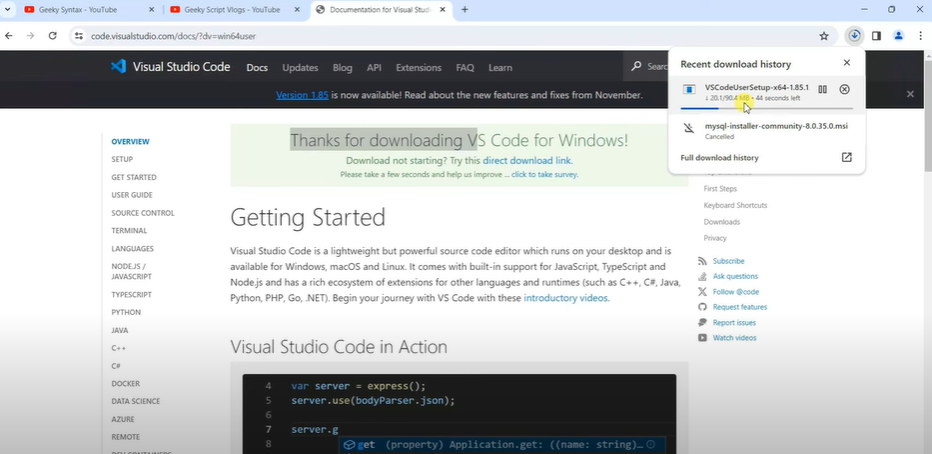
Questions

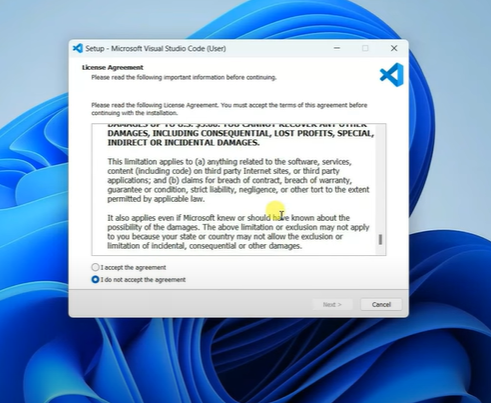
1. Installation of VS Code

**Step 1:** Click on the Windows button to download the installer for your operating system.

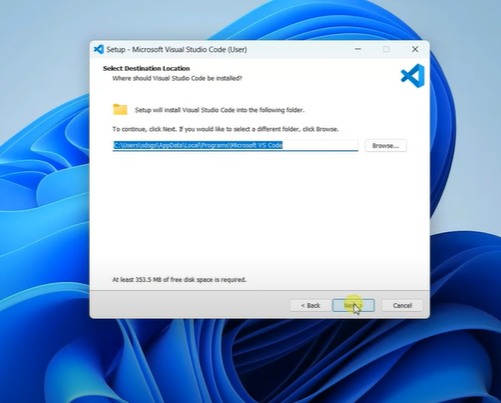


**Step 2:** Install Visual Studio Code

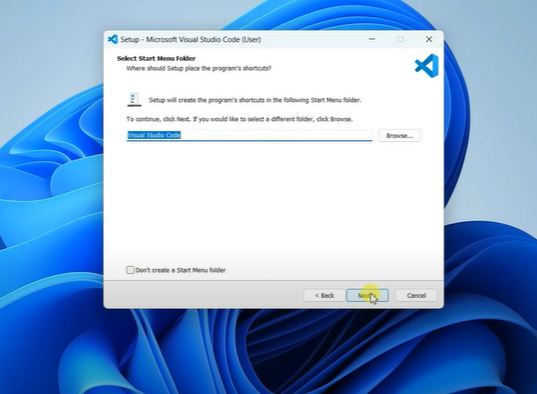
1. Read the license agreement, then check the box to accept the agreement and click Next.



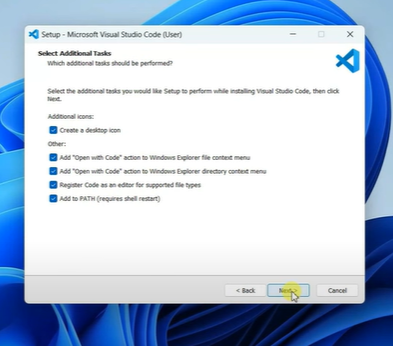
1. Choose the destination folder where you want to install Visual Studio Code.



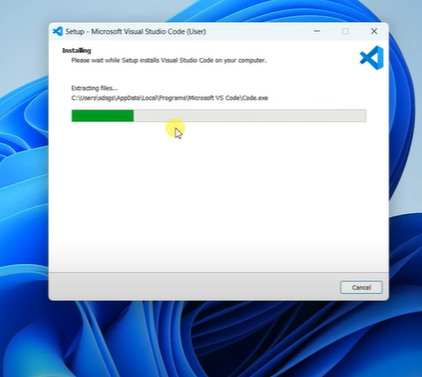
1. Choose the Start Menu folder for the Visual Studio Code shortcuts. The default option is usually fine.



1. Select any additional tasks you would like the installer to perform.

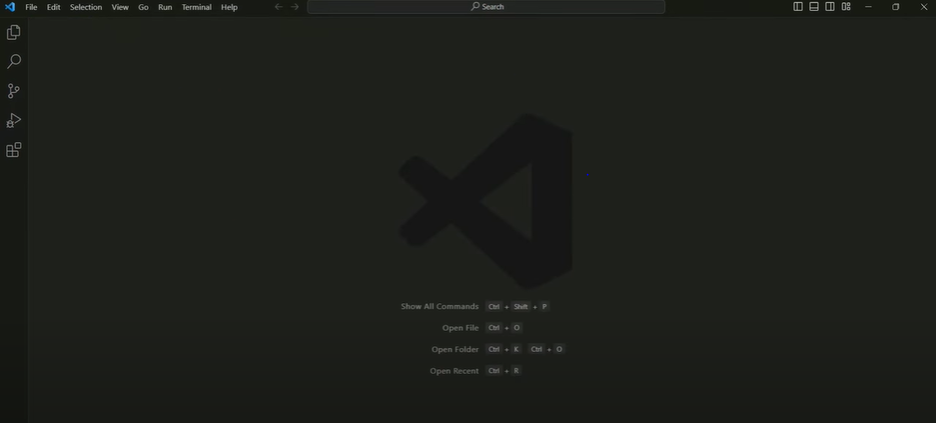


1. Click **Install** to begin the installation.

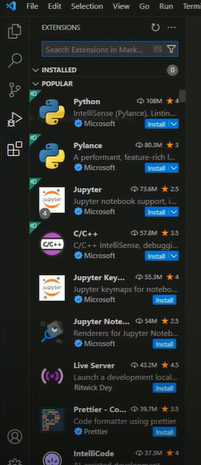


1. I choose to launch Visual Studio Code immediately by checking the box and clicking **Finish**.

**Step 3:** Launch Visual Studio Code



Step 4: Installed extensions like Python and Git.



The prerequisites that may be needed include:

1. Windows 11 Operating System

2. Administrative rights to install software on your machine.

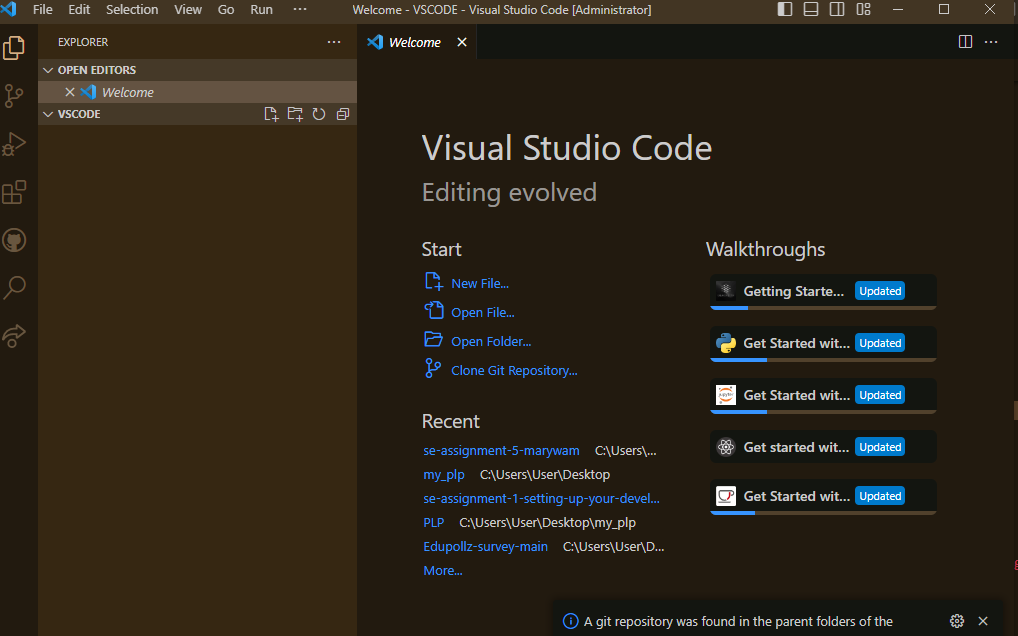
3. Internet Connection that is required to download the installer.

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.

Step 1: Open VS Code

Step 2: Open a folder or create a new workspace.



Step 3 : Configure Settings.

(i) Go to Settings

(ii) Font Size: Set a comfortable font size, e.g., 14.

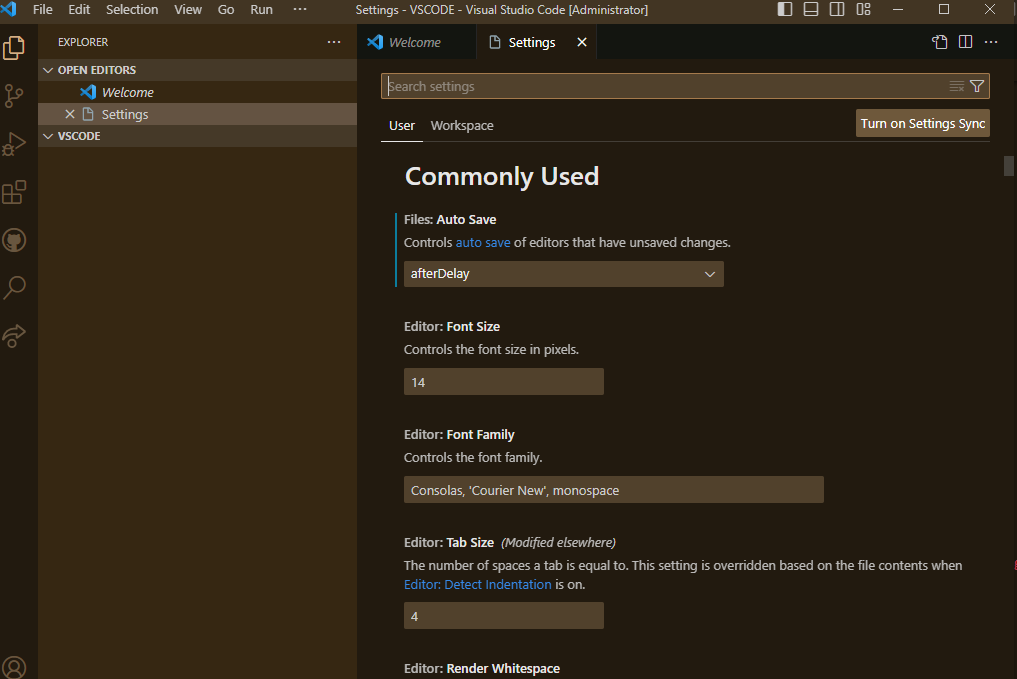
(iii) Tab Size: Set the tab size, e.g., 2 or 4 spaces.

(iv) Format On Save: Enable this setting to auto-format your code on save.

(vi) Auto Save: Set to afterDelay or onWindowChange for automatic saving of files.

(v) Workbench: Color Theme: Choose a theme you find comfortable, such as Dark+

or Light+.



Step 4: Install Key Extensions.

(i) Go to the Extensions view by clicking the square icon on the sidebar.

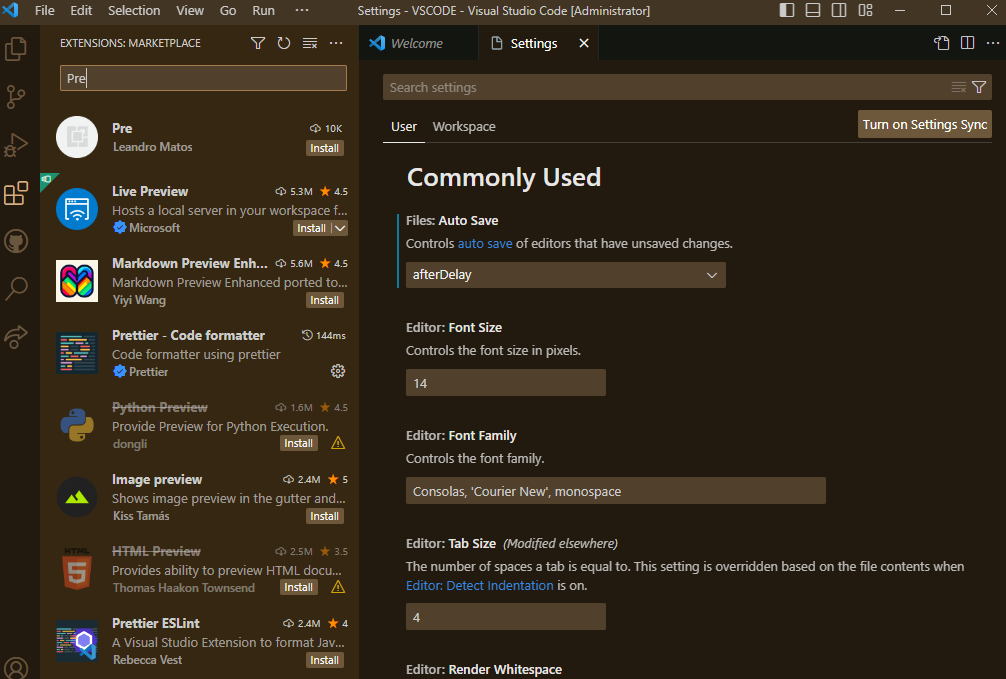
(ii) Install a prettier - Code formatter: For consistent code formatting.

(iii) Install live Server: For launching a local development server with a live reload

feature for static & dynamic pages.

(iv) Install auto Rename Tag: Automatically renames paired HTML/XML tags.

(v) Install gitLens: Enhances the built-in Git capabilities.



Step 5: Set up Version Control

Step 6 :. Configure User and Workspace Settings. User settings apply globally across all projects, while workspace settings apply to the specific project.

**CUSTOMIZING THE USER INTERFACE.**

1. You can move the sidebar (Explorer, Search, Source Control, etc.) to the right side

by right-clicking the activity bar and selecting "Move Side Bar Right".

2. Enable or disable the minimap (a small overview of your code) in View >

Appearance > Show Minimap.

3. Customize the status bar by right-clicking it and selecting or deselecting options as

Needed.

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.

1. Activity Bar

It is located vertically on the far left side of the window.

The Activity Bar provides access to different views and functionalities in VS Code.

Each icon on the Activity Bar represents a different view or extension.

Example:

(i) Explorer - Access and manage files and folders in your workspace.

(ii) Search - Performs text searches across files in your project.

(iii) Source Control - Manage source control repositories (e.g., Git).

(iv) Run and Debug - Start and control debugging sessions.

(v) Extensions - Discover and install extensions to enhance VS Code.



2. Side Bar

It is located directly to the right of the Activity Bar.

The Side Bar displays the contents of the currently selected view in the Activity Bar.

Example:

(i) If the Explorer view is selected, the Side Bar will display a file explorer.

(ii) If the Search View is selected it provides a search interface for finding text in your

files.

(iii) If the Source Control View is selscted,it displays version control information and

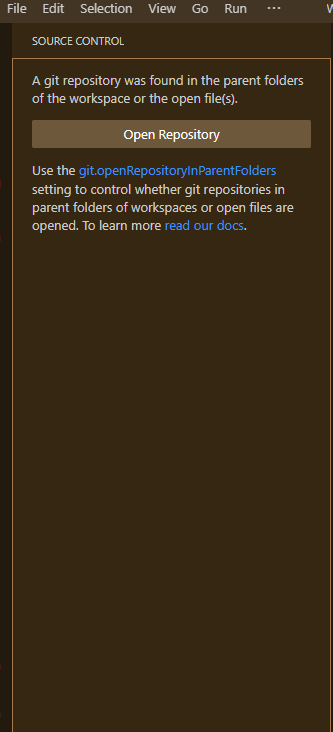
allows you to perform source control operations.

(iv) If the Run and Debug View is selected it will show debugging controls and

information about running processes.

(v) If the Extensions View is selected, it will show a list of installed extensions and

offers recommendations for new extensions.



3. Editor Group

It is located at the Central area of the window, taking up the majority of the space.

The Editor Group is where you write and edit your code. It can be split into multiple

editors.

You can open multiple files in separate editors.

Examples:

(i) Tabs: Each open file is represented by a tab. You can switch between tabs to

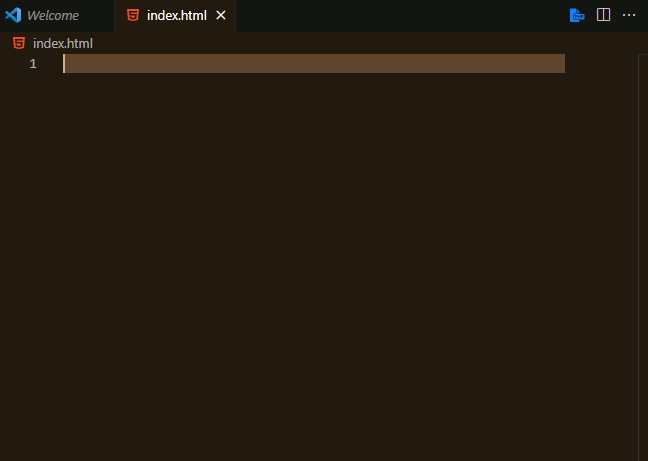
navigate through open files.

(ii) Split Editor: You can split the editor into multiple panes (horizontal or vertical) to

view and edit files side by side.

(iii) Code Editor: The main area where you write and edit your code, with features

like syntax highlighting, IntelliSense (code suggestions), and linting.



4. Status Bar

It is located horizontally along the bottom of the window.

The Status Bar displays information about the current file or project, such as the file

type, line.

It also provides quick access to certain commands and settings.

Examples:

(i) File encoding and line endings.

(ii) Git branch and repository information.

(iii) Language mode and file type.

(iv) Errors and warnings count.

(v) Quick access to commands like "Save All" and "Toggle Word Wrap".



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.

A command palette is a powerful feature in Visual Studio Code (VS Code) that provides quick access to a wide array of commands and functionalities without the need for navigating through menus.

How to Access the Command Palette

Keyboard Shortcut: Press Ctrl + Shift + P or F1.

Menu Access: Click on View in the menu bar and select Command Palette.

Common tasks using the command palette:

(i) File Operations:

Open File: Start typing "Open File" and select the command to open a file.

Save File: Type "Save" to quickly save the current file.

Close File: Type "Close Editor" to close the current file.

(ii) Editing and Navigation:

Go to Line: Type : followed by the line number (e.g., :42) to navigate to a specific

line in the file.

Toggle Sidebar Visibility: Type "Toggle Sidebar Visibility" to show or hide the

sidebar.

Split Editor: Type "Split Editor" to divide the editor into multiple panes for

side-by-side coding.

(iii) Source Control:

Git: Clone: Type "Git: Clone" to clone a repository from a URL.

Git: Commit: Type "Git: Commit" to commit changes with a message.

(iv) Search and Replace:

Find in Files: Type "Find in Files" to open the search functionality.

Replace in Files: Type "Replace in Files" to perform a search and replace

operation across files.

(v) Extensions:

Install Extensions: Type "Extensions: Install Extensions" to browse and install

new extensions.

Disable Extensions: Type "Extensions: Disable" to disable a specific extension.

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.

Role of VS Code extensions

(i) Let you add languages

(ii) Debuggers

(iii) Tools to your installation to support your development workflow.

<https://code.visualstudio.com/docs/editor/extension-marketplace#:~:text=VS%20Code%20extensions%20let%20you,APIs%20used%20by%20VS%20Code.>

How to Find, Install, and Manage Extensions

(i) Finding Extensions

1. Extensions View.

Open the Extensions view by clicking the square icon in the Activity Bar on the side of the window

(ii) Installing Extensions Via Extensions View:

In the Extensions view, type the name of the extension you want to install in the

search bar.

Click on the extension from the search results to open its details page.

(iii) Managing Extensions

1. Enable/Disable Extensions:

In the Extensions view, right-click on an installed extension and choose

"Disable" or "Enable" to toggle its status.

1. Update Extensions:

VS Code typically checks for updates to installed extensions automatically. You

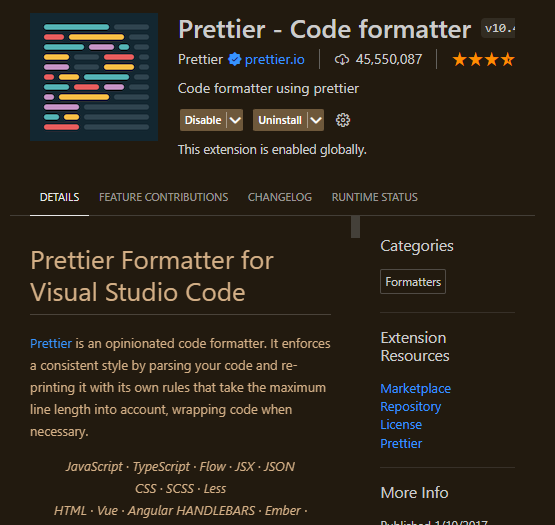
can also manually update extensions by clicking the update button in the

Extensions view when an update is available.

(c) Uninstall Extensions:

In the Extensions view, right-click on an installed extension and select "Uninstall"

to remove it.



Essential Extensions for Web Development

1. Live Server: Launches a local development server with live reload feature for static and dynamic pages.
2. Prettier - Code Formatter: Automatically formats your code to ensure a consistent style.
3. ESLint: Integrates ESLint into VS Code for identifying and fixing JavaScript code issues.
4. GitLens: Enhances Git capabilities within VS Code, providing rich insights into code history and contributors.
5. Git Graph: Provides a visual representation of your Git repository.
6. Debugger for Chrome: Debug JavaScript code in the Chrome browser or other targets that support the Chrome Debugger protocol.
7. Jest: Integrates the Jest testing framework into VS Code.

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?

The integrated terminal in Visual Studio Code (VS Code) allows you to run command-line operations directly within the editor. This feature provides a seamless development experience by consolidating your coding and terminal activities in one place.

**Opening the Integrated Terminal**

1. Keyboard Shortcut:

Press Ctrl + (backtick) or Ctrl + Shift + (backtick) to open the integrated terminal.

1. Menu Access:

Click on View in the menu bar and select Terminal to open a new terminal.

**Using the Integrated Terminal**

1. Splitting the Terminal:

Click the split icon (split terminal icon) in the terminal panel to split the terminal into

multiple panes.

Each pane can run a separate terminal session.

1. Switching Between Terminals:

Use the dropdown menu in the terminal panel to switch between different terminal

sessions.

You can also use keyboard shortcuts like Ctrl + PageUp and Ctrl + PageDown to cycle

through terminals.

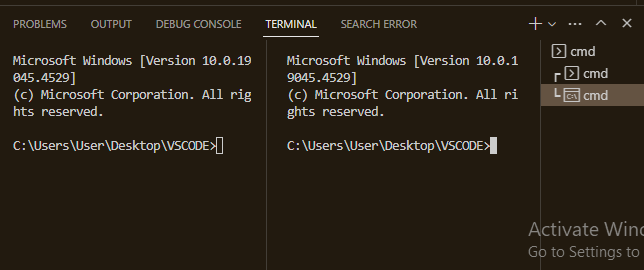
1. Customizing the Terminal:

Right-click on the terminal panel to access options like "Rename", "Select Default Shell",

and "Configure Terminal Settings".

Configure the terminal's appearance and behavior by editing your settings (Ctrl + ,) and

navigating to Terminal settings.



**Advantages of Using the Integrated Terminal Compared to an External Terminal**

1. Convenience and Efficiency:

The integrated terminal allows you to stay within the VS Code environment, reducing the need to switch between different applications.

This helps in maintaining your focus and workflow, as all your development activities are consolidated in one place.

1. Workspace Context:

The integrated terminal opens with the current workspace directory as the default, eliminating the need to navigate to the project directory manually.

Commands and scripts executed in the terminal are contextually relevant to the open project.

1. Customization and Configuration:

Customize the integrated terminal to match your preferences and development needs, such as choosing your default shell (e.g., Bash, PowerShell, Zsh) and configuring terminal behavior through settings.

Easily create and manage multiple terminal sessions, each with its own configuration.

1. Integration with VS Code Features:

The integrated terminal seamlessly integrates with other VS Code features, such as debugging, task running, and source control.

You can run build scripts, version control commands, and test suites directly from the terminal, with outputs linked to the editor.

1. Accessibility:

VS Code provides keyboard shortcuts and commands to interact with the terminal, enhancing accessibility for developers who prefer or rely on keyboard navigation.

1. Synchronized Environment:

The terminal's environment variables are synchronized with the editor, ensuring consistency in paths, configurations, and settings.

You can run editor commands directly from the terminal using the code CLI tool.

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?

1. Creating a New File

(i) Using Explorer:

Open the Explorer view by clicking the Explorer icon in the Activity Bar.

Right-click in the Explorer pane and select New File.

Enter the desired filename and press Enter.

1. Creating a New Folder:

Open the Explorer view (Ctrl + Shift + E).

Right-click in the Explorer pane and select New Folder.

Enter the desired folder name and press Enter.

1. Opening a File:

(i) Using Explorer:

Navigate to the file in the Explorer pane and double-click it to open.

1. Opening a Folder:

(i)Using Explorer:

Click the "Open Folder" button in the Explorer pane if no folder is open, or use the

File > Open Folder menu option.

1. Renaming Files and Folders:

Open the Explorer view (Ctrl + Shift + E).

Right-click on the file or folder you want to rename and select Rename.

Enter the new name and press Enter.

1. Moving Files and Folders:

Drag and drop files and folders within the Explorer pane to move them to a different location.

1. Deleting Files and Folders:

Open the Explorer view (Ctrl + Shift + E).

Right-click on the file or folder you want to delete and select Delete.

Confirm the deletion when prompted.

1. Copying Files and Folders:

Right-click on the file or folder and select Copy.

Right-click in the desired location and select Paste.

1. Explorer View:

Use the Explorer view (Ctrl + Shift + E) to navigate the folder structure of your project. You can expand and collapse directories to find files quickly.

1. Quick Open:

Press Ctrl + P to use the Quick Open feature. Start typing the name of the file you want to open, and VS Code will display a list of matching files. This is one of the fastest ways to switch between files.

1. Go to Symbol:

Press Ctrl + Shift + O to open the Go to Symbol in File dialog, which allows you to navigate directly to symbols (functions, classes, variables, etc.) within the current file.

1. Terminal Navigation:

Use the integrated terminal (Ctrl + ) to navigate directories using command-line commands like cd and ls (Unix/Linux) or dir (Windows).

1. Search Across Files:

Press Ctrl + Shift + F to open the search pane. Enter your search query to find text across all files in your workspace.

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.

Users can find and modify settings through both a graphical interface and a settings

JSON file. Here’s how to access and customize various settings in VS Code:

**Accessing Settings**

1. Settings via GUI:

(i) Keyboard Shortcut: Press Ctrl + , to open the Settings UI.

(ii) Menu Access: Click on File in the menu bar, then select

Preferences > Settings.

1. Settings via JSON File:

(i) Command Palette: Open the Command Palette with Ctrl + Shift + P, then type

Preferences: Open Settings (JSON) and press Enter.

(ii)Menu Access: Click on File in the menu bar, then select Preferences > Settings,

and click the {} icon in the top right to open the settings JSON file.

**Changing the Theme**

1. Settings UI

Open the Settings UI (Ctrl + ,).

In the search bar, type "color theme".

Click on Color Theme under Preferences.

Select your preferred theme from the list.

1. Via Command Palette

Open the Command Palette with Ctrl + Shift + P.

Type Preferences: Color Theme and press Enter.

Choose a theme from the dropdown list that appears.

**Changing the Font Size**

1. Via Settings UI

Open the Settings UI (Ctrl +,).

In the search bar, type "font size".

Adjust the Editor: Font Size setting by entering your preferred font size or using the slider.

**Changing Keybindings**

1. Via Keybindings UI

Open the Command Palette with Ctrl + Shift + P.

Type Preferences: Open Keyboard Shortcuts and press Enter.

This opens the Keyboard Shortcuts editor, where you can search for commands and

change their keybindings by clicking the pencil icon next to a command and pressing

the desired key combination.

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?

**Setting Up and Starting Debugging**

Example: Debugging a Simple JavaScript Program

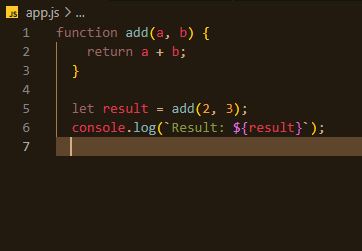
1. Install Node.js:
2. Create a Simple JavaScript Program:

Open VS Code.

Create a new folder for your project.

Open the folder in VS Code (File > Open Folder).

Create a new file named app.js



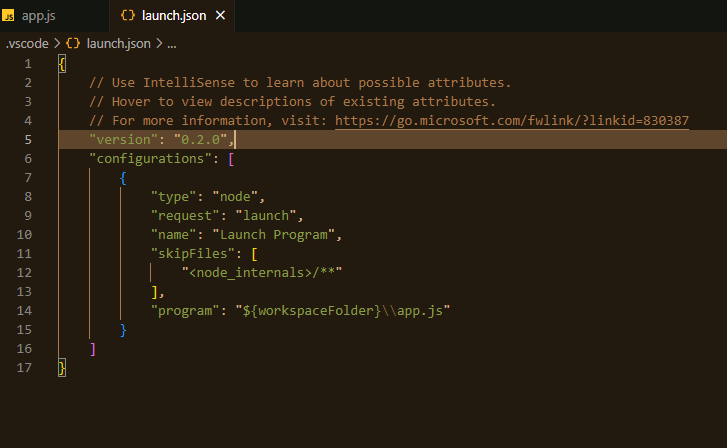
1. Configure the Debugger:

Open the Debug view by clicking the Debug icon in the Activity Bar or pressing Ctrl + Shift + D.

Click the gear icon to open the launch.json file, which contains debug configurations.

If launch.json does not exist, VS Code will prompt you to create one. Select "Node.js" from the environment list.

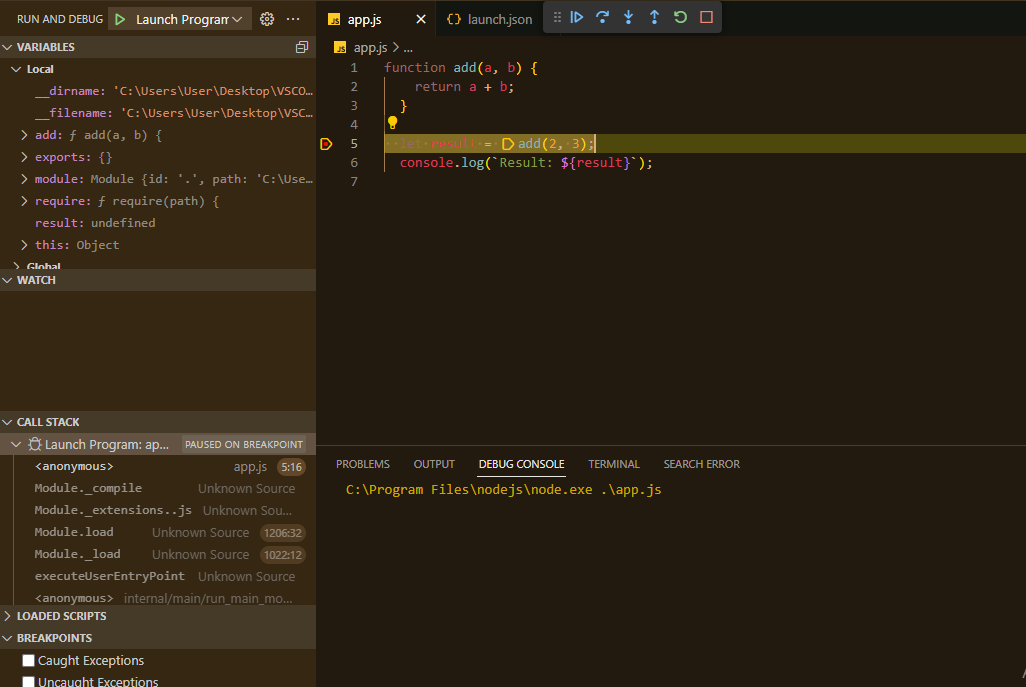
Ensure the launch.json file has the following configuration.



1. Set Breakpoints:

Open app.js and click in the gutter (left margin) next to the line numbers to set

breakpoints. For example, set a breakpoint on the line let result = add(2, 3);.



1. Start Debugging:

In the Debug view, select the configuration "Launch Program" and click the green

play button (or press F5).

**Key Debugging Features in VS Code**

1. Breakpoints:

Breakpoints allow you to pause the execution of your program at specific lines of code to inspect variables and the program state.

Types of breakpoints include line breakpoints, conditional breakpoints, and function breakpoints.

1. Watch Expressions:

Add watch expressions to evaluate and monitor specific variables or expressions while debugging.

You can add a watch expression by clicking the + icon in the "Watch" pane and entering the expression you want to monitor.

1. Call Stack:

The Call Stack pane shows the current execution call stack, allowing you to trace the sequence of function calls that led to the current point in the program.

This helps in understanding the flow of the program and identifying the source of errors.

1. Variables Pane:

The Variables pane displays all variables in the current scope, including local, global, and closure variables.

You can inspect and modify variable values during a debugging session.

1. Debug Console:

The Debug Console allows you to execute commands and evaluate expressions in the context of the currently paused program.

You can access the Debug Console by clicking the "Debug Console" tab in the bottom panel.

1. Step Controls:

Use step controls to navigate through your code during debugging:

Continue (F5): Resume program execution until the next breakpoint.

Step Over (F10): Execute the next line of code, but do not step into functions.

Step Into (F11): Step into the next function call.

Step Out (Shift + F11): Step out of the current function.

1. Inline Values:

VS Code displays inline values for variables right next to their usage in the editor while debugging, making it easier to see the current state of variables.

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 VS Code**

1. Prerequisites

Ensure that Git is installed on your machine. You can download and install it from git-scm.com.

Set up your Git user name and email

1. Initializing a Repository

(i) Open Your Project Folder:

Open VS Code.

Open the folder containing your project by selecting File > Open Folder and navigating to your project directory.

(ii) Initialize the Repository:

Open the Source Control view by clicking the Source Control icon in the Activity Bar or pressing Ctrl + Shift + G.

If your project folder is not already a Git repository, you will see an option to initialize a Git repository. Click the Initialize Repository button.

(iii) Check Git Status:

You will now see a list of untracked files in the Source Control view.

1. Making Commits

(i) Staging Changes:

In the Source Control view, hover over the files you want to commit and click the + icon to stage them.

Alternatively, you can click the + icon next to the Changes header to stage all changes.

(ii) Writing a Commit Message:

After staging the changes, enter a commit message in the input box at the top of the Source Control view.

(iii) Committing Changes:

Click the checkmark icon or press Ctrl + Enter to commit the changes.

1. Pushing Changes to GitHub

(i) Creating a Repository on GitHub:

Go to GitHub and sign in to your account.

Click the + icon in the top right corner and select New Repository.

Fill in the repository name, and description (optional), and set it to public or private.

Click Create Repository.

(ii) Adding a Remote Repository:

Copy the repository URL (HTTPS) from GitHub.

In VS Code, open the integrated terminal by pressing Ctrl + (backtick) or selecting Terminal > New Terminal.

Add the remote repository using the following command:

git remote add origin <https://github.com/Powerlearnproject/se-assignment-5-marywam>

(iii) Pushing Changes to GitHub:

Push the committed changes to GitHub using the following command:

git push -u origin master

If this is your first push, you might need to authenticate with your GitHub credentials.