Software Engineering Assignment 5 – Installation and Navigation of Visual Studio Code (VS Code)

By Izeowayi, Igeh Godwin

Answers:

1. **Steps to Download and Install Visual Studio Code (VS Code) on Windows 11 Operating System:**
   a) Download the Visual Studio Code installer for Windows. You can download the installer from the official Visual Studio website.
   b) Run the installer (VSCodeUserSetup-{version}.exe).
   c) The installer will only take a minute to install all the necessary files.
   d) By default, VS Code will be installed under C:\Users{Username}\AppData\Local\Programs\Microsoft VS Code.
   e) Alternatively, you can also download a Zip archive, extract it and run Code from there.
   f) Once the installation is complete, you can open Visual Studio Code from the console by typing 'code'.

   Prerequisite: Ensure that your system meets the Visual Studio Code system requirements.

2. The initial configurations and settings you can adjust in VS Code for an optimal coding environment are as follows:
   (a) **User Settings:** These are settings that apply globally to any instance of VS Code you open.
   (b) **Workspace Settings:** These are settings stored inside your workspace and only apply when the workspace is opened.
   (c) **Settings Editor:** This allows you to review and change VS Code settings.
   (d) **Settings Groups:** Settings are represented in groups, so that you can navigate to them easily.
   (e) **Modified Settings:** This shows settings you have configured, and their value differs from the default value.
   (f) **Extension Settings:** These are settings contributed by installed VS Code extensions.

   Some important extensions:
   (a) **Languages:** C++, C#, Go, Java, Python
   (b) **Tools:** ESLint, JSHint, PowerShell
   (c) **Debuggers:** PHP XDebug
   (d) **Keymaps:** Vim, Sublime Text, IntelliJ, Emacs, Atom, Brackets, Visual Studio, Eclipse.

3. The main components of the VS Code user interface are:
   (a) **Activity Bar:** Located on the left side of the window, it provides quick access to various aspects of the application, such as:
   - Opening files and folders
   - Managing extensions
   - Accessing settings
   - Debugging tools
   - Version control systems like Git

   (b) **Side Bar:** Adjacent to the Activity Bar, it displays additional information and tools, including:

   - File Explorer: shows the folder structure and files
   - Outline: displays the code structure and symbols
   - Search results
   - Debugging information

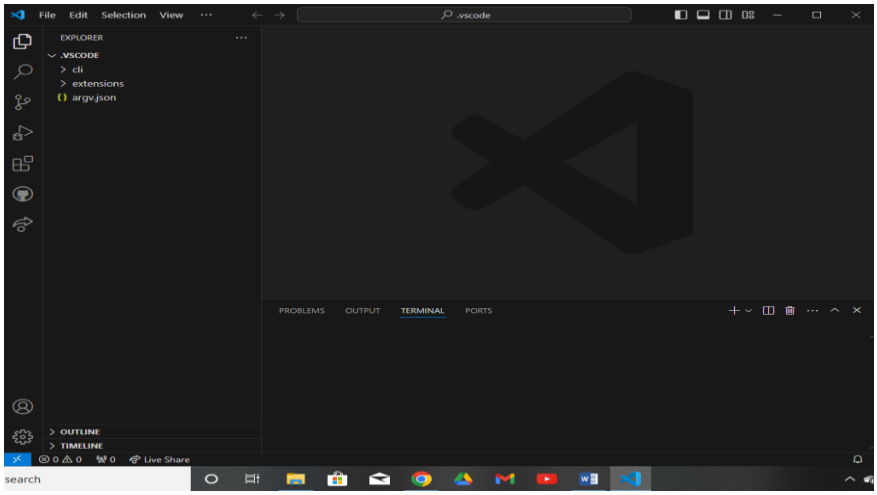
   (c) **Editor Group:** This is the main area where you write and edit code. It consists of:

- Editor: displays the code file
- Tabs: allow multiple files to be open simultaneously
- Editor widgets: provide additional functionality, such as code refactoring or debugging

(d) **Status Bar:** Located at the bottom of the window, it displays:

- Information about the current file, such as its name, language, and encoding
- Error and warning messages
- Debugging information
- Version control status

These components work together to provide a comprehensive and customizable coding environment in VS Code.

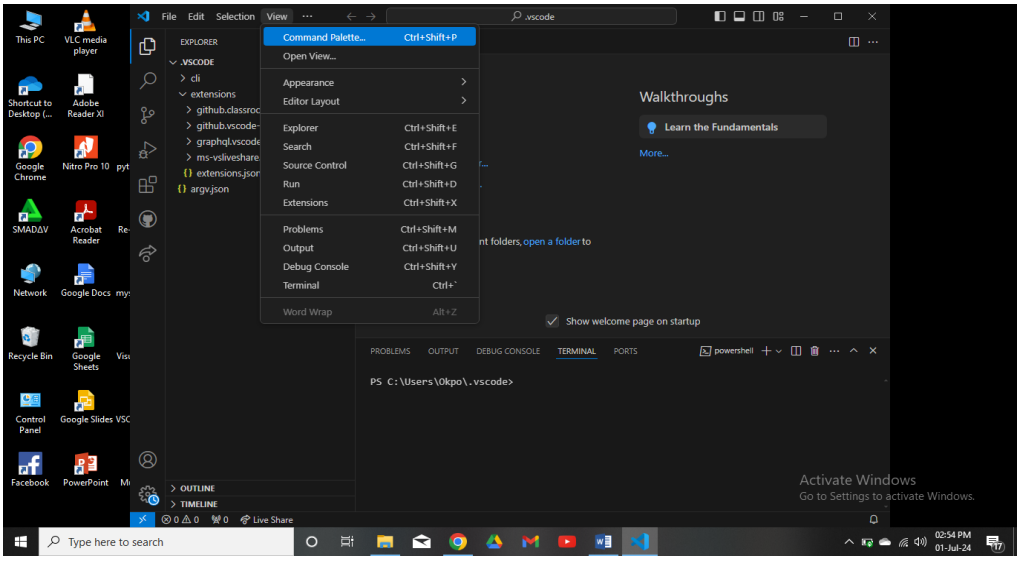


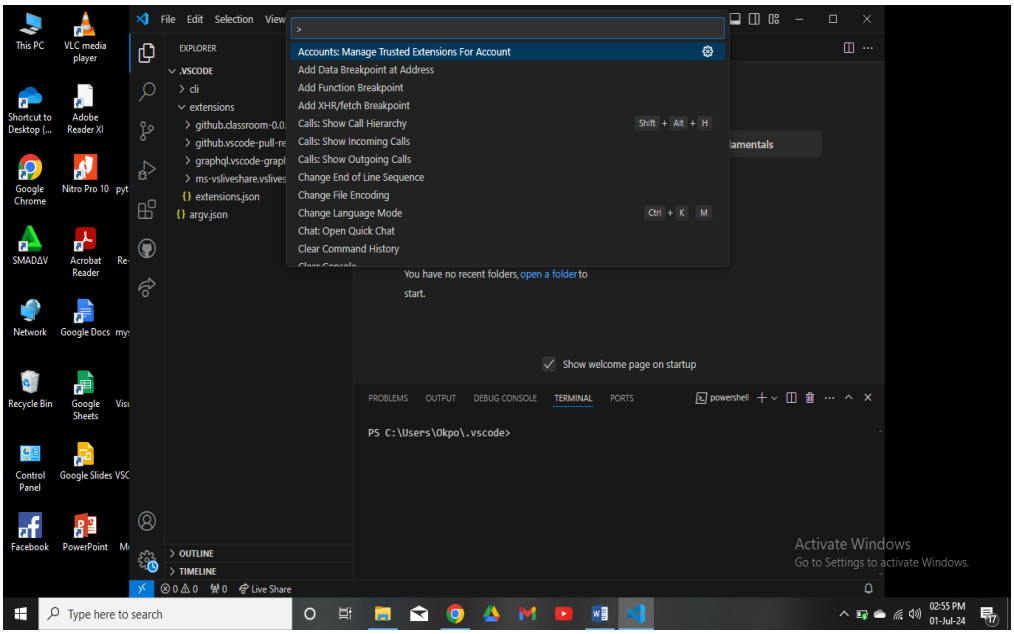
4. The Command Palette is a central hub in VS Code that allows you to access and execute various commands, features, and functionality within the editor. It can be accessed in several ways:
   (a) **Keyboard shortcut:** Press Ctrl + Shift + P (Windows/Linux) or Cmd + Shift + P (Mac) to open the Command Palette.
   (b) **Menu:** Go to View > Command Palette or use the shortcut Ctrl + P (Windows/Linux) or Cmd + P (Mac).
   (c) **Icon:** Click the Command Palette icon in the Activity Bar (looks like a keyboard shortcut symbol).

   Examples of common tasks that can be performed using the Command Palette:

   (a) **Switch editor theme:** Type "Theme" and select a theme to change the editor's appearance.
   (b) **Open a file:** Type the name of a file in your workspace, and select it to open it in the editor.
   (c) **Run a command:** Type "Run Code" to execute a code snippet or a command in the terminal.
   (d) **Debugging:** Type "Debug" to access debugging commands, such as "Start Debugging" or "Step Over".
   (e) **Format code:** Type "Format Document" to format the entire code file or "Format Selection" to format a selected code block.
   (f) **Search:** Type a search query to search for text in your codebase.
   (g) **Extensions:** Type "Extensions" to manage and install extensions.
   (h) **Settings:** Type "Settings" to access and modify VS Code settings.

   The Command Palette is a powerful tool that provides quick access to various features and functionality in VS Code, allowing you to work more efficiently and effectively.





5. Extensions play a crucial role in VS Code, enhancing its functionality and capabilities. They can:
   (a) Provide additional features and tools
   (b) Support new programming languages
   (c) Enhance code editing and debugging experiences
   (d) Integrate with version control systems
   (e) Offer code snippets and templates

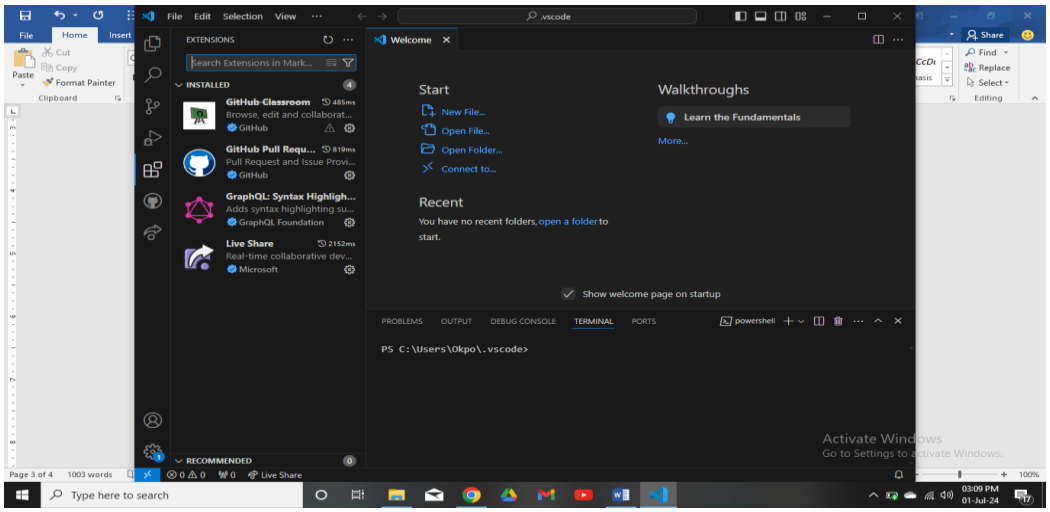
   Users can find, install, and manage extensions in the following ways:
   a) **Extensions Marketplace:** Accessible within VS Code, it allows users to search, browse, and install extensions.
   b) **VS Code Extensions Website:** A website that lists all available extensions, with filters and search functionality.
   c) **Install from VSIX:** Users can install extensions manually by downloading a VSIX file and installing it in VS Code.

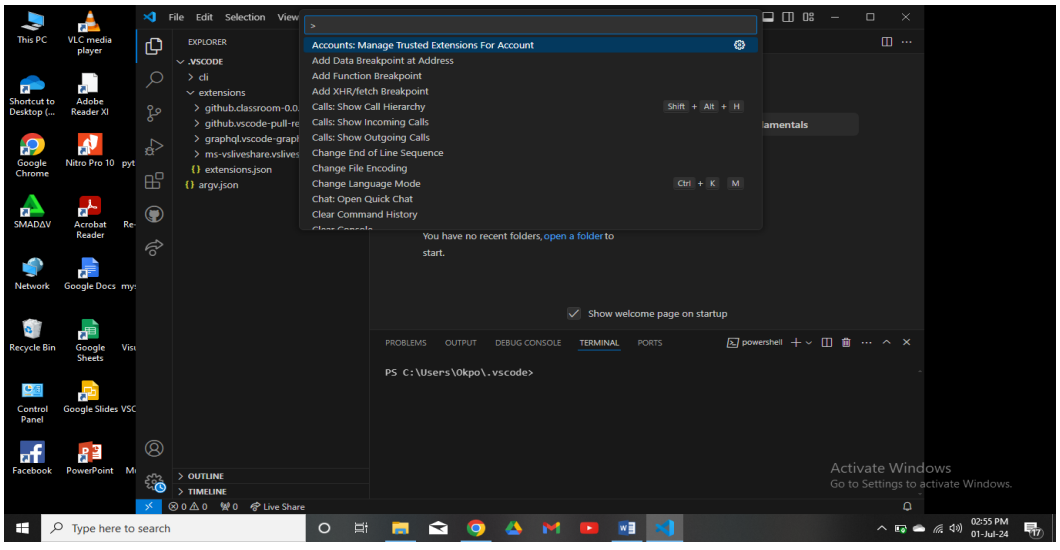
To manage extensions, the tools below would help:
- **Extensions View:** Accessible through the Activity Bar or by pressing Ctrl + Shift + X (Windows/Linux) or Cmd + Shift + X (Mac).
- **Extension Settings:** Allow users to configure extension settings.

Essential extensions for web development:

a) **HTML Snippets:** Provides HTML code snippets and templates.
b) **CSS Intellisense:** Offers CSS code completion, hover information, and formatting.
c) **JavaScript (ES6) code snippets:** Provides JavaScript code snippets and templates.
d) **ESLint:** Integrates ESLint into VS Code for code linting and formatting.
e) **Debugger for Chrome:** Allows debugging of JavaScript code in Chrome.
f) **Live Server:** Launches a local development server for static and dynamic websites.
g) **Path Intellisense:** Provides path completion for imports and links.
h) **Prettier:** Formats code according to predefined styles.
i) **GitHub Pull Requests:** Allows management of GitHub pull requests within VS Code.
j) **Azure Extensions:** Provides tools for Azure development, deployment, and management.

These extensions enhance the web development experience in VS Code, offering features like code completion, debugging, and formatting, making development more efficient and productive.

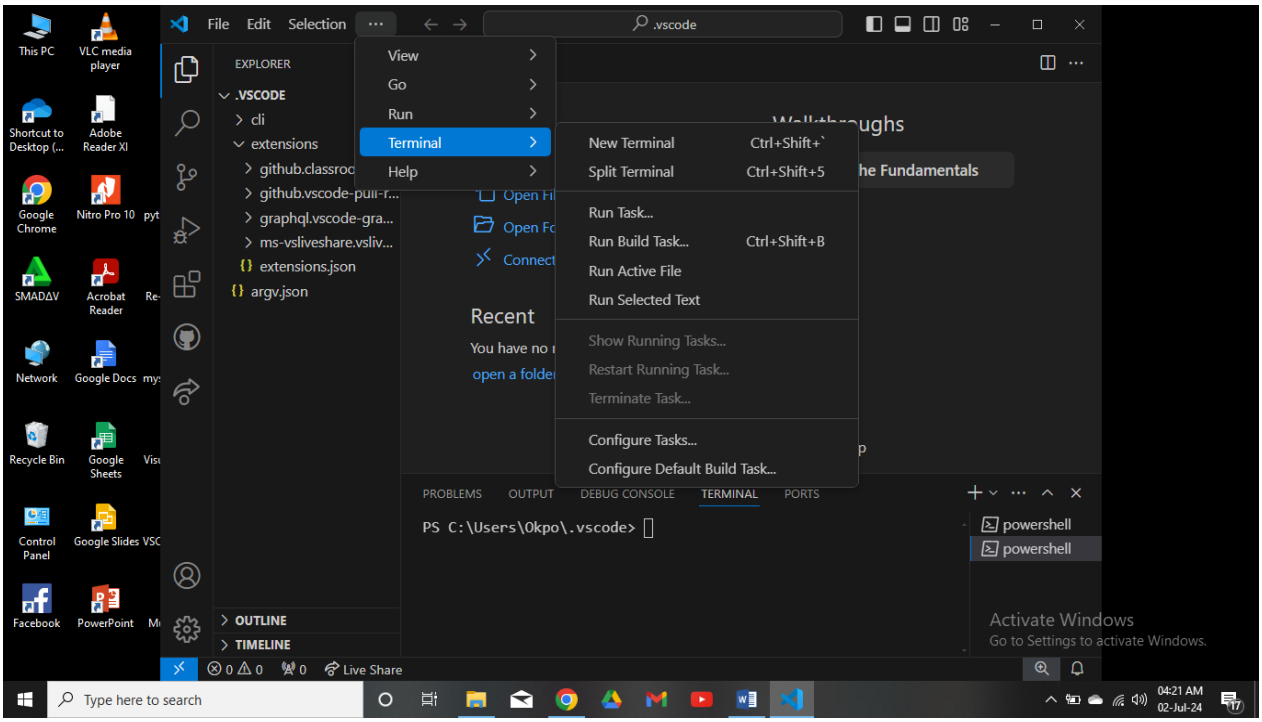




6. To open the integrated terminal in VS Code:
   (a) Press Ctrl + (Windows/Linux) or Cmd + (Mac) to toggle the terminal.
   (b) Alternatively, use the menu: View > Terminal or Terminal > New Terminal.
   (c) You can also use the shortcut Ctrl + Shift + * (Windows/Linux) or *Cmd + Shift + (Mac) to create a new terminal.

   Advantages of using the integrated terminal:
   (a) **Convenience:** No need to switch between applications or windows.
   (b) **Context:** The terminal is aware of the current file and folder, making it easier to execute commands.
   (c) **Integration:** VS Code extensions can interact with the terminal, enhancing the development experience.
   (d) **Multi-terminal support:** You can have multiple terminals open, each with its own context.
   (e) **Better navigation:** Use VS Code's navigation features, like Ctrl + Click, to navigate to files and folders from the terminal.
   (f) **Improved debugging:** The terminal is integrated with the debugger, making it easier to inspect and debug code.
   (g) **Enhanced productivity:** With the terminal integrated, you can stay focused on your code and avoid context switching.

   Compared to an external terminal, the integrated terminal offers a more streamlined and efficient development experience, allowing you to stay within VS Code and leverage its features and extensions.



7. In VS Code, you can create, open, and manage files and folders using the following methods:
   (a) **Creating Files and Folders:**
   - **File:** Press Ctrl + N (Windows/Linux) or Cmd + N (Mac) to create a new file.
   - **Folder:** Press Ctrl + Shift + N (Windows/Linux) or Cmd + Shift + N (Mac) to create a new folder.

   (b) **Opening Files and Folders:**
   - **File:** Double-click a file in the Explorer panel or press Ctrl + O (Windows/Linux) or Cmd + O (Mac) to open a file.
   - **Folder:** Double-click a folder in the Explorer panel or press Ctrl + K (Windows/Linux) or Cmd + K (Mac) to open a folder.

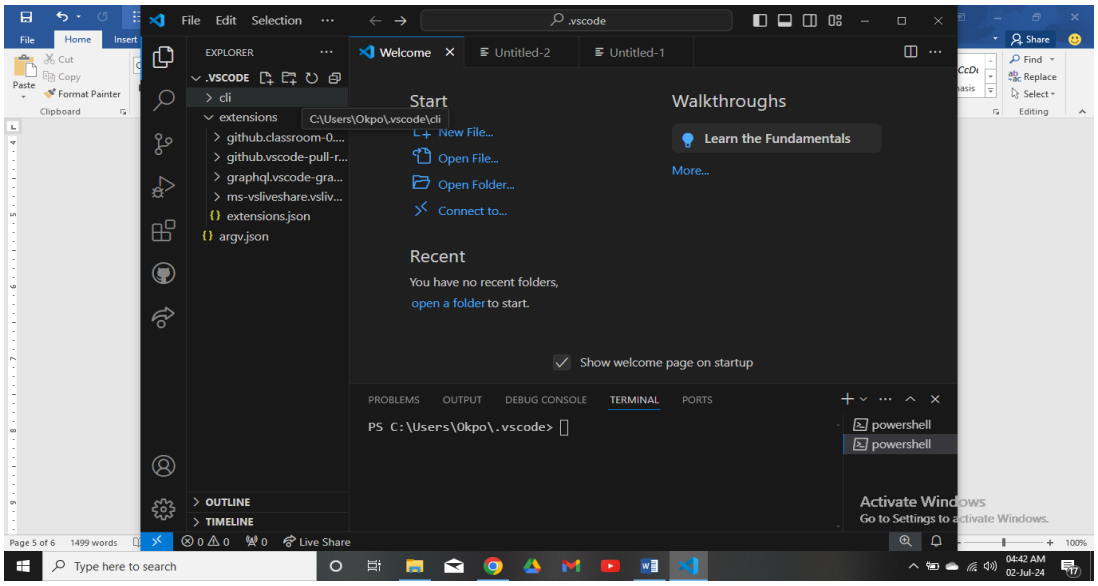
(c) **Managing Files and Folders:**

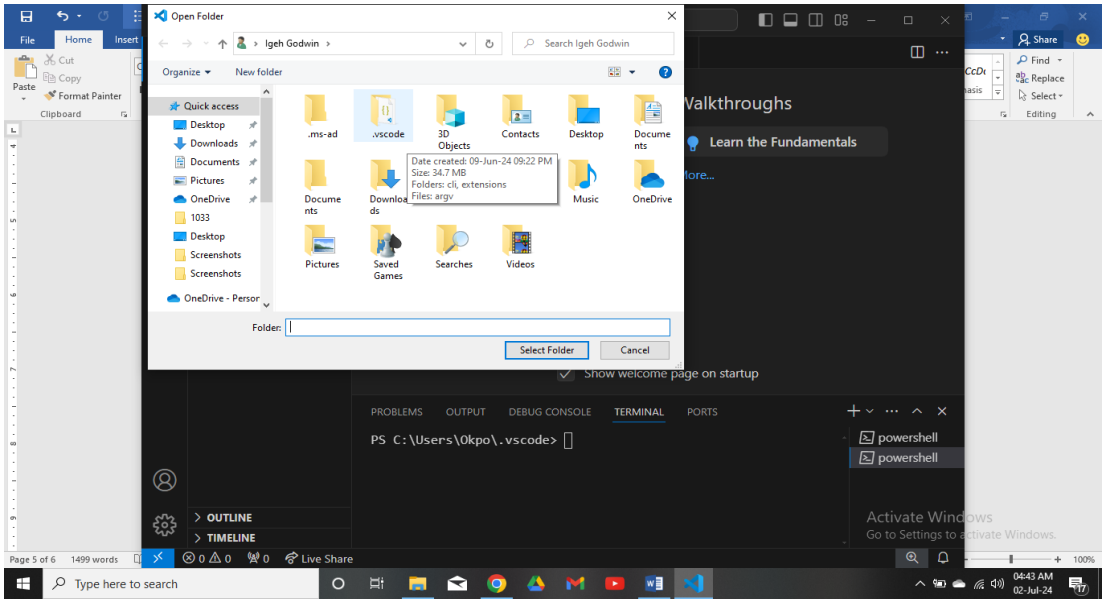
- **Rename:** Click and drag to rename a file or folder.
- **Delete:** Right-click and select "Delete" or press Del (Windows/Linux) or Cmd + Backspace (Mac).
- **Move:** Drag and drop files or folders to move them.

(d) **Navigating between Files and Directories:**

- **Explorer Panel:** Use the Explorer panel to navigate through your project's file structure.
- **Breadcrumbs:** Click on the breadcrumbs in the top-left corner of the editor to navigate through the file path.
- **Go to File:** Press Ctrl + P (Windows/Linux) or Cmd + P (Mac) to search and open a file.
- **Go to Symbol:** Press Ctrl + Shift + O (Windows/Linux) or Cmd + Shift + O (Mac) to search and navigate to a symbol (function, variable, etc.).
- **Navigation History:** Press Ctrl + Tab (Windows/Linux) or Cmd + Tab (Mac) to navigate through your recent files.

By using these methods, you can efficiently create, open, and manage files and folders in VS Code, and navigate between them with ease.





8. Users can find and customize settings in VS Code in the following ways:
   (a) **Settings Editor:** Press Ctrl + , (Windows/Linux) or Cmd + , (Mac) to open the Settings Editor.
   (b) **Settings File:** Open the settings file directly by pressing Ctrl + Shift + P (Windows/Linux) or Cmd + Shift + P (Mac) and searching for "Open Settings (JSON)".
   (c) **Command Palette:** Type "Settings" in the Command Palette (Ctrl + Shift + P or Cmd + Shift + P) to access various settings-related commands.

   Examples of customizations:

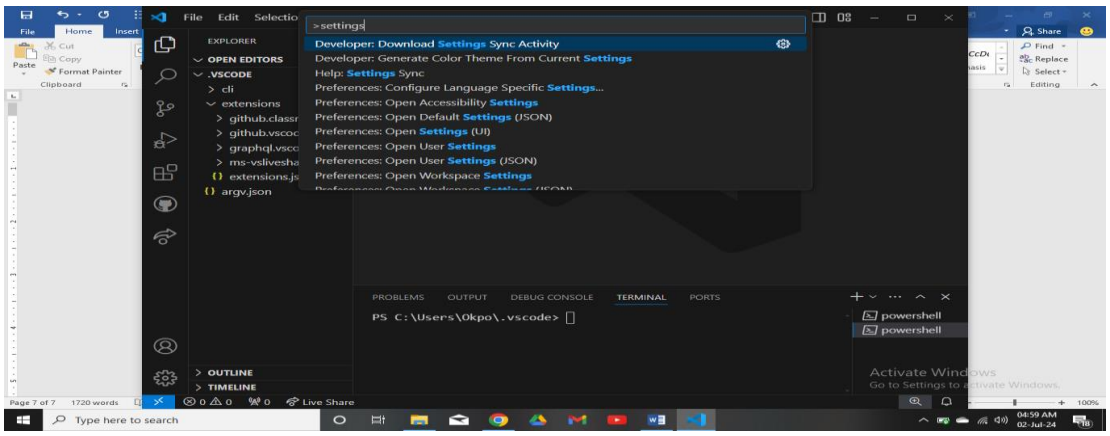
(a) **Change Theme:**
   - In the Settings Editor, search for "theme" and select a theme from the dropdown list.
   - Alternatively, add a line like "theme": "Dark" to the settings file.

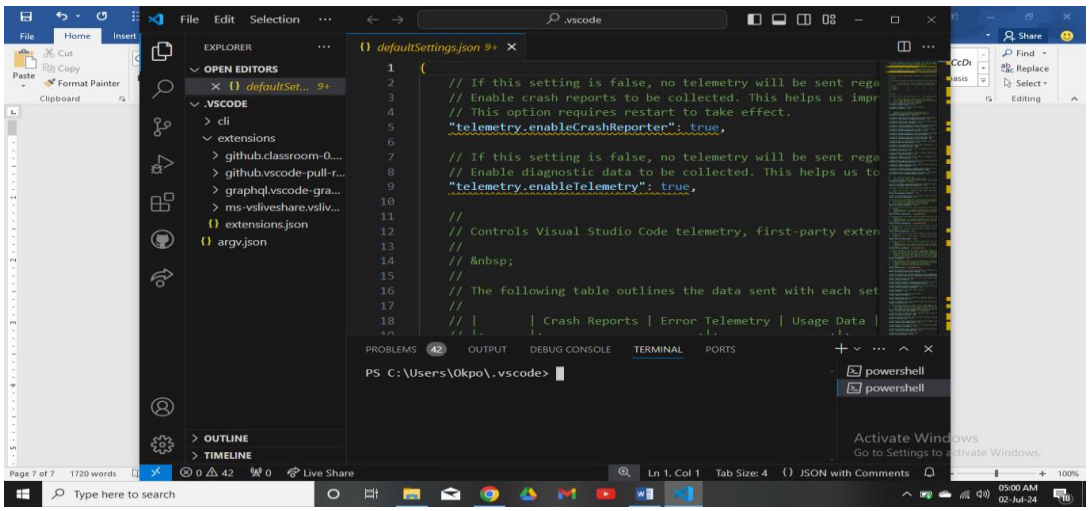
(b) **Change Font Size:**
   - In the Settings Editor, search for "font size" and adjust the value.
   - Alternatively, add a line like "editor.fontSize": 18 to the settings file.

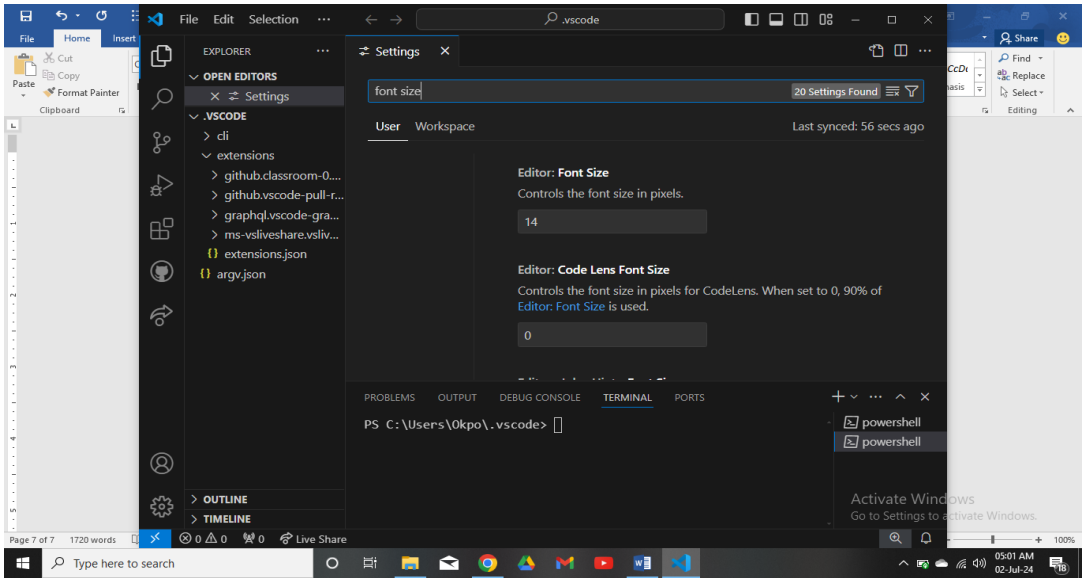
(c) **Change Keybindings:**
   - In the Settings Editor, search for "keybindings" and modify or add new keybindings.
   - Alternatively, add lines like "keybindings": [{"key": "ctrl+shift+f", "command": "editor.formatAll"}] to the settings file.

   Note: Changes in the settings file require a reload of VS Code (Ctrl + R or Cmd + R) to take effect.

   These examples demonstrate how users can access and customize various settings in VS Code to personalize their development environment.





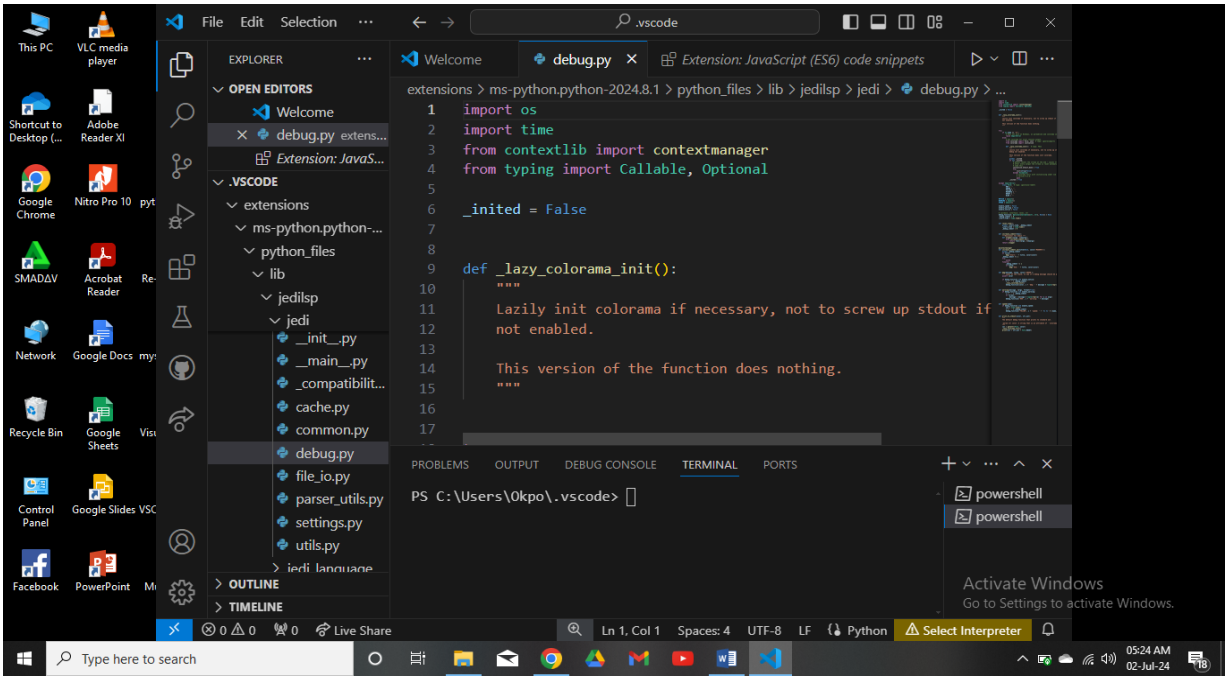


9. Below are the steps to set up and start debugging a simple program in VS Code:
   (a) **Create a new project:** Create a new folder for your project and add your code files to it.
   (b) **Open the project in VS Code:** Open the project folder in VS Code.
   (c) **Install the debugger extension:** Install the appropriate debugger extension for your programming language (e.g., Python, JavaScript, C++).
   (d) **Configure the debugger:** Configure the debugger settings in the launch.json file (automatically created by VS Code).
   (e) **Set breakpoints:** Set breakpoints in your code by clicking on the line number or using the keyboard shortcut F9.
   (f) **Start debugging:** Press F5 or use the "Debug" icon in the Activity Bar to start debugging.
   (g) **Step through code:** Use the debugging controls (Step Over, Step Into, Step Out) to step through your code.

   Some key debugging features available in VS Code include:

   (a) **Breakpoints:** Set breakpoints to pause execution at specific points in your code.
   (b) **Step through code:** Step through your code line by line to examine variables and behavior.
   (c) **Variable inspection:** Inspect variable values and expressions in the "Variables" panel.
   (d) **Call stack:** View the call stack to see the sequence of function calls leading to the current point.
   (e) **Debug console:** Interact with your code using the debug console.
   (f) **Conditional breakpoints:** Set breakpoints that only trigger when specific conditions are met.
   (g) **Logpoints:** Set logpoints to log messages to the console without pausing execution.
   (h) **Profiler:** Use the built-in profiler to analyze performance and identify bottlenecks.

   These features help you efficiently identify and fix errors, making VS Code a powerful debugging tool.



10. Users can integrate Git with VS Code for version control by following these steps:
    (a) **Install the Git extension:** Install the Git extension for VS Code from the Extensions marketplace.
    (b) **Initialize a repository:** Open the project folder in VS Code and initialize a Git repository by running the command "Git: Initialize Repository" from the Command Palette (Ctrl+Shift+P).
    (c) **Configure Git settings:** Configure Git settings, such as setting the author name and email, by running the command "Git: Configure Git Settings" from the Command Palette.
    (d) **Make changes:** Make changes to your code files.
    (e) **Stage changes:** Stage changes by selecting the files and using the "Git: Stage Changes" command from the Command Palette or by using the "git add" command in the terminal.
    (f) **Make commits:** Make commits by using the "Git: Commit Changes" command from the Command Palette or by using the "git commit" command in the terminal.
    (g) **Push changes to GitHub:** Push changes to GitHub by using the "Git: Push Changes" command from the Command Palette or by using the "git push" command in the terminal.

    Here are the detailed steps:
    (a) Open the project folder in VS Code.
    (b) Initialize a Git repository by running the command "Git: Initialize Repository" from the Command Palette (Ctrl+Shift+P).
    (c) Configure Git settings by running the command "Git: Configure Git Settings" from the Command Palette.
    (d) Make changes to your code files.
    (e) Stage changes by selecting the files and using the "Git: Stage Changes" command from the Command Palette or by using the "git add" command in the terminal.
    (f) Make commits by using the "Git: Commit Changes" command from the Command Palette or by using the "git commit" command in the terminal.
    (g) Link your GitHub account to VS Code by running the command "Git: Login to GitHub" from the Command Palette.
    (h) Push changes to GitHub by using the "Git: Push Changes" command from the Command Palette or by using the "git push" command in the terminal.

    Note: You can also use the Git Panel in VS Code to perform these actions.