SE-ASSIGNMENT-5

Installation and navigation of visual studio code.

1.INSTALLATION OF VS CODE.

* Ensure your system meets the basic requirements that is :

-windows 11 OS, administrative privileges to install software, stable internet connection.

* Visit the visual studio code website:

- On the homepage, you’ll see a download button that automatically detects your operating system. Click the "Download for Windows" button.

-This will download the VSCodeSetup.exe file.

* Run the installer:

-locate the VScodeSetup.exe file double click to run the installer

-accept the license agreement and click next

-leave the destination location on its defaults and click next

-you can choose to create a desktop icon and other context menu options.

- click install to start the installation process.

2.FIRST-TIME SETUP.

* Initial configurations

-Theme and appearances. Choose a theme that suits your preference. Go to File > Preferences > Color Theme or press Ctrl+K followed by Ctrl+T and select your desired theme

-Tab size and formatting. search for tab size and format on save in setting and configure them accordingly.

-Auto-save. Enable auto-save to avoid losing work. Go to File > Preferences > Settings, search for "Auto Save," and set it to "afterDelay" or your preferred option.

* Important extensions

-language support .install them from the Extensions view by searching the languages you ant to use eg python, javascript,

-development tools- debugger for chrome allowas you to debug javascript code running in google chrome directly from vs code.

3.USER INTERFACE OVERVIEW.

Here are the main components of the vs code user interface:

* Activity bar:

Located on the left side of the VS code window

It provides quick access to different views and major functions. Contains icons for explorer, search, source control, run and debug and extensions.

* Side bar:

Located adjacent to the activity bar on the left side of the VS code window

It displays the content and tools associated with the selected activity depending on the selected icon in the activity bar.

* Editor group:

Located at the central area of the vs code window

This is the main area were you edit your code it supports multiple editor groups (tabs) to help you manage multiple files at once.

* Status bar:

Located at the bottom of the VS code window.

Displays information about the current state of the editor and workspace including encoding indicates the character encoding of the current file, git status indicates the current git branch and sync status.

4.COMMAND PALETTE.

It is a powerful feature that provides quick access to a wide range of commands and functionalities within the editor. It allows you to perform tasks without needing to navigate through menus or remember complex keyboard shortcuts.

Can be accessed through pressing  Ctrl+Shift+P (Windows/Linux) or Cmd+Shift+P (Mac) to open the Command Palette.

Common tasks using the command palette:

* Changing the color themes:

Type themes and select preferences: color theme and choose a theme for the editor

* Installing extensions:

Type install select extensions: install extensions to open the Extensions view where you can search for and install new extensions.

* Opening settings:

Type setting and select preferences: open settings to open the settings in the user interface.

* Running commands:

Type run and select tasks: run task to run predefined tasks from your tasks.json file

Type debug and select debug: start debugging to start debugging session.

* File operation:

Type >New and select File: New File to create a new file.

Type >Open and select File: Open File to open an existing file.

Type >Save and select File: Save to save the current file.

* Version control:

Type git and select commands like git: clone, git: commit or git:push to perform version control operations.

5.EXTENTIONS IN VS CODE.

How to find, install and manage extensions:

Click on the extensions icon in the activity bar on the side of the window.

Then search desired extension and click the install button.

Extensions can be managed by enabling and disabling them, ensuring they are updated regularly, uninstalling extensions.

Essential extensions for web development.

* HTML, CSS and JavaScript support
* Code formatting and linting e.g prettier-code formatter.
* Frameworks and libraries
* Version control.

6.INTERGRATED TERMINAL.

To open the integrated terminal go to view > terminal from the top of the menu

Using the integrated terminal:

* Opening multiple terminals. Click the +icon in the terminal panel to open multiple terminal instances.
* Splitting terminal. Click on the split terminal icon to create a new terminal instance in a split view.
* Running commands. Type commands as you would in any terminal and press enter to execute them.
* Customize the terminal.

Advantages of using the integrated terminal compared to an external terminal:

* Convenience and efficiency- the integrated terminal allows you to run commands without leaving the VS code environment, streamlining your workflow.
* Context awareness- integrated terminal opens in the context of the current workspace automatically setting the working directory to your projects root folder.
* Unified environment-Having the terminal within the same window as your code editor allows for a more organized and unified workspace. You can see terminal output alongside your code, making it easier to debug and test.
* Integrated features-The integrated terminal supports features like hyperlinks, which allow you to click on URLs directly in the terminal output.
* Accessibility- The integrated terminal is readily accessible through keyboard shortcuts, making it quick and easy to open and use without interrupting your workflow.

7.FILE AND FOLDER MANAGEMENT.

Creating files and folders-Click the New File icon (a piece of paper with a plus sign) in the File Explorer sidebar. Click the New Folder icon (a folder with a plus sign) in the File Explorer sidebar.

To open the file Go to File > Open File or File > Open Folder, browse to the file or folder and select it.

Managing files and folders:

* Renaming files and folders- right-click the file or folder in the file explorer and select rename, type new name and press enter.
* Move files and folders- drag and drop file or folder within the file explorer to move them to a different location.
* Delete files and folders- right-click the file or folder in the file explorer and select delete , confirm the deletion.

Navigating between files and directories efficiently use quick open, breadcrumb navigation or explorer shortcuts.

8.SETTING AND PREFERENCES

Using the menu Go to File > Preferences > Settings (Windows/Linux) or Code > Preferences > Settings (Mac).

Example of customizing settings:

* Changing theme- open settings UI type color theme click on it and select your preferred theme.
* Changing font size- open settings in the search bar type font size under text editor and find fomt size and select your desired value
* Changing key bindings using key bindings UI Go to File > Preferences > Keyboard Shortcuts (Windows/Linux) or Code > Preferences > Keyboard Shortcuts (Mac).

9.DEBUGGING IN VS CODE.

Install Necessary Extensions:

* Make sure you have the appropriate language extension installed. For example, if you are debugging a Python program, install the Python extension. For JavaScript/Node.js, install the Node.js extension.

Open the Project Folder:

* Open your project folder in VS Code by selecting File > Open Folder and choosing the folder containing your project files.

Open the Source File:

* Open the source file you want to debug by clicking on it in the File Explorer.

Add Breakpoints:

* Click in the left margin next to the line number where you want to set a breakpoint. A red dot will appear indicating the breakpoint.

Create a Launch Configuration:

* Open the Run and Debug view by clicking the Run icon in the Activity Bar on the side of the window or by pressing Ctrl+Shift+D.
* Click on create a launch.json file to create a new configuration file. VS Code will prompt you to select the environment (e.g., Node.js, Python, etc.).
* A launch.json file will be created in the .vscode directory in your project folder. This file contains configurations for different debugging scenarios.

Configure the Debugger:

* Customize the launch.json file if needed

Start Debugging:

* Press F5 to start debugging. The debugger will launch your application, and execution will stop at any breakpoints you have set.

Key Debugging Features in VS Code

Breakpoints:

* + Set breakpoints by clicking in the left margin next to the line numbers.
  + Conditional breakpoints can be set by right-clicking on a breakpoint and selecting "Edit Breakpoint". You can then enter an expression that must evaluate to true for the breakpoint to trigger.

Watch Expressions:

* + In the Run and Debug view, you can add watch expressions to monitor the value of variables and expressions. Click the + button in the WATCH section and enter the expression you want to watch.

Call Stack:

* + The CALL STACK section shows the stack of function calls leading up to the current breakpoint. You can click on any frame in the stack to view the corresponding source code.

Variables:

* + The VARIABLES section displays the current value of all variables in the current scope. You can inspect and expand objects to see their properties.

Debug Console:

* + The Debug Console allows you to evaluate expressions and interact with your application while it's running. You can access it from the Run and Debug view or by pressing `Ctrl+`` (backtick).

Step Through Code:

* + Use the toolbar buttons or keyboard shortcuts to control execution:
    - Continue (F5): Resume execution until the next breakpoint.
    - Step Over (F10): Execute the next line of code, stepping over function calls.
    - Step Into (F11): Step into a function call.
    - Step Out (Shift+F11): Step out of the current function.

10.USING SOURCE CONTROL.

#### Initializing a Git Repository

1. **Open Your Project Folder:**
   * Open your project folder in VS Code by selecting File > Open Folder and choosing the folder containing your project files.
2. **Initialize a Repository:**
   * Open the Source Control view by clicking the Source Control icon in the Activity Bar on the side of the window or by pressing Ctrl+Shift+G.
   * If your folder is not already a Git repository, you’ll see an option to initialize a repository. Click Initialize Repository.
   * This will create a .git folder in your project directory, which Git uses to track your changes.

#### Making Commits

1. **Stage Changes:**
   * Any changes made to your files will appear in the Source Control view under the Changes section.
   * To stage a change, click the + icon next to the file. You can also stage all changes by clicking the + icon at the top of the Changes section.
2. **Commit Changes:**
   * Once your changes are staged, they will appear in the Staged Changes section.
   * Enter a commit message in the message box above the Staged Changes section.
   * Click the checkmark icon to commit the changes. You can also commit using the Command Palette (Ctrl+Shift+P) by typing >Git: Commit Staged and pressing Enter.

#### Pushing Changes to GitHub

1. **Set Up Remote Repository:**
   * If you haven’t already set up a remote repository on GitHub, go to GitHub and create a new repository.
   * Copy the repository URL.
2. **Add Remote Repository:**
   * In VS Code, open the integrated terminal by pressing Ctrl+`` (backtick) or selecting View>Terminal`.
   * Add the remote repository by running the following command, replacing <repository-url> with the URL you copied from GitHub:

sh

Copy code

git remote add origin <repository-url>

1. **Push Changes:**
   * To push your committed changes to the remote repository on GitHub, run the following command in the terminal:

sh

Copy code

git push -u origin main

* + If your branch is not named main, replace main with the name of your branch.

#### Additional Git Features in VS Code

1. **Branch Management:**
   * You can create, switch, and delete branches directly from VS Code. Click on the current branch name in the bottom left corner of the window to open the branch menu.
2. **Pulling Changes:**
   * To pull changes from the remote repository, open the Command Palette (Ctrl+Shift+P), type >Git: Pull, and press Enter.
3. **Viewing History:**
   * You can view the commit history by clicking on the ... menu in the Source Control view and selecting View History.
4. **Merging and Resolving Conflicts:**
   * VS Code provides a user-friendly interface for merging branches and resolving conflicts. Conflicting changes will be highlighted, and you can choose which changes to keep directly in the editor.