SE-Assignment-5

1. Installation of VS Code:

Installation of VS Code on Windows 11

Steps:

1. **Download Visual Studio Code Installer:**
   - Open a web browser and go to the Visual Studio Code download page.
   - Click "Download for Windows" to get the installer.
2. **Run the Installer:**
   - Locate the downloaded file (e.g., `VSCodeSetup.exe`) in the "Downloads" folder.
   - Double-click the installer to run it.
3. **Setup Wizard:**
   - Click "Next" on the welcome screen.
   - Accept the license agreement and click "Next".
4. **Select Installation Location:**
   - Choose the default or preferred location and click "Next".
5. **Select Additional Tasks:**
   - Recommended options:
     - Create a desktop icon
     - Add "Open with Code" actions to the Windows Explorer context menu
     - Register Code as an editor for supported file types
     - Add to PATH
   - Click "Next".
6. **Install:**
   - Click "Install" to begin the process.
   - Wait for the installation to complete, then click "Finish" to launch Visual Studio Code.
7. **Optional: Install Extensions:**
   - Click on the Extensions icon in the Activity Bar or press `Ctrl+Shift+X`.
   - Search for and install desired extensions.

1. First-time Setup:

Initial Configurations:

1. **User Settings:**

    - Open the settings by clicking the gear icon in the lower left corner and selecting "Settings" or pressing `Ctrl+,`.
    - Adjust the following key settings:
        - **Theme:** Choose a theme you prefer (e.g., "Dark+" for a dark theme).
        - **Font Size:** Set a comfortable font size (e.g., 14 or 16).
        - **Tab Size:** Adjust the tab size (e.g., 2 or 4 spaces).
2. **Workspace Settings:**
    - Save specific settings for different projects by creating a `.vscode/settings.json` file in your project directory.
3. **Keybindings:**
    - Customize keybindings by selecting "Keyboard Shortcuts" from the gear menu or pressing `Ctrl+K, Ctrl+S`.
    - Adjust shortcuts to match your workflow.
4. **File and Editor Settings:**
    - Enable auto-save: `"files.autoSave": "afterDelay"`.
    - Set line endings to LF or CRLF as needed: `"files.eol": "\n"` or `"files.eol": "\r\n"`.
    - Enable format on save: `"editor.formatOnSave": true`.

Important Extensions:

1. **Programming Languages:**
    - **Python:** Install the "Python" extension by Microsoft.
    - **JavaScript/TypeScript:** Install the "ESLint" and "Prettier - Code formatter" extensions.
    - **C++:** Install the "C/C++" extension by Microsoft.
2. **Version Control:**
    - **Git:** Install the "GitLens" extension to enhance Git capabilities.
3. **Code Formatting:**
    - **Prettier:** Install "Prettier - Code formatter" for consistent code formatting.
    - **ESLint:** Install "ESLint" for linting JavaScript and TypeScript.

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.

Main Components:

1. **Activity Bar:**
    - **Location:** The vertical bar on the far left of the window.
    - **Purpose:** Provides access to various views and functions such as Explorer, Search, Source Control, Run and Debug, and Extensions.
    - **Icons:**
        - **Explorer:** Access and manage files and folders.
        - **Search:** Perform search and replace across files.
        - **Source Control:** Manage version control operations with Git or other SCM providers.
        - **Run and Debug:** Configure and launch debugging sessions.
        - **Extensions:** Browse and install VS Code extensions.
2. **Side Bar:**
    - **Location:** To the right of the Activity Bar.
    - **Purpose:** Displays the content of the selected Activity Bar view.
    - **Examples:**
        - **Explorer:** Shows a tree view of your project files and folders.
        - **Source Control:** Displays Git repository status, change history, and allows commit operations.
        - **Extensions:** Lists installed extensions and provides a search box for finding new ones.
3. **Editor Group:**
    - **Location:** The main central area where you write and edit code.
    - **Purpose:** Hosts open files, divided into multiple tabs if needed.
    - **Features:**
        - **Tabs:** Allows you to switch between multiple open files.
        - **Split Editor:** You can split the editor to view and edit multiple files side by side.
4. **Status Bar:**
    - **Location:** The horizontal bar at the bottom of the window.
    - **Purpose:** Provides information about the current file and workspace.

4. 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. remember to be brief always and straight to the point

It Provides quick access to various commands and features in VS Code.

How to Access:

- **Shortcut: Press `Ctrl+Shift+P`**

Common Tasks:

1. **Open Settings: Type `Preferences: Open Settings`.**
2. **Install Extensions: Type `Extensions: Install Extensions`.**
3. **Toggle Terminal: Type `View: Toggle Terminal`.**
4. **Change Theme: Type `Preferences: Color Theme`.**
5. **Format Document: Type `Format Document`.**
6. **Git Commands: Type `Git: Commit` or `Git: Pull`.**

5.

VS Code Extensions:

- **They add features and functionalities to VS Code (like linters, debuggers, new languages).**
- **How to find them: Open Extensions tab (puzzle piece icon) and browse/search the marketplace.**
- **How to install: Click "Install" next to the extension you want.**

Essential Web Dev Extensions:

- **ESLint (catches errors)**
- **Live Server (live code refresh)**

6.

VS Code Integrated Terminal:

- **Open:**
  - **Go to View > Terminal menu.**
  - **Keyboard shortcut: `Ctrl` + `'` (backtick).**
- **Use it:**
  - **Type your terminal commands (e.g., `git status`, `npm install`).**
  - **Press `Enter` to run the command.**

Advantages:

- **Convenience: No need to switch between VS Code and a separate terminal window.**
- **Integration: Works seamlessly with VS Code features like file navigation and code execution.**

7.

VS Code File & Folder Management:

- **Explorer View: This panel (usually on the left) shows your project folders and files.**
- **Create Files/Folders:**
  - **Right-click in the Explorer view and select "New File" or "New Folder".**
  - **Keyboard shortcut: `Ctrl` + `N` (new file), `Ctrl` + `Shift` + `N` (new folder).**
- **Open Files:**
  - **Double-click the file in the Explorer view.**
  - **Use the "Go to File" feature (`Ctrl` + `T`) to search for files by name.**
- **Navigate Efficiently:**
  - **Use the Explorer view to see and click through folders.**
  - **Use the "Open Recent" feature (`Ctrl` + `Shift` + `O`) to quickly access recently opened files.**
  - **Keyboard shortcuts for navigation (e.g., `Ctrl` + `.` to move up a directory).**

8.

VS Code Settings & Preferences:

- **Access:**
  - **Go to File > Preferences > Settings menu.**
  - **Keyboard shortcut: `Ctrl` + `,` (comma).**
- **Customization Examples:**
  - **Theme: Search for "Theme" and pick a new one from the dropdown.**
  - **Font Size: Search for "Font Size" and adjust the slider.**
  - **Keybindings: Search for "Keyboard Shortcuts" and modify key combinations for specific actions.**

9.

VS Code Debugging: Cliff Notes

1. **Set breakpoints (red dots) in your code.**
2. **Open Run & Debug view (Ctrl+Shift+D).**
3. **Hit Run (play icon) or F5.**
4. **Use Step Over (F10), Step Into (F11), Step Out (Shift+F11) to navigate.**
5. **Check variable values and call stack to debug**

10.

VS Code & Git: Version Control

Integration:

1. **Open your project folder in VS Code.**
2. **Click the Source Control icon (gear icon) and select "Initialize Repository." This creates a local Git repository for your project.**

Making Commits:

1. **Edit your files as usual.**
2. **Stage changes you want to track with Git**
3. **Type a descriptive commit message and click "Commit"**

Pushing to GitHub:

1. **Create a remote repository on GitHub for your project**
2. **In VS Code, find the "Publish branch to GitHub"**
3. **Push your committed changes to GitHub with the "Push" button**