**Dev Setup assignment**

1. **Installing Windows 11**

**Step 1: Before you begin**

Make sure the device you want to install Windows 11 on meets the minimum system requirements. If your device is currently running Windows 10, we recommend you verify the minimum system requirements using the PC Health Check app. We do not recommend installing Windows 11 on a device that doesn't meet requirements. For more info, see Installing Windows 11 on devices that don't meet minimum system requirements.

Windows Update in Settings

If you’re upgrading from Windows 10, we recommend you wait until you're notified through Windows Update that the upgrade is ready for your device. To check if Windows 11 is ready for your device, select Start > Settings > Update & Security > Windows Update > Check for updates.

System requirements

These are the minimum system requirements for installing Windows 11 on a PC. If your device does not meet these requirements, you may not be able to install Windows 11 on your device and might want to consider purchasing a new PC. If you are unsure whether your PC meets these requirements, you can check with your PC Original Equipment Manufacturer (OEM) or, if your device is already running Windows 10, you can use the PC Health Check app to assess compatibility. Note that this app does not check for graphics card or display, as most compatible devices will meet those requirements listed below.

Your device must be running Windows 10, version 2004 or later, to upgrade. Free updates are available through Windows Update in Settings>Update and Security.

**Processor**

1 gigahertz (GHz) or faster with 2 or more cores on a compatible 64-bit processor or System on a Chip (SoC).

**RAM**

4 gigabyte (GB).

**Storage**

64 GB or larger storage device

**System firmware** UEFI, Secure Boot capable

**TPM**

Trusted Platform Module (TPM) version 2.0. Check here for instructions on how your PC might be enabled to meet this requirement.

**Graphics card**

Compatible with DirectX 12 or later with WDDM 2.0 driver.

**Display**

High definition (720p) display that is greater than 9” diagonally, 8 bits per color channel.

**Internet connection and Microsoft account**

Windows 11 Home requires internet connectivity and a Microsoft account during initial device setup.

**Step 2: Create a Bootable USB Drive**

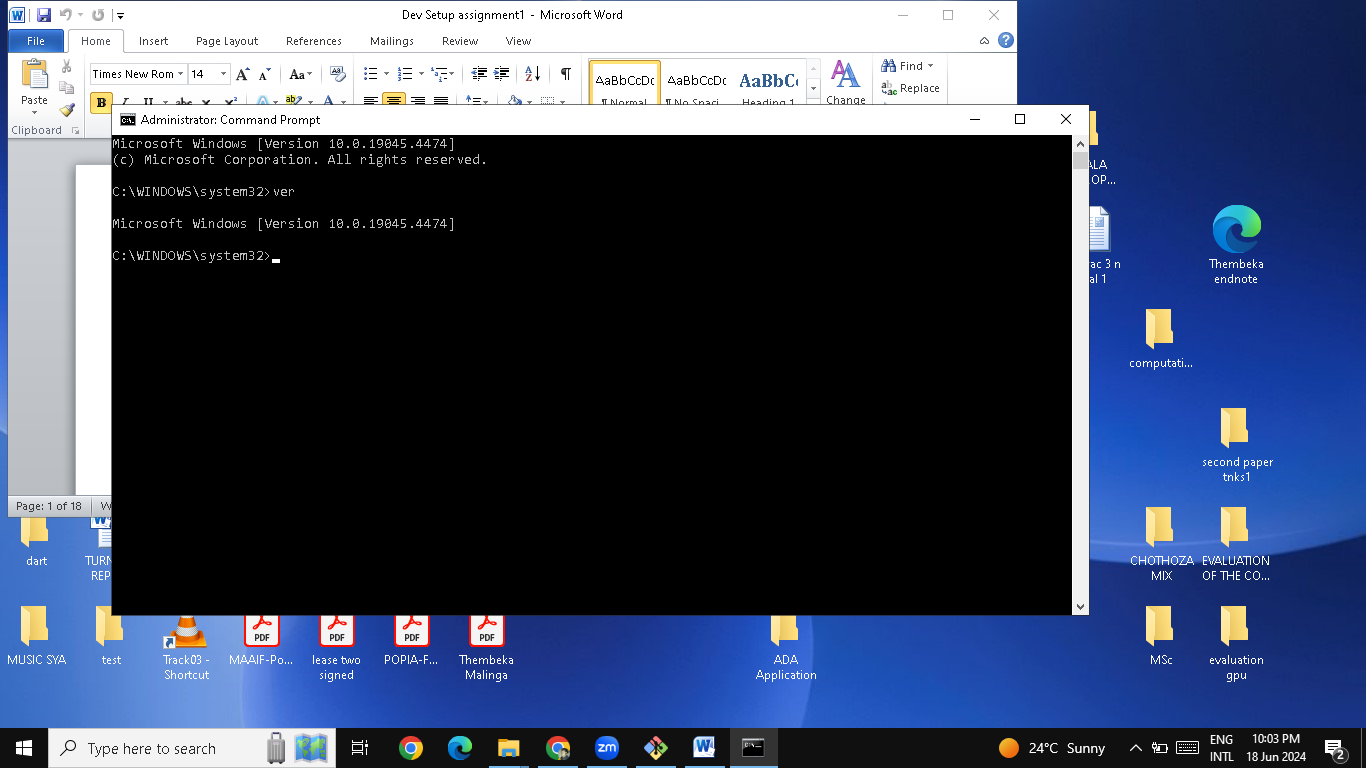
1. **Download Windows 11 ISO**:
   * Go to the official Microsoft website to download the Windows 11 ISO file:  <https://www.microsoft.com/software-download/windows11>
2. **Download and Install the Media Creation Tool**:
   * You can also use the Media Creation Tool from the same page. Download it and run the tool.
3. **Create Installation Media**:
   * Open the Media Creation Tool and select "Create installation media (USB flash drive, DVD, or ISO file) for another PC".
   * Choose the language, edition, and architecture (64-bit).
   * Select "USB flash drive" as the media to use.
   * Insert a USB flash drive with at least 8 GB of space and select it in the tool.
   * Follow the prompts to create the bootable USB drive.

**Step 3: Install Windows 11**

1. **Insert the Bootable USB Drive**:
   * Insert the USB drive into the PC where you want to install Windows 11.
2. **Boot from the USB Drive**:
   * Restart your PC and press the appropriate key (e.g., F12, F2, Esc, or Del) during the boot process to enter the BIOS or Boot Menu.
   * Select the USB drive as the boot device.
3. **Start the Installation Process**:
   * When the Windows Setup screen appears, select your language, time, and keyboard preferences, then click "Next".
   * Click "Install Now".
4. **Enter Product Key**:
   * Enter your Windows 11 product key, or click "I don't have a product key" if you want to enter it later.
5. **Select the Installation Type**:
   * Choose "Custom: Install Windows only (advanced)" to perform a clean installation.
6. **Partition the Drive**:
   * Select the drive where you want to install Windows 11. You can delete existing partitions to create new ones if needed (note that this will erase all data on those partitions).
7. **Install Windows 11**:
   * Follow the on-screen instructions to complete the installation. Your PC will restart several times during the process.

**Step 4: Set Up Windows 11**

1. **Choose Your Region and Keyboard Layout**:
   * Select your region and keyboard layout.
2. **Connect to a Network**:
   * Connect to a Wi-Fi network or plug in an Ethernet cable.
3. **Sign In with Microsoft Account**:
   * You will be prompted to sign in with your Microsoft account. If you don’t have one, you can create one or choose to sign in later.
4. **Set Up Windows**:
   * Follow the prompts to set up your preferences, privacy settings, and other configurations.
5. **Finish and Enjoy Windows 11**:
   * Once the setup is complete, you will be taken to the Windows 11 desktop.



1. **Installing Visual Studio Code on windows**

**Step 1: Download Visual Studio Code**

1. **Visit the Official Website**:
   * Open your web browser and go to the official Visual Studio Code website: [code.visualstudio.com](https://code.visualstudio.com/).
2. **Download the Installer**:
   * Click the "Download for Windows" button. This will download the installer for VS Code.

**Step 2: Install Visual Studio Code**

1. **Run the Installer**:
   * Once the download is complete, locate the installer file (usually named something like VSCodeUserSetup-x64-1.x.x.exe) in your Downloads folder and double-click to run it.
2. **Setup Wizard**:
   * The Visual Studio Code Setup wizard will open. Follow the steps below:

**Welcome Screen**:

* + Click "Next".

**License Agreement**:

* + Read the license agreement, select "I accept the agreement", and click "Next".

**Select Destination Location**:

* + Choose the destination folder where you want to install VS Code. The default location is typically fine. Click "Next".

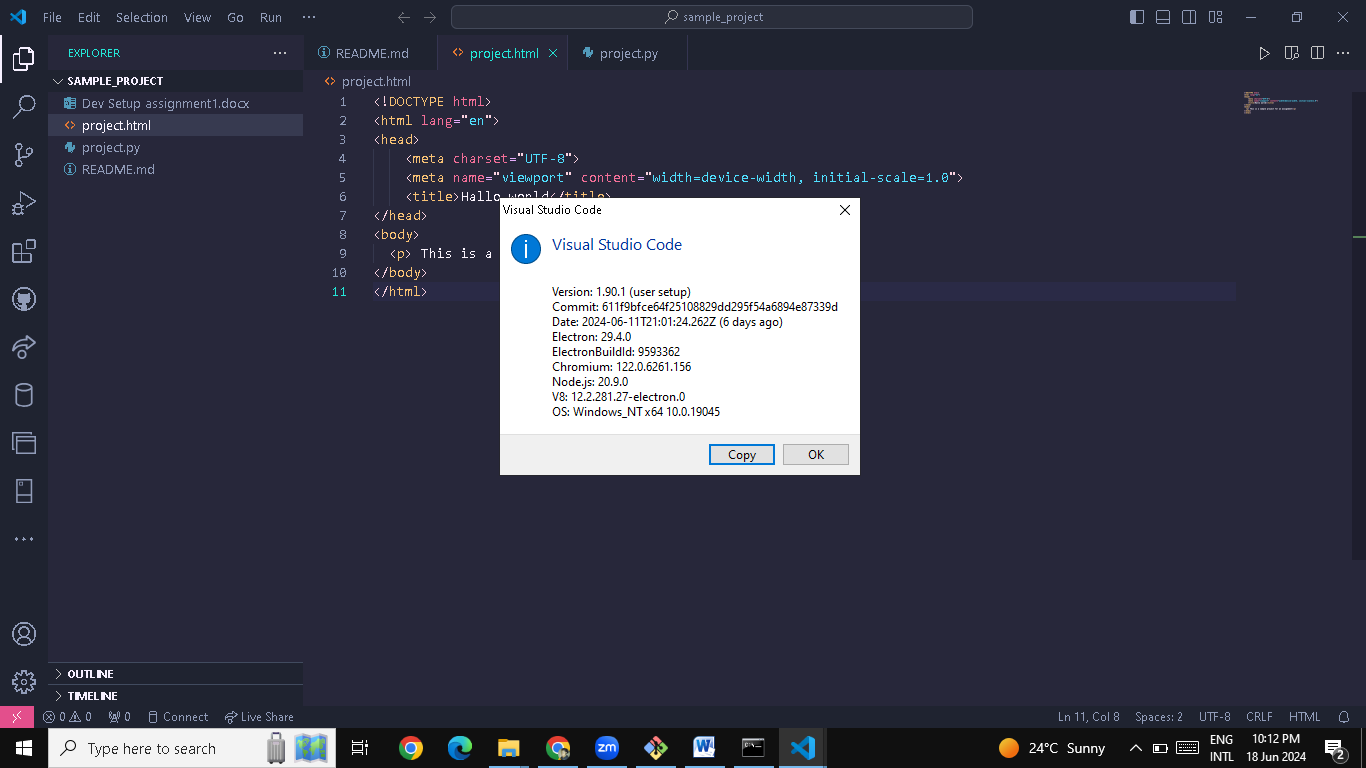
**Select Start Menu Folder**:

* + Choose the Start Menu folder for the program’s shortcuts. Click "Next".

**Select Additional Tasks**:

* + Choose additional tasks you would like to perform while installing VS Code:
    - **Create a desktop icon**: Check this box if you want a shortcut on your desktop.
    - **Add "Open with Code" action to Windows Explorer file context menu**: This is useful for opening files directly from the context menu.
    - **Add "Open with Code" action to Windows Explorer directory context menu**: This is useful for opening directories.
    - **Register Code as an editor for supported file types**.
    - **Add to PATH (available after restart)**: This is useful for accessing VS Code from the command line.
  + Click "Next".

1. **Install**:
   * Click "Install" to begin the installation process. This may take a few minutes.
2. **Completion**:
   * Once the installation is complete, you can choose to launch Visual Studio Code immediately by checking the "Launch Visual Studio Code" box.
   * Click "Finish".

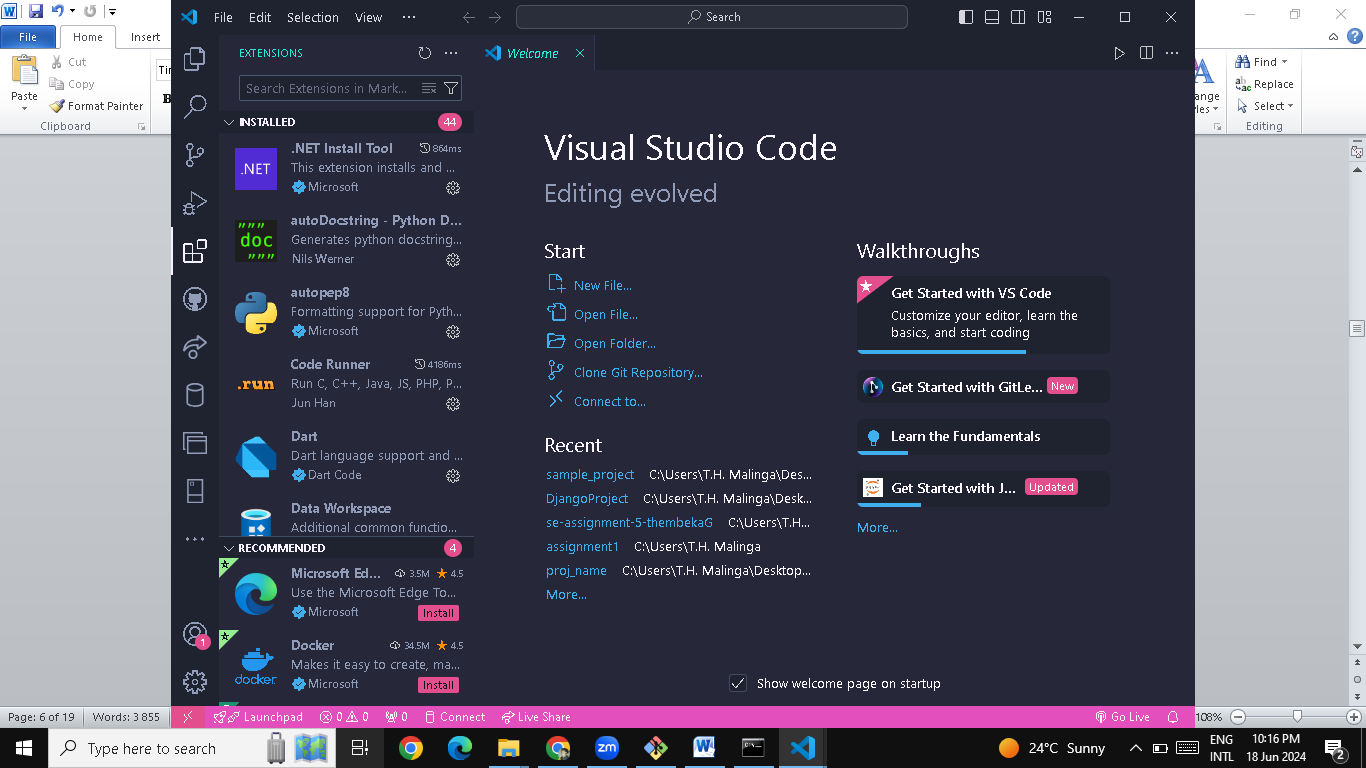


**Step 3: First Launch and Initial Setup**

1. **Launch VS Code**:
   * If you didn’t check the "Launch Visual Studio Code" box during installation, you can open VS Code from the Start menu or the desktop shortcut.
2. **Initial Setup**:
   * When you first open VS Code, you may be presented with a "Welcome" screen. This screen provides helpful tips and links to documentation.
   * You can customize the editor by installing extensions, changing themes, and configuring settings.

**Step 4: Installing Extensions**

1. **Open Extensions View**:
   * Click the Extensions icon in the Activity Bar on the side of the window or use the shortcut Ctrl+Shift+X.
2. **Search for Extensions**:
   * In the Extensions view, you can search for and install extensions to add new features and functionality to VS Code. Some popular extensions include:
     + **Python**: For Python development.
     + **Prettier - Code formatter**: For code formatting.
     + **ESLint**: For linting JavaScript and TypeScript code.
     + **Live Server**: For a live-reload server for static and dynamic pages.
     + **GitLens**: Supercharges the built-in Git capabilities.
3. **Install an Extension**:
   * Click the Install button for the extension you want to add.



1. **Installing Git and Configuring it on Windows**

To use Git with VS Code on Windows, you need to install Git and configure it properly. Here are the detailed steps:

**Step 1: Download and Install Git**

1. **Download Git**:
   * Go to the official Git website: [git-scm.com](https://git-scm.com/).
   * Click on the "Download" button. The website should automatically detect your operating system and provide the correct version of Git for Windows.
2. **Run the Installer**:
   * Once the download is complete, run the installer (Git-2.x.x-64-bit.exe).
3. **Setup Wizard**:
   * Follow the setup wizard. Here are the recommended settings for each step:

**Select Destination Location**:

* + Choose the default location or specify a different one if needed.

**Select Components**:

* + Leave the default options selected. You can optionally add additional icons to the desktop or start menu if you prefer.

**Select Start Menu Folder**:

* + Choose the default folder or specify a different one if needed.

**Choosing the default editor used by Git**:

* + Select "Use Visual Studio Code as Git's default editor".

**Adjusting your PATH environment**:

* + Choose "Git from the command line and also from 3rd-party software" for maximum compatibility.

**Choosing HTTPS transport backend**:

* + Select "Use the OpenSSL library".

**Configuring the line ending conversions**:

* + Choose "Checkout Windows-style, commit Unix-style line endings".

**Configuring the terminal emulator to use with Git Bash**:

* + Select "Use MinTTY (the default terminal of MSYS2)".

**Choosing the default behavior of git pull**:

* + Choose "Default (fast-forward or merge)".

**Choose a credential helper**:

* + Select "Git Credential Manager Core".

**Configuring extra options**:

* + Leave the default options selected (e.g., enable file system caching, enable Git credential manager).

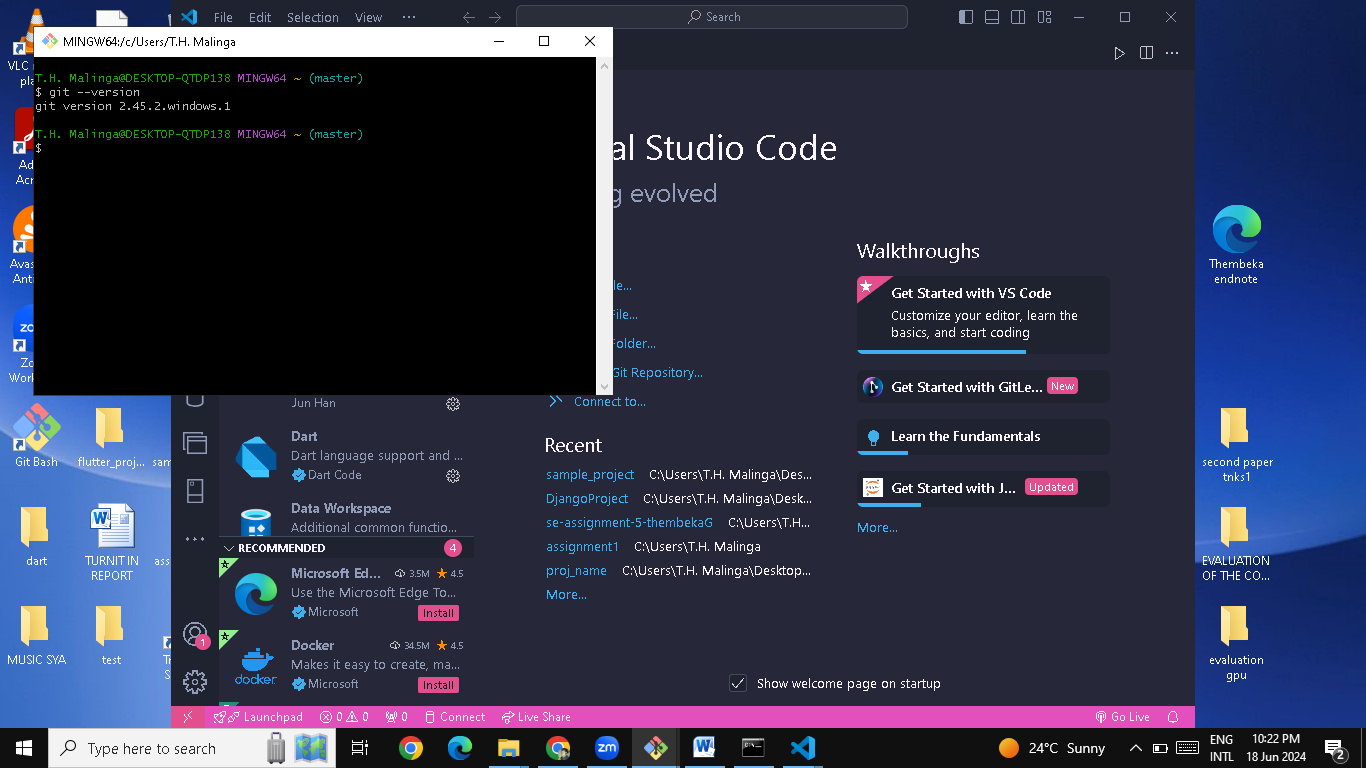
**Configuring experimental options**:

* + Leave these unchecked unless you want to try out experimental features.

1. **Complete the Installation**:
   * Click "Install" to begin the installation.
   * Once the installation is complete, click "Finish".

**Step 2: Verify Git Installation**

1. **Open Git Bash**:
   * After installation, you can open Git Bash from the Start menu or by right-clicking on the desktop.
2. **Check Git Version**:
   * In the Git Bash terminal, type: git --version
   * This should display the installed version of Git.



**Step 3: Configure Git**

1. **Set Your Username and Email**:

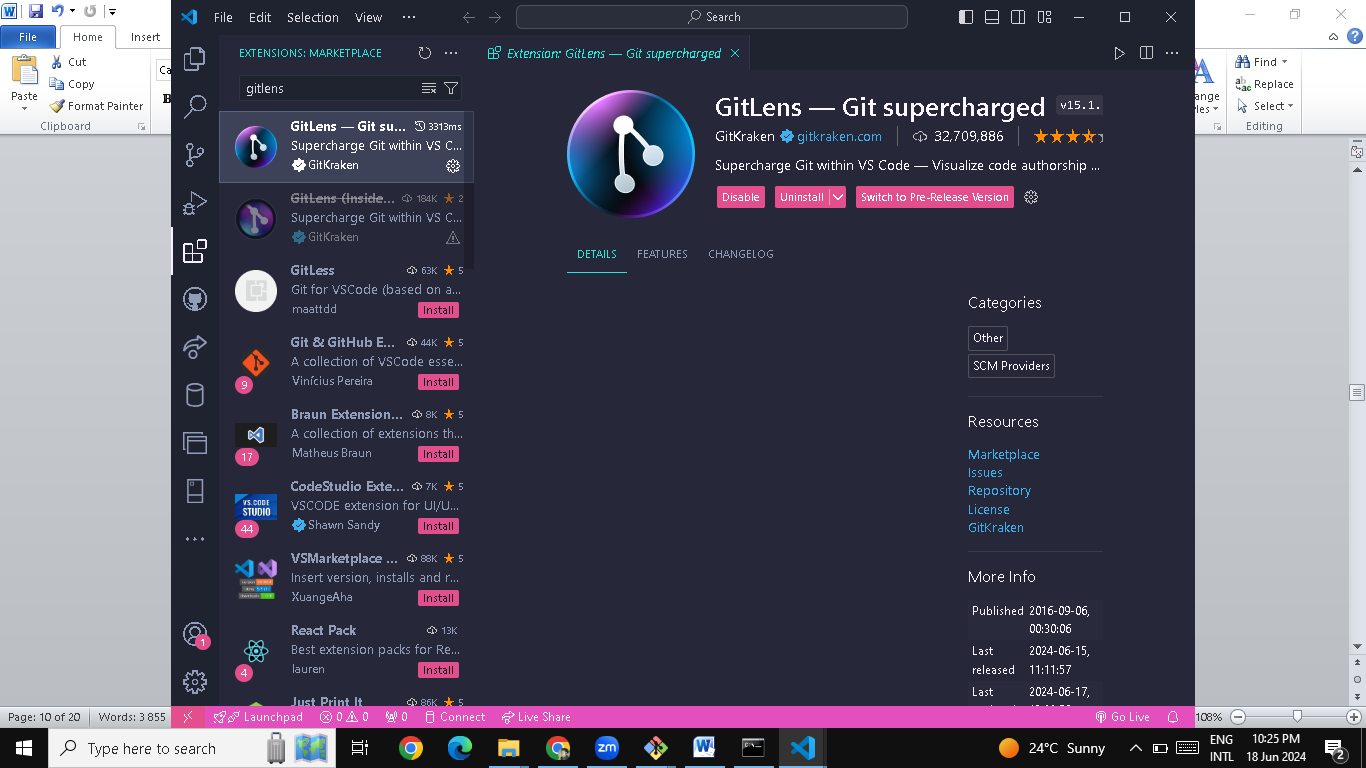
git config --global user.name "thembekaG"

git config --global user.email "94malinga@gmail.com"

1. **Verify Configuration**:
   * To verify your configuration, run: git config --list
   * This will display a list of your Git configuration settings, including your username and email.

**Step 4: Integrate Git with VS Code**

1. **Open VS Code**:
   * Launch Visual Studio Code.
2. **Install Git Extension**:
   * Although VS Code has built-in Git support, you can enhance it with extensions. Open the Extensions view by clicking the Extensions icon in the Activity Bar or by using the shortcut Ctrl+Shift+X.
   * Search for "Git" and install any extensions that add features you might need, such as "GitLens".



1. **Verify Git Integration**:
   * Open the Command Palette with Ctrl+Shift+P
   * Type >Git: Clone to ensure Git commands are recognized.

**Example Workflow: Initializing a Repository and Making Your First Commit**

1. **Open a Folder**:
   * In VS Code, go to File > Open Folder and select the folder you want to use as a Git repository.
2. **Initialize Git Repository**:
   * Open the Source Control view (Ctrl+Shift+G).
   * Click Initialize Repository.
3. **Create a New File**:
   * Create a new file in your project folder, e.g., README.md.
   * Add some content to the file and save it.
4. **Stage and Commit Changes**:
   * In the Source Control view, you’ll see the new file listed under Changes.
   * Hover over the file and click the + icon to stage it.
   * Enter a commit message, like "Initial commit", and click the checkmark icon (✓) to commit.
5. **Push Changes to GitHub**:
   * Open the integrated terminal in VS Code (Ctrl+).
   * Add your GitHub repository as a remote: git remote add origin https://github.com/thembekaG/your-repository.git
   * Push your changes: git push -u origin master
6. **Installing Python**

**Step 1: Download Python**

1. **Visit the Official Python Website**:
   * Open your web browser and go to the official Python website: [python.org](https://www.python.org/).
2. **Download the Installer**:
   * On the Python homepage, click the "Downloads" link.
   * The website should automatically suggest the latest version for Windows. Click the "Download Python 3.12.2" button to download the installer.

**Step 2: Install Python**

1. **Run the Installer**:
   * Once the download is complete, locate the installer file (usually named something like python-3.2.2.exe) in your Downloads folder and double-click to run it.
2. **Setup Wizard**:
   * The Python Setup wizard will open. Follow the steps below:

**Important: Add Python to PATH**:

* + At the bottom of the first window, check the box that says "Add Python 3.12.2 to PATH". This will allow you to run Python from the command line.

**Install Now or Customize Installation**:

* + Click "Install Now" to install Python with the default settings, or choose "Customize Installation" if you want to select specific features and installation paths.

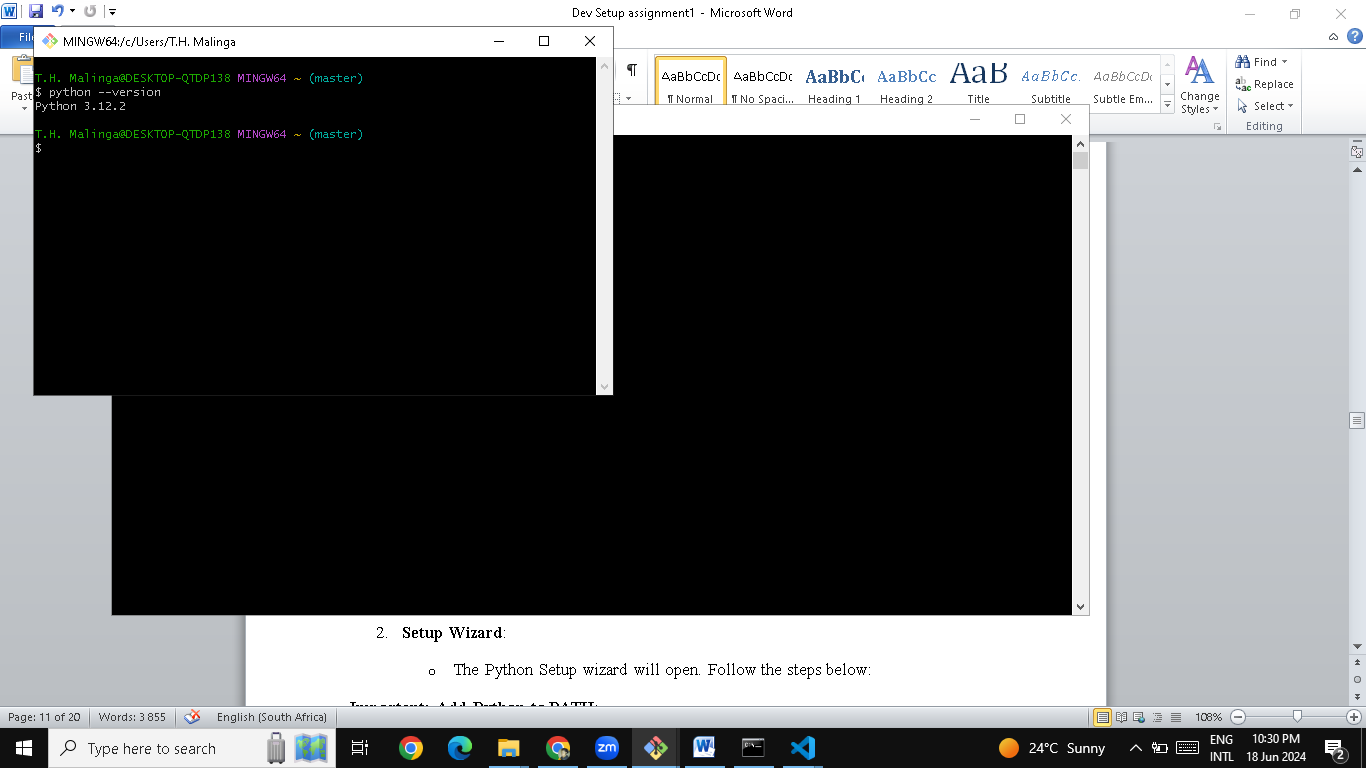
**Advanced Options**:

* + In the "Advanced Options" screen, you can choose to install Python for all users (requires administrative privileges) and to add Python to the environment variables. Both are recommended.

1. **Complete the Installation**:
   * Click "Install" to begin the installation process. This may take a few minutes.
   * Once the installation is complete, click "Close".

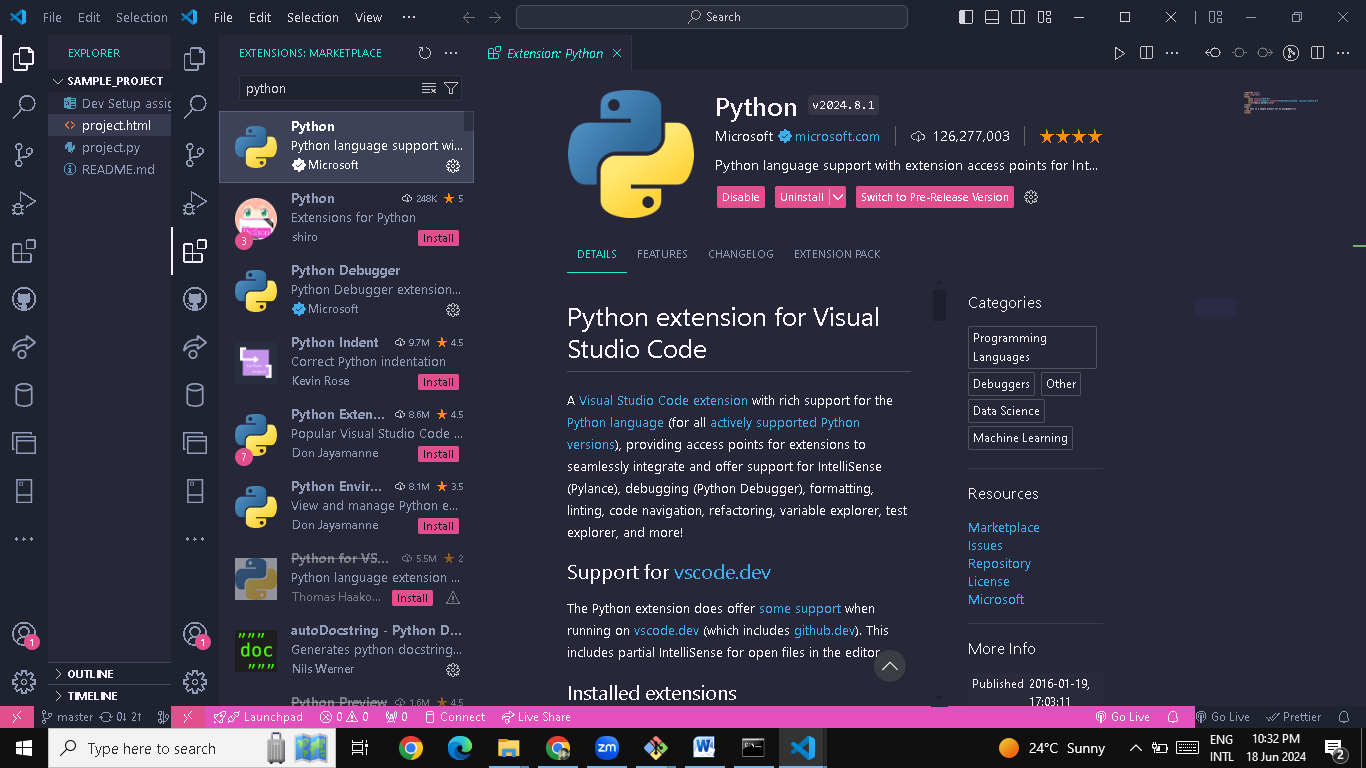
**Step 3: Verify Python Installation**

1. **Open Command Prompt**:
   * Open Command Prompt by typing cmd in the Start menu search bar and pressing Enter.
2. **Check Python Version**:
   * In the Command Prompt, type: python --version
   * This should display the version of Python you installed.



**Step 4: Install and Configure VS Code Extensions**

1. **Open VS Code**:
   * Launch Visual Studio Code.
2. **Install Python Extension**:
   * Open the Extensions view by clicking the Extensions icon in the Activity Bar on the side of the window or by using the shortcut Ctrl+Shift+X.
   * Search for "Python" and install the extension provided by Microsoft.



1. **Install Python-related Extensions (Optional)**:
   * You can install additional extensions like "Pylance" for improved IntelliSense and "Python Docstring Generator" for better documentation support.
2. **Configure Python Interpreter**:
   * Open the Command Palette with Ctrl+Shift+P and type Python: Select Interpreter.
   * Select the Python interpreter you want to use (e.g., the one from your virtual environment).
     1. **Installing PIP**

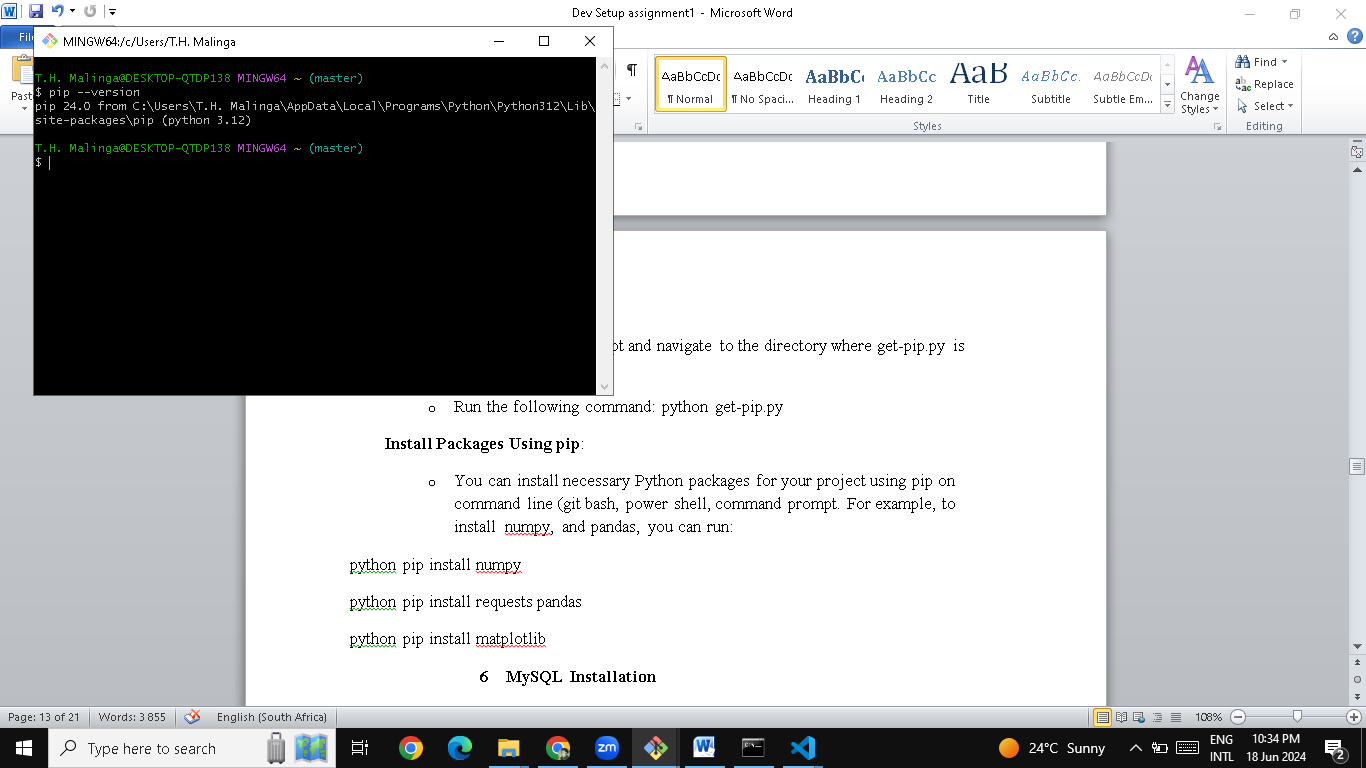
**1. Check pip Version:**

o In the Command Prompt, type: pip --version

o This should display the version of pip installed.

**Install pip**:

* If pip is not installed, you can install it using the following steps:
  + Download get-pip.py from the [official repository](https://bootstrap.pypa.io/get-pip.py).
  + Open Command Prompt and navigate to the directory where get-pip.py is saved.
  + Run the following command: python get-pip.py



**Install Packages Using pip**:

* + You can install necessary Python packages for your project using pip on command line (git bash, power shell, command prompt. For example, to install numpy, and pandas, you can run:

python pip install numpy

python pip install requests pandas

python pip install matplotlib

* + 1. **MySQL Installation**

**Step 1: Download MySQL Installer**

1. **Visit the Official MySQL Website**:
   * Open your web browser and go to the MySQL download page: <https://dev.mysql.com/downloads/windows/installer/5.7.html>
2. **Download MySQL Installer**:
   * On the MySQL Installer page, you’ll see two download options: "Web Community" (smaller size) and "Full" (includes all MySQL products).
   * Click the "Download" button next to "Windows (x86, 32-bit), MSI Installer" to download the installer.
3. **Begin the Download**:
   * You may be prompted to log in or sign up for an Oracle account, but you can skip this step by clicking on "No thanks, just start my download."

**Step 2: Install MySQL**

1. **Run the Installer**:
   * Once the download is complete, locate the installer file (mysql-installer-community-5.7.x.x.msi) in your Downloads folder and double-click to run it.
2. **Setup Wizard**:
   * The MySQL Installer setup wizard will open. Follow the steps below:

**Choose Setup Type**:

* + You will be presented with different setup types (Developer Default, Server only, Client only, Full, Custom).
  + For most users, the "Developer Default" option is sufficient. Select it and click "Next".

**Check Requirements**:

* + The installer will check for required software components. If any components are missing, the installer will prompt you to download and install them. Click "Execute" to install the required components.

**Installation**:

* + The installer will display the list of products to be installed. Click "Execute" to begin the installation process. This may take a few minutes.

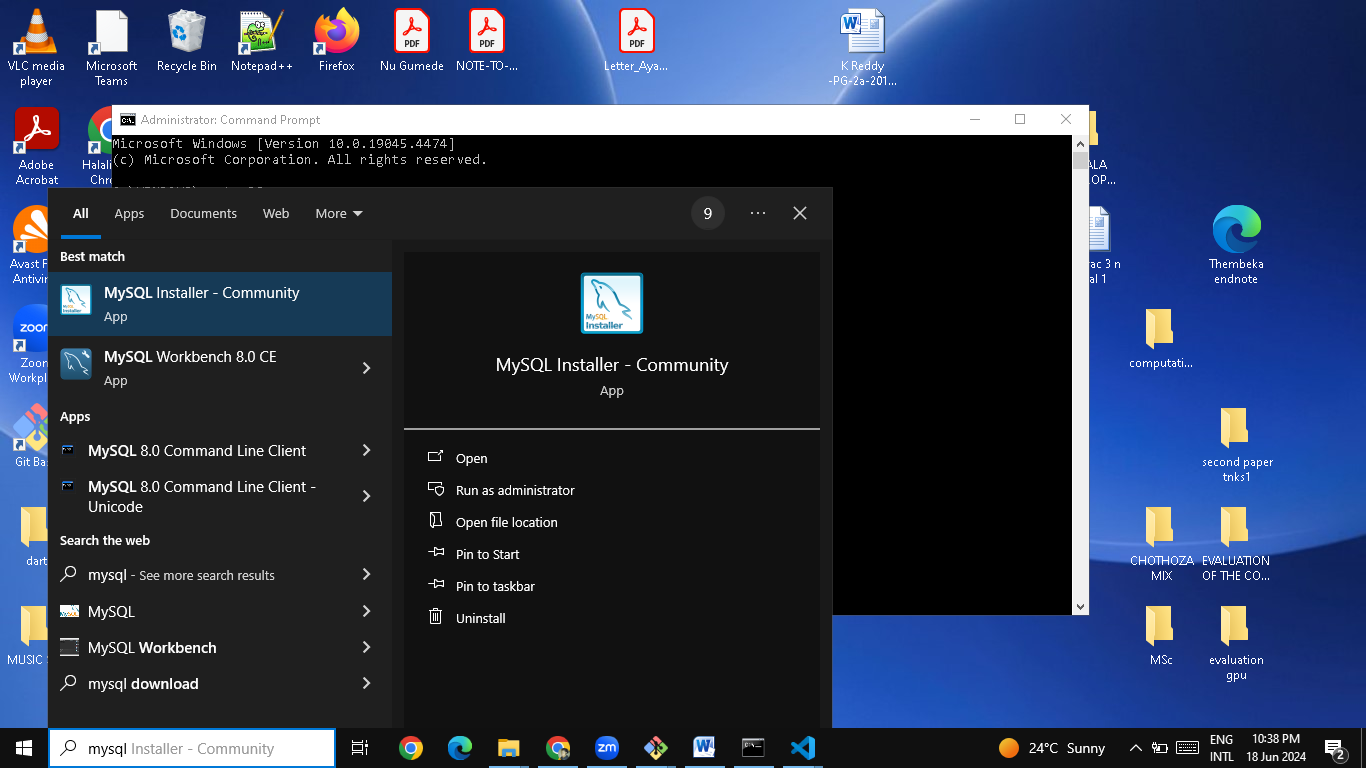
**Configuration**:

* + After the installation is complete, you will be guided through the configuration process.

1. **Configure MySQL Server**:
   * **Type and Networking**:
     + Select the "Standalone MySQL Server" configuration type. Click "Next".
     + Leave the default settings for networking (TCP/IP, Port 3306) and click "Next".
   * **Authentication Method**:
     + Choose the "Use Strong Password Encryption for Authentication (RECOMMENDED)" option and click "Next".
   * **Accounts and Roles**:
     + Set up the root account password. Make sure to remember this password as it will be required to access the MySQL server.
     + Optionally, you can create additional user accounts by clicking "Add User" and filling in the details. Click "Next" when done.
   * **Windows Service**:
     + Choose to run MySQL as a Windows service and set the service name. Ensure that "Start the MySQL Server at System Startup" is checked. Click "Next".
   * **Apply Configuration**:
     + Review the configuration settings and click "Execute" to apply them.
2. **Complete the Installation**:
   * Once the configuration is applied, click "Finish" to complete the installation process.

**Step 3: Verify MySQL Installation**

1. **Start MySQL Workbench**:
   * MySQL Workbench is a graphical user interface for working with MySQL databases. You can launch it from the Start menu.



1. **Connect to the MySQL Server**:
   * In MySQL Workbench, click on the "Local instance MySQL" to connect to your newly installed MySQL server.
   * Enter the root password you set during installation.
2. **Verify the Connection**:
   * If the connection is successful, you will see the MySQL Workbench dashboard where you can start creating and managing databases.

**Step 4: Additional Tools and Configuration (Optional)**

1. **Install MySQL Shell**:
   * MySQL Shell is a command-line interface for managing MySQL databases. You can install it via the MySQL Installer by selecting it from the available products.
2. **Configure Environment Variables**:
   * To access MySQL from the command line easily, you may want to add the MySQL binary directory to your system's PATH environment variable.
   * The default installation path is usually C:\Program Files\MySQL\MySQL Server 5.7\bin.
   * Add this path to the system's PATH environment variable via System Properties > Environment Variables.

**7 Installing Docker**

1. **Download Docker Desktop**:
   * Visit the Docker website and download Docker Desktop for Windows: Docker Desktop for Windows.
2. **Run the Installer**:
   * Once the download is complete, run the installer and follow the installation steps.
   * Ensure that you check the option to "Use the WSL 2 based engine" (this is recommended for better performance).
3. **Complete Installation**:
   * Follow the prompts to complete the installation and restart your computer if necessary.

**Step 2: Verify Docker Installation**

1. **Launch Docker Desktop**:
   * After installation, open Docker Desktop from the Start menu or desktop shortcut.
2. **Verify Docker Installation**:
   * Open Command Prompt or PowerShell and run: docker --version
   * This command should display the version of Docker installed.

**Step 3: Pull a Docker Image**

1. **Choose a Base Image**:
   * You can use Docker Hub to find official images for various programming languages and frameworks. For example, for a Python development environment, you might use the official Python image.
2. **Pull the Image**:
   * In Command Prompt or PowerShell, run: docker pull python:3.9
   * This command pulls the official Python 3.9 image from Docker Hub.

**Step 4: Create a Dockerfile**

1. **Create a Project Directory**:
   * Create a directory for your project and navigate into it:

mkdir my-python-app

cd my-python-app

1. **Create a Dockerfile**:
   * Inside your project directory, create a file named Dockerfile with the following contents: # Use the official Python image from the Docker Hub from python:3.9

# Set the working directory in the container

WORKDIR /app

# Copy the current directory contents into the container at /app

COPY . /app

# Install any needed packages specified in requirements.txt

RUN pip install --no-cache-dir -r requirements.txt

# Make port 80 available to the world outside this container

EXPOSE 80

# Define environment variable

ENV NAME World

# Run app.py when the container launches

CMD ["python", "app.py"]

1. **Create requirements.txt and app.py**:
   * Create a requirements.txt file with your project dependencies: flask
   * Create an app.py file with a simple Flask application: from flask import Flask

**Step 5: Build and Run the Docker Container**

1. **Build the Docker Image**:
   * In the project directory, build the Docker image using the Dockerfile: docker build -t my-python-app.
2. **Run the Docker Container**:
   * Run the Docker container from the image you just built.

**8 Extensions and plugins**

1. **Open VS Code**:
   * Launch Visual Studio Code.
2. **Access Extensions**:
   * Click on the Extensions icon in the Activity Bar on the side of the window or use the shortcut Ctrl+Shift+X.

**Step 2: Search for Extensions**

1. **Search Bar**:
   * In the Extensions view, you’ll see a search bar at the top. Type in keywords related to the functionality you’re looking for, such as "Python", "JavaScript", "linting", "Git", etc.
2. **Browse Extensions**:
   * Browse through the list of available extensions. Each extension listing shows the name, publisher, rating, and a brief description.

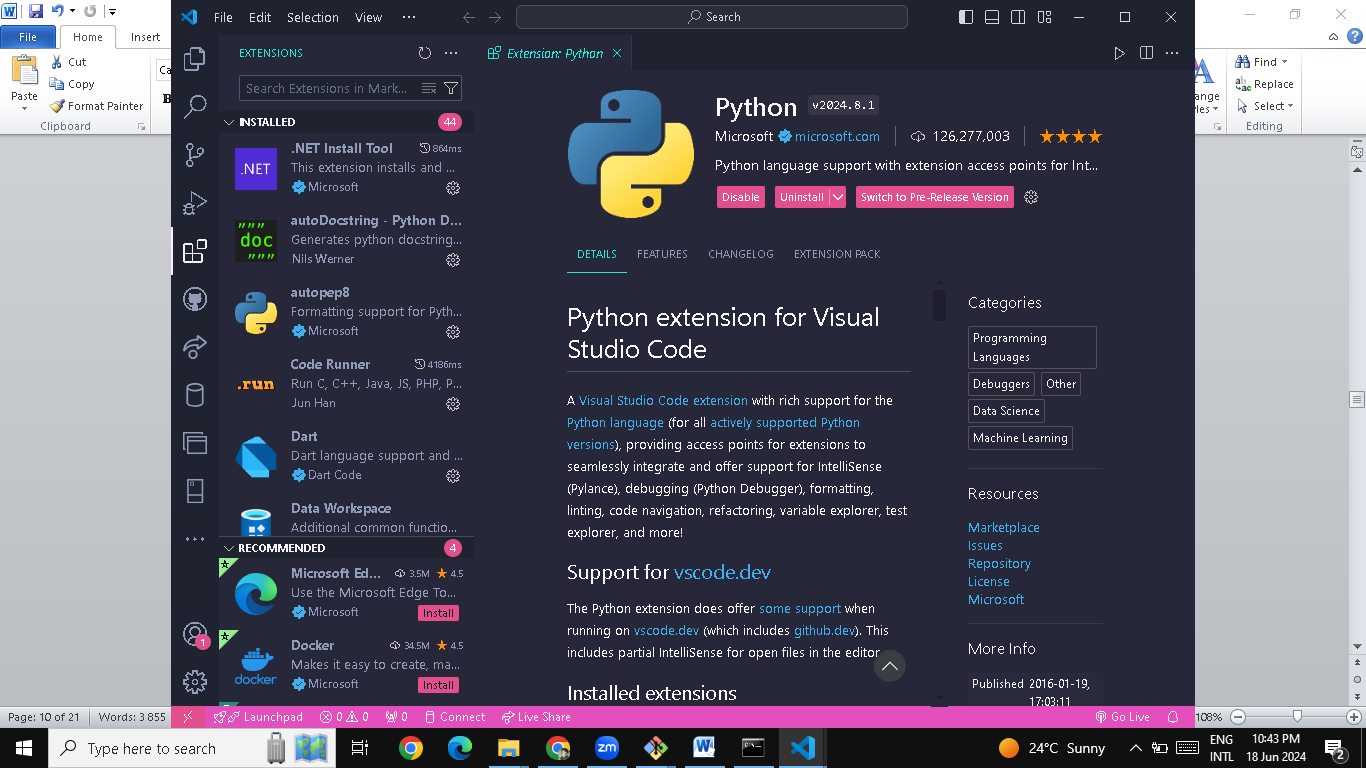
**Step 3: Install Extensions**

1. **Select an Extension**:
   * Click on an extension to view more details, including installation instructions, usage, and screenshots.
2. **Install**:
   * Click the Install button to add the extension to VS Code. Once installed, some extensions may require additional setup or configuration.

**Step 4: Popular Extensions for Various Functionalities**

Here are some popular extensions categorized by functionality:

1. **Syntax Highlighting**:
   * **Python**: [Python Extension for VS Code](https://marketplace.visualstudio.com/items?itemName=ms-python.python)
   * **JavaScript/TypeScript**: [ESLint](https://marketplace.visualstudio.com/items?itemName=dbaeumer.vscode-eslint)
2. **Linting**:
   * **ESLint**: Provides linting for JavaScript and TypeScript.
   * **Pylint**: Integrated with the Python extension to lint Python code.
3. **Code Formatting**:
   * **Prettier**: [Prettier - Code formatter](https://marketplace.visualstudio.com/items?itemName=esbenp.prettier-vscode)
   * **Black**: [Python formatting using Black](https://marketplace.visualstudio.com/items?itemName=ms-python.black-formatter)
4. **Version Control Integration**:
   * **GitLens**: [GitLens — Git supercharged](https://marketplace.visualstudio.com/items?itemName=eamodio.gitlens)
   * **GitHub Pull Requests and Issues**: [GitHub Pull Requests and Issues](https://marketplace.visualstudio.com/items?itemName=GitHub.vscode-pull-request-github)
5. **Debugging**:
   * **Python**: Integrated with the Python extension.
   * **Debugger for Chrome**: [Debugger for Chrome](https://marketplace.visualstudio.com/items?itemName=msjsdiag.debugger-for-chrome)
6. **Docker**:
   * [Docker](https://marketplace.visualstudio.com/items?itemName=ms-azuretools.vscode-docker): Provides tools for working with Docker containers.
7. **Database Management**:
   * **SQL Server**: [SQL Server (mssql)](https://marketplace.visualstudio.com/items?itemName=ms-mssql.mssql)
   * **MongoDB**: [MongoDB for VS Code](https://marketplace.visualstudio.com/items?itemName=mongodb.mongodb-vscode)
8. **Additional Enhancements**:
   * **Live Server**: [Live Server](https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer): Launches a local development server with live reload.
   * **Path Intellisense**: [Path Intellisense](https://marketplace.visualstudio.com/items?itemName=christian-kohler.path-intellisense): Autocompletes filenames.



**Step 5: Manage Installed Extensions**

1. **View Installed Extensions**:
   * In the Extensions view, you can see a list of installed extensions by clicking on the "Installed" tab.
2. **Disable or Uninstall Extensions**:
   * Hover over an installed extension and click on the gear icon to see options to disable or uninstall the extension.
3. **Extension Settings**:
   * Some extensions have configurable settings. You can access these by clicking the gear icon and selecting "Extension Settings".

**Step 6: Keep Extensions Updated**

1. **Automatic Updates**:
   * By default, VS Code will automatically update extensions when new versions are available. You can check for updates manually by clicking on the gear icon in the Extensions view and selecting "Check for Extension Updates".