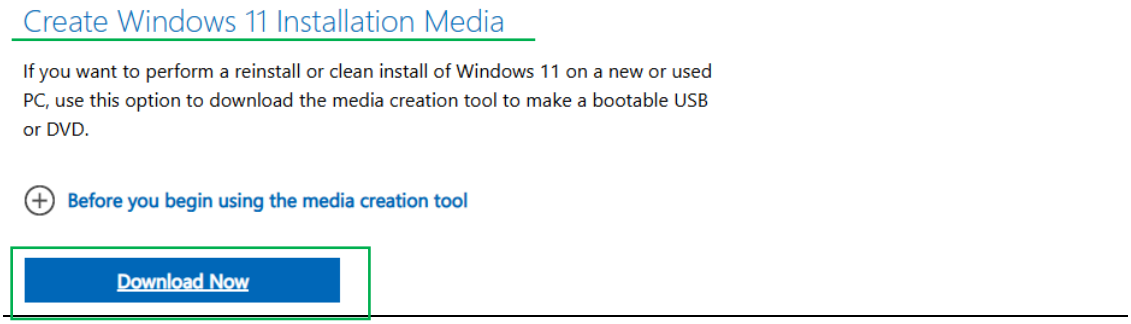
DOCUMENTATION COMPILED BY:

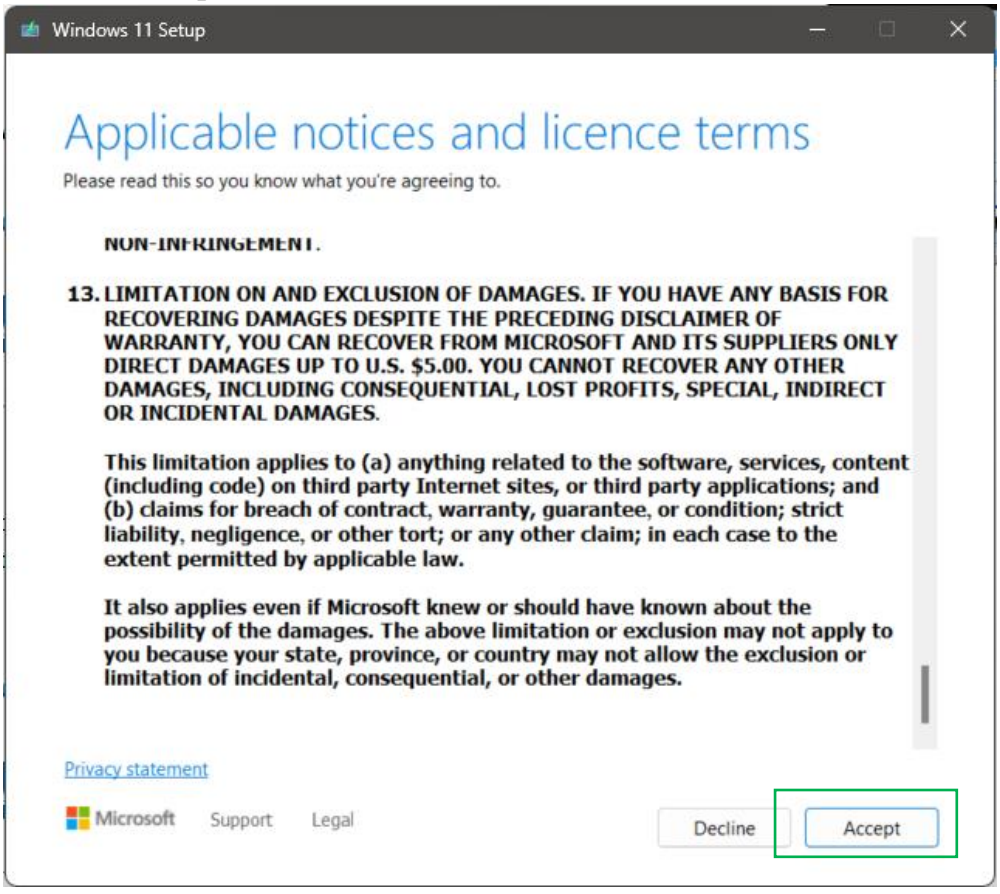
VICTOR MUNGAI

Victormungai01@gmail.com

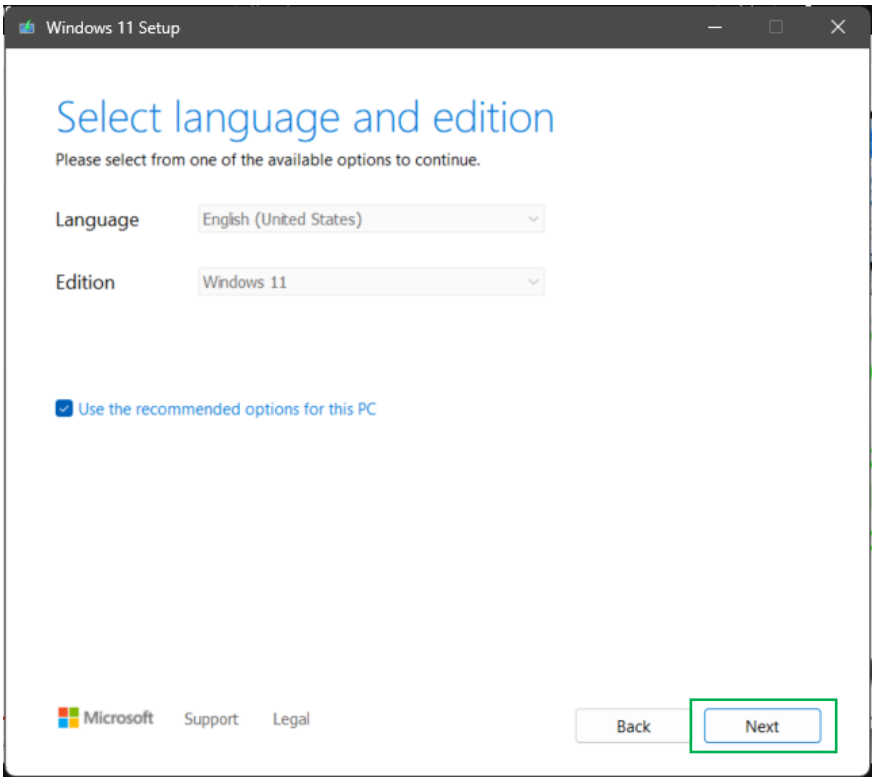
1. **DOWNLOADING AND INSTALLING WINDOWS 11. (shown on visuals by a green box)**
   - ✓ click on this link to direct to Microsoft website https://www.microsoft.com/software-download/windows11
   - ✓ on create windows 11 installation media, click on download now



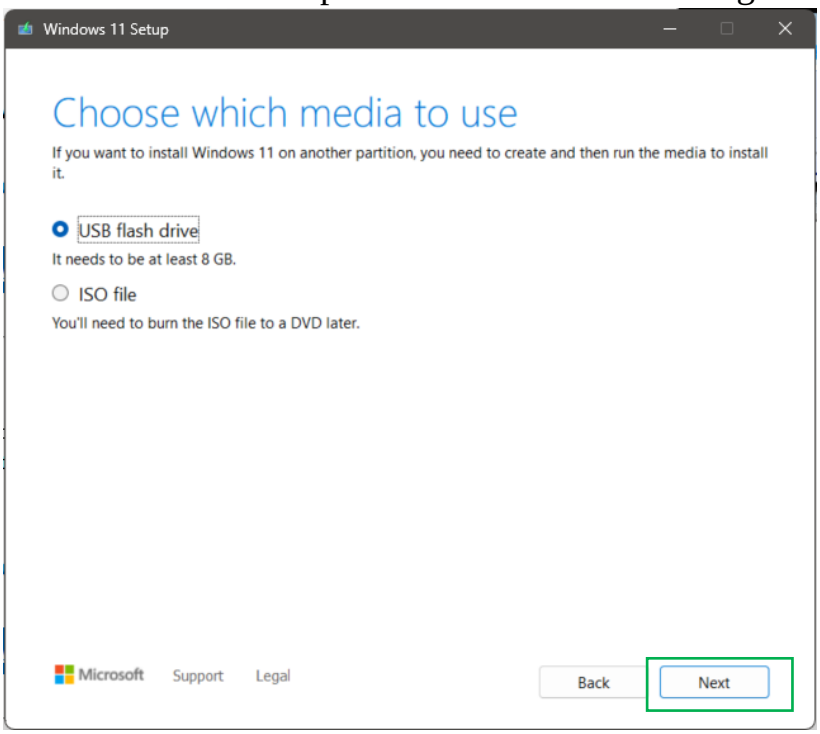
   - ✓ once the download is finished, locate it and run the app, a licence agreement appears. Click on accept.



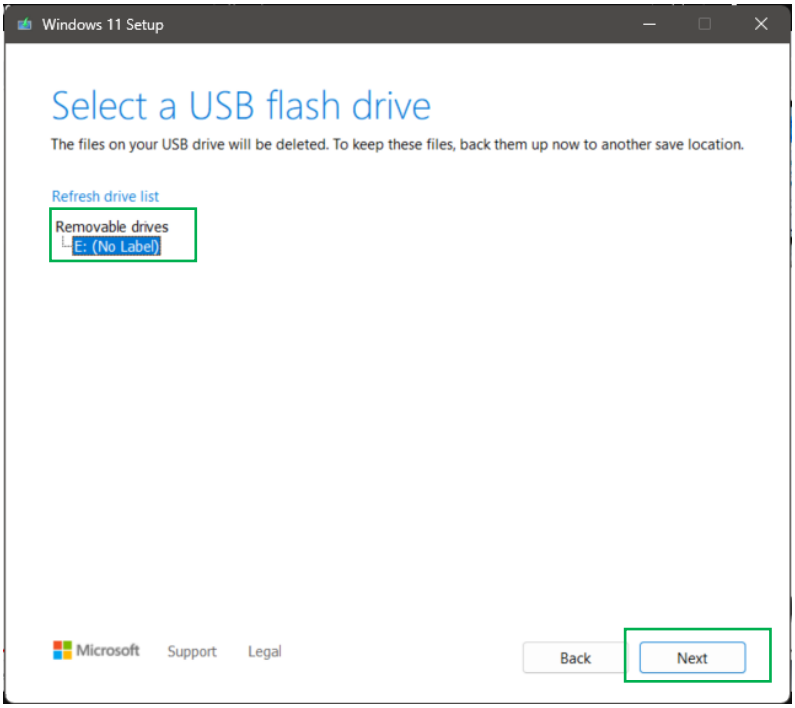
   - ✓ Select language and edition, click on next



✓ Choose which media to use, select USB flash drive and ensure that its completely empty and has a minimum space of 8GB before selecting it. Click on next



✓ Select the USB flash drive you want in case you have more than one inserted. Click on next



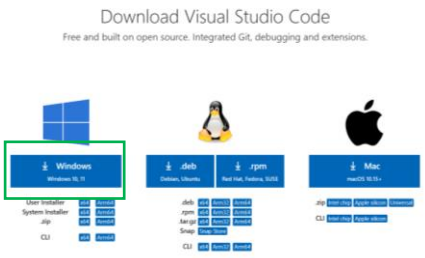
- ✓ The download will start, verify and create windows 11 media once complete. Once it is done click on finish

- ✓ Eject the USB from the PC and connect it to the new PC
- ✓ Turn on the PC and press the key that opens the boot-device selection menu(can be F1 but varies depending on PC model)
- ✓ Select boot from USB flash drive. Windows Setup starts. Click Install now. Once complete, remove USB flash drive.
- ✓ Customize settings such as keyboard, date and time. Enjoy.

2. SETTING UP MY DEVELOPMENT ENVIRONMENT

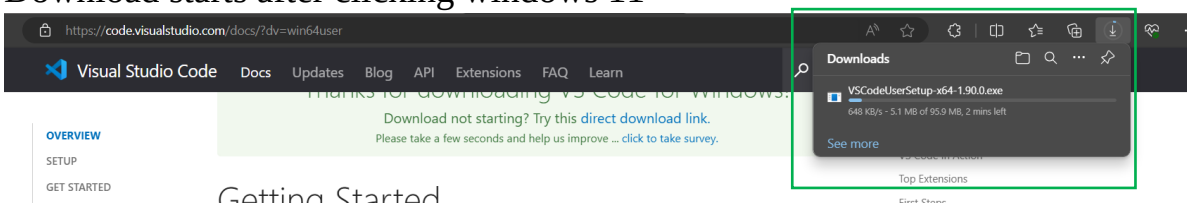
I. Downloading & Installing VS Code

- ✓ Click on this link to open VS Code website https://code.visualstudio.com/Download

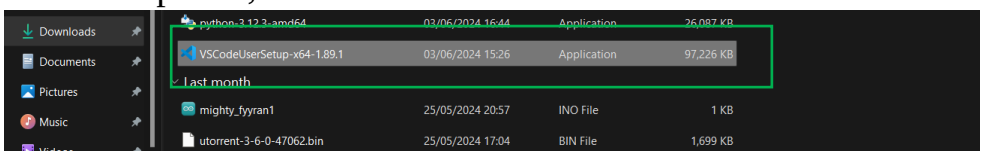


Click on windows 11

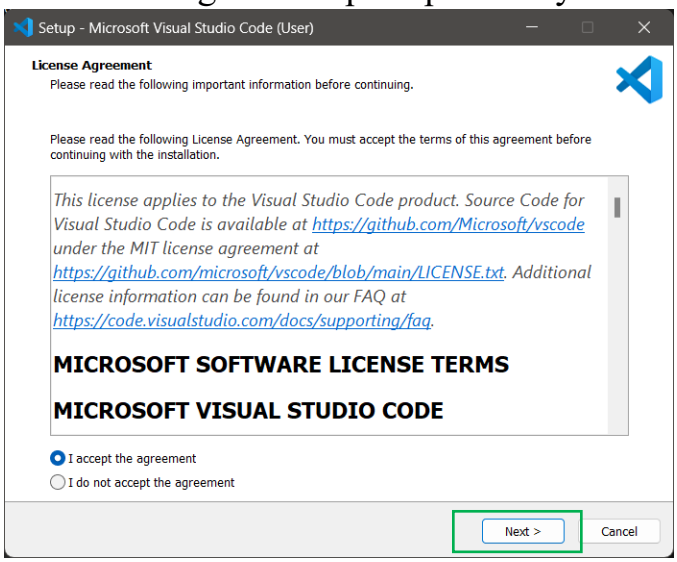
✓ Download starts after clicking windows 11

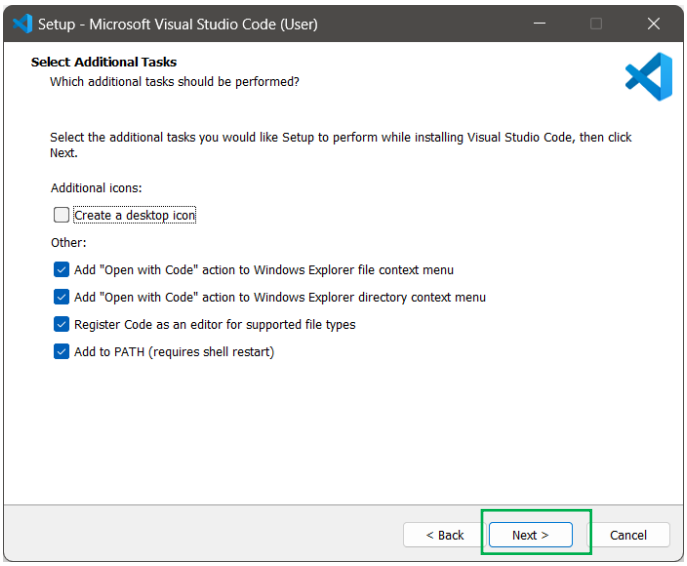


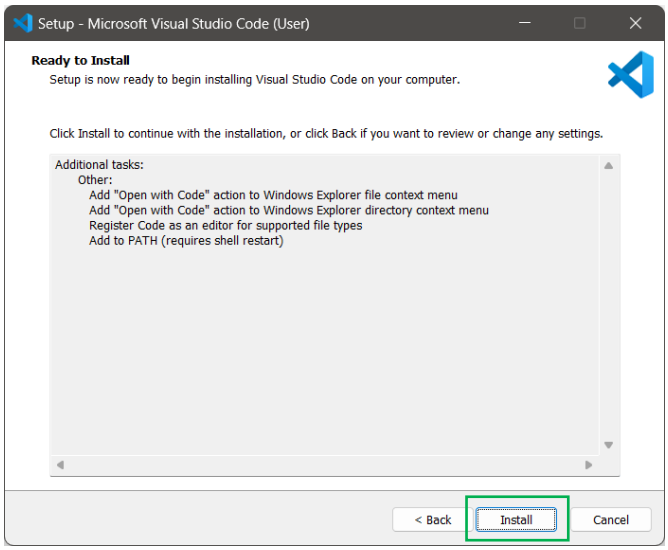
✓ Once completed, locate it and run it



✓ Follow through all the prompts until you've clicked on install





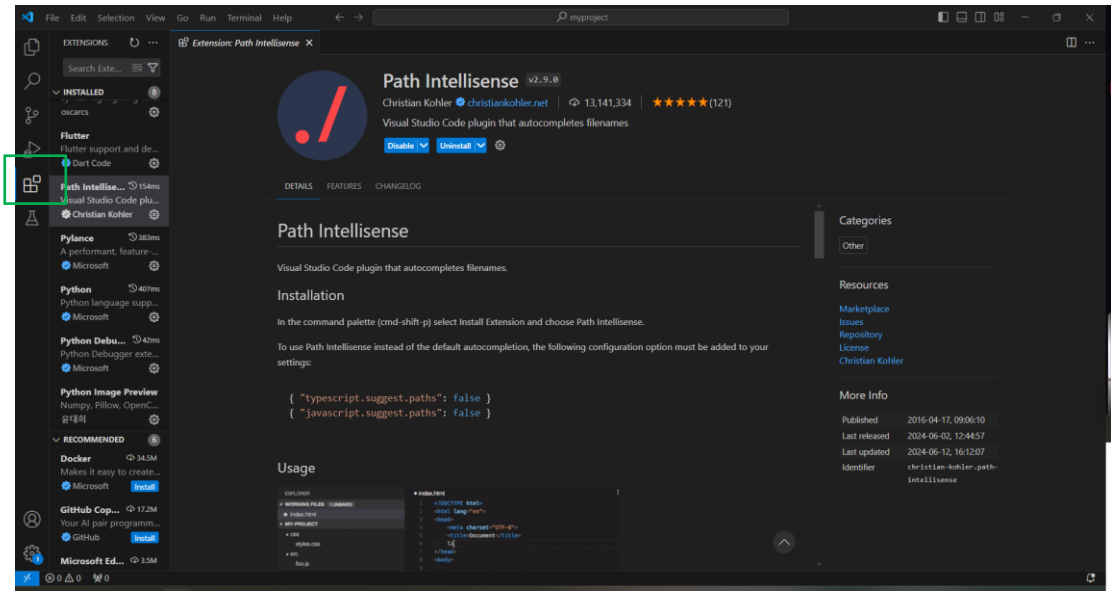


II. Exploring Extensions in VS Code

Launch VS Code. Extensions enhance functionality and coding experience.

They can be found on the activity bar or by ctrl+shift+x. Look for the desired extension and click on install or uninstall if no longer needed.

Essential extensions include path intellisense, css peek, HTML snippets, Git lens, python, dart, flutter

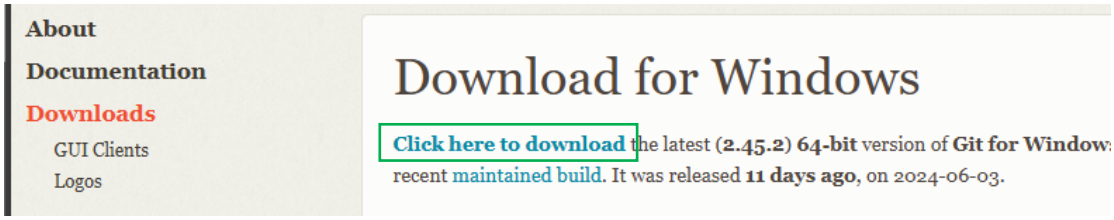


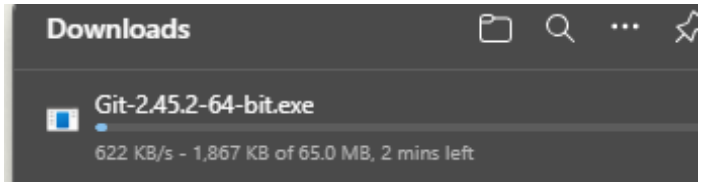
III. Downloading & Installing Git

✓ Click on this link to direct to Git website https://git-scm.com/downloads

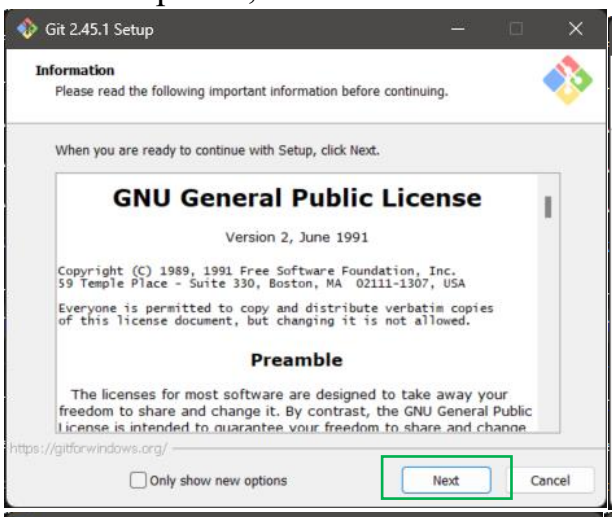
✓ Click on windows

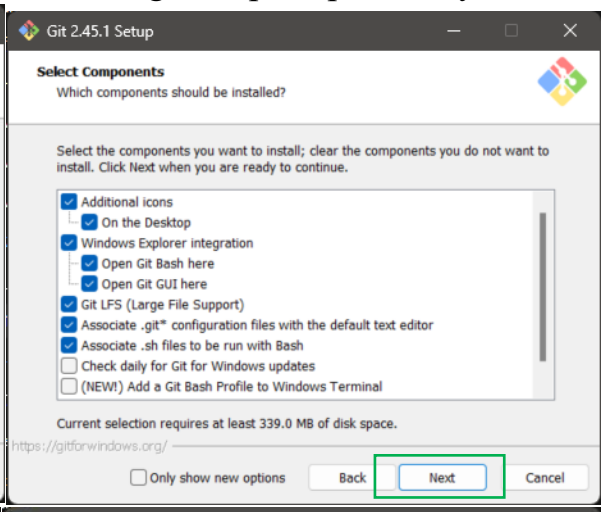
✓ Click on 'Click here to download'. The download will start

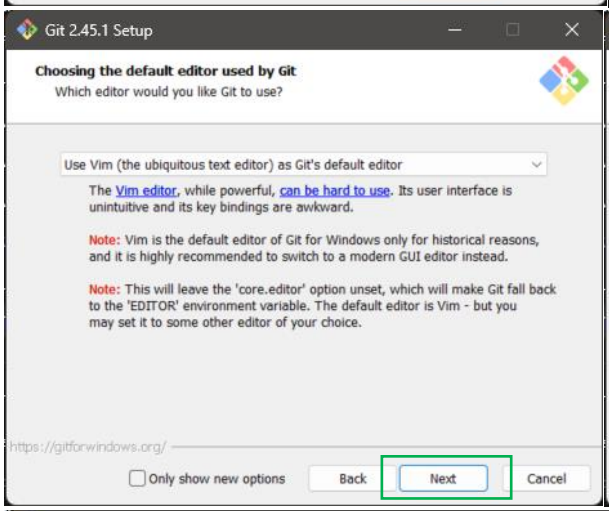


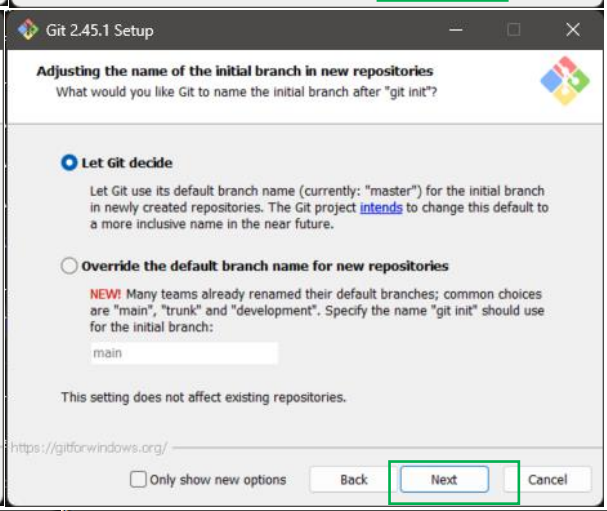


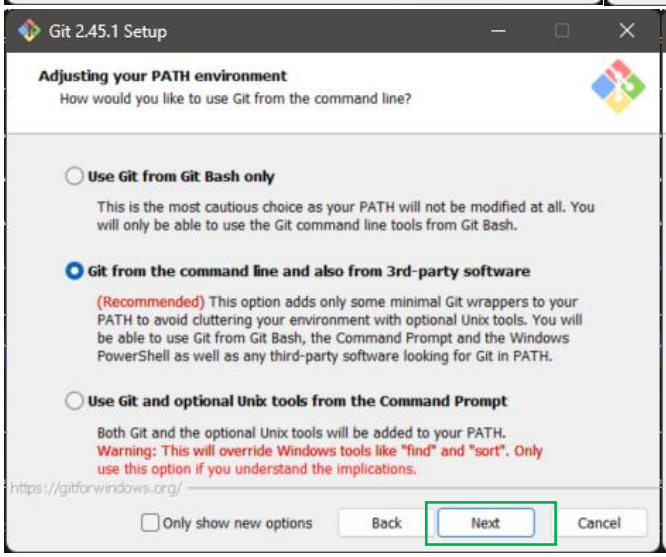
✓ Once completed, locate it and run it. Follow through the prompts until you've installed

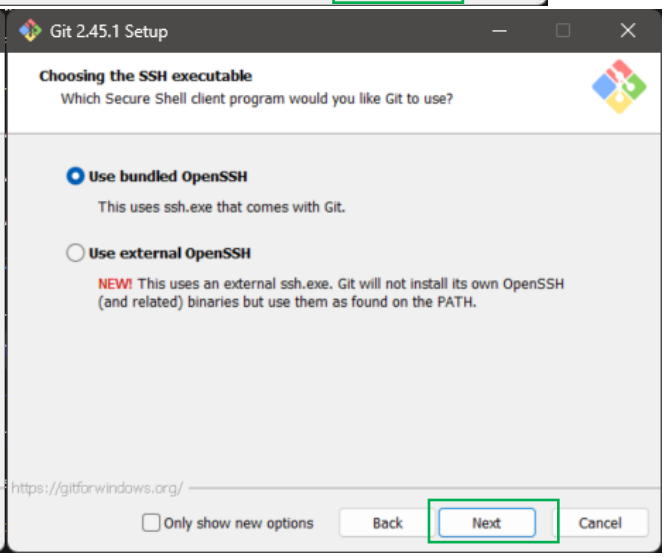


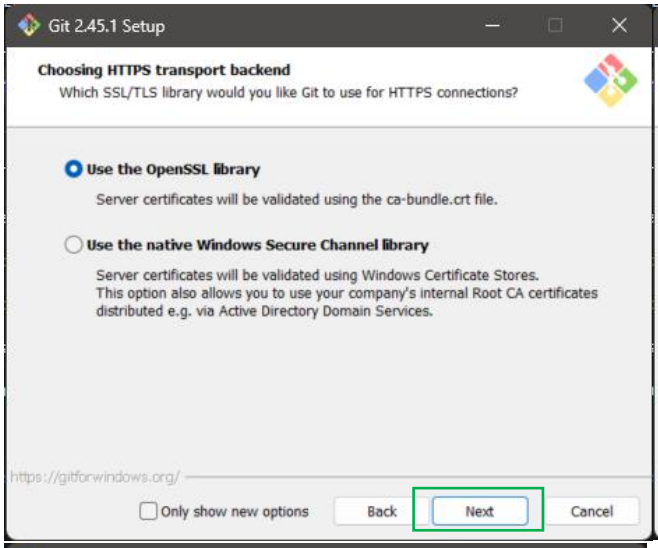


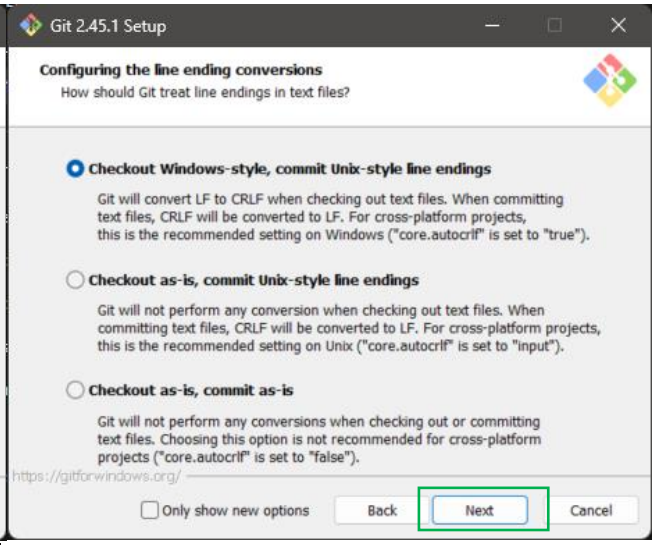


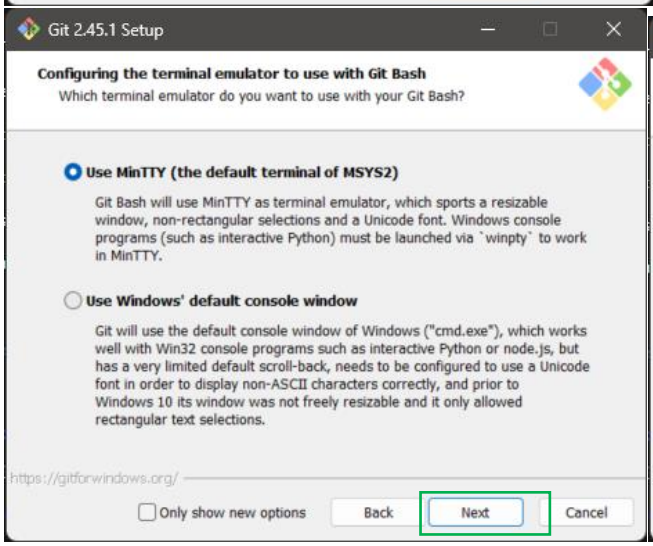


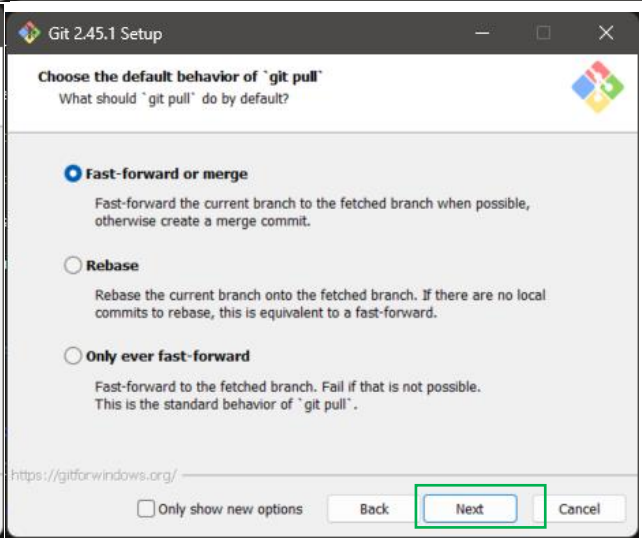


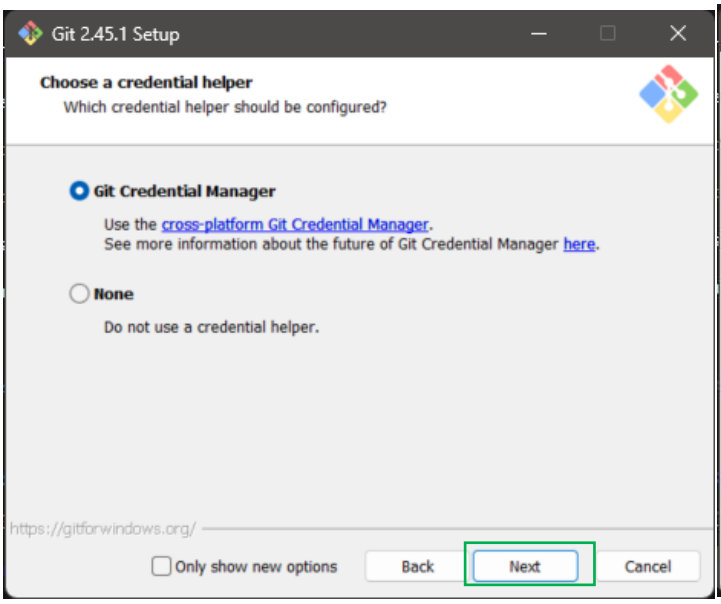


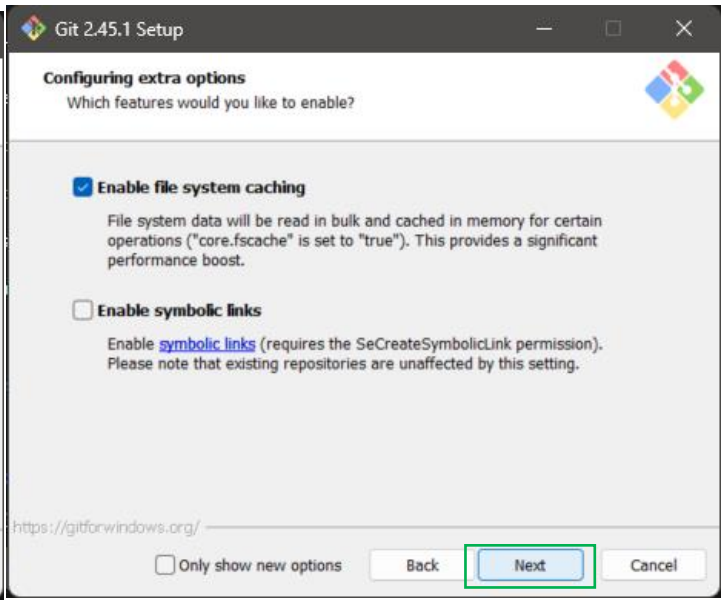

Git 2.45.1 Setup
Choosing HTTPS transport backend
Which SSL/TLS library would you like Git to use for HTTPS connections?
Use the OpenSSL library
Server certificates will be validated using the ca-bundle.crt file.
Use the native Windows Secure Channel library
Server certificates will be validated using Windows Certificate Stores.
This option also allows you to use your company's internal Root CA certificates
distributed e.g. via Active Directory Domain Services.
https://gitforwindows.org/
Only show new options
Back
Next
Cancel

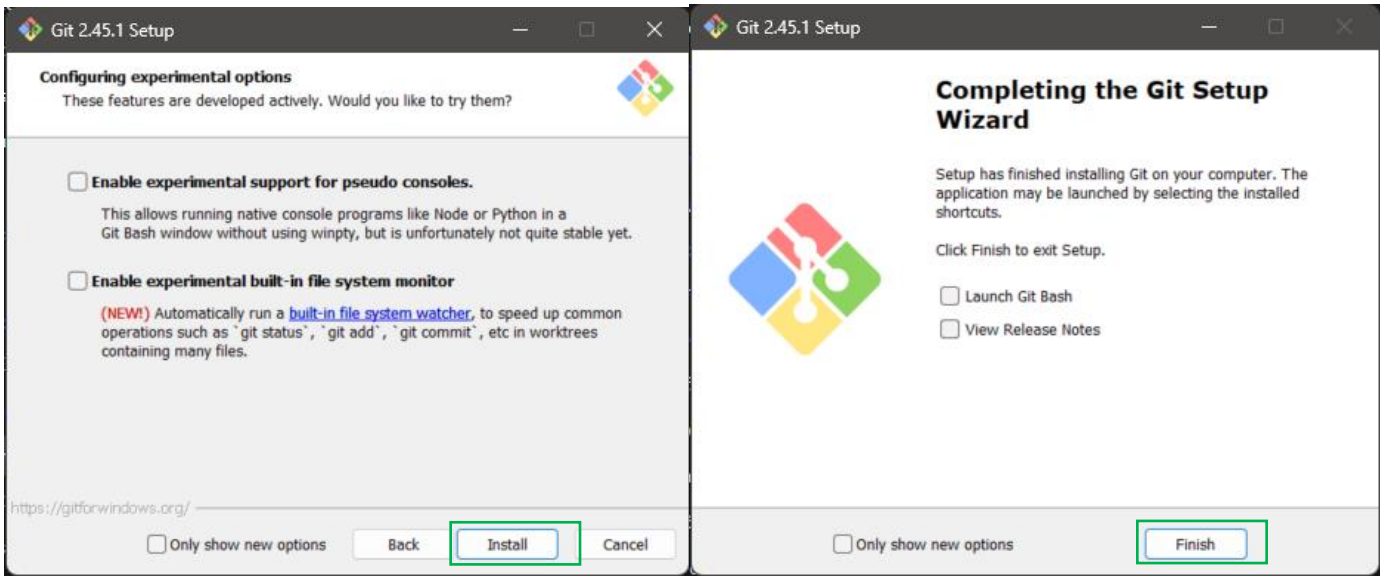

Git 2.45.1 Setup
Configuring the line ending conversions
How should Git treat line endings in text files?
Checkout Windows-style, commit Unix-style line endings
Git will convert LF to CRLF when checking out text files. When committing
text files, CRLF will be converted to LF. For cross-platform projects,
this is the recommended setting on Windows ("core.autocrlf" is set to "true").
Checkout as-is, commit Unix-style line endings
Git will not perform any conversion when checking out text files. When
committing text files, CRLF will be converted to LF. For cross-platform projects,
this is the recommended setting on Unix ("core.autocrlf" is set to "input").
Checkout as-is, commit as-is
Git will not perform any conversions when checking out or committing
text files. Choosing this option is not recommended for cross-platform
projects ("core.autocrlf" is set to "false").
https://gitforwindows.org/
Only show new options
Back
Next
Cancel


Git 2.45.1 Setup
Configuring the terminal emulator to use with Git Bash
Which terminal emulator do you want to use with your Git Bash?
Use MinTTY (the default terminal of MSYS2)
Git Bash will use MinTTY as terminal emulator, which sports a resizable
window, non-rectangular selections and a Unicode font. Windows console
programs (such as interactive Python) must be launched via `winpty` to work
in MinTTY.
Use Windows' default console window
Git will use the default console window of Windows ("cmd.exe"), which works
well with Win32 console programs such as interactive Python or node.js, but
has a very limited default scroll-back, needs to be configured to use a Unicode
font in order to display non-ASCII characters correctly, and prior to
Windows 10 its window was not freely resizable and it only allowed
rectangular text selections.
https://gitforwindows.org/
Only show new options
Back
Next
Cancel


Git 2.45.1 Setup
Choose the default behavior of `git pull`
What should `git pull` do by default?
Fast-forward or merge
Fast-forward the current branch to the fetched branch when possible,
otherwise create a merge commit.
Rebase
Rebase the current branch onto the fetched branch. If there are no local
commits to rebase, this is equivalent to a fast-forward.
Only ever fast-forward
Fast-forward to the fetched branch. Fail if that is not possible.
This is the standard behavior of `git pull`.
https://gitforwindows.org/
Only show new options
Back
Next
Cancel

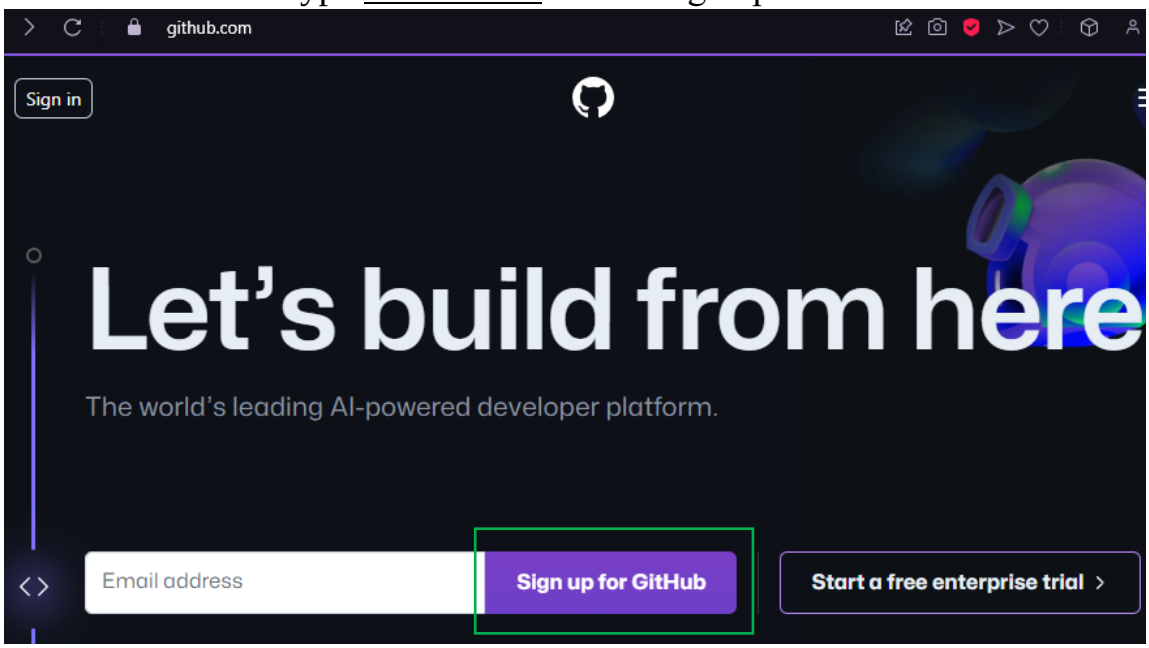

Git 2.45.1 Setup
Choose a credential helper
Which credential helper should be configured?
Git Credential Manager
Use the cross-platform Git Credential Manager.
See more information about the future of Git Credential Manager here.
None
Do not use a credential helper.
https://gitforwindows.org/
Only show new options
Back
Next
Cancel


Git 2.45.1 Setup
Configuring extra options
Which features would you like to enable?
Enable file system caching
File system data will be read in bulk and cached in memory for certain
operations ("core.fscache" is set to "true"). This provides a significant
performance boost.
Enable symbolic links
Enable symbolic links (requires the SeCreateSymbolicLink permission).
Please note that existing repositories are unaffected by this setting.
https://gitforwindows.org/
Only show new options
Back
Next
Cancel

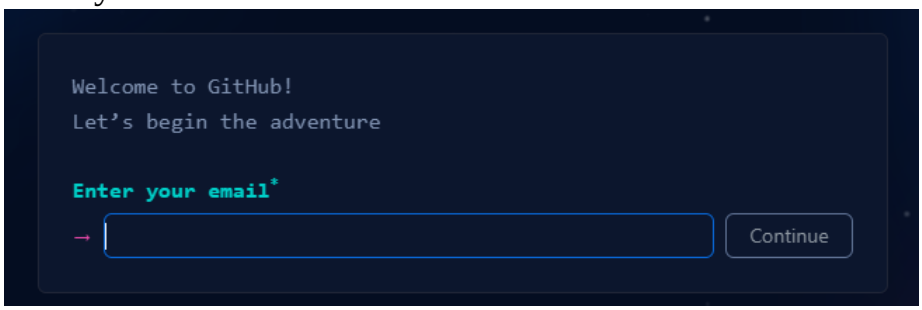


IV. Creating A GitHub Account

✓ Go to browser and type Github.com. Click 'sign up for GitHub'



✓ Enter your email

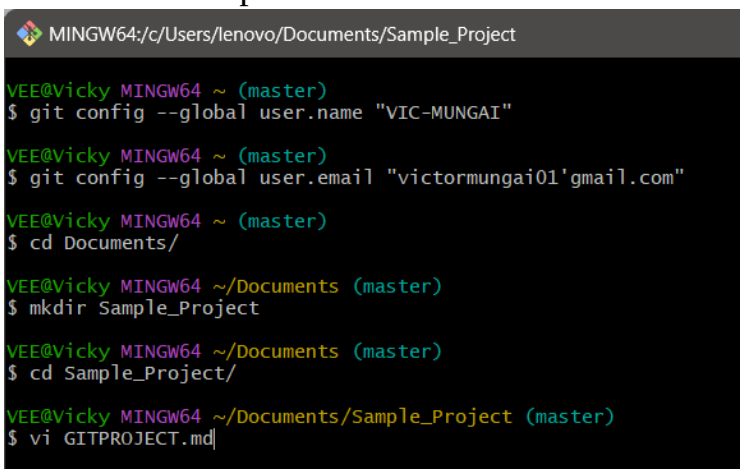


✓ Create a Password and username

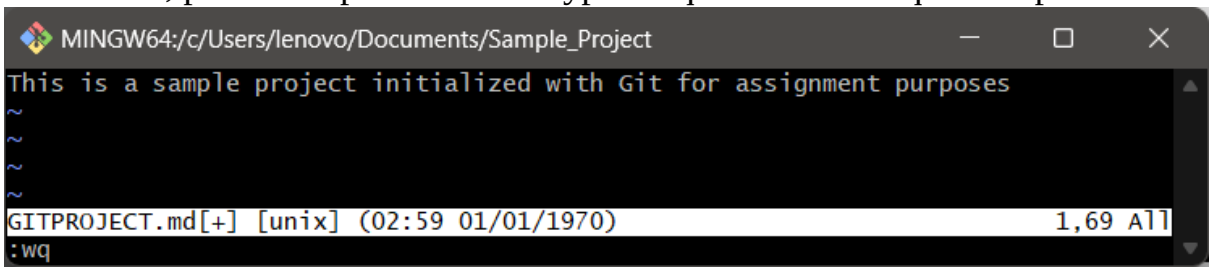
✓ Click on continue and verify your GitHub account

V. Sample Project Initialized with Git

✓ Run Git Bash as administrator
✓ To configure git user: 'git config --global user.name "VIC-MUNGAI" '
✓ Configure the email: 'git config –global user.email victormungai01@gmail.com '
✓ To navigate to a directory: 'cd Documents/'
✓ To create a folder: 'mkdir Sample_Project'
✓ To navigate to that folder: 'cd Sample_Project/'
✓ To create a sample file in this folder: 'vi GITPROJECT.md'



✓ Press 'I' to insert and type 'This is a sample project initialized with Git for assignment purposes'
✓ Once done, press Escape button and type ':wq' to write and quit and press enter



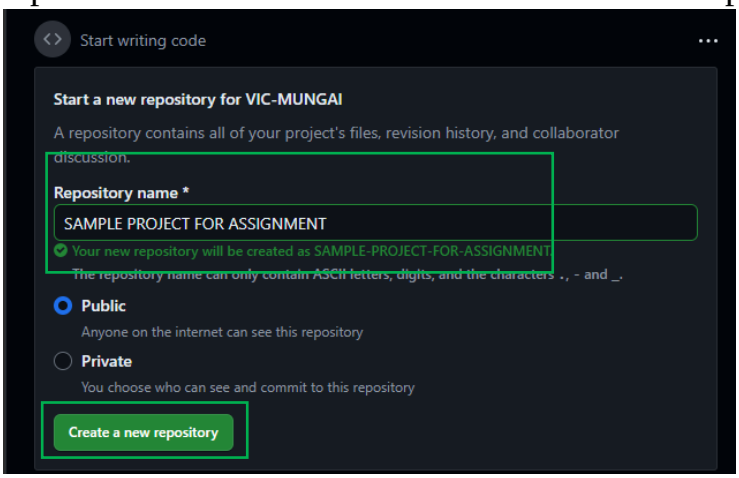
✓ To make this a git repository: 'git init'
✓ To see the files there: 'git status'

```
MINGW64:/c/Users/lenovo/Documents/Sample_Project

VEE@Vicky MINGW64 ~/Documents/Sample_Project (master)
$ git init
Initialized empty Git repository in C:/Users/lenovo/Documents/Sample_Project/.gi

VEE@Vicky MINGW64 ~/Documents/Sample_Project (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        GITPROJECT.md

nothing added to commit but untracked files present (use "git add" to track)
```

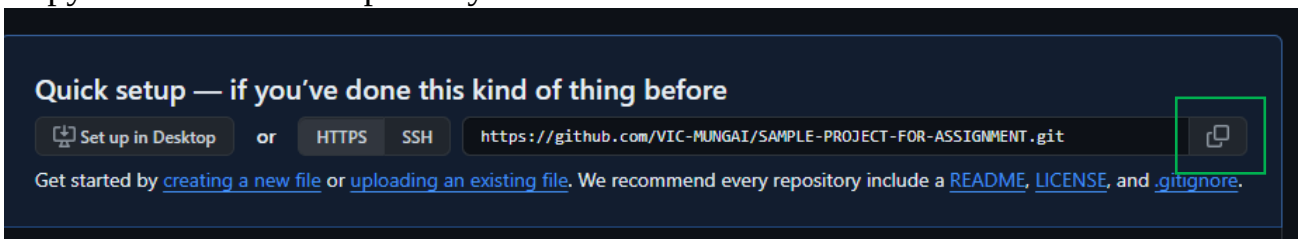
✓ To add all these files so as to track them: ‘git add .’

```
VEE@Vicky MINGW64 ~/Documents/Sample_Project (master)
$ git add .
warning: in the working copy of 'GITPROJECT.md', LF will be replaced by CRLF the
 next time Git touches it
```

✓ Open GitHub in the browser and create a new repository.



✓ Copy the link for that repository



✓ In Git Bash, to start pushing the project to GitHub,
follow these commands: ‘git add .’
‘git commit -m “Committing my sample project” ’
‘git remote add origin ‘URLcopied’ ’
‘ git push -u origin
branchname ’

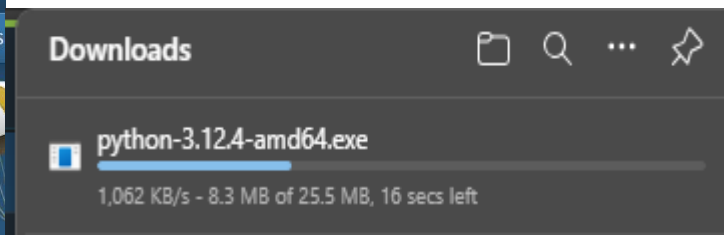
```
VEE@Vicky MINGW64 ~/Documents/Sample_Project (master)
$ git commit -m "committing my sample project with .gitignore included"
[master 26d42ac] committing my sample project with .gitignore included
 1 file changed, 2 insertions(+)
 create mode 100644 .gitignore

VEE@Vicky MINGW64 ~/Documents/Sample_Project (master)
$ git remote add origin https://github.com/VIC-MUNGAI/SAMPLE-PROJECT-FOR-ASSIGNM
ENT.git
```

```
VEE@Vicky MINGW64 ~/Documents/Sample_Project (master)
$ git push -u origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 617 bytes | 205.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/VIC-MUNGAI/SAMPLE-PROJECT-FOR-ASSIGNMENT.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

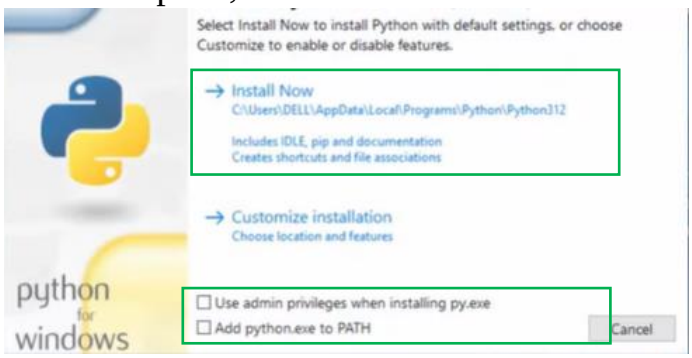
✓ Successfully created a GitHub repository initialized with Git. The link to the repository: https://github.com/VIC-MUNGAI/SAMPLE-PROJECT-FOR-ASSIGNMENT.git

VI. Downloading & Installing Python

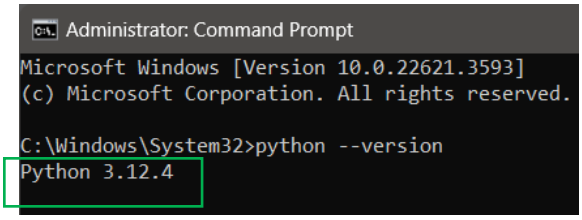
✓ Click on this link to direct to Python website. https://www.python.org/downloads/ Click 'Download Python 3.12.4'. Download starts after clicking.



✓ Once complete, locate it and run it. Click on install now after ticking these boxes



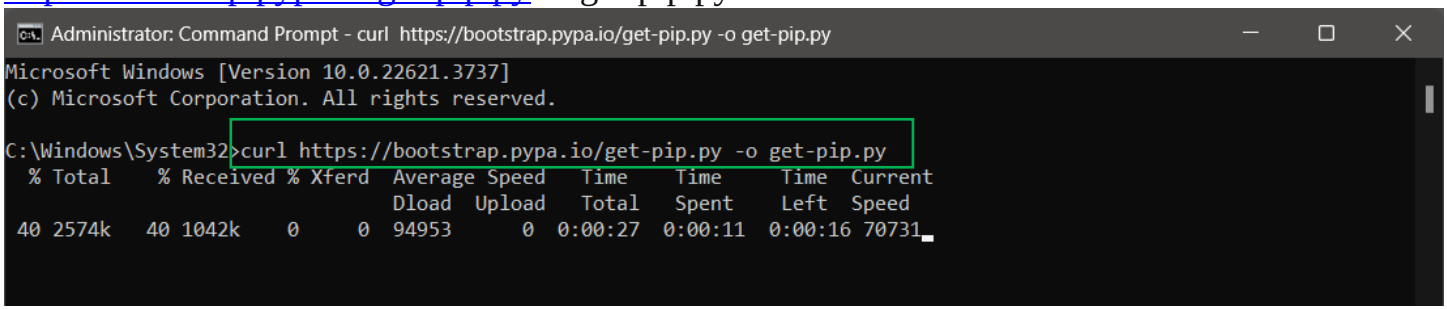
✓ Run the CMD as admin and run this command 'python --version'



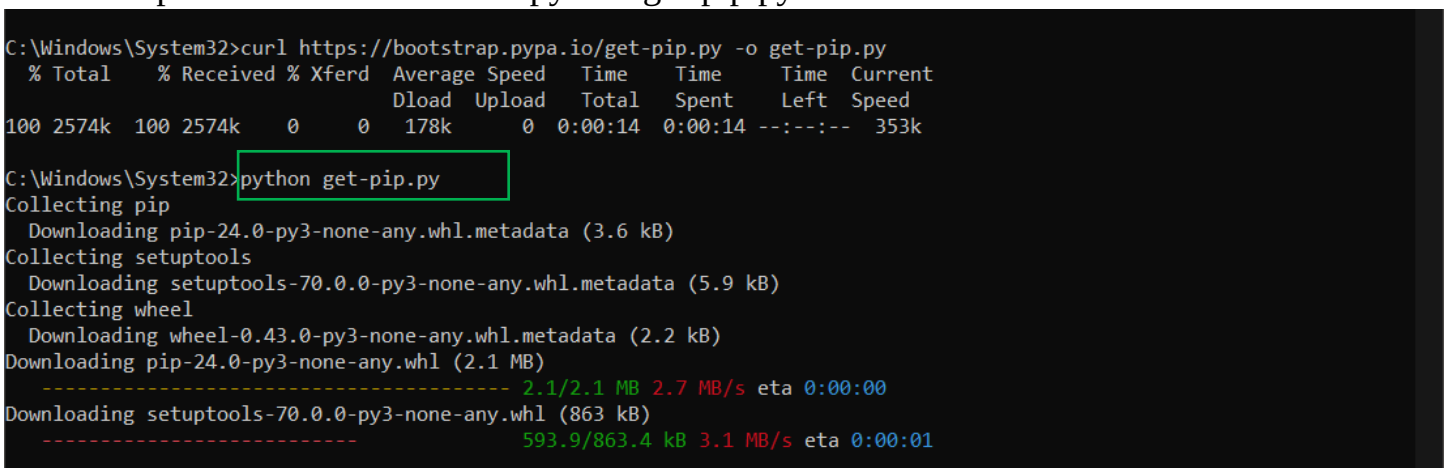
✓ Python successfully installed.

VII. Installing Pip in Python

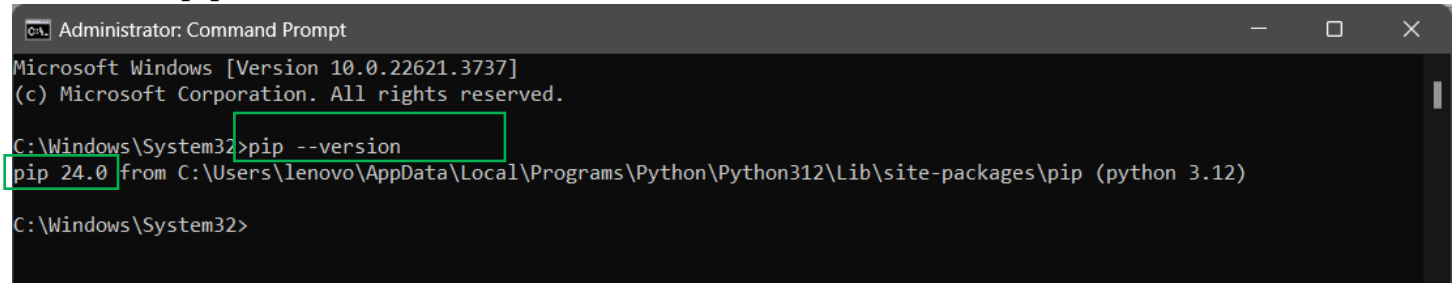
✓ Run command prompt as administrator. Run this command ‘curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py’



✓ Once complete run this command ‘python get-pip.py’

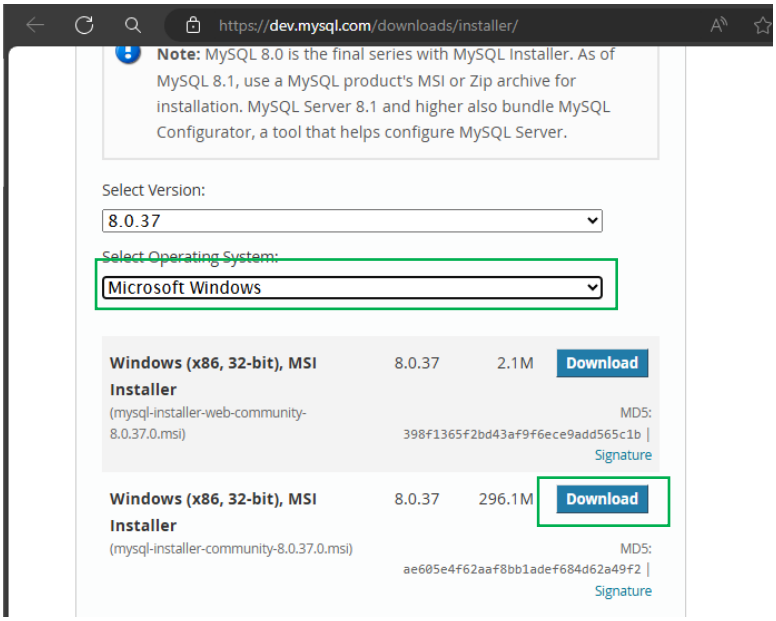


✓ The installation will successfully complete. To check the version installed, run this command ‘pip --version’

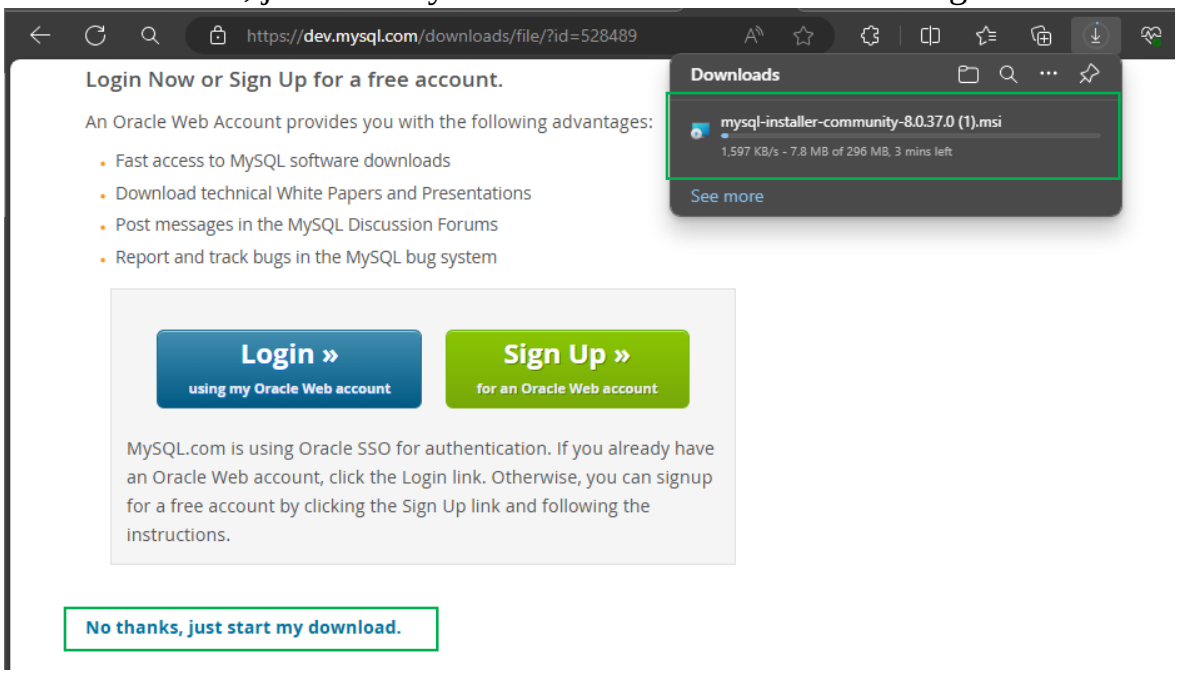


VIII. Downloading & Installing MYSQL Database

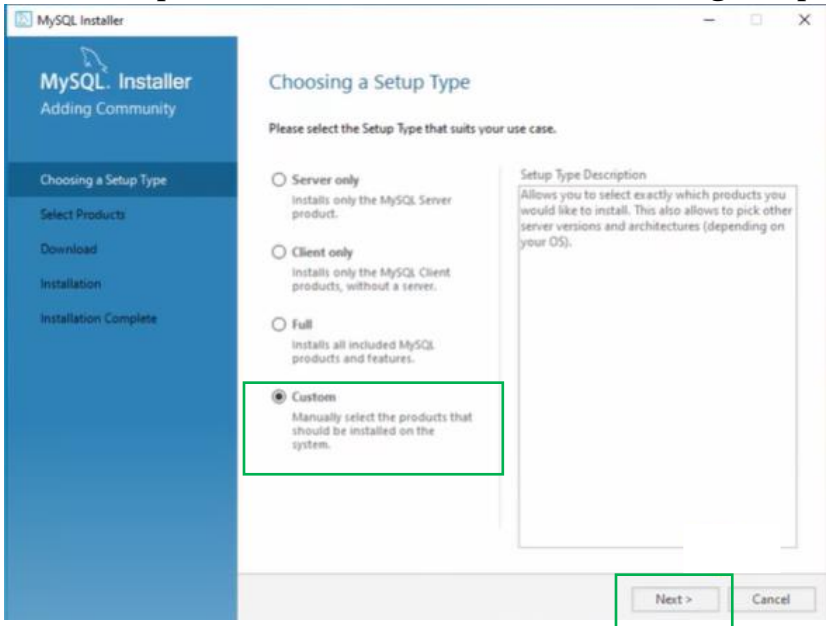
✓ Click on this link to direct to MYSQL website https://dev.mysql.com/downloads/installer/
Under select OS, select Microsoft Windows, then click download for 'Windows (x86,)'



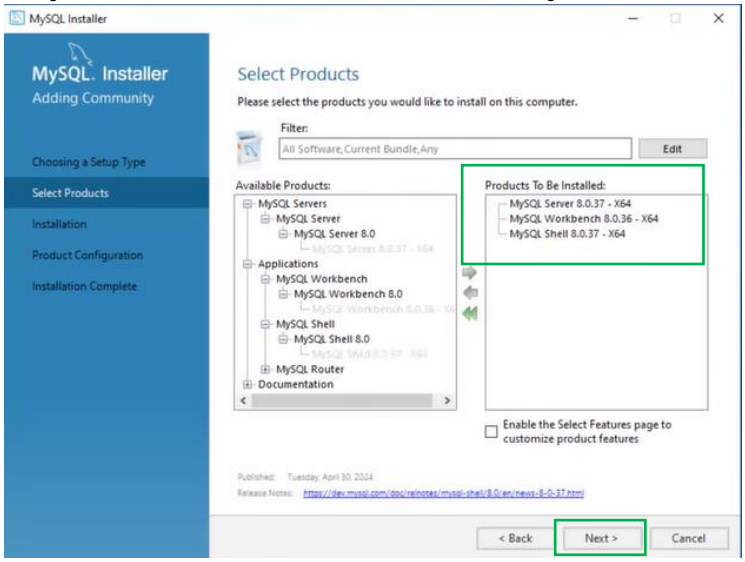
✓ Click No thanks, just start my download. The download will begin



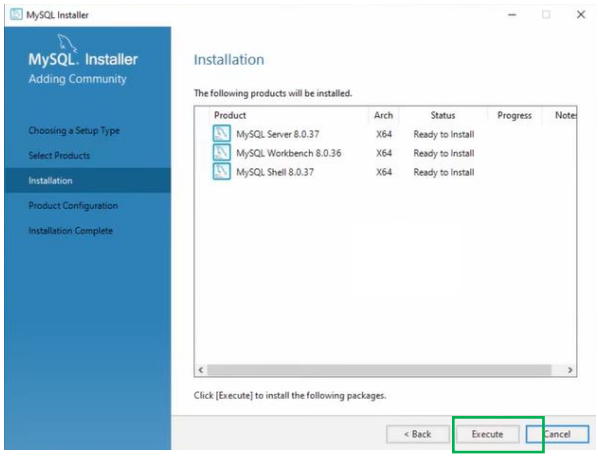
✓ Once complete, locate it and run it. On choosing setup type, Click on custom then next



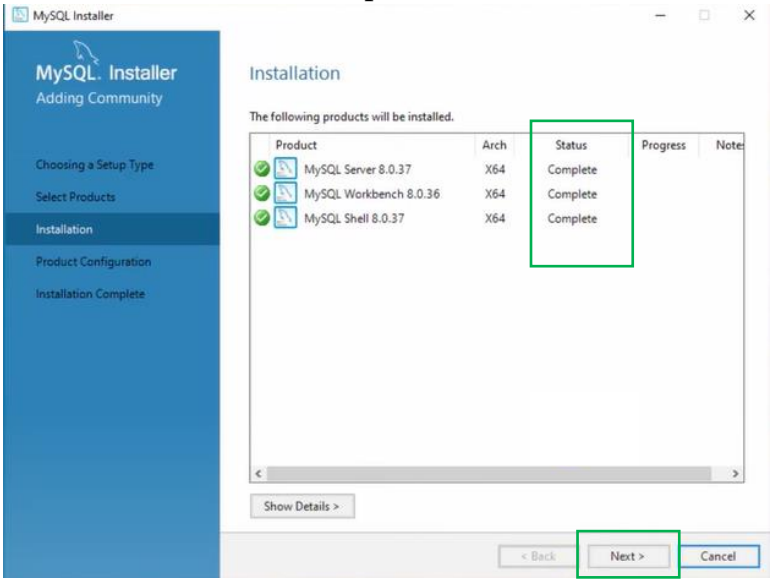
✓ On select products, select ‘MYSQL server 8.0.37’ under MySQL servers. Also select ‘MySQL workbench 8.0.36’ and ‘MySQL Shell 8.0.37’ under applications. Click next



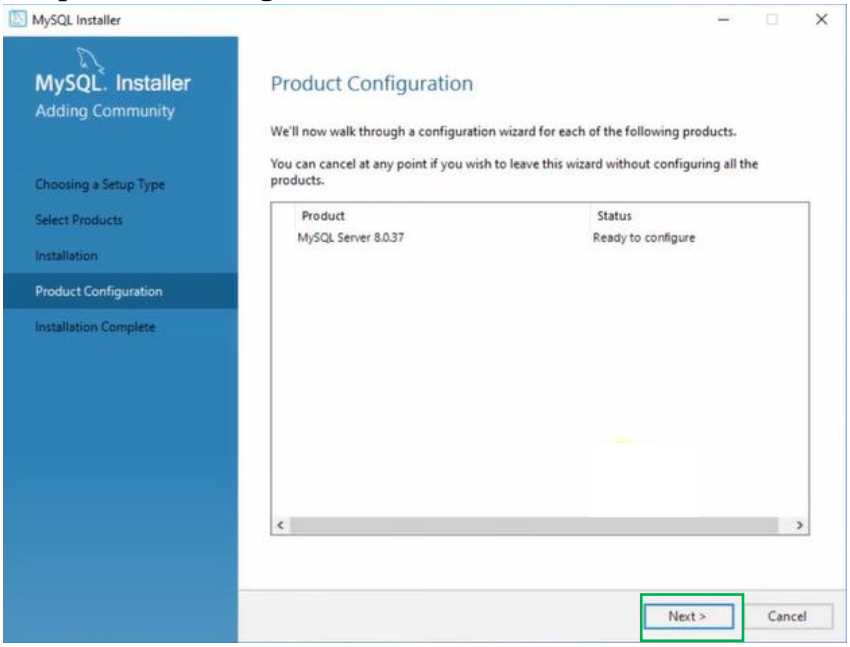
✓ Under installation, click execute



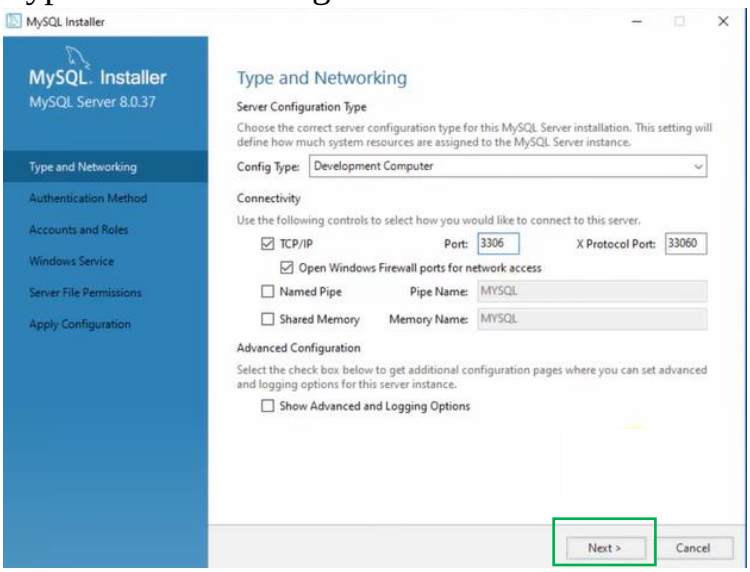
✓ Once installation is complete, click next



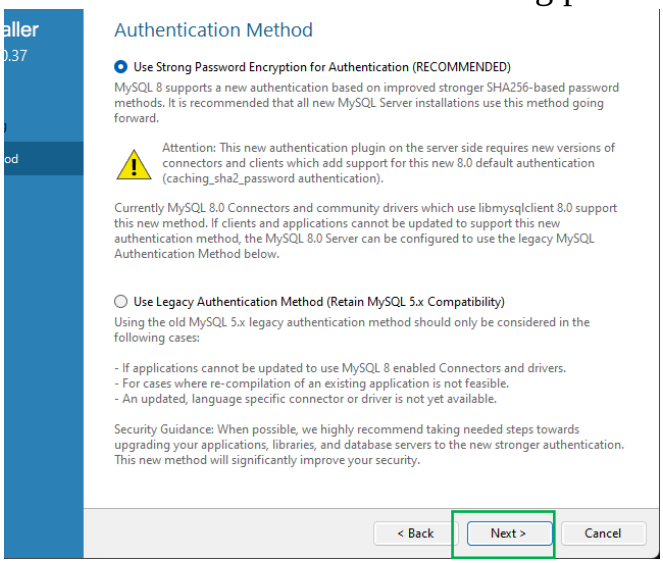
✓ On product configuration click next



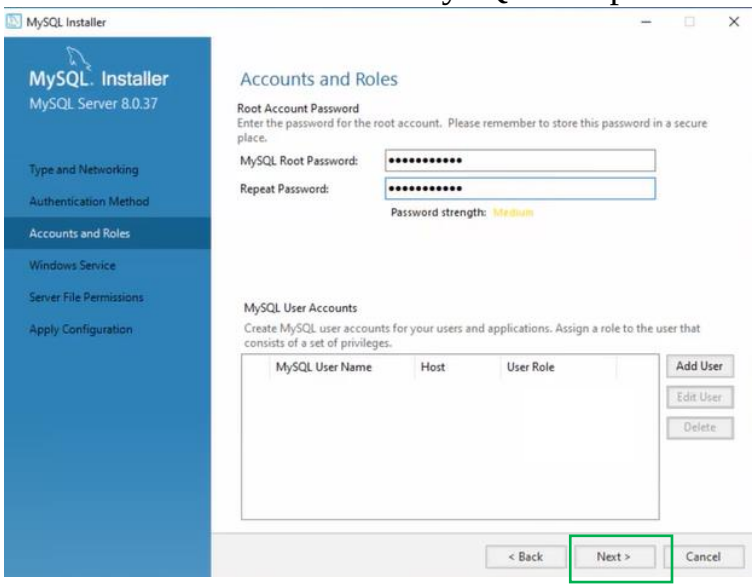
✓ Type and Networking: click next



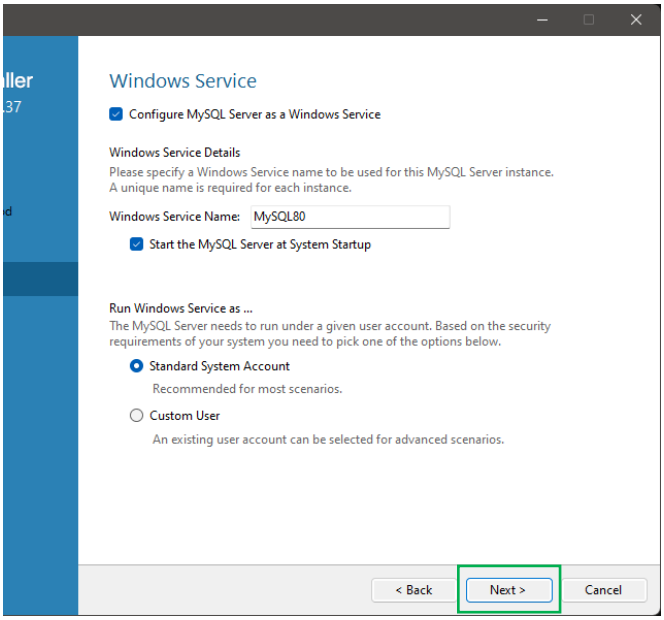
✓ Authentication method: Use a strong password encryption for authentication. Click next



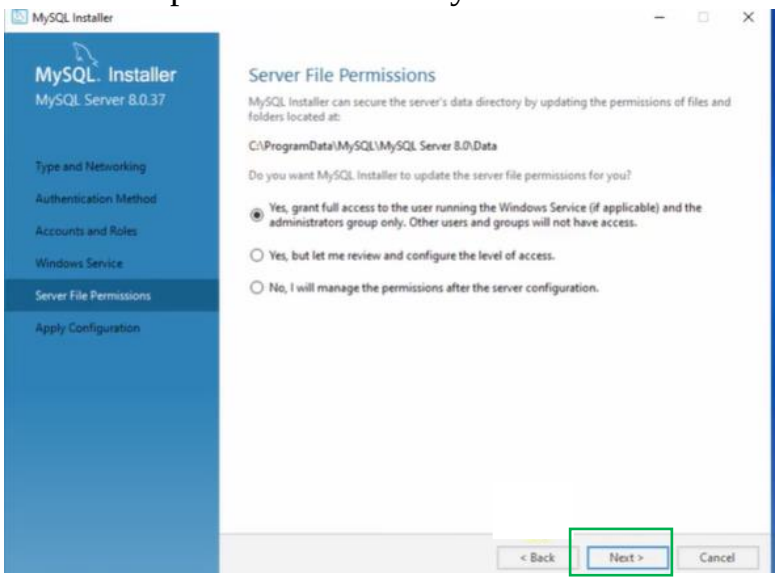
✓ Accounts and roles: create MySQL root password and repeat it. Click next



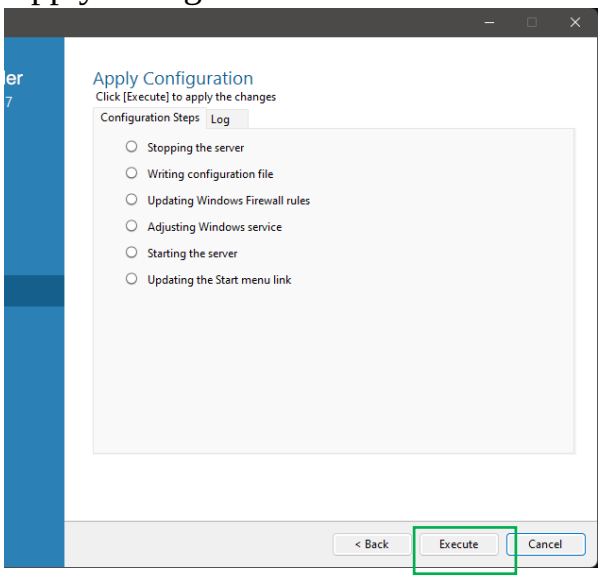
✓ Windows service: click next



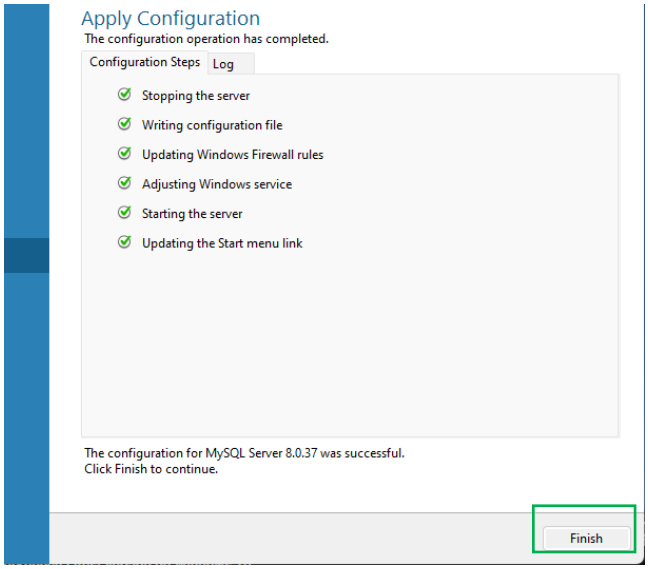
✓ Server file permissions: click yes then next



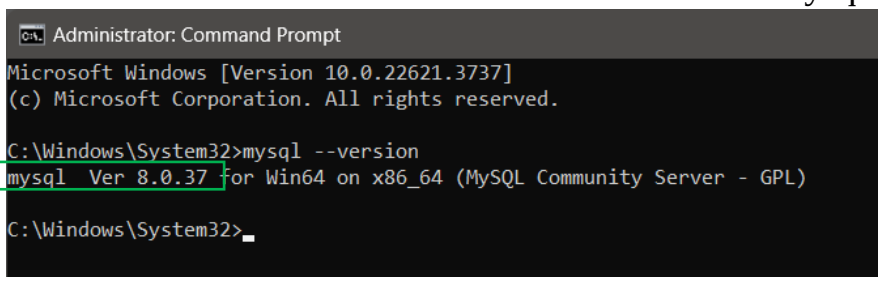
✓ Apply configuration: click execute



✓ Click finish



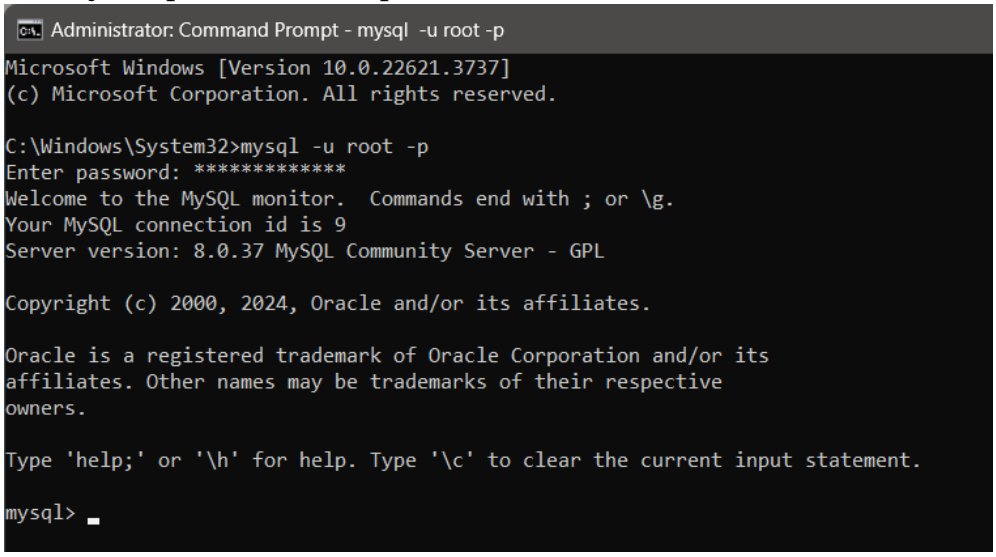
✓ Run CMD as administrator and run this command ‘mysql --version’



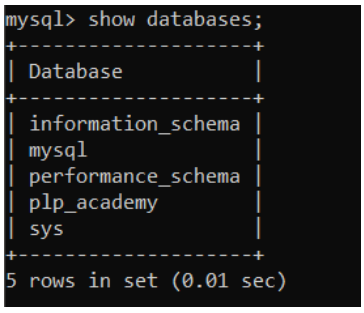
✓ Successfully installed

IX. Configuring A Database

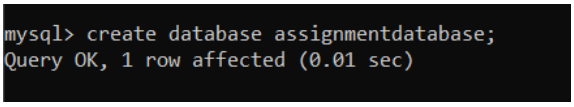
✓ Run CMD as administrator. To start MySQL server, run this command ‘mysql -u root -p’. Enter your password and press enter



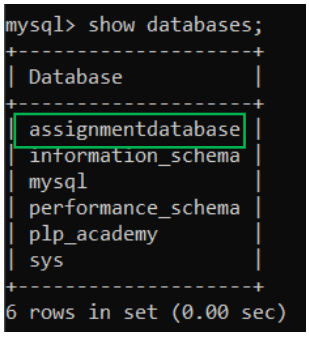
✓ Run this command to check databases ‘show databases;’



✓ To create a database, run this command ‘create database ‘databasename’;’



✓ Type ‘show databases;’ to see whether it has been created and press enter



✓ Successfully created. This can also be done using MySQL workbench.

3. CHALLENGES FACED DURING SETUP

i. Configuration complexity

There were numerous settings and parameters that were overwhelming, especially during installation of MySQL.

Solution: Sourcing others who had successfully installed their software and asking for assistance. Documenting my setup so that in future I can make amendments and install on other devices smoothly.

ii. Performance Optimization

The PC would at times lag due to overloading of the memory and performing difficult tasks

Solution: Closely monitoring resource usage. Closing apps that are not in use. Clearing temp files.

iii. Little knowledge on coding

Following the limited knowledge and skills on coding as a beginner, typos, syntax errors, logical errors, poor knowledge of variables was the order of the day.

Solution: I practised writing these codes and running commands every day. I learnt from my trainers new variables and improved on coding by learning the different coding environments and their requirements

iv. Missing steps

Errors when running a program would appear when a step was skipped, an example being failure to install MySQL workbench which would result in consuming a lot of time trying to figure out what went wrong.

Solution: Documenting the process, locating the missed step and reinstalling the program

4. CONCLUSION

Reflecting on these challenges and the strategies employed to overcome them helped to streamline the setup process and improve the development workflow. This reflection process is essential for continuous improvement and helps prevent similar issues in future projects.