TRUE)
    {
        printf("File %s already existed. Rewrite? Y/N? :", filename);
        if ( toupper( getchar() ) == 'N') return FALSE;
    }
    FILE* f = fopen( filename , "w");
    fflush(stdin);
    do
    {
        c = getchar();
        if (c!=CTRL_Z) fputc(c,f);
    }
    while (c != CTRL_Z);
    fclose(f);
    return TRUE;
}

int main()
{
    char s[21]="f1.txt";
    writef(s);
}
```

Snyk to analyze code report: ***Example, code security, DemoFile.c found 2 issues in line 33 and 36, error none check null pointer. On the Synk Code Issue, AI suggest 03 solution to fix it.***



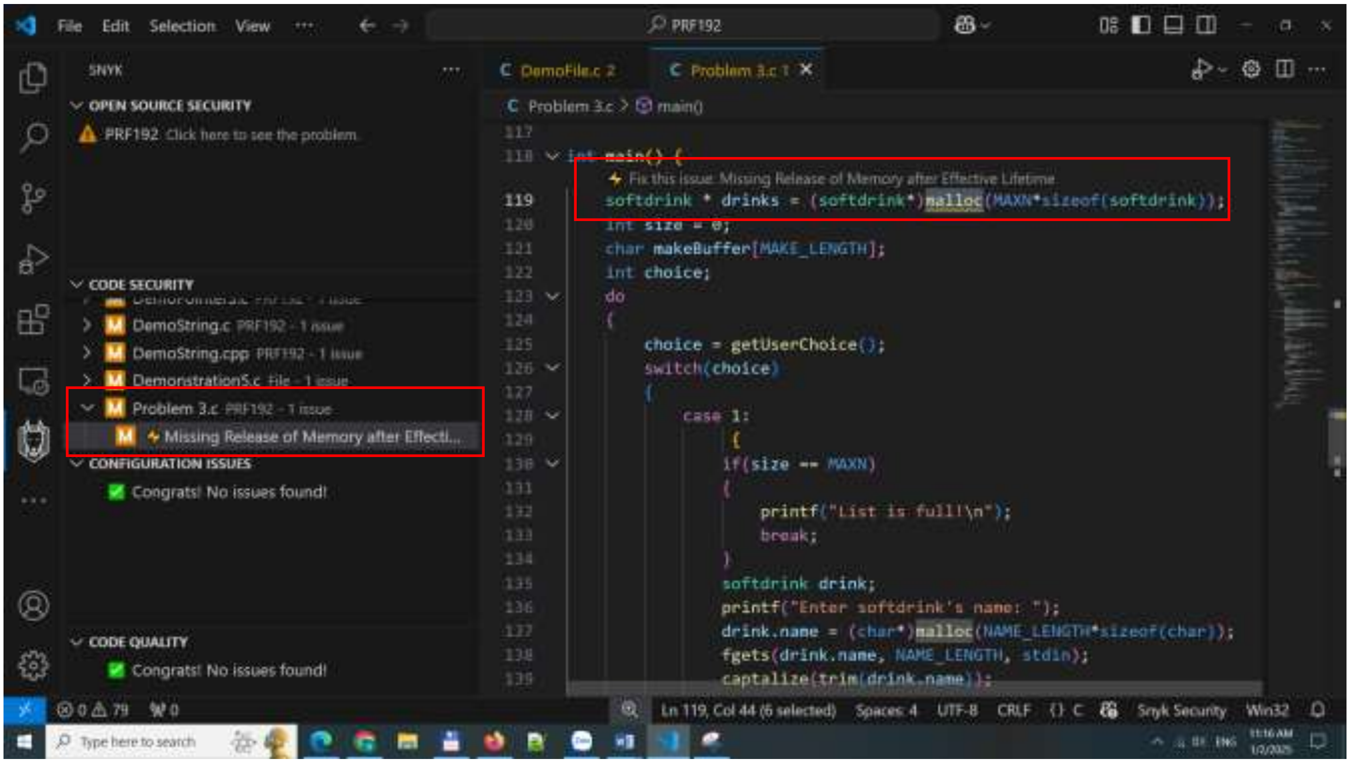
Dereference Of A NULL Pointer

DATA FLOW - 1 STEP

FIXED CODE EXAMPLES

3.2.3. DeepCode AI fix.

Additionally, in the project folder, in file Problem3.c, Snyk detected an error: Missing Release of Memory after Effective Lifetime at line 119.



SNYK

OPEN SOURCE SECURITY

⚠ PRF192 Click here to see the problem.

CODE SECURITY

- DemoString.c PRF192 - 1 issue
- DemoString.cpp PRF192 - 1 issue
- DemonstrationS.c File - 1 issue
- Problem 3.c PRF192 - 1 issue
  - Missing Release of Memory after Effecti...

CONFIGURATION ISSUES

✓ Congrats! No issues found!

CODE QUALITY

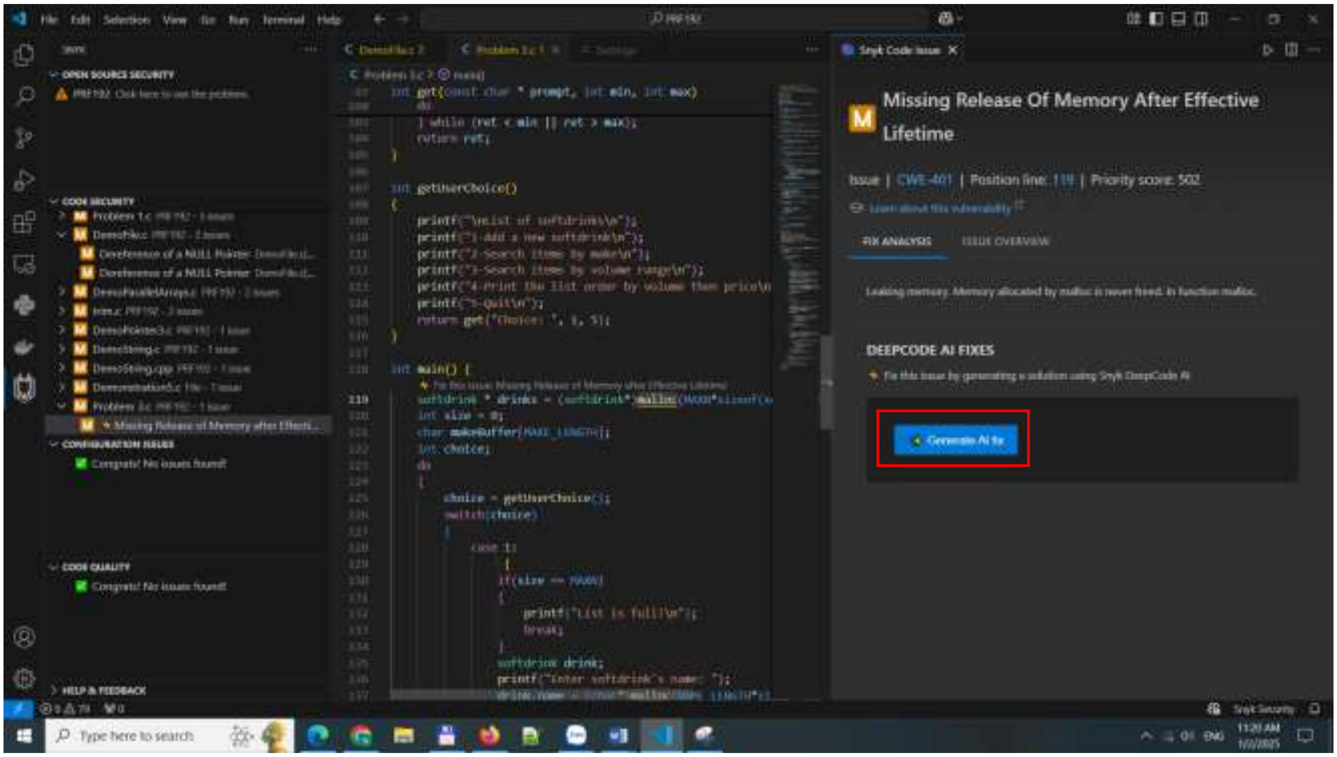
✓ Congrats! No issues found!

Problem 3.c > main()

```
117
118 int main() {
    ⚡ Fix this issue: Missing Release of Memory after Effective Lifetime
119     softdrink * drinks = (softdrink*)malloc(MAXN*sizeof(softdrink));
120     int size = 0;
121     char makeBuffer[MAKE_LENGTH];
122     int choice;
123     do
124     {
125         choice = getUserChoice();
126         switch(choice)
127         {
128             case 1:
129                 {
130                 if(size == MAXN)
131                 {
132                     printf("List is full!\n");
133                     break;
134                 }
135                 softdrink drink;
136                 printf("Enter softdrink's name: ");
137                 drink.name = (char*)malloc(NAME_LENGTH*sizeof(char));
138                 fgets(drink.name, NAME_LENGTH, stdin);
139                 captalize(trim(drink.name));
```

Ln 119, Col 44 (6 selected) Spaces: 4 UTF-8 CRLF {} C Snyk Security Win32

We click **Generate AI fix** to receive suggestions to help fix errors.



```
int get(const char * prompt, int min, int max)
    do
    } while (ret < min || ret > max);
    return ret;
}

int getUserChoice()
{
    printf("\nList of softdrinks\n");
    printf("1-Add a new softdrink\n");
    printf("2-Search items by make\n");
    printf("3-Search items by volume range\n");
    printf("4-Print the list order by volume then price\n
    printf("5-Quit\n");
    return get("Choice: ", 1, 5);
}

int main() {
    softdrink * drinks = (softdrink*)malloc(MAXN*sizeof(s
    int size = 0;
    char makeBuffer[MAKE_LENGTH];
    int choice;
    do
    {
        choice = getUserChoice();
        switch(choice)
        {
            case 1:
            {
            if(size == MAXN)
            {
                printf("List is full!\n");
                break;
            }
            softdrink drink;
            printf("Enter softdrink's name: ");
            drink.name = (char*)malloc(NAME_LENGTH*si
```

Missing Release Of Memory After Effective Lifetime

Issue | CWE-401 | Position line: 119 | Priority score: 502

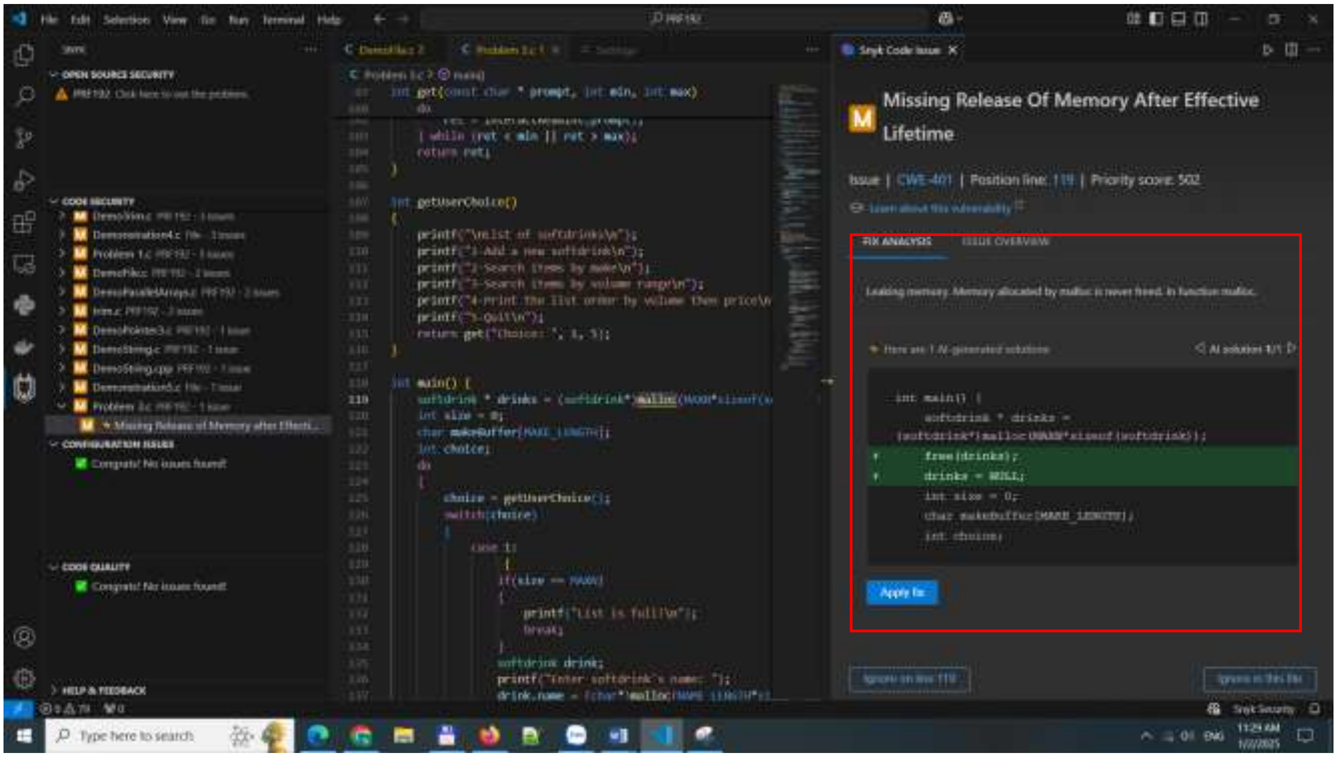
Learn about this vulnerability

Leaking memory. Memory allocated by malloc is never freed. In function malloc.

DEEPCODE AI FIXES

Fix this issue by generating a solution using Snyk DeepCode AI

Snyk AI suggests a solution: "***Leaking memory. Memory allocated by malloc is never freed".*** We click **Apply fix** and check logic code.



Missing Release Of Memory After Effective Lifetime

Issue | CWE-401 | Position line: 119 | Priority score: 502

Learn about this vulnerability

Leaking memory. Memory allocated by malloc is never freed. In function malloc.

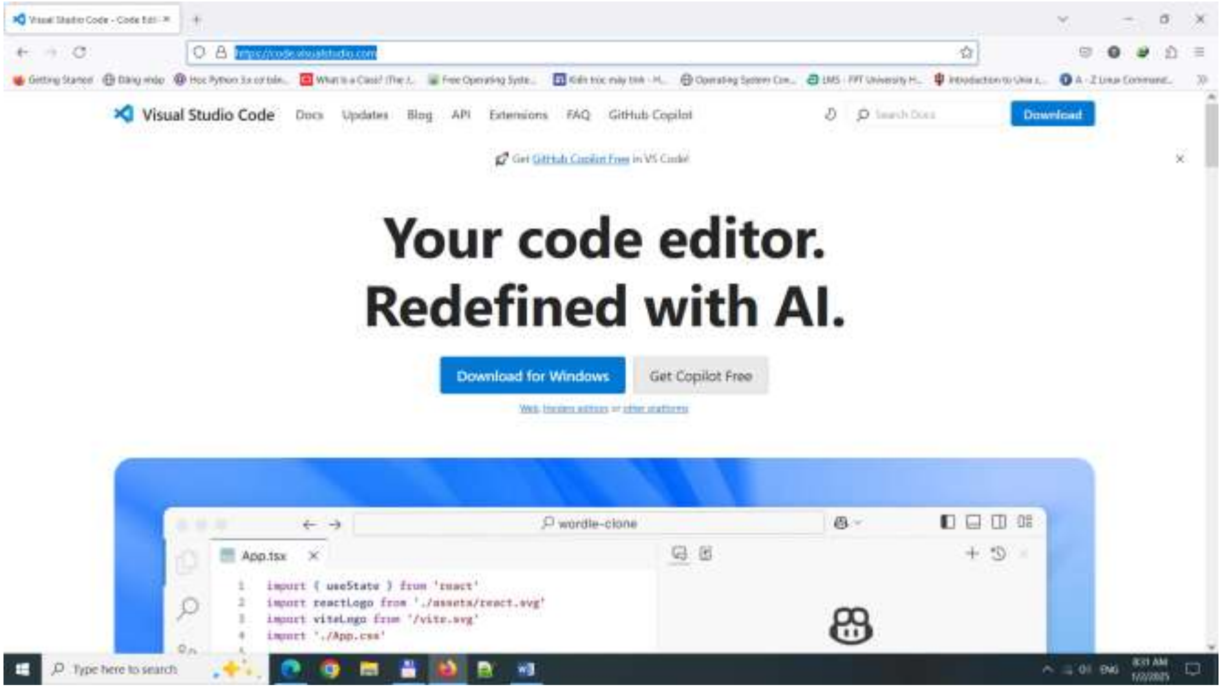
Here are 1 AI-generated solutions   AI solution 1/1

```
int main() {
    softdrink * drinks =
(softdrink*)malloc(MAXN*sizeof(softdrink));
+   free(drinks);
+   drinks = NULL;
    int size = 0;
    char makebuffer[MAKE_LENGTH];
    int choice;
```

IV. Visual Studio Code IDE (VS Code)

4.1. Download

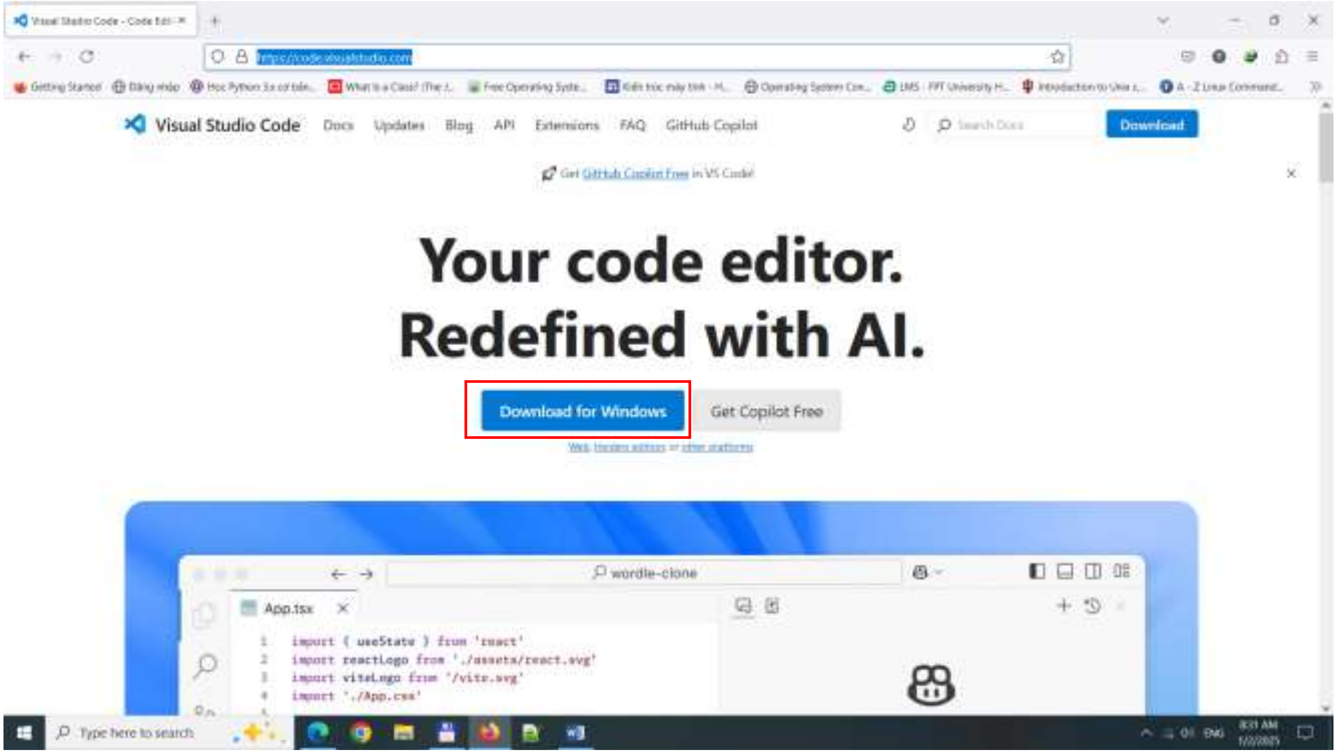
4.1.1. Go to the website: https://code.visualstudio.com/



Your code editor.
Redefined with AI.

```
import { useState } from 'react'
import reactLogo from './assets/react.svg'
import viteLogo from '/vite.svg'
import './App.css'
```

4.1.2. Choose **Download for Windows** to download

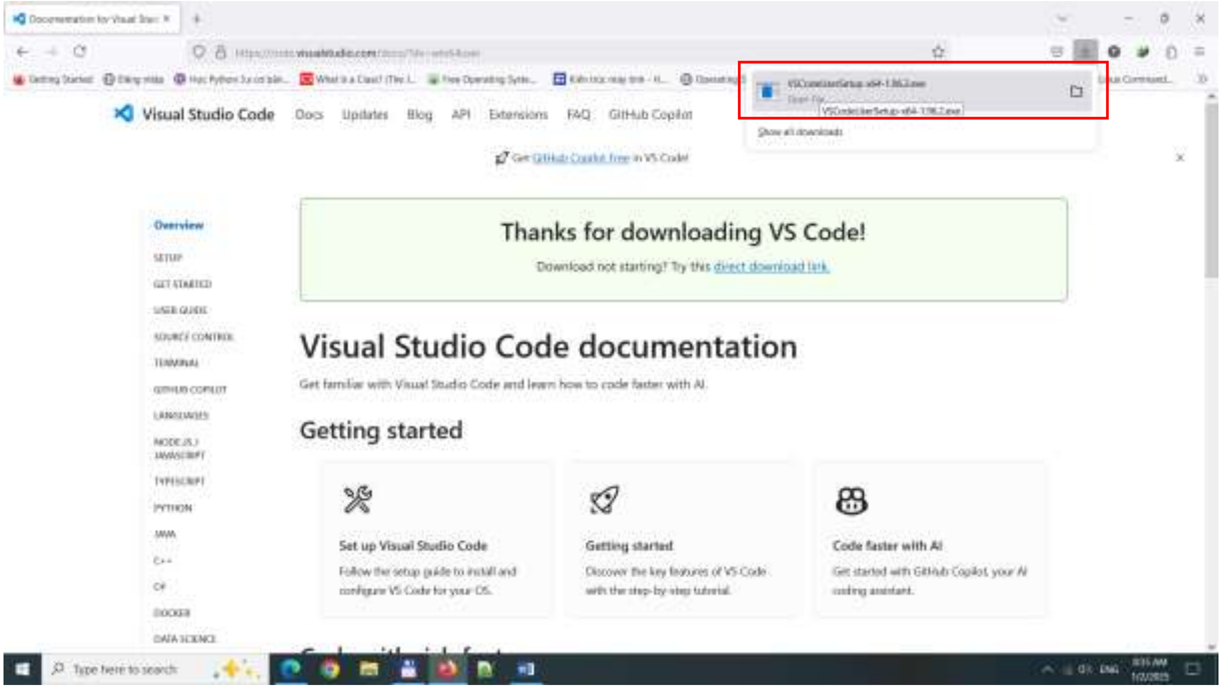


Your code editor.
Redefined with AI.

```
import { useState } from 'react'
import reactLogo from './assets/react.svg'
import viteLogo from '/vite.svg'
import './App.css'
```

4.2. Install

4.2.1. Open **VSCodeUserSetup-x64-1.96.2.exe**



Get GitHub Copilot free in VS Code!

Thanks for downloading VS Code!

Download not starting? Try this direct download link.

Visual Studio Code documentation

Get familiar with Visual Studio Code and learn how to code faster with AI.

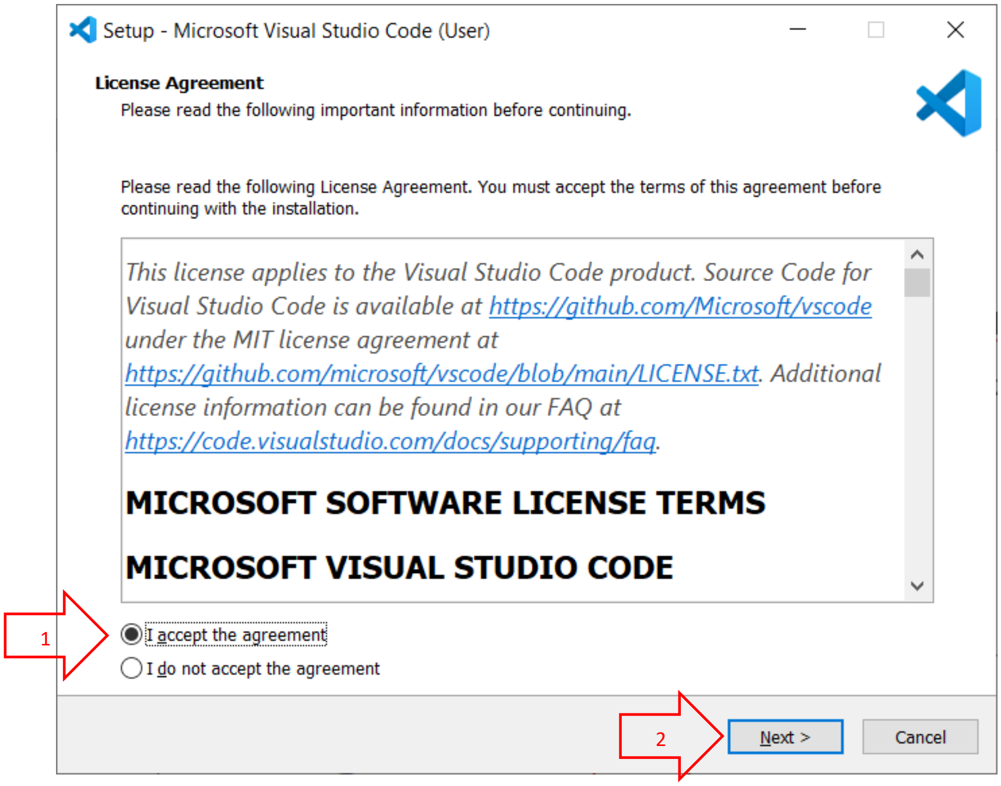
Getting started

Set up Visual Studio Code
Follow the setup guide to install and configure VS Code for your OS.

Getting started
Discover the key features of VS Code with the step-by-step tutorial.

Code faster with AI
Get started with GitHub Copilot, your AI coding assistant.

4.2.2. After the Installer opens, it will ask you to accept the terms and conditions of the Visual Studio Code. Click on **I accept the agreement** and then click the **Next** button.



License Agreement

Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

This license applies to the Visual Studio Code product. Source Code for Visual Studio Code is available at https://github.com/Microsoft/vscode under the MIT license agreement at https://github.com/microsoft/vscode/blob/main/LICENSE.txt. Additional license information can be found in our FAQ at https://code.visualstudio.com/docs/supporting/faq.

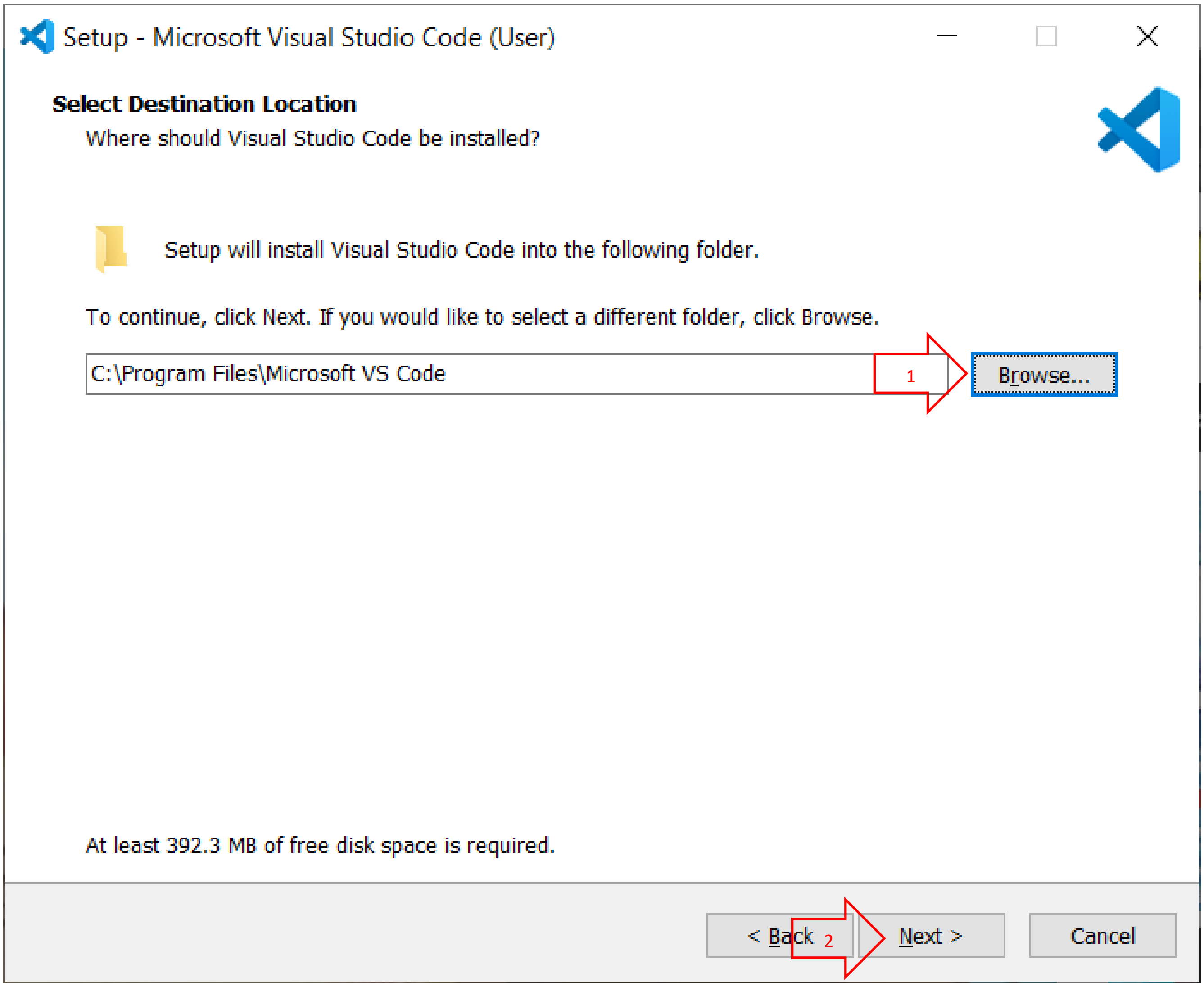
MICROSOFT SOFTWARE LICENSE TERMS

MICROSOFT VISUAL STUDIO CODE

● I accept the agreement

○ I do not accept the agreement

4.2.3. By default, the Visual Studio Code installation location displayed on the screen. However, you can choose a different location by clicking the "**Browse**" button and selecting a different directory. Then click on the **Next** button.



Setup - Microsoft Visual Studio Code (User)

Select Destination Location

Where should Visual Studio Code be installed?

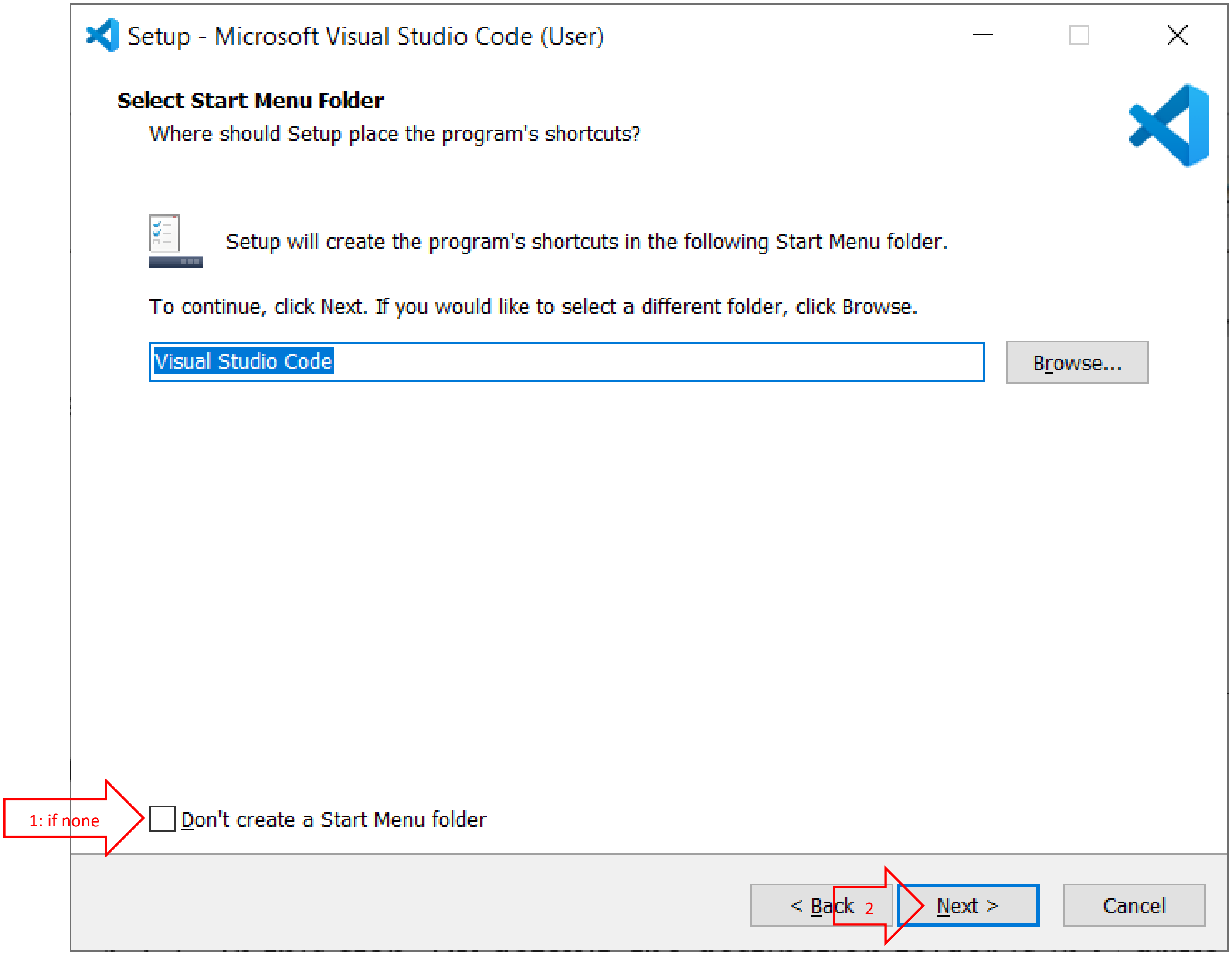
Setup will install Visual Studio Code into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files\Microsoft VS Code

At least 392.3 MB of free disk space is required.

4.2.4. Select **Additional Tasks**. You can choose to create a start menu folder for VS Code or leave it unchecked if you prefer not to have a start menu entry. Then click on the **Next** button.



Setup - Microsoft Visual Studio Code (User)

Select Start Menu Folder

Where should Setup place the program's shortcuts?

Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

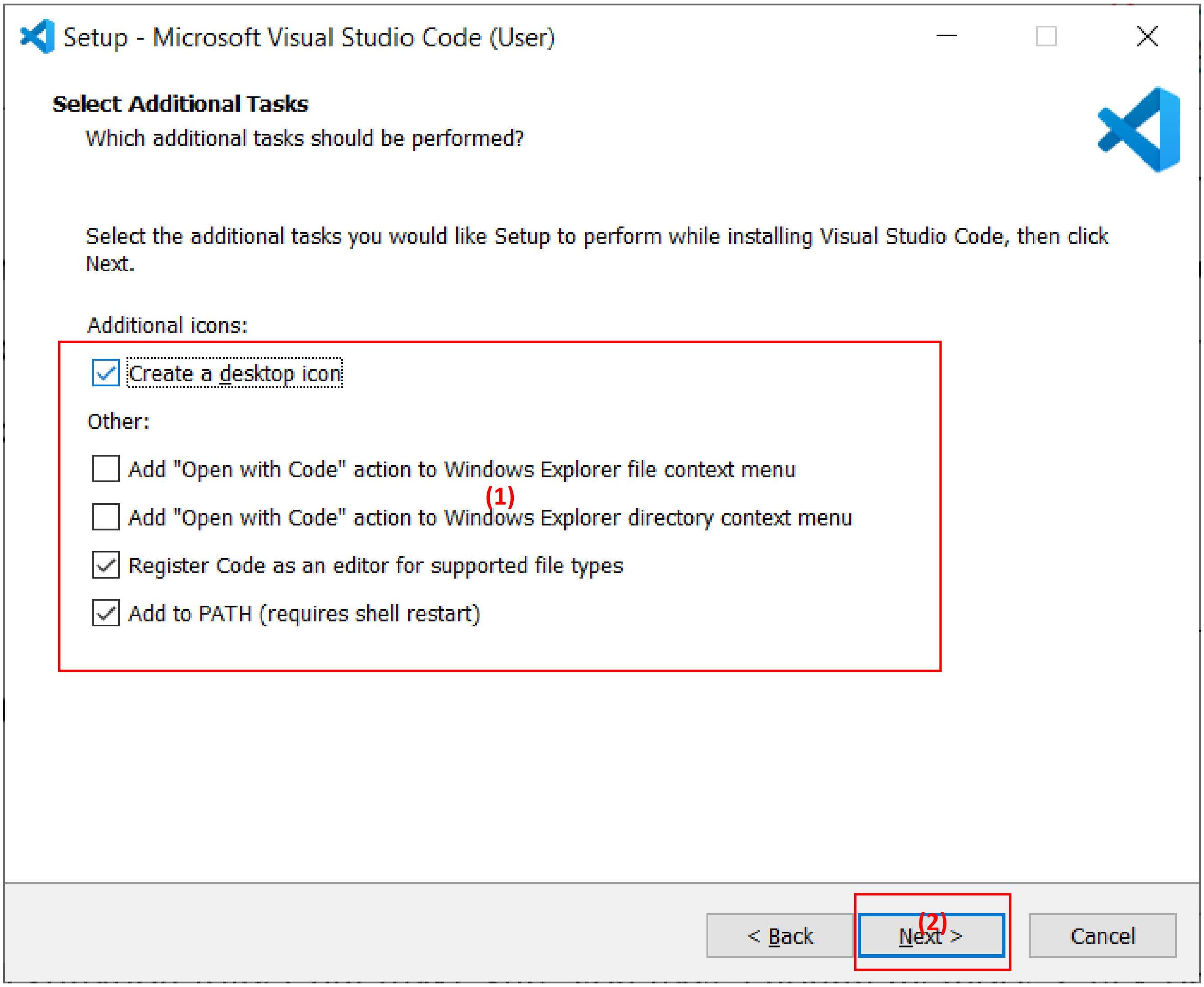
Visual Studio Code

☐ Don't create a Start Menu folder

1: if none

2

4.2.5. The installer may provide additional options, such as creating a desktop icon, adding Visual Studio Code to the PATH environment variable, or installing additional components. Review these options and make your selections based on your preferences. Then click on the **Next** button.



Setup - Microsoft Visual Studio Code (User)

Select Additional Tasks

Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Visual Studio Code, then click Next.

Additional icons:

☑ Create a desktop icon

Other:

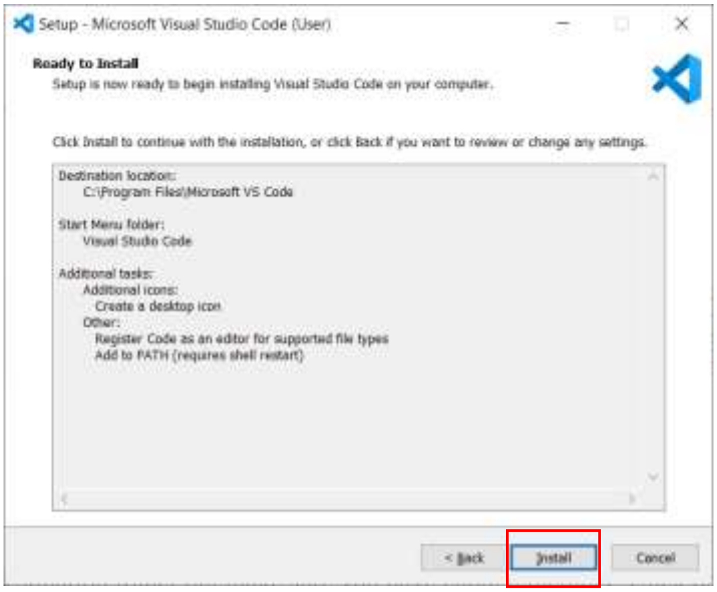
□ Add "Open with Code" action to Windows Explorer file context menu

□ Add "Open with Code" action to Windows Explorer directory context menu

☑ Register Code as an editor for supported file types

☑ Add to PATH (requires shell restart)

4.2.6. Click the **Install** button to begin the installation process.



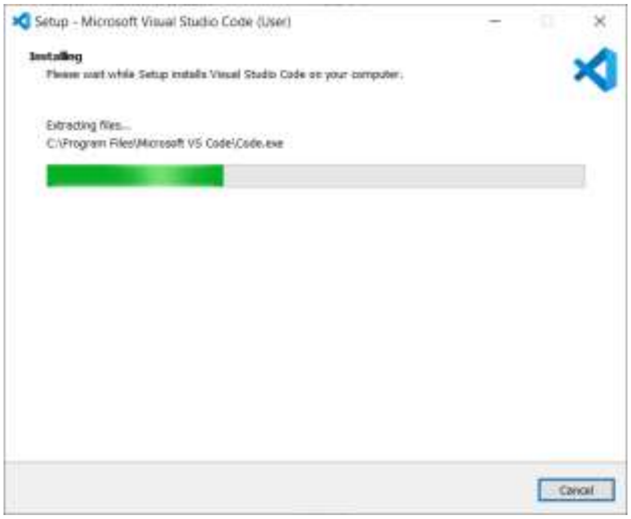
Setup - Microsoft Visual Studio Code (User)

Ready to Install

Setup is now ready to begin installing Visual Studio Code on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

```
Destination location:
    C:\Program Files\Microsoft VS Code

Start Menu folder:
    Visual Studio Code

Additional tasks:
    Additional icons:
      Create a desktop icon
    Other:
      Register Code as an editor for supported file types
      Add to PATH (requires shell restart)
```



Setup - Microsoft Visual Studio Code (User)

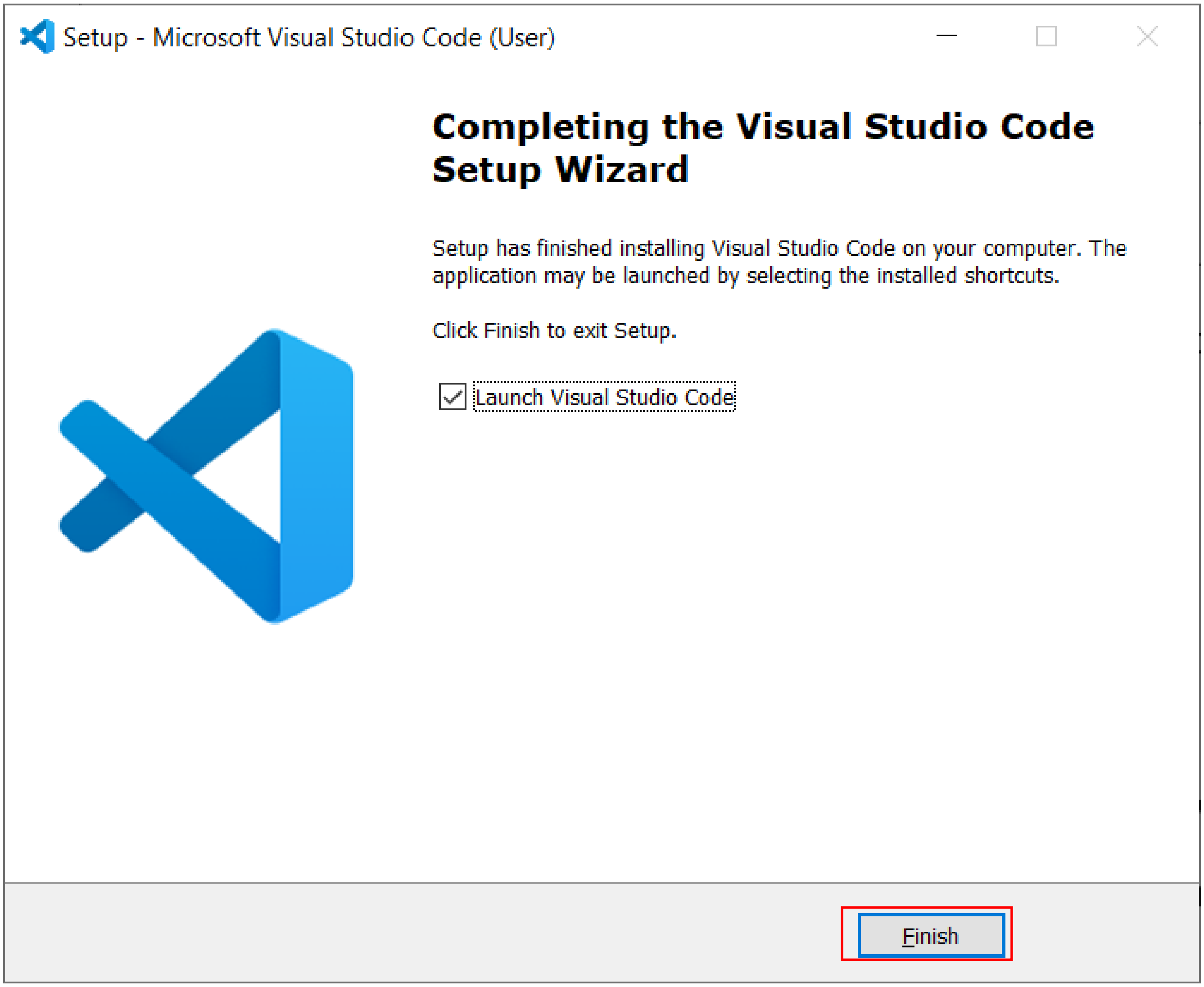
Installing

Please wait while Setup installs Visual Studio Code on your computer.

Extracting files...

C:\Program Files\Microsoft VS Code\Code.exe

4.2.7. Once the installation is complete, you can launch Visual Studio Code by clicking the **Finish** button in the installer:



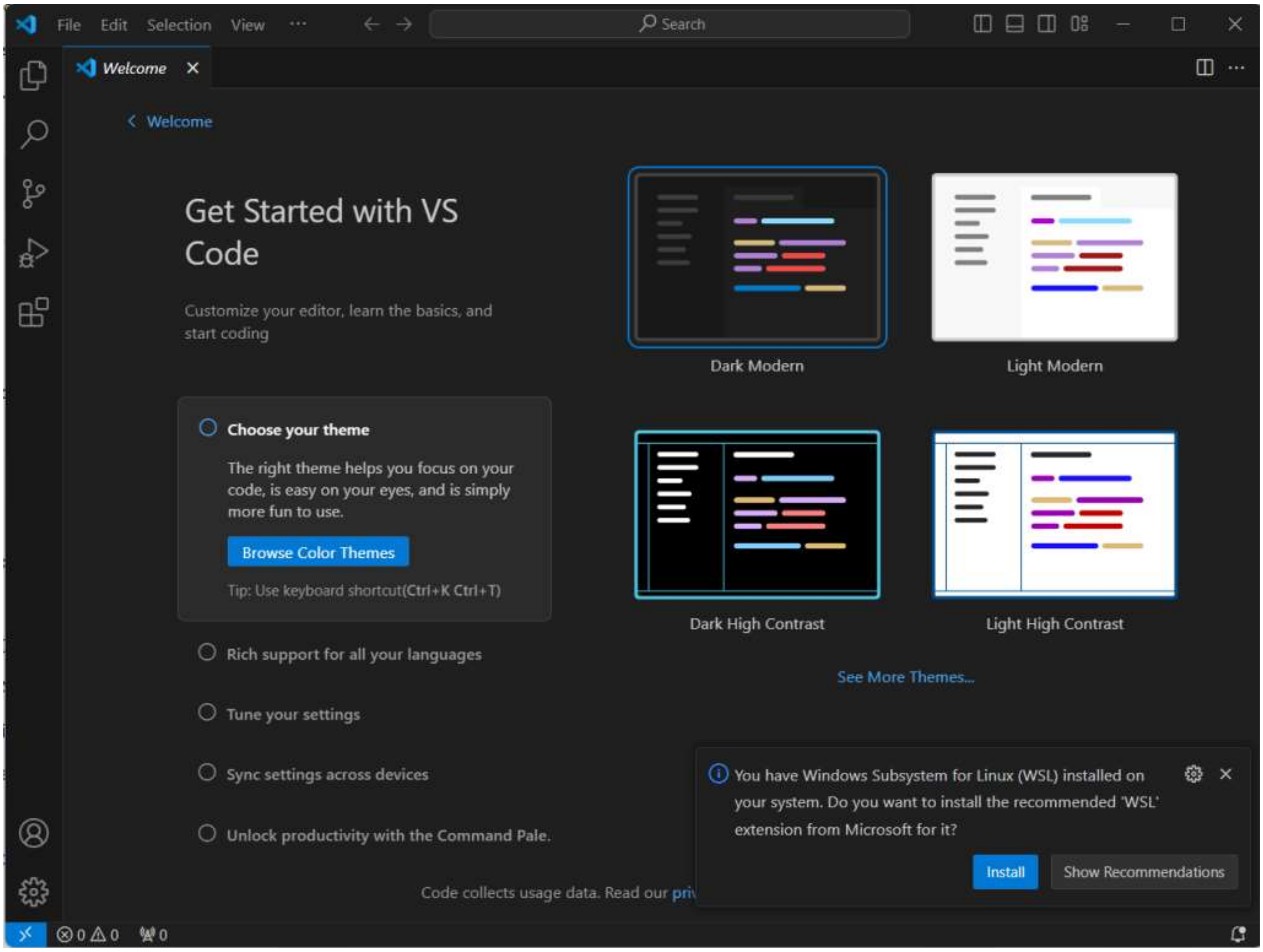
Setup - Microsoft Visual Studio Code (User)

Completing the Visual Studio Code Setup Wizard

Setup has finished installing Visual Studio Code on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

☑ Launch Visual Studio Code



Get Started with VS Code

Customize your editor, learn the basics, and start coding

○ Choose your theme

The right theme helps you focus on your code, is easy on your eyes, and is simply more fun to use.

Tip: Use keyboard shortcut(Ctrl+K Ctrl+T)

○ Rich support for all your languages

○ Tune your settings

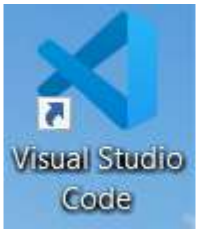
○ Sync settings across devices

○ Unlock productivity with the Command Pale.

Code collects usage data. Read our pri

You have Windows Subsystem for Linux (WSL) installed on your system. Do you want to install the recommended 'WSL' extension from Microsoft for it?

4.2.8. The icon of **Visual Studio Code** on Desktop.

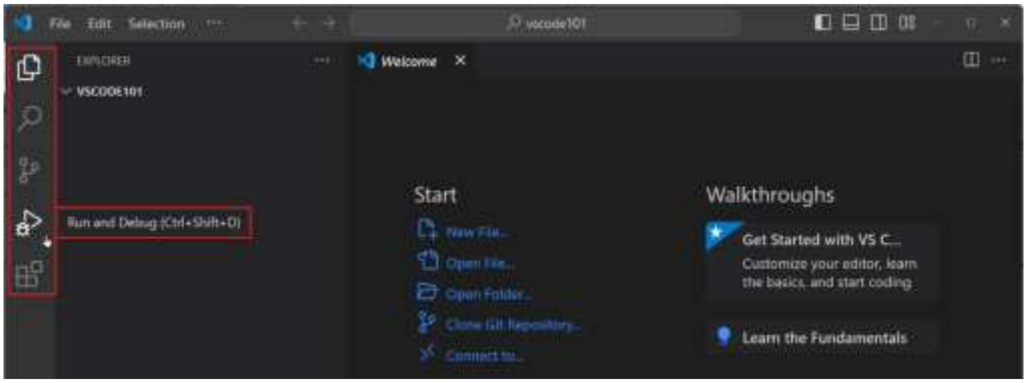


4.3. The basic user interface

Now that you have a folder open in VS Code, we take a quick tour of the user interface.

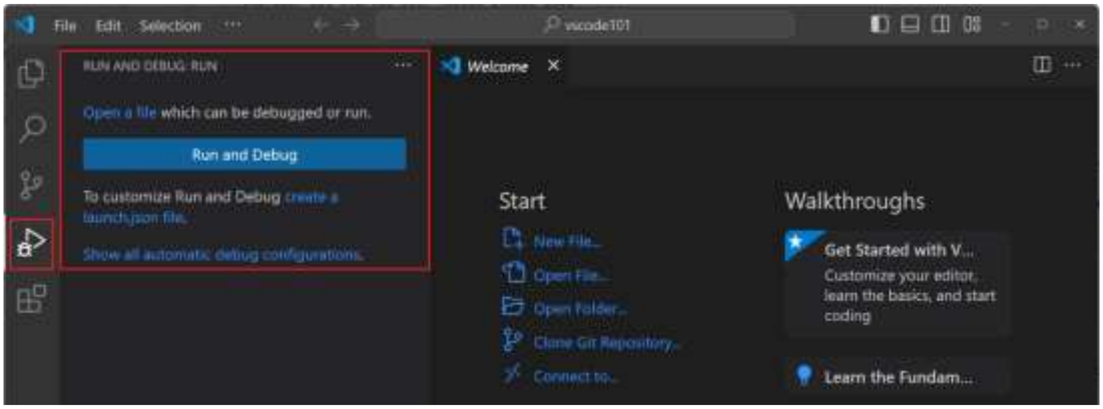
4.3.1. Switch between views with the Activity Bar

4.3.1.1. Use the Activity Bar to switch between different views.



Customize your editor, learn the basics, and start coding

4.3.1.2. When you select a view in the Activity Bar, the **Primary Side Bar** opens to show view-specific information. For example, the Run and Debug view enables you to configure and start debugging sessions.



Open a file which can be debugged or run.

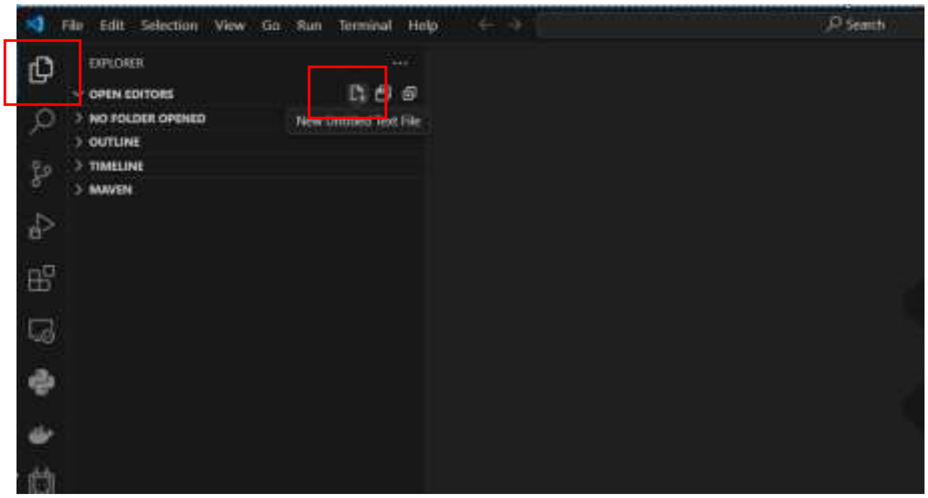
To customize Run and Debug create a launch.json file.

Show all automatic debug configurations.

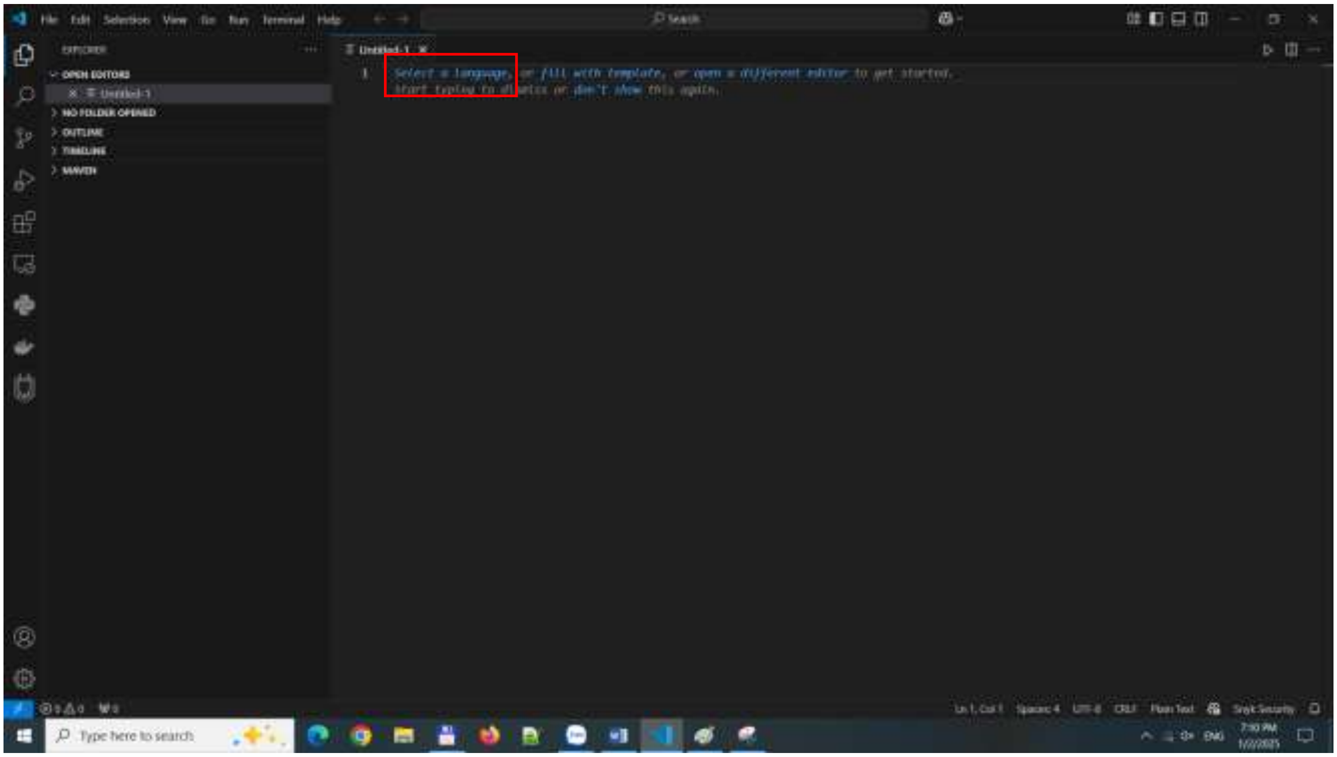
Customize your editor, learn the basics, and start coding

4.3.2. View and edit files with the Editor

4.3.2.1. Select the Explorer view in the Activity Bar, and select the **New Untitled Text File** button to create a new file in your workspace.

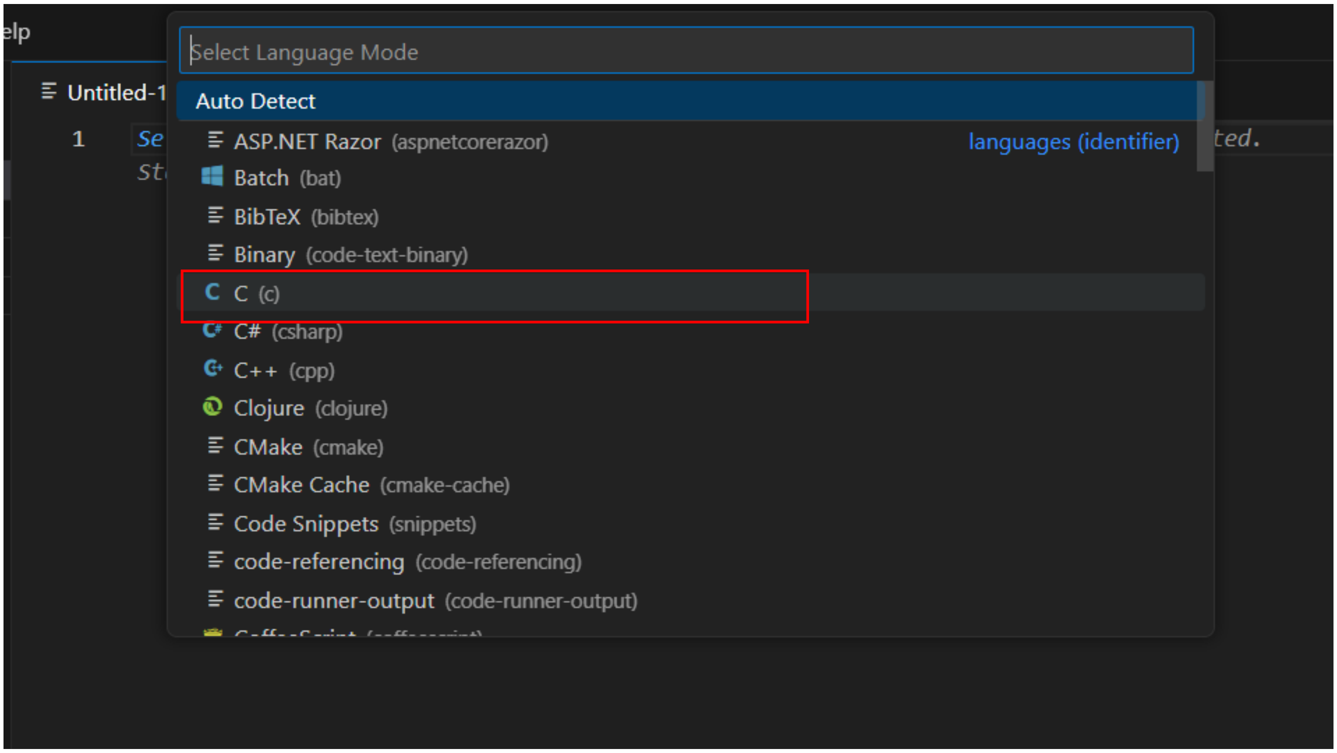


4.3.2.2. Click **Select a language**.



Select a language, or fill with template, or open a different editor to get started.
Start typing to dismiss or don't show this again.

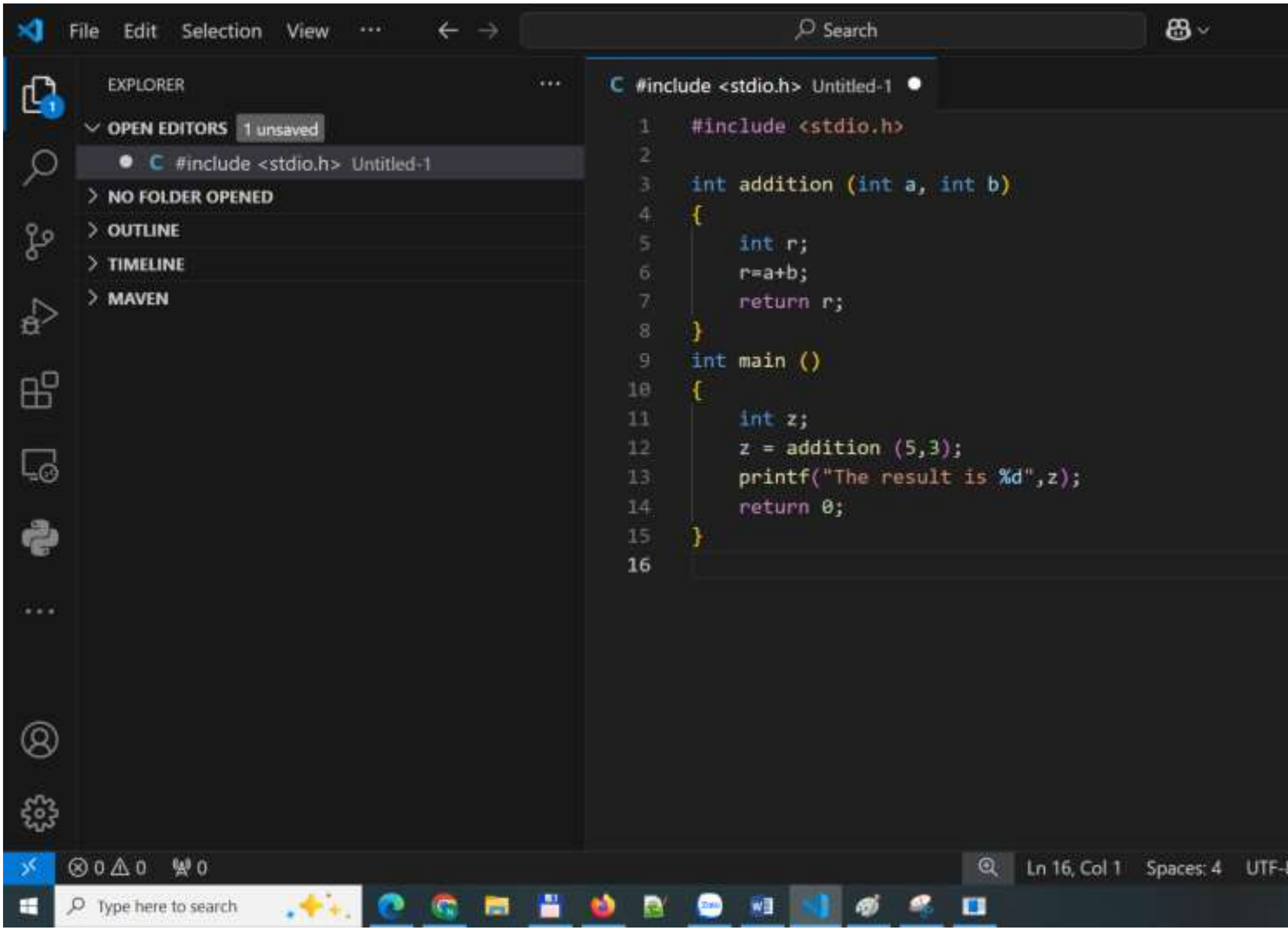
4.3.2.3. If the programming language is C, choose C (c); if the language is C++, choose C++ (cpp). For example, if the programming language is C, choose C(c)



Select Language Mode
Auto Detect
ASP.NET Razor (aspnetcorerazor)
languages (identifier)
Batch (bat)
BibTeX (bibtex)
Binary (code-text-binary)
C (c)
C# (csharp)
C++ (cpp)
Clojure (clojure)
CMake (cmake)
CMake Cache (cmake-cache)
Code Snippets (snippets)
code-referencing (code-referencing)
code-runner-output (code-runner-output)
CoffeeScript (coffeescript)
Untitled-1

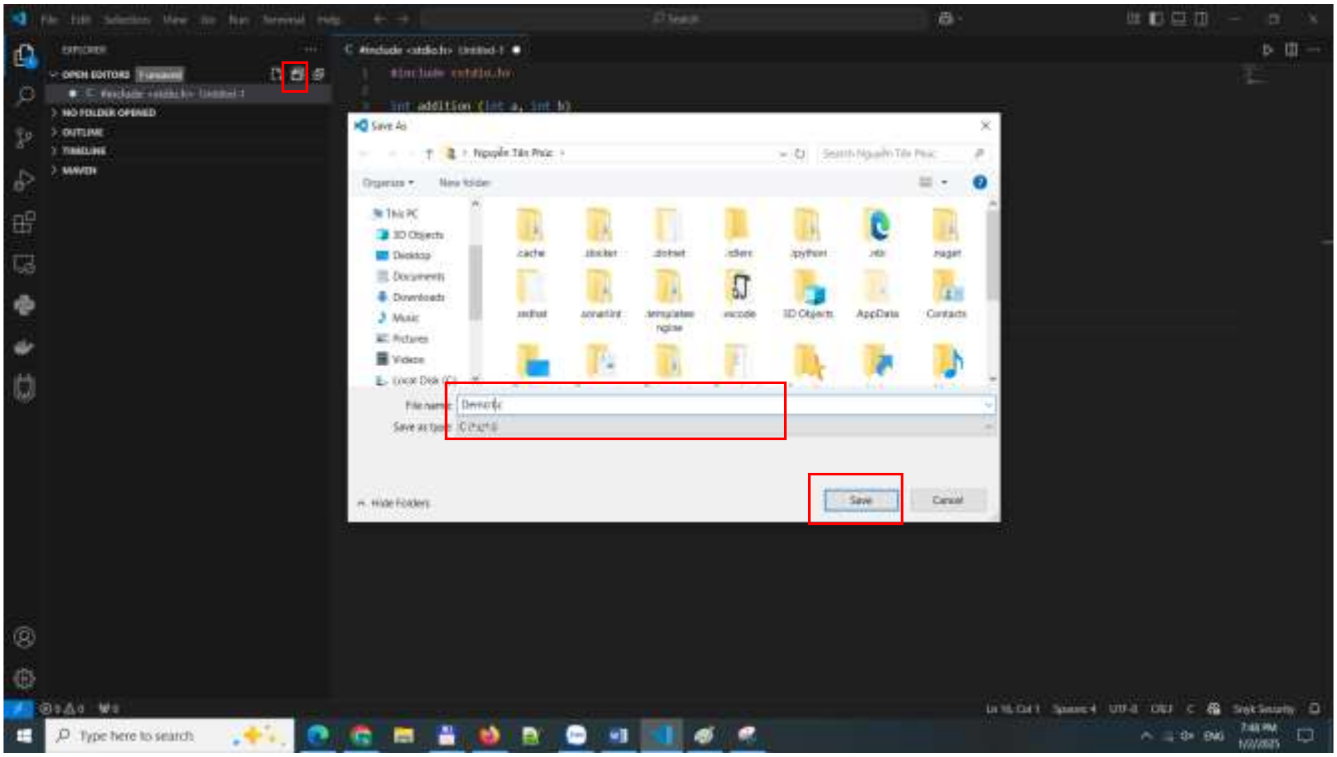
4.3.2.4. Start typing some HTML code in the index.html file.

As you type, you should see suggestions popping up that help you complete your code (*IntelliSense*). You can use the **Up** and **Down** keys to navigate the suggestions, and **Tab** to insert the selected suggestion.



```c
#include <stdio.h>

int addition (int a, int b)
{
    int r;
    r=a+b;
    return r;
}
int main ()
{
    int z;
    z = addition (5,3);
    printf("The result is %d",z);
    return 0;
}
```

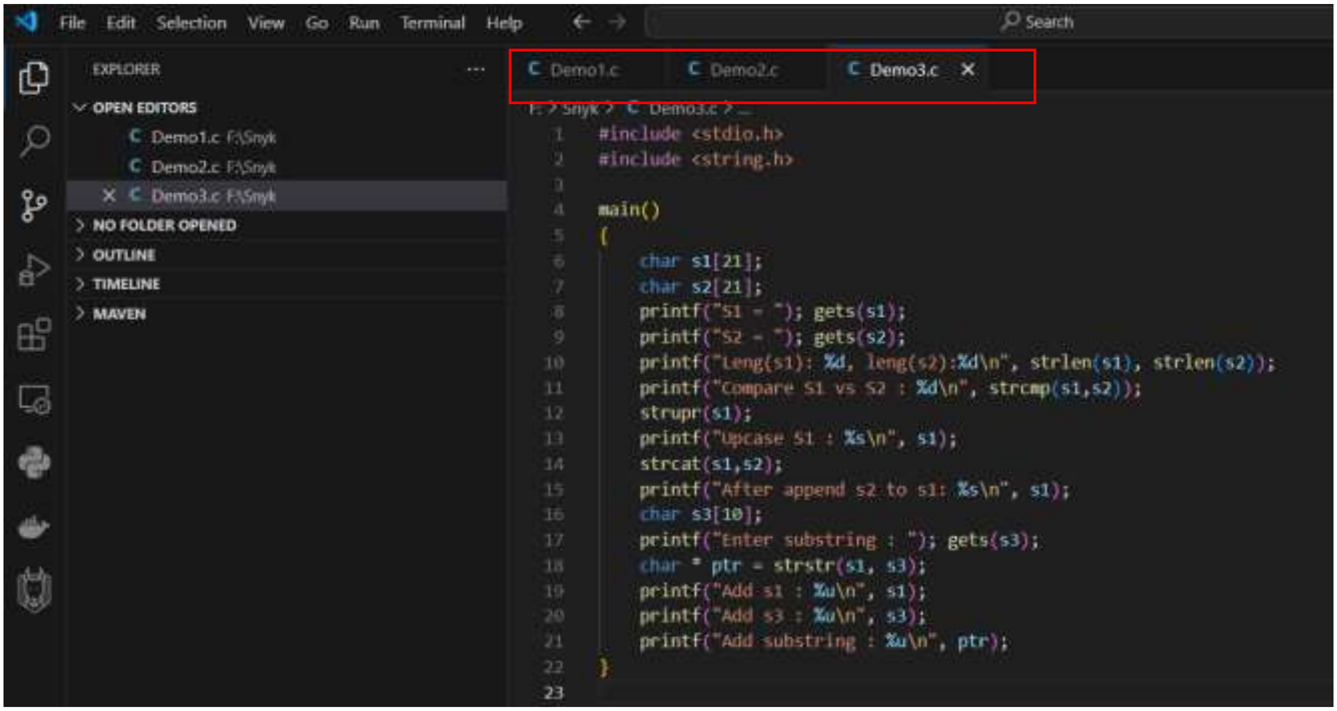
4.3.2.5. Click the icon **Save All**, then type the file name and choose **Save**. A file added to your workspace and an Editor opens in the main area of the window. Example, file name is Demo1.c



```
#include <stdio.h>

int addition (int a, int b)
```

File name: Demo1.c
Save as type: C (*.c;*.h)

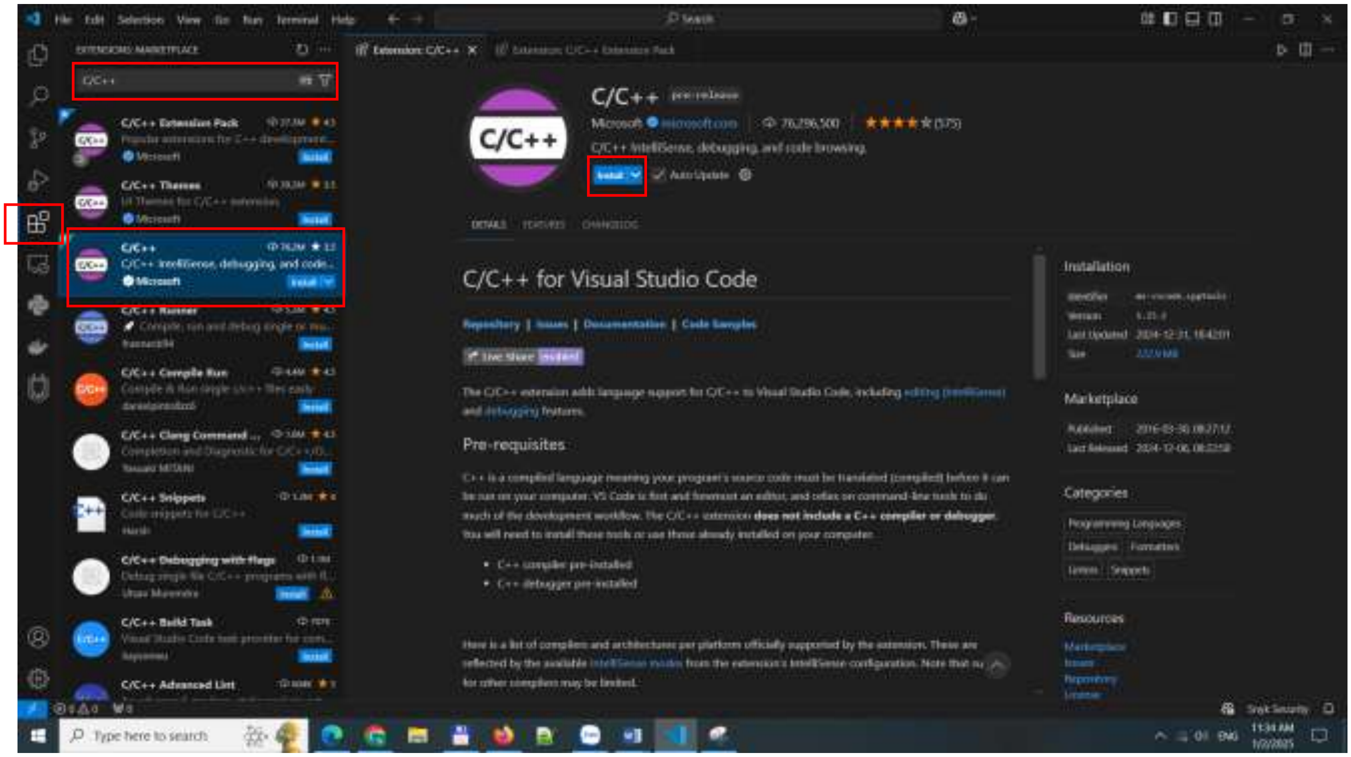
4.3.2.6. Add more files to your workspace and notice that each file opens a new Editor tab. You can open as many editors as you like and view them side by side vertically or horizontally.



```
#include <stdio.h>
#include <string.h>

main()
{
    char s1[21];
    char s2[21];
    printf("S1 = "); gets(s1);
    printf("S2 = "); gets(s2);
    printf("Leng(s1): %d, leng(s2):%d\n", strlen(s1), strlen(s2));
    printf("Compare S1 vs S2 : %d\n", strcmp(s1,s2));
    strupr(s1);
    printf("Upcase S1 : %s\n", s1);
    strcat(s1,s2);
    printf("After append s2 to s1: %s\n", s1);
    char s3[10];
    printf("Enter substring : "); gets(s3);
    char * ptr = strstr(s1, s3);
    printf("Add s1 : %u\n", s1);
    printf("Add s3 : %u\n", s3);
    printf("Add substring : %u\n", ptr);
}
```

4.4. Configure run C/C++ on VS Code

4.4.1. Install the **C/C++** extension for VS Code. You can install the C/C++ extension by searching for '**C/C++**' in the Extensions view (Ctrl+Shift+X).



C/C++ pre-release

Microsoft microsoft.com 76,296,500 ★★★★★ (575)

C/C++ IntelliSense, debugging, and code browsing.

C/C++ for Visual Studio Code

Repository | Issues | Documentation | Code Samples

The C/C++ extension adds language support for C/C++ to Visual Studio Code, including editing (IntelliSense) and debugging features.

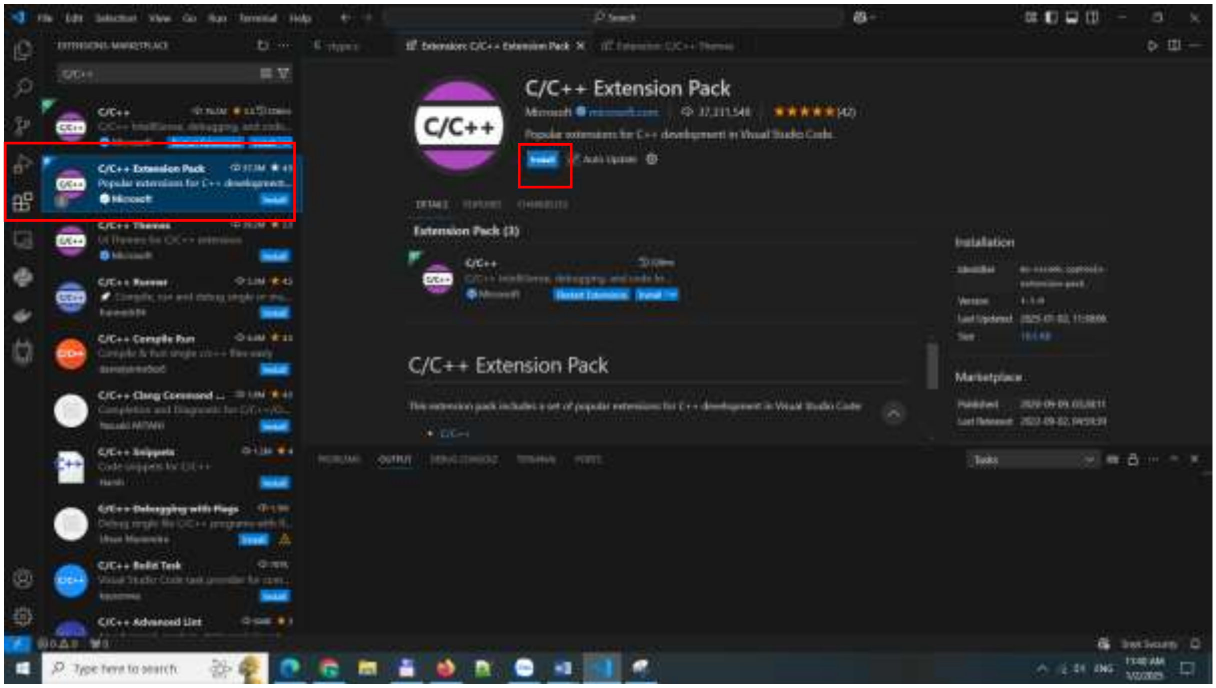
Pre-requisites

C++ is a compiled language meaning your program's source code must be translated (compiled) before it can be run on your computer. VS Code is first and foremost an editor, and relies on command-line tools to do much of the development workflow. The C/C++ extension **does not include a C++ compiler or debugger**. You will need to install these tools or use those already installed on your computer.

- C++ compiler pre-installed
- C++ debugger pre-installed

Here is a list of compilers and architectures per platform officially supported by the extension. These are reflected by the available IntelliSense modes from the extension's IntelliSense configuration. Note that su... for other compilers may be limited.

4.4.2. Install the **C/C++ Extension Pack** for VS Code.



C/C++ Extension Pack

Microsoft microsoft.com 37,331,548 ★★★★★ (42)

Popular extensions for C++ development in Visual Studio Code.

Extension Pack (3)

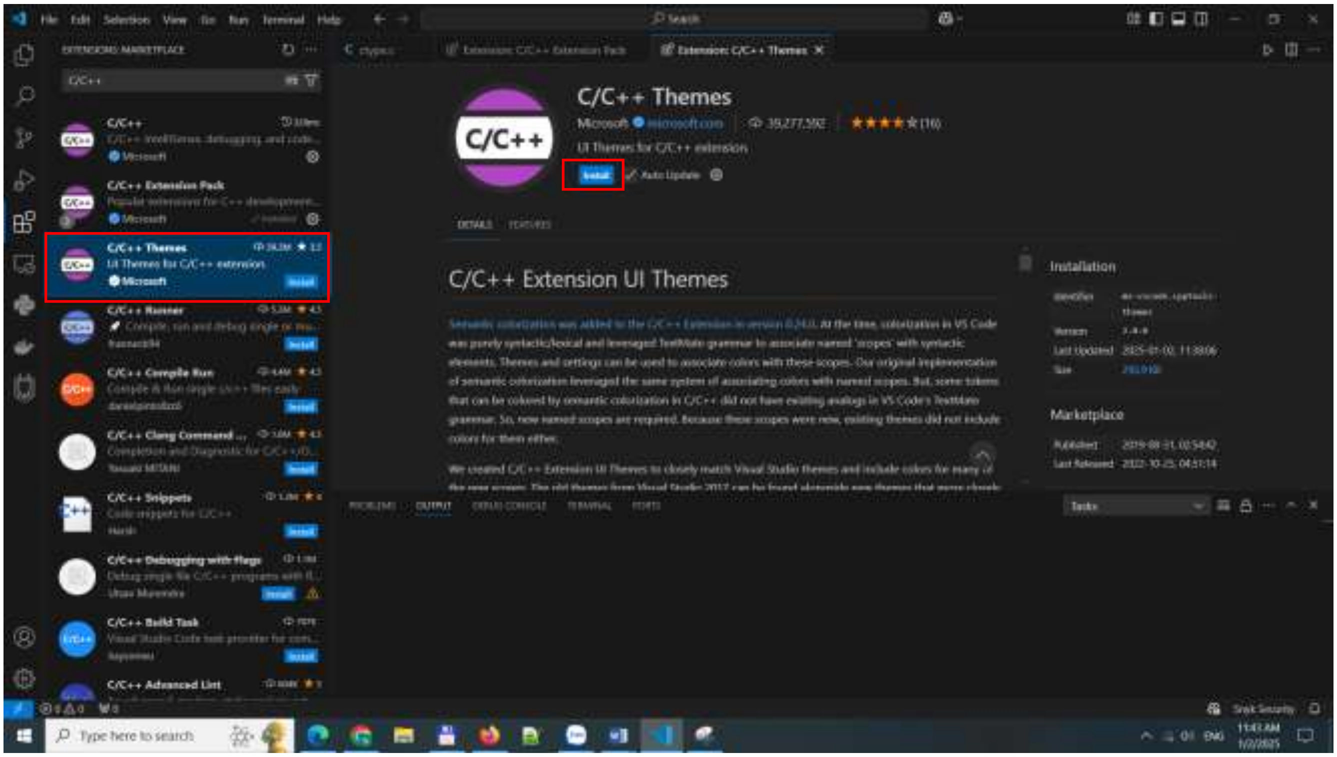
C/C++

C/C++ Extension Pack

This extension pack includes a set of popular extensions for C++ development in Visual Studio Code:

- C/C++

4.4.3. Install the **C/C++ Themes** for VS Code.



C/C++ Extension UI Themes

Semantic colorization was added to the C/C++ Extension in version 0.24.0. At the time, colorization in VS Code was purely syntactic/lexical and leveraged TextMate grammar to associate named 'scopes' with syntactic elements. Themes and settings can be used to associate colors with these scopes. Our original implementation of semantic colorization leveraged the same system of associating colors with named scopes. But, some tokens that can be colored by semantic colorization in C/C++ did not have existing analogs in VS Code's TextMate grammar. So, new named scopes are required. Because these scopes were new, existing themes did not include colors for them either.

We created C/C++ Extension UI Themes to closely match Visual Studio themes and include colors for many of the new scopes. The old themes from Visual Studio 2017 can be found alongside new themes that more closely

Installation

Identifier ms-vscode.cpptools-themes

Version 2.0.0

Last Updated 2025-01-02, 11:38:06

Size 293.91KB

Marketplace

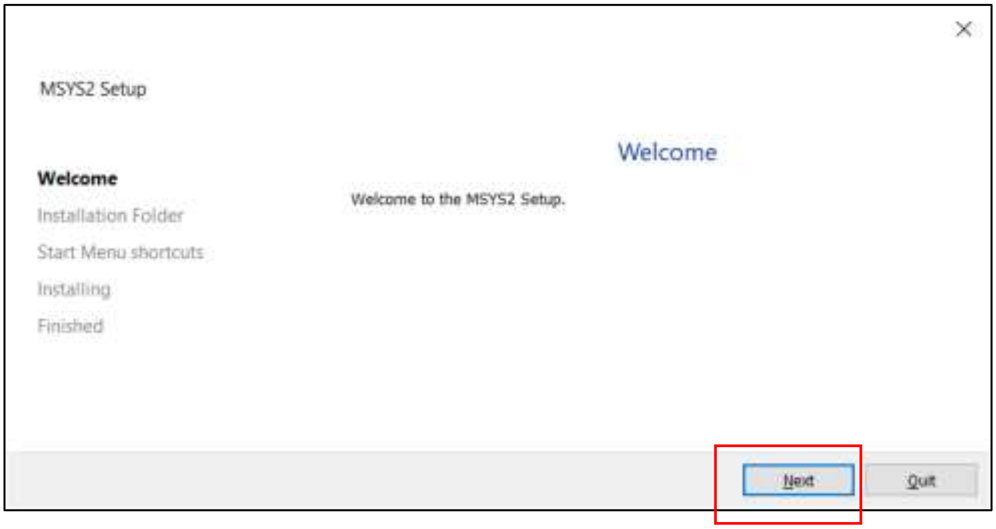
Published 2019-08-31, 02:54:42

Last Released 2022-10-25, 04:51:14

4.4.4. Installing the MinGW-w64 toolchain

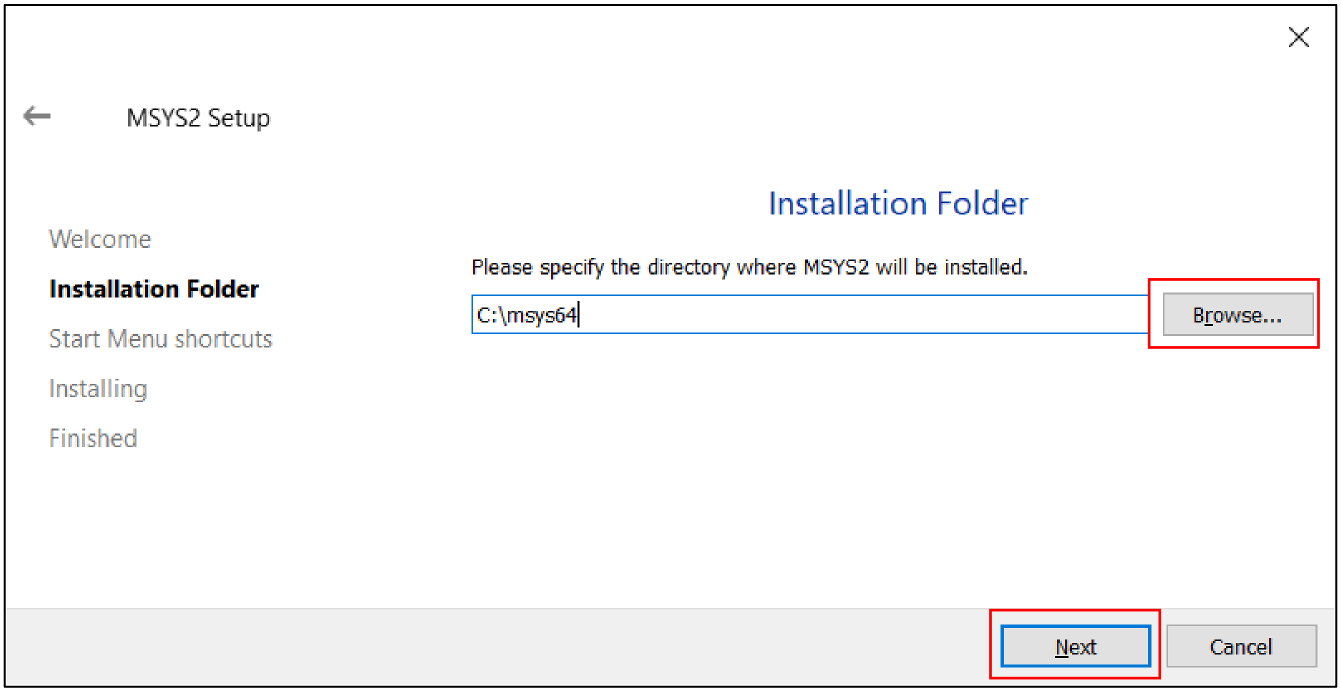
4.4.4.1. Get the latest version of MinGW-w64 at link: https://github.com/msys2/msys2-installer/releases/download/2024-01-13/msys2-x86_64-20240113.exe

4.4.4.2. Run **msys2-x86_64-20240113.exe.** Click **Next**



Welcome to the MSYS2 Setup.

4.4.4.3. Choose your desired Installation Folder. Click **Next**

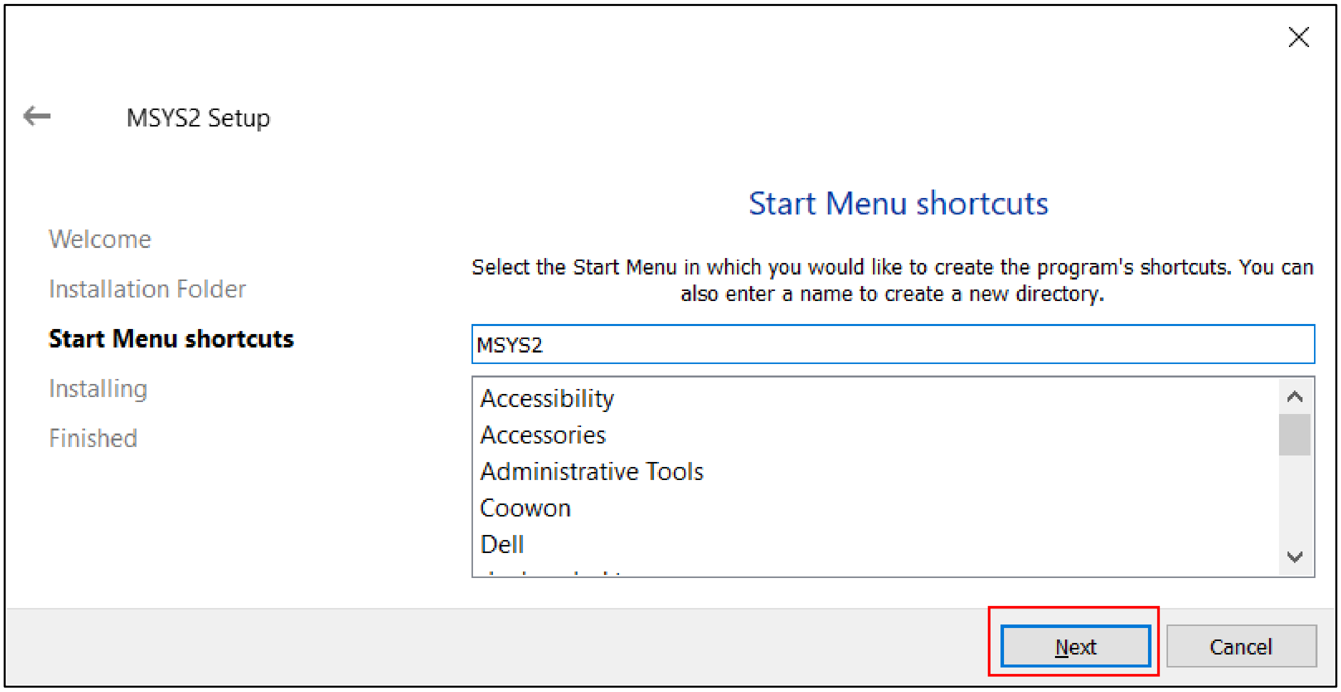


Installation Folder

Please specify the directory where MSYS2 will be installed.

C:\msys64

4.4.4.4. Setting the start menu shortcuts. Click **Next**

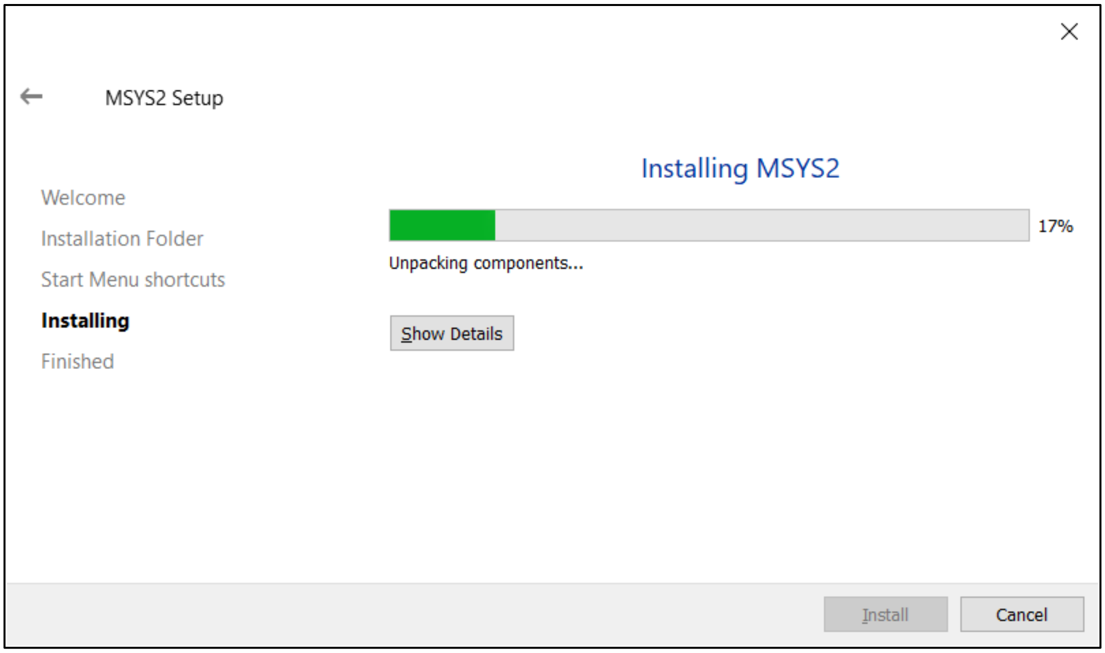


Start Menu shortcuts

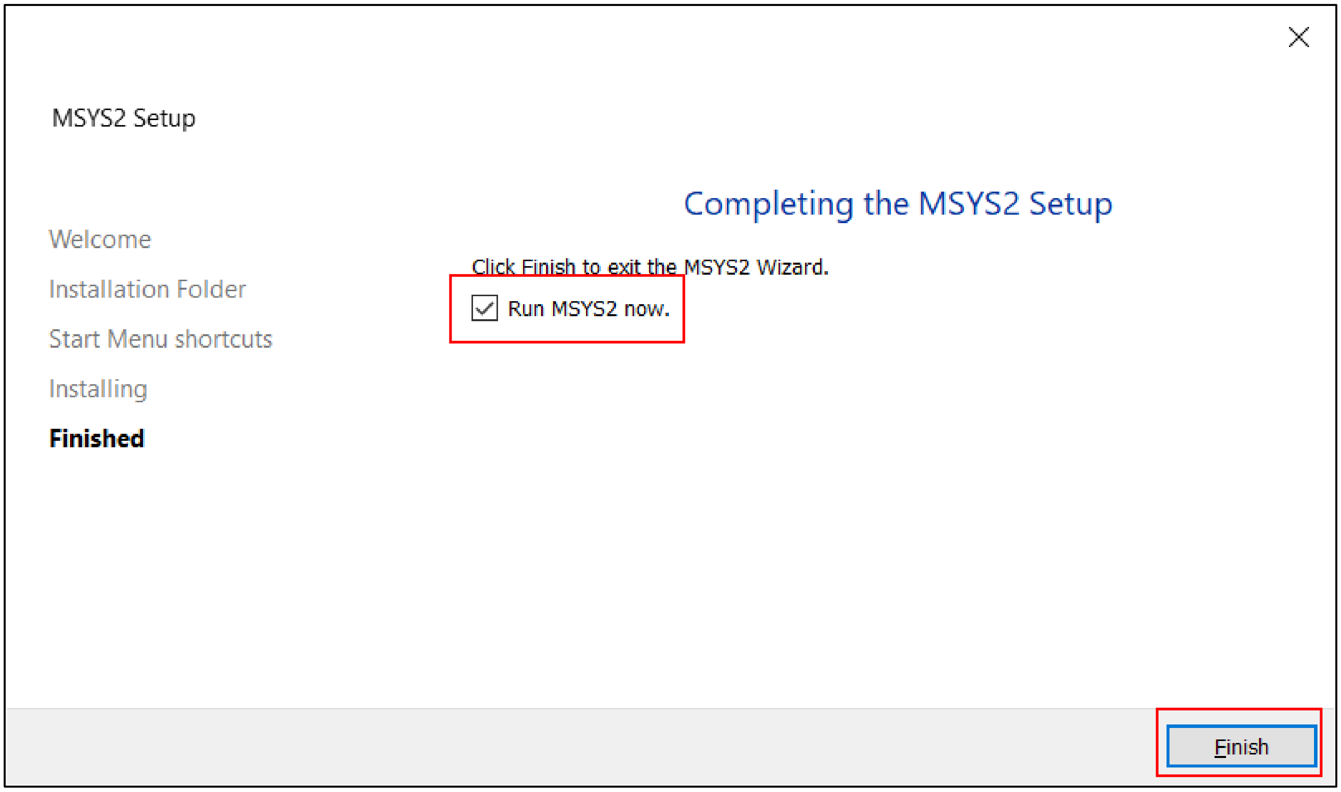
Select the Start Menu in which you would like to create the program's shortcuts. You can also enter a name to create a new directory.

MSYS2

4.4.4.5. Installing



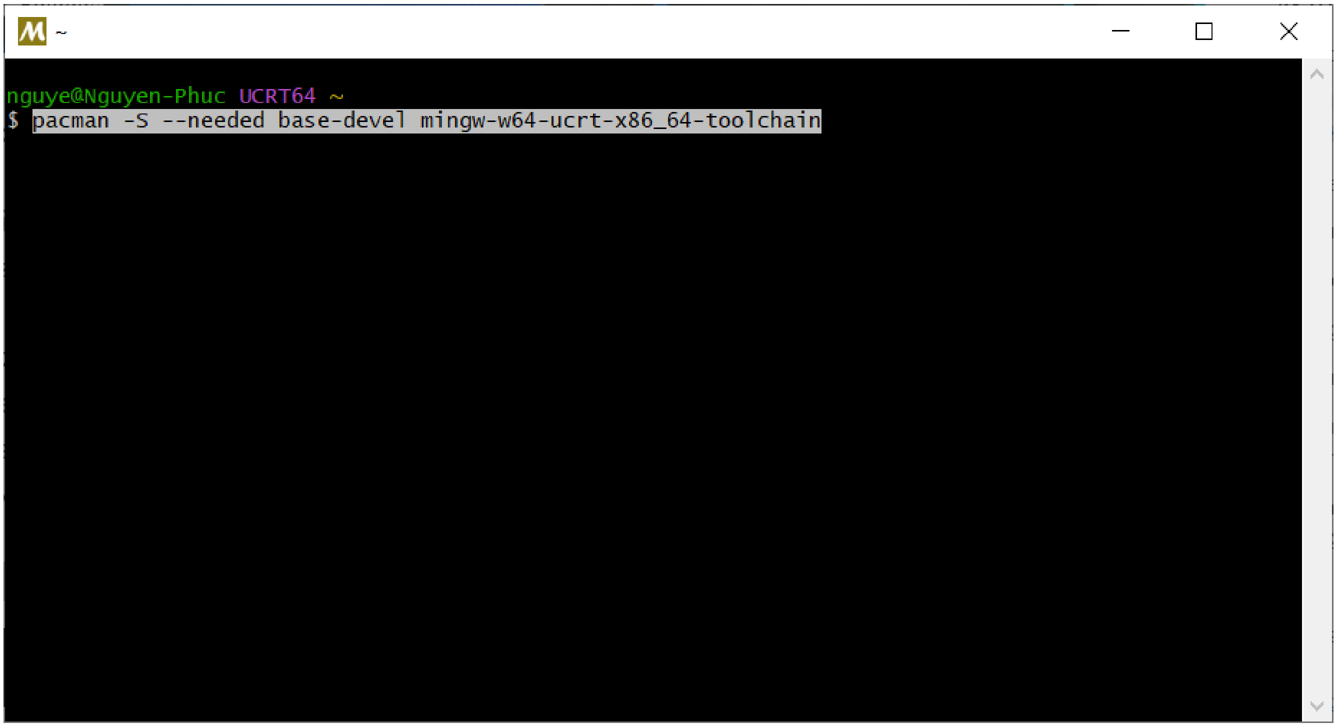
4.4.4.6. Check **Run MSYS2 now** → Click **Finish**



Click Finish to exit the MSYS2 Wizard.

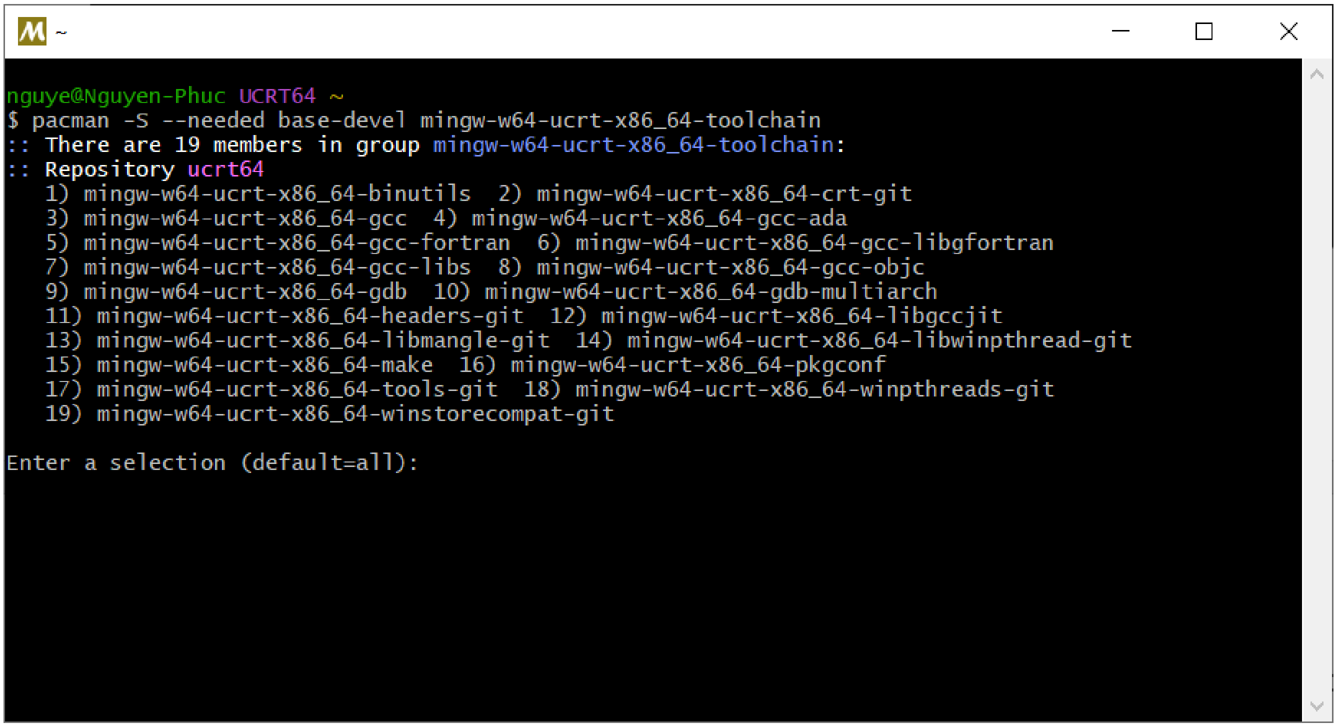
☑ Run MSYS2 now.

4.4.4.7. In this terminal, install the MinGW-w64 toolchain by running the following command: Run **pacman -S --needed base-devel mingw-w64-ucrt-x86_64-toolchain**



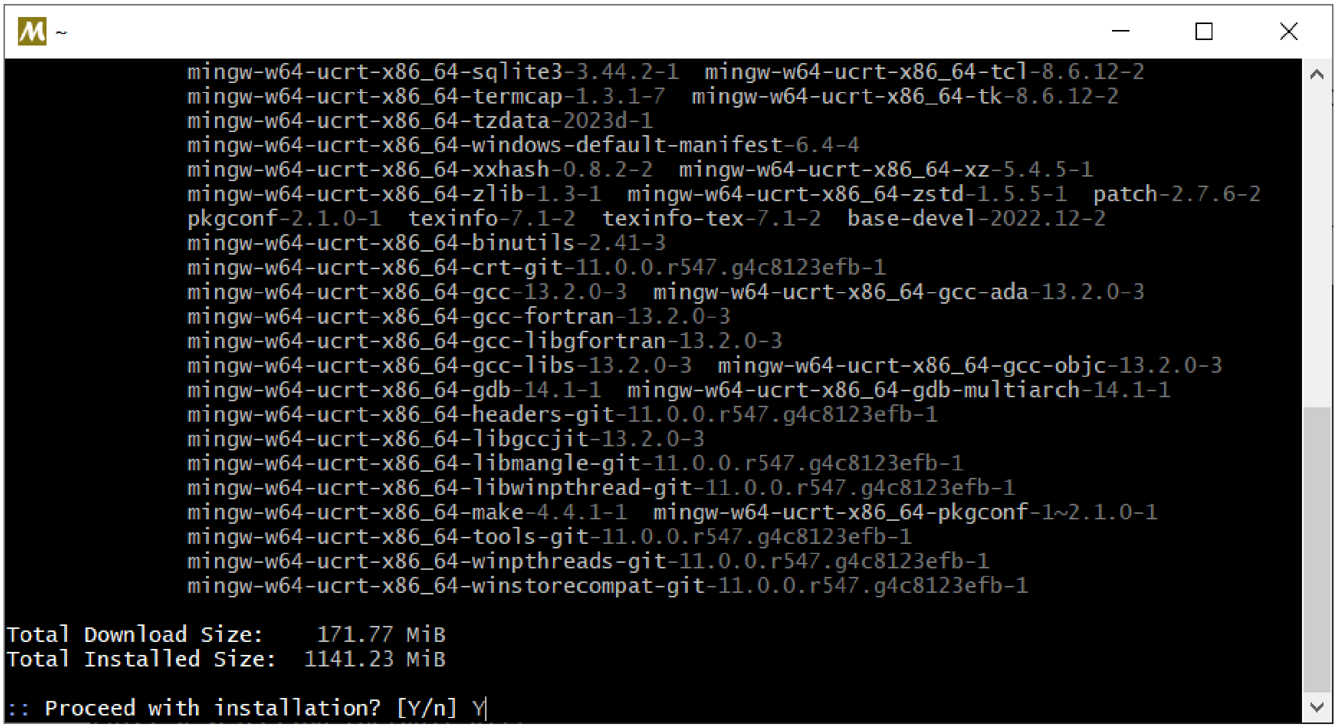
```
nguye@Nguyen-Phuc UCRT64 ~
$ pacman -S --needed base-devel mingw-w64-ucrt-x86_64-toolchain
```

4.4.4.8. Accept the default number of packages in the toolchain group by pressing **Enter**.

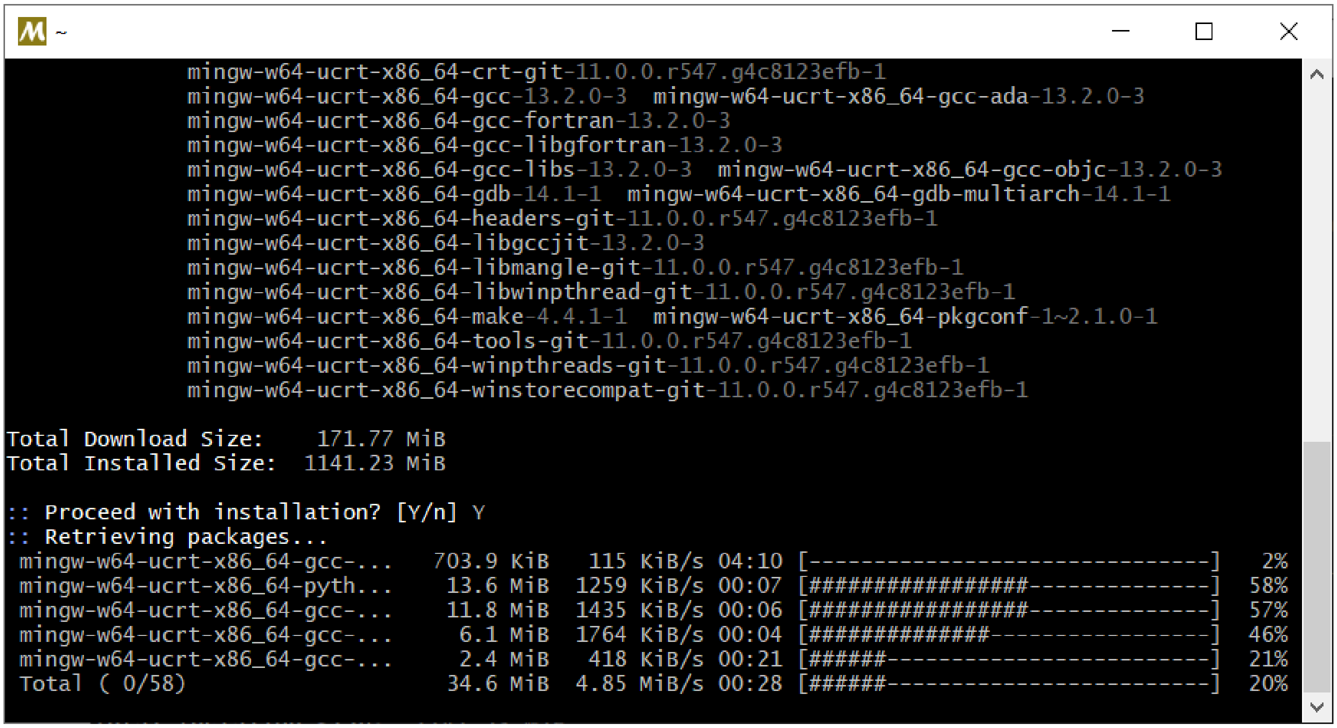


```
nguye@Nguyen-Phuc UCRT64 ~
$ pacman -S --needed base-devel mingw-w64-ucrt-x86_64-toolchain
:: There are 19 members in group mingw-w64-ucrt-x86_64-toolchain:
:: Repository ucrt64
   1) mingw-w64-ucrt-x86_64-binutils  2) mingw-w64-ucrt-x86_64-crt-git
   3) mingw-w64-ucrt-x86_64-gcc  4) mingw-w64-ucrt-x86_64-gcc-ada
   5) mingw-w64-ucrt-x86_64-gcc-fortran  6) mingw-w64-ucrt-x86_64-gcc-libgfortran
   7) mingw-w64-ucrt-x86_64-gcc-libs  8) mingw-w64-ucrt-x86_64-gcc-objc
   9) mingw-w64-ucrt-x86_64-gdb  10) mingw-w64-ucrt-x86_64-gdb-multiarch
   11) mingw-w64-ucrt-x86_64-headers-git  12) mingw-w64-ucrt-x86_64-libgccjit
   13) mingw-w64-ucrt-x86_64-libmangle-git  14) mingw-w64-ucrt-x86_64-libwinpthread-git
   15) mingw-w64-ucrt-x86_64-make  16) mingw-w64-ucrt-x86_64-pkgconf
   17) mingw-w64-ucrt-x86_64-tools-git  18) mingw-w64-ucrt-x86_64-winpthreads-git
   19) mingw-w64-ucrt-x86_64-winstorecompat-git

Enter a selection (default=all):
```

4.4.4.9. Enter **Y** when prompted whether to proceed with the installation.



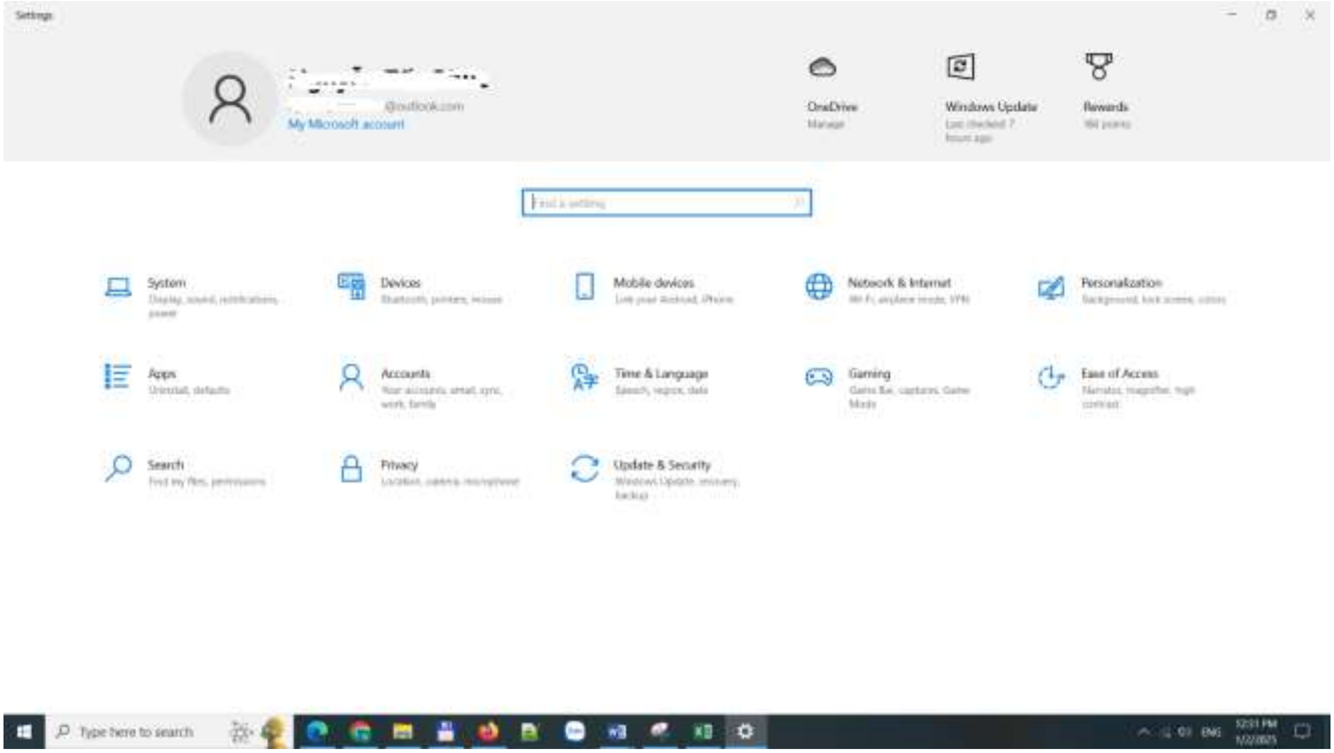
```
              mingw-w64-ucrt-x86_64-sqlite3-3.44.2-1  mingw-w64-ucrt-x86_64-tcl-8.6.12-2
              mingw-w64-ucrt-x86_64-termcap-1.3.1-7  mingw-w64-ucrt-x86_64-tk-8.6.12-2
              mingw-w64-ucrt-x86_64-tzdata-2023d-1
              mingw-w64-ucrt-x86_64-windows-default-manifest-6.4-4
              mingw-w64-ucrt-x86_64-xxhash-0.8.2-2  mingw-w64-ucrt-x86_64-xz-5.4.5-1
              mingw-w64-ucrt-x86_64-zlib-1.3-1  mingw-w64-ucrt-x86_64-zstd-1.5.5-1  patch-2.7.6-2
              pkgconf-2.1.0-1  texinfo-7.1-2  texinfo-tex-7.1-2  base-devel-2022.12-2
              mingw-w64-ucrt-x86_64-binutils-2.41-3
              mingw-w64-ucrt-x86_64-crt-git-11.0.0.r547.g4c8123efb-1
              mingw-w64-ucrt-x86_64-gcc-13.2.0-3  mingw-w64-ucrt-x86_64-gcc-ada-13.2.0-3
              mingw-w64-ucrt-x86_64-gcc-fortran-13.2.0-3
              mingw-w64-ucrt-x86_64-gcc-libgfortran-13.2.0-3
              mingw-w64-ucrt-x86_64-gcc-libs-13.2.0-3  mingw-w64-ucrt-x86_64-gcc-objc-13.2.0-3
              mingw-w64-ucrt-x86_64-gdb-14.1-1  mingw-w64-ucrt-x86_64-gdb-multiarch-14.1-1
              mingw-w64-ucrt-x86_64-headers-git-11.0.0.r547.g4c8123efb-1
              mingw-w64-ucrt-x86_64-libgccjit-13.2.0-3
              mingw-w64-ucrt-x86_64-libmangle-git-11.0.0.r547.g4c8123efb-1
              mingw-w64-ucrt-x86_64-libwinpthread-git-11.0.0.r547.g4c8123efb-1
              mingw-w64-ucrt-x86_64-make-4.4.1-1  mingw-w64-ucrt-x86_64-pkgconf-1~2.1.0-1
              mingw-w64-ucrt-x86_64-tools-git-11.0.0.r547.g4c8123efb-1
              mingw-w64-ucrt-x86_64-winpthreads-git-11.0.0.r547.g4c8123efb-1
              mingw-w64-ucrt-x86_64-winstorecompat-git-11.0.0.r547.g4c8123efb-1

Total Download Size:    171.77 MiB
Total Installed Size:  1141.23 MiB

:: Proceed with installation? [Y/n] Y
```



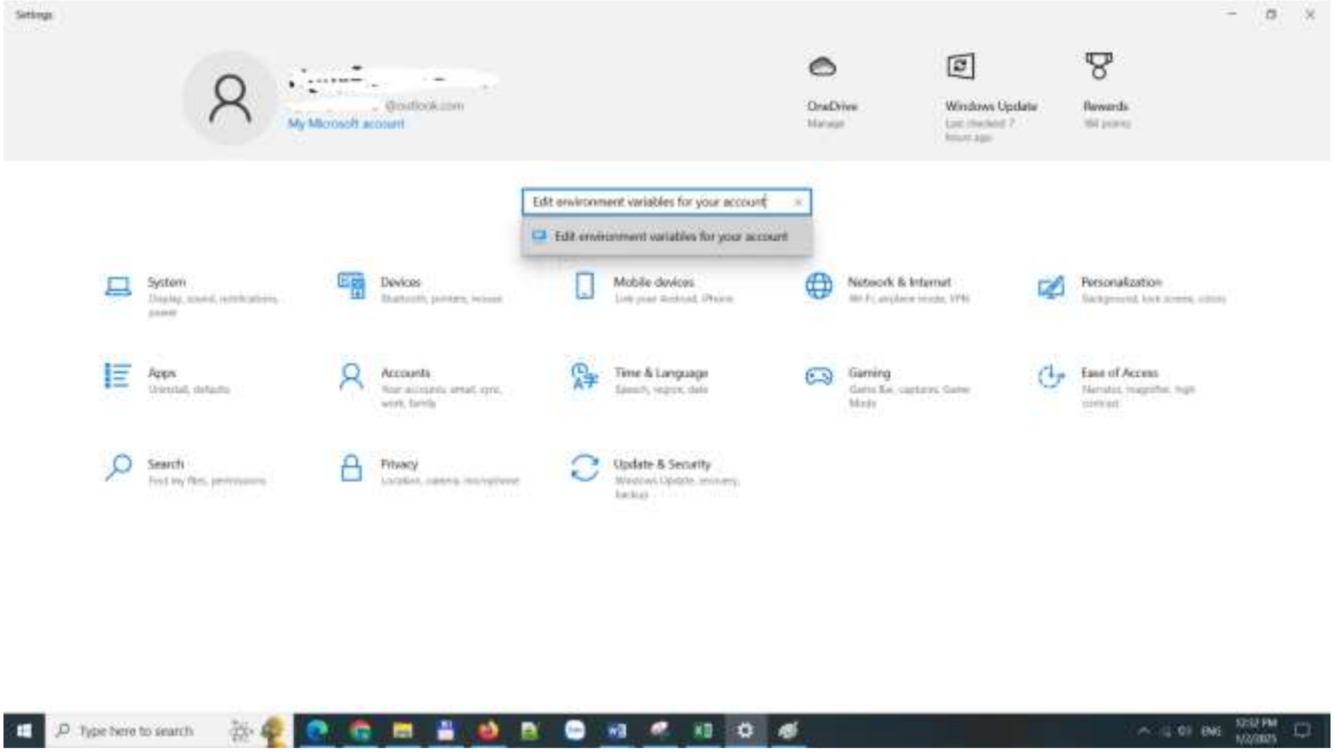
```
              mingw-w64-ucrt-x86_64-crt-git-11.0.0.r547.g4c8123efb-1
              mingw-w64-ucrt-x86_64-gcc-13.2.0-3  mingw-w64-ucrt-x86_64-gcc-ada-13.2.0-3
              mingw-w64-ucrt-x86_64-gcc-fortran-13.2.0-3
              mingw-w64-ucrt-x86_64-gcc-libgfortran-13.2.0-3
              mingw-w64-ucrt-x86_64-gcc-libs-13.2.0-3  mingw-w64-ucrt-x86_64-gcc-objc-13.2.0-3
              mingw-w64-ucrt-x86_64-gdb-14.1-1  mingw-w64-ucrt-x86_64-gdb-multiarch-14.1-1
              mingw-w64-ucrt-x86_64-headers-git-11.0.0.r547.g4c8123efb-1
              mingw-w64-ucrt-x86_64-libgccjit-13.2.0-3
              mingw-w64-ucrt-x86_64-libmangle-git-11.0.0.r547.g4c8123efb-1
              mingw-w64-ucrt-x86_64-libwinpthread-git-11.0.0.r547.g4c8123efb-1
              mingw-w64-ucrt-x86_64-make-4.4.1-1  mingw-w64-ucrt-x86_64-pkgconf-1~2.1.0-1
              mingw-w64-ucrt-x86_64-tools-git-11.0.0.r547.g4c8123efb-1
              mingw-w64-ucrt-x86_64-winpthreads-git-11.0.0.r547.g4c8123efb-1
              mingw-w64-ucrt-x86_64-winstorecompat-git-11.0.0.r547.g4c8123efb-1

Total Download Size:    171.77 MiB
Total Installed Size:  1141.23 MiB

:: Proceed with installation? [Y/n] Y
:: Retrieving packages...
 mingw-w64-ucrt-x86_64-gcc-...   703.9 KiB   115 KiB/s 04:10 [-------------------------------]   2%
 mingw-w64-ucrt-x86_64-pyth...    13.6 MiB  1259 KiB/s 00:07 [#################--------------]  58%
 mingw-w64-ucrt-x86_64-gcc-...    11.8 MiB  1435 KiB/s 00:06 [#################--------------]  57%
 mingw-w64-ucrt-x86_64-gcc-...     6.1 MiB  1764 KiB/s 00:04 [##############-----------------]  46%
 mingw-w64-ucrt-x86_64-gcc-...     2.4 MiB   418 KiB/s 00:21 [######-------------------------]  21%
 Total ( 0/58)                    34.6 MiB  4.85 MiB/s 00:28 [######-------------------------]  20%
```

4.4.4.10. Add the path of your **MinGW-w64** bin folder to the Windows **PATH** environment variable by using the following steps:

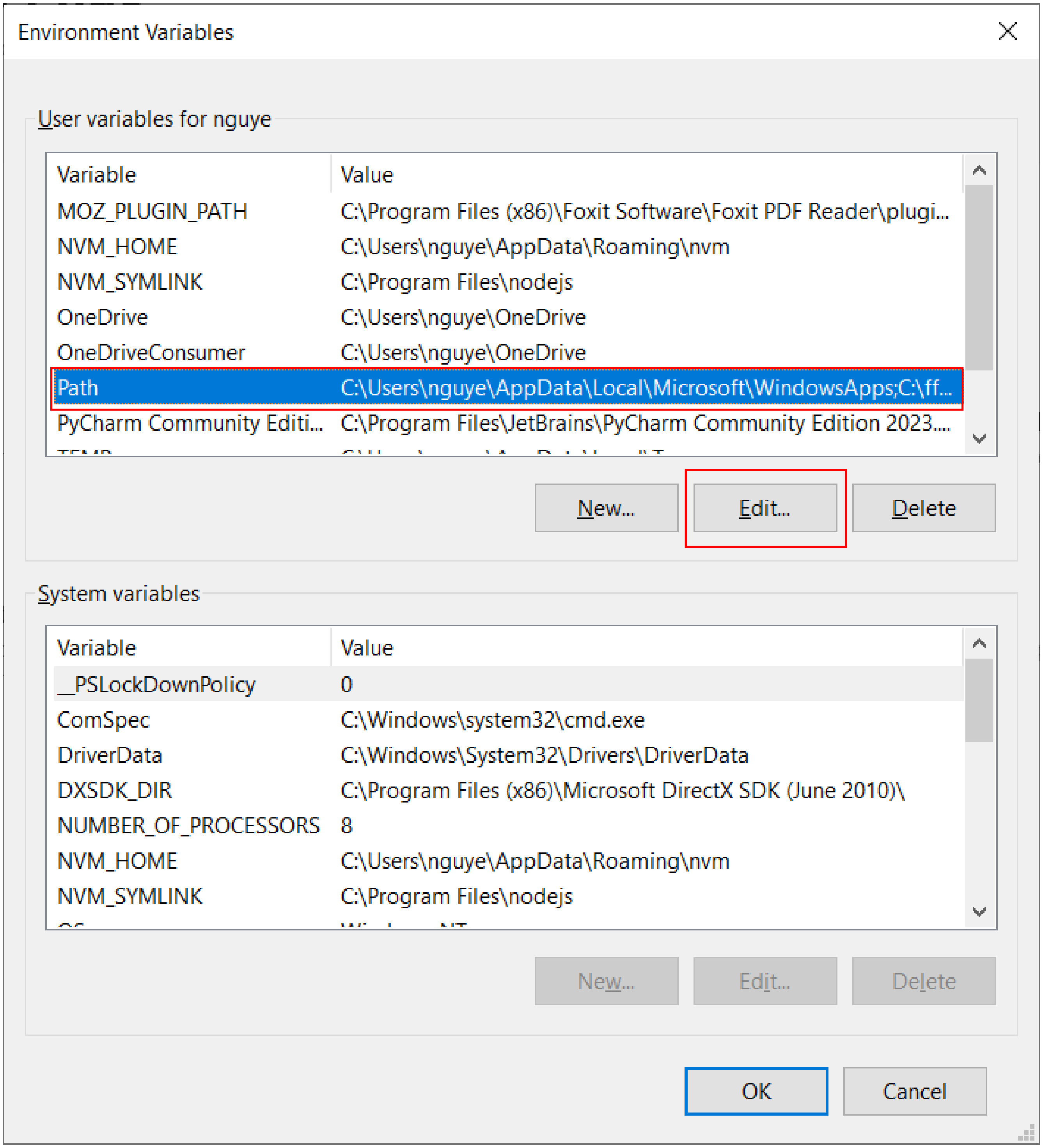
1. In the Windows search bar, type **Settings** to open your Windows Settings.


Settings
[redacted]
@outlook.com
My Microsoft account
OneDrive
Manage
Windows Update
Last checked 7 hours ago
Rewards
Find a setting
System
Display, sound, notifications, power
Devices
Bluetooth, printers, mouse
Mobile devices
Link your Android, iPhone
Network & Internet
Wi-Fi, airplane mode, VPN
Personalization
Background, lock screen, colors
Apps
Uninstall, defaults
Accounts
Your accounts, email, sync, work, family
Time & Language
Speech, region, date
Gaming
Game Bar, captures, Game Mode
Ease of Access
Narrator, magnifier, high contrast
Search
Find my files, permissions
Privacy
Location, camera, microphone
Update & Security
Windows Update, recovery, backup
Type here to search
ENG

2. Search for **Edit environment variables for your account**.


Edit environment variables for your account
Settings
[redacted]
@outlook.com
My Microsoft account
OneDrive
Manage
Windows Update
Last checked 7 hours ago
Rewards
Edit environment variables for your account
System
Display, sound, notifications, power
Devices
Bluetooth, printers, mouse
Mobile devices
Link your Android, iPhone
Network & Internet
Wi-Fi, airplane mode, VPN
Personalization
Background, lock screen, colors
Apps
Uninstall, defaults
Accounts
Your accounts, email, sync, work, family
Time & Language
Speech, region, date
Gaming
Game Bar, captures, Game Mode
Ease of Access
Narrator, magnifier, high contrast
Search
Find my files, permissions
Privacy
Location, camera, microphone
Update & Security
Windows Update, recovery, backup
Type here to search
ENG

3. In your **User variables**, select the **Path** variable and then select **Edit**.



Environment Variables

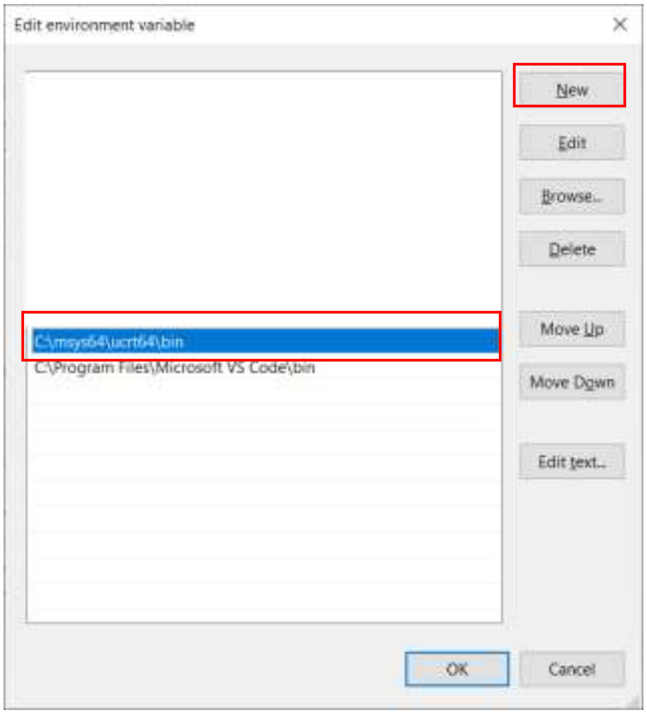
User variables for nguye

Variable	Value
MOZ_PLUGIN_PATH	C:\Program Files (x86)\Foxit Software\Foxit PDF Reader\plugi...
NVM_HOME	C:\Users\nguye\AppData\Roaming\nvm
NVM_SYMLINK	C:\Program Files\nodejs
OneDrive	C:\Users\nguye\OneDrive
OneDriveConsumer	C:\Users\nguye\OneDrive
Path	C:\Users\nguye\AppData\Local\Microsoft\WindowsApps;C:\ff...
PyCharm Community Editi...	C:\Program Files\JetBrains\PyCharm Community Edition 2023....

System variables

Variable	Value
__PSLockDownPolicy	0
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
DXSDK_DIR	C:\Program Files (x86)\Microsoft DirectX SDK (June 2010)\
NUMBER_OF_PROCESSORS	8
NVM_HOME	C:\Users\nguye\AppData\Roaming\nvm
NVM_SYMLINK	C:\Program Files\nodejs

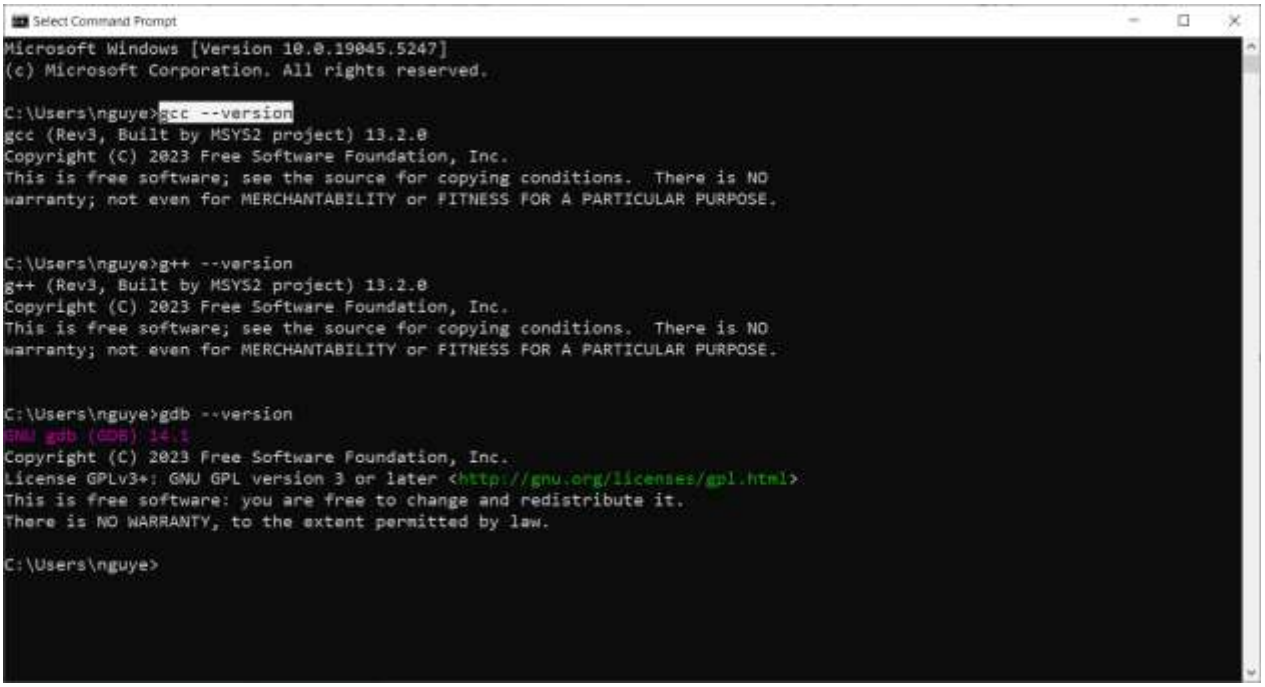
4. Select **New** and add the MinGW-w64 destination folder you recorded during the installation process to the list. If you used the default settings above, then this will be the path: C:\msys64\ucrt64\bin.



C:\msys64\ucrt64\bin
C:\Program Files\Microsoft VS Code\bin

5. Select **OK**, and then select **OK** again in the **Environment Variables** window to update the PATH environment variable. You have to restart windows for the updated PATH environment variable to be available.

6. Check your **MinGW installation**: open a new Command Prompt and type: gcc –version, g++ --version, gdb --version



```
Microsoft Windows [Version 10.0.19045.5247]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nguye>gcc --version
gcc (Rev3, Built by MSYS2 project) 13.2.0
Copyright (C) 2023 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.


C:\Users\nguye>g++ --version
g++ (Rev3, Built by MSYS2 project) 13.2.0
Copyright (C) 2023 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.


C:\Users\nguye>gdb --version
GNU gdb (GDB) 14.1
Copyright (C) 2023 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

C:\Users\nguye>
```