Experiment-2

Aim: **To Understand Version Control System, GIT installation & Github Account**

Theory:

1. Version Control System

Version control - also known as source control or revision control - is an important software development practice for tracking and managing changes made to code and other files. It is closely related to source code management.

With version control, every change made to the code base is tracked. This allows software developers to see the entire history of who changed what at any given time — and roll back from the current version to an earlier version if they need to. It also creates a single source of truth.

Version control (or source control or revision control) serves as a safety net to protect the source code from irreparable harm, giving the development team the freedom to experiment without fear of causing damage or creating code conflicts.

If developers code concurrently and create incompatible changes, version control identifies the problem areas so that team members can quickly revert changes to a previous version, compare changes, or identify who committed the problem code through the revision history. With a version control system (VCS), a software team can solve an issue before progressing further into a project. Through code reviews, software teams can analyze earlier versions to understand the changes made to the code over time.

Depending on a team's specific needs and development process, a VCS can be local, centralized, or distributed. A local VCS stores source files within a local system, a centralized VCS stores changes in a single server, and a distributed VCS involves cloning a Git repository.

2.Benefits of version control

- Quality

  Teams can review, comment, and improve each other's code and assets.

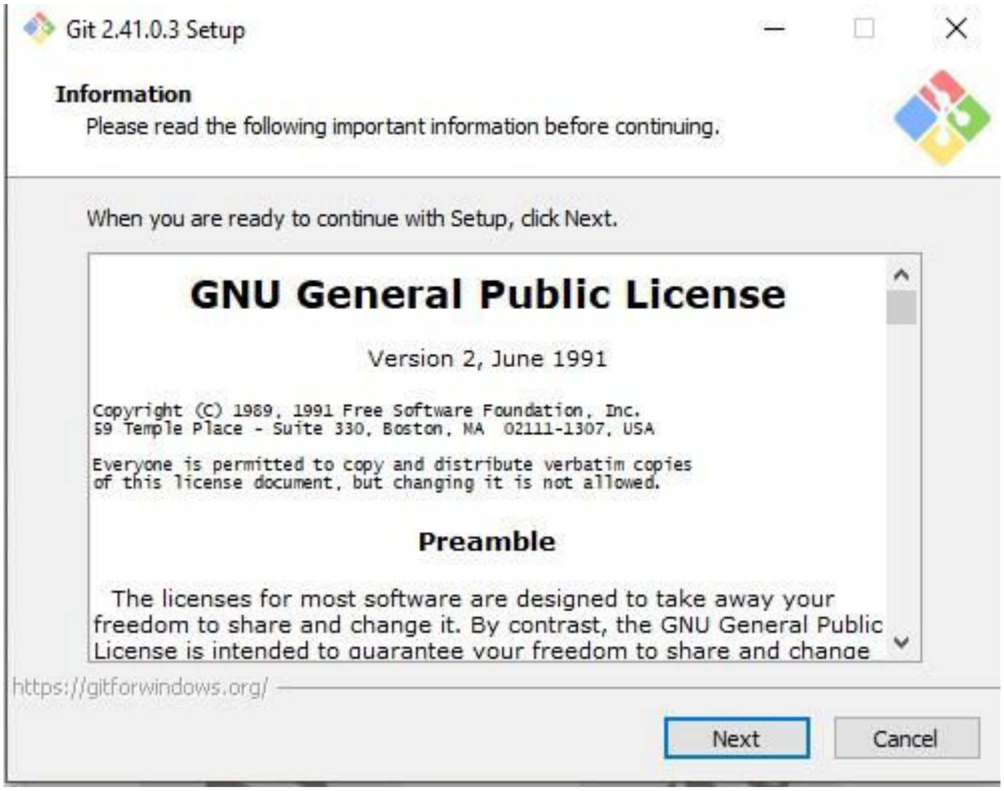
- Acceleration

  Branch code, make changes, and merge commits faster. □

Visibility

Understand and spark team collaboration to foster greater release build and release patterns. Better visibility improves everything from project management to code quality.

2. GIT Installation



Git 2.41.0.3 Setup

Information

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

GNU General Public License

Version 2, June 1991

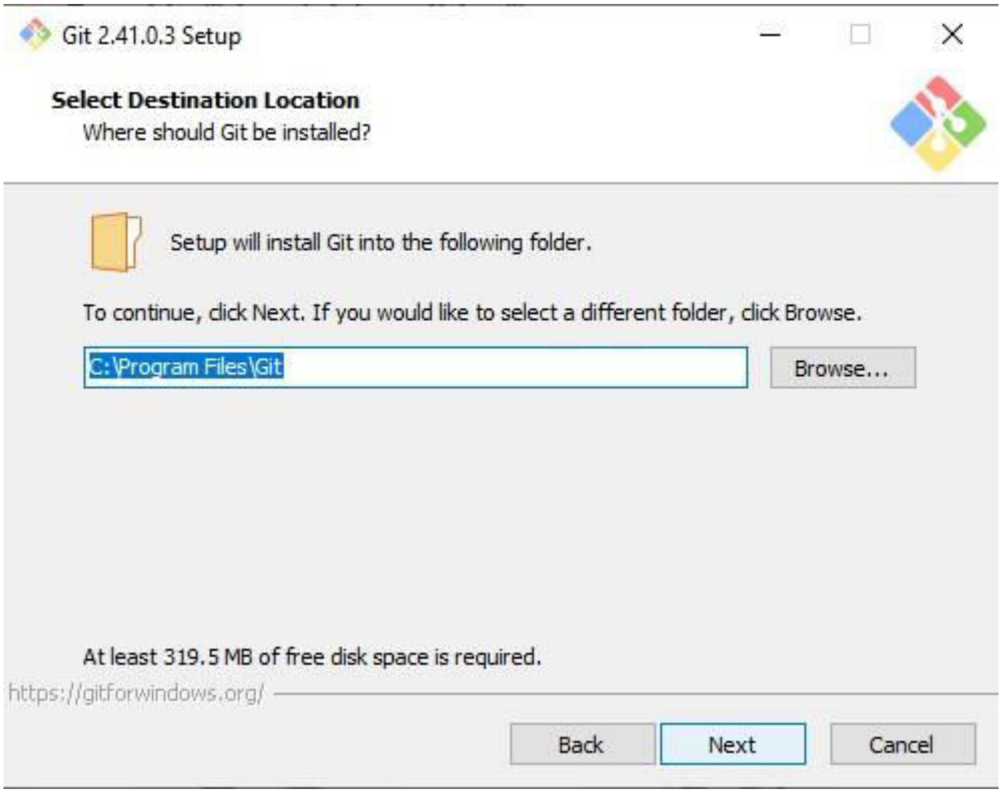
```
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA  02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
```

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change

step -1 : select the file path for installation



Git 2.41.0.3 Setup

Select Destination Location

Where should Git be installed?

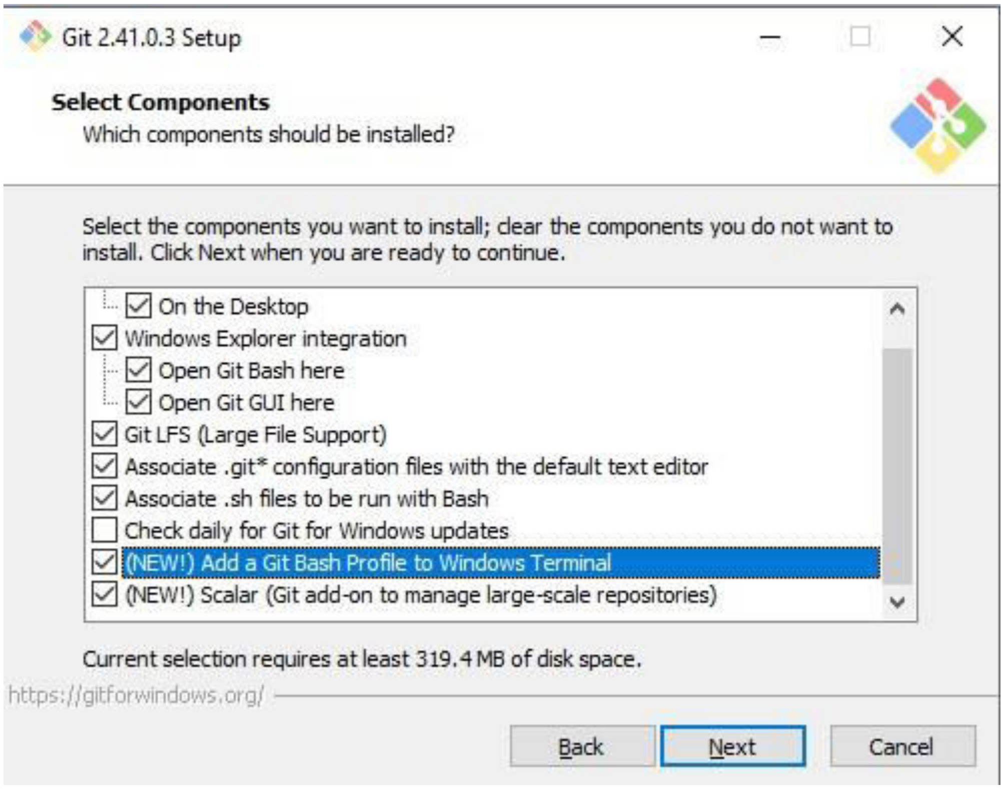
Setup will install Git into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files\Git

At least 319.5 MB of free disk space is required.

step -2 : select the required components for installation



Git 2.41.0.3 Setup

Select Components

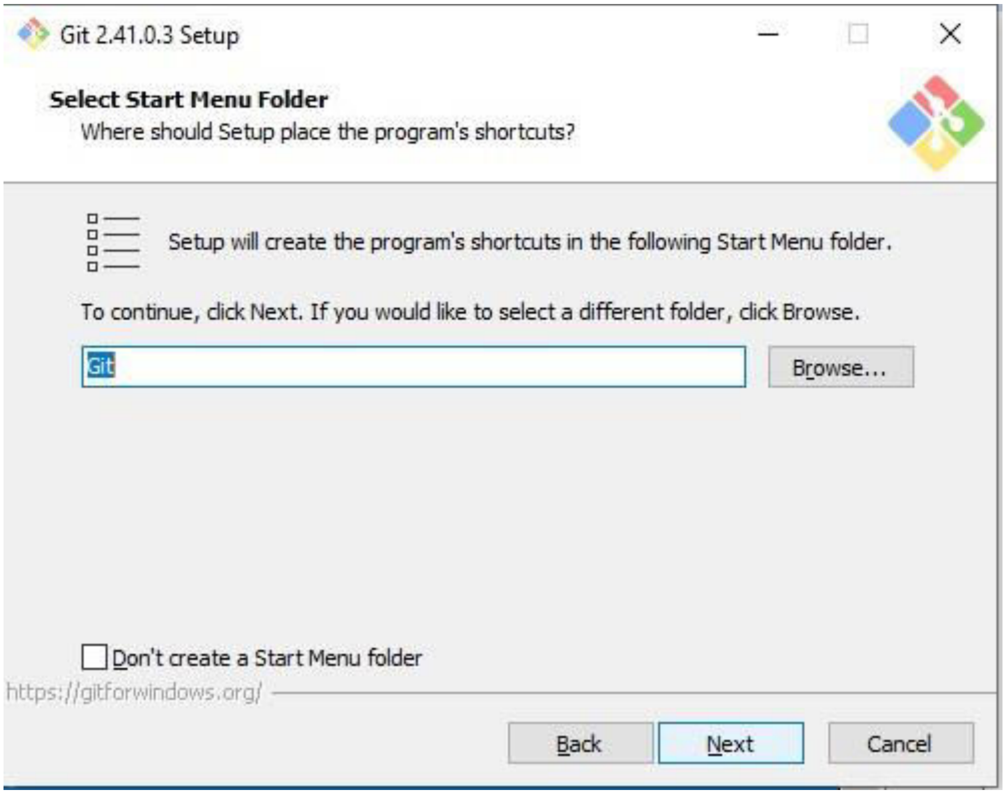
Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- ☑ On the Desktop
- ☑ Windows Explorer integration
  - ☑ Open Git Bash here
  - ☑ Open Git GUI here
- ☑ Git LFS (Large File Support)
- ☑ Associate .git* configuration files with the default text editor
- ☑ Associate .sh files to be run with Bash
- ☐ Check daily for Git for Windows updates
- ☑ (NEW!) Add a Git Bash Profile to Windows Terminal
- ☑ (NEW!) Scalar (Git add-on to manage large-scale repositories)

Current selection requires at least 319.4 MB of disk space.

step -3 : select the folder name for setup



Git 2.41.0.3 Setup

Select Start Menu Folder

Where should Setup place the program's shortcuts?

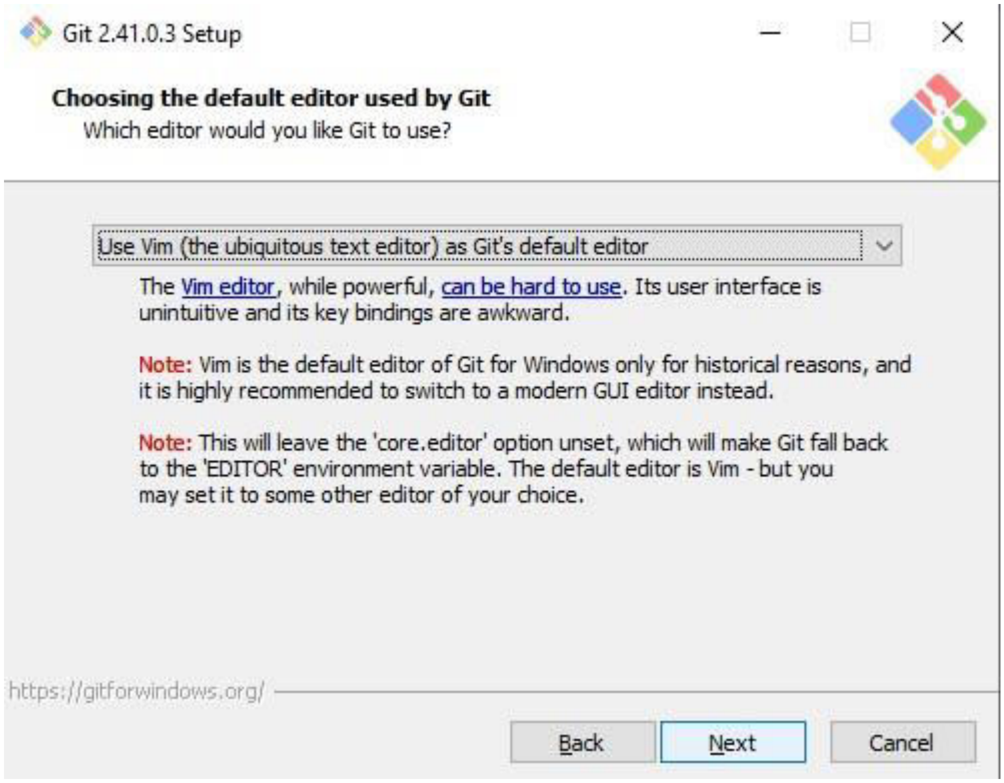
Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

Git

☐ Don't create a Start Menu folder

step -4 : select the default text editor for using Git



Git 2.41.0.3 Setup

Choosing the default editor used by Git

Which editor would you like Git to use?

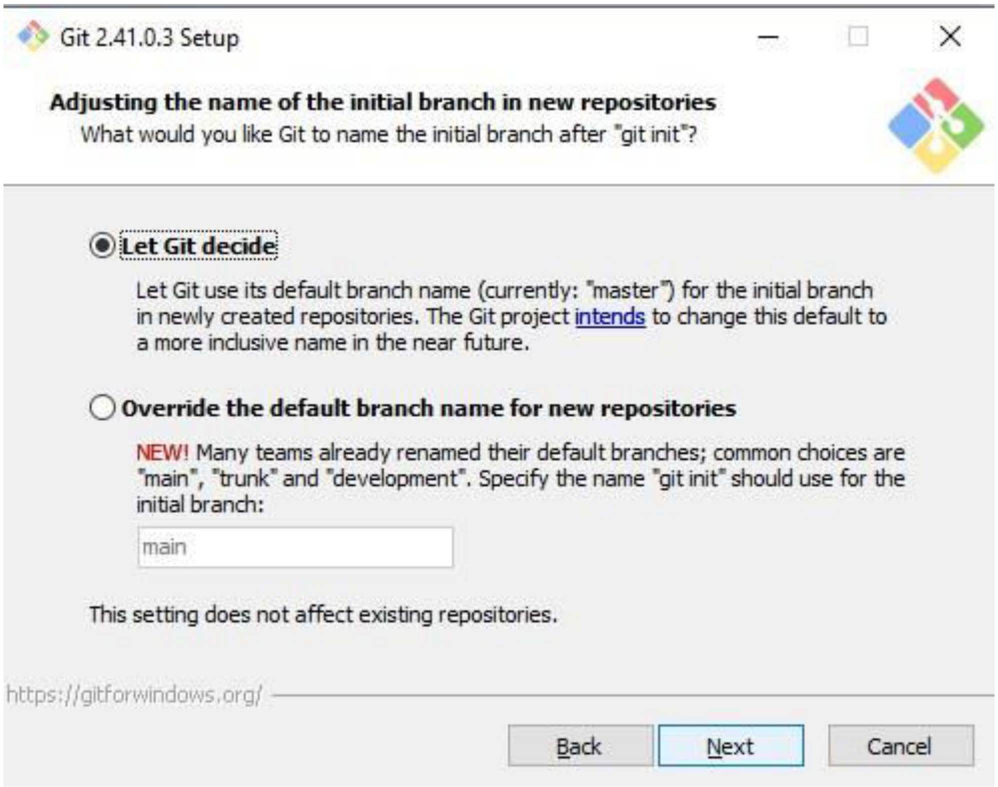
Use Vim (the ubiquitous text editor) as Git's default editor

The Vim editor, while powerful, can be hard to use. Its user interface is unintuitive and its key bindings are awkward.

Note: Vim is the default editor of Git for Windows only for historical reasons, and it is highly recommended to switch to a modern GUI editor instead.

Note: This will leave the 'core.editor' option unset, which will make Git fall back to the 'EDITOR' environment variable. The default editor is Vim - but you may set it to some other editor of your choice.

step -5 : setup required formats



Git 2.41.0.3 Setup

Adjusting the name of the initial branch in new repositories

What would you like Git to name the initial branch after "git init"?

● **Let Git decide**

Let Git use its default branch name (currently: "master") for the initial branch in newly created repositories. The Git project intends to change this default to a more inclusive name in the near future.

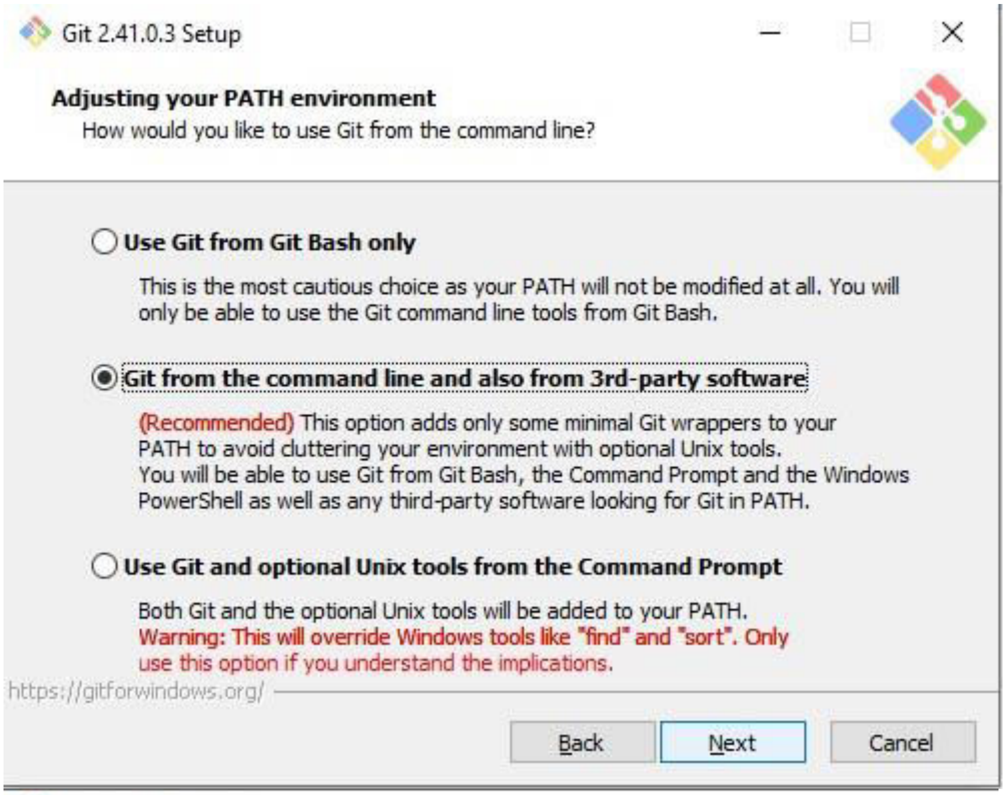
○ **Override the default branch name for new repositories**

NEW! Many teams already renamed their default branches; common choices are "main", "trunk" and "development". Specify the name "git init" should use for the initial branch:

main

This setting does not affect existing repositories.

step-6 : Authorise necessary permissions



Git 2.41.0.3 Setup

Adjusting your PATH environment

How would you like to use Git from the command line?

○ **Use Git from Git Bash only**

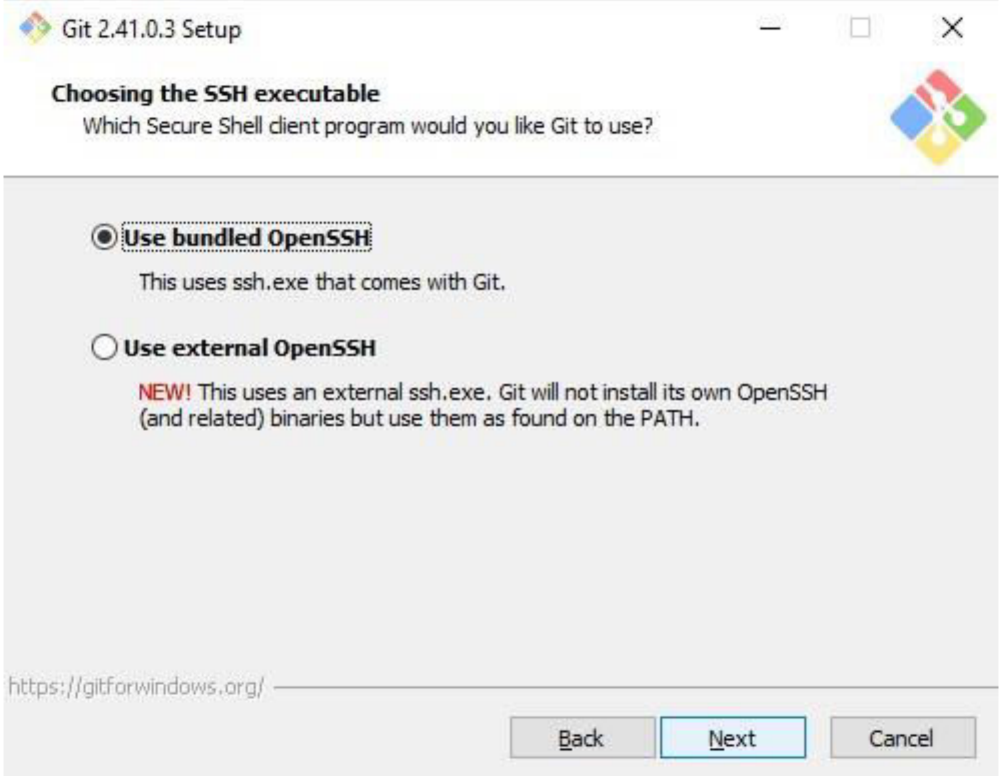
This is the most cautious choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.

● **Git from the command line and also from 3rd-party software**

(Recommended) This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools.
You will be able to use Git from Git Bash, the Command Prompt and the Windows PowerShell as well as any third-party software looking for Git in PATH.

○ **Use Git and optional Unix tools from the Command Prompt**

Both Git and the optional Unix tools will be added to your PATH.
Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.



Git 2.41.0.3 Setup

Choosing the SSH executable

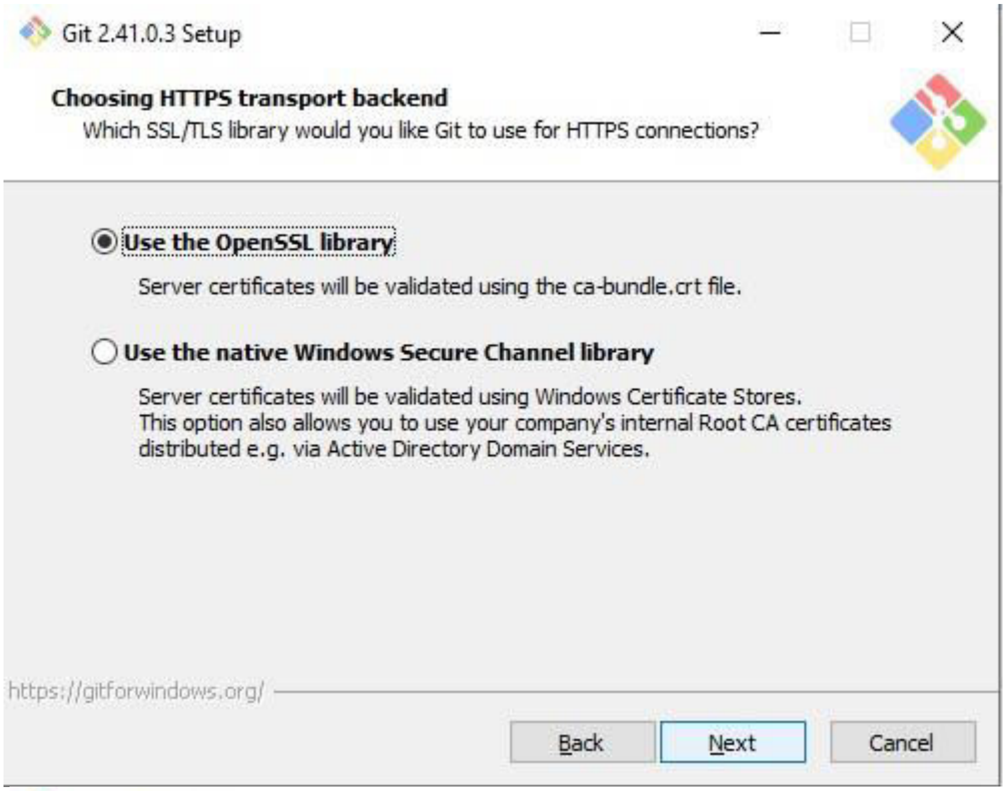
Which Secure Shell client program would you like Git to use?

● **Use bundled OpenSSH**

This uses ssh.exe that comes with Git.

○ **Use external OpenSSH**

NEW! This uses an external ssh.exe. Git will not install its own OpenSSH (and related) binaries but use them as found on the PATH.



Choosing HTTPS transport backend

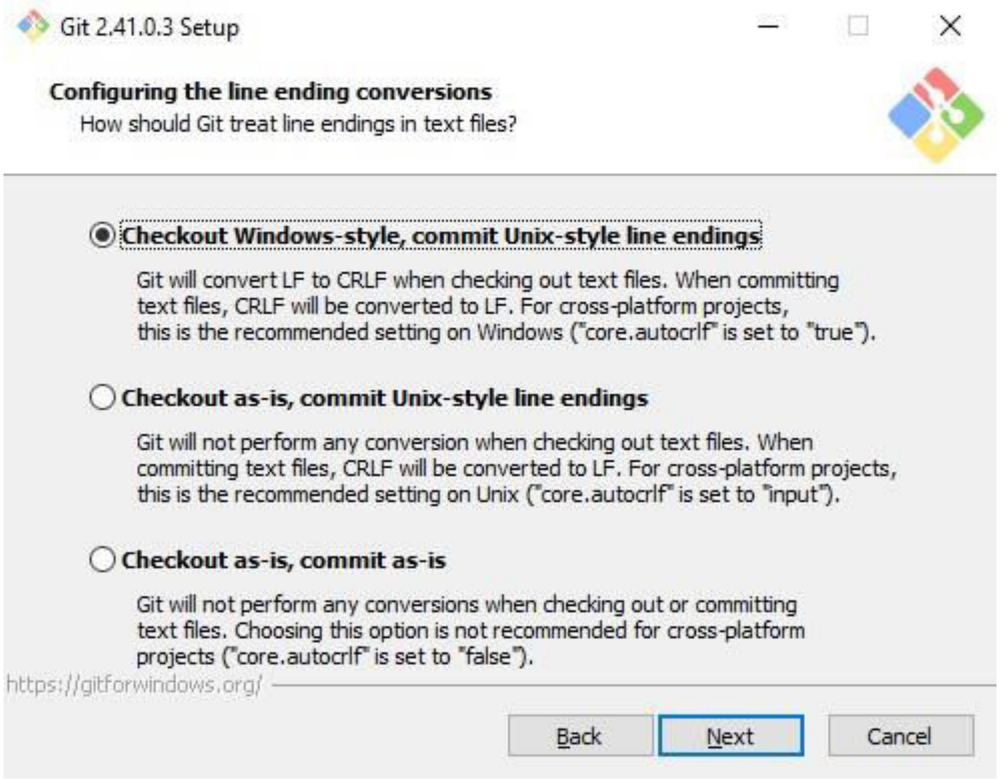
Which SSL/TLS library would you like Git to use for HTTPS connections?

● **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.



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").

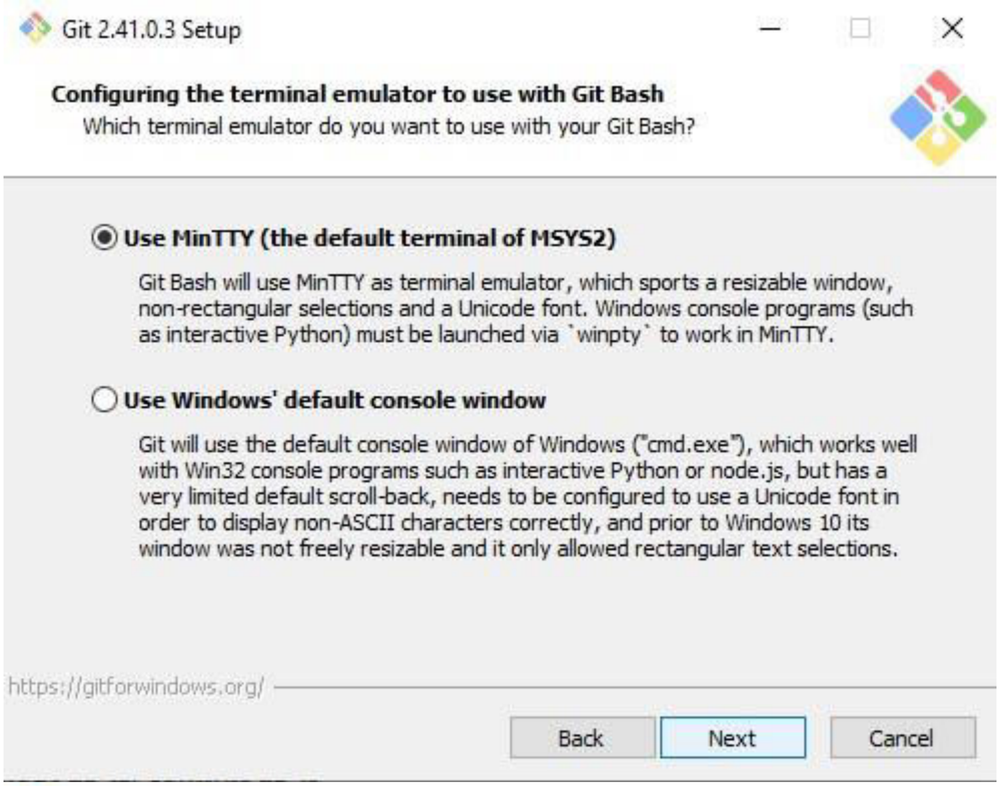
○ **Checkout as-is, commit Unix-style line endings**

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").

Step-7: Configure the BASH terminal



Git 2.41.0.3 Setup

Configuring the terminal emulator to use with Git Bash

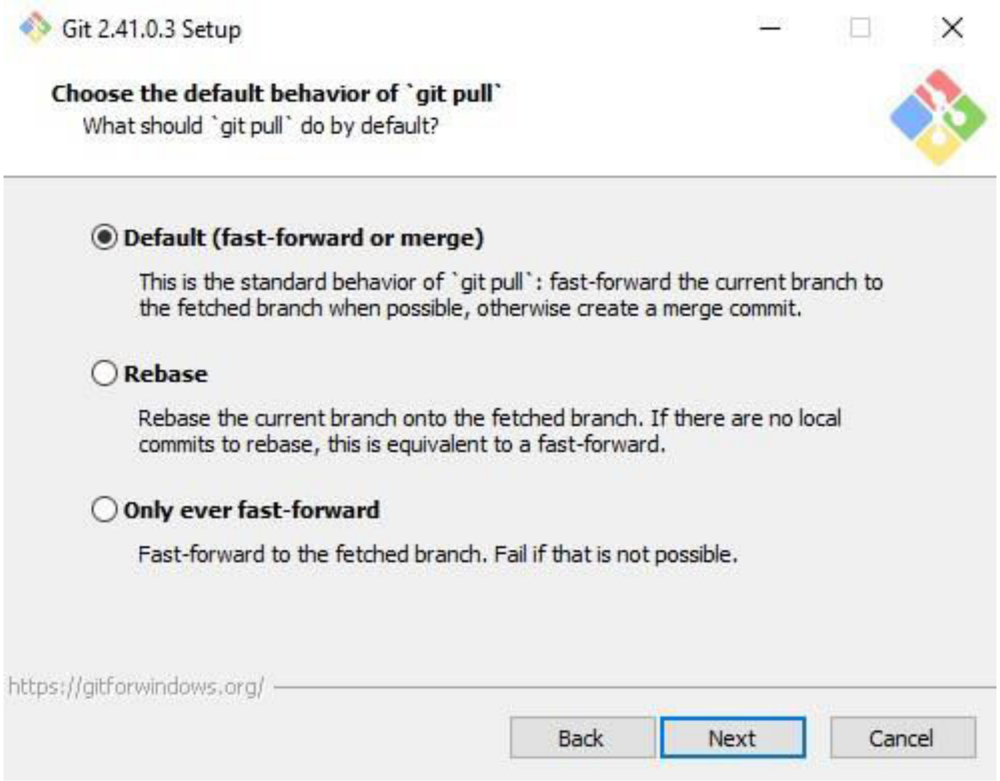
Which terminal emulator do you want to use with your Git Bash?

● **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.



Git 2.41.0.3 Setup

Choose the default behavior of `git pull`

What should `git pull` do by default?

● **Default (fast-forward or merge)**

This is the standard behavior of `git pull`: fast-forward the current branch to the fetched branch when possible, otherwise create a merge commit.

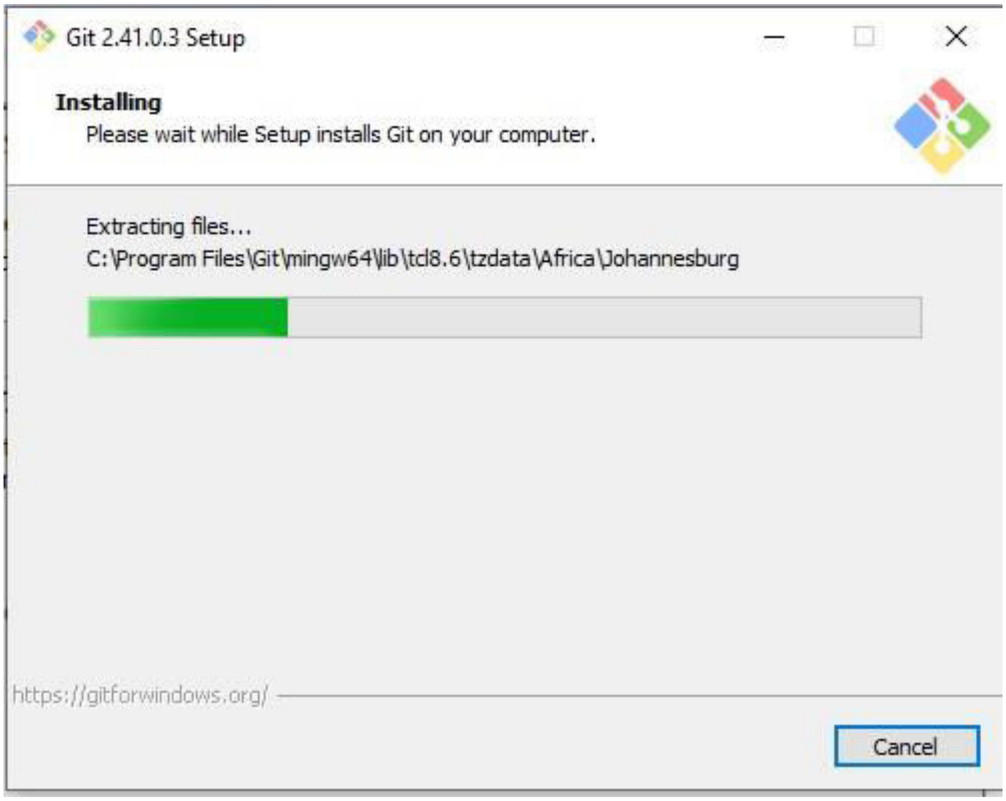
○ **Rebase**

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.

Step-8: Click Install


Git 2.41.0.3 Setup

Installing

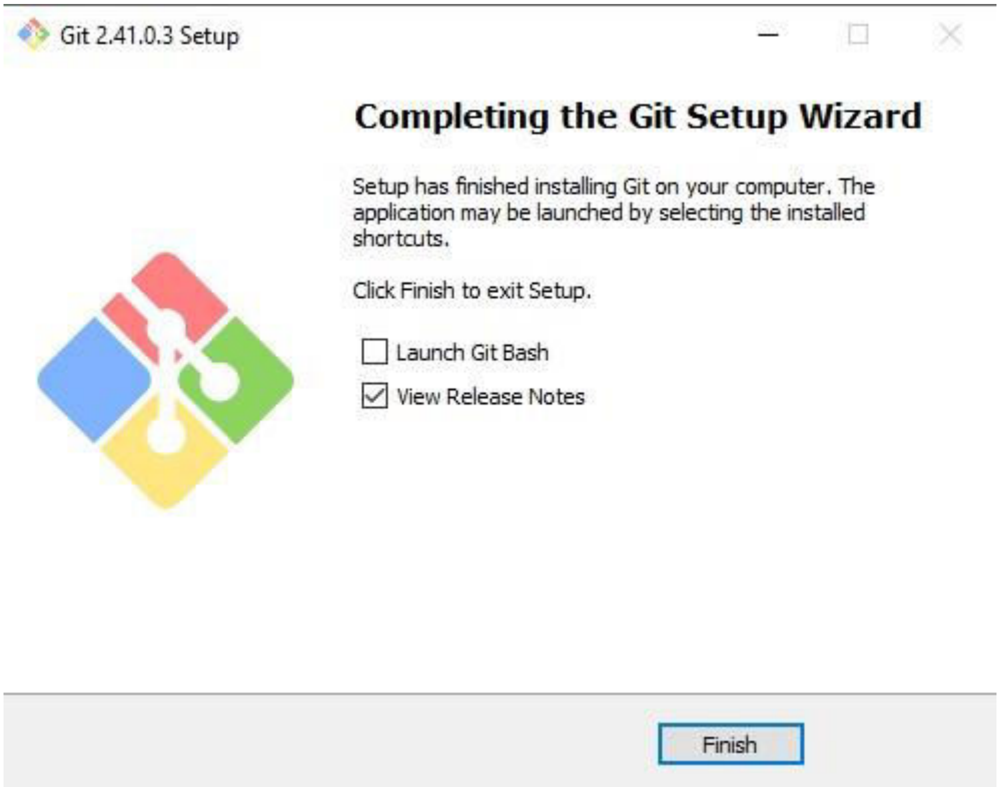
Please wait while Setup installs Git on your computer.

Extracting files...

C:\Program Files\Git\mingw64\lib\tcl8.6\tzdata\Africa\Johannesburg

https://gitforwindows.org/

Step-9: Setup Complete


Git 2.41.0.3 Setup

Completing the Git Setup Wizard

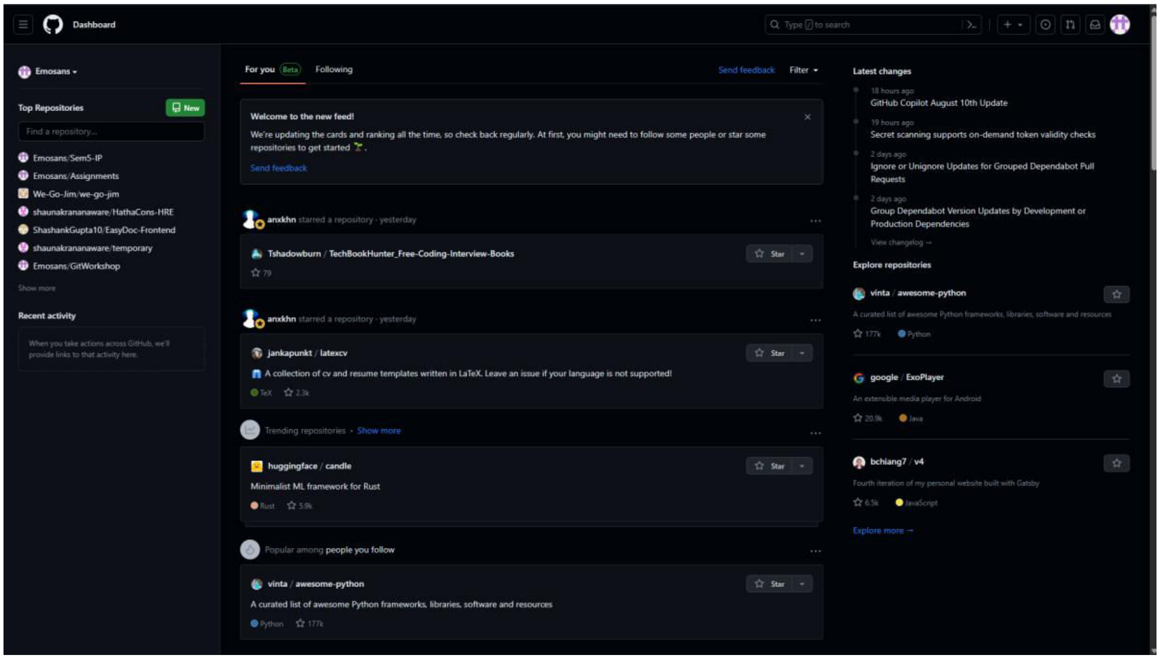
Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

☐ Launch Git Bash

☑ View Release Notes

3.GITHUB ACCOUNT



Top Repositories

- Emosans/Sem5-IP
- Emosans/Assignments
- We-Go-Jim/we-go-jim
- shaunakrananaware/HathaCons-HRE
- ShashankGupta10/EasyDoc-Frontend
- shaunakrananaware/temporary
- Emosans/GitWorkshop

Recent activity

When you take actions across GitHub, we'll provide links to that activity here.

Welcome to the new feed!

We're updating the cards and ranking all the time, so check back regularly. At first, you might need to follow some people or star some repositories to get started.

Send feedback

anxkhn starred a repository · yesterday

Tshadowburn / TechBookHunter_Free-Coding-Interview-Books

☆ 79

anxkhn starred a repository · yesterday

jankapunkt / latexcv

A collection of cv and resume templates written in LaTeX. Leave an issue if your language is not supported!

● TeX ☆ 2.3k

Trending repositories · Show more

huggingface / candle

Minimalist ML framework for Rust

● Rust ☆ 5.9k

Popular among people you follow

vinta / awesome-python

A curated list of awesome Python frameworks, libraries, software and resources

● Python ☆ 177k

Latest changes

18 hours ago
GitHub Copilot August 10th Update

19 hours ago
Secret scanning supports on-demand token validity checks

2 days ago
Ignore or Unignore Updates for Grouped Dependabot Pull Requests

2 days ago
Group Dependabot Version Updates by Development or Production Dependencies

Explore repositories

vinta / awesome-python

A curated list of awesome Python frameworks, libraries, software and resources

☆ 177k ● Python

google / ExoPlayer

An extensible media player for Android

☆ 20.9k ● Java

bchiang7 / v4

Fourth iteration of my personal website built with Gatsby

☆ 6.5k ● JavaScript

<u>Conclusion</u>: Basic knowledge regarding Git bash installation and Version Control System to effectively track changes augmented with git and github.