Comprehensive Documentation for Setting Up Development Environment

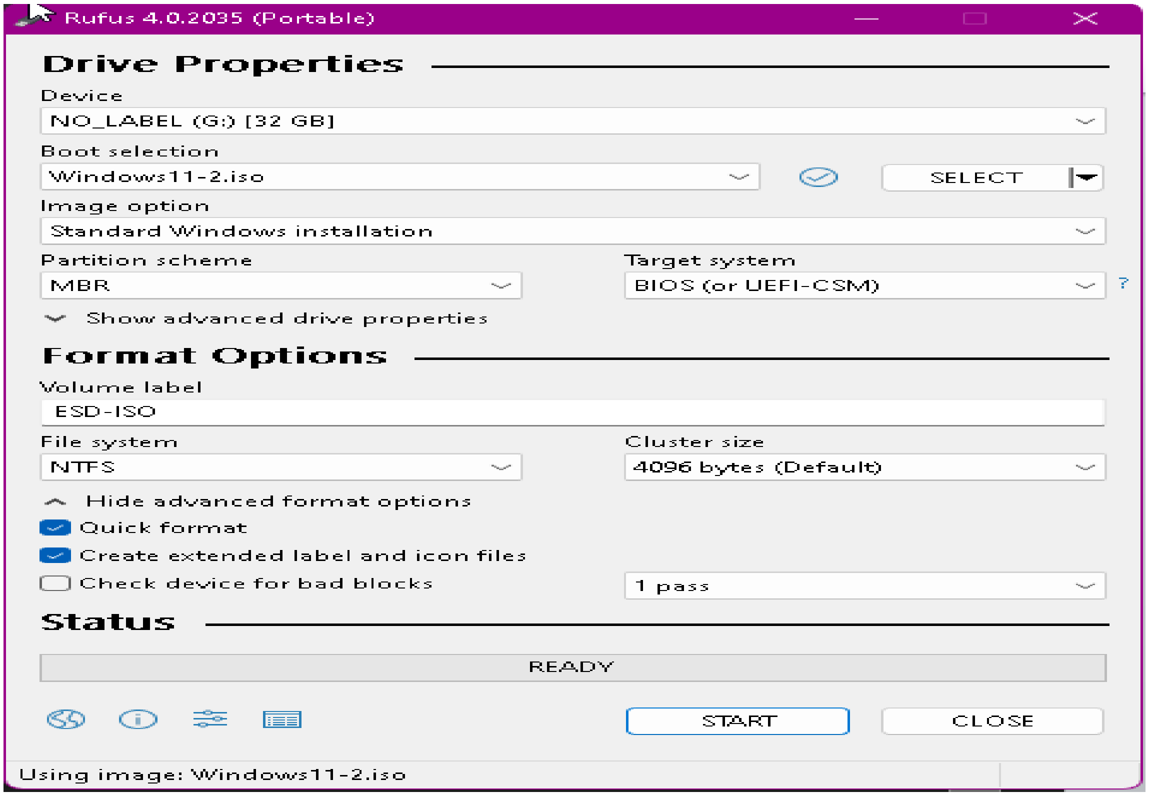
Table of Contents

Introduction

This document provides a comprehensive guide of the process I used to set up a development environment, focusing on installing necessary tools and configuring systems to enhance productivity. Below is the step-by-step illustration.

Installing Windows 11

1. Downloading of Windows 11:
   a) I visited the official Microsoft website: https://www.microsoft.com/software- download/windows11 and I downloaded the ISO format of the latest windows 11 OS to my computer.
   b) I also downloaded Rufus software from Rufus official website: Rufus - Create bootable USB drives the easy way
2. Installing Windows 11:
   a) I created a bootable USB drive using Rufus. I used Rufus because it makes it easy to customize the bootable USB drive, Rufus allows one to tweak the installation at the ISO burning stage to essentially turn off some of the annoying system requirements for Microsoft's latest OS.



Drive Properties

Device: NO_LABEL (G:) [32 GB]

Boot selection: Windows11-2.iso

Image option: Standard Windows installation

Partition scheme: MBR

Target system: BIOS (or UEFI-CSM)

Show advanced drive properties

Format Options

Volume label: ESD-ISO

File system: NTFS

Cluster size: 4096 bytes (Default)

Hide advanced format options

☑ Quick format

☑ Create extended label and icon files

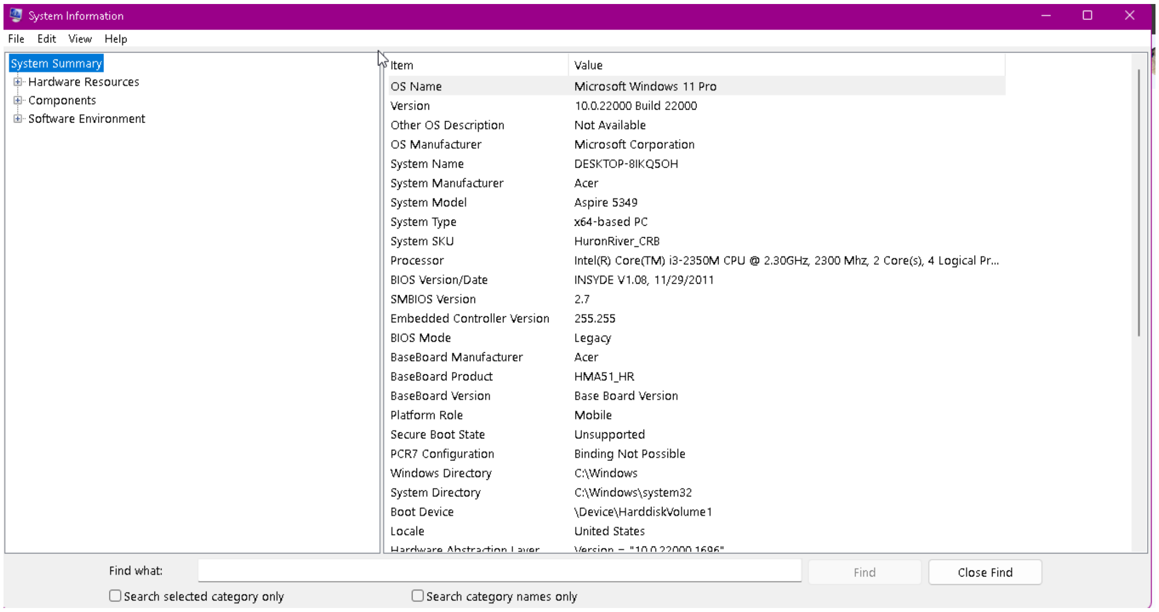
□ Check device for bad blocks — 1 pass

Status

READY

Using image: Windows11-2.iso

[image 1]



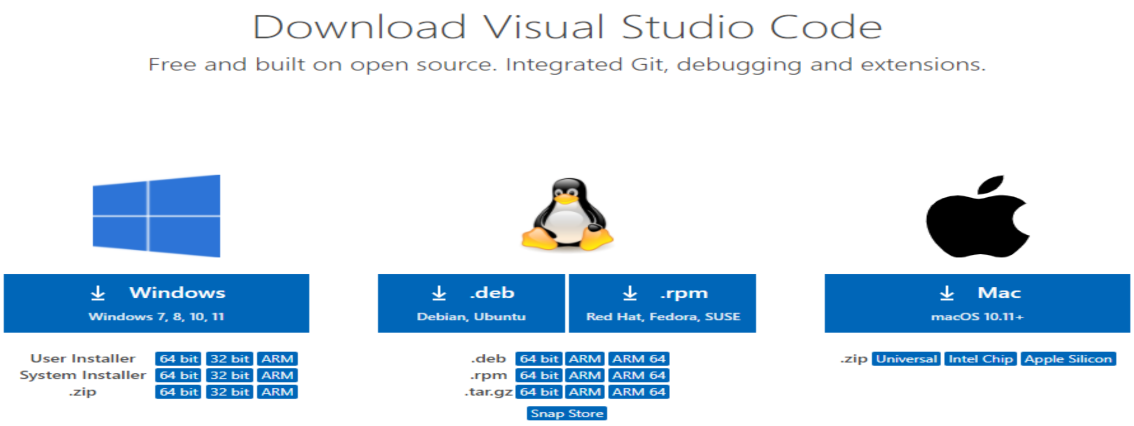
Item	Value
OS Name	Microsoft Windows 11 Pro
Version	10.0.22000 Build 22000
Other OS Description	Not Available
OS Manufacturer	Microsoft Corporation
System Name	DESKTOP-8IKQ5OH
System Manufacturer	Acer
System Model	Aspire 5349
System Type	x64-based PC
System SKU	HuronRiver_CRB
Processor	Intel(R) Core(TM) i3-2350M CPU @ 2.30GHz, 2300 Mhz, 2 Core(s), 4 Logical Pr...
BIOS Version/Date	INSYDE V1.08, 11/29/2011
SMBIOS Version	2.7
Embedded Controller Version	255.255
BIOS Mode	Legacy
BaseBoard Manufacturer	Acer
BaseBoard Product	HMA51_HR
BaseBoard Version	Base Board Version
Platform Role	Mobile
Secure Boot State	Unsupported
PCR7 Configuration	Binding Not Possible
Windows Directory	C:\Windows
System Directory	C:\Windows\system32
Boot Device	\Device\HarddiskVolume1
Locale	United States
Hardware Abstraction Layer	Version = "10.0.22000.1696"

[image 2]

Installing Visual Studio Code

1. Downloading Visual Studio Code:

a) I downloaded the latest version Visual Studio Code software for Windows 11 from the official website: https://code.visualstudio.com/Download .


Download Visual Studio Code

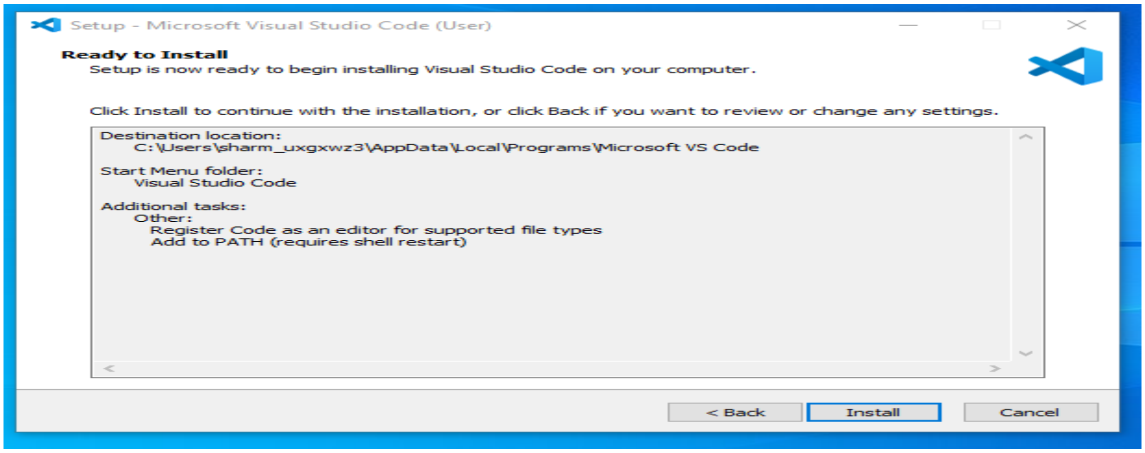
Free and built on open source. Integrated Git, debugging and extensions.

[image 3]

b) I choose Visual Studio Code because it is the most popular code editor and the IDEs provided by Microsoft for writing different programs and languages. It allows the users to develop new code bases for their applications and allow them to successfully optimize them.

2. Installing Visual Studio Code:

a) I accessed the installer from the downloads folder, and double clicked it to get it running.


Setup - Microsoft Visual Studio Code (User)

Ready to Install

Setup is now ready to begin installing Visual Studio Code on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

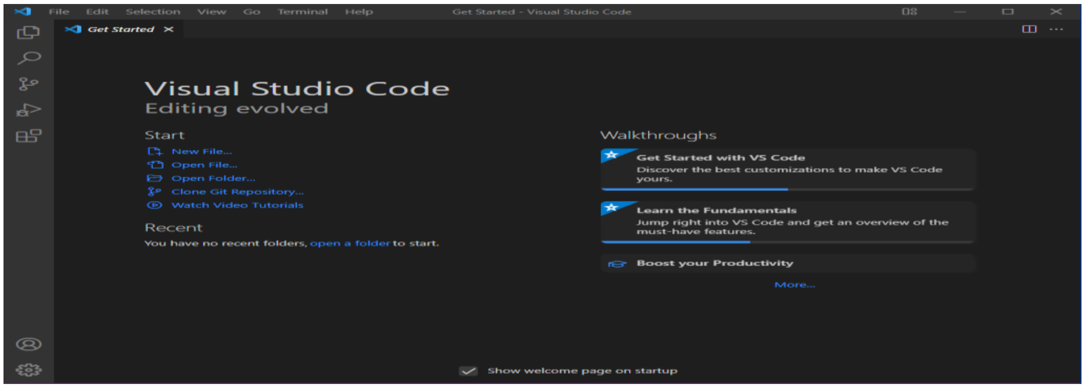
```
Destination location:
      C:\Users\sharm_uxgxwz3\AppData\Local\Programs\Microsoft VS Code

Start Menu folder:
      Visual Studio Code

Additional tasks:
      Other:
         Register Code as an editor for supported file types
         Add to PATH (requires shell restart)
```

[image 4]

b) Having followed the screen prompts, I successfully finished the installation, and launched Visual Studio Code as shown on image 5



Visual Studio Code
Editing evolved

Recent
You have no recent folders, open a folder to start.

Discover the best customizations to make VS Code yours.

Jump right into VS Code and get an overview of the must-have features.

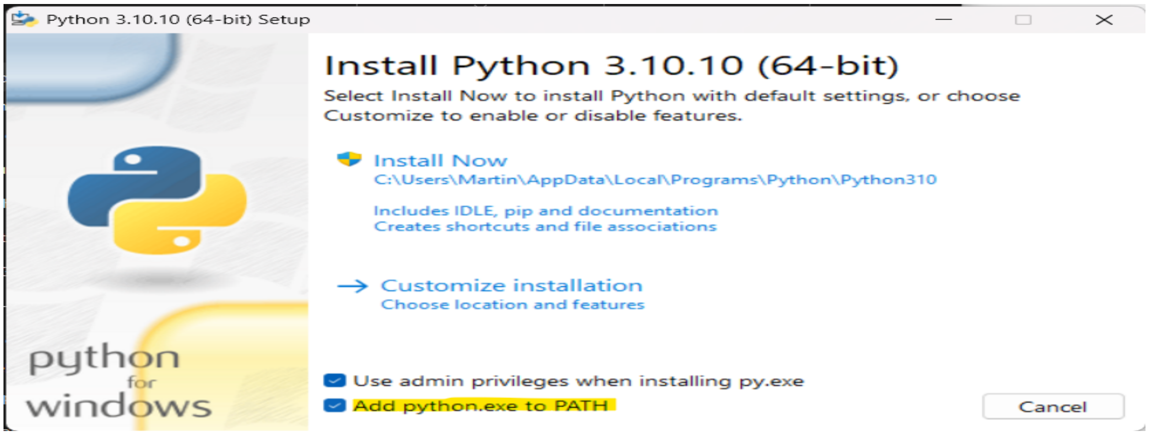
✓ Show welcome page on startup

[image 5]

Installing Python

1. Downloading Python:
    a) I downloaded the latest Python installer from the official website: Download Python | Python.org
2. Installing Python:
    a) I accessed downloads folder and double clicked python installer to open it, I followed the installation prompts, ensuring I checked "Add Python to PATH" as illustrated on image 6.

NOTE: I choose Python 3.10 because it has a wide range of modules it supports compared to the newest version.



Install Python 3.10.10 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

C:\Users\Martin\AppData\Local\Programs\Python\Python310

Includes IDLE, pip and documentation
Creates shortcuts and file associations

Choose location and features

☑ Use admin privileges when installing py.exe
☑ Add python.exe to PATH

[image 6]

Package Manager (pip)

1. Verifying pip Installation:

a) Since python 3.4 and above comes pip ready, I verified pip by running: “””pip –version””” on the windows command prompt and it was a success.

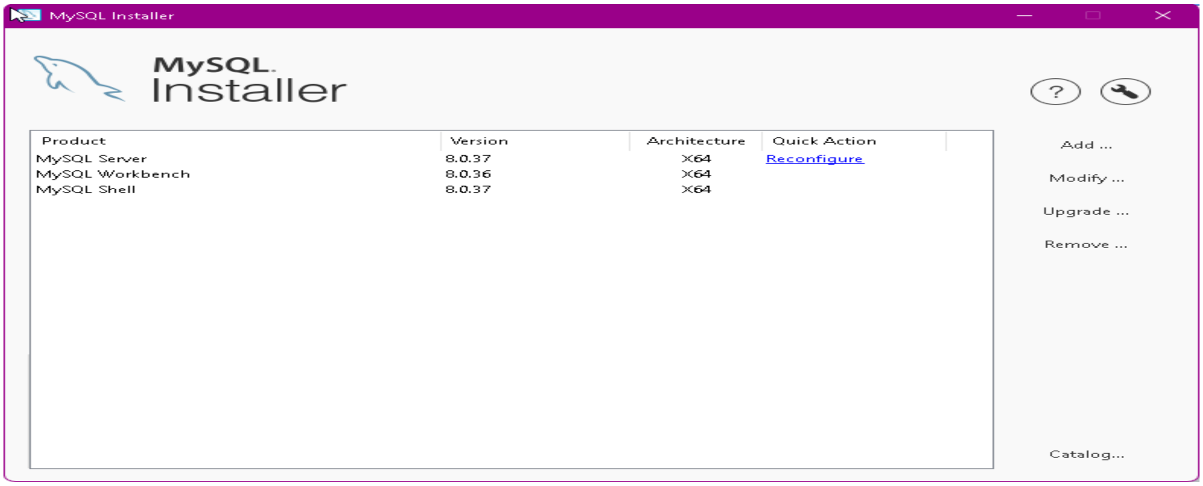
b) I also run “””python -m pip install --upgrade pip””” on the **CMD** to ensure pip was up to date.

Installing and Configuring MySQL

1. Downloading MySQL:
   a) I downloaded he latest version from MySQL download page: https://dev.mysql.com/downloads/windows/installer/

2. Installing MySQL:
   a) I run the installer and followed the configuration prompts:
      - I choose custom installation, because I wanted to set up mySql Server, mySql Workbench, and mySql Shell as shown in image 7.

   b) I did set up the root password and configured as needed.



Product	Version	Architecture	Quick Action
MySQL Server	8.0.37	X64	Reconfigure
MySQL Workbench	8.0.36	X64	
MySQL Shell	8.0.37	X64	

[image 7]

3. Verifying MySql:
   a) I opened the **MySQL Command Line Client** from **CMD**, I entered the password. And I got connected to **MySQL Server.**

Installing Git and connecting to GitHub

1. Installing Git:
   a) I downloaded Git from: [Git - Downloading Package (git-scm.com)](git-scm.com).

   b) I double clicked the installer from the downloads folder and followed all the installation prompts.

2. Configuring Git:
   a) I verified if Git installation was a success by running “””git –version””” command on Microsoft powerShell, which brught back the Git verson.

b) I then started Git Bash so that I can configure Git. I entered this commands: “””git config --global user.name ‘My Name’”””
“””git config --global user.email ‘myr.email@example.com’”””

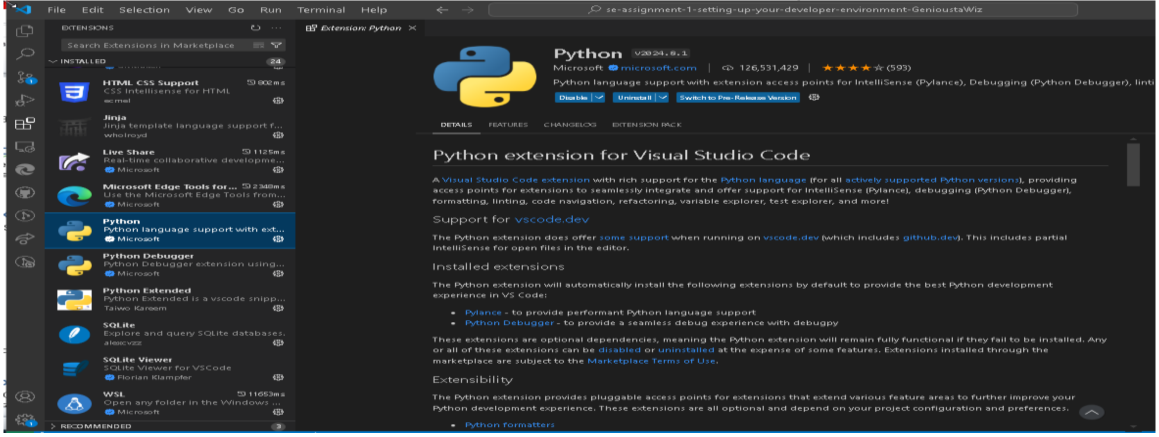
NB: I went straight to configuring Git because I already had a Github account.

Setting Up Development Enviroments.

1. For the development environment, I used Python's virtual environment (venv) module to set up each project with its own isolated dependencies.

Exploring Extensions and Plugins

1. Visual Studio Code Extensions:
   a) I opened Visual Studio Code application and clicked the Extensions icon in the Activity Bar.
   b) I searched for and installed essential extensions such as:
      - Python by Microsoft.
      - Python Debugger by Microsoft.
      - SQLite Viewer by Florian Klamper.
      - HTML CSS Support by Ecmel.
      - Gitlens by GitKraken.
      - GitHub Pull Requests by Github.
      - git-commit-plugin by Redjue.
      - And several others that I deemed of importance.



HTML CSS Support 802ms
CSS Intellisense for HTML
ecmel

Jinja
Jinja template language support f...
wholroyd

Live Share 1125ms
Real-time collaborative developme...
Microsoft

Microsoft Edge Tools for... 2340ms
Use the Microsoft Edge Tools from...
Microsoft

Python
Python language support with ext...
Microsoft

Python Debugger
Python Debugger extension using...
Microsoft

Python Extended
Python Extended is a vscode snipp...
Taiwo Kareem

SQLite
Explore and query SQLite databases.
alexcvzz

SQLite Viewer
SQLite Viewer for VSCode
Florian Klampfer

WSL 11653ms
Open any folder in the Windows ...
Microsoft

Python v2024.8.1
Microsoft microsoft.com | 126,531,429 | ★★★★☆ (593)
Python language support with extension access points for IntelliSense (Pylance), Debugging (Python Debugger), linti

Python extension for Visual Studio Code

A Visual Studio Code extension with rich support for the Python language (for all actively supported Python versions), providing access points for extensions to seamlessly integrate and offer support for IntelliSense (Pylance), debugging (Python Debugger), formatting, linting, code navigation, refactoring, variable explorer, test explorer, and more!

Support for vscode.dev

The Python extension does offer some support when running on vscode.dev (which includes github.dev). This includes partial IntelliSense for open files in the editor.

Installed extensions

The Python extension will automatically install the following extensions by default to provide the best Python development experience in VS Code:

- Pylance - to provide performant Python language support
- Python Debugger - to provide a seamless debug experience with debugpy

These extensions are optional dependencies, meaning the Python extension will remain fully functional if they fail to be installed. Any or all of these extensions can be disabled or uninstalled at the expense of some features. Extensions installed through the marketplace are subject to the Marketplace Terms of Use.

Extensibility

The Python extension provides pluggable access points for extensions that extend various feature areas to further improve your Python development experience. These extensions are all optional and depend on your project configuration and preferences.

- Python formatters

Reflection on Challenges and Solutions

1. Configuring Git and GitHub

a) **Problem:** I had difficulty in configuring user details or connecting to GitHub repositories.
b) **Solution:**

- I referred to the detailed guides on GitHub's official documentation.
- **I** Ensured Git is properly installed by running git --version in the terminal.

2. Installing MySQL

a) **Problem:** Installation errors, configuration issues, or unable to connect to MySQL server.

b) **Solution:**

- I followed the official MySQL installation guide for Windows operating system: MySQL Installation Guide.
- I ensured MySQL is installed correctly by running “””mysql -u root –p””” on windows **CMD**

3. Configuring and Using Virtual Environments (venv)

a) **Problem:** I had Issues with creating and activating virtual environments.

b) **Solution:**

- I referred to the official venv documentation for detailed instructions.
- I ensured I had Python installed and accessible in Environment variables PATH.
- I created a virtual environment using this command “””python -m venv myenv”””
- I activated the virtual environment by running this command “””myenv\Scripts\activate”””, and I was able to install modules for the test project by running “””pip install package-name”””

.

----- END -----