**Installation Guide for Windows 11**

**1. Check System Requirements**

Before starting, ensure your PC meets the minimum system requirements for Windows 11. These typically include a compatible 64-bit processor, 4GB RAM, 64GB storage, UEFI firmware with Secure Boot capability, TPM version 2.0, and a DirectX 12 compatible graphics card.

**2. Backup Your Data**

It's crucial to back up all important data on your computer to prevent data loss during the installation process.

**3. Download Windows 11**

Go to the official Microsoft website or use the Windows Update tool to download the Windows 11 installation files.

**4. Create Installation Media**

You can create a bootable USB drive using the Media Creation Tool provided by Microsoft. This tool helps in preparing the installation media for Windows 11.

**5. Prepare Your PC**

Insert the bootable USB drive into your PC.

Restart your computer.

**6. Boot from USB Drive**

During startup, enter the BIOS/UEFI settings of your PC (usually by pressing Del, F2, or Esc key).

Set the boot order to prioritize the USB drive.

**7. Install Windows 11**

Once the PC boots from the USB drive, follow the on-screen instructions to begin the installation of Windows 11.

Choose your language, time, and keyboard preferences.

**8. Select Installation Type**

Click on "Install now" to start the installation process.

**9. Enter Product Key**

If prompted, enter your Windows 11 product key. You can also skip this step and enter it later.

**10. Accept License Terms**

Read and accept the Microsoft Software License Terms.

**11. Choose Installation Type**

Select the type of installation you want (e.g., Upgrade or Custom installation).

**12. Select Drive**

Choose the drive where you want to install Windows 11.

**13. Install Windows 11**

Windows will now install. Your PC will restart several times during this process.

**14. Set Up Your PC**

Follow the on-screen prompts to customize your settings, such as creating user accounts and connecting to the internet.

**15. Install Drivers and Updates**

After installation, install necessary drivers for your hardware and update Windows 11 to ensure it's running smoothly.

**16. Restore Data**

Restore your backed-up data onto your new Windows 11 installation.

**Installation Of Visual Studio Code on Windows 11**

**1. Download Visual Studio Code Installer**

Open your web browser and go to the official Visual Studio Code website: Visual Studio Code.

Click on the "Download for Windows" button to download the installer.

**2. Run the Installer**

Once the download is complete, locate the downloaded installer file (typically named something like VSCodeSetup.exe) and double-click to run it.

**3. User Account Control (UAC) Prompt**

If prompted by User Account Control, click "Yes" to allow the installer to make changes to your device.

**4. Setup Wizard**

The Visual Studio Code Setup Wizard will open. Click on "Next" to proceed.

**5. Select Installation Location**

Choose the destination folder where you want to install Visual Studio Code or leave the default location. Click on "Next".

**6. Select Start Menu Folder**

Choose whether to create a Start Menu folder for Visual Studio Code shortcuts or leave the default option. Click on "Next".

**7. Additional Tasks**

Optionally, select any additional tasks you want, such as creating a desktop shortcut or adding VS Code to the PATH. Click on "Next".

**8. Install**

Click on "Install" to begin the installation process.

**9. Installation Progress**

Wait for the installation to complete. This may take a few moments.

**10. Completion**

Once the installation is finished, click on "Finish" to exit the Setup Wizard.

**11. Launch Visual Studio Code**

Visual Studio Code is now installed on your Windows 11 PC. You can launch it from the desktop shortcut or from the Start Menu.

**12. Update Extensions (Optional)**

Upon launching, VS Code may prompt you to install recommended extensions or updates. Follow the prompts to customize your installation further.

**Version Control System on Windows 11**

**1. Install Git**

Download and Install Git

Go to the official Git website: Git Downloads.

Click on the appropriate download link for Windows. This will download the Git installer (Git-2.XX.X.X-64-bit.exe).

Double-click the downloaded installer to launch it.

Follow the prompts in the installer:

Click "Next" on the first screen.

Choose the destination folder (leave as default is recommended).

Select components (leave as default unless you have specific needs).

Choose the default editor (usually Visual Studio Code or Nano).

Adjust your PATH environment (choose "Git from the command line and also from 3rd-party software").

Choose HTTPS transport backend.

Configure the line endings conversions (choose the default).

Click "Install" to begin the installation process.

Wait for the installation to complete. Once done, click "Finish".

**2. Configure Git**

Set Up Your Identity

Open Git Bash (installed with Git) from the Start Menu or desktop shortcut.

Configure your username and email by running these commands in Git Bash, replacing Your Name and your\_email@example.com with your details:

bash

Copy code

git config --global user.name "Your Name"

git config --global user.email "your\_email@example.com"

Verify Configuration

Verify that Git has been installed correctly and configured by running:

bash

Copy code

git --version

git config --global --list

This should display the Git version and your configured settings.

**3. Create a GitHub Account**

Go to GitHub and sign up for a free account if you don't already have one.

Follow the prompts to set up your account with a username, password, and email address.

**4. Initialize a Git Repository**

Create a New Project Folder

Create a new directory on your computer where you want to initialize your Git repository. You can do this through File Explorer or Git Bash:

bash

Copy code

mkdir myproject

cd myproject

Initialize Git Repository

Inside your project directory, initialize a Git repository:

bash

Copy code

git init

**5. Make Your First Commit**

Create a Sample File

Create a new file within your project directory. You can do this by right-clicking inside the directory, selecting "New" > "Text Document," and naming it README.md.

Add and Commit Changes

Add this file to your Git repository and make your first commit:

bash

Copy code

git add README.md

git commit -m "Initial commit"

Replace "Initial commit" with a meaningful commit message describing the changes you've made.

**6. Connect Git Repository to GitHub**

Create a New Repository on GitHub

Log in to your GitHub account.

Click on the "+" sign in the top right corner and select "New repository".

Give your repository a name, optionally add a description, choose public or private, and click "Create repository".

Connect Local Repository to GitHub

Follow the instructions under the "…or push an existing repository from the command line" section in your GitHub repository's page:

bash

Copy code

git remote add origin https://github.com/your\_username/your\_repository.git

git branch -M main

git push -u origin main

Replace your\_username with your GitHub username and your\_repository with the name of your repository.

**7. Verify on GitHub**

Refresh your GitHub repository page. You should see your README.md file and the commit message you added.

**Python Installation on Windows 11**

**1. Download Python Installer**

Go to the official Python website: Python Downloads.

Download the latest stable version of Python for Windows by clicking on the "Download Python X.X.X" button. As of now, Python 3.10 is the latest stable version.

**2. Run the Python Installer**

Once the download is complete, locate the downloaded installer file (python-X.X.X.exe, where X.X.X is the version number) and double-click to run it.

The Python Installation Wizard will open.

**3. Install Python**

On the first screen of the wizard, make sure to check the box that says "Add Python X.X to PATH". This option ensures that Python is added to your system PATH, allowing you to run Python from the command line easily.

Click on "Install Now" to start the installation process.

**4. Wait for Installation to Complete**

The installer will now install Python on your system. This process may take a few moments.

**5. Verify Installation**

Once the installation is complete, you can verify it by opening a Command Prompt or PowerShell window and typing:

bash

Copy code

python --version

This command should display the installed Python version.

**6. Install Additional Tools (Optional)**

Depending on your development needs, you may want to install additional tools such as:

pip: Python's package installer. It should be installed by default with Python 3.4 and above.

Python IDEs or Editors: Tools like Visual Studio Code, PyCharm, or IDLE for a more integrated development environment.

Compilers or Interpreters: Install compilers or interpreters like Cython, PyPy, or IronPython for specific Python development tasks.

**7. Set Up Environment Variables (Optional)**

If not already set during installation, you may need to manually set up environment variables:

Right-click on "This PC" or "My Computer" on your desktop or in File Explorer.

Select "Properties" > "Advanced system settings" > "Environment Variables".

Under "System variables", edit the "Path" variable to include the directory where Python is installed (e.g., C:\PythonXX or C:\Users\YourUsername\AppData\Local\Programs\Python\PythonXX).

**8. Test Python Installation**

Open a new Command Prompt or PowerShell window and run a simple Python script to ensure everything is set up correctly:

bash

Copy code

python

This will open the Python interactive shell (>>>). You can type Python code here to test.

python

Copy code

print("Hello, Python!")

Press Ctrl+Z followed by Enter (or Ctrl+D on macOS/Linux) to exit the Python shell.

**pip Installation on Windows 11**

**1. Ensure Python is Installed**

Before installing pip, ensure Python is installed on your system. You can follow the earlier provided steps to install Python on Windows 11 if you haven't already.

**2. Download get-pip.py**

Open your web browser and go to the get-pip.py page.

Right-click on the page and select "Save As" to download get-pip.py to your computer.

**3. Open Command Prompt or PowerShell**

Open Command Prompt or PowerShell with administrative privileges. This ensures you can install pip system-wide.

**4. Navigate to the Directory**

Use cd command to navigate to the directory where you saved get-pip.py. For example:

bash

Copy code

cd C:\Users\YourUsername\Downloads

**5. Install pip**

Run the following command to install pip:

bash

Copy code

python get-pip.py

This command will execute get-pip.py script using Python, which installs pip along with its dependencies.

**6. Verify Installation**

After installation, you can verify pip installation by checking its version:

bash

Copy code

pip --version

This command should display the installed pip version.

**7. Update pip (Optional)**

It's a good practice to update pip to the latest version:

bash

Copy code

python -m pip install --upgrade pip

**8. Test pip**

You can test pip by installing a package. For example, install numpy:

bash

Copy code

pip install numpy

This command installs the numpy package. You can replace numpy with any other package you wish to install.

**MySQL Database Installation on Windows 11**

**1. Download MySQL Installer**

Go to the official MySQL website: MySQL Downloads.

Download the MySQL Installer for Windows (typically the "MySQL Installer for Windows" version).

**2. Run the MySQL Installer**

Locate the downloaded installer file (mysql-installer-web-community-X.X.X.X.msi) and double-click to run it.

**3. MySQL Installer Setup**

The MySQL Installer window will open. Click on "Install MySQL Products".

**4. Choose Setup Type**

Select "Server only" or "Developer Default" setup type depending on your requirements. Click "Next".

**5. MySQL Server Setup**

Accept the license terms and click "Next".

**6. Choose a Setup Type**

Select "Standalone MySQL Server / Classic MySQL Replication". Click "Next".

**7. Check Requirements**

MySQL Installer will check if all the requirements are met. If there are any warnings or errors, address them before proceeding.

**8. Installation**

Click "Execute" to start the installation process. This will download and install MySQL Server and other necessary components.

**9. Configure MySQL Server**

Once the installation is complete, click on "Next" to configure MySQL Server.

**10. Configuration Type**

Select "Standalone MySQL Server / Classic MySQL Replication". Click "Next".

**11. Type and Networking**

Choose "Use Strong Password Encryption" and set a root password for MySQL. Click "Next".

**12. Windows Service**

Choose whether to configure MySQL Server as a Windows Service. Click "Next".

**13. Apply Configuration**

Review the configuration summary and click "Execute" to apply the configuration.

**14. Complete Installation**

Once configuration is complete, click "Finish" to exit the MySQL Installer.

**15. Verify Installation**

Open Command Prompt or PowerShell and navigate to MySQL bin directory (e.g., C:\Program Files\MySQL\MySQL Server X.X\bin).

Log in to MySQL server using the following command:

bash

Copy code

mysql -u root -p

Enter the root password when prompted. You should see the MySQL prompt (mysql>), indicating that you have successfully logged in to MySQL Server.

**Installation of Extensions in Visual Studio Code**

**1. Open Visual Studio Code**

Launch Visual Studio Code from the Start Menu or desktop shortcut.

**2. Open Extensions View**

Click on the Extensions icon in the Activity Bar on the side of the window, or press Ctrl+Shift+X (Cmd+Shift+X on macOS).

**3. Search for Extensions**

In the Extensions view, you can search for extensions by name or functionality using the search bar.

**4. Install an Extension**

Find an extension you want to install and click on the "Install" button next to it.

Alternatively, you can install an extension by clicking on the extension and then clicking "Install".

**5. Manage Installed Extensions**

After installation, the "Install" button changes to "Installed". You can manage your installed extensions by disabling, enabling, uninstalling, or updating them as needed.

**6. Recommended Extensions**

VS Code also suggests recommended extensions based on the files you open and the languages you use. You can explore these recommendations in the Extensions view.

**7. Popular Extensions**

Explore popular extensions by clicking on the "Popular" tab in the Extensions view. These extensions are widely used and cover a variety of functionalities.

**8. Themes and Color Themes**

You can also install themes and color themes to customize the appearance of VS Code. These are available under the "Themes" section in the Extensions view.