### **Setting up Development Environment**

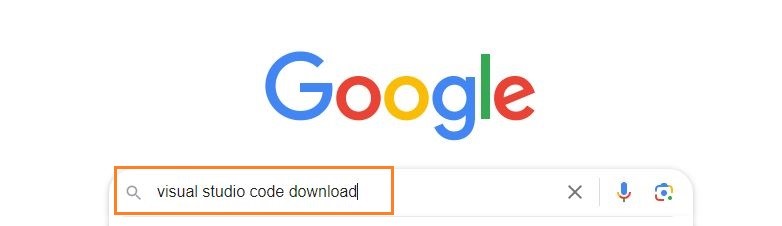
Document detailing the setup process with step-by-step instructions including a reflection on the challenges faced during setup and strategies employed to overcome them.

### **Operating System: Windows 11**

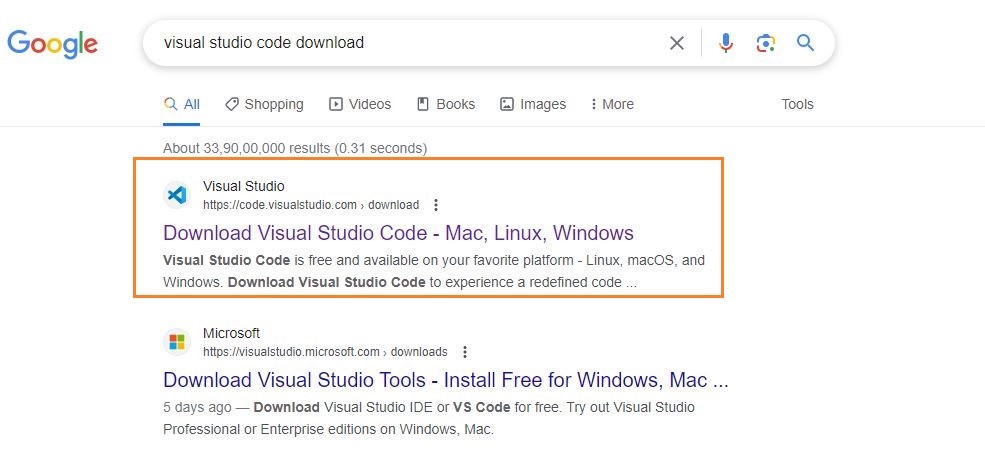
## This was pre-installed on the PC so I did not need to download or install.

## **Installation Steps of VS Code (Text Editor/IDE)**

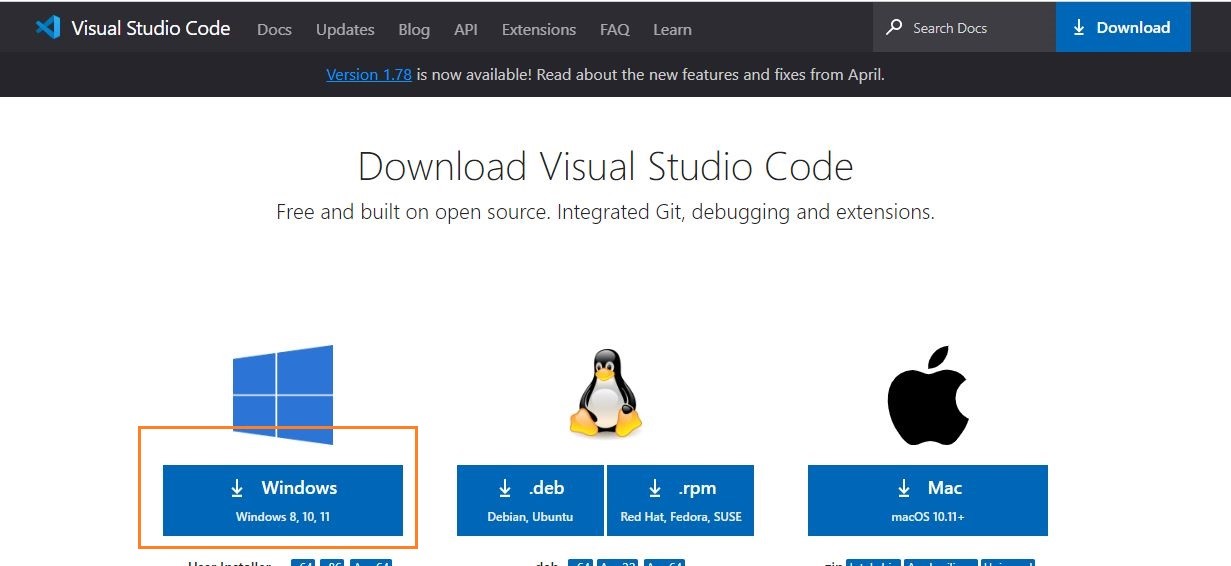
1. Open Google and type **Visual Studio Code download** in the search bar.



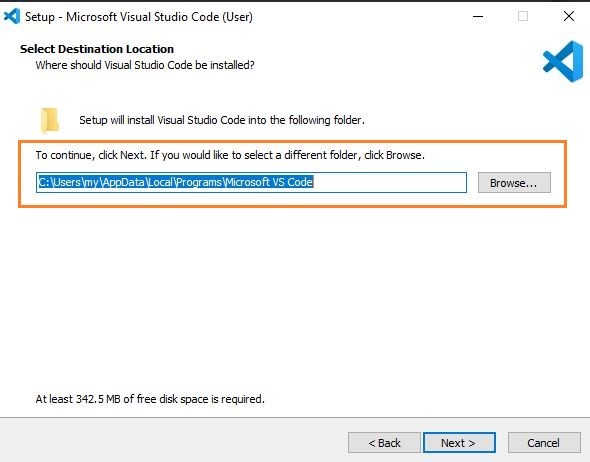
1. Click on the link below.



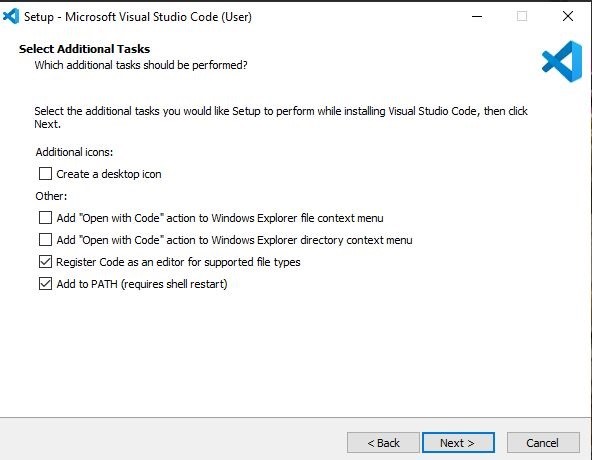
1. In this case we are selecting Windows.



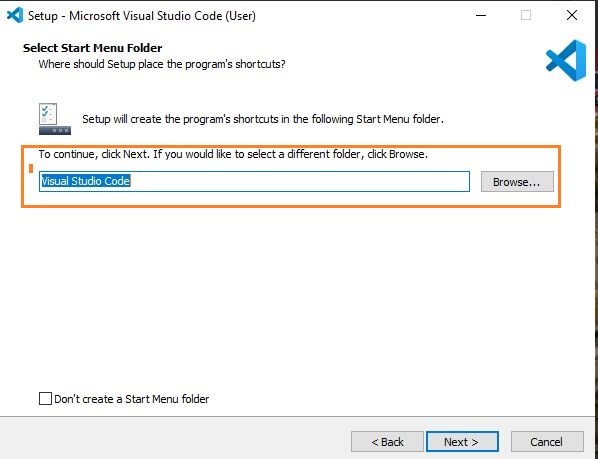
1. The file will be downloaded onto your system. Open the file and then click on **Install**. After downloading the VS Code file, the official site will display a Thanks message for downloading the file.
2. Now accept the license agreement
3. Then it prompts for the file location, where you want to save the VS Code file. Browse the location and then click on **Next**.



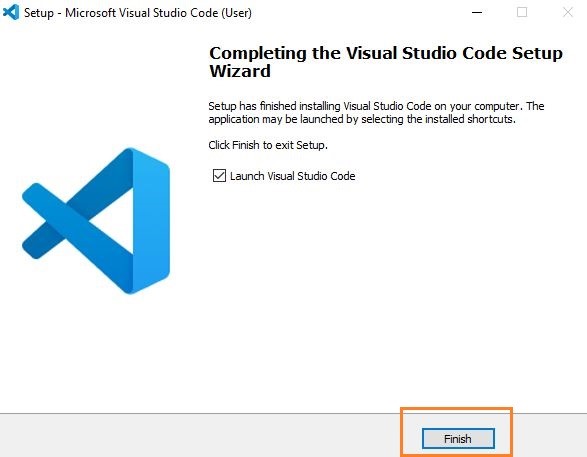
1. Next, you see the prompt for the additional task which we want the VS Code to perform. At this step, choose the default settings and then click on **next**.



1. The next prompt is how you want the VS Code on your startup. Change according to your convenience and click on **Next**.



1. The installation of VS Code will now begin.
2. At this step, we have completed installing VS Code, click on **Finish**.



**Explore Extensions:**

* Open Visual Studio Code.
* Click on view then extensions or press Ctrl+Shift+X.
* Search for and install extensions based on your preferences and project requirements eg. Python, Flutter, Django

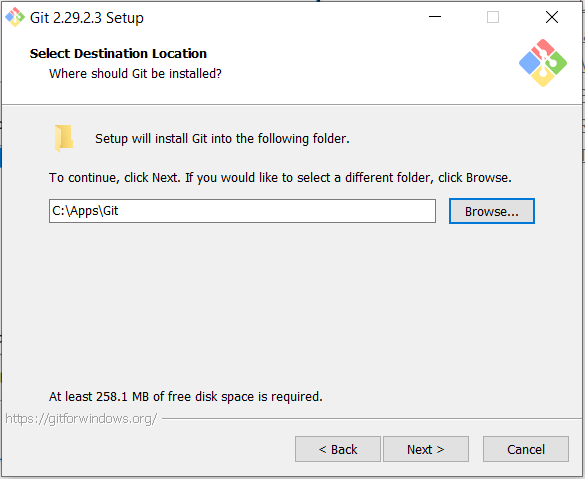
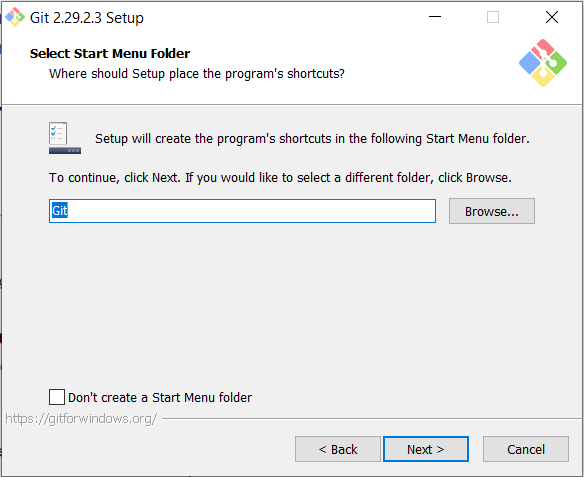
#### **Reflection:**

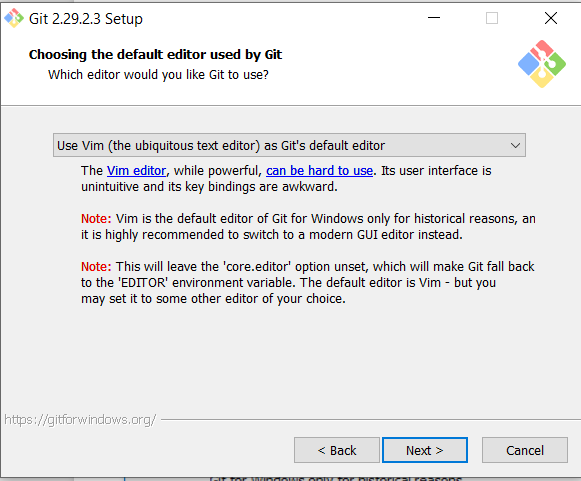
Setting up Visual Studio Code was straightforward and intuitive. The download and installation process was smooth, and launching the application was quick. Exploring and installing extensions was easy, and I found a variety of useful extensions to enhance my development experience.

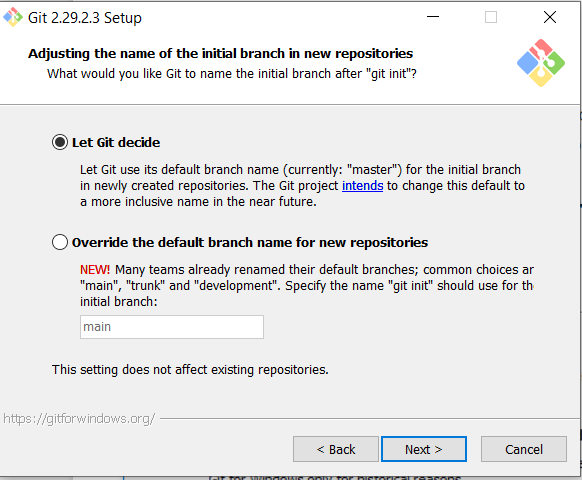
However, one challenge I faced was customizing settings to align with my specific preferences. While the Settings UI in VS Code is user-friendly, understanding and configuring some advanced settings required additional research. To overcome this challenge, I utilized online resources, documentation, and community forums to find solutions and optimize my VS Code setup according to my needs.

Overall, Visual Studio Code provides a robust and versatile development environment with a wide range of features and extensions, making it suitable for various programming languages and workflows.

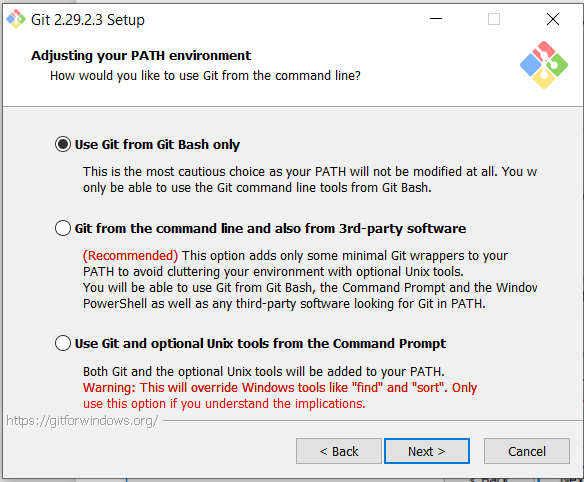
## **Git download and installation (Version control system)**

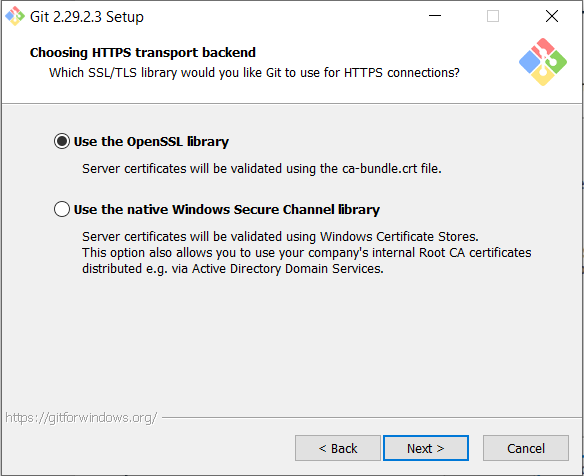
Go to the [Git For Windows](https://gitforwindows.org/) website and click on the [Download] button to download and save the installer. Once downloaded double-click on the .exe file to begin installation.  
 **Select Installation Folder**  
  
 **Select Start Menu Folder** Accept the suggested name and click [Next>]  
  
 **Choose Default Editor**

This will determine which text editor will open to input commit notes and other text messages interactively.  
  
 **Select Default Branch Name**

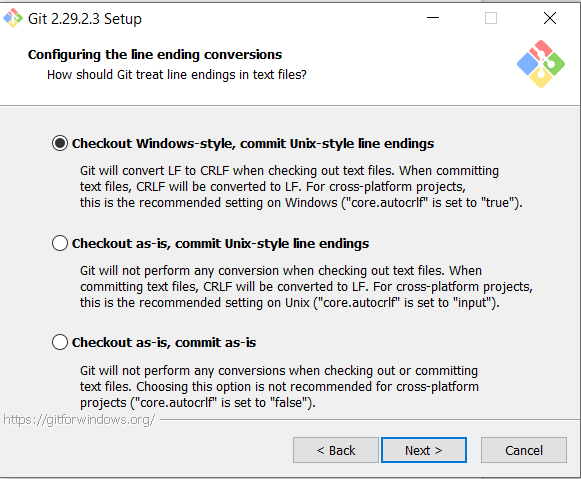
Let Git decide and click [Next>]  


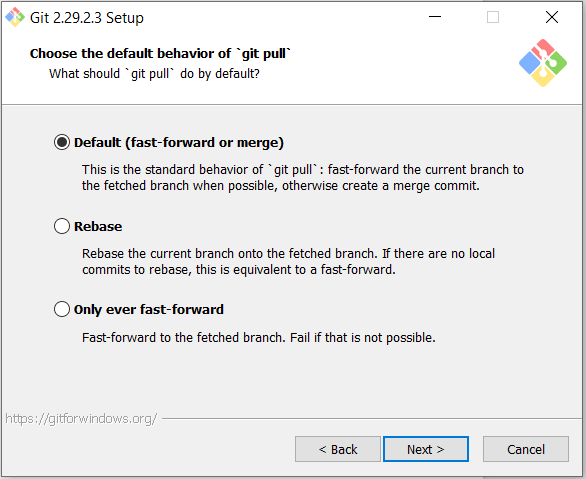
**Adjust path**

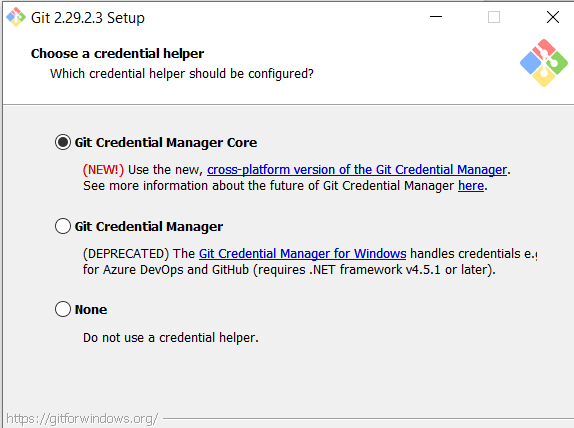
Make sure you select “Use Git from Git Bash only”  
  
 **Select HTTPS transport**

Specify the library to use for secure HTTP connections. Go for OpenSSL unless you have a reason to do otherwise  


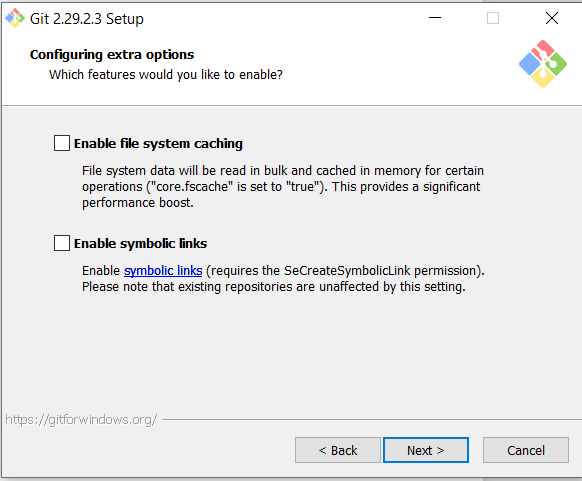
**Configure Line Ending Conversions**

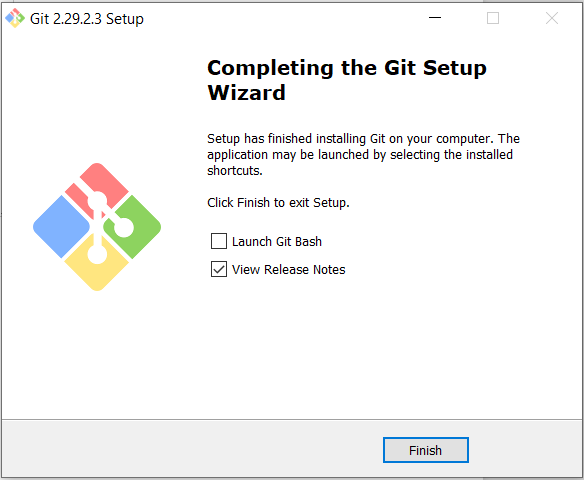
Make sure you select “Checkout Windows-style, commit Unix-style line endings”  
  
 **Select Behaviour of Pull Command**

Select Default  
  
**Select Credential Helper**

Go with the recommended choice  


**Configure Extra Options**

Leave unchecked and click [Next >]  
  
   
**Installation Complete**

Click on [Finish].   


Open Git Bash and run the following commands to configure Git:  
git config --global user.name "Your Name"

git config --global user.email "[your\_email@example.com](mailto:your_email@example.com)"

To verify that Git is installed correctly, run the following command:  
git --version

You should see the installed Git version displayed in the Command Prompt or Git Bash.

#### **Reflection:**

Setting up Git Bash provided a convenient command-line interface for Git on Windows. The installation process was straightforward, and launching Git Bash was easy from the Start menu.

Configuring Git within Git Bash was similar to the Command Prompt, and setting my username and email was done using familiar Git commands.

One challenge I encountered was navigating the Git Bash interface for the first time, as it differs slightly from the Command Prompt. However, with some exploration and practice, I quickly became accustomed to using Git Bash for version control tasks.

Overall, Git Bash provides a lightweight and efficient way to interact with Git on Windows, and with Git configured and ready to use, I'm prepared to manage version control for my projects effectively.

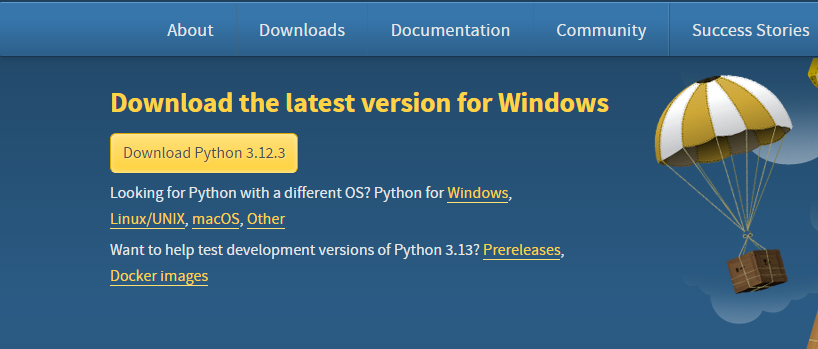
**Download and installation of Python for Windows 11**

Visit the Downloads Section of the Python website

Open a browser and type Python Download or paste link (https://www.python.org/downloads/)

Choose the Python Version

Click on the latest version to download the installer.



Once the download is complete, run the installer program. On Windows, it is a .exe file.

Run the Python Installer for how to install Python on the Windows downloads folder

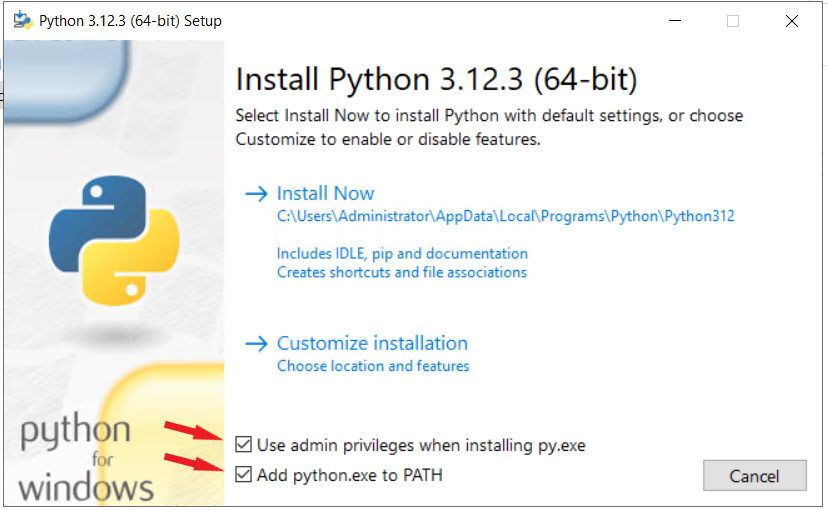
Make sure to mark **Add Python to PATH** otherwise you will have to do it explicitly. It will start installing Python on Windows. Download Python Windows executable installer

The following instructions walk you through the installation process:

1. Launch the Python Installer that you downloaded.
2. There are two checkboxes in the installation window:

Administrator Rights. Whether Python is installed for all system users or only the current user is determined by this argument. You can modify the Python installation folder with this option.

Include Python in PATH. After installation, the executable is added to the PATH variable by using the second option.



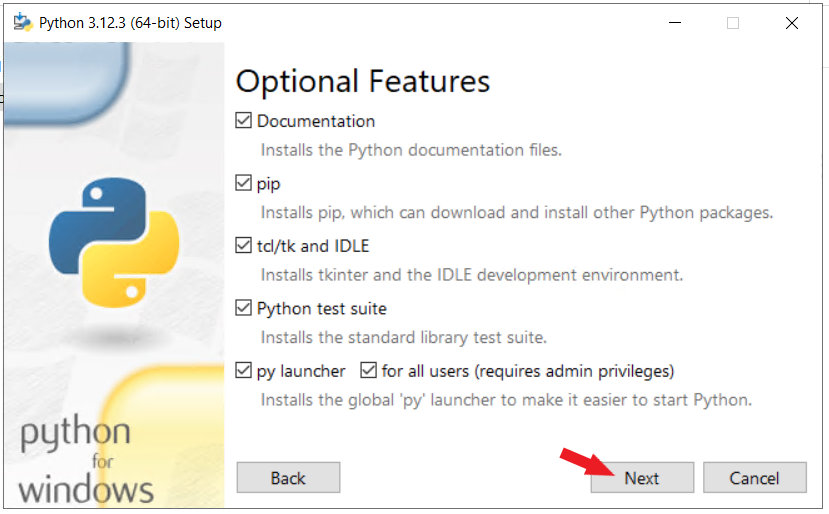
Check both items.

To proceed with the suggested installation, select Install Now (if you choose to skip the next two steps).

3. Python is installed by default for the current user at C:\Users\[user]\AppData\Local\Programs\Python\Python[version]. It comes with extra documentation, the PIP package manager, and the standard Python editor, IDLE. Additionally, the installer sets up the required file associations and shortcuts when possible.

Modifying the installation enables you to alter these parameters and options.

4. Select the installation features that are optional. Although Python functions without these features, having them adds to the program's usefulness.

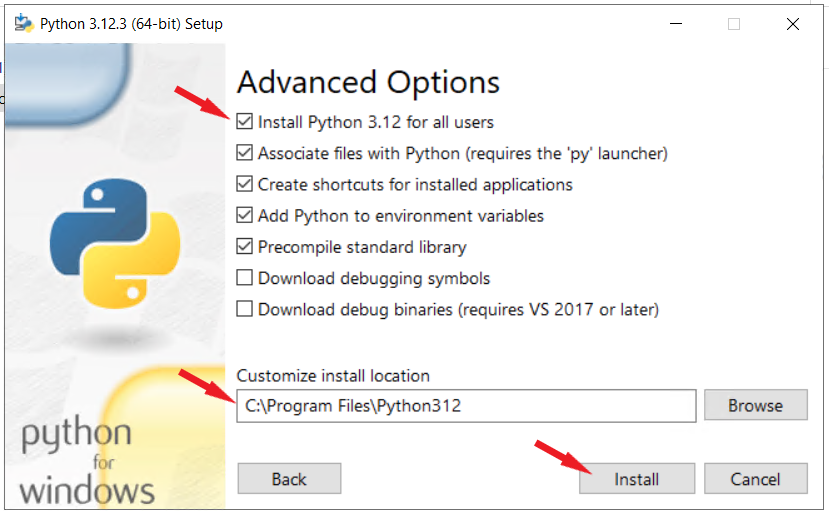


To go to the screen with the Advanced Options, click Next.

5. Advanced options are included in the installation customization section.

Make the decision to install Python for every user or not. The install location is altered by this option to C:\Program Files\Python[version]. Since it doesn't have spaces in the path and is accessible by all users, C:\Python[version] is a popular option when choosing the location manually. Both routes could result in problems installing packages because of administrative rights.

Additional advanced options are to add Python to PATH, create file associations, and create shortcuts.



Click Install to begin the installation after making the necessary selections.

After installation is complete click on Close.

**Reflection:** Python was selected for its versatility and extensive library support. Installing Python from Python.org included ensuring it was added to the system PATH for easy command-line access. Managing Python packages using pip simplified dependency management for Python projects.

**Challenges:**

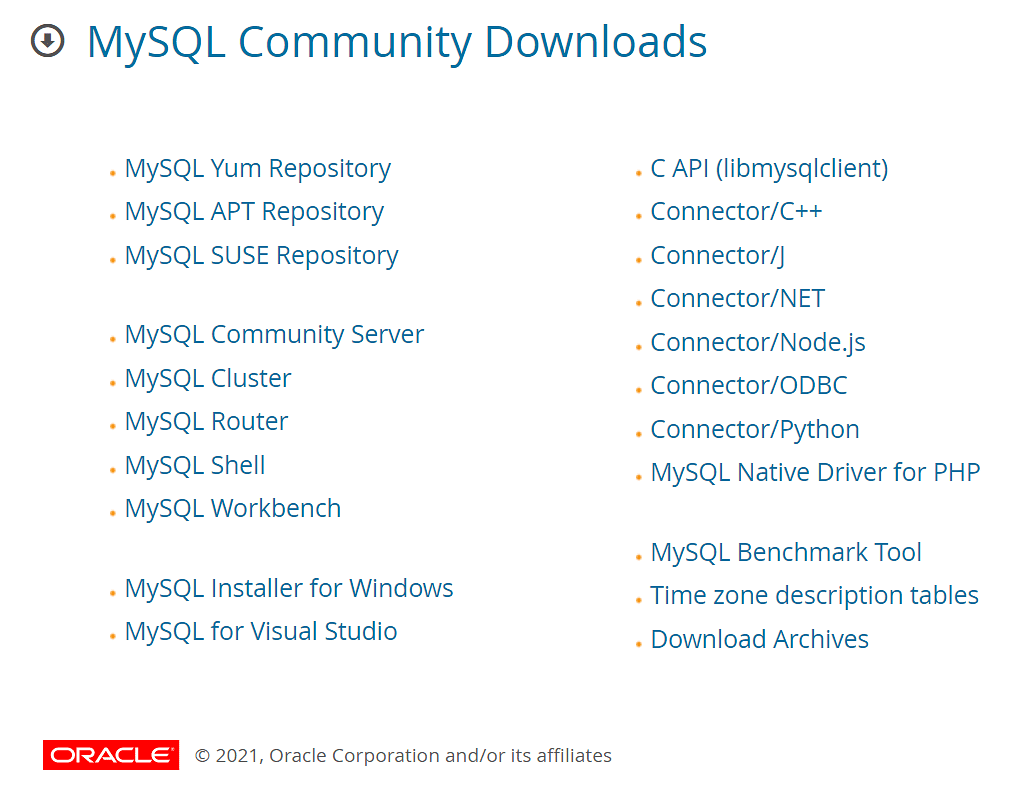
* **Dependency Management:** Managing dependencies and ensuring compatibility between different Python packages and versions required careful attention. Setting up virtual environments helped isolate project dependencies and avoid conflicts.

**MySQL : The installation process is similar to other operating systems.**

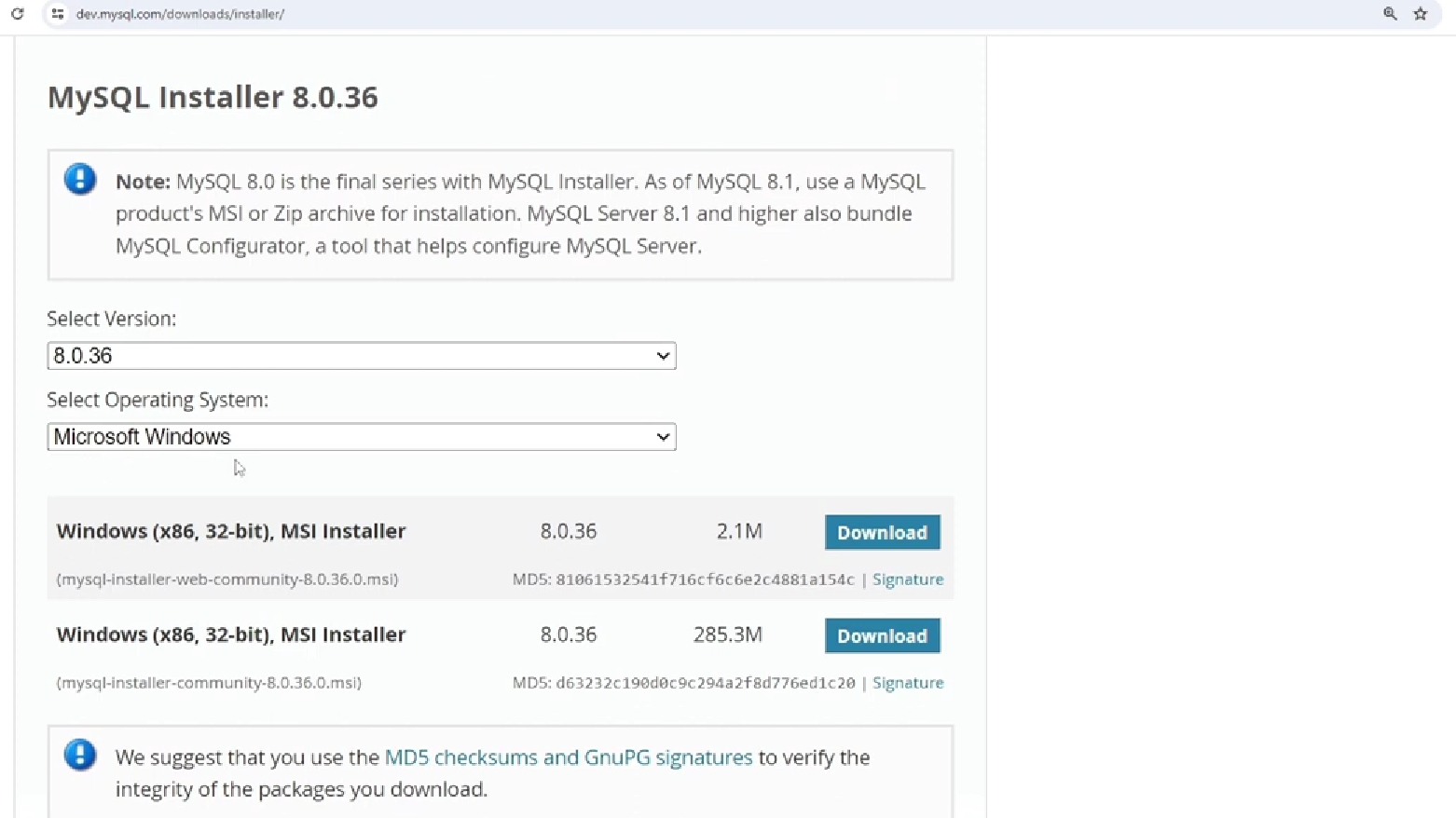
1. Open the MySQL website on a browser.

2. Select the Downloads option.

3. Select MySQL Installer for Windows.

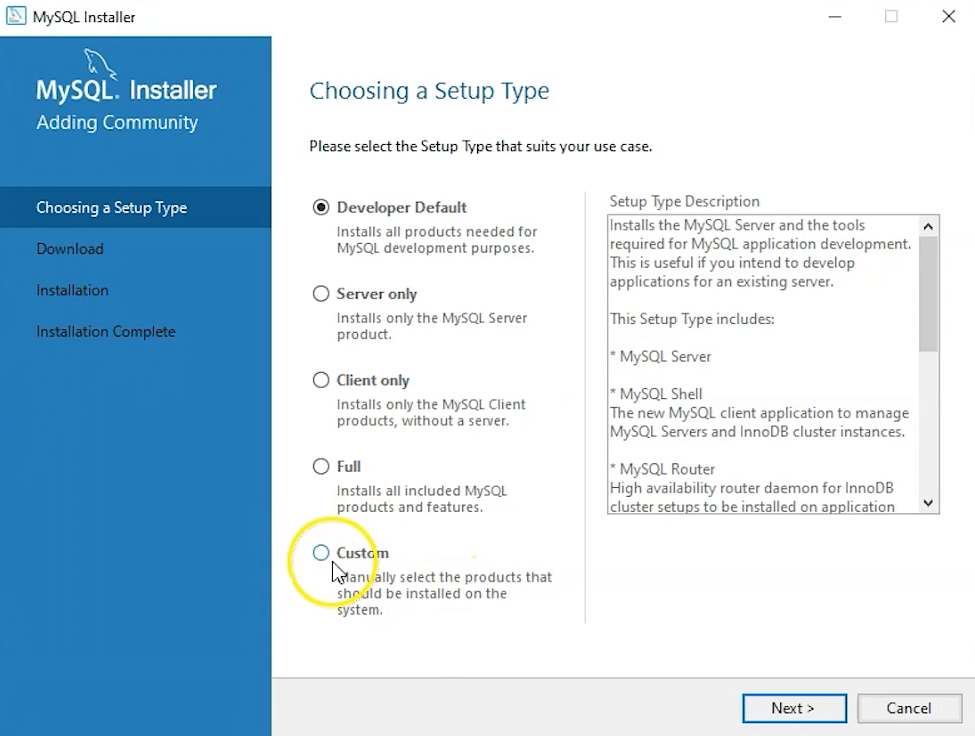


4. Choose Windows installer and click on download.



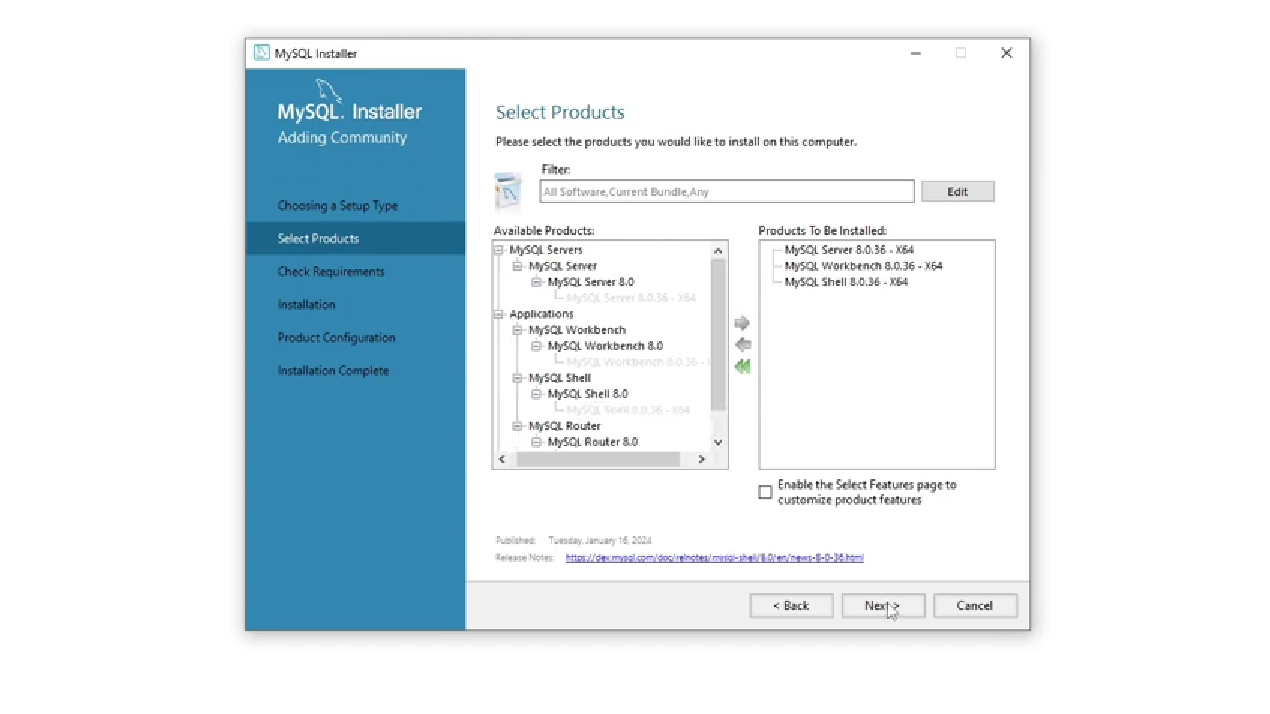
5. After the download, open the installer.

6. It will ask for permission; when it does, click Yes. The installer will then open. Now, it will ask to choose the setup type. Here, select Custom.

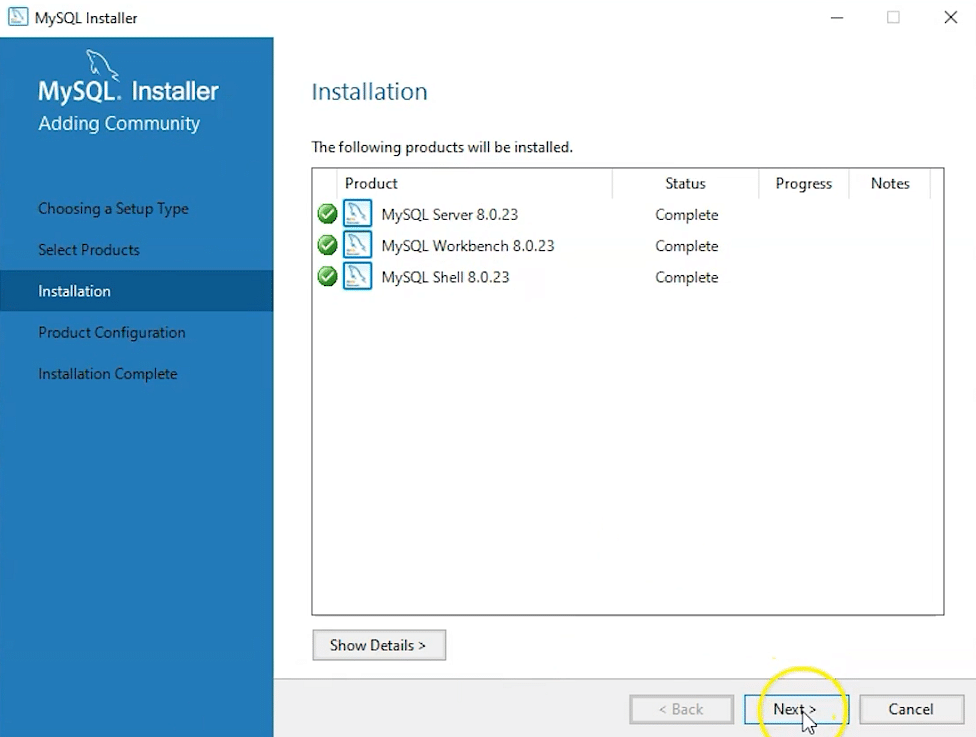


7. Click on Next. With this, you will install MySQL server, MySQL Workbench, and MySQL shell.

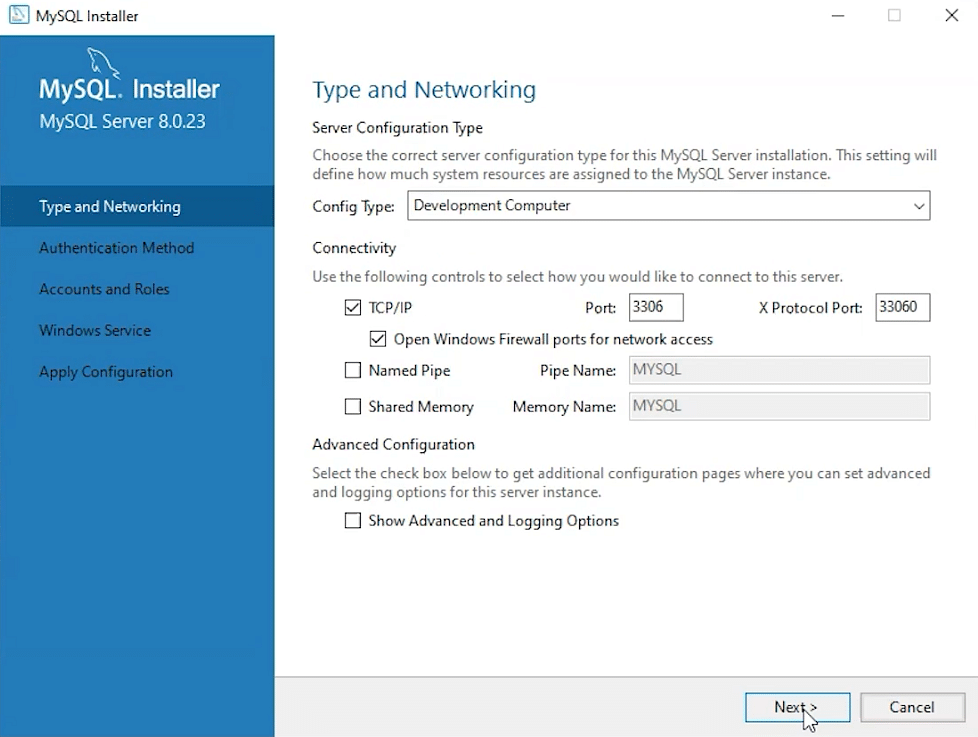
8. Open MySQL Servers, select the server you want to install, and move it to the Products/Features to be installed window section. Now, expand Applications, choose MySQL Workbench and MySQL shell. Move both of them to ‘Products/Features to be installed’.



9. Click on the Next button. Now, click on the Execute button to download and install the MySQL server, MySQL Workbench, and the MySQL shell.

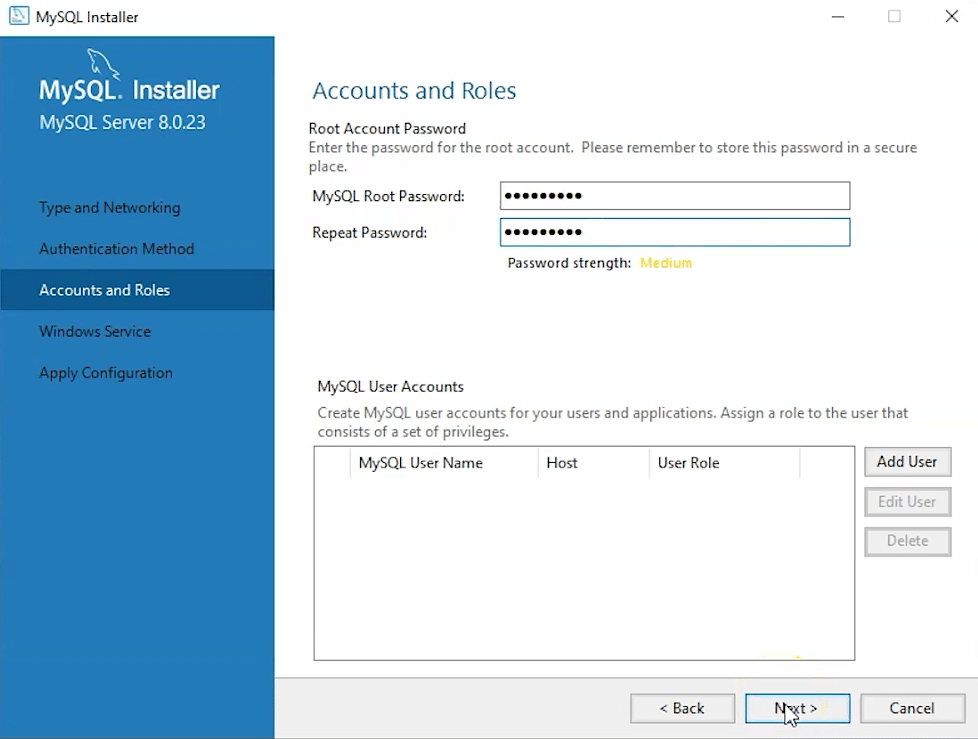


10. Once the product is ready to configure, click on Next. Under Type and Networking, go with the default settings and select Next.

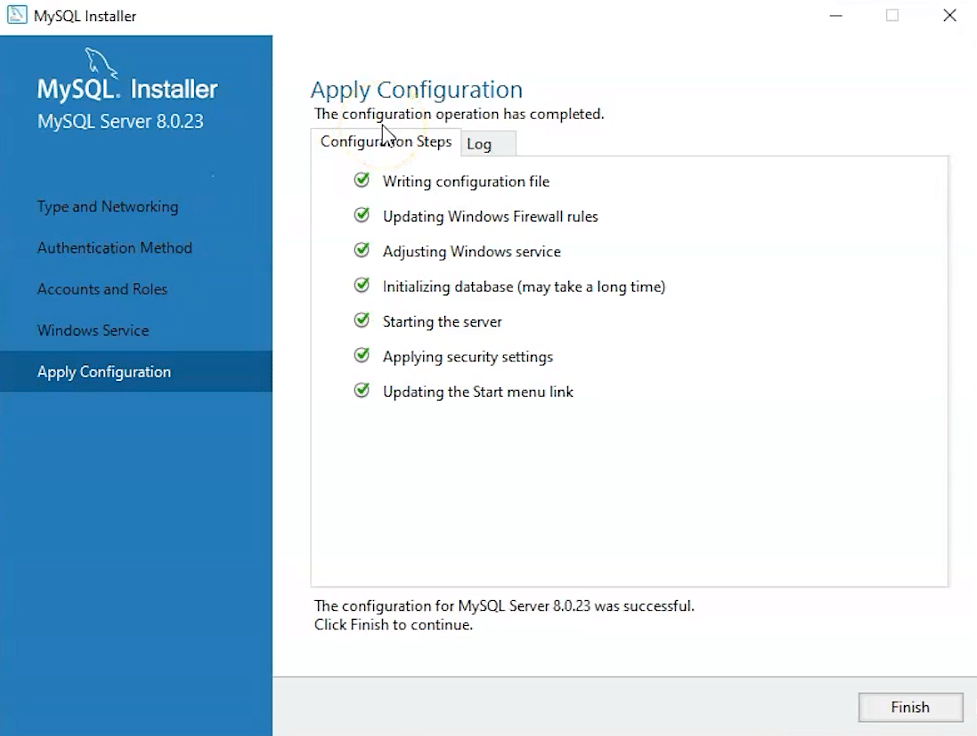


11. For authentication, use the recommended strong password encryption.

12. Set your MySQL Root password and click on next.



13. Go for the default windows service settings and under apply configuration, click on execute. Once the configuration is complete, click on finish.



14. Complete the installation. This will now launch the MySQL Workbench and the MySQL Shell.

Once MySQL is installed, select the Local instance and enter the password.

**Challenges:**

* During the MySQL installation, understanding and selecting the appropriate components (MySQL Server, Workbench) based on project requirements and system capabilities required careful consideration.
* Configuring MySQL settings, Understanding the implications of different configuration options, especially related to security and performance, required consulting MySQL documentation and online resources.

### **Reflection:**

By following installation guides, consulting documentation, I successfully configured MySQL. Documenting the setup process and reflecting on challenges and strategies employed provided insights into maintaining robust and scalable database solutions for future projects.

Overall: Setting up the developer environment involved overcoming various challenges understanding new technologies. Each component contributed to creating a robust workspace conducive to coding, debugging, version control, and collaboration. The process included exploring documentation, seeking help from online resources, and leveraging community forums to optimize setup and resolve issues effectively.

By documenting the setup process and reflecting on encountered challenges and strategies, I gained valuable insights and skills necessary to maintain and enhance a productive developer environment for future projects.