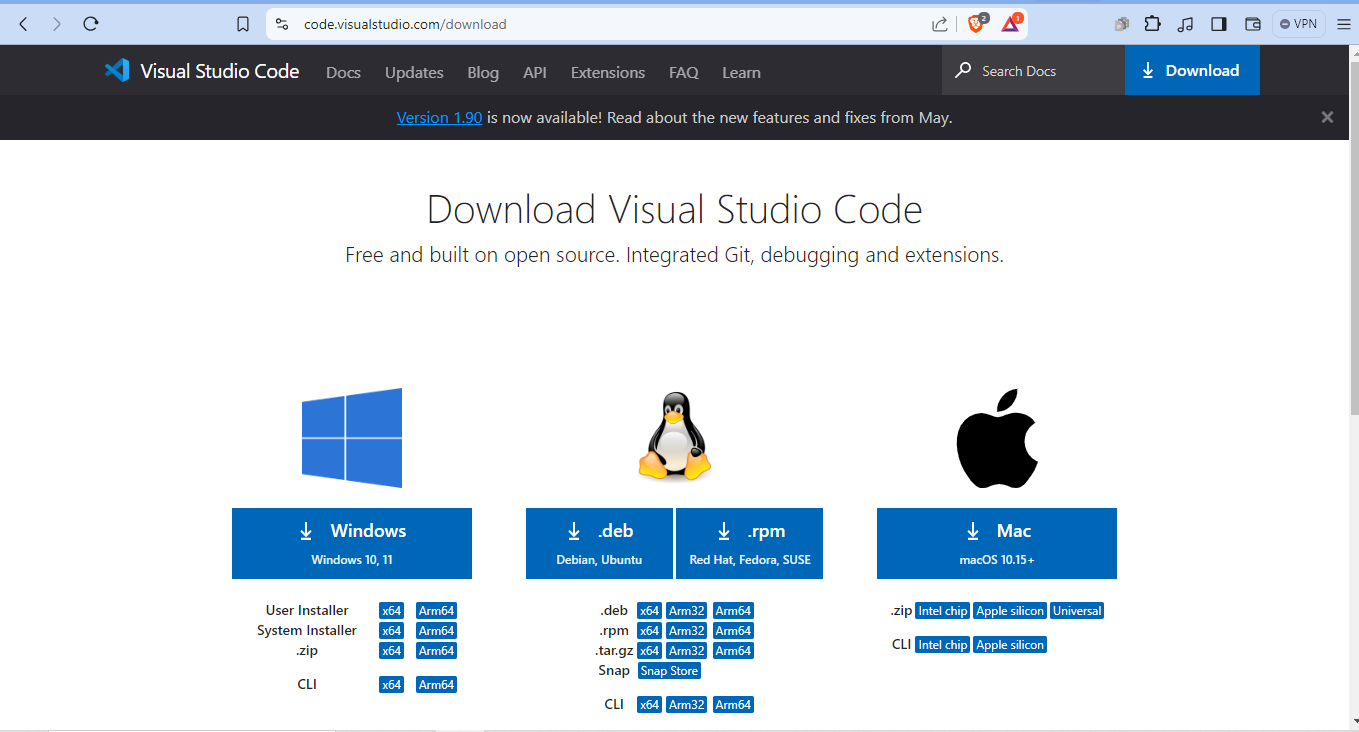
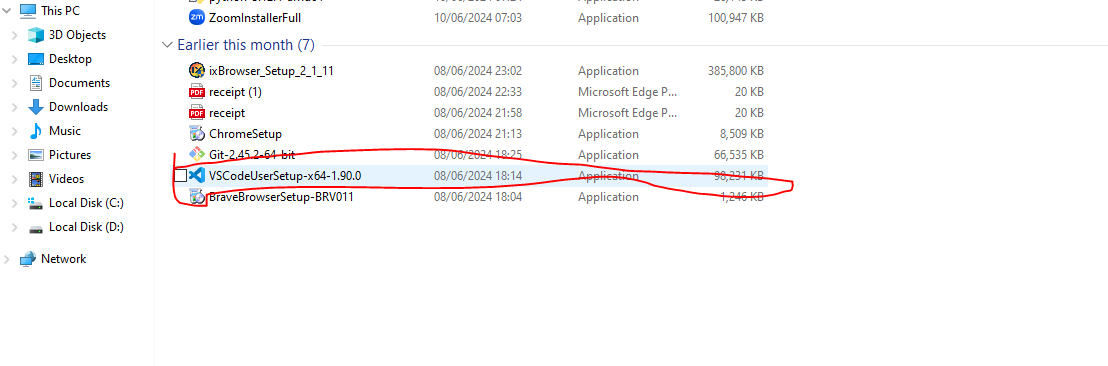
**This is a Visual studio installation documentation**

Step 1. Open a web browser and go to the official Visual Studio Code website.



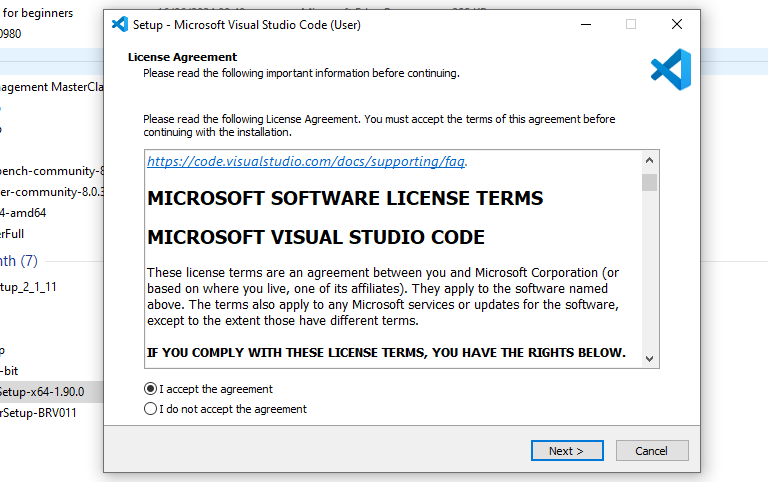
Step 2: Click on the "Download for Windows" button. This will download the installer (VSCodeUserSetup-x64-1.x.x.exe).

step 3: Installation:

Locate the downloaded installer file as shown above and double-click to run it.

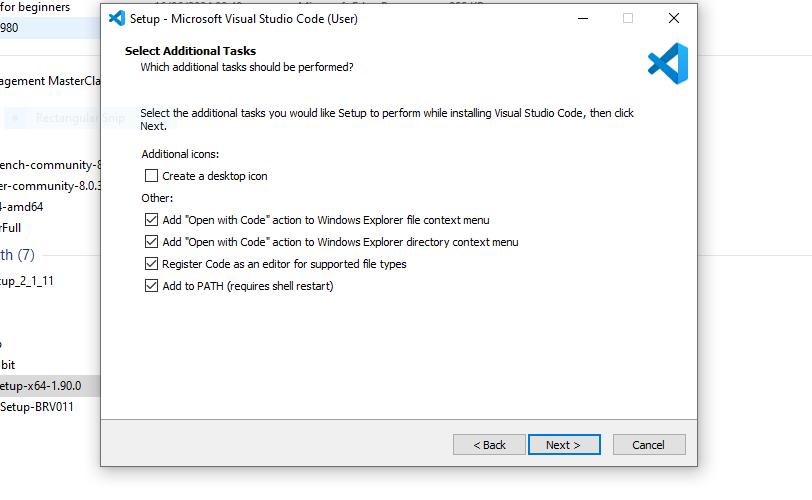
Follow the setup wizard instructions:

Accept the license agreement.



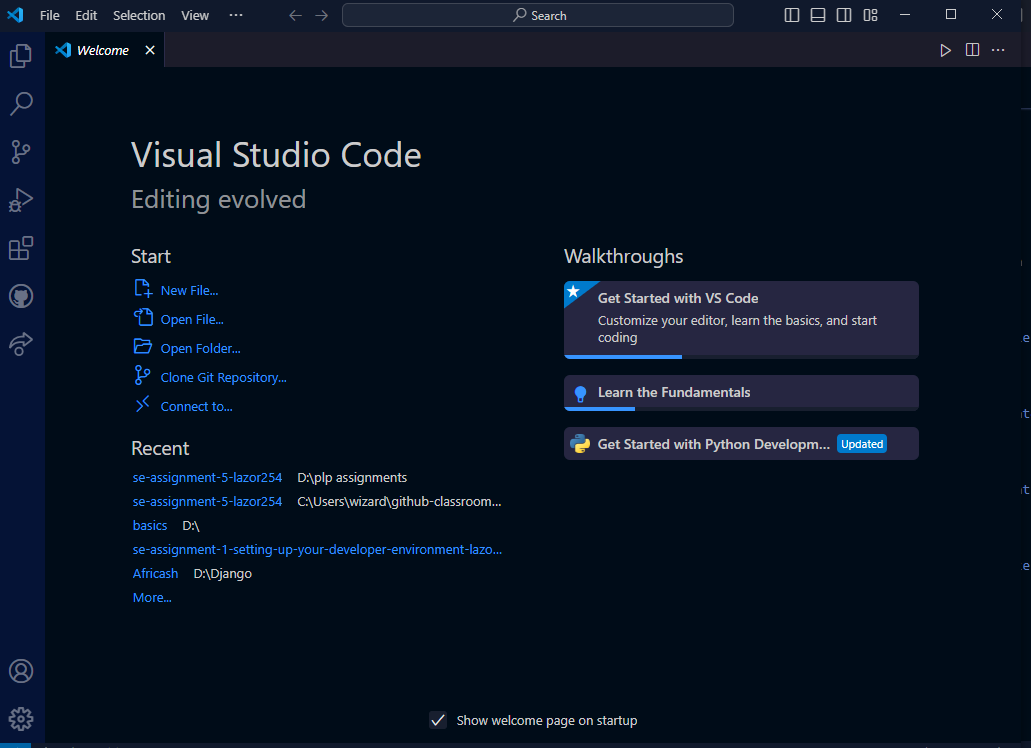
Choose the installation location (default is usually fine).

Select additional tasks (such as creating a desktop icon and adding to PATH).



Click "Install" to begin the installation.

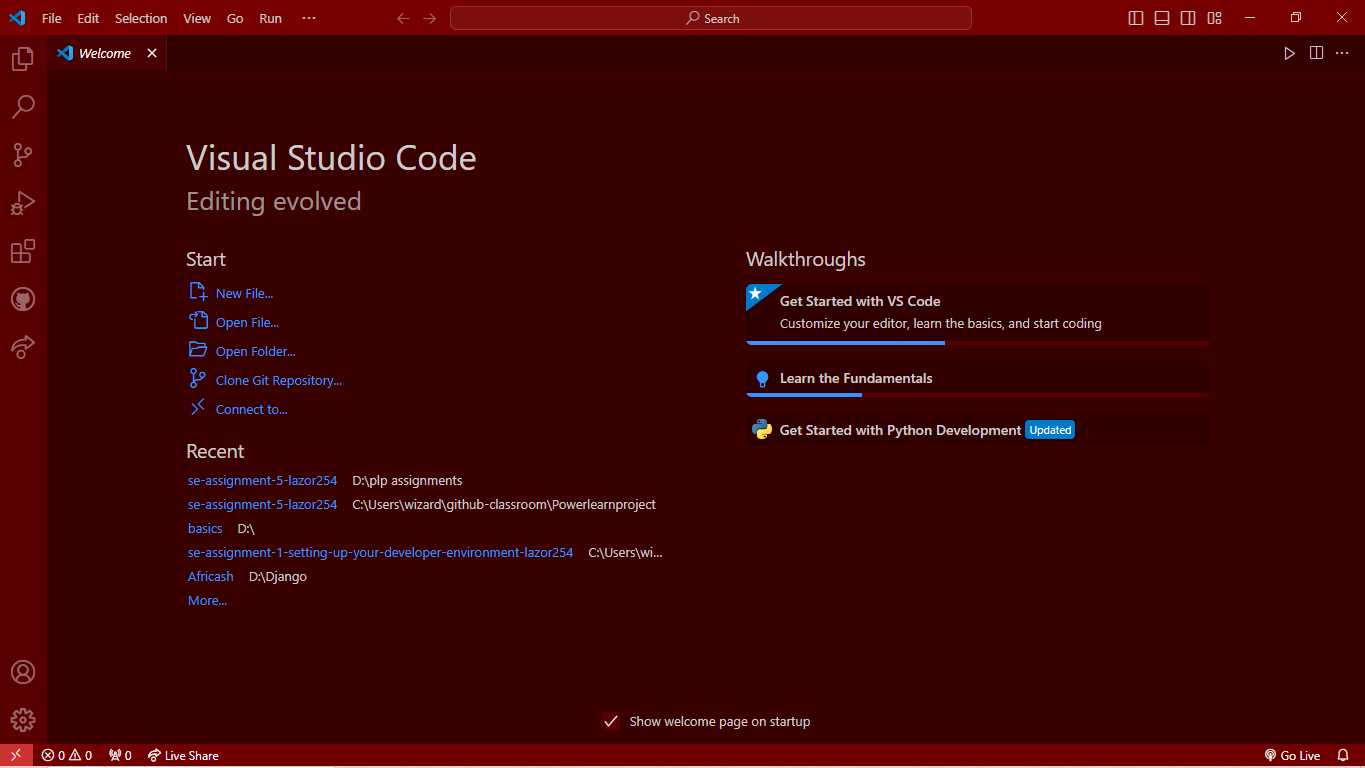
Once the installation is complete, click "Finish" to launch VS Code.



step 4: Initial Configurations and Settings:

Theme and Appearance:

Go to File > Preferences > Color Theme to choose a theme that suits your preference.



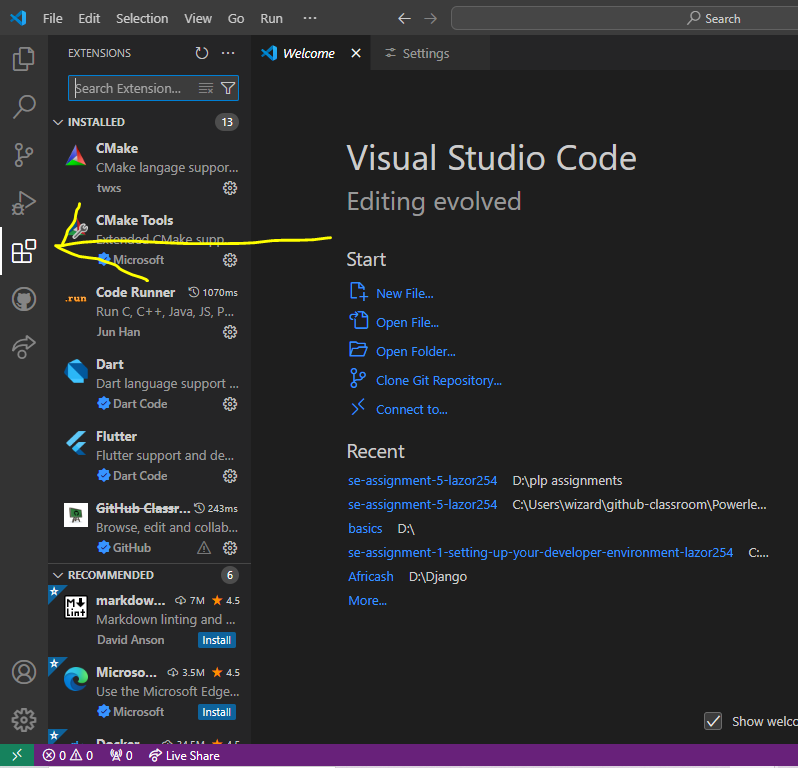
Font Size and Family:

Navigate to File > Preferences > Settings.

Search for "Font Size" and "Font Family" to adjust these settings for better readability.

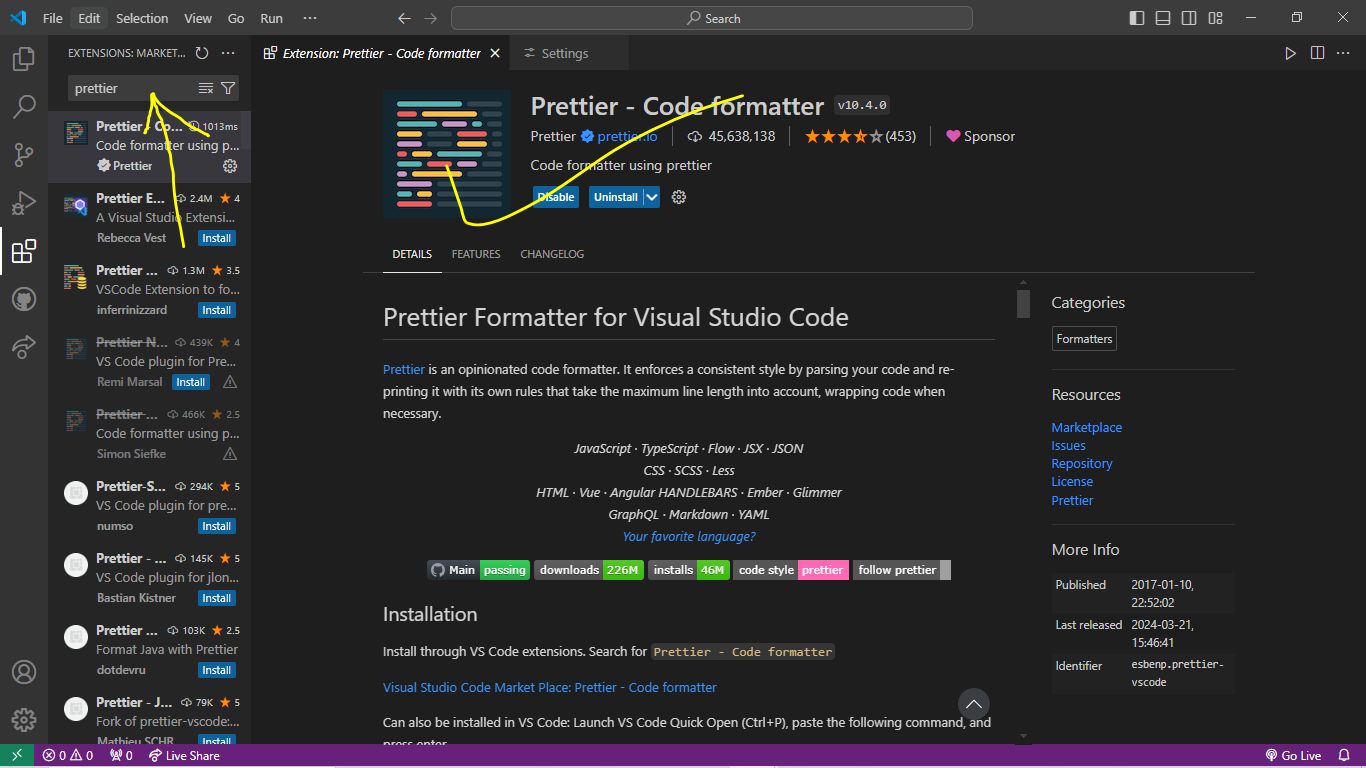
Step 5: Extensions:

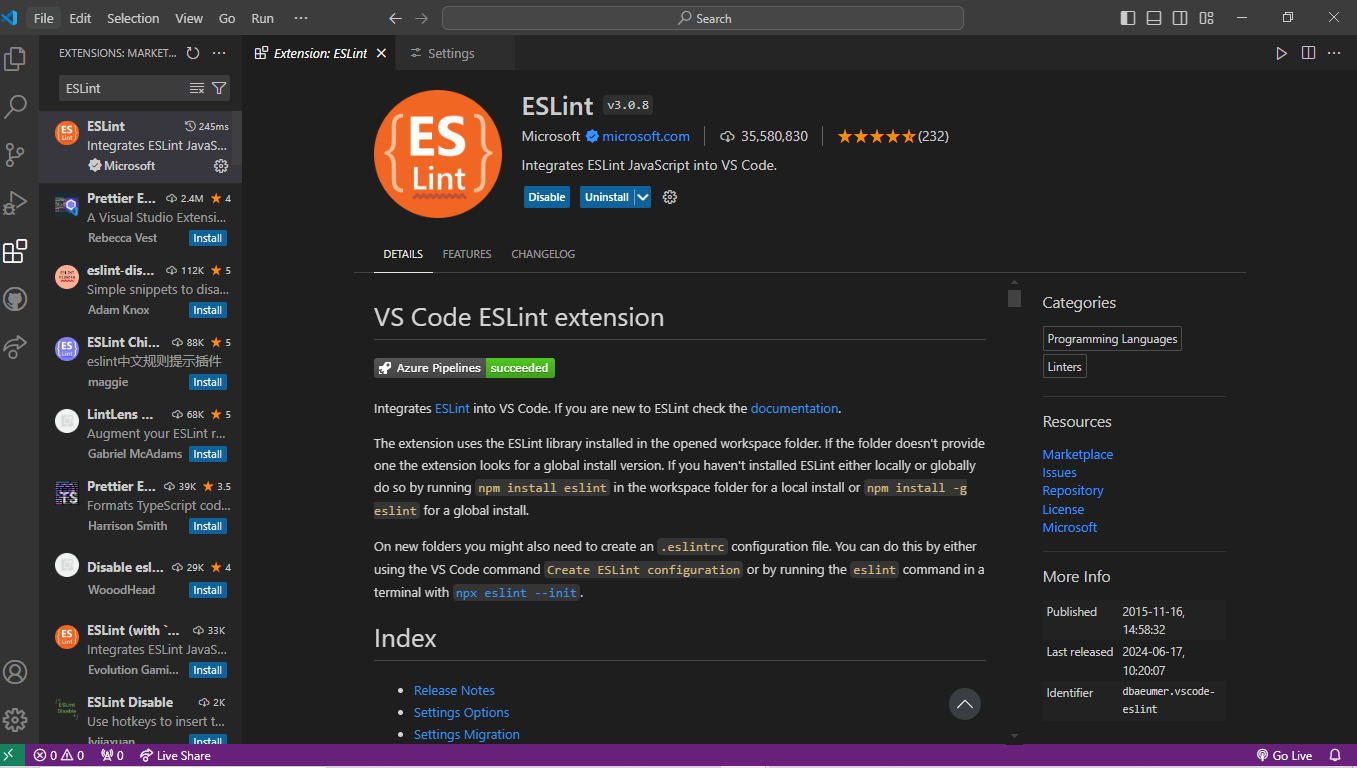
Open the Extensions view by clicking on the Extensions icon in the Activity Bar or pressing Ctrl+Shift+X.



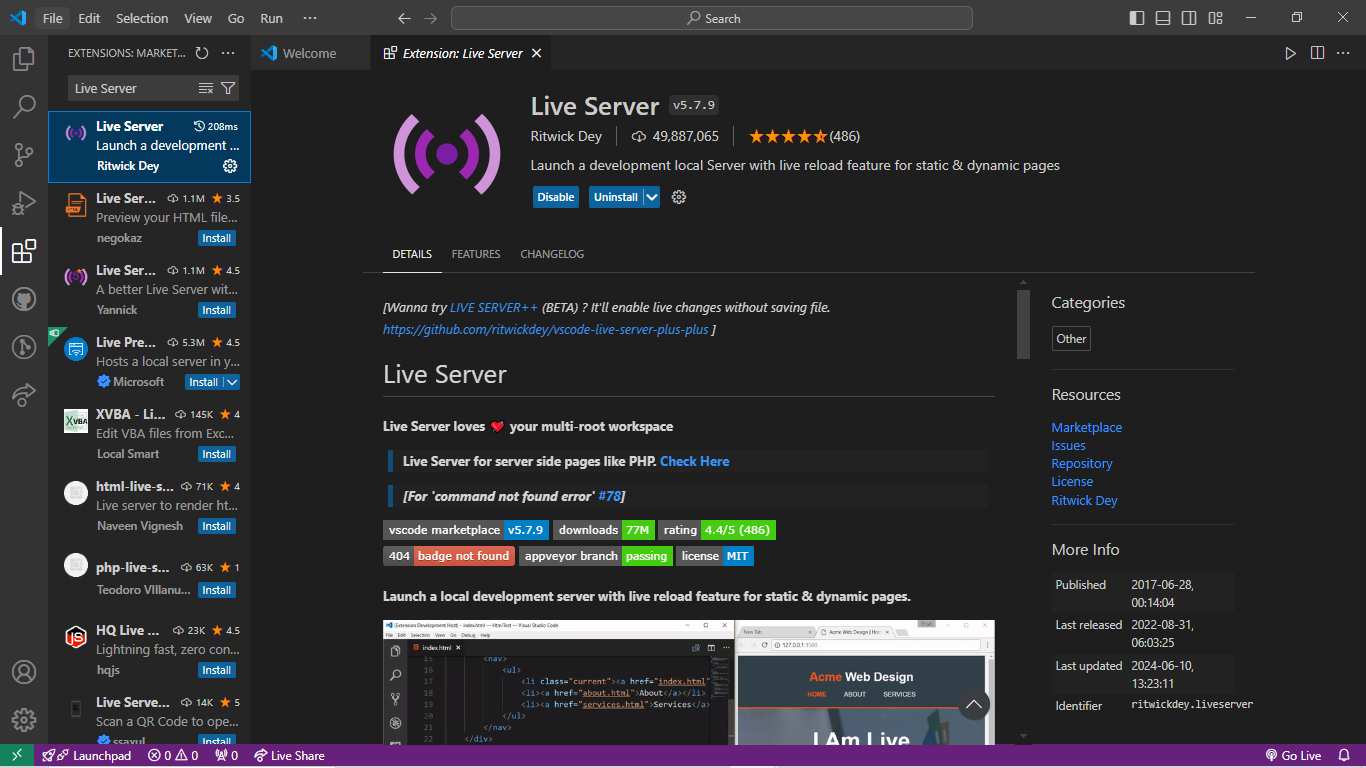
Essential extensions for an optimal coding environment:

Prettier: Code formatter.





Live Server: Local server with live reload.



Step 6: Workspace Settings:

Configure specific settings for your projects by opening File > Preferences > Settings and selecting the Workspace tab.

step 7: Main Components of the VS Code User Interface:

Activity Bar:

Located on the left side of the window.

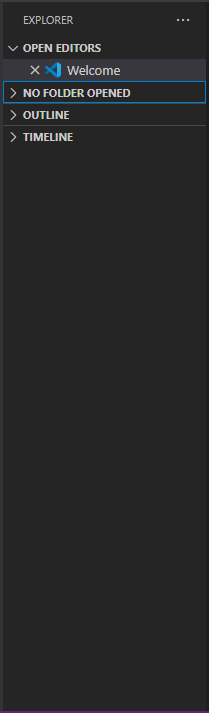
Provides access to different views like Explorer, Search, Source Control, Run and Debug, and Extensions.



Side Bar:

Adjacent to the Activity Bar.

Displays contextual information and actions for the selected view (e.g., file explorer, search results, source control changes).



Editor Group:

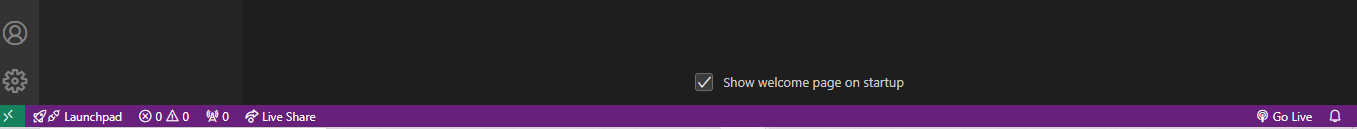
Central area where you edit your files.

Supports multiple editors side-by-side.

Status Bar:

Located at the bottom of the window.

Shows information about the current workspace, such as file encoding, line endings, Git branch, and errors.

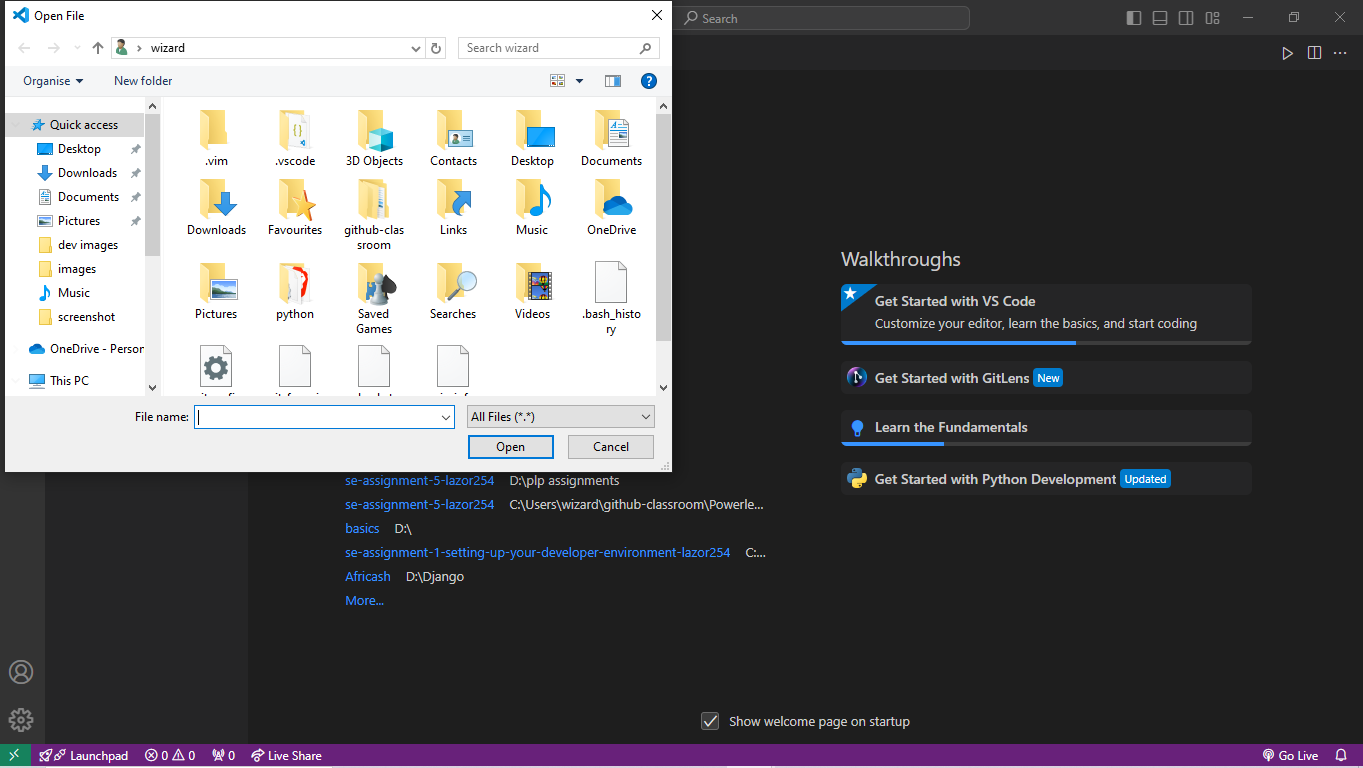


step 8: Command Palette

The Command Palette is a powerful tool that allows you to access all VS Code commands.

It can be accessed by pressing Ctrl+Shift+P or F1.

Common Task Using Command Palette: Open file:



step 9: Finding Extensions:

Click the Extensions icon in the Activity Bar or press Ctrl+Shift+X.

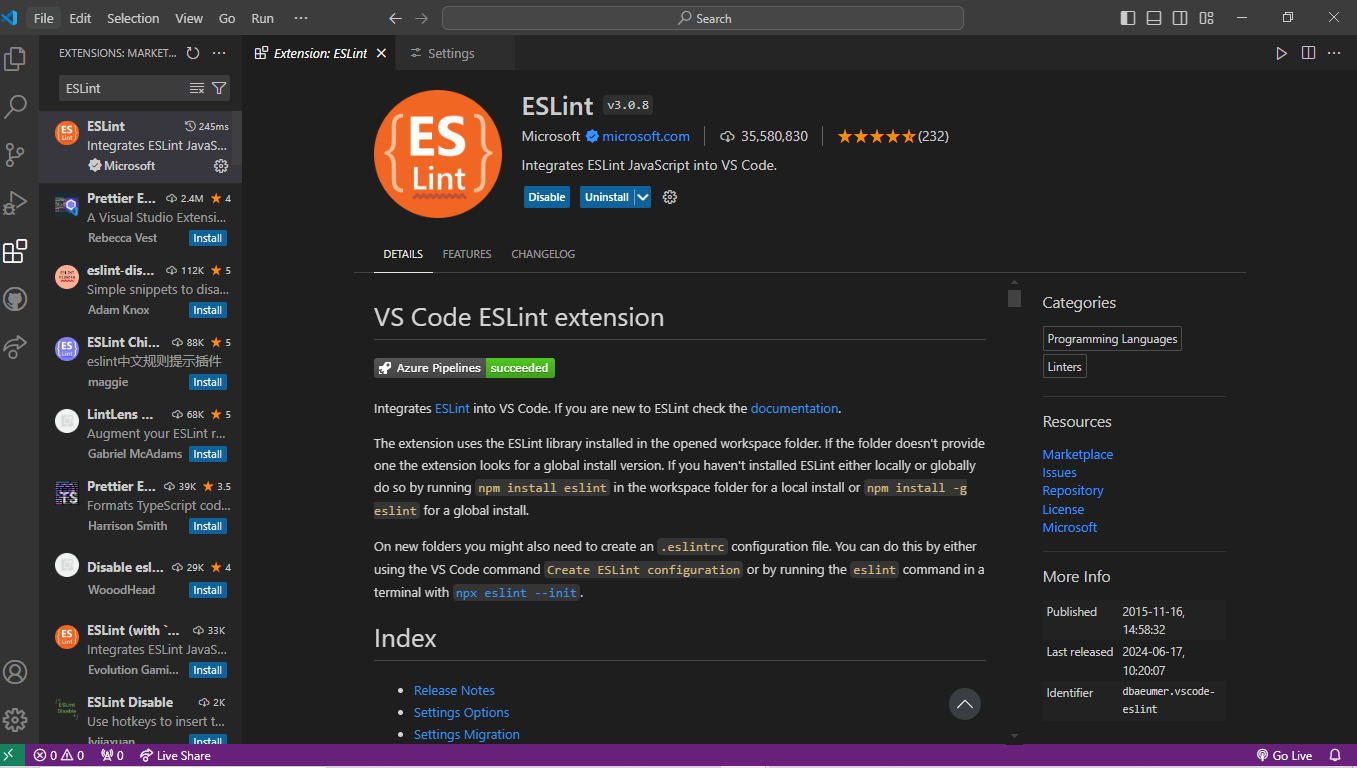
Installing Extensions:

Search for the extension you want.

Click on the "Install" button to install it.

Managing Extensions:

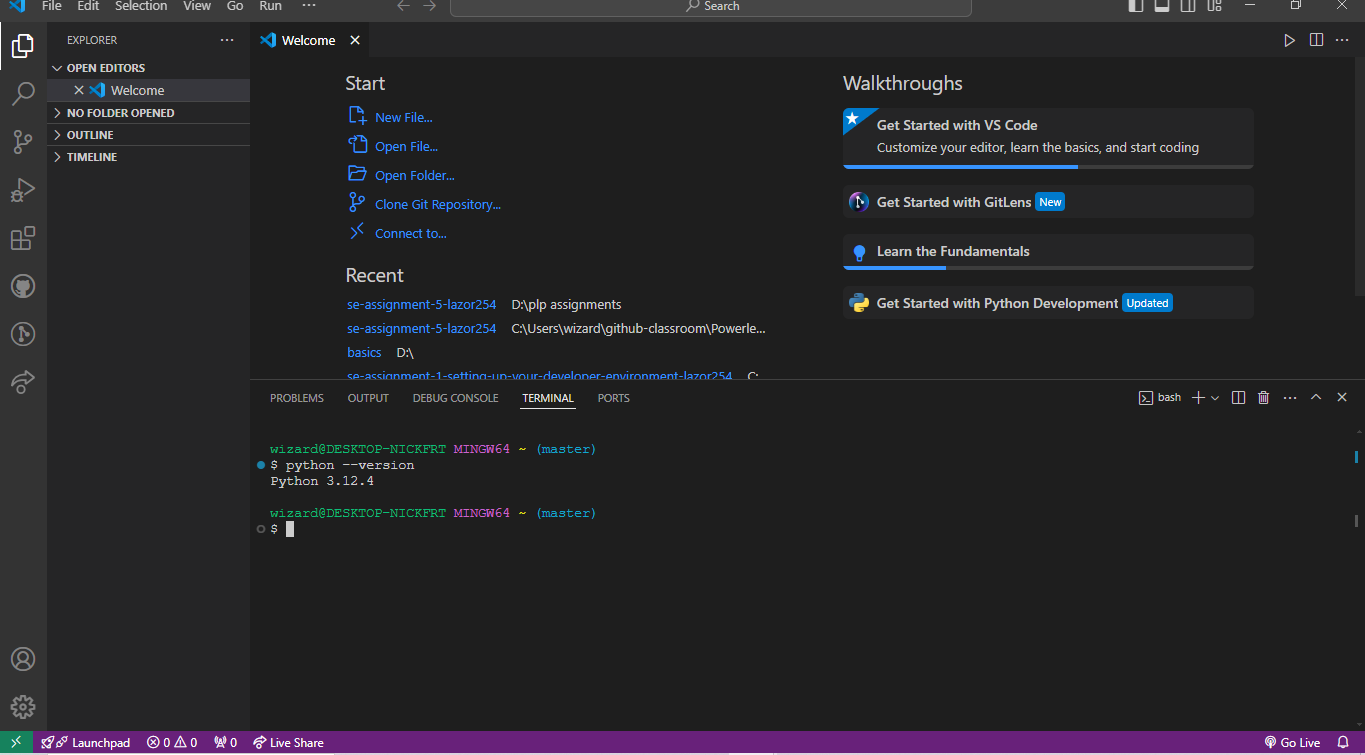
Use the Extensions view to enable, disable, or uninstall extensions.



step 10: Opening the Terminal:

Use the shortcut Ctrl+J or navigate to View > Terminal.

Using the Terminal:



You can run command-line tools directly within VS Code.

Supports multiple terminals and split views.

Advantages of Using the Integrated Terminal:

Seamless workflow: No need to switch between different windows.

Environment consistency: Uses the same settings and environment as your project.

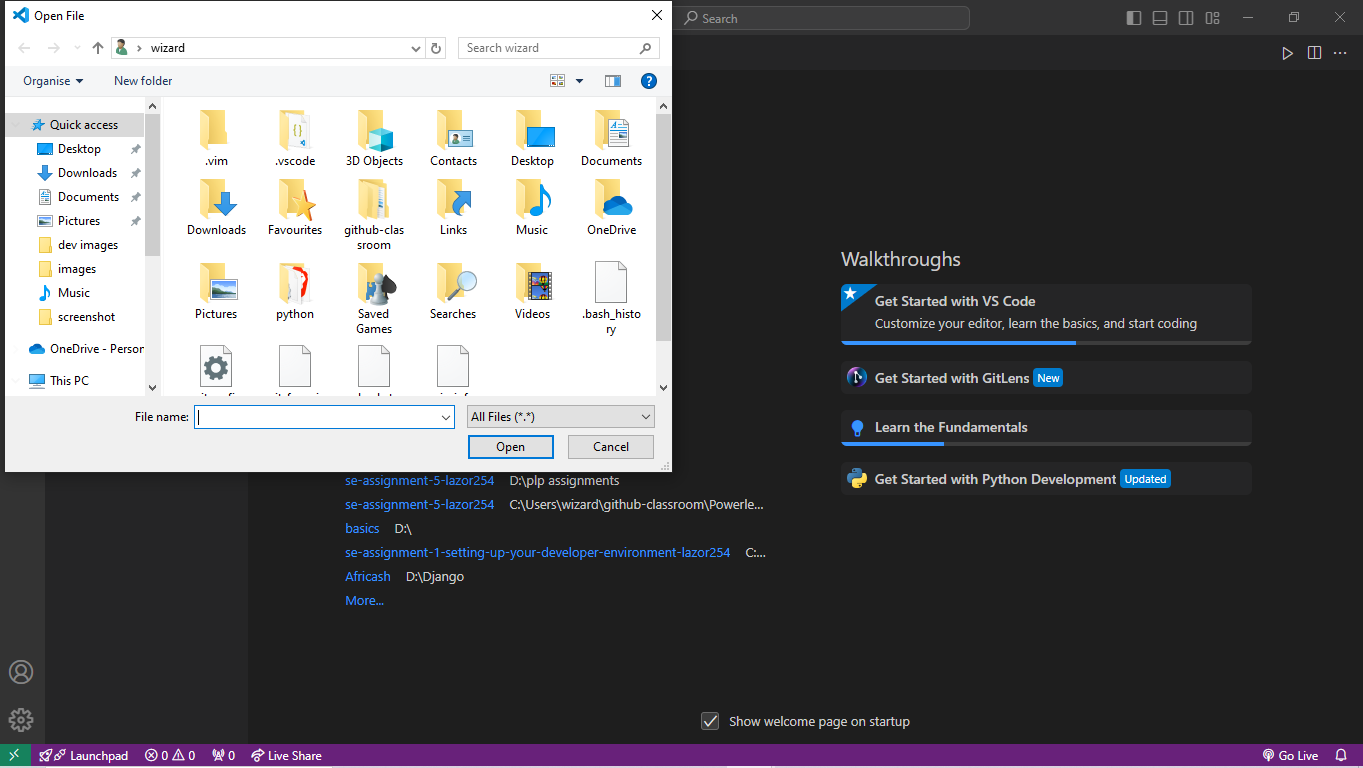
Integrated with VS Code: Supports tasks, debugging, and version control.

Also changes made can be visible one on one

Step 11: Creating, Opening, and Managing Files and Folders:

Creating Files and Folders:

Right-click in the Explorer view and select "New File" or "New Folder".



Use the File > New File or File > New Folder menu options.

Opening Files:

Double-click a file in the Explorer view.

Use the File > Open File menu option.

Managing Files and Folders:

Move files by dragging and dropping them within the Explorer.

Rename by right-clicking and selecting "Rename".

Navigating Between Files and Directories:

Step 12: Finding and Customizing Settings:

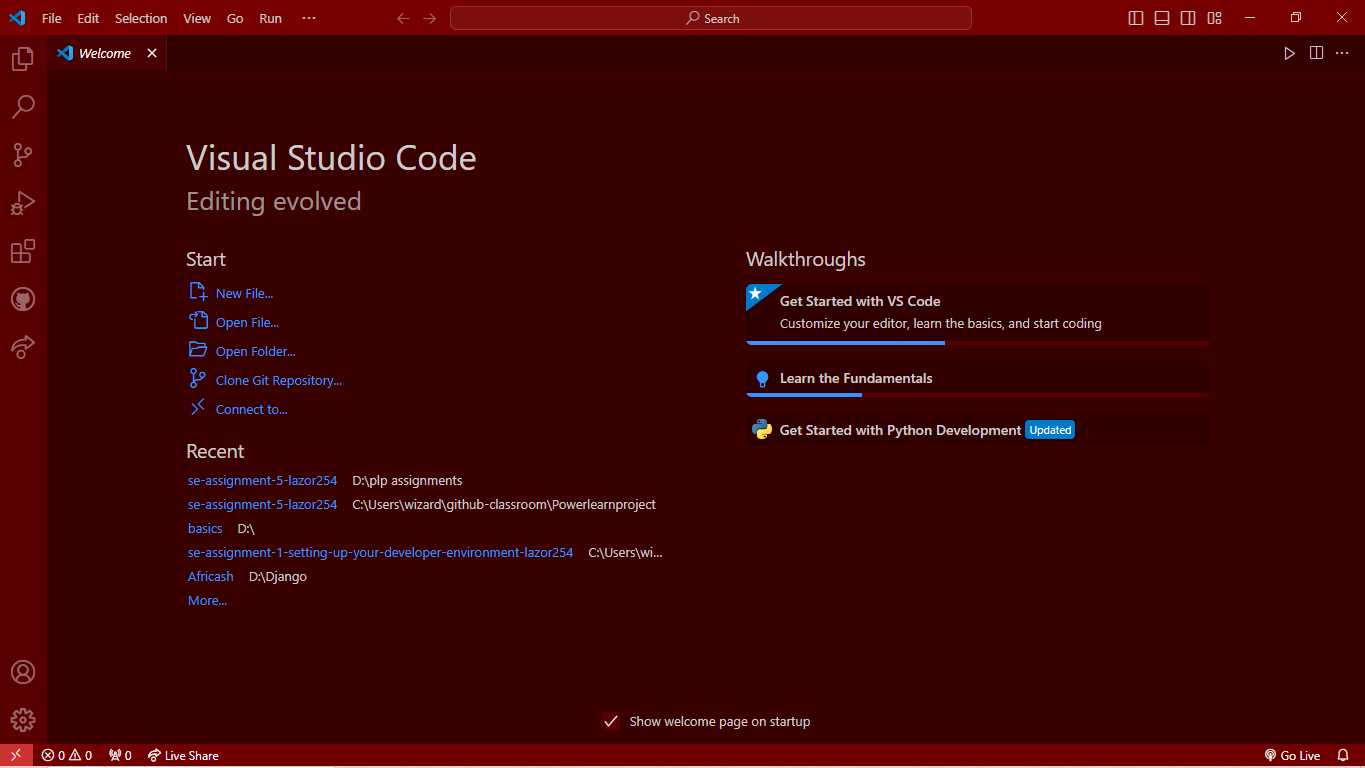
Opening Settings:

Navigate to File > Preferences > Settings or use Ctrl+,

Changing Theme:

Go to File > Preferences > Color Theme to select a new theme.

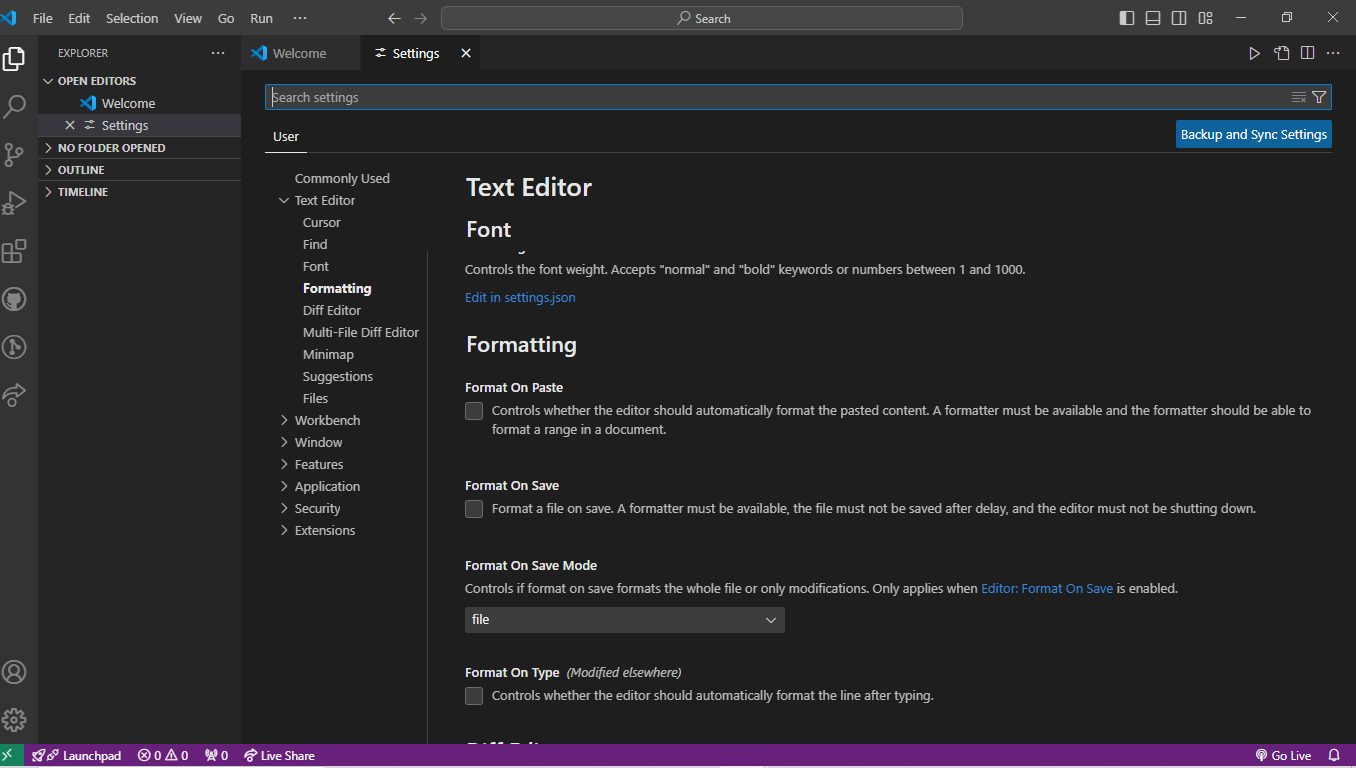
Changing Font Size:



Search for "Font Size" in the settings and adjust as needed.

Changing Keybindings:

Go to File > Preferences > Keyboard Shortcuts to customize keybindings.



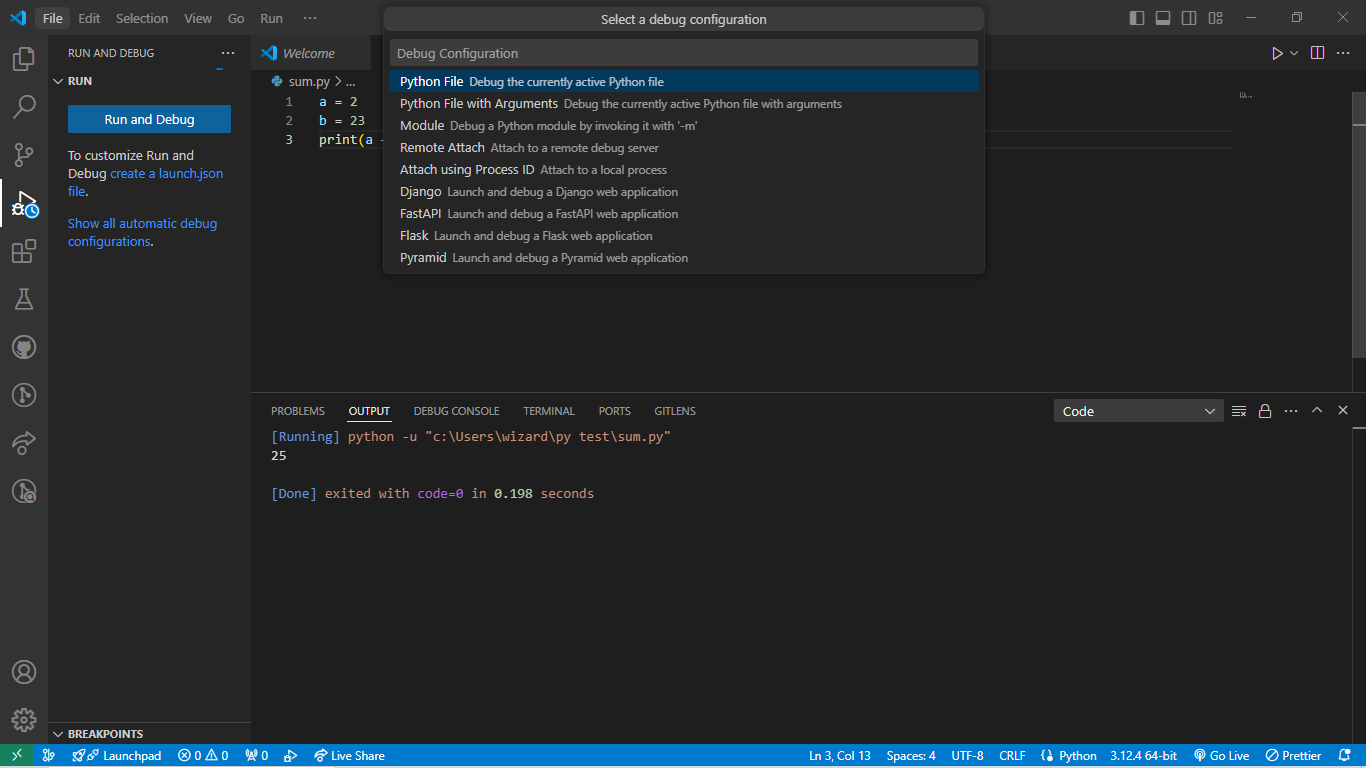
Step 13: Steps to Set Up and Start Debugging a Simple Program:

Open the Debug View:

Click on the Run and Debug icon in the Activity Bar or press Ctrl+Shift+D.

Configure the Debugger:

Click on python debugger to set up debugging configurations.



Key Debugging Features:

-Breakpoints and conditional breakpoints.

-Watch expressions and variable inspection.

-Call stack and loaded scripts view.

-Integrated terminal for running debug commands

Step 14: pushing files to GitHub

You need to have a GitHub account and git.

Inside your GitHub create a new repository name it the way you like. You can customize it as well with descriptions and making it public.

Then in your visual studio you choose the type of terminal you want either PowerShell, cmd or gitbash. But for this you will definitely want to use the gitbash terminal.

Open your gitbash terminal and navigate to the directory where your project is located.

Initialize a new Git repository by running the “git init” command

Add the files in your project to the staging area by running the “git add .”

command.

Commit the changes to the local repository by running the “git commit –m “my commit” command.

Push the changes to your GitHub repository by running the “git push –u "your branch”

And this will be successful you just need to refresh your GitHub repository to see the committed changes