Nicky Masimula

# SE-Assignment-5

Installation and Navigation of Visual Studio Code (VS Code)

Instructions:

Answer the following questions based on your understanding of the installation and navigation of Visual Studio Code (VS Code). Provide detailed explanations and examples where appropriate.

Questions:

1. Installation of VS Code:
   - Describe the steps to download and install Visual Studio Code on Windows 11 operating system. Include any prerequisites that might be needed.

Prerequisites

Administrator Privileges: You need administrative rights on your Windows 11 computer to install software.

Internet Connection: Ensure you have a stable internet connection to download the installer.

Steps to Download and Install Visual Studio Code on Windows 11

Download the Installer:Open your web browser and go to the official Visual Studio Code website: https://code.visualstudio.com/.

Click on the Download for Windows button. This downloads the latest stable version of the VS Code installer (VSCodeSetup-{version}.exe).

Run the Installer:

Once the download completes, locate the downloaded installer file (usually in your Downloads folder) and double-click it (VSCodeSetup-{version}.exe).If prompted by the User Account Control (UAC), click Yes to allow the installer to make changes to your device.

Install Visual Studio Code:

The installer will launch. Click Next to proceed with the installation.

Review the license agreement, select I accept the agreement, and click Next.

Choose the destination folder where you want to install VS Code or leave the default location, and then click Next.

Optionally, you can choose to create shortcuts for launching VS Code in the Start Menu and on the desktop. Make your selections and click Next.

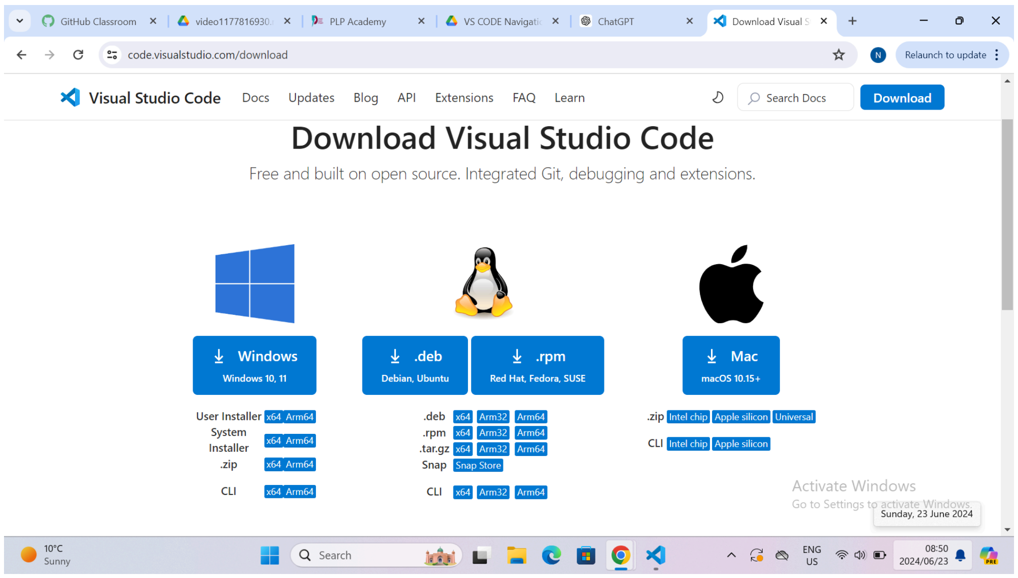
Click Install to begin the installation process.

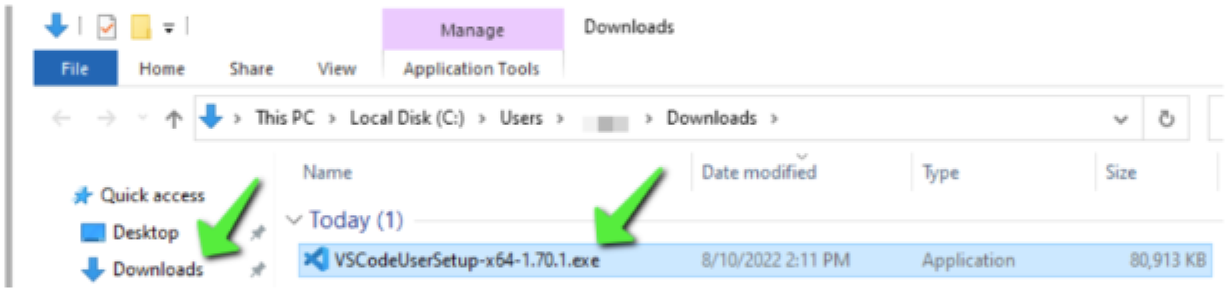
Completing the Installation:

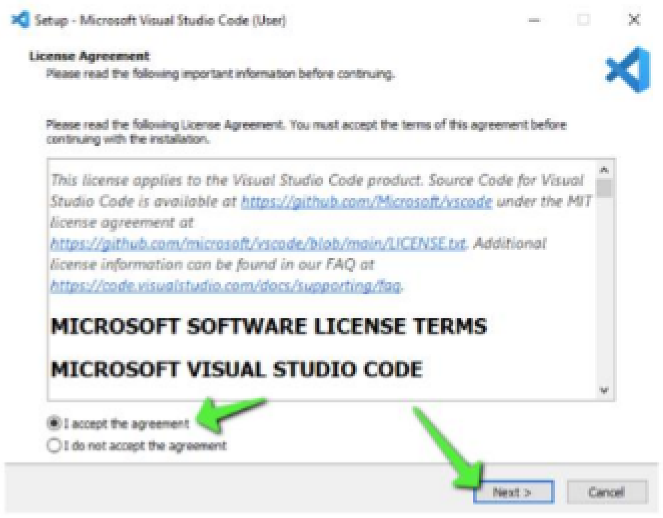
Once the installation completes, click Finish.

Launch Visual Studio Code:

Refer to the images below:







Setup - Microsoft Visual Studio Code (User)

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.

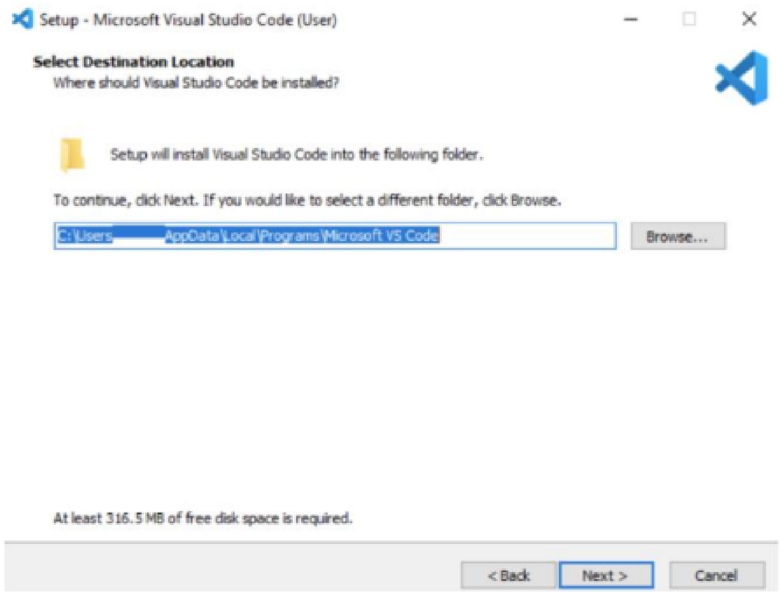
This license applies to the Visual Studio Code product. Source Code for Visual Studio Code is available at https://github.com/Microsoft/vscode under the MIT license agreement at https://github.com/microsoft/vscode/blob/main/LICENSE.txt. Additional license information can be found in our FAQ at https://code.visualstudio.com/docs/supporting/faq.

MICROSOFT SOFTWARE LICENSE TERMS

MICROSOFT VISUAL STUDIO CODE

[✓ I accept the agreement]

I do not accept the agreement



Setup - Microsoft Visual Studio Code (User)

Select Destination Location

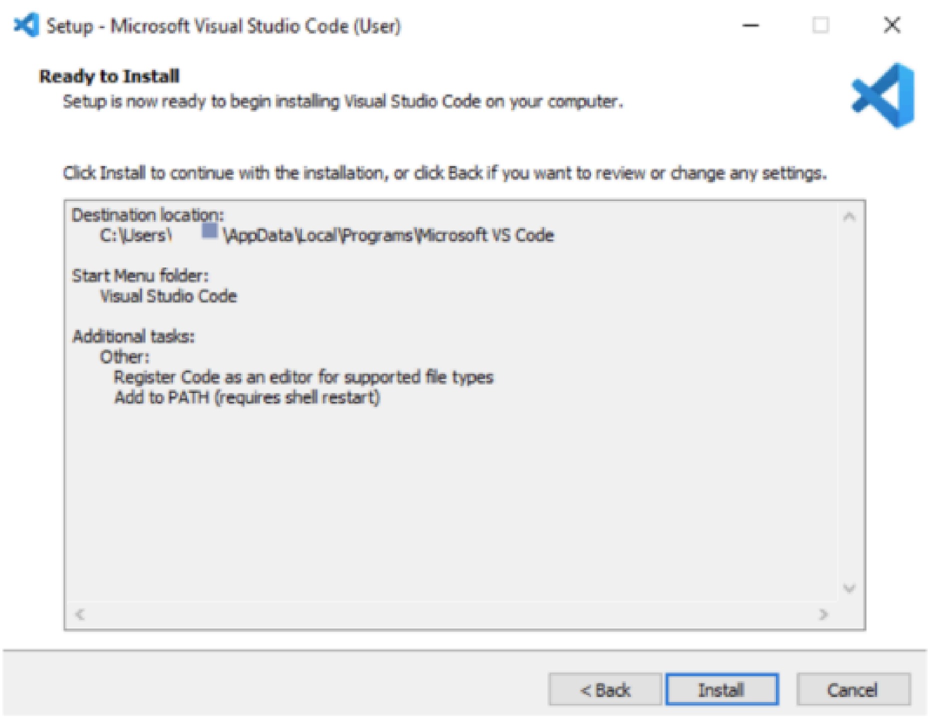
Where should Visual Studio Code be installed?

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:\Users\[redacted]AppData\Local\Programs\Microsoft VS Code

At least 316.5 MB of free disk space is required.



Setup - Microsoft Visual Studio Code (User)

Ready to Install

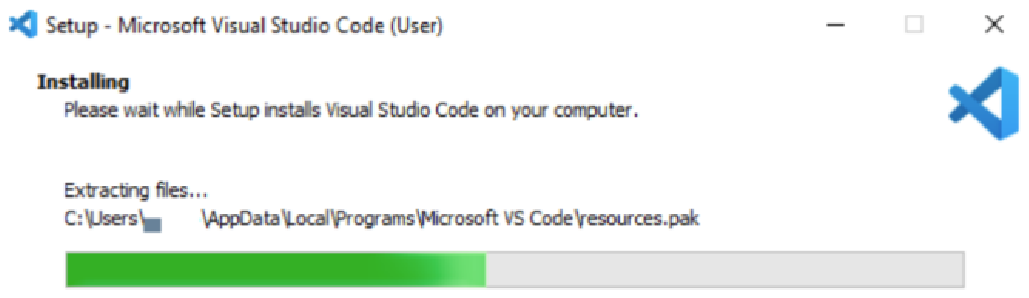
Setup is now ready to begin installing Visual Studio Code on your computer.

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

```
Destination location:
    C:\Users\ [redacted] \AppData\Local\Programs\Microsoft VS Code

Start Menu folder:
    Visual Studio Code

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

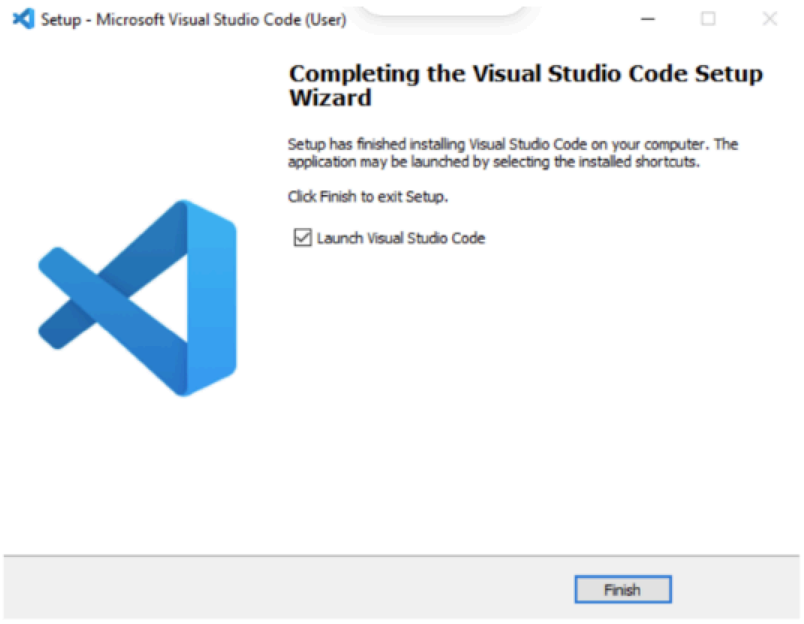


Setup - Microsoft Visual Studio Code (User)

Installing

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

```
Extracting files...
C:\Users\ [redacted] \AppData\Local\Programs\Microsoft VS Code\resources.pak
```



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

2. First-time Setup:

- After installing VS Code, what initial configurations and settings should be adjusted for an optimal coding environment? Mention any important settings or extensions.

Initial Configurations

Theme and Color Scheme:

Choose a theme that is comfortable for your eyes and enhances readability. You can change the theme by clicking on the gear icon in the bottom-left corner (Settings) and selecting Color Theme.

Font and Font Size:

Adjust the editor font and font size to your liking. Search for editor.fontFamily and editor.fontSize in the Settings to modify these preferences.

Indentation and Formatting:

Set your preferred indentation style (tabs or spaces) and configure formatting rules for languages you commonly use. Adjust settings like editor.tabSize and editor.insertSpaces in the Settings.

3. User Interface Overview:

- Explain the main components of the VS Code user interface. Identify and describe the purpose of the Activity Bar, Side Bar, Editor Group, and Status Bar.

Here are the main components and features of VS Code:

Activity Bar: Located on the far left, the activity bar provides quick access to different views and panels, including those mentioned in the Sidebar and other extensions you've installed.

Sidebar: Located on the left-hand side, the sidebar provides quick access to different views and panels:

Explorer: Shows the file structure of your project, allowing you to navigate files and folders.

Search: Provides tools for finding and replacing text across files.

Source Control: Integrates with version control systems like Git, allowing you to manage changes to your codebase.

Extensions: Enables you to browse and install extensions that add functionality to VS Code.

Editor: This is the primary workspace where you write your code. VS Code supports syntax highlighting, code completion, and code formatting for various programming languages

Status Bar: Located at the bottom, the status bar displays information such as the current programming language, line and column number, Git branch status, and notifications from installed extensions.

4. Command Palette:
   - What is the Command Palette in VS Code, and how can it be accessed? Provide examples of common tasks that can be performed using the Command Palette.

Command Palette: Accessed via Ctrl+Shift+P (or Cmd+Shift+P on macOS), the command palette provides a quick way to access and execute commands, such as opening files, running tasks, and invoking extensions.

5. Extensions in VS Code:

- Discuss the role of extensions in VS Code. How can users find, install, and manage extensions? Provide examples of essential extensions for web development.

VS Code is highly extensible through extensions, which can add new languages, themes, debuggers, and other tools. Extensions can be installed and managed through the Extensions view in the Sidebar.

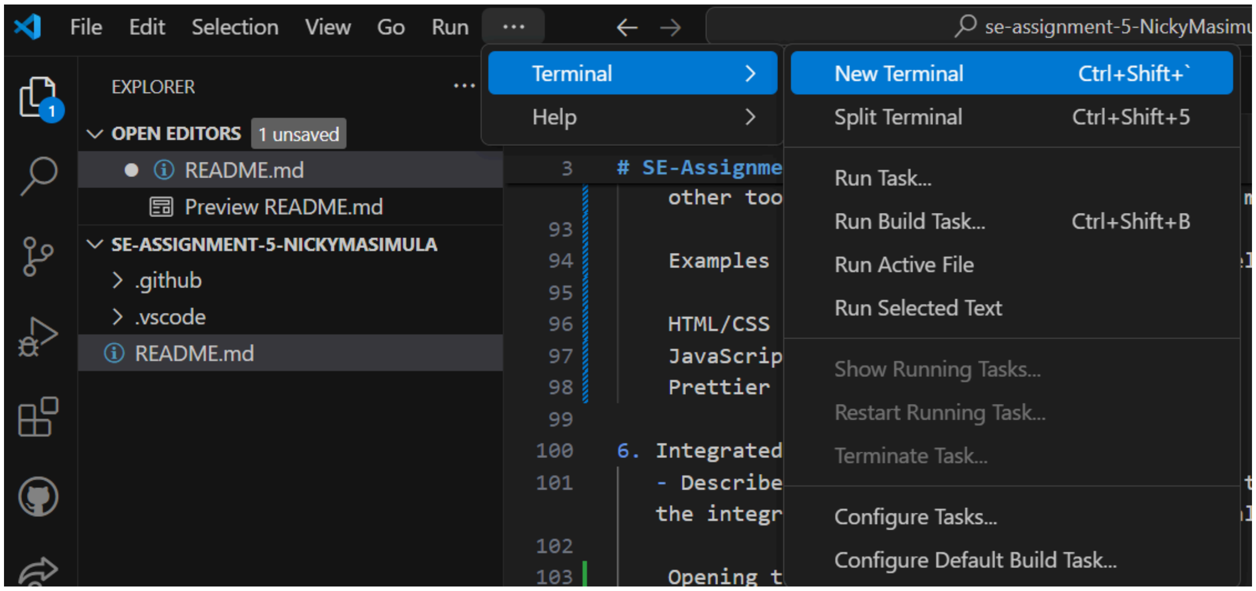
Examples of essential extensions for web development are as follows:
HTML/CSS
JavaScript/TypeScript
Prettier - Code formatter

6. Integrated Terminal:
   - Describe how to open and use the integrated terminal in VS Code. What are the advantages of using the integrated terminal compared to an external terminal?

Opening the Integrated Terminal
Open VS Code: Start by opening Visual Studio Code on your computer.

Access the View Menu:
You can open the integrated terminal in several ways. You can open it with Ctrl + (Cmd + on Mac) or use the View menu in the top menu bar of VS Code.

Refer to the image below:



Advantages of using integrated terminal:
You don't need to switch between different applications or windows, which can save time and reduce distractions

The integrated terminal automatically opens at the root of your workspace. This means it always starts in the context of your project, making it easier to navigate files and run commands relevant to your current work.

You can easily interact with files and folders within your VS Code workspace directly from the terminal. For example, you can open files by typing code filename or code . to open the entire workspace.

You can customize the integrated terminal's appearance and behavior through VS Code's settings and extensions. This includes adjusting font size, colors, shell configurations, and more, which can enhance your comfort and productivity.

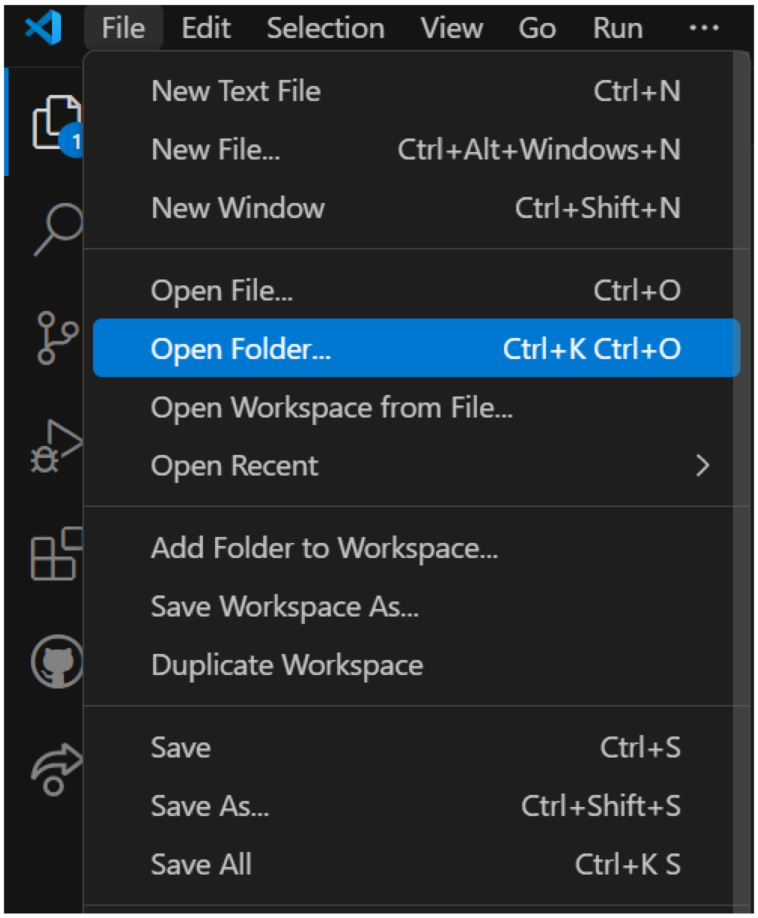
VS Code's integrated terminal can be extended with additional functionality through extensions available in the marketplace. This allows you to tailor the terminal to fit specific needs such as additional shell support or integration with external tools.

When debugging applications within VS Code, the integrated terminal provides seamless interaction with debug commands and outputs, enhancing the debugging experience.

7. File and Folder Management:
  - Explain how to create, open, and manage files and folders in VS Code. How can users navigate between different files and directories efficiently?

Creating a New File or Folder:
To create a new file, you can use one of the following methods:
Click on the Explorer icon in the Activity Bar on the left (or use Ctrl+Shift+E on Windows/Linux, Cmd+Shift+E on macOS) to open the Explorer view.
Right-click on the parent folder where you want to create the file.
Select New File from the context menu, then type the desired filename and press Enter.
To create a new folder, follow the above steps but choose New Folder instead.

Refer to the image below:



8. Settings and Preferences:
   - Where can users find and customize settings in VS Code? Provide examples of how to change the theme, font size, and keybindings.

In Visual Studio Code (VS Code), users can find and customize settings through several methods:

Settings Icon in the Sidebar:
Look for a gear icon ⚙️ in the Activity Bar (usually on the left side).
Clicking this icon opens the Settings view.

Command Palette:
Use the keyboard shortcut Ctrl+Shift+P (Windows/Linux) or Cmd+Shift+P (Mac) to open the Command Palette.

Menu Bar:
On macOS, click on Code in the menu bar, then select Preferences and Settings.
On Windows and Linux, click on File, then select Preferences and Settings.
Type Preferences: Open Settings (JSON) to directly edit the settings in JSON format, or Preferences: Open Settings (UI) to open the graphical interface for settings.

Keyboard Shortcuts:
Use Ctrl+, (Windows/Linux) or Cmd+, (Mac) to directly open the Settings view.

Search Settings:
Once in the Settings view, there's a search bar at the top.
Type keywords related to the setting you want to customize (e.g., font size, theme) to quickly find relevant settings.

Customizing Settings:

User Settings vs Workspace Settings:
User settings affect all instances of VS Code.
Workspace settings are specific to the current workspace/project.

Editing Settings:
Settings can be edited in JSON format or using the graphical UI.
Click on the pencil icon next to a setting to edit its value.
To reset a setting, hover over it and click on the reset icon (curved arrow).

Extensions Settings:
Some extensions also add their own settings, which can be found in the same Settings view under the Extensions category or directly under their own section.

Examples of how to change the theme, font size and keybindings:
Changing the Theme:
Using the Settings:
Press Ctrl+, (Windows/Linux) or Cmd+, (Mac) to open the Settings.
In the search bar, type Color Theme.
Click on the drop-down under "Color Theme" and select a theme from the list.

Changing the Font Size:
Using the Settings:
Press Ctrl+, (Windows/Linux) or Cmd+, (Mac) to open the Settings.
In the search bar, type Font Size.
Use the slider or enter a numeric value to adjust the font size under the "Editor: Font Size" section.

Changing Keybindings:

Using the Settings:

- Press `Ctrl+,` (Windows/Linux) or `Cmd+,` (Mac) to open the Settings.
- In the search bar, type `Keybindings`

9. Debugging in VS Code:
   - Outline the steps to set up and start debugging a simple program in VS Code. What are some key debugging features available in VS Code?

Steps to Set Up and Start Debugging

Open Your Project in VS Code:
Launch VS Code.
Open your project folder using File > Open Folder... or by dragging the folder into the VS Code window.

Configure Debugging Environment:
Click on the Run and Debug icon in the Activity Bar on the left (or press Ctrl+Shift+D).
Click on the Configure gear icon or select a debug configuration if one is already available.
If no configuration exists, click on Add Configuration... and select your programming language.
This action generates a launch.json file in the .vscode directory with default configurations.
Edit launch.json (if needed):

Open launch.json (created in step 2) and customize configurations as necessary. For example, you might specify the program entry point, arguments, environment variables, etc.

Set Breakpoints:
Open the file containing the code you want to debug.
Click in the gutter next to the line number where you want to set a breakpoint. A red dot appears, indicating the breakpoint.

Start Debugging:
Press F5 or click Start Debugging in the “Run and Debug view”.
VS Code launches the debugger and starts running your program.
Execution pauses at the breakpoints, allowing you to inspect variables, step through code, and analyze program state.
Visual Studio Code (VS Code) offers several powerful debugging features that help developers diagnose and resolve issues in their code efficiently. Here are some key debugging features available in VS Code:

Breakpoints:
Set breakpoints in your code by clicking in the gutter next to a line number. When the program execution reaches a breakpoint, it pauses, allowing you to inspect variables and the call stack.

Variable Inspection:
View the current values of variables in your code during debugging. Variables can be inspected in the Variables view, where you can see their names, types, and current values.

Watch Expressions:
Define custom expressions to monitor during debugging. Watch expressions allow you to track specific variables, properties, or complex expressions and see how their values change as you step through your code.

Call Stack Navigation:
Navigate through the call stack to understand the flow of execution and identify where issues might be occurring. The Call Stack view shows the sequence of function calls leading to the current point of execution.

Debug Console:
Access the Debug Console to interactively execute commands and view output from your program during debugging. This is useful for logging messages or executing expressions to understand program behavior.

Step Through Code:
Use step-by-step debugging to control the execution flow:
Step Over: Execute the current line and move to the next line in the current file.
Step Into: Move into a function call and begin debugging inside that function.
Step Out: Execute the remaining lines of the current function and return to the caller.

Conditional Breakpoints:
Set breakpoints that only trigger when specific conditions are met. This feature helps you debug specific scenarios without stopping at every occurrence of a breakpoint.

Exception Handling:
Configure VS Code to break execution on exceptions or specific types of exceptions. This allows you to catch and investigate errors as they occur in your code.

Debugging Configuration:
Customize debugging configurations in launch.json to specify how your program should be debugged. You can define program entry points, arguments, environment variables, and more.

Multi-threaded Debugging:
Debug applications that involve multiple threads or processes. VS Code supports debugging scenarios where multiple threads are running simultaneously, providing tools to inspect and manage thread-specific state.

Remote Debugging:
Debug applications running on remote machines or environments (e.g., Docker containers, remote servers). VS Code allows you to attach to remote processes and debug them as if they were local.

Integrated Terminal:
Access an integrated terminal directly within VS Code for running commands, debugging scripts, and interacting with your development environment without switching to an external terminal.

10. Using Source Control:
   - How can users integrate Git with VS Code for version control? Describe the process of initializing a repository, making commits, and pushing changes to GitHub.

Integrating Git with Visual Studio Code (VS Code) for version control allows you to efficiently manage your codebase, track changes, and collaborate with others using GitHub or other Git hosting services.

Here’s a step-by-step guide on how to initialize a repository, make commits, and push changes to GitHub using VS Code:

Open Your Project in VS Code:
Launch VS Code.
Open your project folder using File > Open Folder or drag the folder into the VS Code window.

Initialize Git Repository:

Open the Source Control view in VS Code by clicking on the Source Control icon in the Activity Bar on the left (Ctrl+Shift+G).

Click Initialize Repository or use the Initialize Repository button in the Source Control view.

Alternatively, you can initialize Git via the terminal:

Open the integrated terminal in VS Code (`Ctrl+``).

Navigate to your project directory.

Run the command:

```
git init
```

Making Commits:

Make changes to your files within VS Code.

In the Source Control view, all modified files will appear.

To stage changes, click the + button next to each file you want to include in the commit.

Alternatively, stage all changes using the + button at the top of the Source Control view.

Commit Changes:

Once your changes are staged, enter a commit message in the text box at the top of the Source Control view.

Press Ctrl+Enter or click the checkmark icon to commit your changes.

Alternatively, commit via the terminal:

```
git commit -m "Your commit message"
```

Push Changes to GitHub:

Push your committed changes to GitHub by opening the Command Palette (Ctrl+Shift+P) and typing *Git: Push.*

Select the remote repository (origin if you followed the default naming).

Enter your GitHub credentials (username and password or access token) if prompted.

VS Code will push your changes to the remote repository on GitHub.

References:

PLP LMS - https://plpacademy.powerlearnproject.org/

ChatGPT - https://openai.com/chatgpt/

Youtube - https://www.youtube.com/results?search_query=freecodecamp+vs+code

VS Code website - https://code.visualstudio.com/download