Project – 2

Optimizing Development Workflow with VS Code: A Comprehensive Guide

Submitted By:

Nishigandha Patil

INDEX

INTRODUCTION TO VS CODE

Visual Studio Code (VS Code) is a free, open-source code editor developed by Microsoft. It runs on various platforms like Windows, macOS, and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages and runtimes such as C++, C#, Java, Python, PHP, Go, .NET etc.

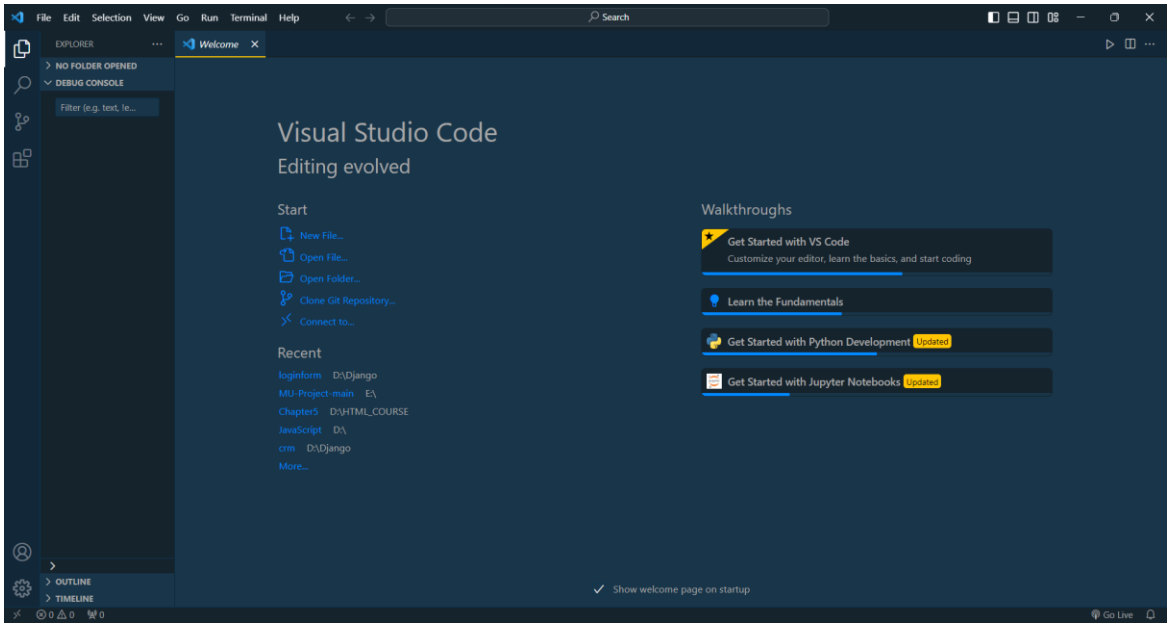
➢ Significance of VS Code:

1. **Language Support:** It supports multiple programming languages and frameworks.
2. **Extensions:** The rich ecosystem of extensions enhances functionality. Developers can add functionalities, themes, and tools through extensions.
3. **Performance:** Known for its speed and minimal resource usage, VS Code provides a smooth coding experience even with large projects, aiding developer productivity.
4. **Version Control:** Seamless integration with Git, allowing for version control operations such as committing, branching, merging, and viewing commit history within the editor.
5. **IntelliSense:** Provides intelligent code completion and suggestions.
6. **Customization:** Developers can personalize their workspace with different themes, settings, and extensions to match their preferences and workflow.
7. **Cross-Platform:** Works on Windows, macOS, and Linux.
8. **Community and Support:** A large and active community contributing extensions, providing support, and continuously improving the editor.

1. Installing VS Code:

Go to https://code.visualstudio.com/download > Download for "Windows" > Run the installer (VSCodeUserSetup-{version}.exe) > Install > Launch.

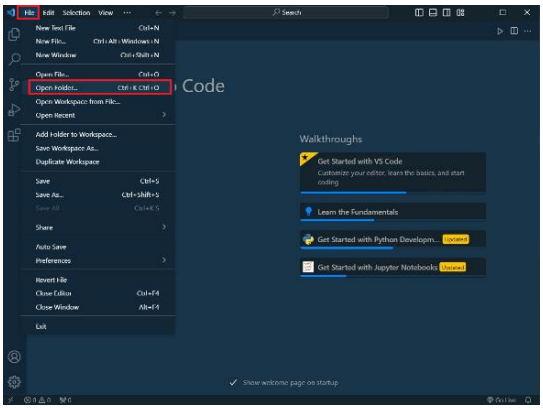
➢ Here's what you'll see when you open VS Code:

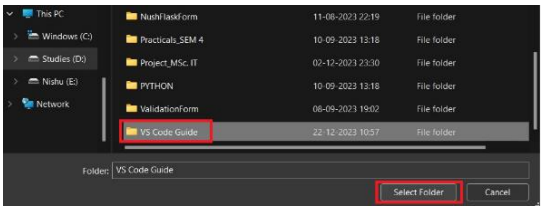


2. Creating and managing projects and files in VS Code:

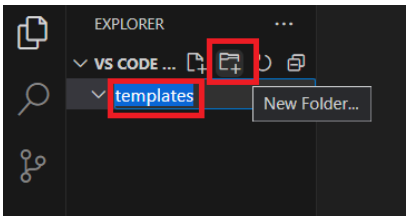
a. Creating a Project

- **Create Folder:** Open VS Code > File> Open Folder > Choose Folder > Select Folder.

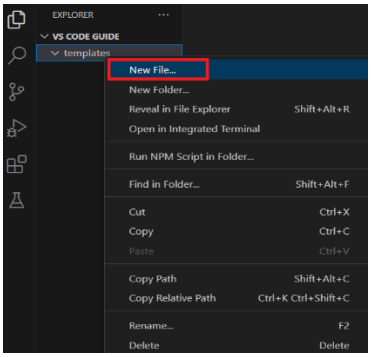


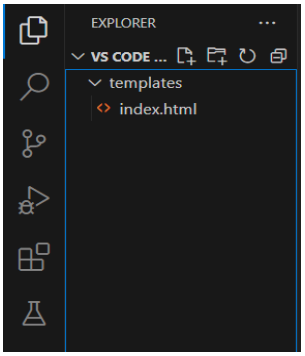


- Within this folder, you can create another folder.
  Click on "New Folder" icon > Add name of folder.

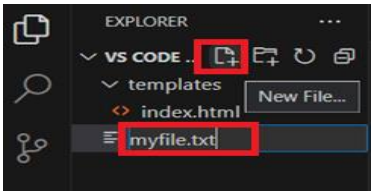


- **Add Files:** You can add or create files by:
  i. Right Clicking on folder > New File > Add name of file.





ii. Clicking on the "New File" icon.



b. Managing Files

- **File Navigation:** Use the File Explorer to navigate through files and folders. Double-clicking on a file will open it in the editor.
- **Tabs & Open Editors:** The tabs at the top of the editor window display open files. You can also access "Open Editors" from the sidebar to view all open files.

c. Working with Projects

- **Extensions:** Install relevant extensions for your project using the “Extensions” view to enhance functionality.
- **Git Integration:** VS Code integrates with Git. Initialize Git in your project folder to track changes.
- **Task Running:** Use the integrated terminal or tasks feature to run scripts or commands for your project.
- **Workspaces:** Create workspaces to manage multiple projects together.

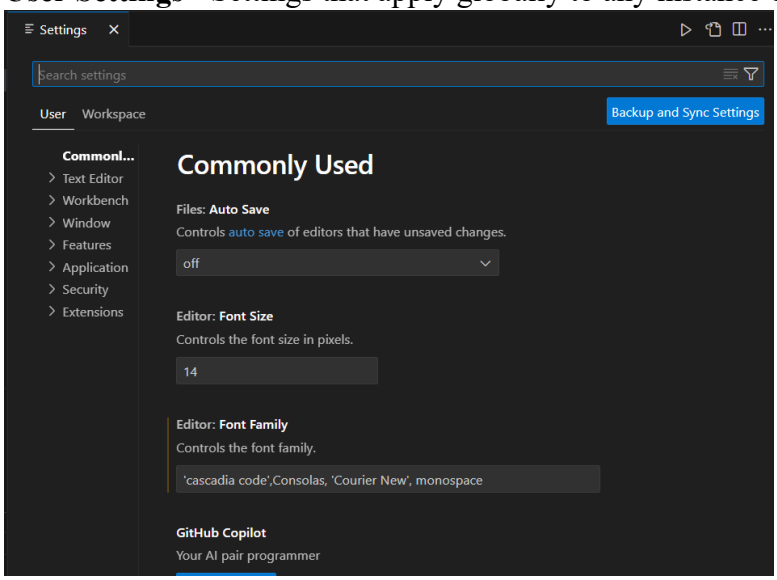
➢ **Useful Shortcuts:**

- **Ctrl + N:** Create a new file.
- **Ctrl + Shift + N:** Create a new window.
- **Ctrl + S:** Save current file.

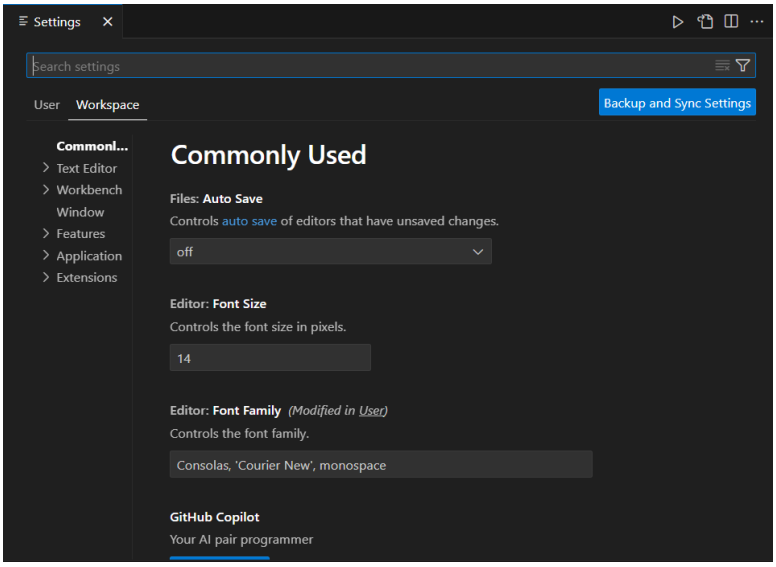
3. Settings:

You can configure VS Code through its various settings. Every part of VS Code's editor, user interface, and functional behavior has options to modify.
VS Code provides several different scopes for settings. When you open a workspace, you will see at least the following two scopes:

a) User Settings - Settings that apply globally to any instance of VS Code you open.



b) Workspace Settings - Settings stored inside your workspace and only apply when the workspace is opened.

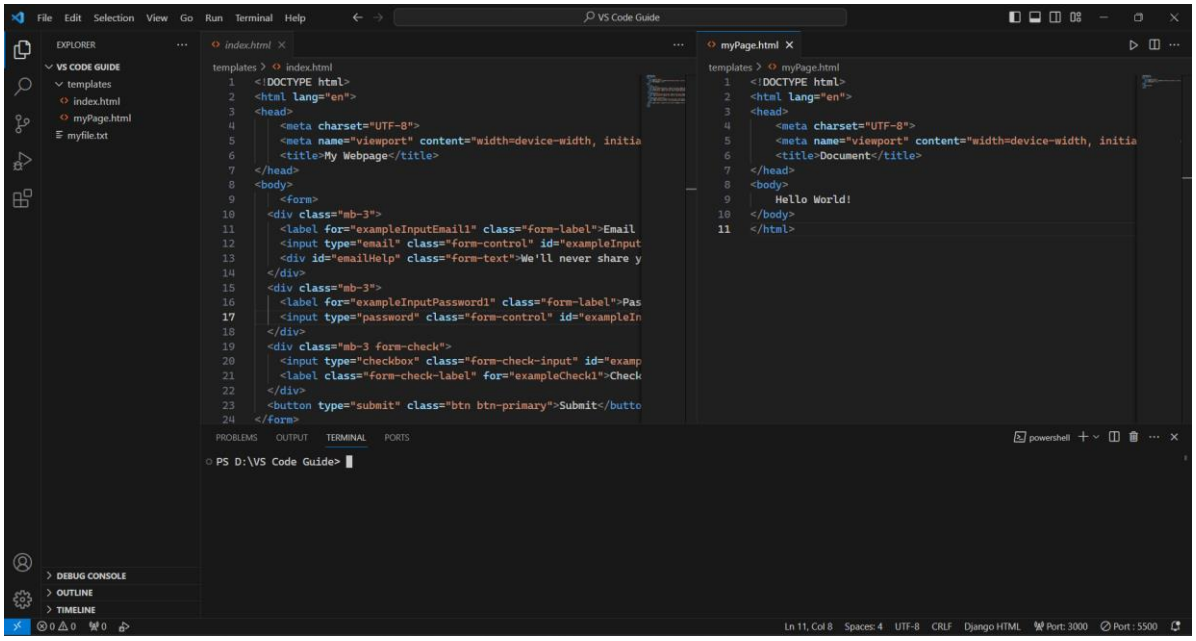


- **Types of Settings:**
  - **Editor Settings:** Control editor behavior, such as indentation, tab size, font, and line wrapping.
  - **File and Folder Settings:** Manage file associations, exclude certain files or folders from searches, and set preferences for file encoding.
  - **Extensions Settings:** Customize behavior and settings for installed extensions.
  - **Workspace Settings:** Store project-specific configurations, ensuring consistent settings for a particular workspace.
  - **Keybindings:** Customize or redefine keyboard shortcuts to match personal preferences or emulate other editors.
  - **Theme Settings:** Adjust theme-specific configurations, like color schemes or icon packs.

- **Configuration Methods:**
  - **GUI:** Access settings through the Settings UI (Ctrl+, or Cmd+,). Modify settings using checkboxes, dropdowns, and text fields in the graphical interface.
  - **JSON:** Edit settings directly in JSON format by accessing the settings.json file. These settings override GUI settings and provide more granular control.

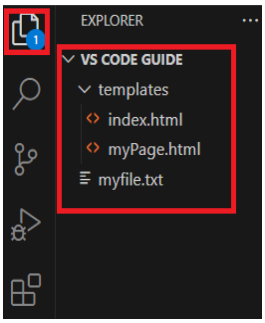
- **For More information:** https://code.visualstudio.com/docs/getstarted/settings

USER INTERFACE

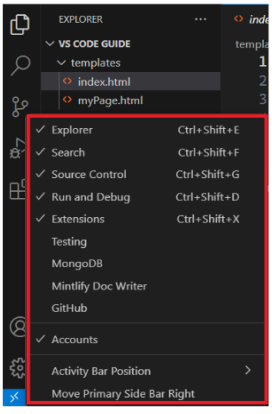


The UI is divided into five main areas:

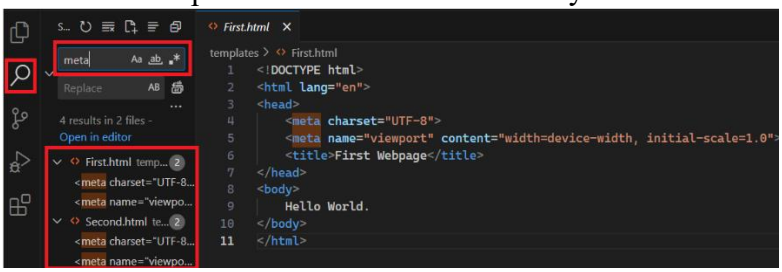
1. **Activity Bar** - Located on the far left-hand side, this lets you switch between views and gives you additional context-specific indicators, like the number of outgoing changes when Git is enabled.
    a. **Explorer:** Provides access to the file system. You can navigate files and folders, create new files, and manage your project structure.



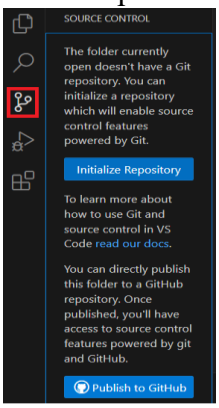
You can remove a View entirely (right-click the Activity Bar and uncheck any view that you don't want to show on acivity bar).



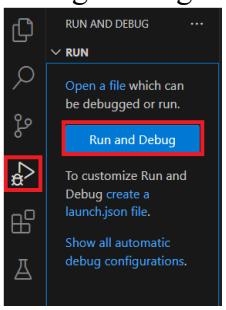
b. Search: Allows you to search within your project or workspace for specific text or files. It provides a search functionality across the entire codebase.



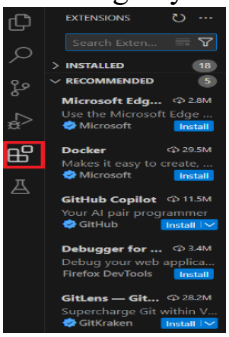
c. Source Control: This view integrates with version control systems like Git, which allows you to manage changes, commits, branches, and other source control operations.



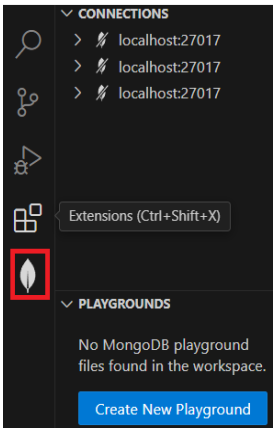
d. Run and Debug: Used to run and debug your code, set breakpoints, and manage configurations for debugging.



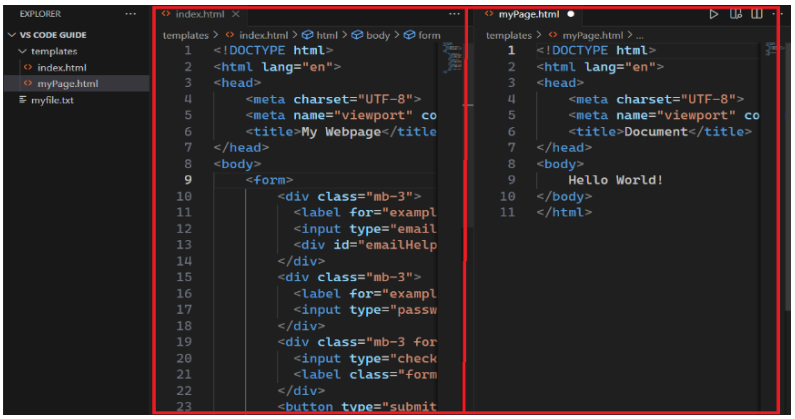
e. Extensions: You can search, explore, install, enable, and manage extensions according to your need that extend the functionality of VS Code.



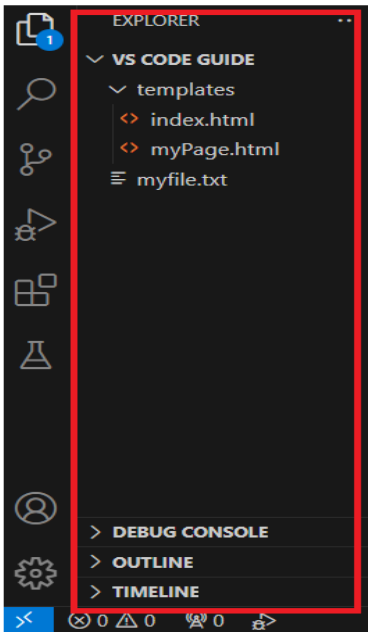
f. **Custom views** - Views contributed by extensions.
Eg. MongoDb Extension View.



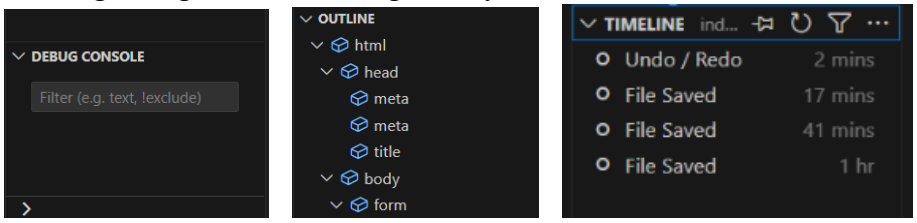
2. **Editor** - The main area to edit your files. You can open as many editors as you like side by side vertically and horizontally.



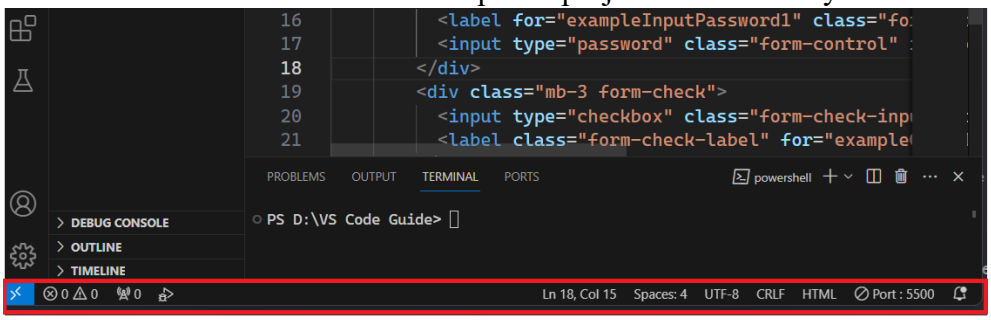
3. **Side Bar** - Contains different views like the Explorer to assist you while working on your project. You can access Secondary Side Bar by pressing **Ctrl+Alt+B** where you can drag and drop another views (eg. Debug Console).



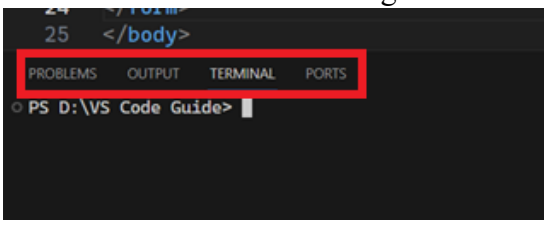
- **Debug Console:** Useful in Monitoring and troubleshooting during debugging sessions.
- **Outline:** Assists in navigating and understanding the structure of the code by presenting an overview of functions, classes, methods, and other significant elements within the code file.
- **Timeline:** The Timeline (if available via extensions) offer features related to tracking changes or visualizing history of edits in the file.



4. **Status Bar** - Information about the opened project and the files you edit.



5. **Panel** - An additional space for views below the editor region. By default, it houses output, debug information, errors and warnings, and an integrated terminal. Panel can also be moved to the left or right for more vertical space.



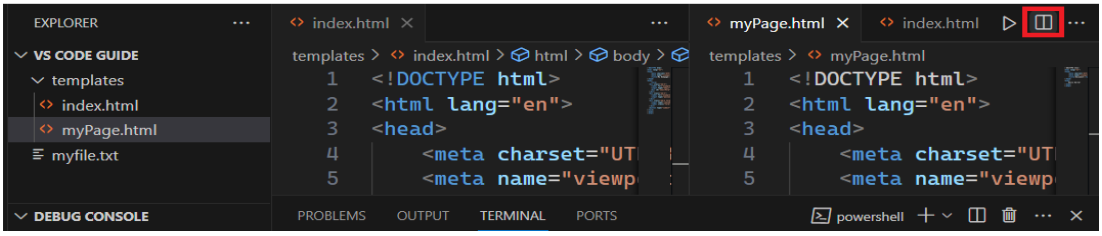
- **Problems:** Lists errors, warnings, and other issues detected in your code. It helps in identifying and fixing problems within your project.
- **Output:** Displays various output streams generated by tasks, extensions, or external processes. It can show build output, terminal output, and more.
- **Terminal:** VS Code has an integrated terminal panel that allows you to execute command-line tasks without leaving the editor.

➢ Features of Editor:

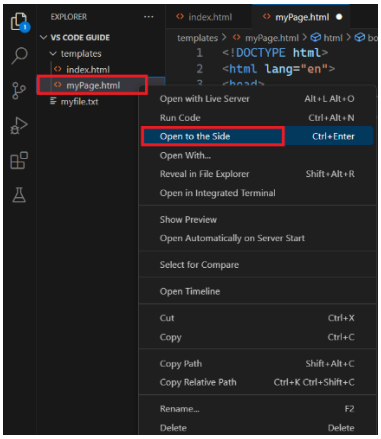
1. Side by side editing

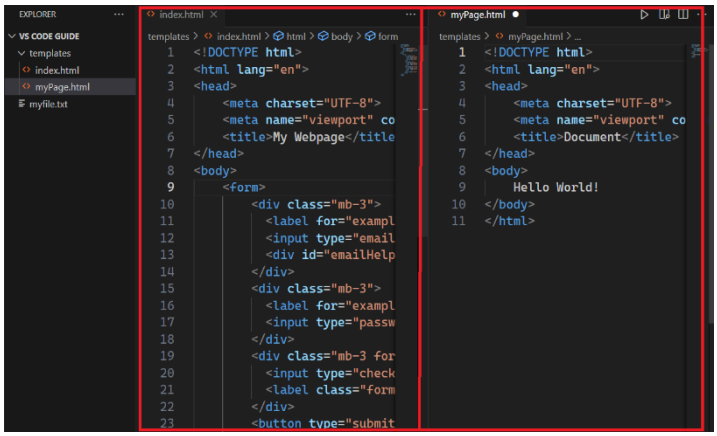
You can open as many editors as you like side by side vertically and horizontally.
Multiple ways of opening another editor to the side of the existing one:

- "Alt" + click on a file in the Explorer.
- "Ctrl+\" to split the active editor into two.
- Click the Split Editor button in the upper right of an editor.



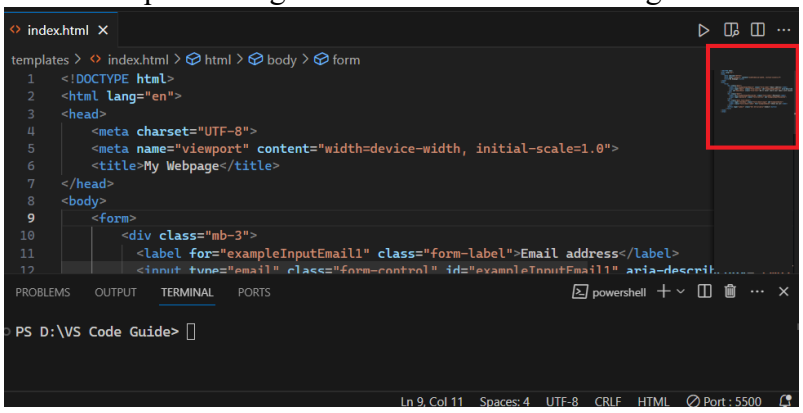
- Drag and drop a file to any side of the editor region.
- Right Click on file > Open to the Side (ctrl + Enter).





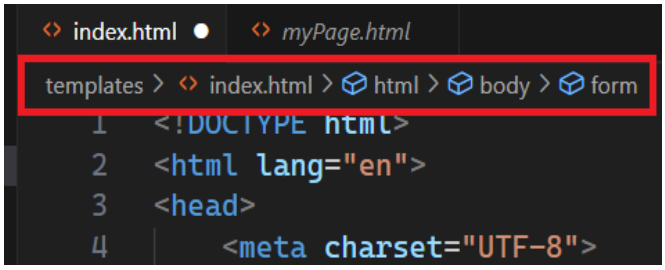
2. Minimap

Shown on right side of the editor. A Minimap gives you a overview of your code, which is useful for quick navigation and code understanding.



3. Breadcrumbs

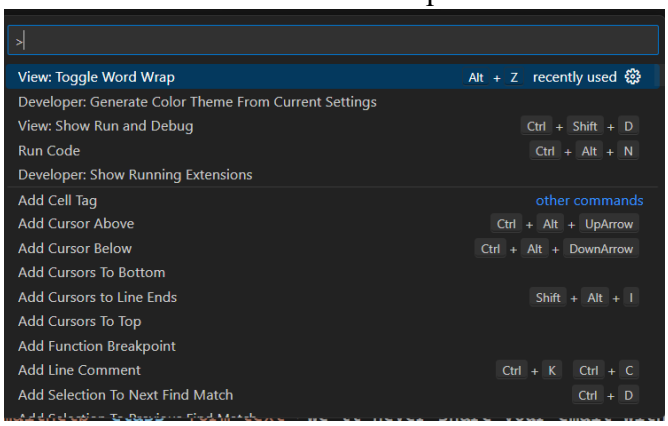
The editor has a navigation bar above its contents called Breadcrumbs. It shows the current location and allows you to quickly navigate between folders, files, and symbols.



4. Command Palette

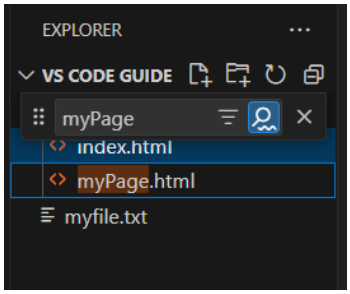
Shortcut: "Ctrl+Shift+P"

Here, you have access to all of the functionality of VS Code, including keyboard shortcuts for the most common operations.



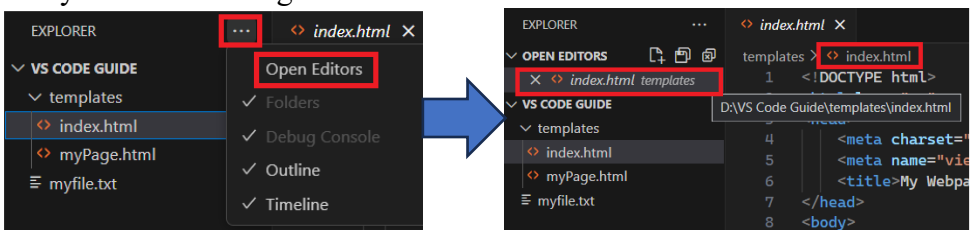
5. Advanced tree navigation

In File Explorer, use "Ctrl+F" to open Find control and type the file name you want to match. If match found it will be highlighted.



6. Open Editors

Provides a list of active files or previews. These are files you previously opened in VS Code that you were working on.



BASIC CODING AND CUSTOMIZATION

1. Writing and editing code

VS Code's advanced editor features can significantly enhance your coding experience. You can write code in multiple programming languages using VS Code and can edit the code with VS Code's advanced Features.

a. Keyboard Shortcuts for editing:

Sr. No	Action	Windows	MacOS
1	Move a line up or down	Alt + ↑ or ↓	Option + ↑ or ↓
2	Duplicate a line or selection	Shift + Alt + ↑ or ↓	Shift + Option + ↑ or ↓
3	Select the next occurrence	Ctrl + D	Ctrl + D
4	Navigate between occurrences	Ctrl + K + D	Shift + Command + G and Command + G

For more shortcuts:
Visit: https://code.visualstudio.com/shortcuts/keyboard-shortcuts-windows.pdf

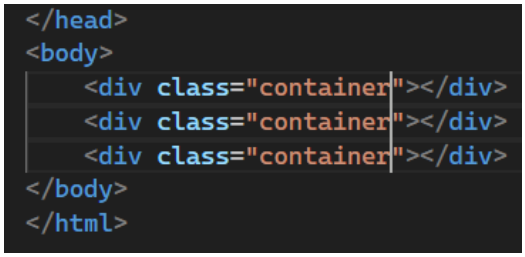
b. Multiple selections (multi-cursor):

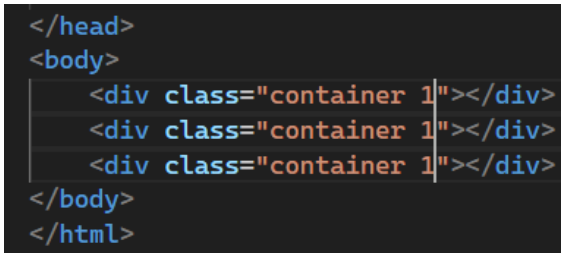
VS Code supports multiple cursors for fast simultaneous edits. Each cursor operates independently.

- To add secondary cursors: **Alt+Click**.

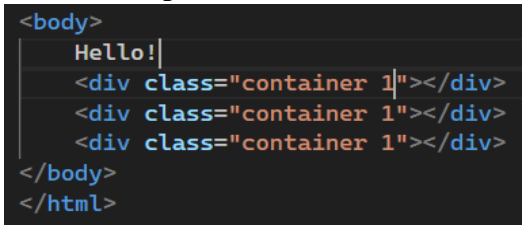
- To add more cursors:

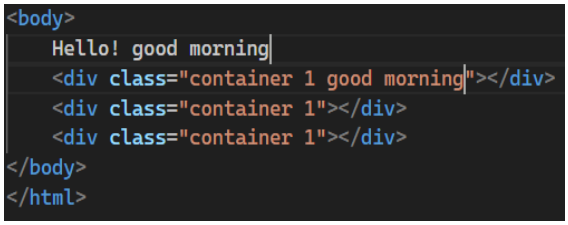
Ctrl+Alt+Down that insert cursors below.



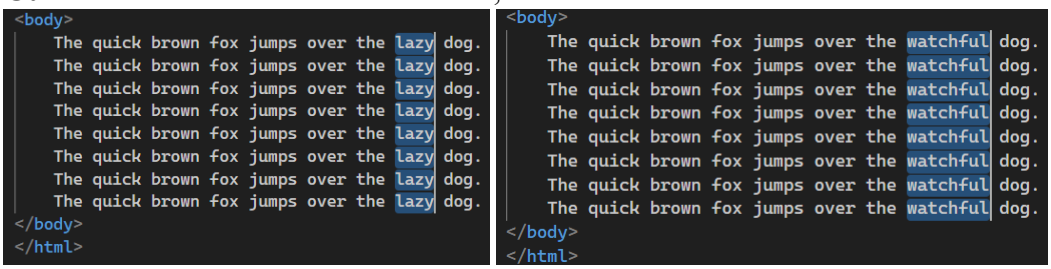


Ctrl+Alt+Up that insert cursors above.





- **Ctrl+D** selects the word at the cursor, or the next occurrence of the current selection.

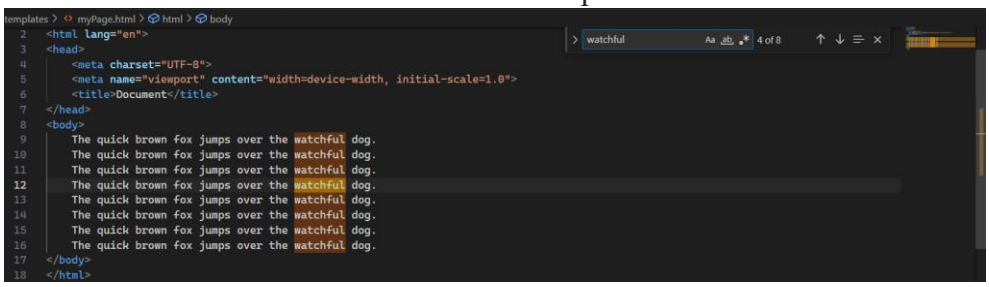


c. Shrink/expand selection:

Quickly shrink or expand the current selection.
Press: **Shift+Alt+Right (expand)** and **Shift+Alt+Left (shrink)**

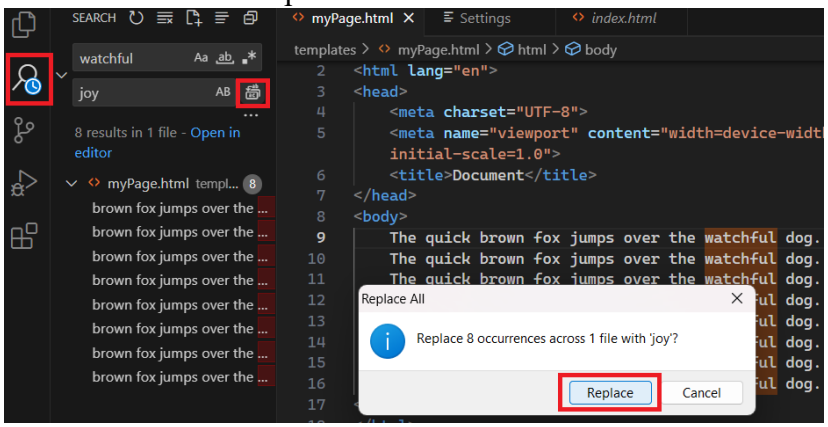
d. Find:

VS Code allows you to quickly find text in the currently opened file.
Press **Ctrl+F** to open the Find Widget in the editor.
If there are more than one matched, you can press “Enter” and “Shift+Enter” to navigate to next or previous result.
Press Ctrl+Enter to insert a new line in the input box.

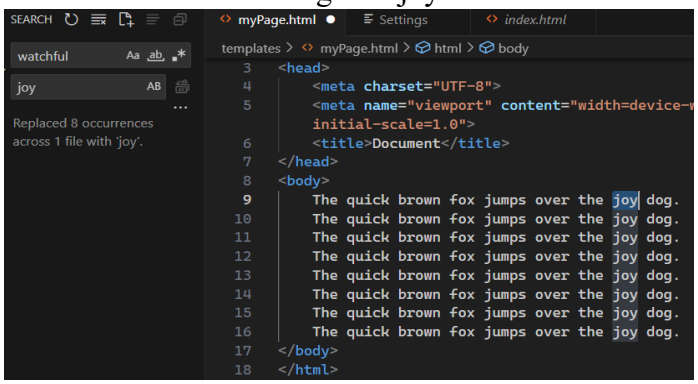


e. Search and Replace:

You can search and replace across files.

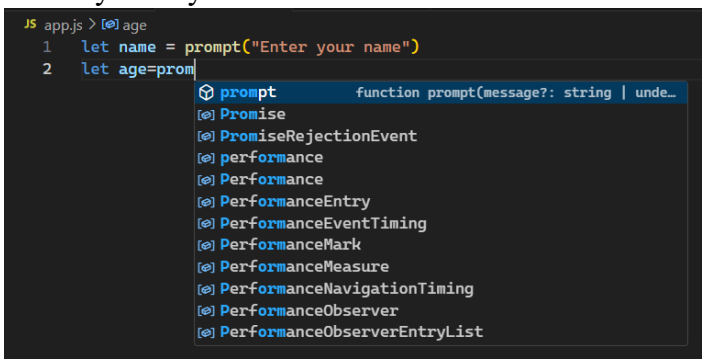


The word watchful changes to joy



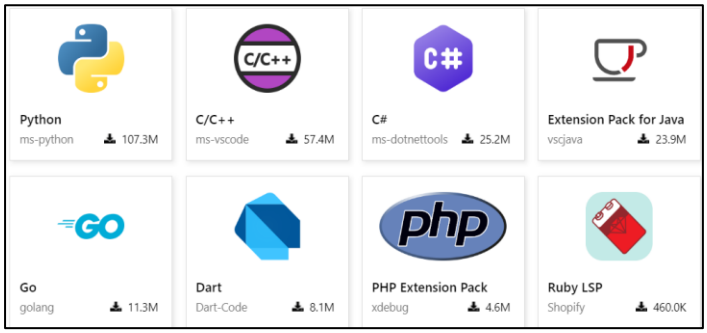
f. IntelliSense:

A language service provides intelligent code completions based on language semantics and an analysis of your source code.



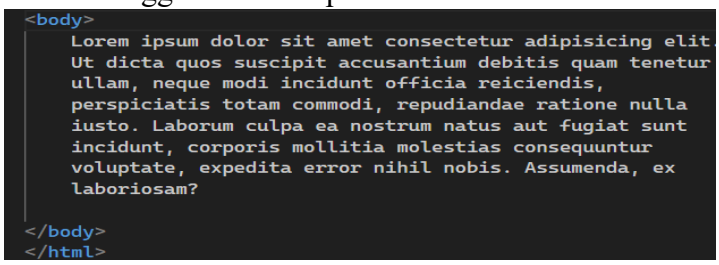
VS Code supports word-based completions for any programming language but can also be configured to have richer IntelliSense by installing a language extension.

Below are the most popular language extensions in the Marketplace. Download the extensions using Extension View.



g. Word Wrap:

You can toggle word wrap for the VS Code session with **Alt+Z.**



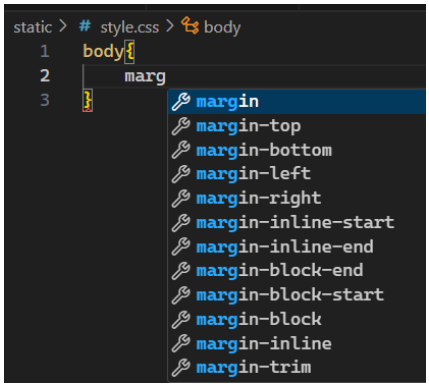
2. Emmet:

Emmet is a powerful toolkit for web developers that significantly speeds up HTML and CSS workflow.

- **Emmet for HTML:**

You can use most of the Emmet actions with multi-cursors as well.

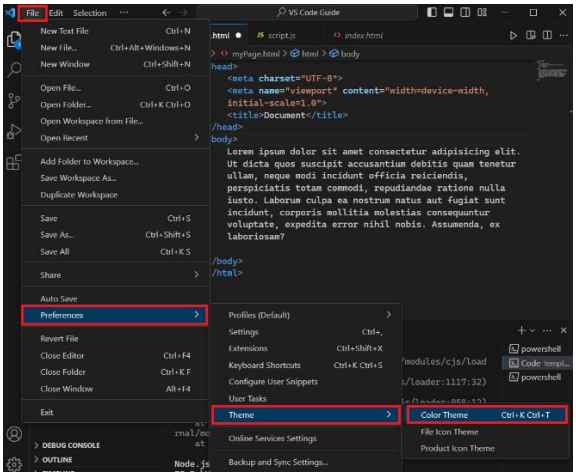
- **Emmet for CSS:**

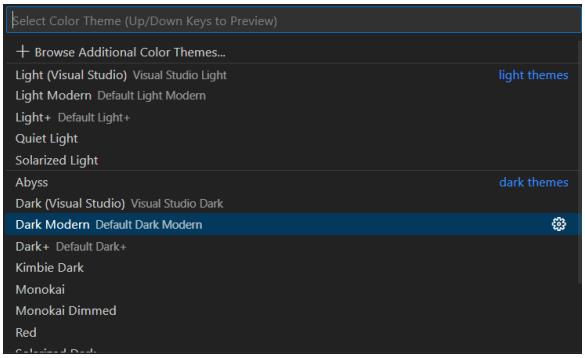


3. Customization:

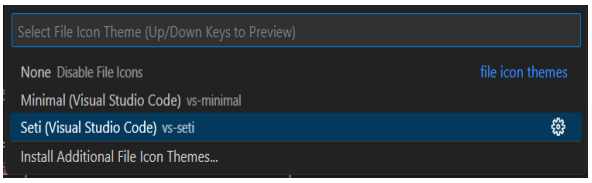
Users can customize existing themes or create their own by modifying colors, fonts, sizes, and other visual elements using settings or theme-specific configuration files.

a. **Themes:** Customize the appearance of the editor with different themes available in the Marketplace. Each theme has its own syntax highlighting styles.
   - **Color Themes:** These themes define the color scheme used throughout the editor. They affect syntax highlighting, text colors, background colors, and more.
     Click File > Preferences > Theme > Color Theme.

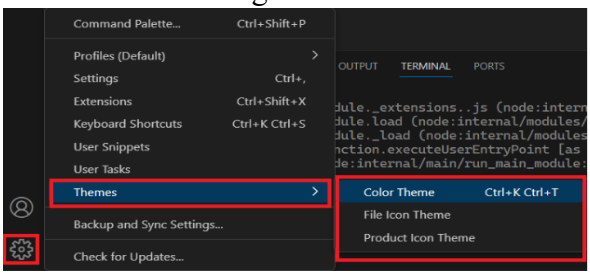




- **File Icon Themes:** Icon themes change the icons used in the Explorer panel to represent files, folders, and various elements in your project.
  Click File > Preferences > Theme > File Icon Theme.

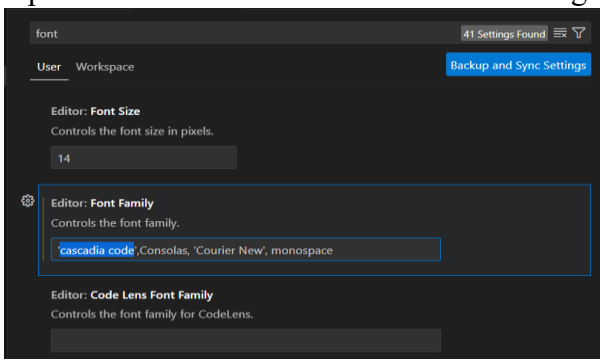


OR Click on Settings > Themes > Color Theme/File Icon Theme



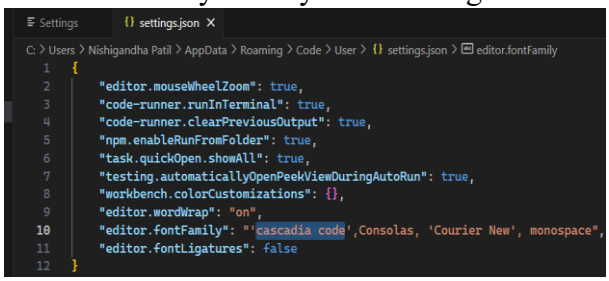
b. Fonts:

Open VS Code > File > Preferences > Settings > Search “Font” > Edit Font Family > Save.



JSON Settings:

You can directly modify these settings in the **settings.json** file by adding lines like:

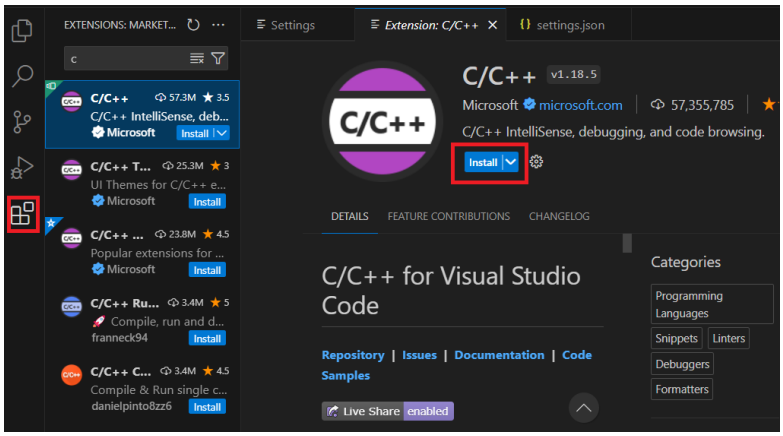


EXTENSIONS AND PRODUCTIVITY TOOLS

1. Extensions

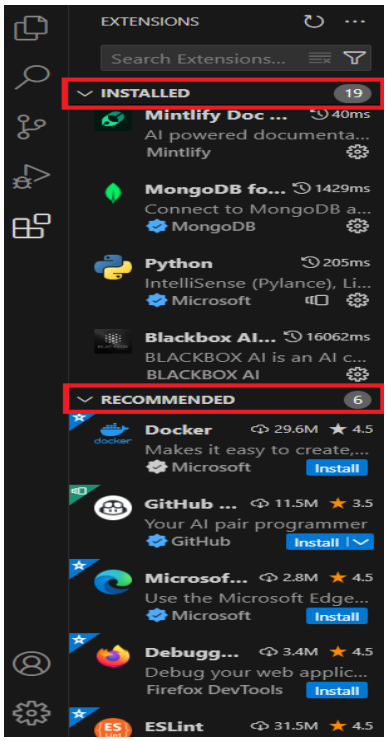
Extensions are like plugins or modules that offer additional features, language support, tools, themes, and integrations with external services. You can browse and install extensions from within VS Code by clicking on the Extension View from Activity Bar.

Click on Extension view > search for Extension > Install.



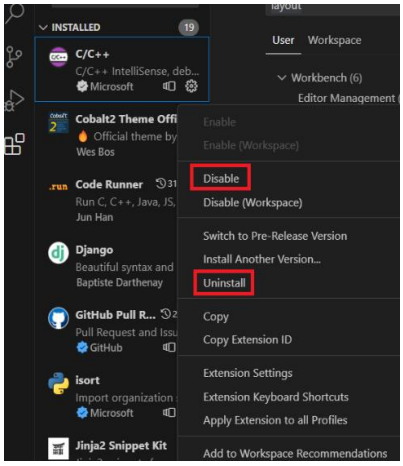
a. Managing extensions

By default, the Extensions view will show the “Installed extensions” and “recommended extensions”.



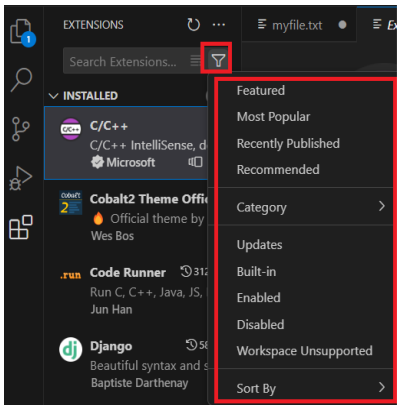
b. Disable/Uninstall extension:

VS Code makes it easy to manage your extensions. You can install, disable, update, and uninstall extensions through the Extensions view



c. Extensions view filters

The Extensions view search box supports filters to help you find and manage extensions.



➢ **Popular extensions used for different programming languages in VS Code:**

a. Python:

- **Python:** Official extension for Python development.
- **Django:** Adds support for Django framework.
- **Jupyter:** Enables working with Jupyter Notebooks.
- **Pylance:** Language server for Python with enhanced features.
- **autoDocstring:** Helps generate docstrings for Python functions.

b. HTML/CSS:

- **HTML CSS Support:** Adds autocompletion for HTML and CSS.
- **Live Server:** Launches a development server for live previews.
- **CSS Peek:** Allows peeking into CSS definitions.

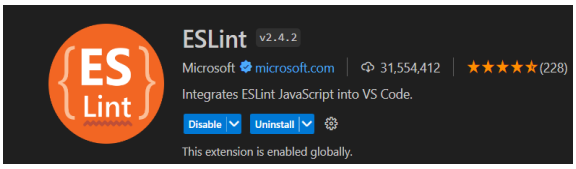
c. JavaScript/TypeScript:

- **ESLint:** Linter for JavaScript and TypeScript.
- **Prettier:** Code formatter for consistent code styling.
- **Debugger for Chrome:** Debug JavaScript in Chrome browser.
- **npm Intellisense:** Autocompletes npm modules in import statements.
- **Jest:** Support for testing with Jest.

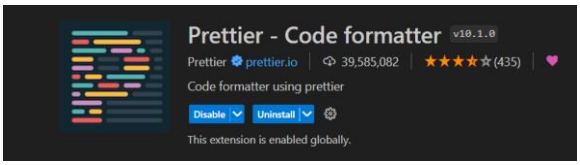
d. **Java:**
   - **Java Extension Pack:** Collection of extensions for Java development.
   - **Spring Boot Extension Pack:** Support for Spring Boot development.
   - **CheckStyle for Java:** Enforces coding standards using CheckStyle.
   - **Debugger for Java:** Allows debugging Java code.
   - **Maven for Java:** Maven integration for managing Java projects.

2. Productivity Tools:

a. **Linters and Code Formatters:** Tools that ensure code quality by identifying errors, enforcing coding standards, and formatting code automatically.
Extensions > ESLint

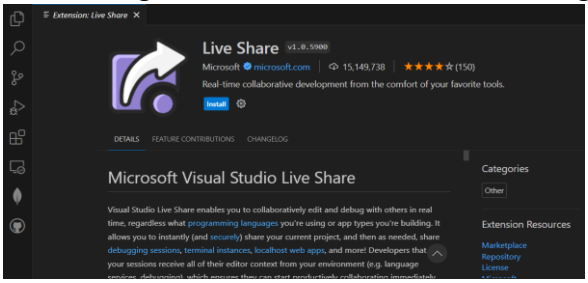


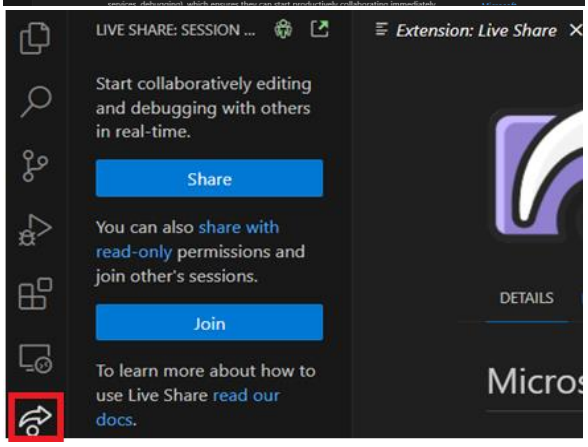
Extensions > Prettier



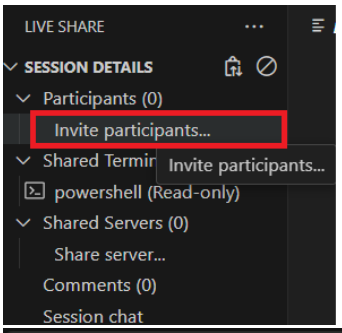
b. **Task Runners:** Tools that automate repetitive tasks like building, testing, and deploying code.

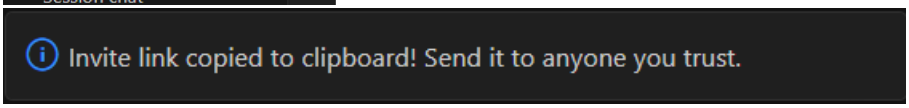
c. **Live Share and Collaboration Tools:** Features allowing real-time collaboration, live sharing of code, and simultaneous editing among team members.





To share: Click on **Share**> Invite participants > Share the Invite link.



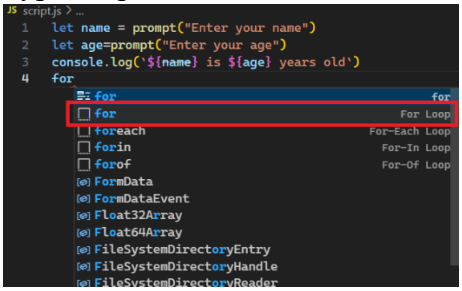


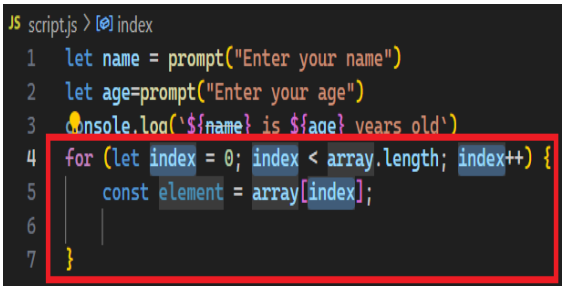
To Join: Click on **Join** > Paste the Invite Link.

d. Snippet Libraries: Collections of code snippets that help in quickly inserting commonly used code patterns. Snippets files are written in JSON, support C-style comments, and can define an unlimited number of snippets

- **Built-in snippets**

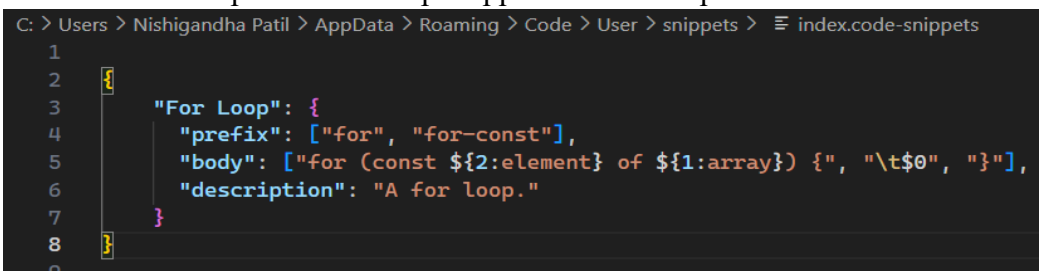
VS Code has built-in snippets for a number of languages such as: JavaScript, TypeScript, Markdown, and PHP.





- **Create your own snippets**

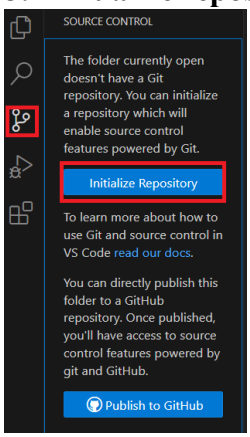
You can easily define your own snippets without any extension.
go to File > Preferences > Configure User Snippets > select the language (by language identifier) for which the snippets should appear, or the New Global Snippets file option if they should appear for all languages.
Below is an example of a for loop snippet for JavaScript



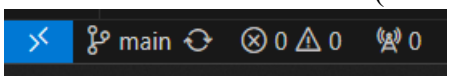
VERSION CONTROL INTEGRATION

In VS Code, version control is integrated, primarily through the use of Git, a widely-used distributed version control system.

- **To use version control in VS Code:**
  1. **Install Git:** visit https://git-scm.com/downloads
  2. **Open Folder:** File > Open Folder
  3. **Initialize repository:** Source Control > Initialize repository

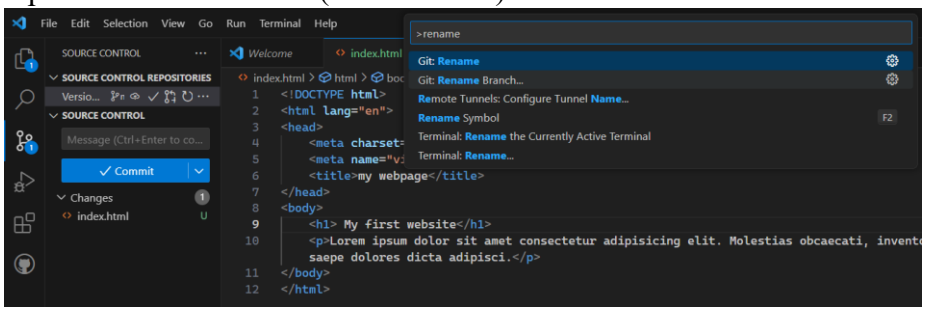


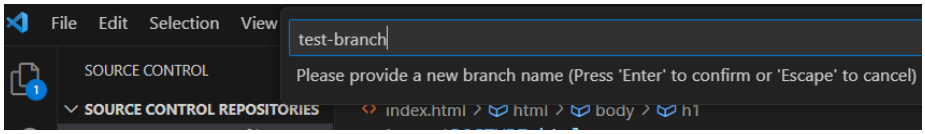
"main" is the default branch (lower left)

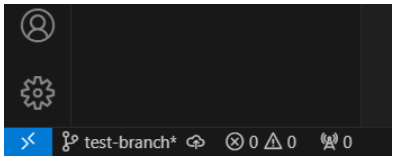


4. **Rename a branch:**

Open Command Palette (Ctrl+Shift+P)> Search for "Git: Rename Branch"







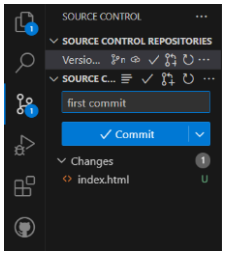
you can again rename it to "main".

5. **File version control status:**

U - Untracked file (not added to repository)
A - Added file
M - Modified file

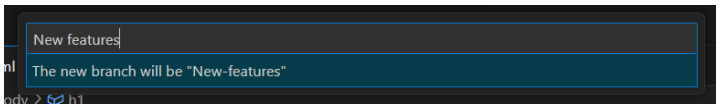
6. Commit file:

Give name of Commit > Click on “Commit” ☑.



7. Create a branch:

Open Command Palette (Ctrl+Shift+P) > search for “Git: Create Branch” and type name of the branch.



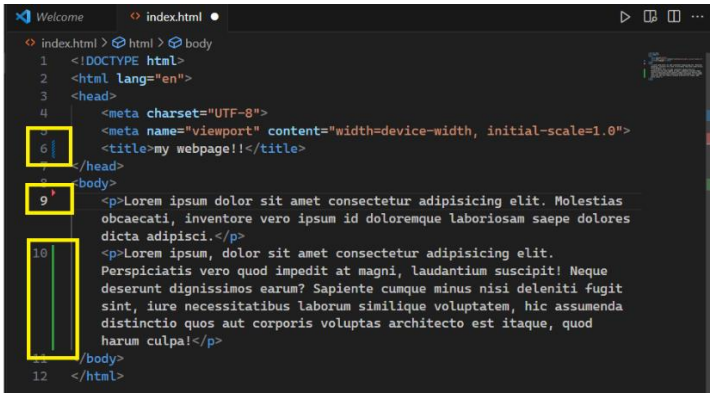
8. Gutter indicators

If you open a folder that is a Git repository and begin making changes, VS Code will add useful annotations to the gutter and to the overview ruler.

A **red** triangle: indicates where lines have been deleted

A **green** bar: indicates new added lines

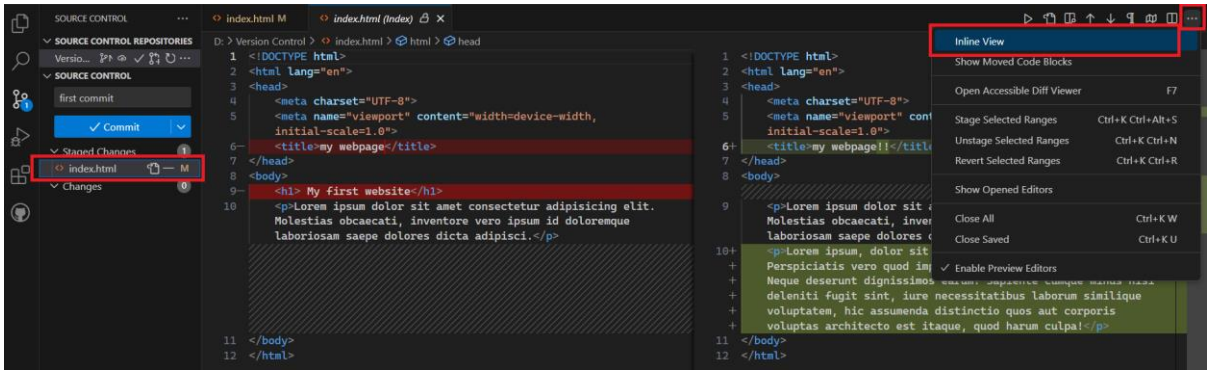
A **blue** bar: indicates modified lines



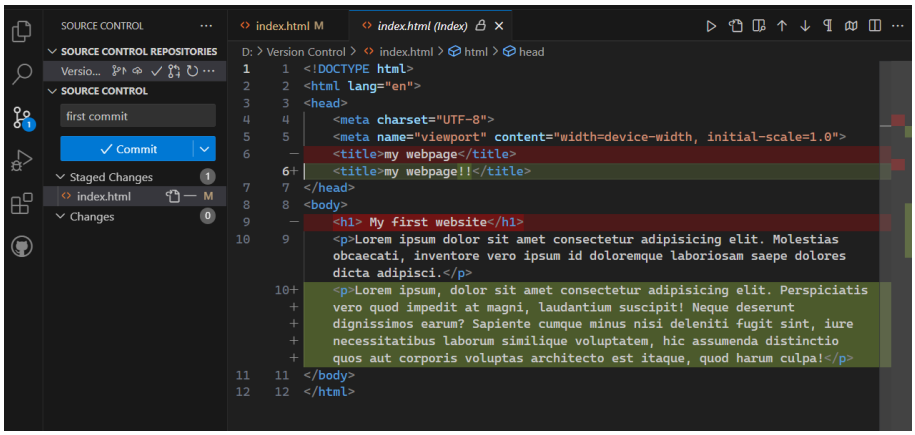
9. Diff editor:

This action opens the Diff Editor view, showing the two files side by side with their differences highlighted. Additions, deletions, and modifications will be visually distinct.

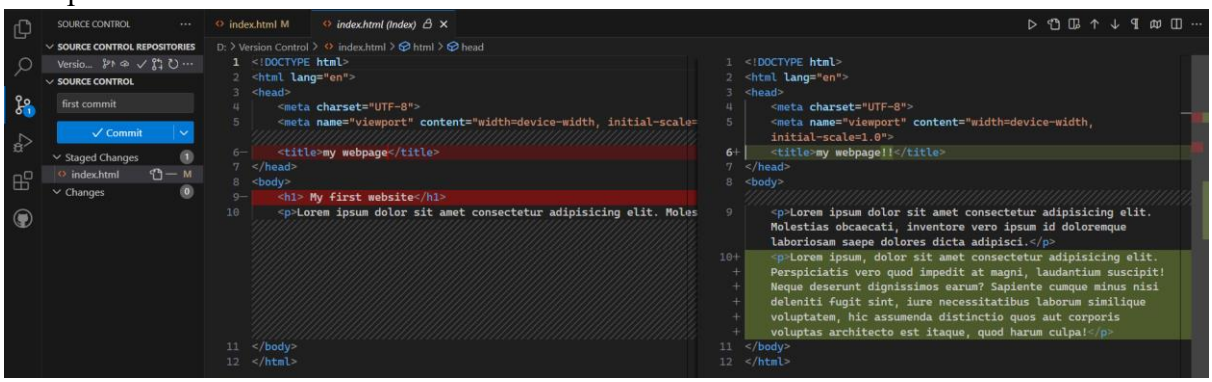
Click on the file > Inline View button



Inline View:

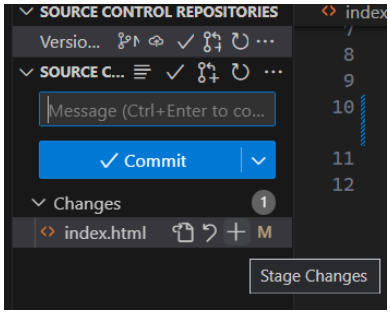


Comparison between initial code and edited code:



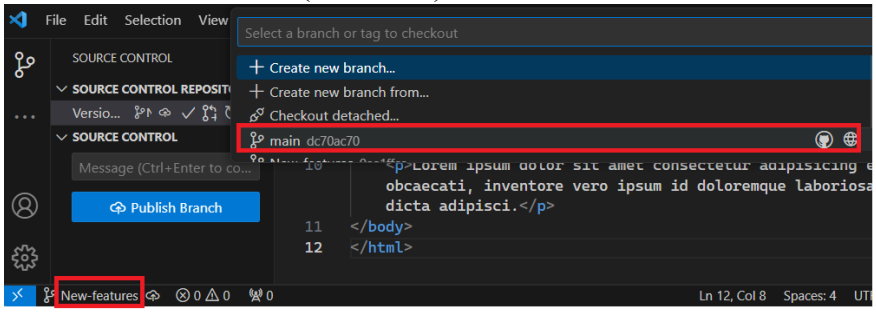
10. Stage changes:

Click Stage Changes ➕ button



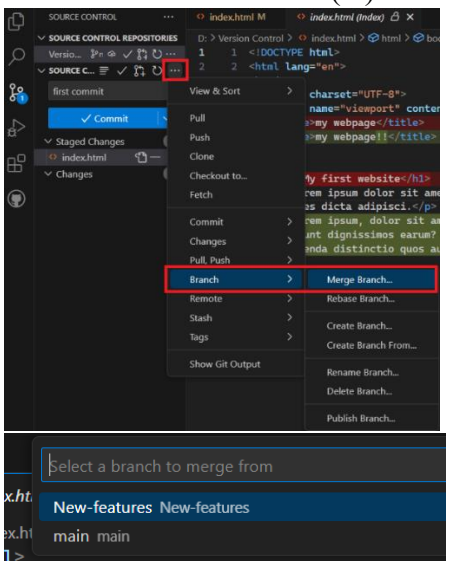
11. Switch branches:

Status bar branch item (lower left)

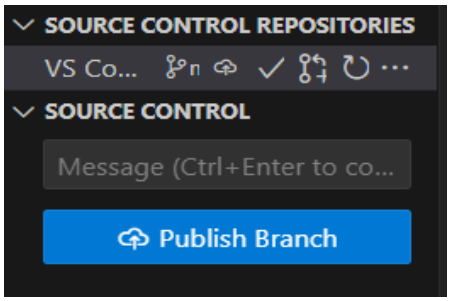


12. Merge branch:

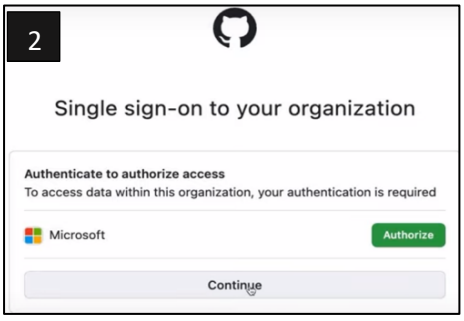
Views and More Actions (...) > Branch > Merge Branch

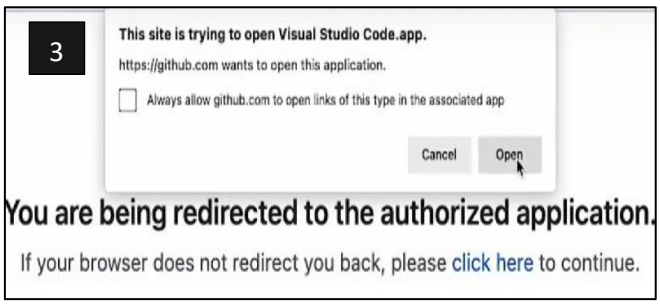


13. Publish branch to GitHub:

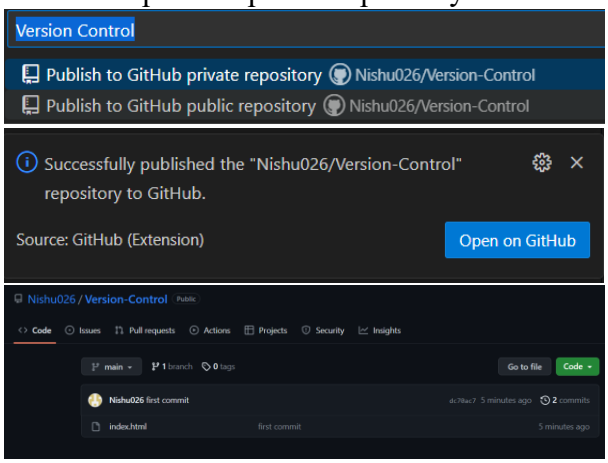


- **Authentication Process will start:**



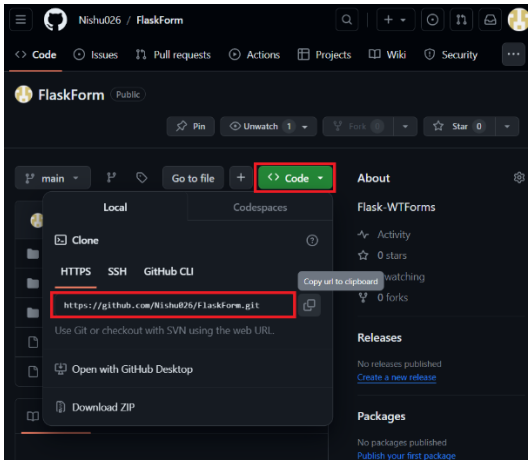


Publish to private/ public repository:

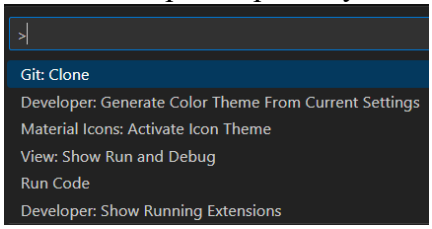


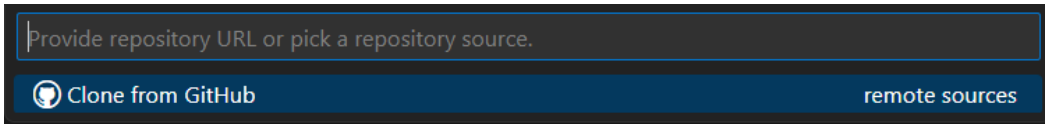
14. Clone repository:

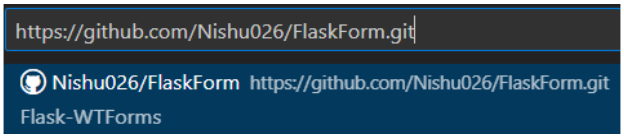
GitHub > Open repository >Copy URL

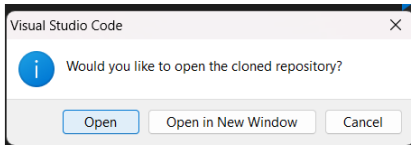


Open VS Code> Open Command Pallet > search for “Git: Clone” > Clone from GitHub
Paste the copied repository URL> Select Folder > Open.

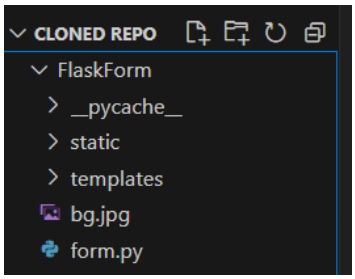






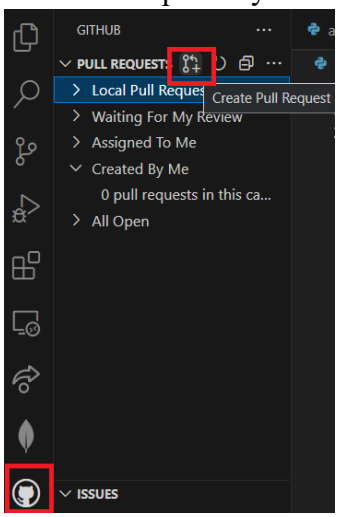


Cloned Repository in VS Code:

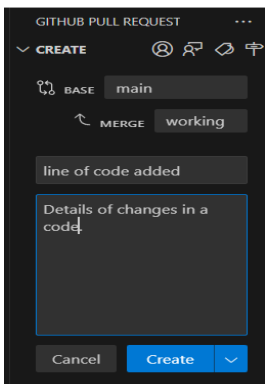


15. Pull request:

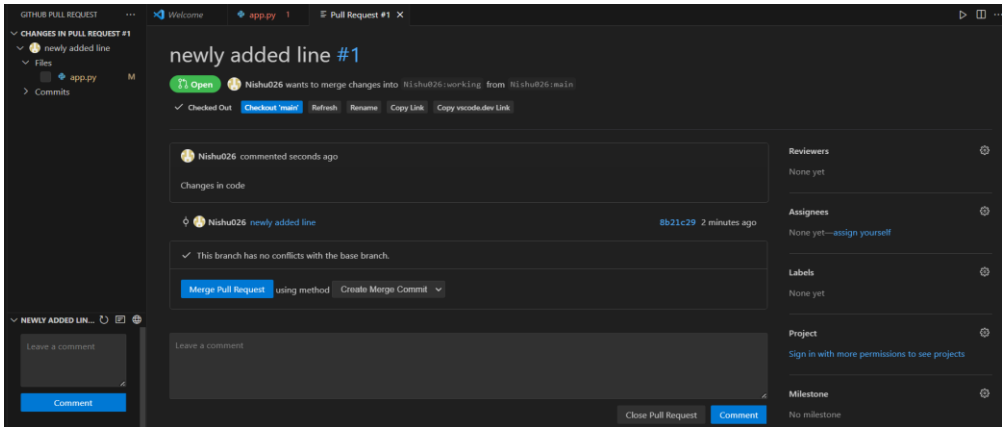
Initialize Repository > Commit changes > Click GitHub View> Create Pull Request



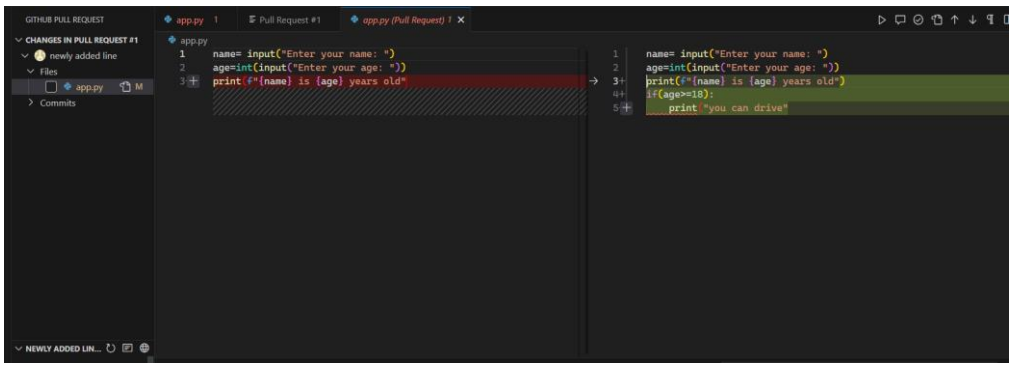
A new Create view will be displayed where you can select the base repository and base branch you'd like your pull request to target as well as fill in the title and description.



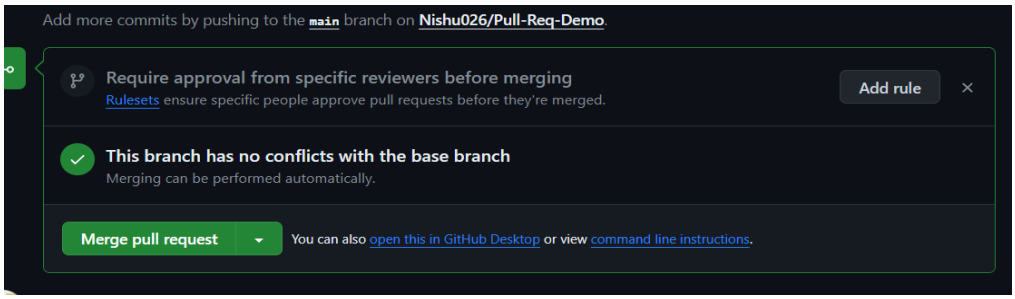
Checkout main branch



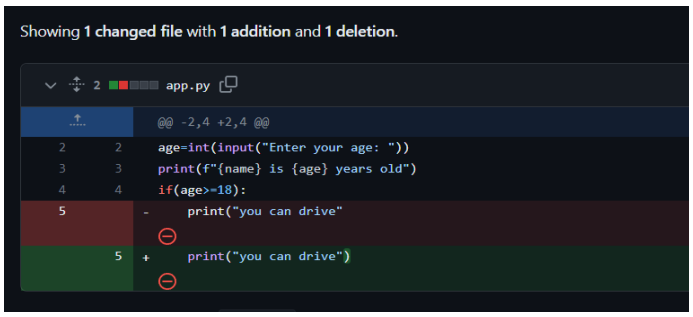
Diff view will open with previous code and newly added.



Fix the errors (if any) > Merge pull request.



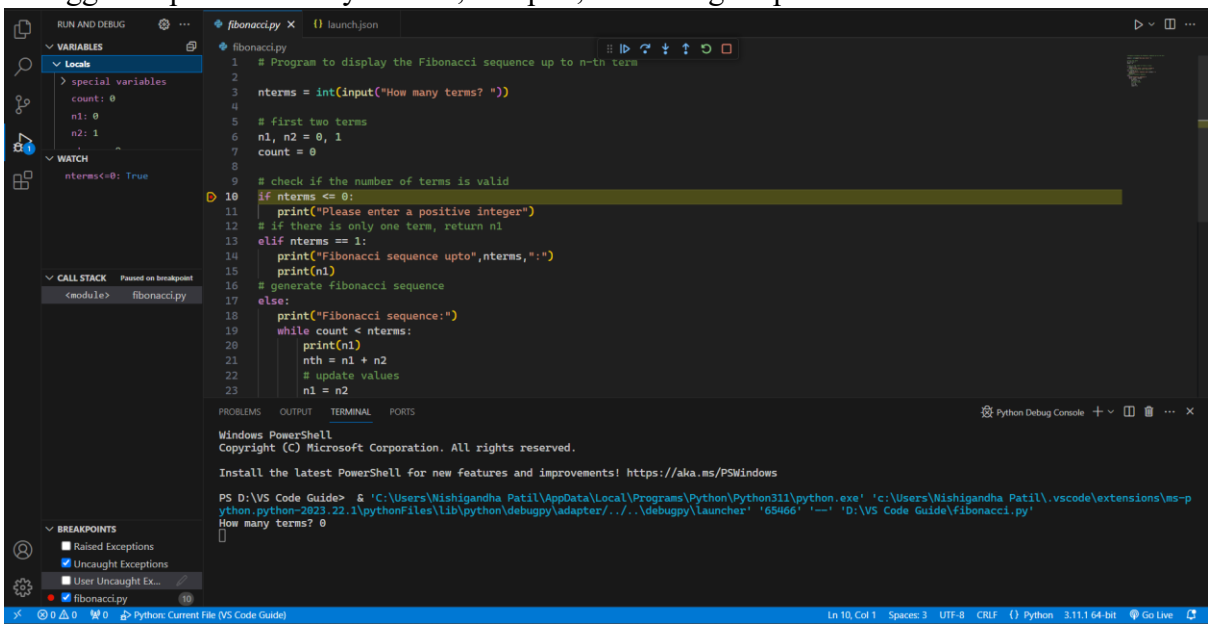
Changes reflected in GitHub repository:



For more information visit: https://code.visualstudio.com/docs/sourcecontrol/github

DEBUGGING AND CODE ANALYSIS

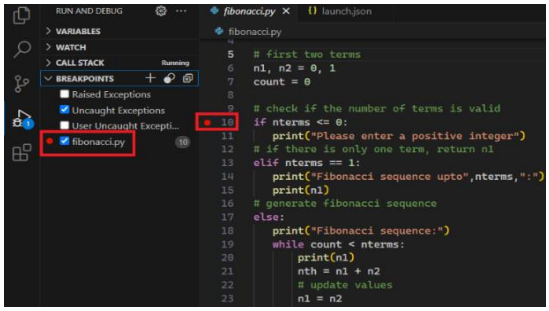
One of the key features of VS Code is its great debugging support. VS Code's built-in debugger helps accelerate your edit, compile, and debug loop.



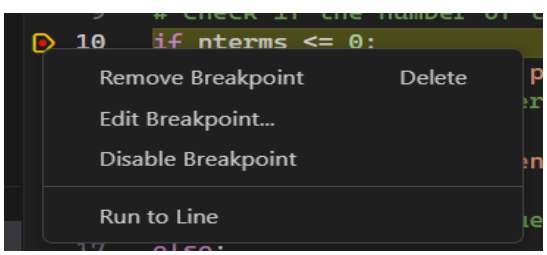
1. Adding Breakpoints:

Breakpoints can be toggled by clicking on the editor margin. Finer breakpoint control (enable/disable/reapply) can be done in the Run and Debug view's BREAKPOINTS section.

- Breakpoints in the editor margin are normally shown as red filled circles.
- Disabled breakpoints have a filled gray circle.
- When a debugging session starts, breakpoints that cannot be registered with the debugger change to a gray hollow circle.
- If the debugger supports breaking on different kinds of errors or exceptions, those will also be available in the Breakpoints view.

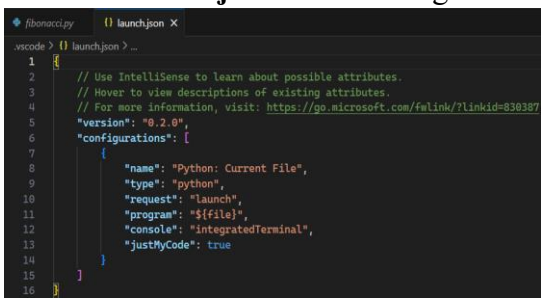


Managing Breakpoints: Right click on breakpoint > Remove/Edit/Disable Breakpoint

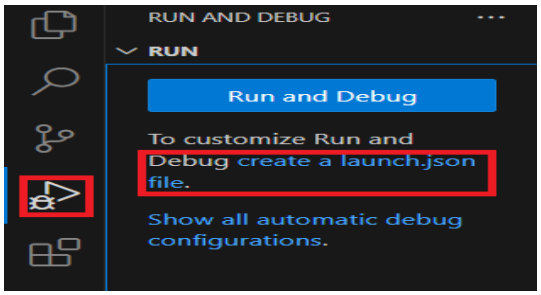


2. Run and Debug:

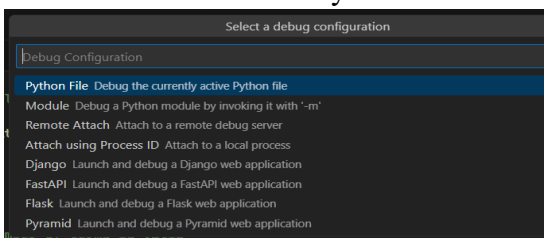
Create a **launch.json** file to configure and save debugging setup details.



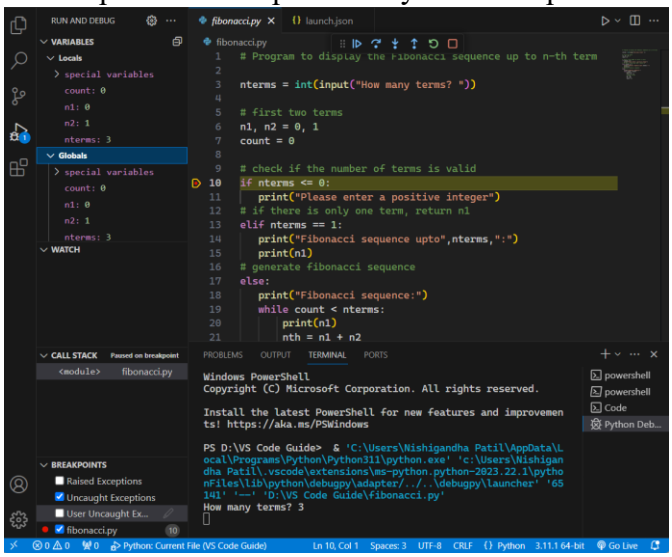
To run or debug a simple app in VS Code, select “Run and Debug” view or press F5.



VS Code will try to automatically detect your debug environment, but if this fails, you will have to choose it manually:



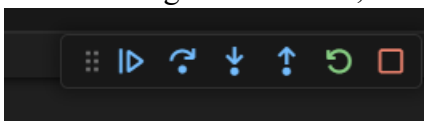
Once you click on Run and Debug option the debugger will run and program execution will stop at the breakpoint and you can inspect the variables.



- **Variables Panel:** Displays all variables in the current scope.
- **Watch Panel:** You can add variables you want to monitor.
- **Call stack:** Refers to the list of function calls that are currently active in a program's execution. When you run a program, each function call is added to the call stack.

3. Debug actions:

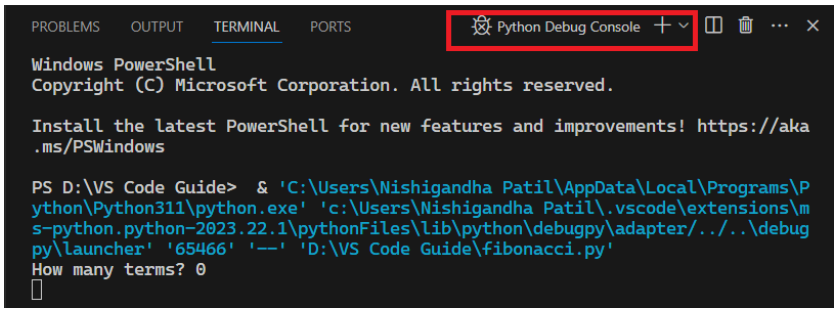
Once a debug session starts, the Debug toolbar will appear on the top of the editor.



Sr. No	Action	Explanation
1.	Continue/Pause F5	**Continue:** Resume normal program execution (up to the next breakpoint). **Pause:** Inspect code executing at the current line and debug line-by-line.
2.	Step Over F10	Execute the next method as a single command without inspecting or following its component steps.
3.	Step Into F11	Enter the next method to follow its execution line-by-line.
4.	Step Out Shift+F11	When inside a method or subroutine, return to the earlier execution context by completing remaining lines of the current method as though it were a single command.
5.	Restart Ctrl+Shift+F5	Terminate the current program execution and start debugging again using the current run configuration.
6.	Stop Shift+F5	Terminate the current program execution.

4. Debug Console:

VS Code includes a debug console where you can execute code snippets and evaluate expressions in the context of your debugging session, aiding in variable inspection and experimentation.

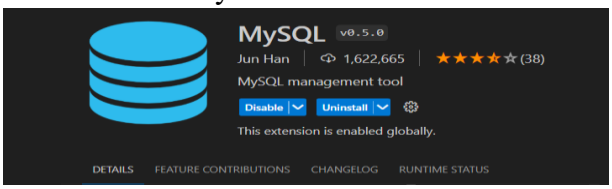


WORKING WITH DATABASES AND REMOTE SERVERS

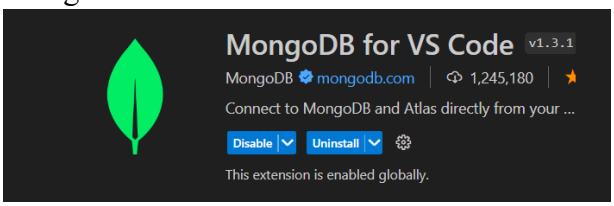
1. Extensions Installation:

For Database Management:

- **SQL Tools:** "SQL Server (mssql)" or "MySQL" allow you to connect and query databases directly from VS Code.

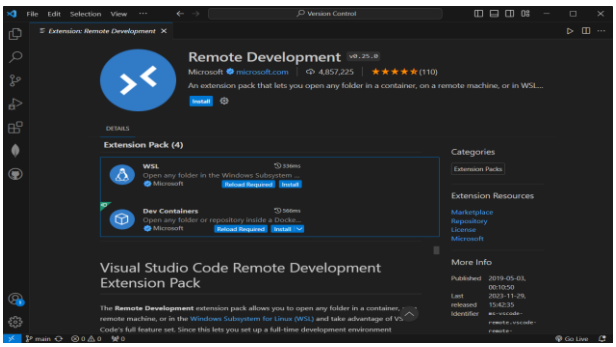


- **MongoDB:** "MongoDB for VS Code" extension offers features to interact with your MongoDB instance.



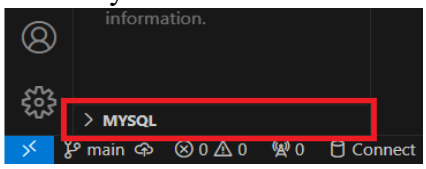
For Remote Server Access:

- **Remote Development:** Install the "Remote Development" extension pack by Microsoft in VS Code. This pack includes extensions for SSH, Containers, and WSL (Windows Subsystem for Linux) that enable you to work on a remote server directly within VS Code.



2. Connecting to Databases:

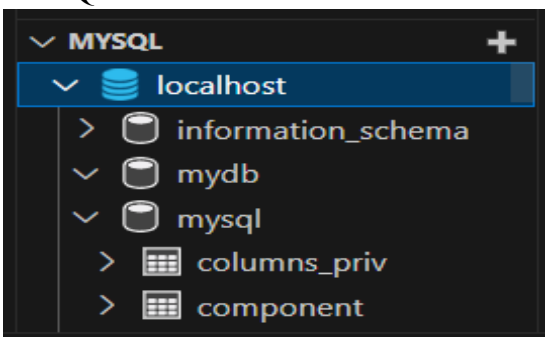
- **SQL Databases (e.g., MySQL, PostgreSQL, SQL Server):** After installing the respective extension, use the provided interface to establish a connection by providing necessary details such as server address, username, password, etc.



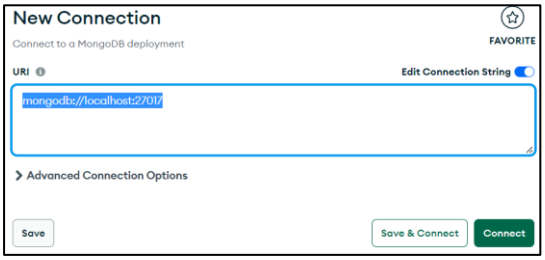
Click on "+" to add connections

Add host > User >Password > Port number

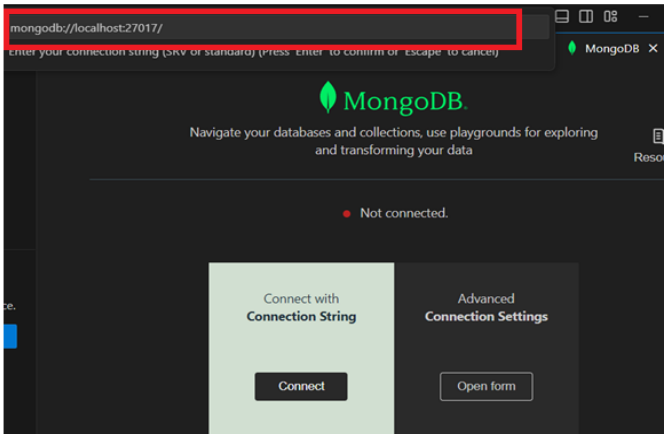
MYSQL connected to VS Code.



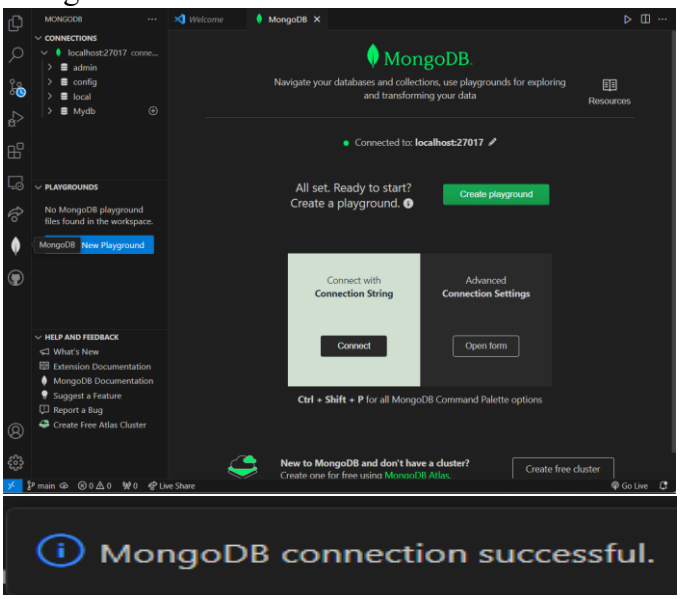
- **MongoDB:** Use the "MongoDB for VS Code" extension to connect to your MongoDB instance by providing connection details.
  Copy Connection URI from MongoDB Paste in Connections (VS Code)



Paste that URI in Connections section in VS Code.

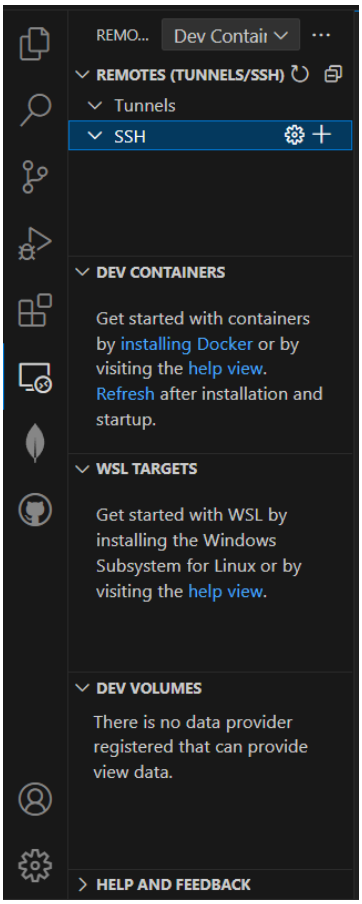


MongoDB connected to VS Code.



3. Using Remote Development:

- **SSH:** With the "Remote Development" pack, you can connect to a remote server via SSH. Use the "Remote-SSH" extension to open folders, edit files, and run commands on a remote machine as if it were local.
- **WSL or Containers:** Similarly, if you're using WSL or Docker containers for your development environment, the "Remote Development" pack supports these scenarios as well.



4. Working with Remote Servers:

Once connected to a remote server using the Remote Development tools, you can edit files, run commands in the terminal, and perform debugging as if you were working on your local machine.

Access files, databases, and execute commands through the integrated terminal in VS Code.

CONCLUSION

Visual Studio Code is an invaluable asset in the development process which offers several important features. Firstly, it equips individuals with a versatile, powerful, and widely-used integrated development environment (IDE) that supports various programming languages. Secondly, mastering VS Code enhances efficiency through its customizable interface, extensive extensions marketplace, and built-in features like debugging, version control, and IntelliSense for code completion. Lastly, understanding VS Code fosters collaboration and adaptability, as it allows seamless integration with different tools and facilitates teamwork in coding projects, enabling individuals to thrive in diverse software development environments.

Learning Outcomes:

- Efficient Coding Practices: Understanding how to navigate and utilize an integrated development environment (IDE) enhances coding efficiency.
- Customization Skills: Learning to personalize settings, themes, and extensions allows for tailored coding environments.
- Debugging Proficiency: Gaining insights into debugging tools and techniques improves the ability to identify and fix code issues effectively.
- Version Control Mastery: Understanding version control systems, especially with Git integration, streamlines code management and collaboration.
- Extension Utilization: Exploring and using various extensions expands functionalities and supports specific programming languages and frameworks.
- Enhanced Productivity: Leveraging VS Code's features optimizes workflows, making coding faster, smoother, and more enjoyable.
- Adaptability to Diverse Projects: Acquiring knowledge applicable across different projects and languages enhances versatility as a developer.