**Setup Development Environment**

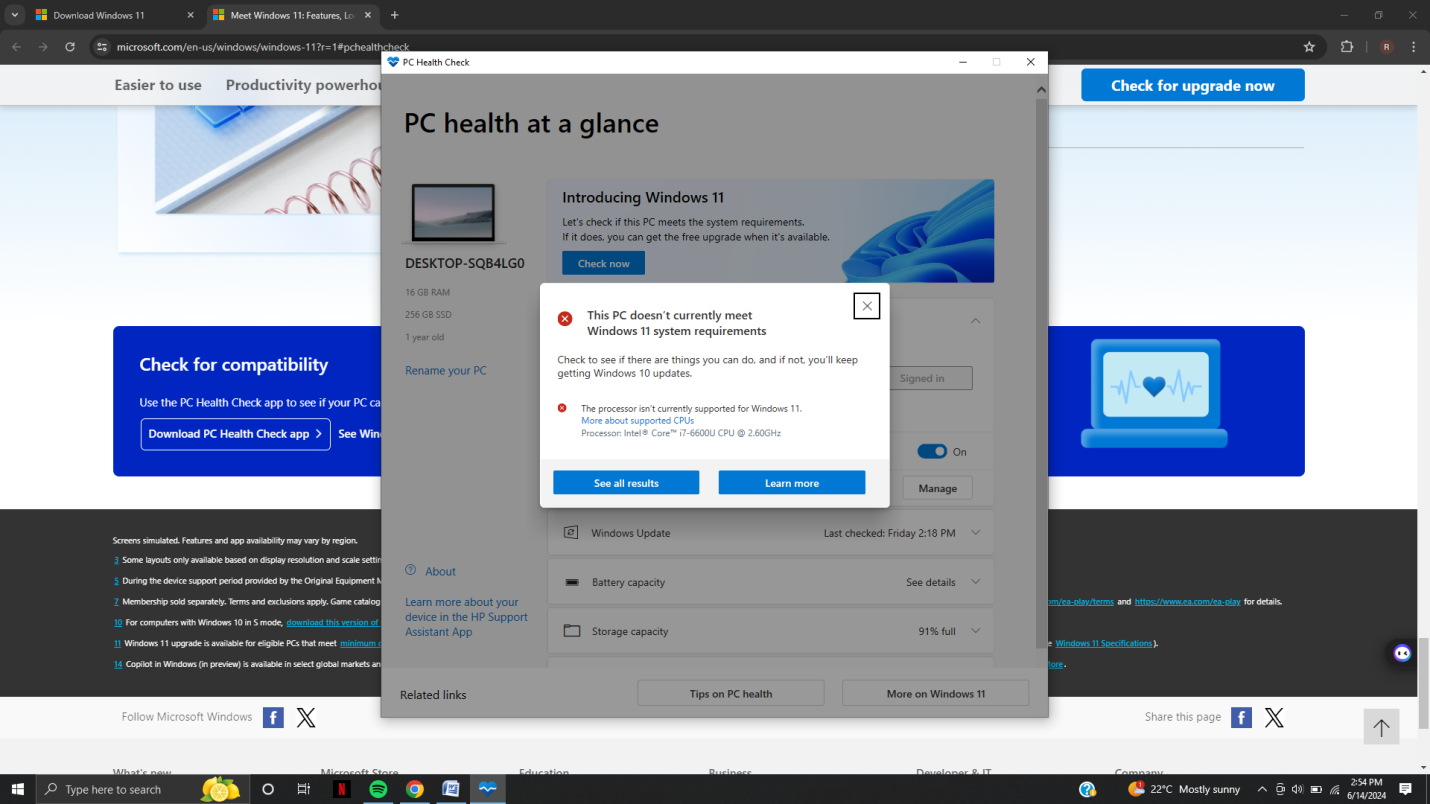
**Assignment: Setting Up Your Developer Environment**

Google docs link : <https://docs.google.com/document/d/1LlPlyBx6giu3CdDMA-QMsRQWMrSD0lSgbDFnXdM4ISk/edit?usp=sharing>

**Objective:** This assignment aims to familiarize you with the tools and configurations necessary to set up an efficient developer environment for software engineering projects. Completing this assignment will give you the skills required to set up a robust and productive workspace conducive to coding, debugging, version control, and collaboration.

**Tasks:**

1. Select Your Operating System (OS): Choose an operating system that best suits your preferences and project requirements. Download and Install Windows 11. <https://www.microsoft.com/software-download/windows11>



Obtain Windows 11 Installation Media

You can download Windows 11 using the following methods:

**Method 1: Windows 11 Installation Assistant**

Step 1: Visit the official Windows 11 download page.

Step 2: Click on the Download now button under the Windows 11 Installation Assistant.

Step 3: Run the downloaded Windows11InstallationAssistant.exe file.

Step 4: Follow the on-screen instructions to download and install Windows 11.

**Method 2: Create Windows 11 Installation Media**

Step 1: On the Windows 11 download page, click Download now under Create Windows 11 Installation Media.

Step 2: Run the MediaCreationToolW11.exe file.

Step 3: Accept the license terms and select your preferred language and edition.

Step 4: Choose the installation media (USB flash drive or ISO file).

If you select USB flash drive, ensure it has at least 8 GB of space and is empty.

If you select ISO file, save it on your PC and then burn it to a DVD or create a bootable USB.

**4. Install Windows 11**

Using Windows 11 Installation Assistant:

Step 1: Follow the on-screen instructions provided by the assistant to upgrade your current Windows version to Windows 11.

*Using Installation Media (USB/DVD):*

Step 1: Insert the USB flash drive or DVD into your PC.

Step 2: Restart your PC and boot from the USB or DVD.

You may need to press a key (like F12, Esc, Del, or a similar key) to enter the boot menu and select your boot device.

Step 3: Once booted, select your language preferences and click Next.

Step 4: Click Install now.

Step 5: Enter your product key if prompted (or select I don’t have a product key if upgrading).

Step 6: Choose the installation type:

Upgrade: Keeps your files, settings, and applications.

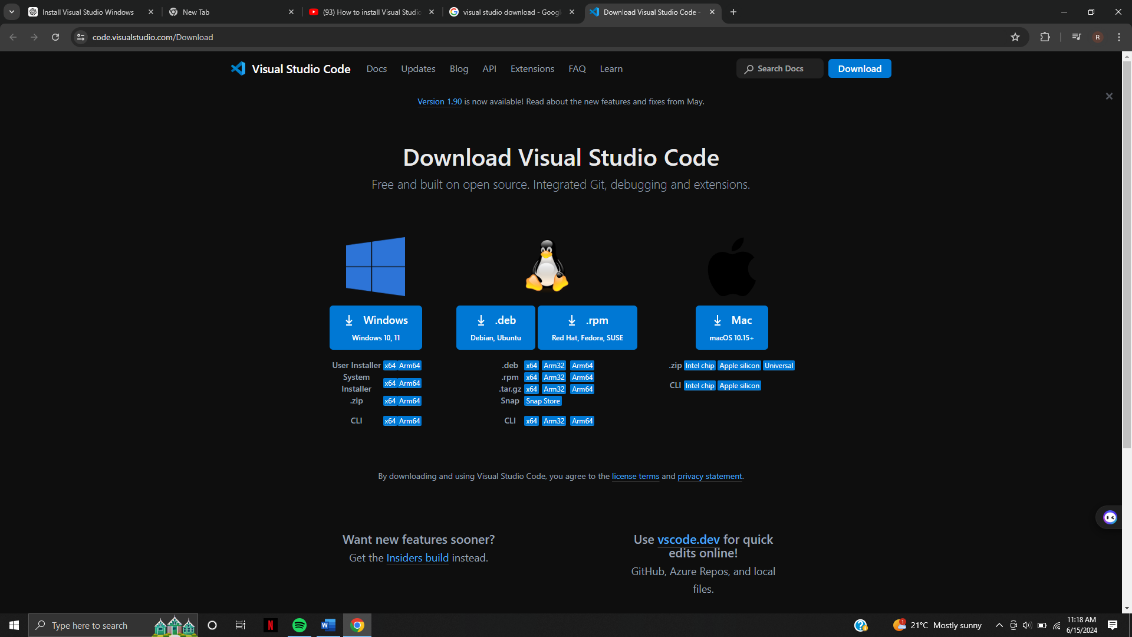
Custom: A clean install that doesn’t keep your files, settings, or applications.

Step 7: Select the drive where you want to install Windows 11 and click Next.

Step 8: Follow the on-screen instructions to complete the installation.

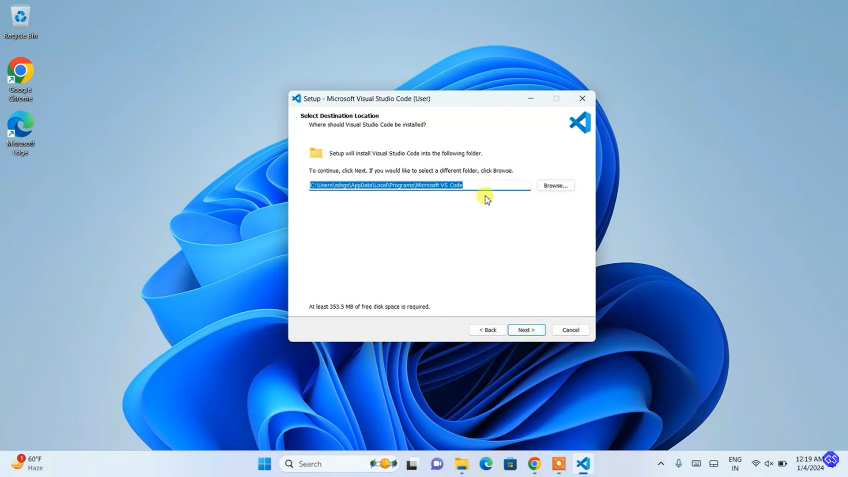
### Install a Text Editor or Integrated Development Environment (IDE): Select and install a text editor or IDE suitable for your programming languages and workflow. Download and Install Visual Studio Code. <https://code.visualstudio.com/Download> Step 1: Download Visual Studio Installer

1. Open Web Browser: Launch your preferred web browser (e.g., Chrome, Firefox, Edge).
2. Go to the Visual Studio Website: Navigate to the official Visual Studio download page by entering the URL: https://visualstudio.microsoft.com/
3. Select Version: On the Visual Studio homepage, click on the "Download" button under the edition you want to install. There are different editions available:
   1. Visual Studio Community: Free for individual developers, opensource projects, academic research, education, and small professional teams.
   2. Visual Studio Professional: For small teams with advanced features (requires a paid license).
   3. Visual Studio Enterprise: For larger teams with advanced features (requires a paid license).



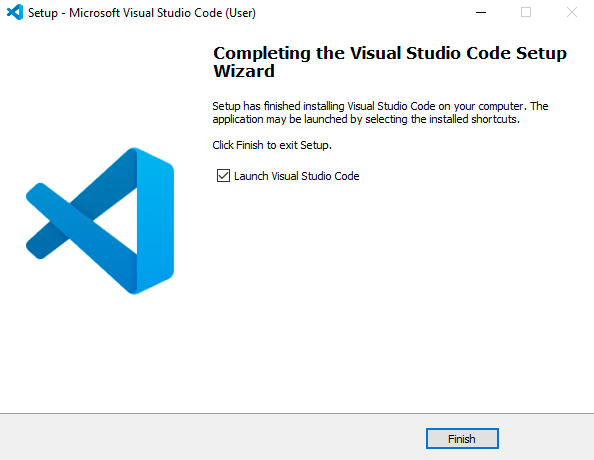
**Step 2: Run the Visual Studio Installer**

1. **Run the Installer**: Once the installer file is downloaded, open it. You might need to allow the installer to make changes to your device.
   * On your web browser, click the downloaded file (often located in the bottom left corner of the window).
   * If prompted by User Account Control, click "Yes" to allow the installer to run.



**Step 3: Choose Workloads**

1. **Workload Selection Screen**: After the installer initializes, you'll see a list of workloads, which are groups of related tools and libraries for different types of development.
   * **Example Workloads**: .NET desktop development, ASP.NET and web development, Desktop development with C++, Game development with Unity, etc.
2. **Select Workloads**: Check the boxes for the workloads that are relevant to your development needs. You can always modify your installation later to add or remove workloads.



**Step 4: Individual Components (Optional)**

1. **Individual Components Tab**: If you need more specific components, click on the "Individual components" tab.
   * Here, you can select individual components without selecting an entire workload.
   * This option is useful if you know exactly what you need.

**Step 5: Language Packs (Optional)**

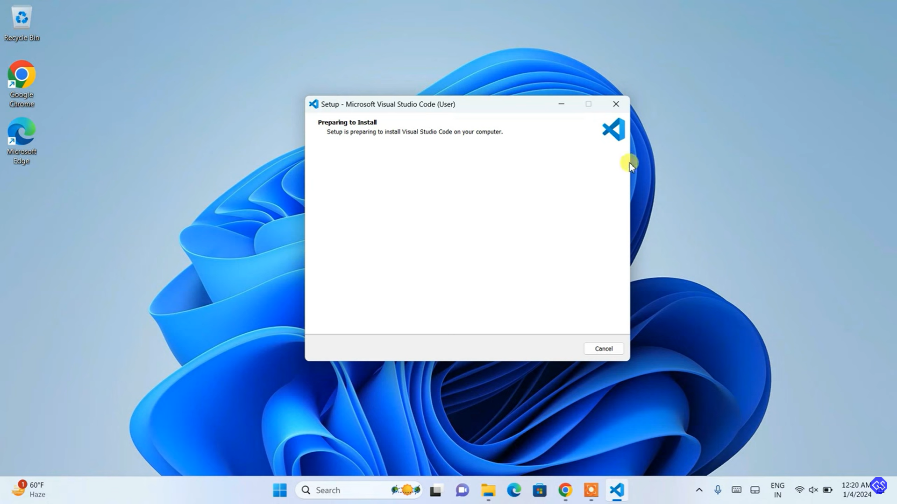
1. **Language Packs Tab**: If you prefer Visual Studio in a language other than English, you can select language packs from the "Language packs" tab.

**Step 6: Installation Location**

1. **Installation Path**: You can choose the installation locations for both Visual Studio and the download cache. The default locations are usually fine, but you can change them if needed.

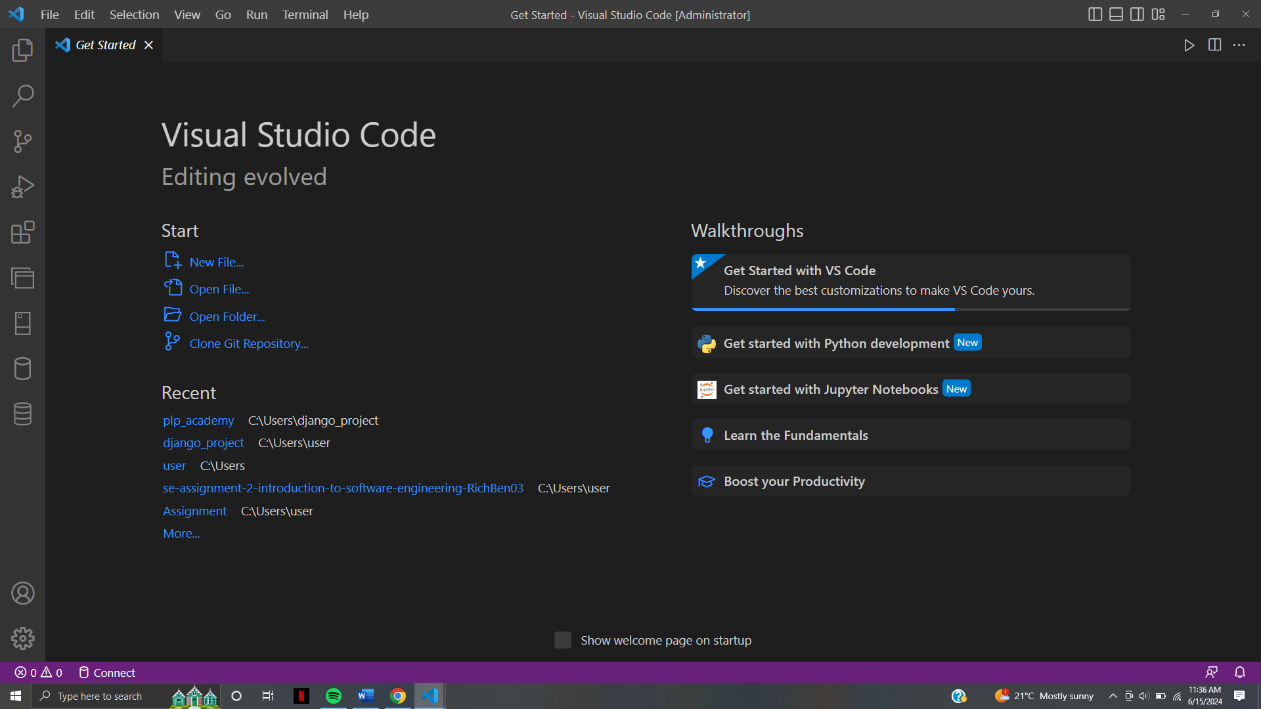
**Step 7: Install**

* 1. **Install Button**: After making your selections, click the "Install" button. The installer will start downloading and installing the selected components. This process may take some time depending on your internet speed and the number of components selected.
  2. **Optional Steps**: During the installation, you can continue working on your computer. Visual Studio installer allows you to pause and resume the installation process.



**Step 8: Launch Visual Studio**

1. **Complete Installation**: Once the installation is complete, you will see a "Launch" button. Click it to start Visual Studio.
2. **Sign In**: When you first launch Visual Studio, you might be prompted to sign in with your Microsoft account. This step is optional for the Community edition but required for Professional and Enterprise editions.
3. **Start Developing**: After signing in (if required), you can start creating or opening projects and developing your applications.



1. Set Up Version Control System: Install Git and configure it on your local machine. Create a GitHub account for hosting your repositories. Initialize a Git repository for your project and make your first commit. <https://github.com>

**Step 1: Install Git**

**Download Git**: Go to the official Git website: https://git-scm.com/

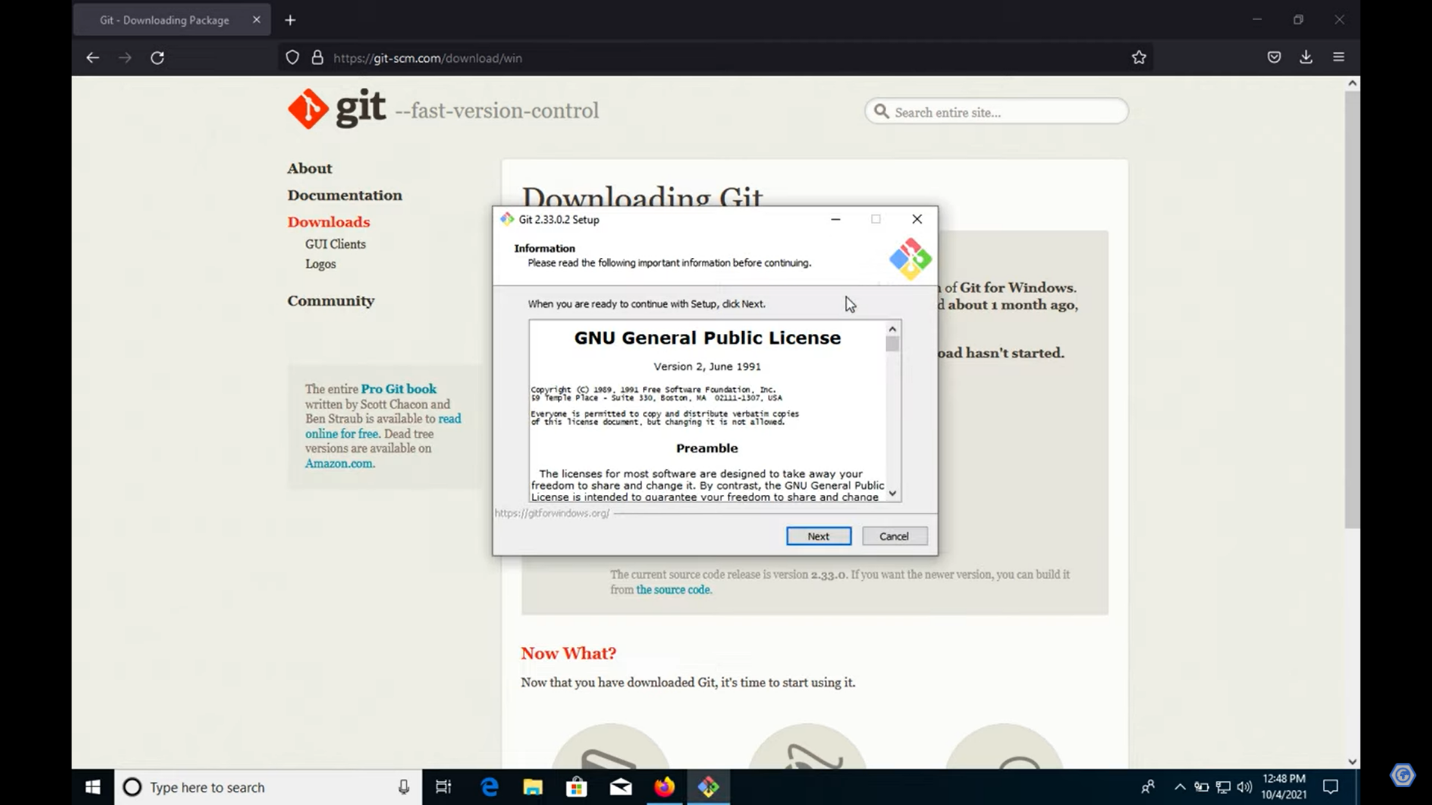
**Download Git Installer**: Click on the "Download" button for your operating system (Windows, macOS, or Linux).

* + For Windows, you’ll download an .exe file.



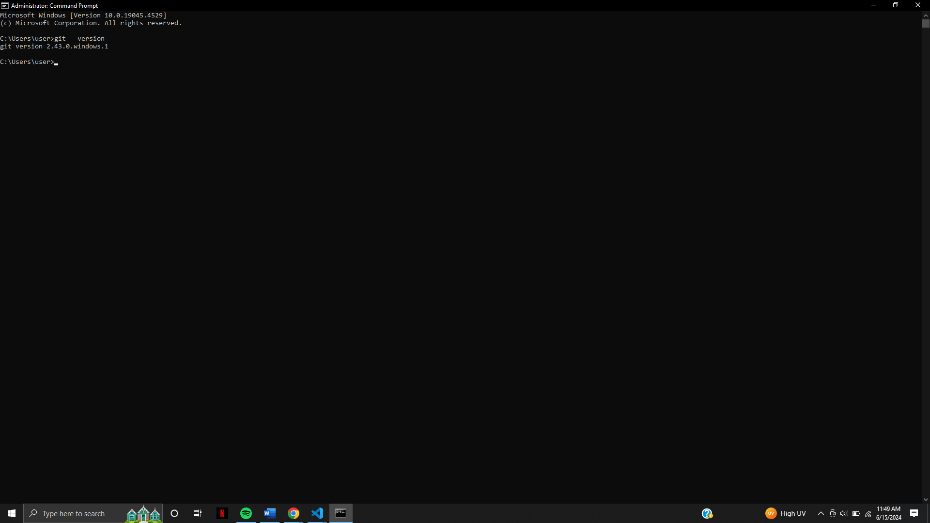
**Run the Installer**: Open the downloaded installer file.

* + Follow the prompts in the installation wizard.
  + You can keep the default options unless you have specific requirements.



**Verify Installation**:

* + Open a command prompt (Windows) or terminal (macOS/Linux).
  + Type git --version and press Enter. You should see the version of Git installed.



**Step 2: Configure Git**

**Open Command Prompt/Terminal**: Open a command prompt (Windows) or terminal (macOS/Linux).

**Set Your Username**:

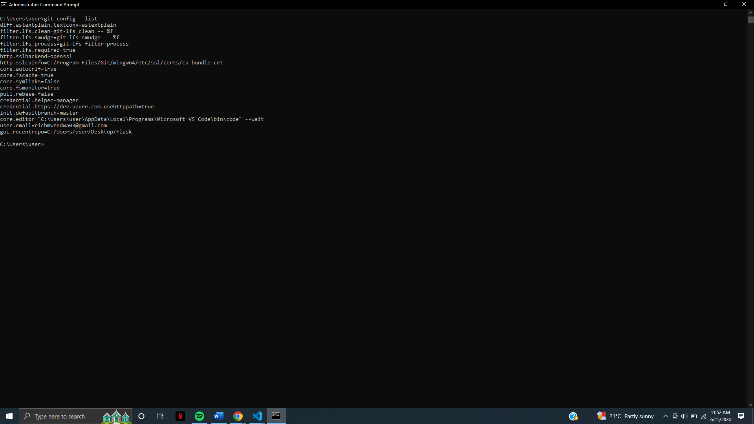
* + Run the command: git config --global user.name "Your Name"
  + Replace "Your Name" with your actual name.

**Set Your Email**:

* + Run the command: git config --global user.email "your-email@example.com"
  + Replace "your-email@example.com" with your actual email address.

**Verify Configuration**:

* + Run the command: git config --list
  + This will display your configuration settings.



**Step 3: Create a GitHub Account**

**Go to GitHub Website**: Open your web browser and go to https://github.com/

**Sign Up**: Click on the "Sign up" button.

* + Follow the prompts to create a new account. You’ll need to provide a username, email address, and password.

**Verify Email**: GitHub will send you a verification email. Check your email and follow the instructions to verify your account.

**Step 4: Create a New Repository on GitHub**

**Log In**: Log in to your GitHub account.

**Create Repository**:

* + Click the "+" icon in the top right corner and select "New repository".
  + Fill in the repository details: repository name, description (optional), and choose between public or private.
  + Optionally, you can add a README file, a .gitignore file, and a license.
  + Click "Create repository".

**Step 5: Initialize a Git Repository Locally**

**Open Command Prompt/Terminal**: Open a command prompt (Windows) or terminal (macOS/Linux).

**Navigate to Project Directory**:

* + Use the cd command to navigate to the directory where your project is located. For example: cd path/to/your/project

**Initialize Git Repository**:

* + Run the command: git init
  + This initializes a new Git repository in your project directory.

**Step 6: Make Your First Commit**

**Add Files to Staging Area**:

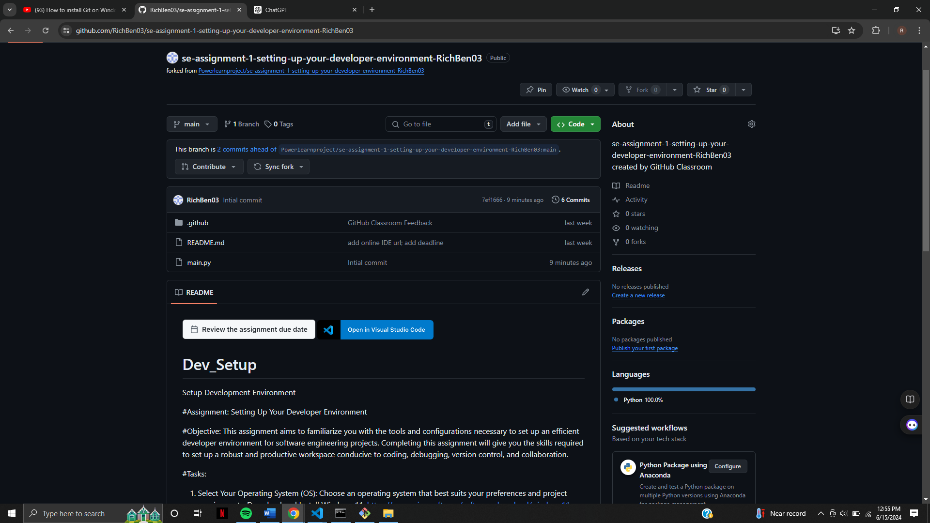
* + Run the command: git add .
  + This adds all files in the current directory to the staging area.

**Commit Files**:

* + Run the command: git commit -m "Initial commit"
  + This commits the files to the repository with the message "Initial commit".

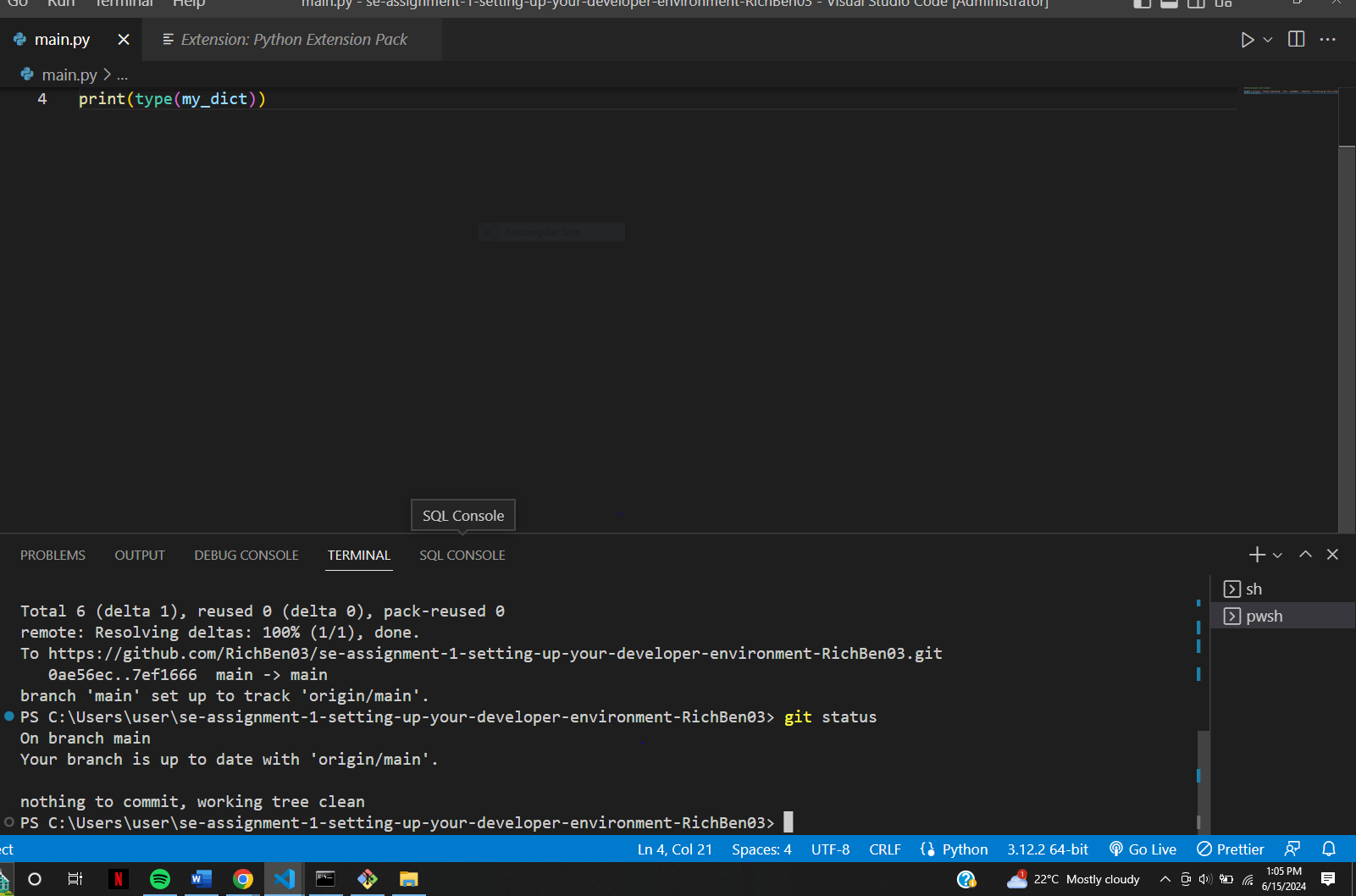
**Step 7: Connect Local Repository to GitHub**

**Add Remote Repository**:

* + Go back to your GitHub repository page.
  + Copy the repository URL (HTTPS or SSH).
  + In your command prompt/terminal, run: git remote add origin <repository-url>
  + Replace <repository-url> with the URL you copied.
  + 

**Push Changes to GitHub**:

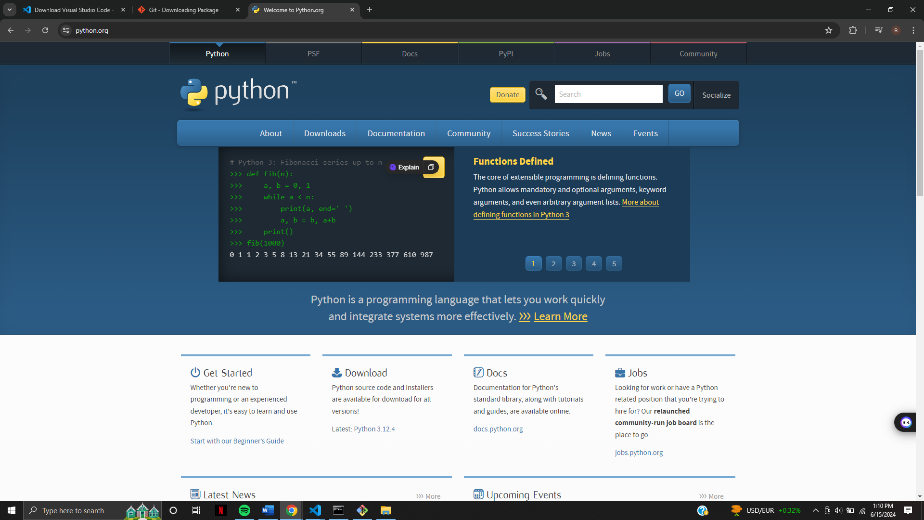
* + Run the command: git push -u origin master
  + This pushes your local repository to GitHub and sets the remote repository as the default for future pushes.



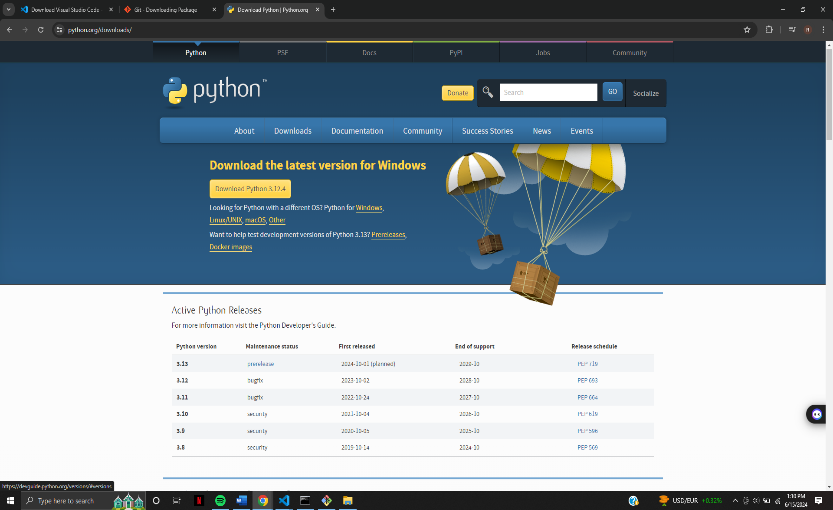
1. Install Necessary Programming Languages and Runtimes: Instal Python from http://wwww.python.org programming language required for your project and install their respective compilers, interpreters, or runtimes. Ensure you have the necessary tools to build and execute your code.

**Step 1: Install Python**

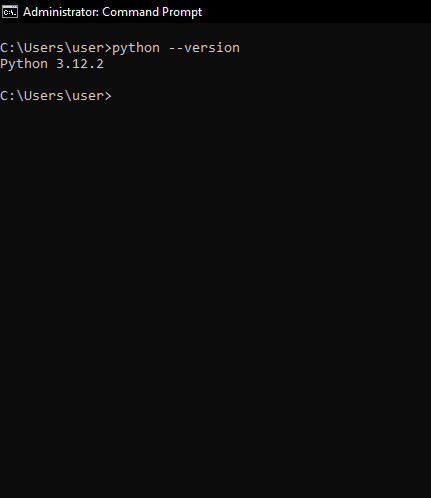
1. **Go to Python's Official Website**:
   1. Open your web browser and go to <https://www.python.org/>



1. **Download Python Installer**:
   1. Click on the "Downloads" tab.
   2. The page should automatically detect your operating system (Windows, macOS, or Linux).
   3. Click on the "Download Python [version]" button to download the installer.



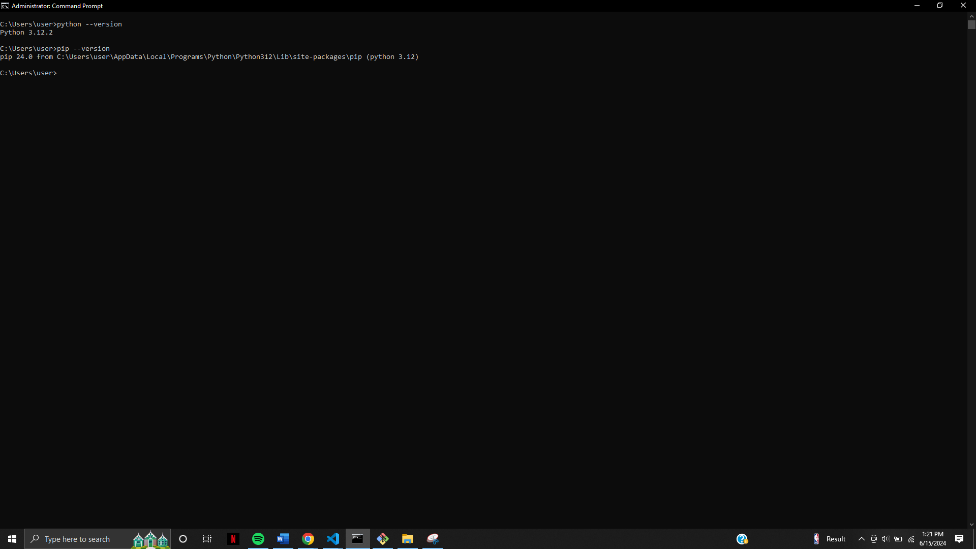
1. **Run the Python Installer**:
   1. Open the downloaded installer file.
   2. **Important**: Check the box that says "Add Python to PATH".
   3. Click "Install Now" or choose "Customize installation" if you want to select specific features.
2. **Verify Installation**:
   1. Open a command prompt (Windows) or terminal (macOS/Linux).
   2. Type python --version and press Enter. You should see the version of Python you installed.
   3. Type pip --version to verify that pip (Python's package manager) was installed.



1. Install Package Managers: If applicable, install package managers like pip (Python).

### Step 2: Install Necessary Packages with pip

1. **Open Command Prompt/Terminal**:
   * Open a command prompt (Windows) or terminal (macOS/Linux).
2. **Install Packages**:
   * **Use pip** to install any necessary packages. For example, to install the requests library, type: pip install requests
   * You can install multiple packages by listing them separated by spaces: pip install requests flask numpy

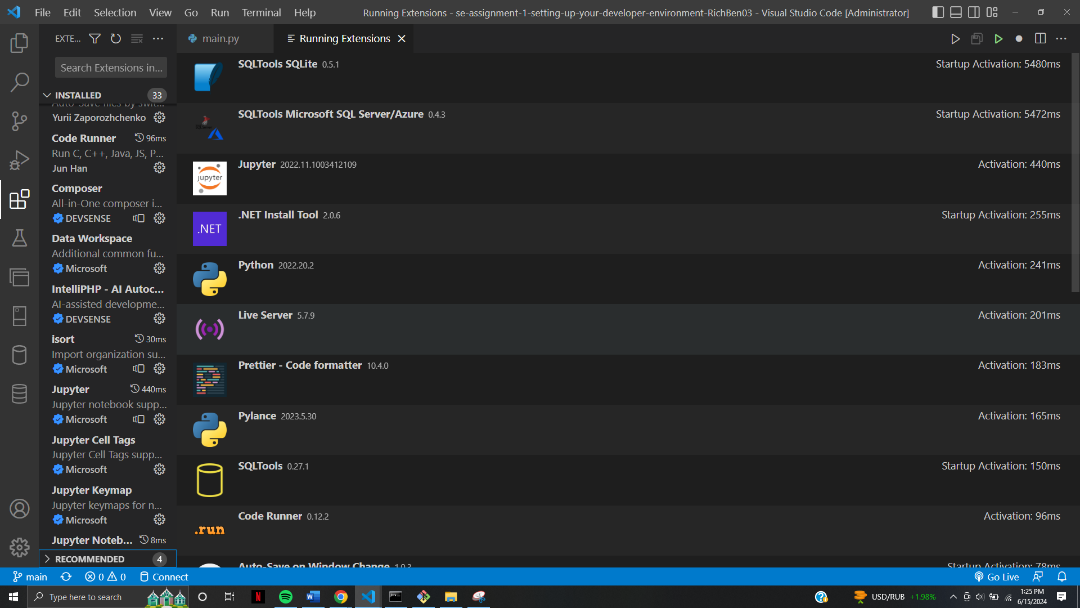


1. Configure a Database (MySQL): Download and install MySQL database. <https://dev.mysql.com/downloads/windows/installer/5.7.html>
2. **Go to MySQL Download Page**:
   * Open your web browser and go to https://dev.mysql.com/downloads/windows/installer/5.7.html
3. **Download MySQL Installer**:
   * Scroll down to the "MySQL Installer" section.
   * Choose the appropriate installer (usually the "Windows (x86, 32-bit), MSI Installer").
   * Click "Download".
4. **Run the MySQL Installer**:
   * Open the downloaded installer file.
   * Choose the setup type (e.g., Developer Default, Server only).
   * Click "Next" and follow the prompts.
   * If you chose "Developer Default", it will install MySQL Server, MySQL Workbench, and other tools.
5. **Configure MySQL Server**:
   * During the installation, you will be prompted to configure the MySQL Server.
   * Choose a configuration type (usually "Standalone MySQL Server").
   * Set the root password and create additional users if needed.
   * Click "Next" to complete the configuration.
6. **Start MySQL Server**:
   * The installer should start the MySQL server automatically. You can verify this by opening MySQL Workbench or running the mysql command in a terminal.

**Step 4: Verify MySQL Installation**

1. **Open MySQL Workbench**:
   * Open MySQL Workbench from the Start menu (Windows) or Applications folder (macOS).
2. **Connect to Local Instance**:
   * Click on "Local instance MySQL" to connect to the MySQL server.
   * Enter the root password you set during installation.
   * You should see the MySQL Workbench interface with your database instance.

**Step 5: Create a Database**

1. **Open MySQL Workbench**:
   * Open MySQL Workbench and connect to your MySQL server.
2. **Create a New Database**:
   * Click on the "Schemas" tab.
   * Right-click in the Schemas panel and select "Create Schema".
   * Enter a name for your new database and click "Apply".
3. **Create Tables**:
   * You can now create tables within your new database using the Workbench interface or by running SQL commands.
4. Set Up Development Environments and Virtualization (Optional): Consider using virtualization tools like Docker or virtual machines to isolate project dependencies and ensure consistent environments across different machines.  
   
5. Explore Extensions and Plugins: Explore available extensions, plugins, and add-ons for your chosen text editor or IDE to enhance functionality, such as syntax highlighting, linting, code formatting, and version control integration.  
   

Document Your Setup: Create a comprehensive document outlining the steps you've taken to set up your developer environment. Include any configurations, customizations, or troubleshooting steps encountered during the process.

#Deliverables:

Document detailing the setup process with step-by-step instructions and screenshots where necessary.

A GitHub repository containing a sample project initialized with Git and any necessary configuration files (e.g., .gitignore).

A reflection on the challenges faced during setup and strategies employed to overcome them.

#Submission: Submit your document and GitHub repository link through the designated platform or email to the instructor by the specified deadline.

#Evaluation Criteria:\*\*

Completeness and accuracy of setup documentation.

Effectiveness of version control implementation.

Appropriateness of tools selected for the project requirements.

Clarity of reflection on challenges and solutions encountered.

Adherence to submission guidelines and deadlines.

Note: Feel free to reach out for clarification or assistance with any aspect of the assignment.