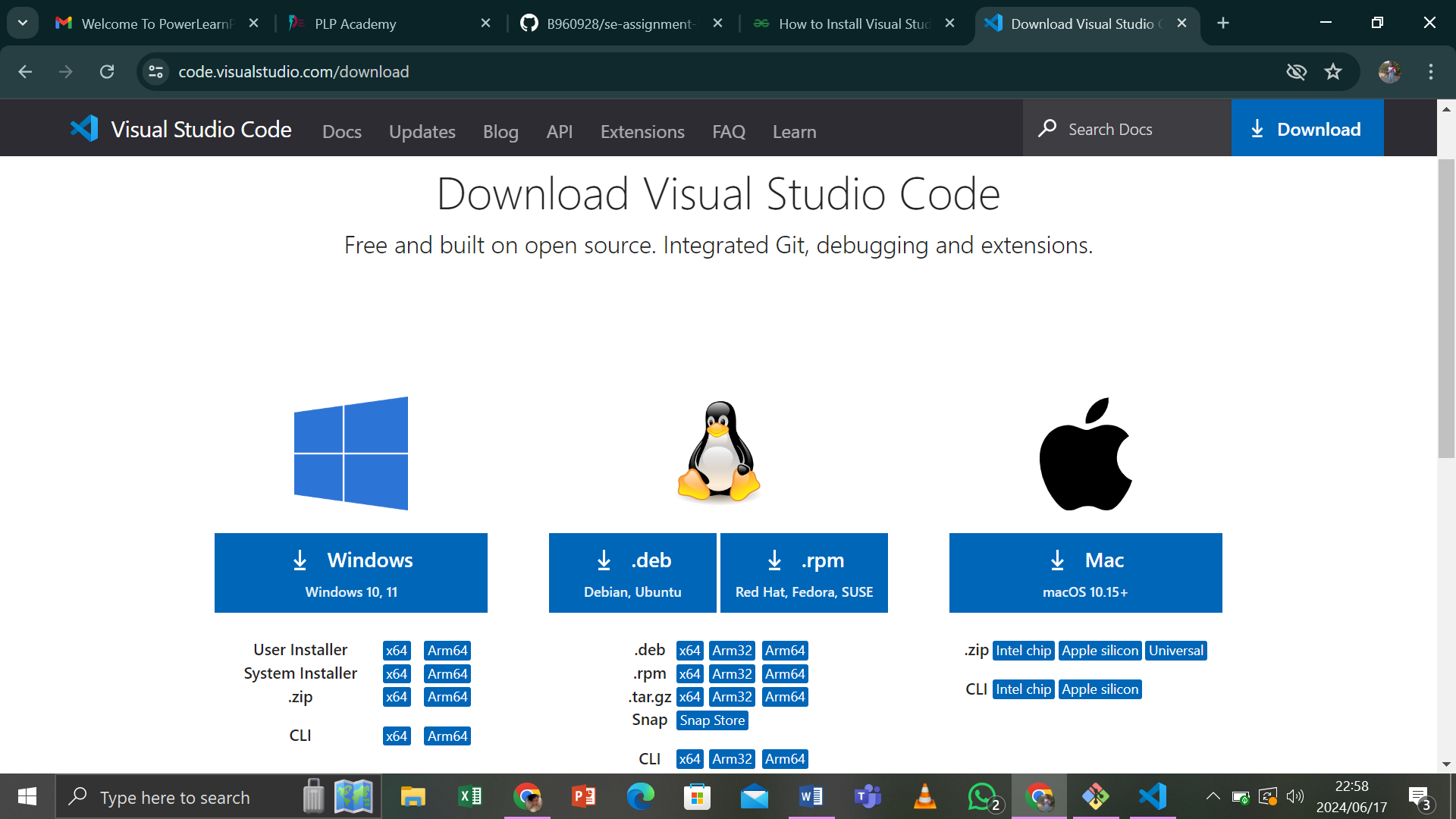
Visual Studio Code Setup Documentation

1. Installation of VS Code:

* Download the VS Code file from the Official Website (<https://code.visualstudio.com/download>)
* Execute the download file
* Accept the Terms & Conditions
* Clicks on the install Visual Studio Code for Windows
* Wait for the installation to complete
* Clicks on the Launch button to start it



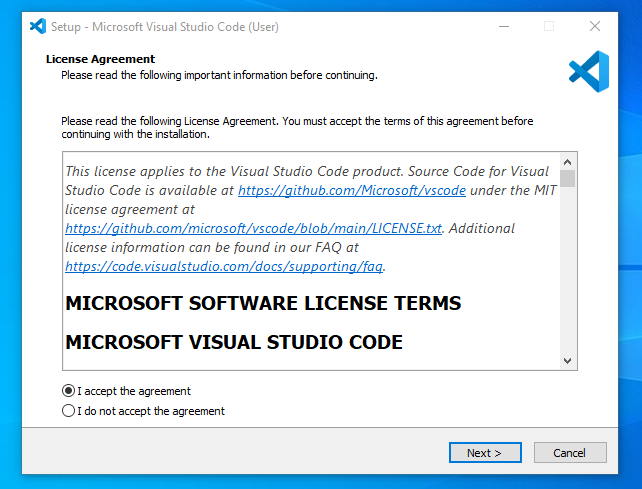
1. Steps to Install Visual Studio Code on Windows

Step 1: Press the “download for windows” button on the website to start the download of the VS Code application.

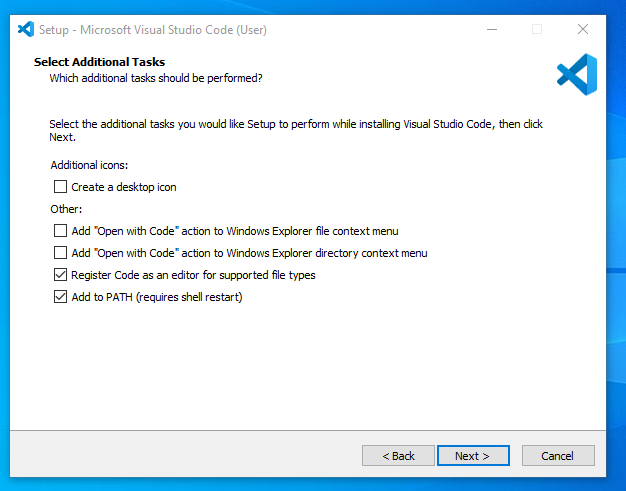
Step 2: When the download finishes, then the Visual Studio Code Icon appears in the download folder

Step 3: Click on the Installer icon to start the installation process of the Visual Studio Code.

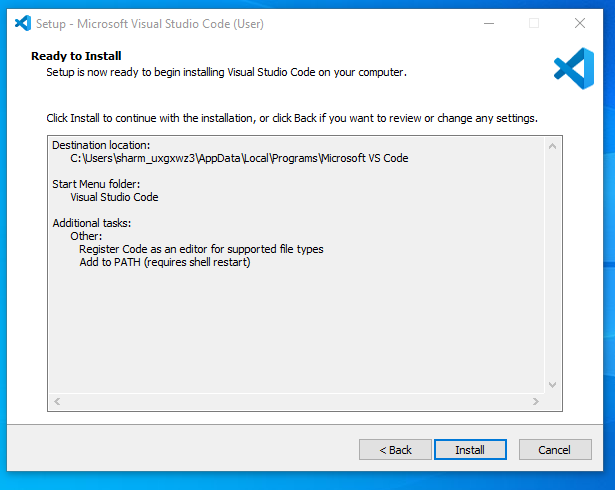
Step 4: After the Installer opens, it will ask you to accept the terms and conditions of the VS Code. Click on I ACCEPT THE AGREEMENT and then click the next button.

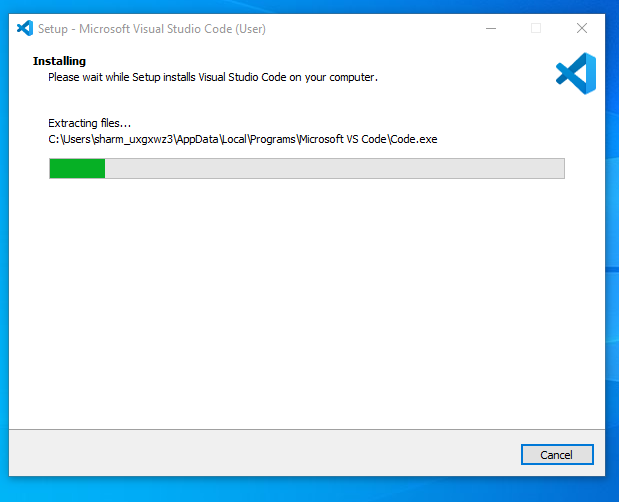


Step 5: Choose the location data for running the Visual Studio Code. It will then ask you to browse the location. Then clicks on the Next button

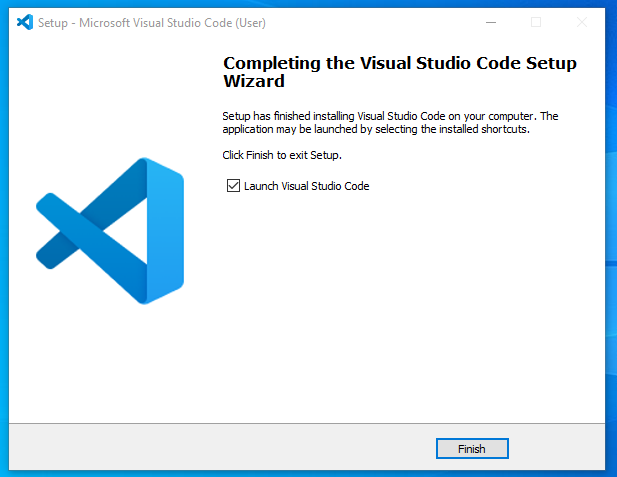


Step 6: Then it will ask to begin the Installation Setup. Click on the Install button

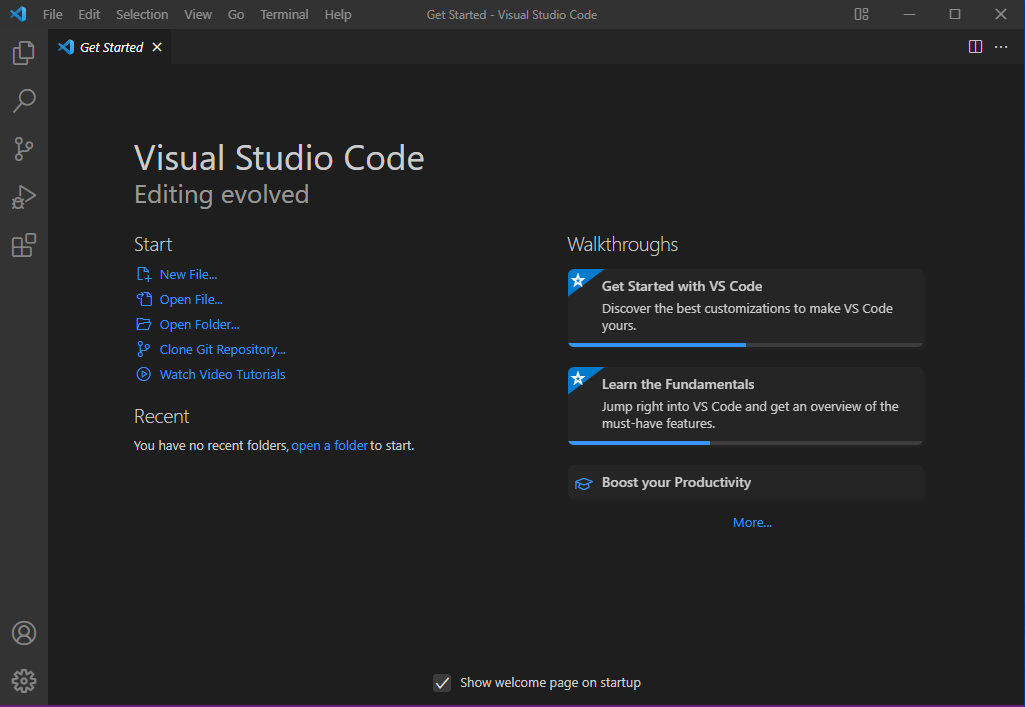


Step 7: After clicking on Install, it will take about 1 minute to Install the VS Code on the device

Step 8: After the Installation setup. Tick the ‘Launch VS Code’ checkbox and the click next



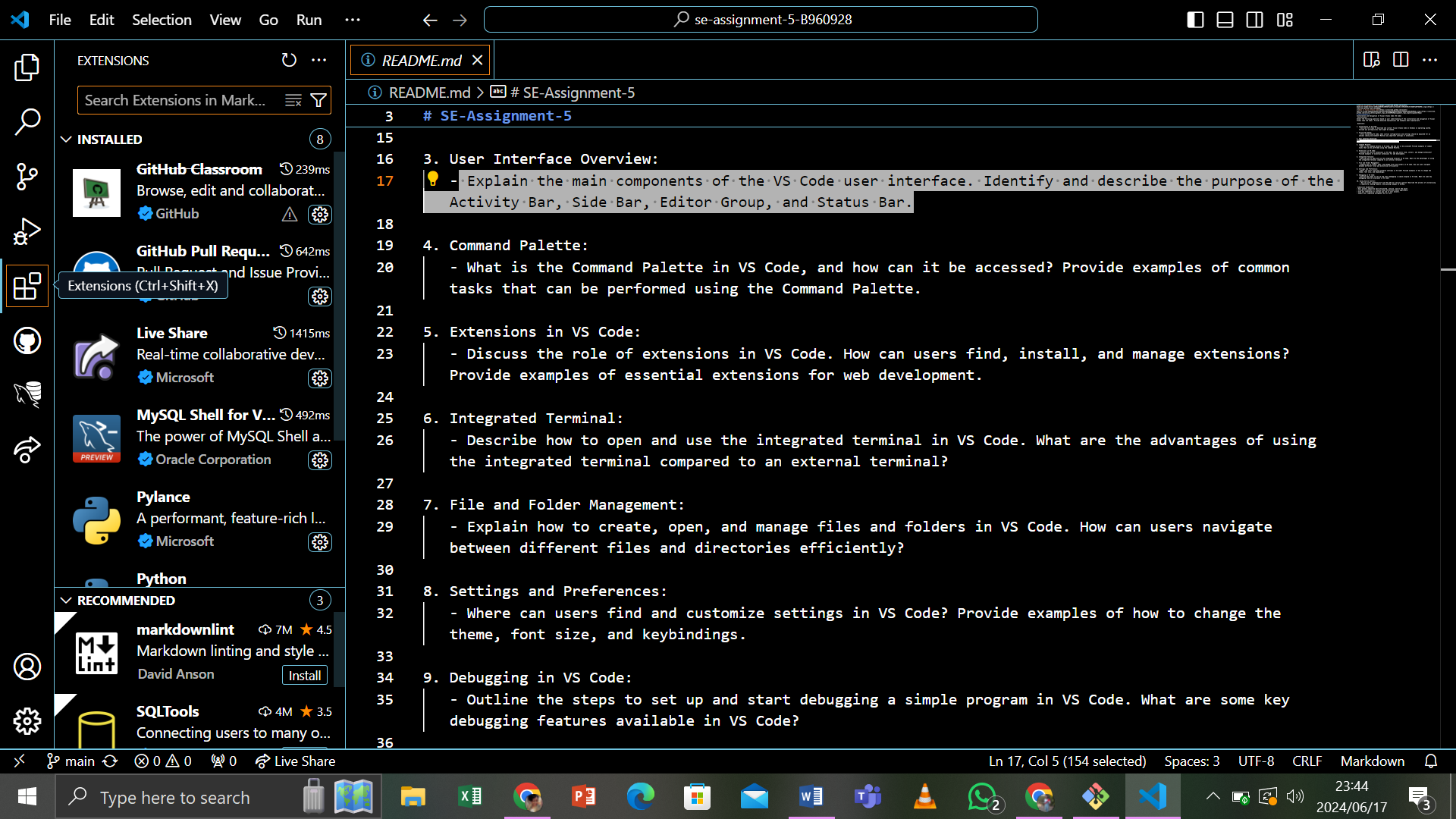
Step 10: Visual Studio Code window opens successfully. Now you can create a new file in the VS Code window and choose a Language of yours to begin your programming journey



This is how I successfully Installed VS Code on Windows system. The steps mentioned in the above guideline can be used in any kind of Windows Browsers for Downloading and Installation VS Code IDE on the device as well.

1. User Interface Overview:
2. Main components of VS Code User Interface:

* An explorer on the left, showing all the file and folders you have access to, and an editor on the right, showing the content of the files you have opened



1. Identify and describe the purpose of the Components:
2. Activity bar:

* Located on the far left, provides access to different views like Explore, Search, Source Control, Run and Debug Extension
* The Activity Bar is a core navigation surface in VS Code. Extensions can contribute view containers to the Activity Bar that appear.

DO: Use an icon that matches the default Activity Bar item icon style

: Use a clear, obvious name for the View Containers associated with the item

DON’T: Duplicate an existing icon

: Use an Activity Bar item to open a Webview panel

1. Side Bars:

* The primary and secondary Sidebars consists of one or more Views contributed by a view container. Extensions can contribute views to an existing view container (for Example, Explorer) or they can contribute an entirely new view container

DO: Group related Views and content together

: Use clear, descriptive names for View containers and their Views

DON’T: Use an excessive number of view containers. A single view container (such as a Sidebar with views unique to that extension) is generally enough for most extensions

: Add content to the Sidebar that could be a simple command

: Repeat existing functionality

1. Editor Group:

* Can appear in the editor toolbar. You can either add an icon as a quick action or add menu item under the over menu (…)

DO: Show only when contextually appropriate

: Use icon from the icon library

: Use the overflow menu for secondary actions

DON’T: Add more than one icon

: Add custom colors

1. Status Bar

* Sits at the bottom of the VS Code workbench and displays information and actions that relate to your workspace.

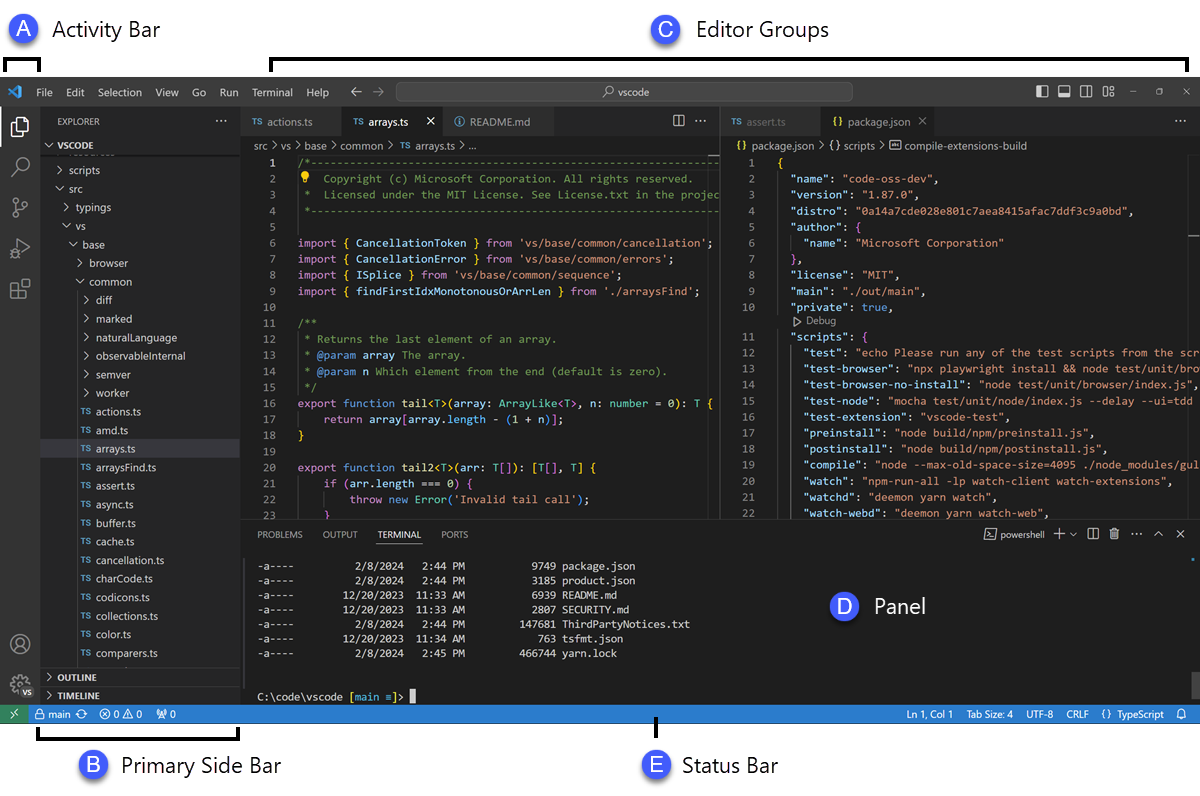
DO: Use short text labels

: Use icon only when necessary

: Place primary (global) items on the left

DON’T: Add custom colors

: Add more than one icon (unless necessary)



4.Command Palette

1. What is the Command Palette in VS Code:

* Is where all Commands are found. It’s important that your command names are labelled appropriately so users can easily find them.

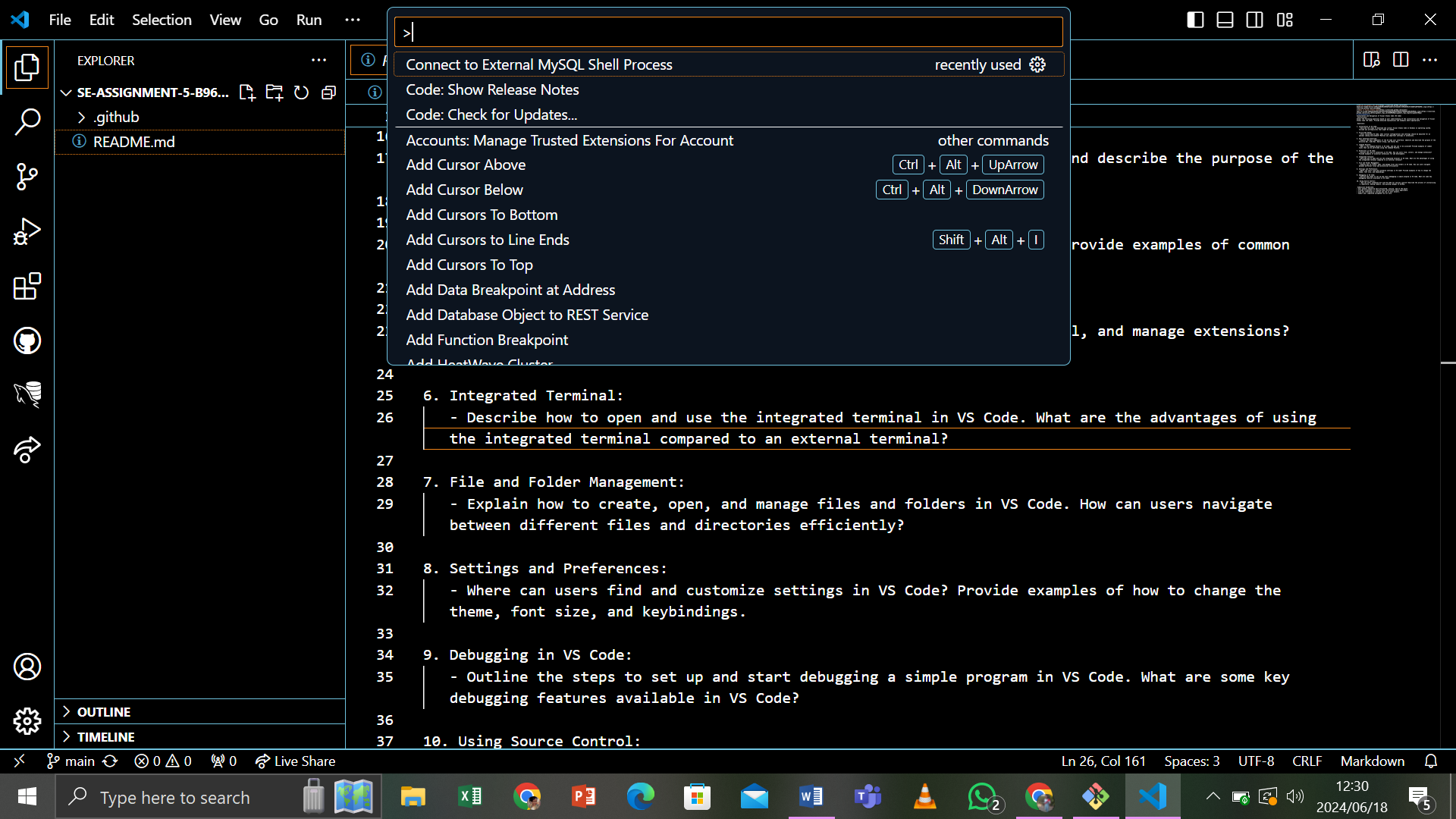
DO: Add keyboard shortcuts where appropriate

: Use clear names for commands

: Group commands together in the same category

DON’T: Overwrite existing keyboard shortcuts

: Use emoji’s in command names



2. How to access Command Palette on VS Code

5. Extensions in VS Code

1. The role of Extensions in VS Code:

* VS Code Extensions let us add languages, debuggers, and tools to your Installation to support your development workflow.

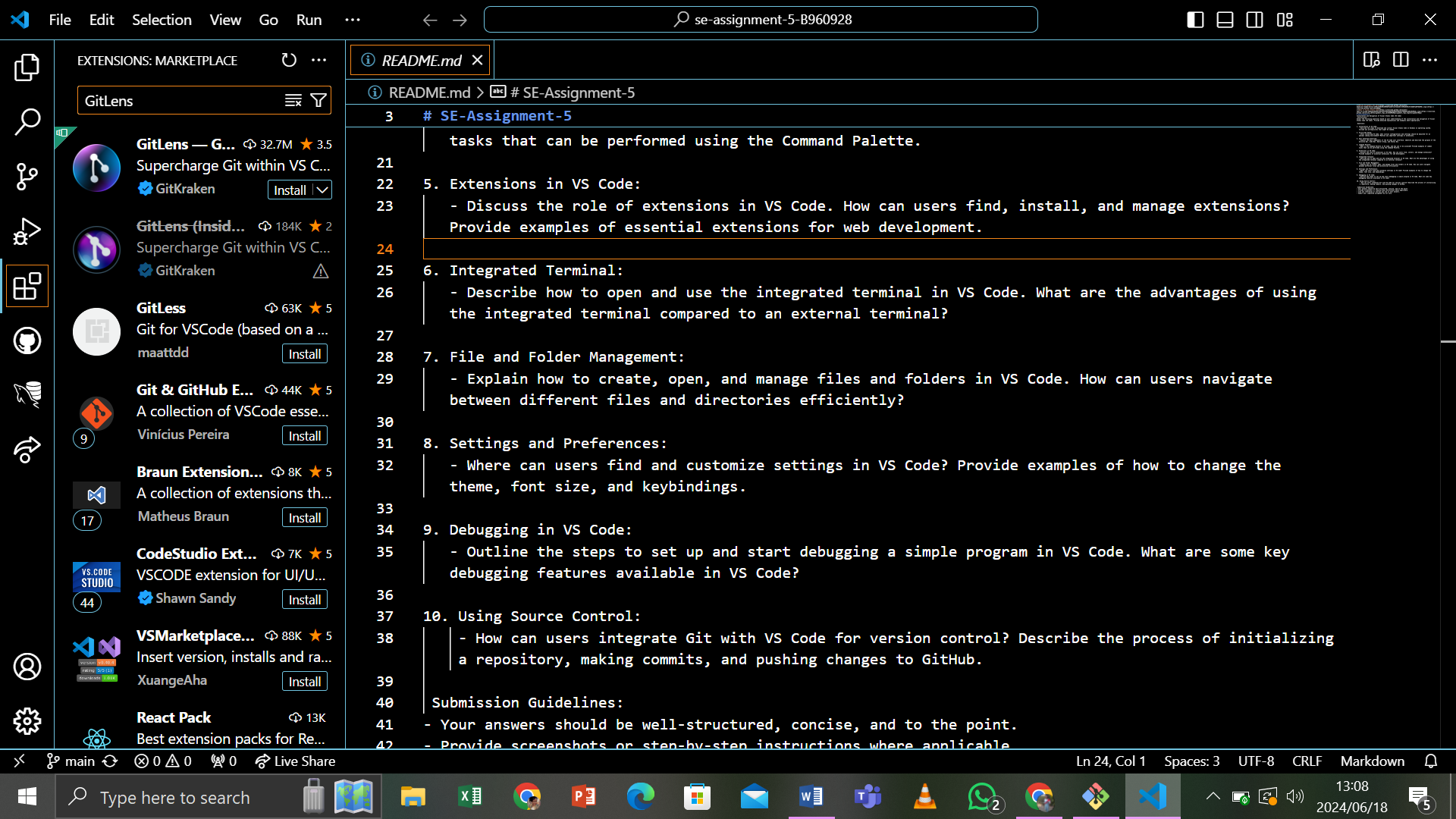
2. Browse for Extensions:

* You can Browse and Install extensions from within VS Code. Bring up the Extensions view by clicking on the Extensions Icon in the Activity Bar on the side of VS Code or the View
* Extensions Command: (Ctrl+Shift+X)

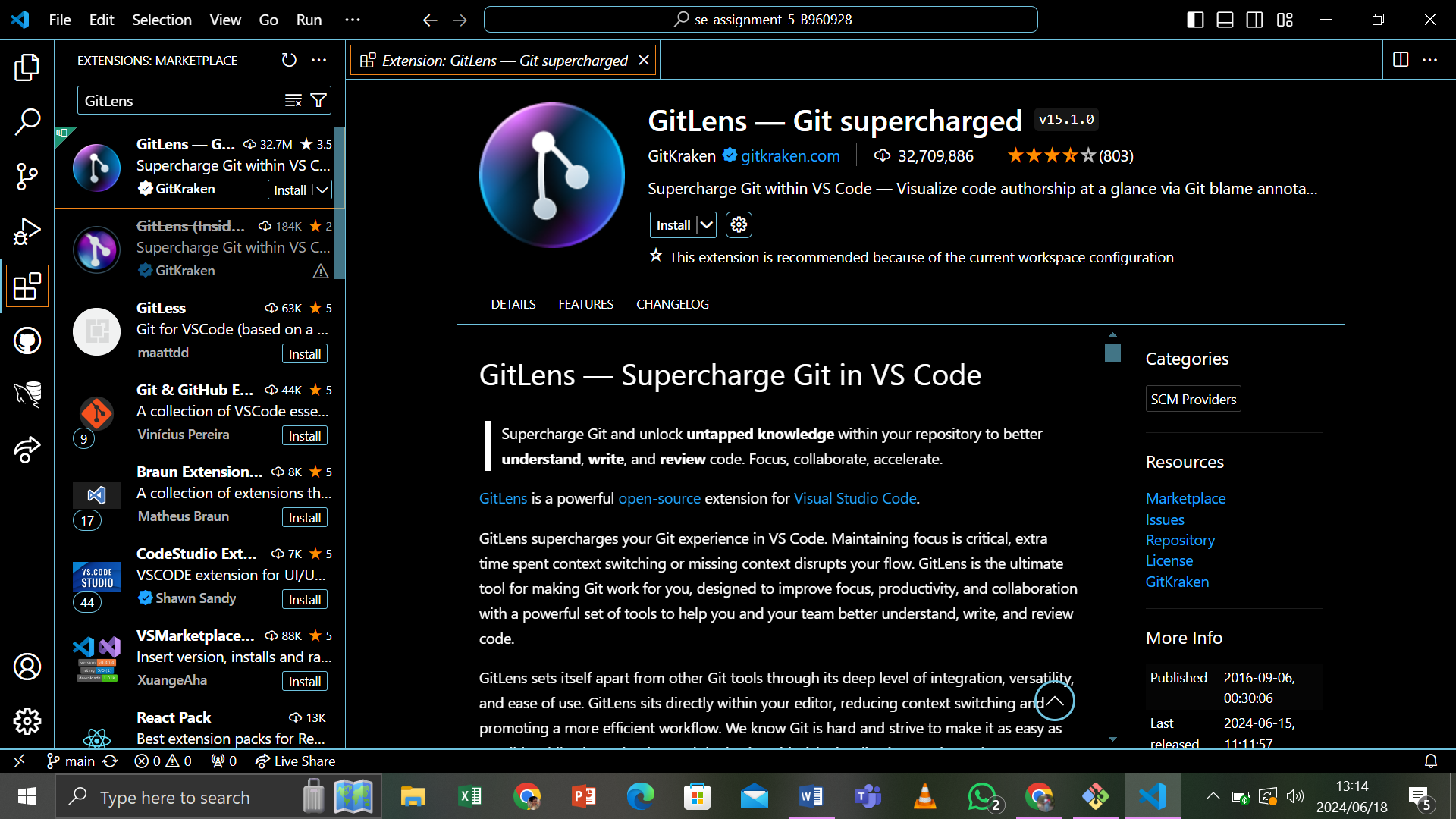


3. Install an Extension:

* To Install an Extension, select the Install button. Once the Installation is complete, the Install button will change to the Manage gear button.
* For example: Let’s Install the popular

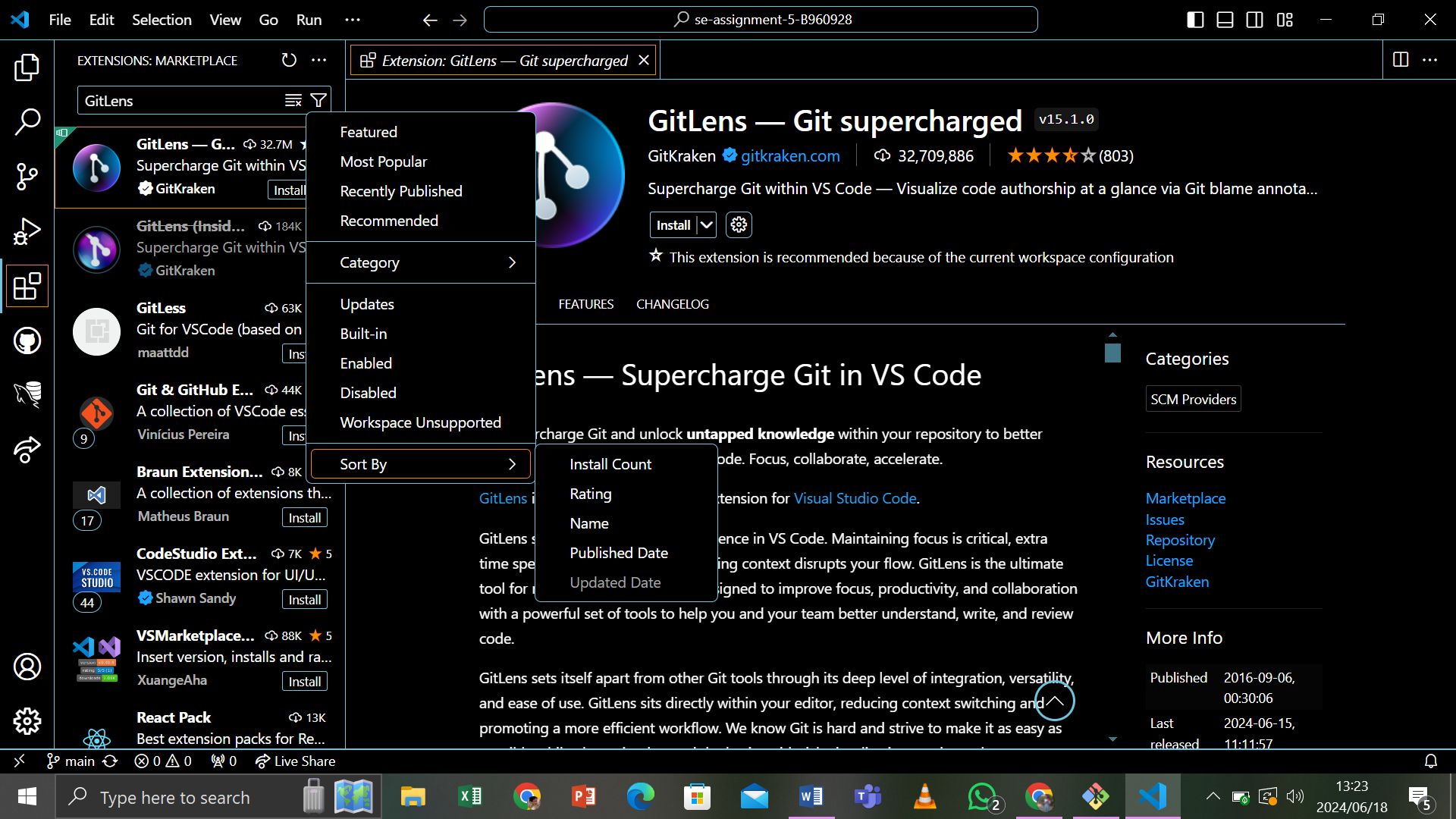


* An Extension is uniquely identified by its publisher and extension IDs. If you select the GitLens highlight extension, you will see the Extension details page, where you can find the Extension ID, in this case wayou.vscode-gitlens-highligth. Knowing the Extension ID can be helpful if there are several similarly named extension.



4. Extension View Filter and Command:

* You can Filter the Extensions view with Filter Extension context menu.



* There are Filter to show:

> The list of outdated extensions that can be updated.

> The list of currently enabled/disabled extensions.

> The list of recommended extensions based on your workspace

> The list of globally popular extensions.

5. Configuring Extensions:

* VS Code extensions may have very different configurations and requirements. Some extensions contribute settings to VS Code, which can be modified in the settings editor.
* Extensions may also require installation and setup od additional componets like compliers, debuggers, and command-line tools.
* Consult the extension’s README (visible in the extension view details page)

6. Command Line Extension Management:

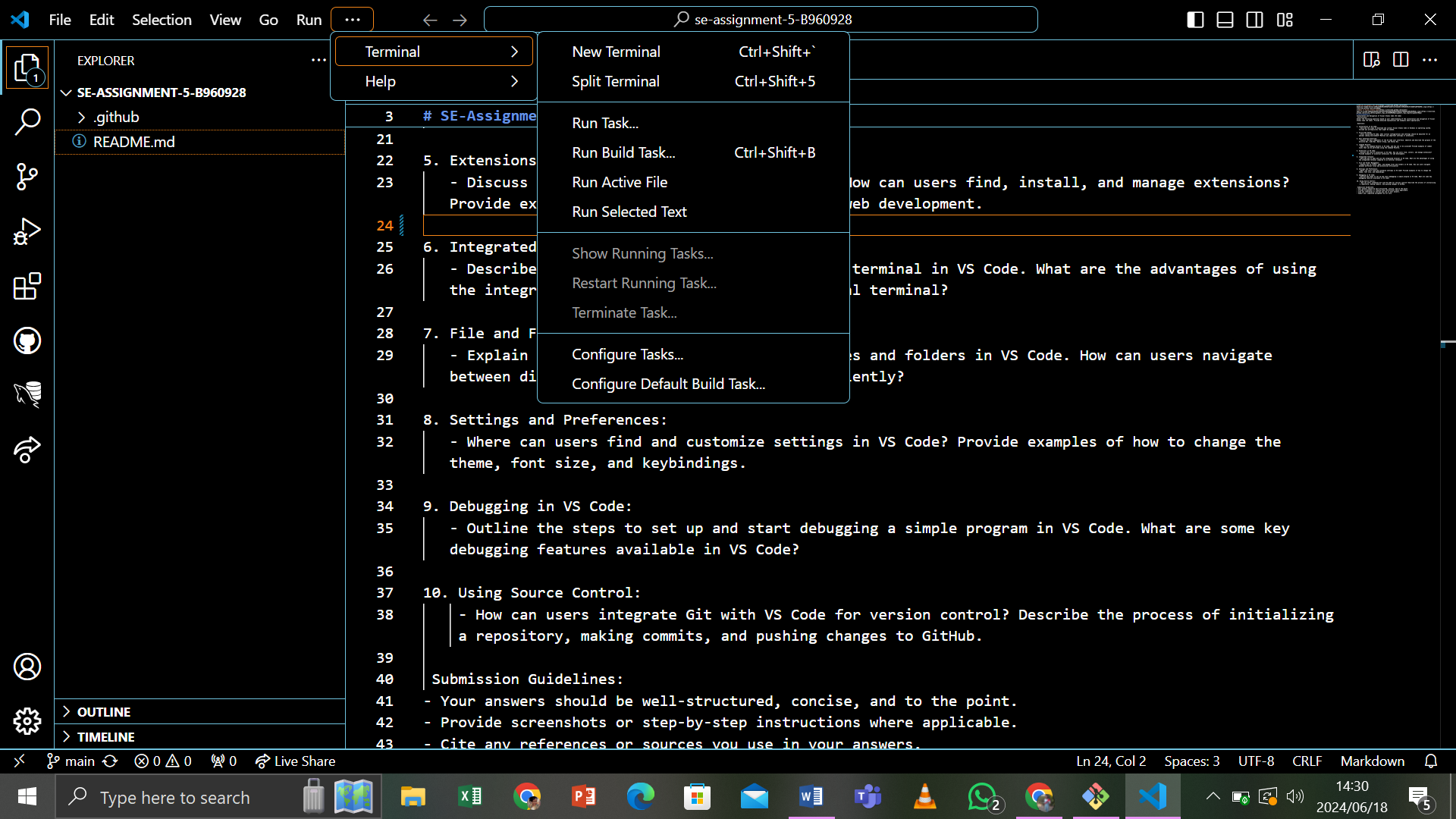
* To make it easier to automate and configure VS Code, it is possible to list, Install, and uninstall extensions from the command-line.
* When identifying an extension, provide the full name of the form publisher.extension, for Example: ms-python.python .

6. Integrated Terminal

* VS Code includes a full featured integrated terminal that starts at the root of your workspace. It provides integration with the editor to support features like Links and Error detection.
* The integrated terminal can run commands such as mkdir and git just like a standalone terminal.

1. Open a terminal as follows:

* From the menu, use the Terminal > New Terminal or View > Terminal menu commands
* From the Command Palette (Ctrl+Shift+X), use the View: Toggle Terminal command
* In the Explorer, you can use the Open in integrated Terminal context menu command to open a new terminal from a folder
* To toggle the terminal panel, use the Ctrl+ keyboard shortcut
* To create a new terminal, use the Ctrl+Shift+ keyboard shortcut



2. Using the Terminal:

* Execute commands as you would in any terminals
* Create multiple terminals and switch between them

Advantages of Using the Integrated Terminal:

* Direct Interaction with project files
* Supports multiple terminal sessions
* Provides a consistent environment within the editor
* Integrated with VS Code features (e.g., debugging, source control)

Example: To run a Node.js server, open the terminal with Ctrl+, navigate to the project directory, and run node serve.js

7. File and Folder Management

1. Creating, Opening and Managing File and Folders

1. 1. Creating Files/Folders:

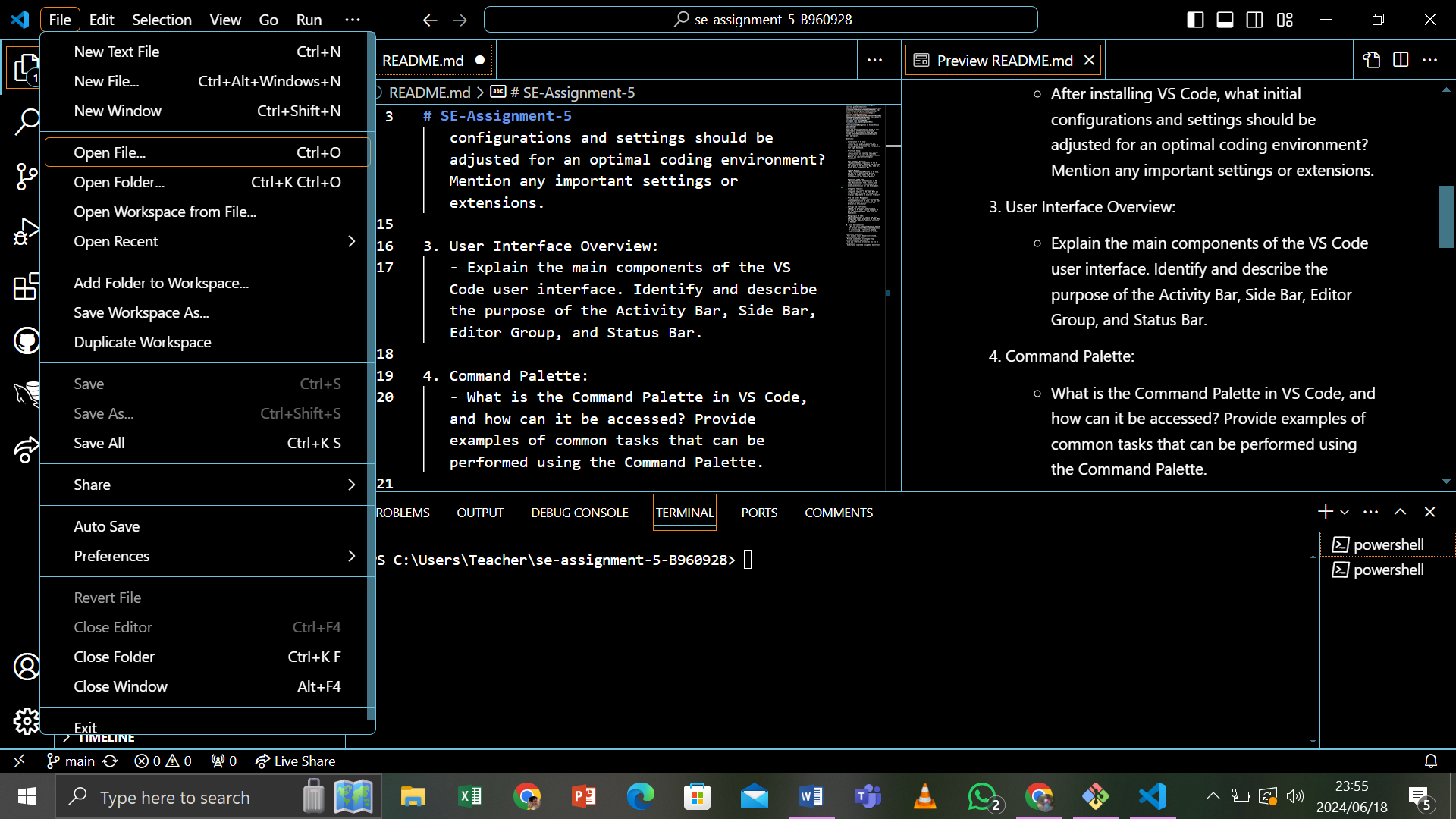
* Right click in the Explorer view and select “New File” or “New Folder”
* Use the Command Palette (Ctrl+Shift+P) and type “New File” or “New Folder”

1.2. Opening Files:

* Double click a file in the Explorer view to open it in the Editor
* Use the file > Open File Menu Options

1.3. Managing Files:

* Use the context menu in the Explorer view to rename, delete or move files and folders



2. Navigating Between Files and Directories Efficiently

* Use the Explorer view to browse directories
* Use Ctrl+P to quickly open a file name
* Use the breadcrumbs feature at the top of the editor to navigate the file

Example:

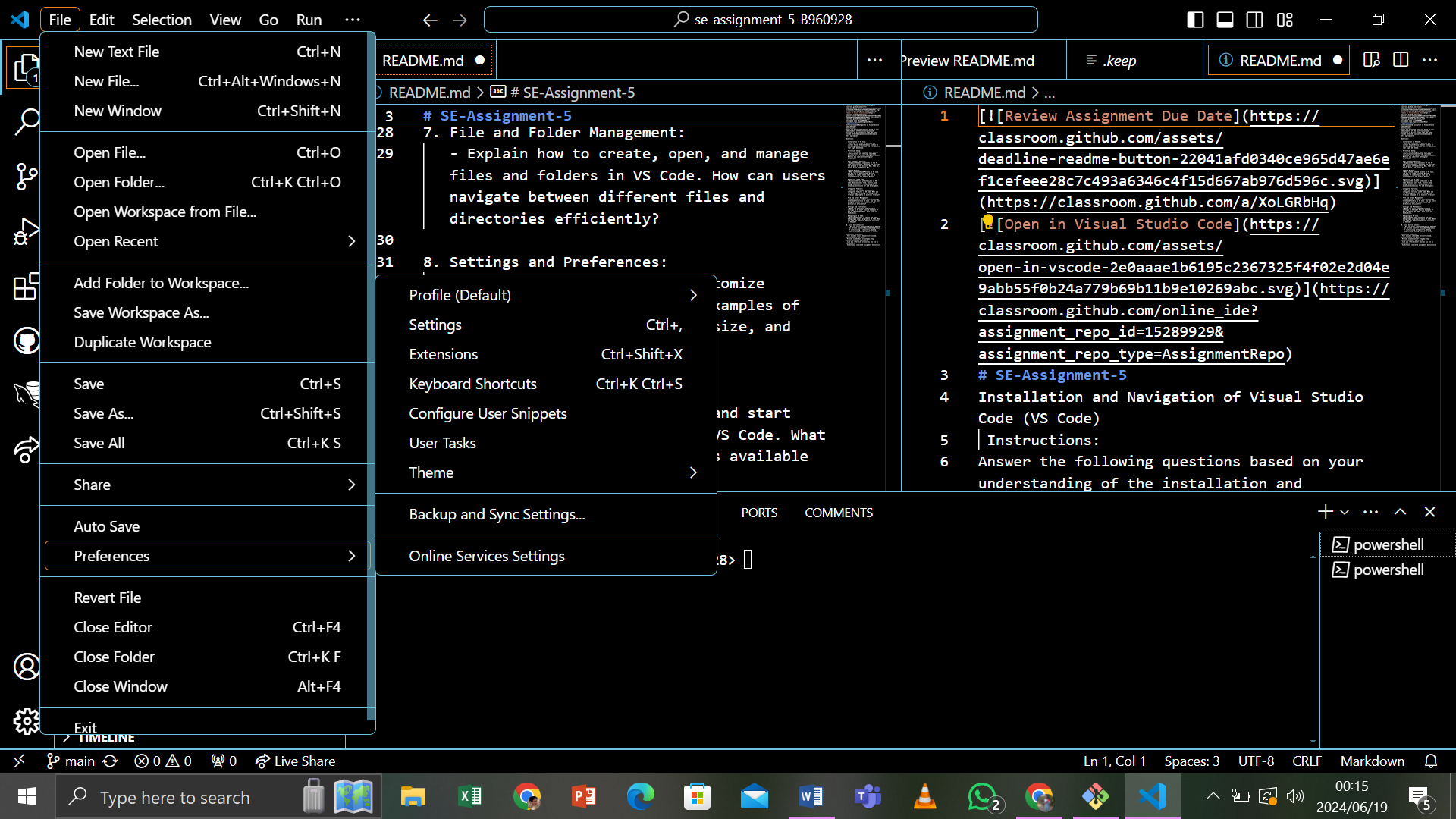
* To quickly open index. Html in your project, press Ctrl+P, type index. Html, and press enter.
* To create a new JavaScript file, right click in the Explorer view, select “New File” and name it script.js.

8. Settings and Preferences

1. Finding and Customizing Settings:

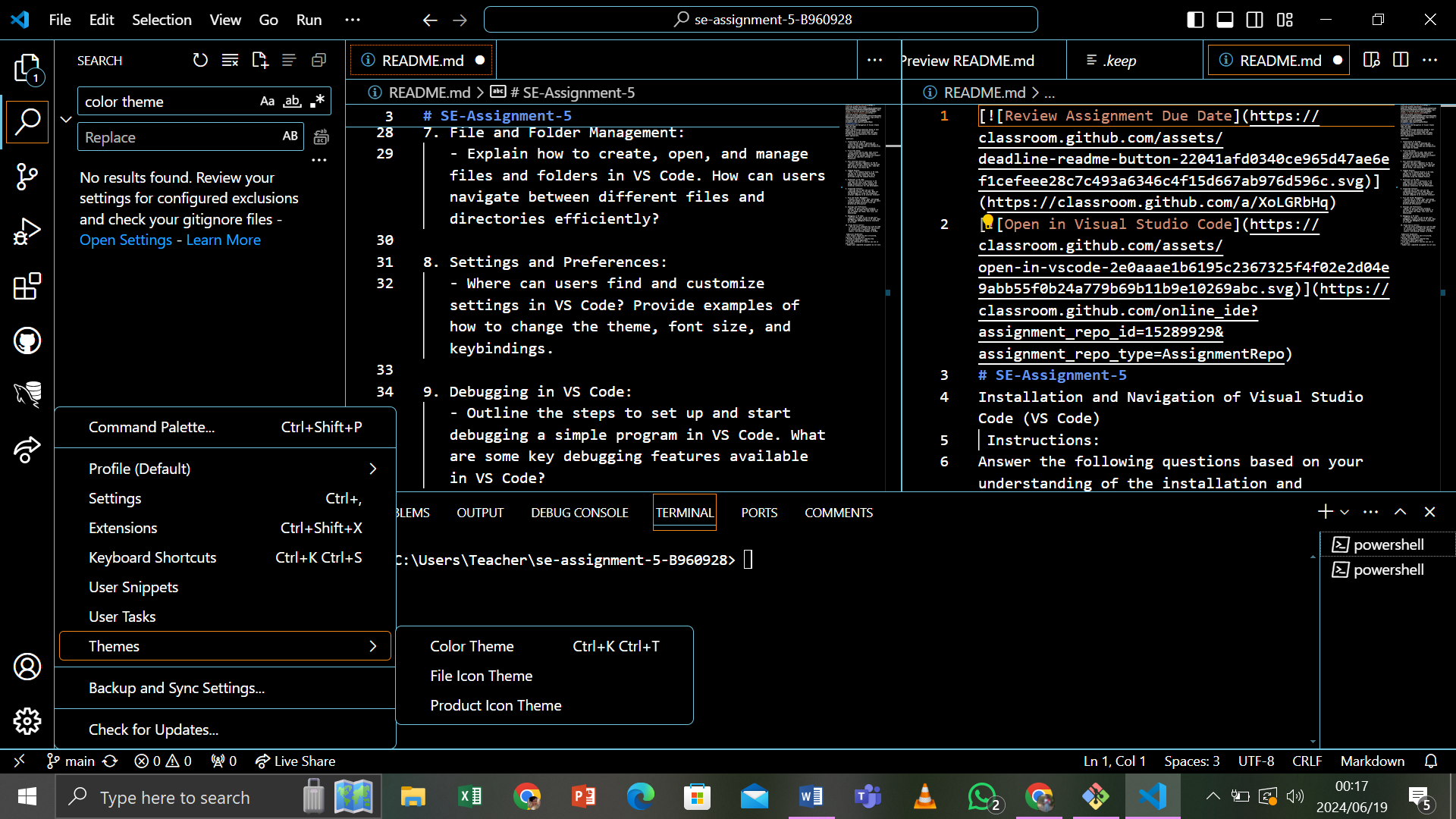
1.1. Accessing Settings:

* Go to file > Preferences > Setting (Ctrl+)



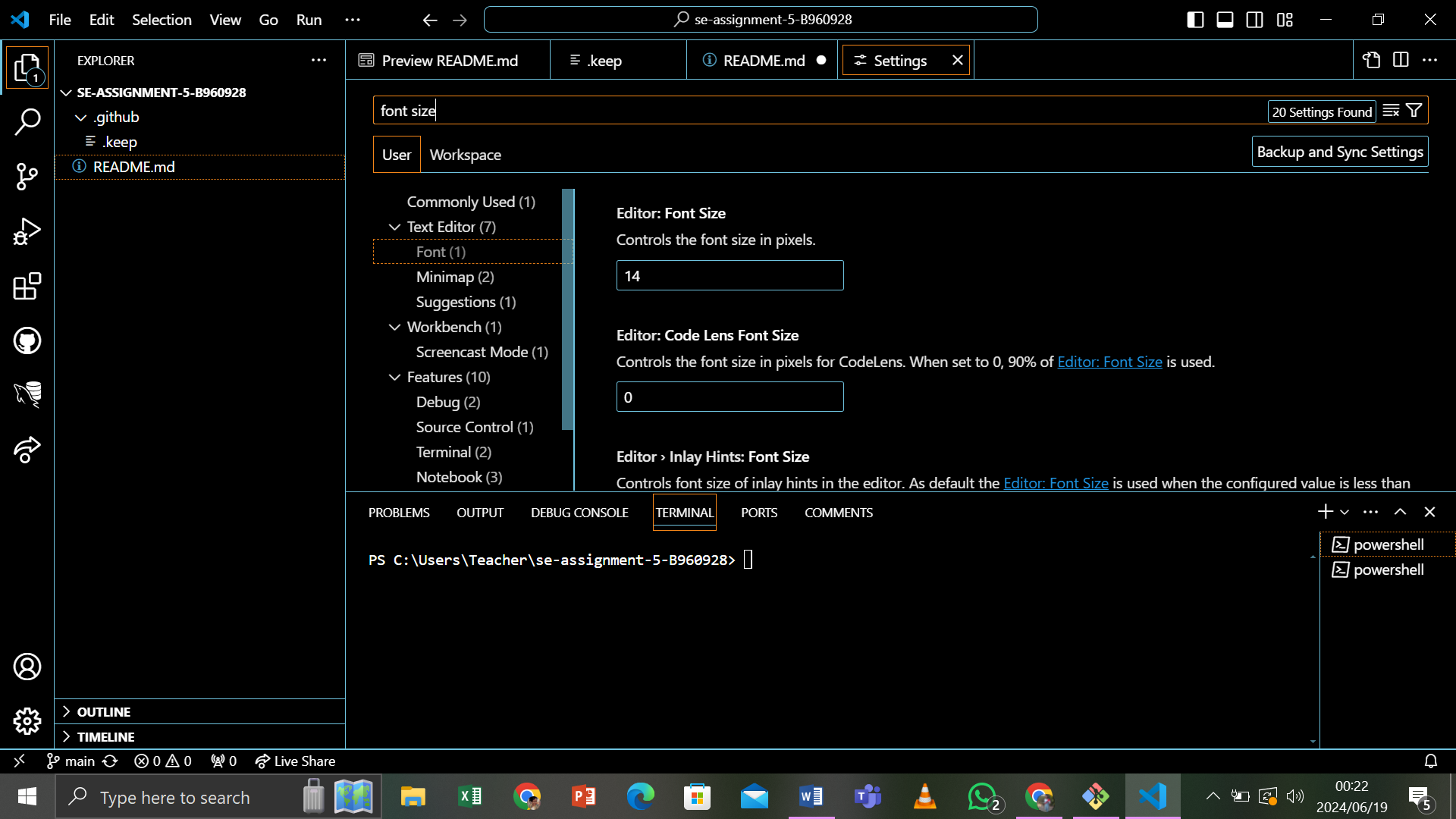
1.2. Changing Theme:

* Setting Icon for “Color Theme” and select the preferred theme



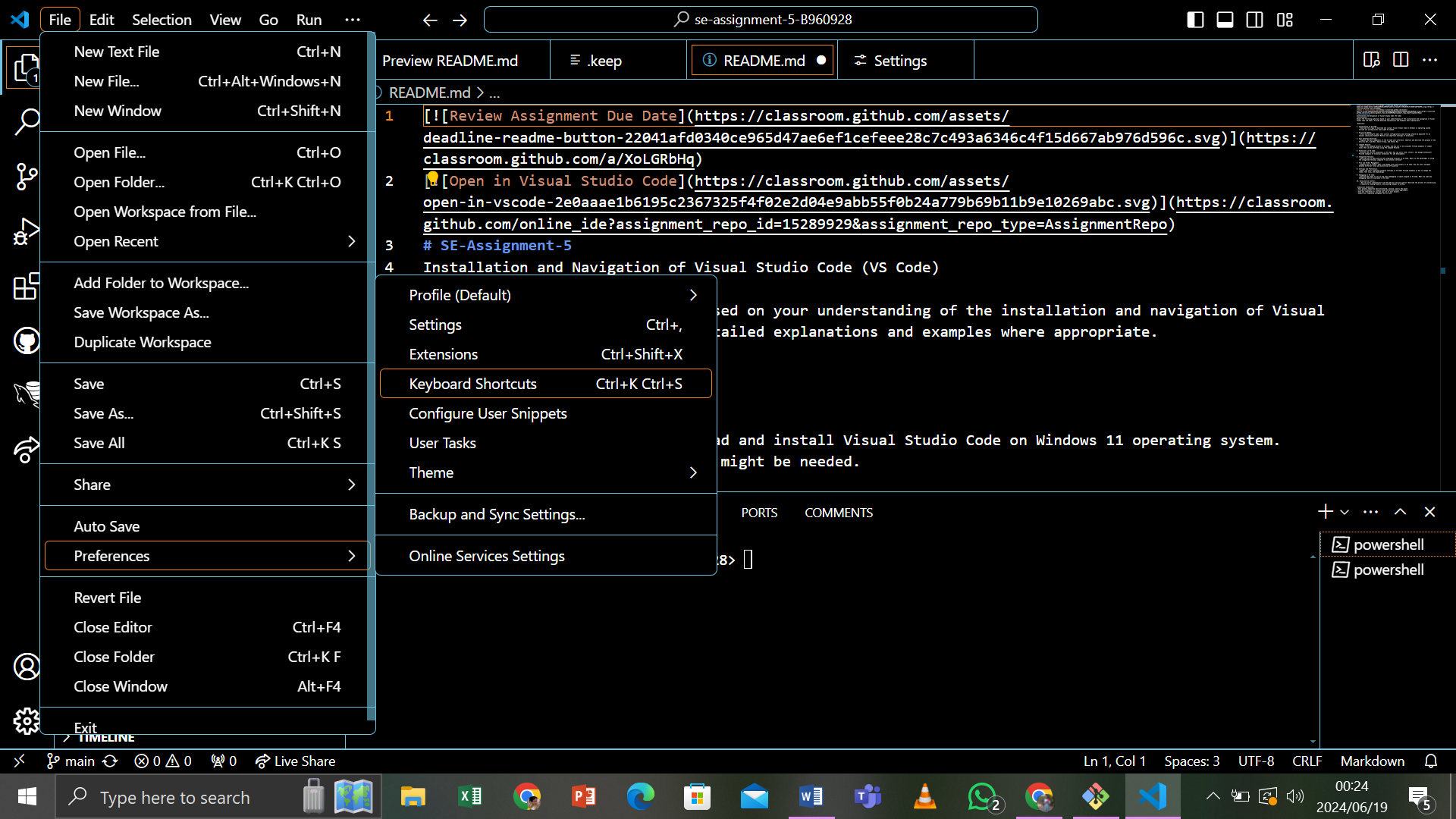
1.3. Adjusting Font Size:

* Search for “Font Size” and set the desired size



1.4. Modifying Keybindings:

* Go to File > Preferences > keyboard shortcut or press (Ctrl+K, Ctrl+S)
* Search for the Command and set new keybindings



Examples:

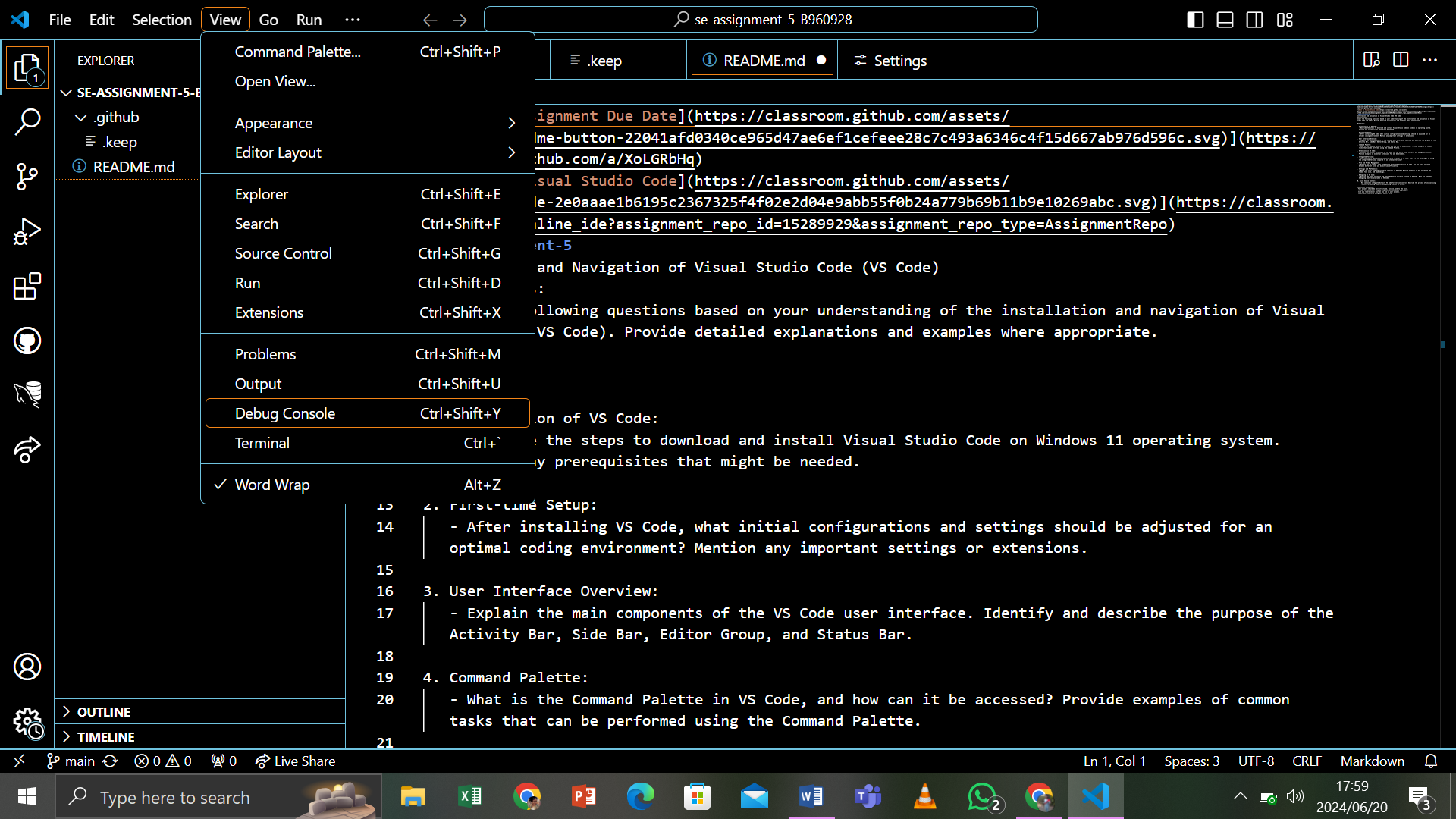
* To set a customer keybiniding or formatting code, go to File > Preferences > keyboard Shortcut, search for “Format Document”, click the pencil icon next to it, and press the desired key combination.

9. Debugging in VS Code

1. Setting up and Starting Debugging:

1.1. Open the Debug View:

* Click on the Debug Icon in the Activity Bar or press Ctrl+Shift+Y



1.2. Configure Debugging:

* Click on the Gear Icon to create a Launch. json file with debugging configurations.

1.3. Set Breakpoints:

* Click in the gutter next to the line number where you want to add a breakpoint.

1.4. Start Debugging:

* Click the green play button in the Debug view or press F5

2. Key Debugging Features:

* Breakpoints: Pause execution at specific lines
* Call Stack: View the call stack and navigate through function calls
* Debug Console: Execute commands and evaluate expressions during debugging
* Watch: Monitor variables and expressions
* Variable: Inspect the current state of variables

Example:

* For debugging a Node.js application, Open the Debug view, configure launch.json with a configuration for Node.js, set breakpoint in your code, and start debugging by pressing F5. Use the debug console to inspect variables and evaluate expressions in real-time.

10. Using Source Control:

1. Integrating Git with VS Code:

1.1. Initializing a Repository:

* Open the Source Control view by clicking the Source Control Icon in the Activity Bar
* Click “Initialize Repository” to create a new Git Repository
* Alternatively, open the terminal and run git initi in your project

1.2. Making Commits:

* Stage changes using the Source Control view or terminal (git add.)
* Commit changes using the Source Control view or terminal (git commit-m)

1.3. Pushing Changes to GitHub:

* Add a remote repository: git remote add origin <repository –URL)
* Push changes: git push –u or origin main

1. Setup Documentation:

* This document with detailed steps and screenshots.

1. GitHub Repository:

* (<https://github.com/B960928/BrightMoima.git>).

1. Reflection:

* Included in the challenges and solutions section.