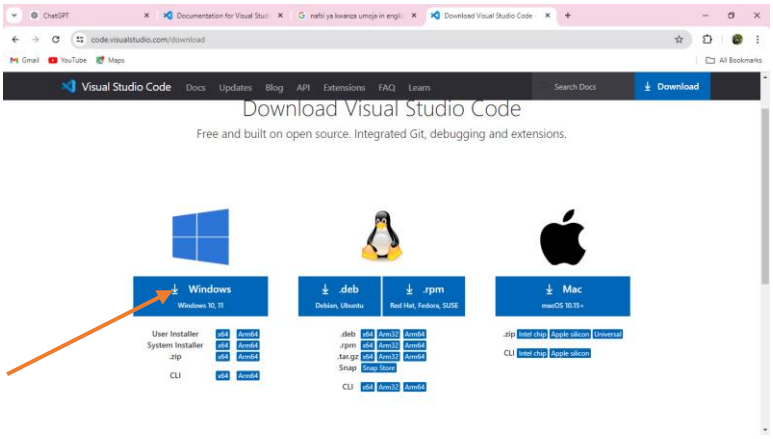
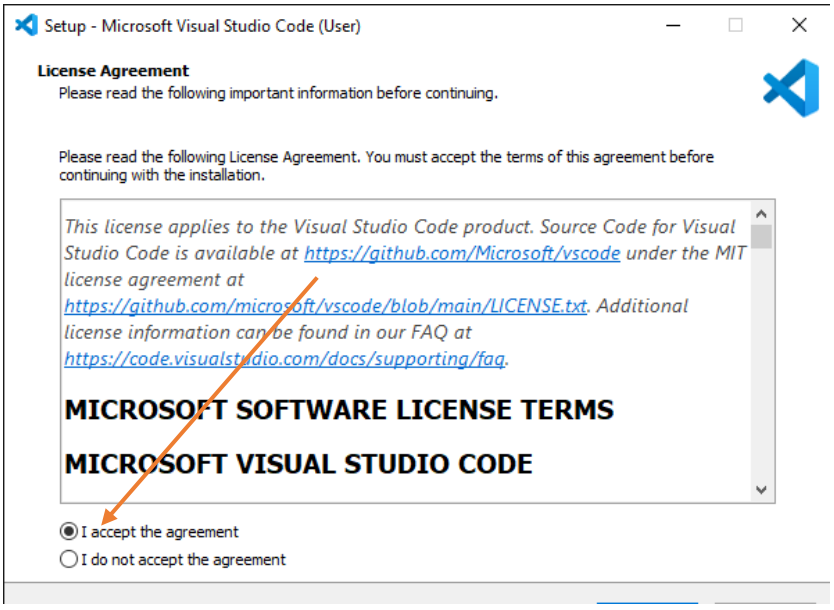
Installation of VS Code:

Steps to Download and Install Visual Studio Code on Windows 11:

1. **Download VS Code:**
    - I visited the [Visual Studio Code download page](code.visualstudio.com/download).



    - I clicked on the "Download for Windows" button.
2. **Install VS Code:**
    - Once the download was complete, I opened the downloaded file (VSCodeSetup.exe).



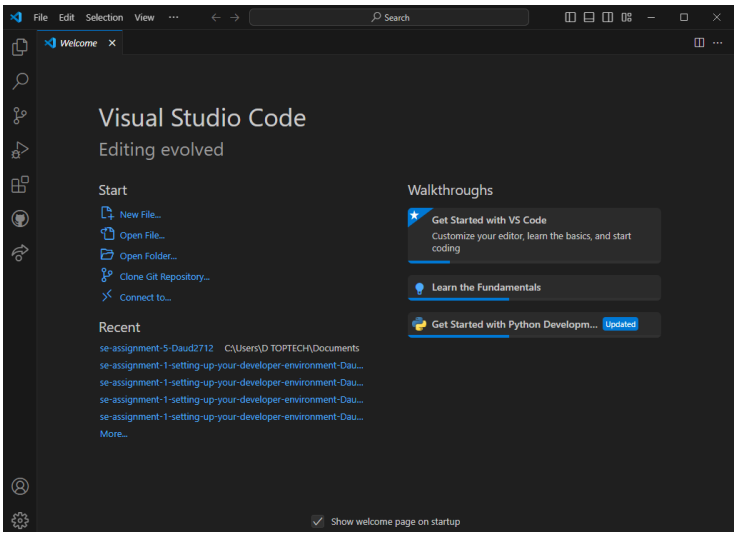
    - I followed the installation wizard steps:
        - Accepted the license agreement.
        - Chose the destination folder.
        - Selected the additional tasks I wanted to perform (e.g., creating a desktop icon, adding to PATH).
        - Clicked "Install" to begin the installation.

3. **Prerequisites:**
    - I ensured I had administrative privileges to install software on my machine.
    - While no specific prerequisites are needed, it's recommended to have Git installed for version control integration. I downloaded Git from [git-scm.com](git-scm.com).

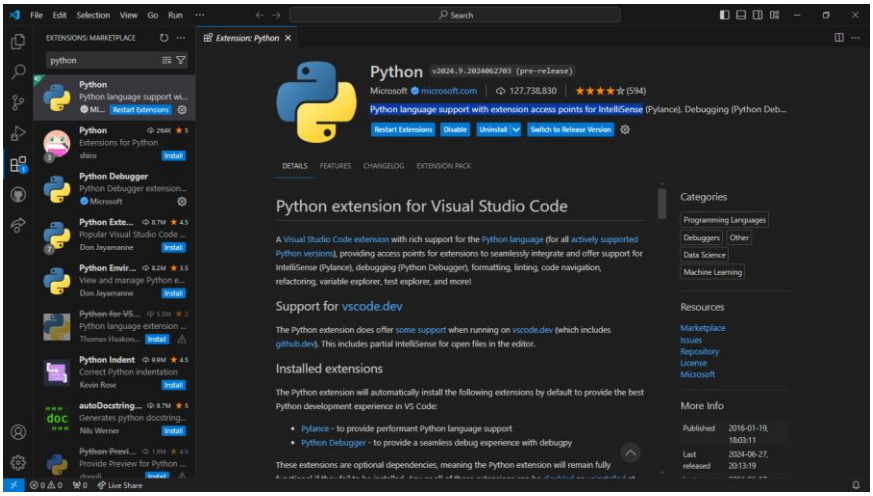
First-time Setup:

Initial Configurations and Settings:

1. **Theme and Appearance:**
    - I opened VS Code.

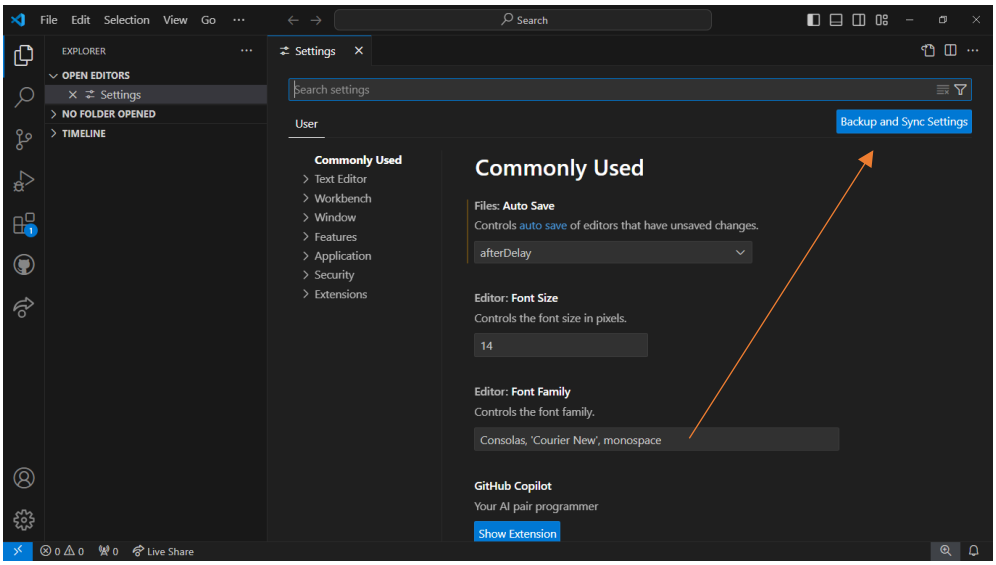


    - I went to File > Preferences > Color Theme to select a theme.
2. **Extensions:**
    - I clicked on the Extensions icon in the Activity Bar on the side of the window or pressed Ctrl+Shift+X.
    - I searched for and installed essential extensions such as:
        - **Python language** (Python extension for Visual Studio Code)



- **ESLint** (JavaScript Linter)
- **Live Server** (Launch a development local Server)
- **Python** (for Python development)

3. **Settings:**
    - I opened settings by navigating to File > Preferences > Settings or pressed Ctrl+,.



    - I adjusted settings such as font size, tab size, and auto-save.

User Interface Overview:

Main Components of the VS Code User Interface:

1. **Activity Bar:**
    - Located on the far left.
    - Provides access to different views like Explorer, Search, Source Control, Run & Debug, and Extensions.
2. **Side Bar:**
    - Displays the content of the selected view from the Activity Bar.
    - Example: The Explorer view shows files and folders in the current project.
3. **Editor Group:**
    - The main area where I edit files.
    - Supports multiple tabs and split views.
4. **Status Bar:**
    - Located at the bottom of the window.
    - Shows information like the current file's encoding, line/column numbers, and language mode.

Command Palette:

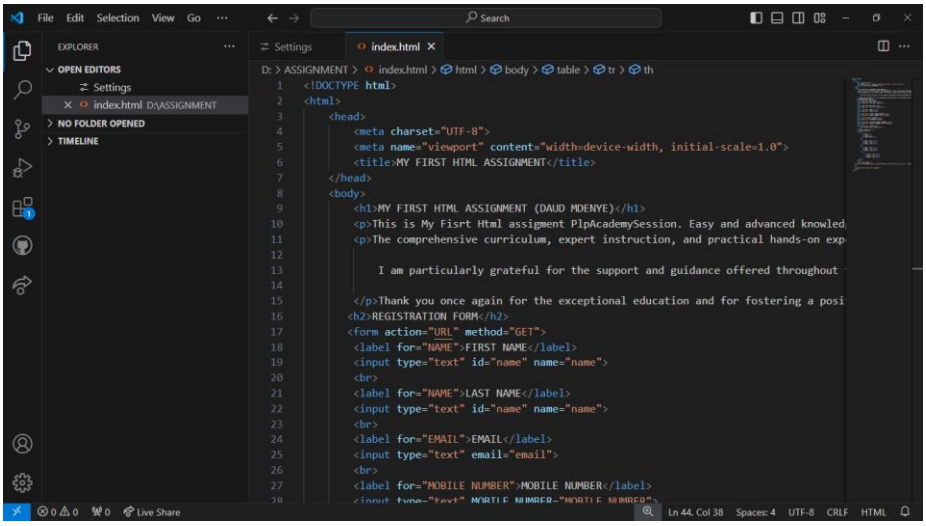
Description and Access:

- The Command Palette provides quick access to many commands and features in VS Code.
- I accessed it by pressing Ctrl+Shift+P or F1.
- Examples of common tasks:
    - Open files (Ctrl+P)
    - Change the theme (> Preferences: Color Theme)
    - Install extensions (> Extensions: Install Extensions)

Extensions in VS Code:

Role and Management of Extensions:

- Extensions enhance VS Code's functionality by adding features such as language support, debuggers, and tools.
- **Finding and Installing Extensions:**
    - I clicked on the Extensions icon in the Activity Bar or pressed Ctrl+Shift+X.
    - I searched for the desired extension and clicked "Install".
- **Managing Extensions:**
    - I viewed installed extensions by clicking on the Extensions icon.
    - I disabled or uninstalled extensions as needed.
- **Essential Extensions for Web Development:**
    - **HTML Snippets**



    - **CSS IntelliSense**
    - **JavaScript (ES6) code snippets**
    - **Debugger for Chrome**

Integrated Terminal:

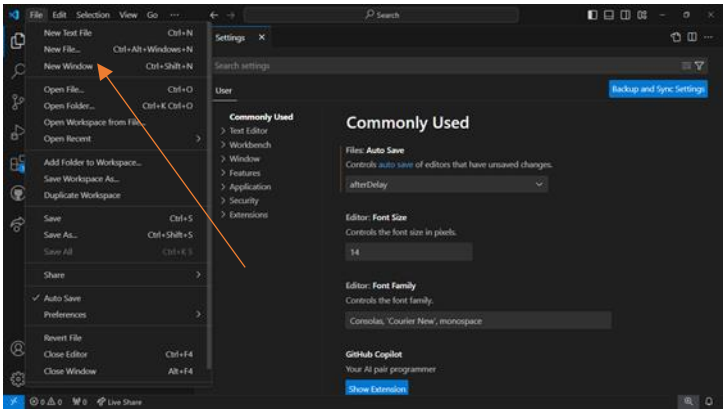
Usage and Advantages:

1. **Opening the Integrated Terminal:**
    - I went to View > Terminal or pressed `Ctrl+`` (backtick).
2. **Advantages:**
    - Integrated with the editor for seamless workflow.
    - Supports multiple terminal instances.
    - Provides the same environment as an external terminal but within VS Code.

File and Folder Management:

Creating, Opening, and Managing Files and Folders:

1. **Creating:**
    - I right-clicked in the Explorer view and selected New File or New Folder.
    - I used Ctrl+N for a new file.



2. **Opening:**
    - I double-clicked a file in the Explorer view.
    - I used File > Open File or Ctrl+O.
3. **Managing:**
    - I used the Explorer view to move, rename, and delete files/folders.
    - I navigated between files with Ctrl+P and the Quick Open feature.

Settings and Preferences:

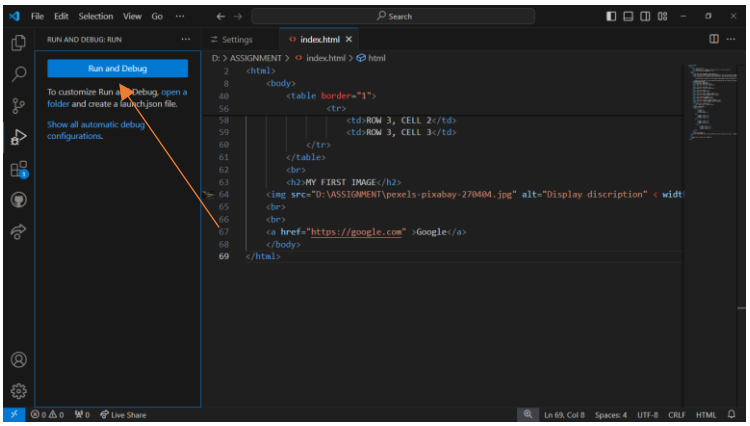
Customizing Settings:

1. **Accessing Settings:**
    - I went to File > Preferences > Settings or pressed Ctrl+,.
2. **Examples:**
    - **Change Font Size:**
        - Searched for "Font Size" in settings and adjusted the value.

Debugging in VS Code:

Setting Up and Starting Debugging:

1. **Steps to Debug:**
    - I opened the file I wanted to debug.
    - I set breakpoints by clicking in the gutter next to the line numbers.
    - I went to the Run & Debug view by clicking the Run icon in the Activity Bar or pressing Ctrl+Shift+D.
    - I clicked on "Run and Debug" and selected the appropriate environment.



2. **Key Debugging Features:**
    - Breakpoints
    - Watch variables
    - Call stack
    - Step in, over, and out

Using Source Control:

Integrating Git with VS Code:

1. **Initializing a Repository:**
    - I opened the project folder in VS Code.
    - I opened the Source Control view by clicking the Source Control icon in the Activity Bar.
    - I clicked "Initialize Repository".
2. **Making Commits:**
    - I staged changes by clicking the + icon next to changed files.
    - I entered a commit message and clicked the checkmark icon to commit.
3. **Pushing Changes to GitHub:**
    - I set up a remote repository on GitHub.
    - I used the integrated terminal to run:

git remote add origin <repository-url>

git push -u origin main

References:

- Visual Studio Code Documentation
- GitHub

  https://github.com/Daud2712/se-assignment-1-setting-up-your-developer-environment-Daud2712.git