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

 Check System Requirements:

•  Make sure your PC meets the minimum system requirements for Windows 11. These typically include a compatible processor, 4 GB RAM, 64 GB storage, UEFI firmware with Secure Boot capability, TPM version 2.0, and DirectX 12 compatible graphics / WDDM 2.x.

Check Windows Update:

•  Windows 11 is available as a free upgrade to eligible Windows 10 PCs through Windows Update. Microsoft is gradually rolling out the update, so it might not be immediately available for all devices.

Check Compatibility:

•  Use the PC Health Check app from Microsoft to see if your current Windows 10 PC is eligible for the free upgrade to Windows 11.

Upgrade via Windows Update:

•  If your PC is eligible and the update is available, you'll receive a notification in Windows Update when Windows 11 is ready for your device.

•  Go to Settings > Update & Security > Windows Update and click on Check for updates.

•  If Windows 11 is available, you should see an option to Download and install.

 Use Media Creation Tool (Optional):

•  If you want to perform a clean installation or upgrade manually, you can use the Windows 11 Installation Assistant or the Media Creation Tool provided by Microsoft.

•  Download the Media Creation Tool from the official Microsoft website.

•  Run the tool and follow the on-screen instructions to create a bootable USB drive or directly upgrade your PC to Windows 11.

 Activate Windows 11:

•  If your current Windows 10 installation is genuine and activated, Windows 11 should automatically activate after the upgrade.

•  If you performed a clean installation, you might need to enter your Windows 10 product key or digital license key to activate Windows 11.

•  After installation, make sure to update your drivers and install any pending updates through Windows Update to ensure your PC functions smoothly with Windows 11.

Struggles

* I can only provide documentation since well i have already installed it
* Could not recover my previous data

2. 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

   Download Visual Studio Code:

•  Go to the official Visual Studio Code website at https://code.visualstudio.com/.

•  Click on the "Download for Windows" button (assuming you're using Windows; for other operating systems, select the appropriate download link).

 Run the Installer:

•  Once the download completes, locate the downloaded installer file (usually in the Downloads folder) and double-click on it to run the installer.

Accept License Agreement:

•  The installer will open. Click on "Next" to proceed with the installation.

 Choose Installation Options:

•  You can choose the destination folder where VS Code will be installed. By default, it installs in the C:\Program Files\Microsoft VS Code directory.

•  Optionally, you can select additional tasks such as creating shortcuts or adding to the PATH.

 Install:

•  Click on "Next" and then "Install" to begin the installation process. Wait for the installer to complete.

 Launch Visual Studio Code:

•  Once the installation finishes, you can choose to launch VS Code immediately by leaving the checkbox "Launch Visual Studio Code" checked.

•  Setup and Extensions:

 On the first launch, VS Code may ask you to install recommended extensions based on the files it detects in your system. You can choose to install these or skip for now.

 Install Extensions:

•VS Code supports a wide range of extensions for different programming languages and workflows. You can install extensions from the Extensions Marketplace (accessible via the Extensions view in the Activity Bar).

Start Coding:

•  Once configured and set up to your liking, start using Visual Studio Code for your programming tasks.

3. 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>

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

Download the installer for your operating system (Windows, macOS, Linux) and run it.

Configure Git:

During the installation, you can choose the default options or customize them as per your preference.

Open a terminal (Command Prompt on Windows, Terminal on macOS/Linux) to verify the installation you can do that using git bash appliction (git --version)

Configure your name and email address for Git:

bash

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

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

 Create a GitHub Account:

Sign Up:

Go to https://github.com/ and sign up for a GitHub account if you don't have one.

Follow the instructions to create your account.

Initialize a Git Repository:

Create a Local Repository:

Create a new directory (folder) for your project on your local machine.

Open a terminal or command prompt and navigate (cd) to your project directory.

Initialize Git:

Initialize a Git repository in your project directory:

git init

This command initializes an empty Git repository in your current directory.

Create a GitHub Repository:

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".

Fill in the repository name, description, and choose if it's public or private.

 Link Local Repository to GitHub:

Add Remote Repository:

On the GitHub repository page, you'll see instructions for linking your local repository to the remote GitHub repository.

Add the GitHub repository as a remote to your local repository:

bash

git remote add origin https://github.com/yourusername/yourrepository.git

Replace yourusername with your GitHub username and yourrepository with the name of your GitHub repository.

4. 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.

Download Python:

Visit the official Python website at https://www.python.org/.

Navigate to the Downloads section.

Download the latest version of Python by clicking on the "Download Python 3.11" button (where 3.11is the version number).

Choose the appropriate installer for your operating system (Windows, macOS, Linux).

Install Python:

Windows:

Double-click the downloaded installer (.exe file).

Check the box that says "Add Python 3.11 to PATH" during the installation setup (this is important for running Python commands from the command prompt).

Click "Install Now" and follow the prompts to complete the installation

Set Up Environment

Consider setting up a virtual environment for your Python projects using virtualenv or venv to manage dependencies and project-specific packages.

bash

**# Install virtualenv**

pip install virtualenv

**# Create a virtual environment**

virtualenv myenv

**# Activate the virtual environment**

**# On Windows**

myenv\Scripts\activate

5. Install Package Managers:

   If applicable, install package managers like pip (Python).

6. Configure a Database (MySQL):

   Download and install MySQL database. https://dev.mysql.com/downloads/windows/installer/5.7.html

   Download MySQL Installer:

Visit the official MySQL website: https://dev.mysql.com/downloads/mysql/.

Click on "MySQL Installer for Windows" or "MySQL Community Server" depending on your operating system.

Run the Installer:

Once the installer is downloaded, double-click to run it.

Choose Setup Type:

Select "Developer Default" or "Custom" setup type based on your preference.

Developer Default: Installs MySQL Server, MySQL Shell, MySQL Workbench, connectors, and other tools commonly used for development.

Custom: Allows you to choose specific components to install.

MySQL Server Installation:

During installation, you will be prompted to configure MySQL Server.

Choose a setup type (e.g., Server only, Server and Client, etc.) and follow the prompts to complete the installation.

Configure MySQL Server:

Set up MySQL Server with a root password (make sure to remember this password as it grants administrative access).

Optionally, configure other server settings like port number and service name.

Install MySQL Workbench :

If you want a graphical tool for managing MySQL databases, select to install MySQL Workbench during the installation process.

Follow the prompts to complete the installation of MySQL Workbench.

Start MySQL Server:

After installation, MySQL Server usually starts automatically as a Windows service.

You can verify if MySQL Server is running by checking in the Services panel (services.msc) or by opening MySQL Workbench and connecting to the server.

Verify Installation:

Open MySQL Workbench (if installed) or use the MySQL command-line client to connect to the MySQL Server using the root user and the password you set during installation

mysql -u root -p

Enter the root password when prompted. If successful, you will see the MySQL command-line prompt (mysql>).

7. 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.

8. 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.