Assignment No 3

Name – Yuvraj Pardeshi

Roll No – 23546

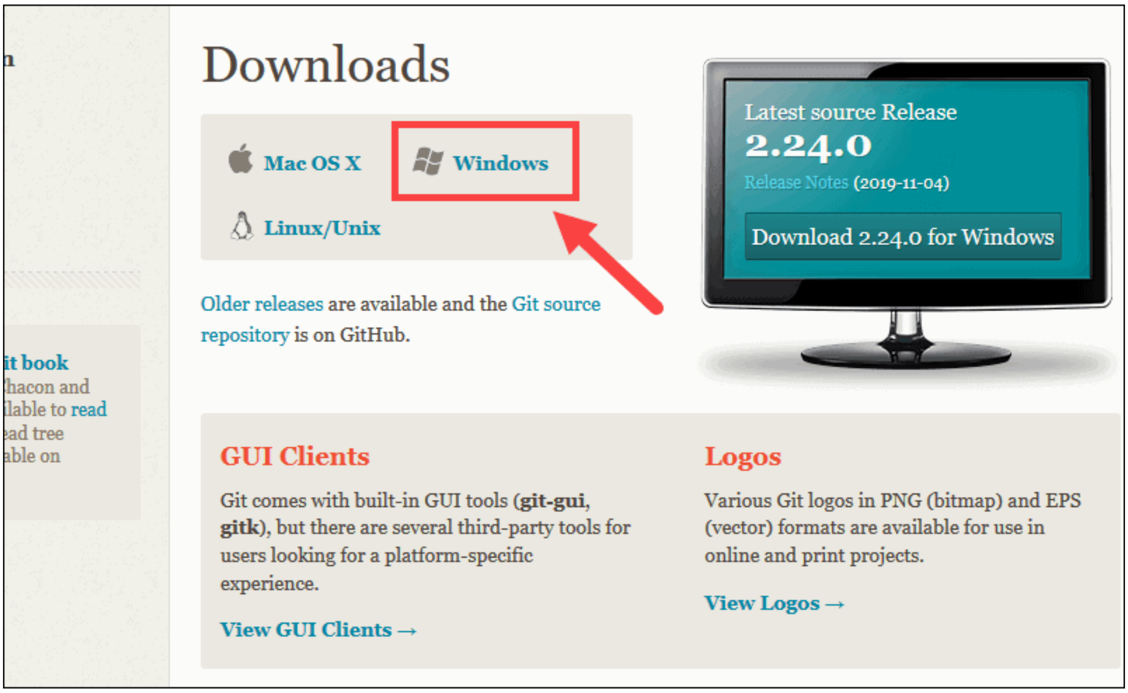
Sub – Web Technology

Topic - Create version control account on GitHub and using Git commands to create repository and push your code to GitHub.

Installation of git for windows -

Download Git for Windows

1. Browse to the official Git website: https://git-scm.com/downloads
2. Click the download link for Windows and allow the download to complete.



Older releases are available and the Git source repository is on GitHub.

GUI Clients

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

View GUI Clients →

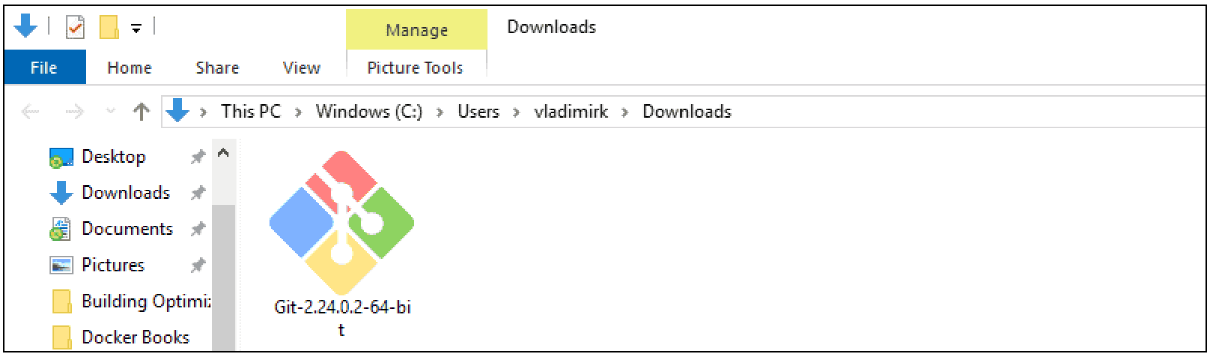
Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

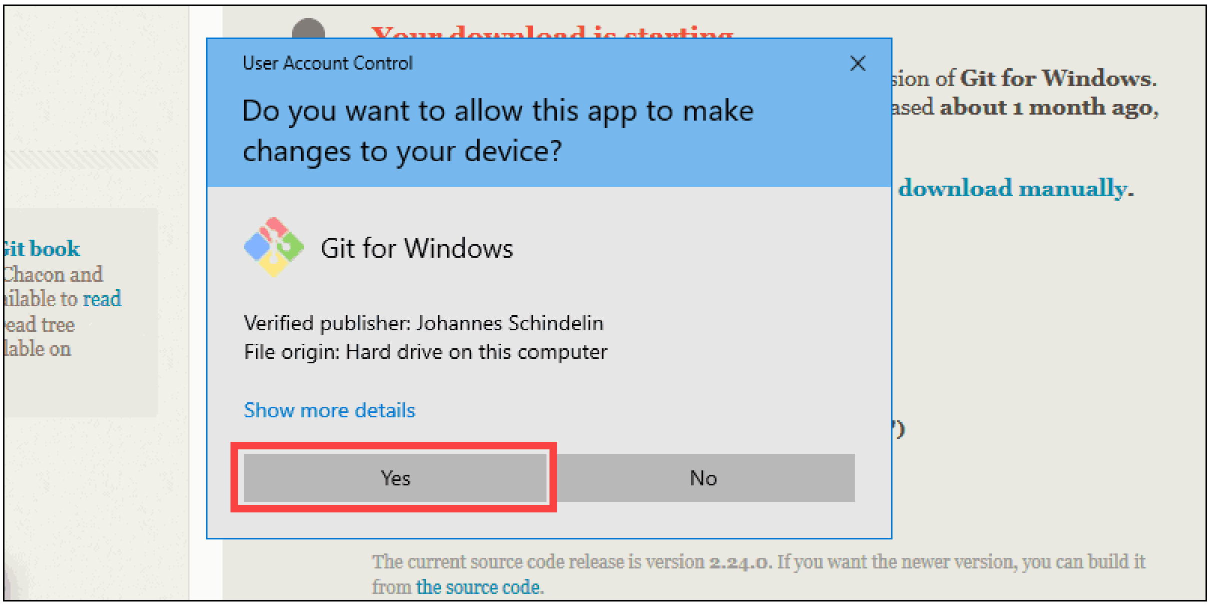
View Logos →

Extract and Launch Git Installer

3. Browse to the download location (or use the download shortcut in your browser). Double-click the file to extract and launch the installer.



4. Allow the app to make changes to your device by clicking **Yes** on the User Account Control dialog that opens.



User Account Control

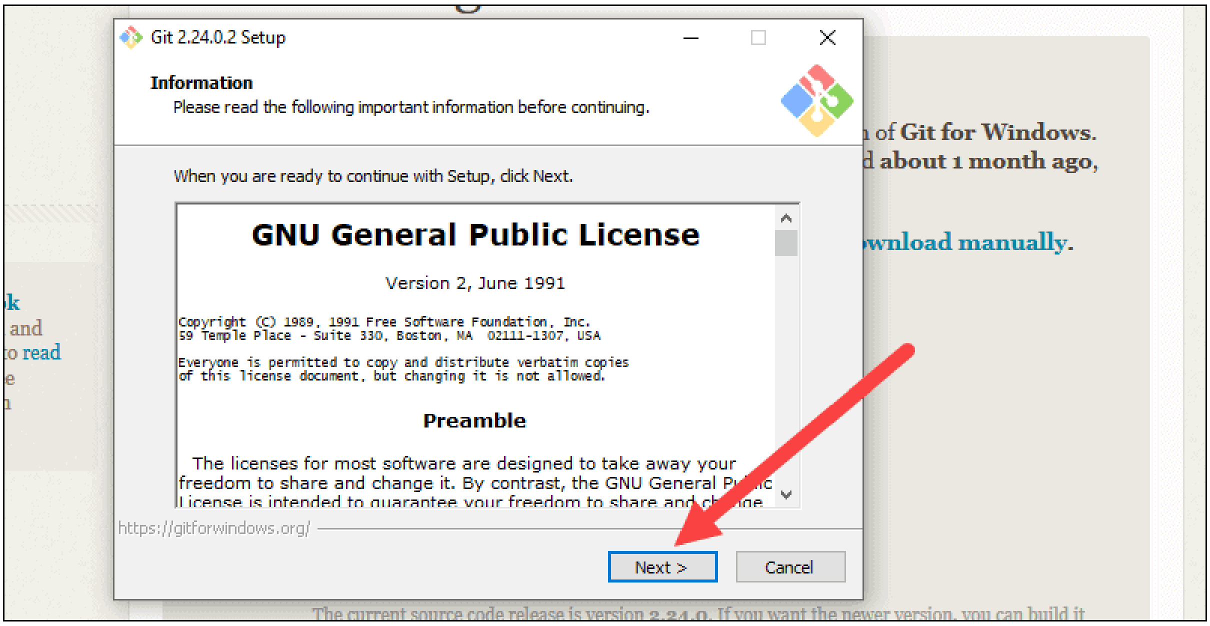
Do you want to allow this app to make changes to your device?

Git for Windows

Verified publisher: Johannes Schindelin
File origin: Hard drive on this computer

The current source code release is version 2.24.0. If you want the newer version, you can build it from the source code.

5. Review the GNU General Public License, and when you're ready to install, click **Next**.



Git 2.24.0.2 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

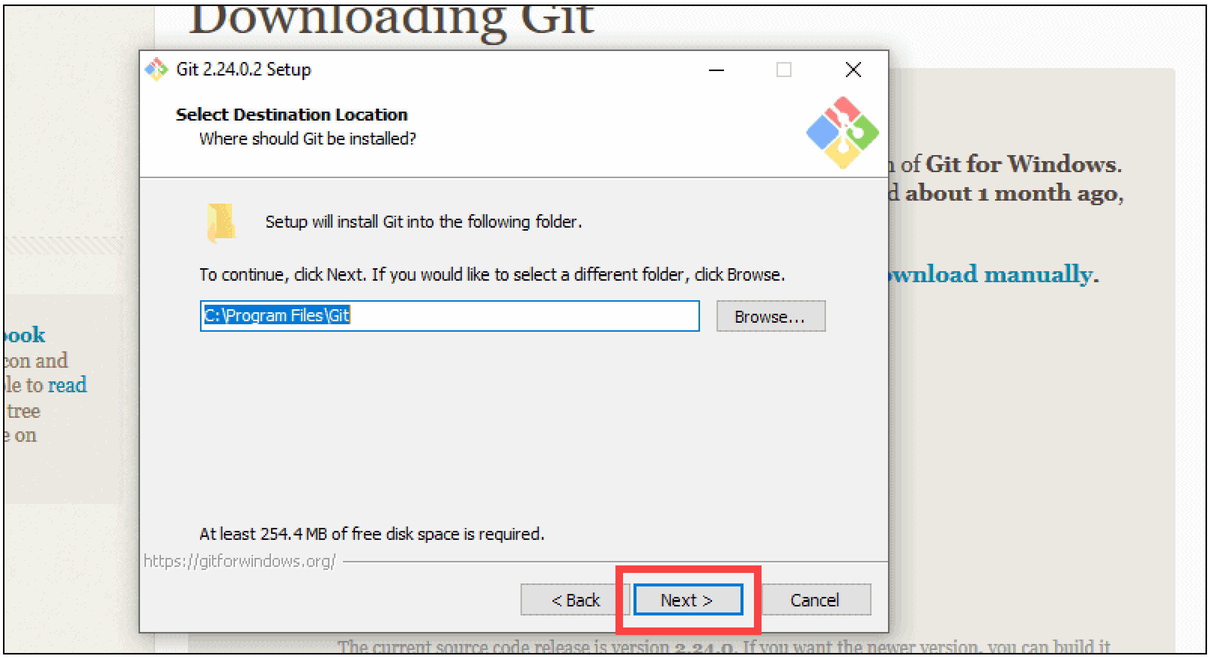
```
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA  02111-1307, USA

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

https://gitforwindows.org/
Next >
Cancel
n of Git for Windows.
d about 1 month ago,
ownload manually.
The current source code release is version 2.24.0. If you want the newer version, you can build it

6. The installer will ask you for an installation location. Leave the default, unless you have reason to change it, and click **Next**.



Git 2.24.0.2 Setup

Select Destination Location

Where should Git be installed?

Setup will install Git into the following folder.

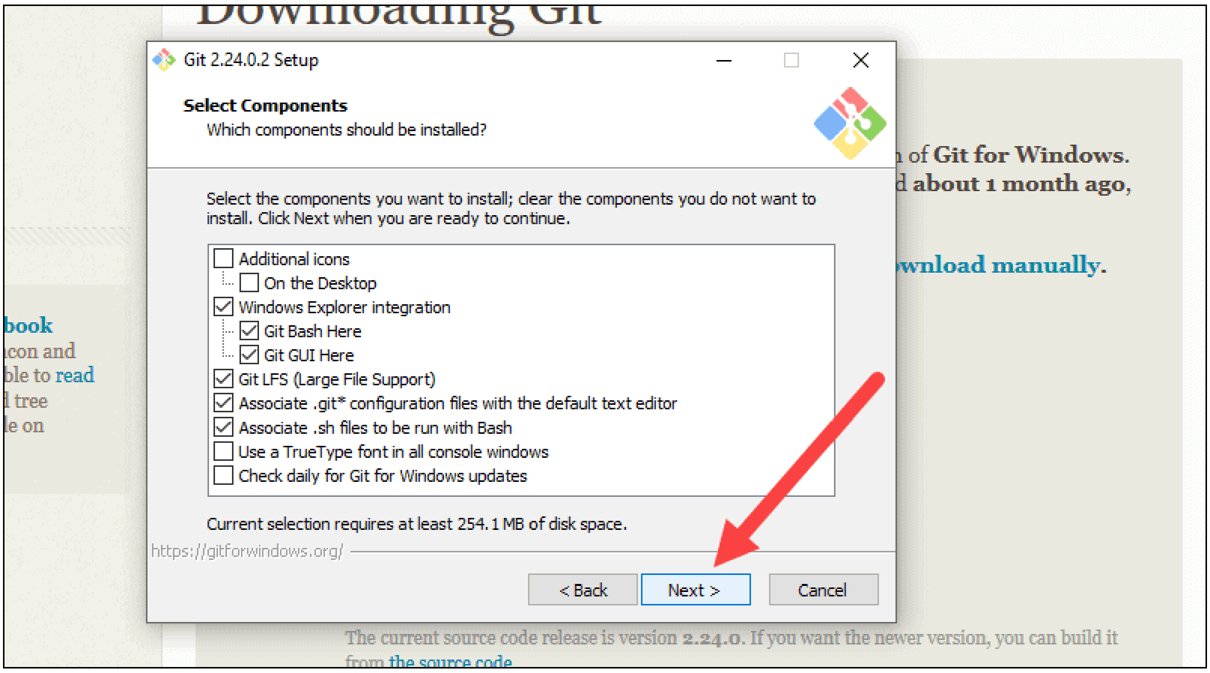
To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files\Git

At least 254.4 MB of free disk space is required.

Downloading Git
Browse...
https://gitforwindows.org/
< Back
Next >
Cancel
n of Git for Windows.
d about 1 month ago,
ownload manually.
The current source code release is version 2.24.0. If you want the newer version, you can build it

7. A component selection screen will appear. Leave the defaults unless you have a specific need to change them and click **Next**.



Git 2.24.0.2 Setup

Select Components

Which components should be installed?

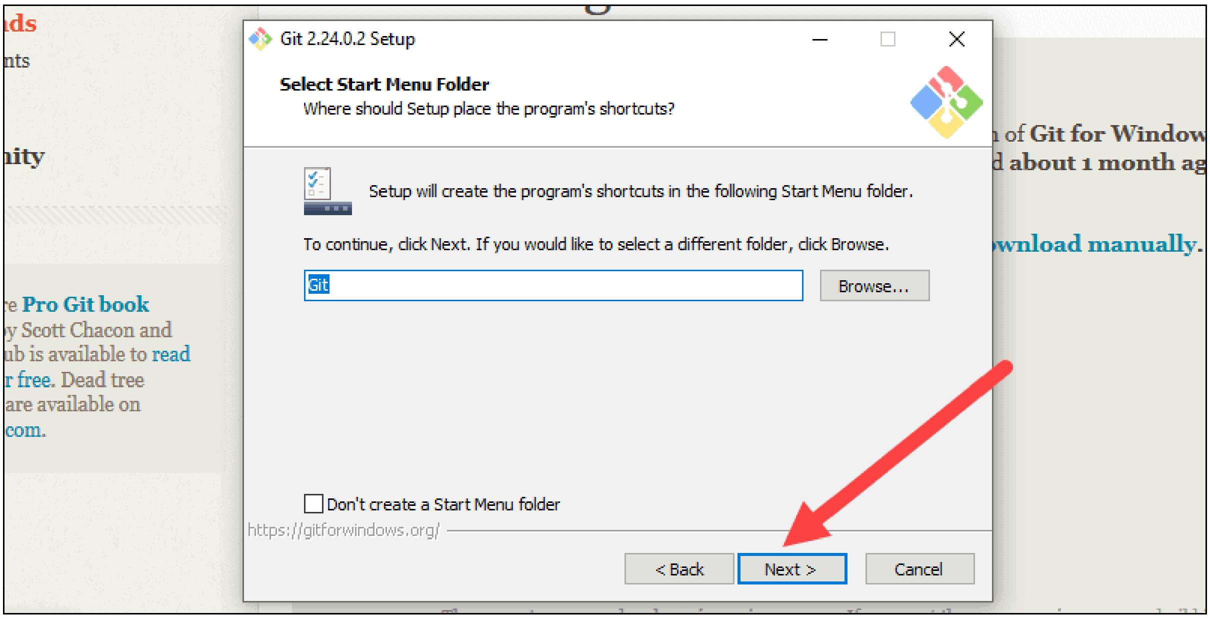
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- □ Additional icons
  - □ On the Desktop
- ☑ Windows Explorer integration
  - ☑ Git Bash Here
  - ☑ 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
- □ Use a TrueType font in all console windows
- □ Check daily for Git for Windows updates

Current selection requires at least 254.1 MB of disk space.

https://gitforwindows.org/

8. The installer will offer to create a start menu folder. Simply click **Next**.



Git 2.24.0.2 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.

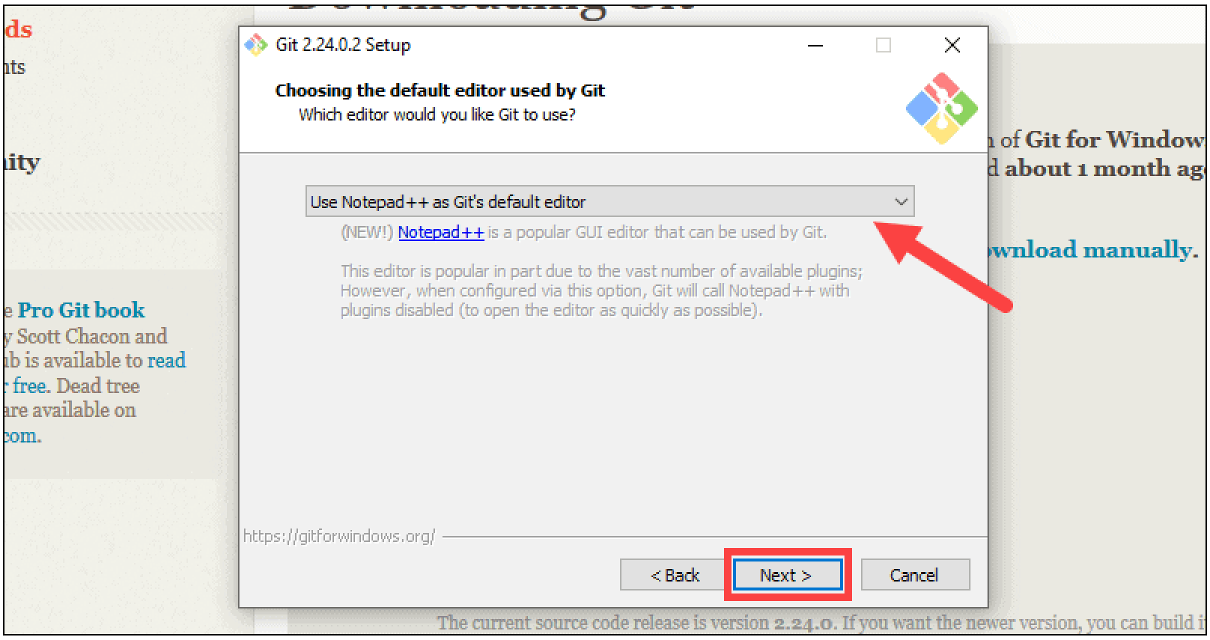
To continue, click Next. If you would like to select a different folder, click Browse.

Git

□ Don't create a Start Menu folder

https://gitforwindows.org/

9. Select a text editor you'd like to use with Git. Use the drop-down menu to select Notepad++ (or whichever text editor you prefer) and click **Next**.



Git 2.24.0.2 Setup

Choosing the default editor used by Git

Which editor would you like Git to use?

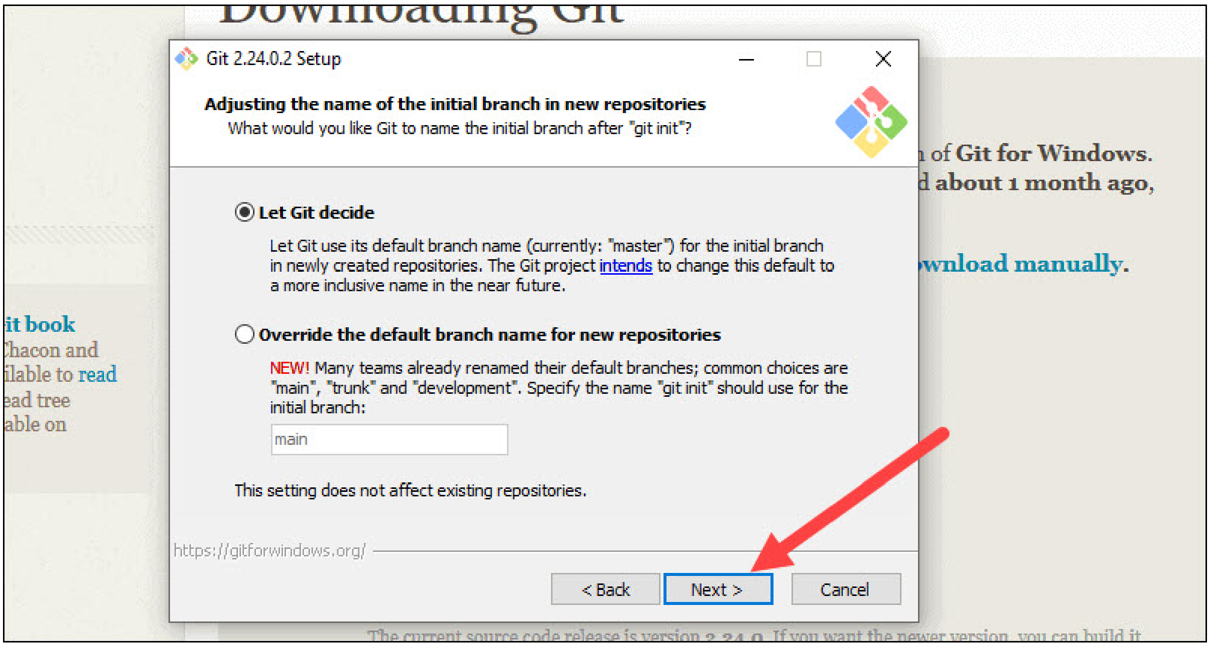
Use Notepad++ as Git's default editor

(NEW!) Notepad++ is a popular GUI editor that can be used by Git.

This editor is popular in part due to the vast number of available plugins; However, when configured via this option, Git will call Notepad++ with plugins disabled (to open the editor as quickly as possible).

https://gitforwindows.org/

10. The next step allows you to choose a different name for your initial branch. The default is 'master.' Unless you're working in a team that requires a different name, leave the default option and click **Next.**



Git 2.24.0.2 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**

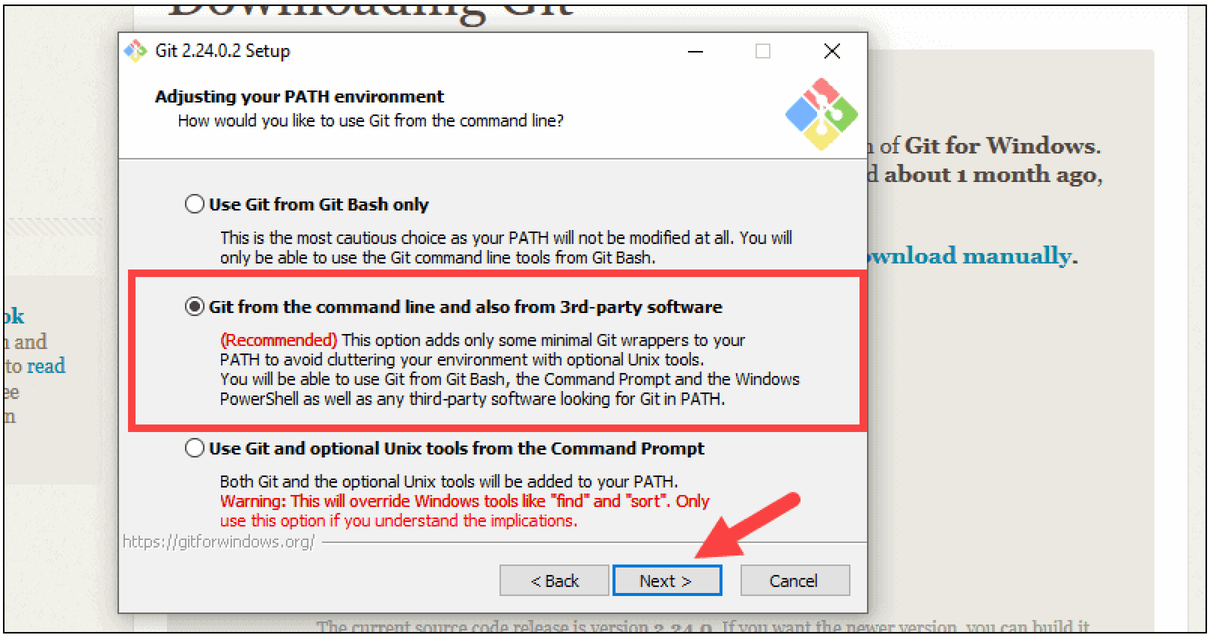
NEW! Many teams already renamed their default branches; common choices are "main", "trunk" and "development". Specify the name "git init" should use for the initial branch:

main

This setting does not affect existing repositories.

https://gitforwindows.org/

11. This installation step allows you to change the **PATH environment**. The **PATH** is the default set of directories included when you run a command from the command line. Leave this on the middle (recommended) selection and click **Next**.



Git 2.24.0.2 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.

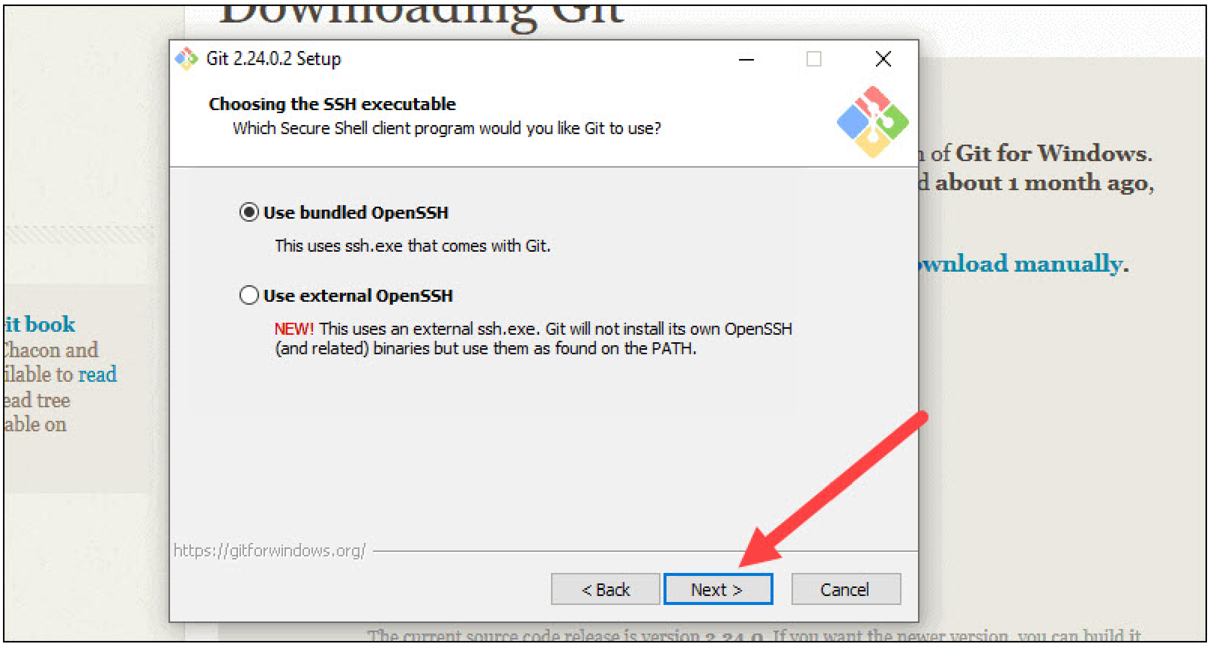
○ **Use Git and optional Unix tools from the Command Prompt**

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.

https://gitforwindows.org/

Server Certificates, Line Endings and Terminal Emulators

12. The installer now asks which SSH client you want Git to use. Git already comes with its own SSH client, so if you don't need a specific one, leave the default option and click **Next.**



Git 2.24.0.2 Setup

Choosing the SSH executable

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

● **Use bundled OpenSSH**

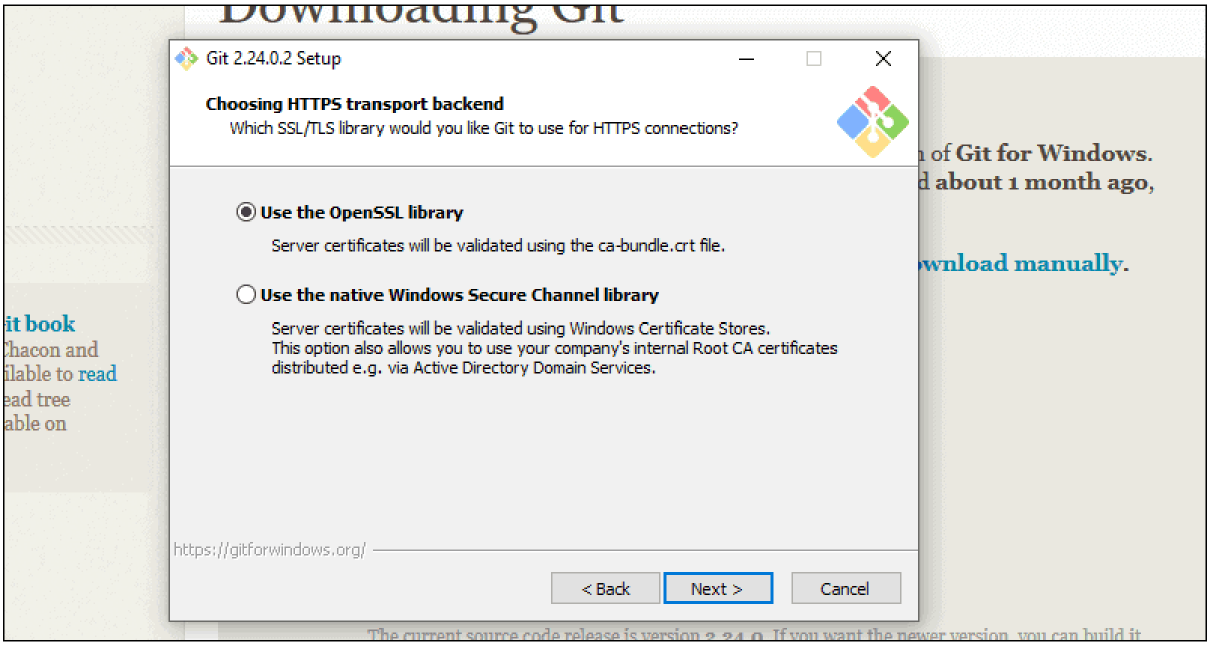
This uses ssh.exe that comes with Git.

○ **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.

https://gitforwindows.org/

13. The next option relates to server certificates. Most users should use the default. If you're working in an Active Directory environment, you may need to switch to Windows Store certificates. Click **Next**.



Git 2.24.0.2 Setup

Choosing HTTPS transport backend

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

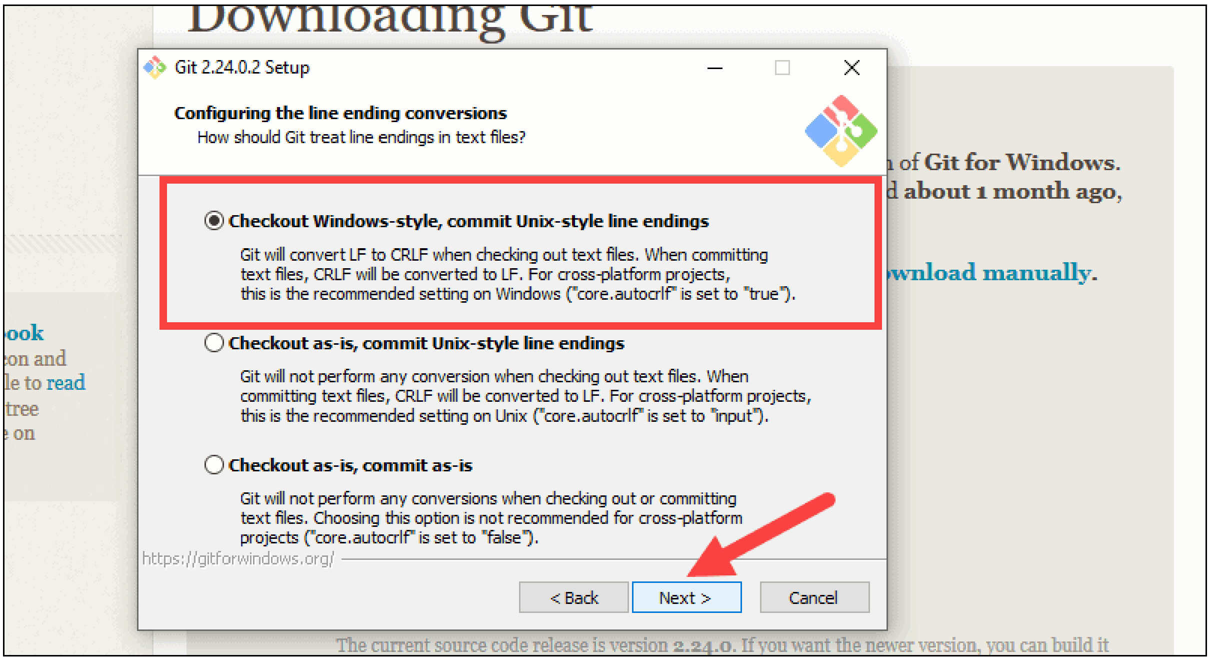
● **Use the OpenSSL library**

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.

14. The next selection converts line endings. It is recommended that you leave the default selection. This relates to the way data is formatted and changing this option may cause problems. Click **Next**.



Git 2.24.0.2 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**

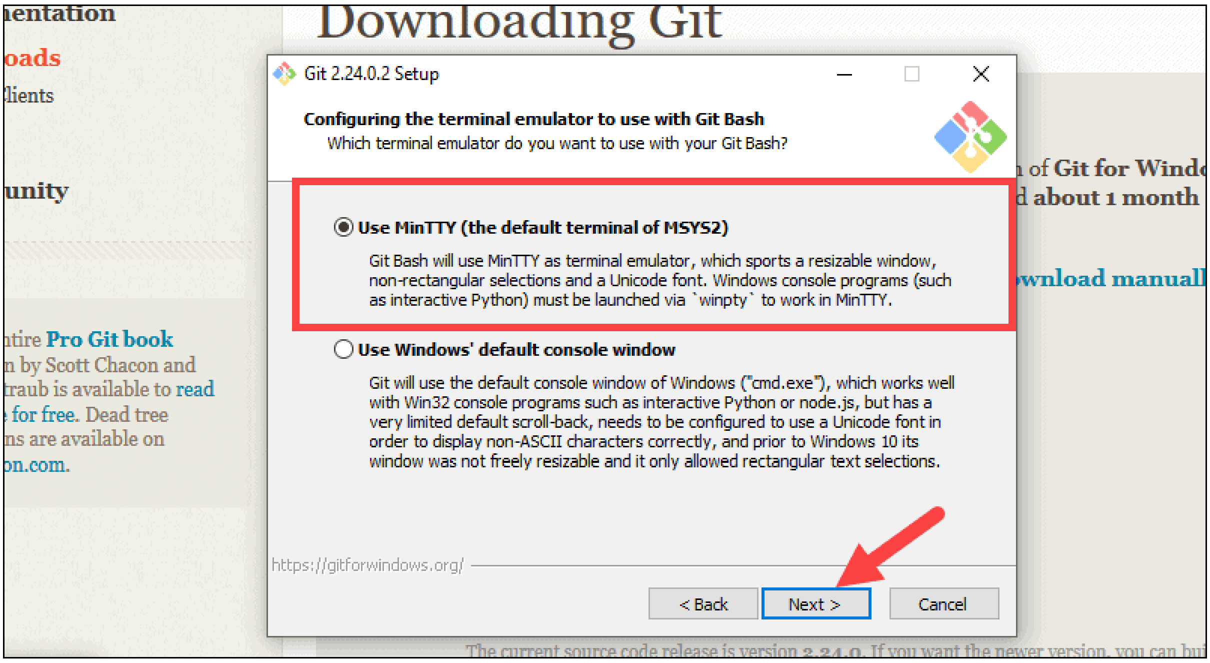
Git will not perform any conversion when checking out text files. When
committing text files, CRLF will be converted to LF. For cross-platform projects,
this is the recommended setting on Unix ("core.autocrlf" is set to "input").

○ **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").

Downloading Git
of Git for Windows.
about 1 month ago,
wnload manually.
ook
on and
le to read
tree
on
https://gitforwindows.org/
< Back
Next >
Cancel
The current source code release is version 2.24.0. If you want the newer version, you can build it

15. Choose the terminal emulator you want to use. The default MinTTY is recommended, for its features. Click **Next**.



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

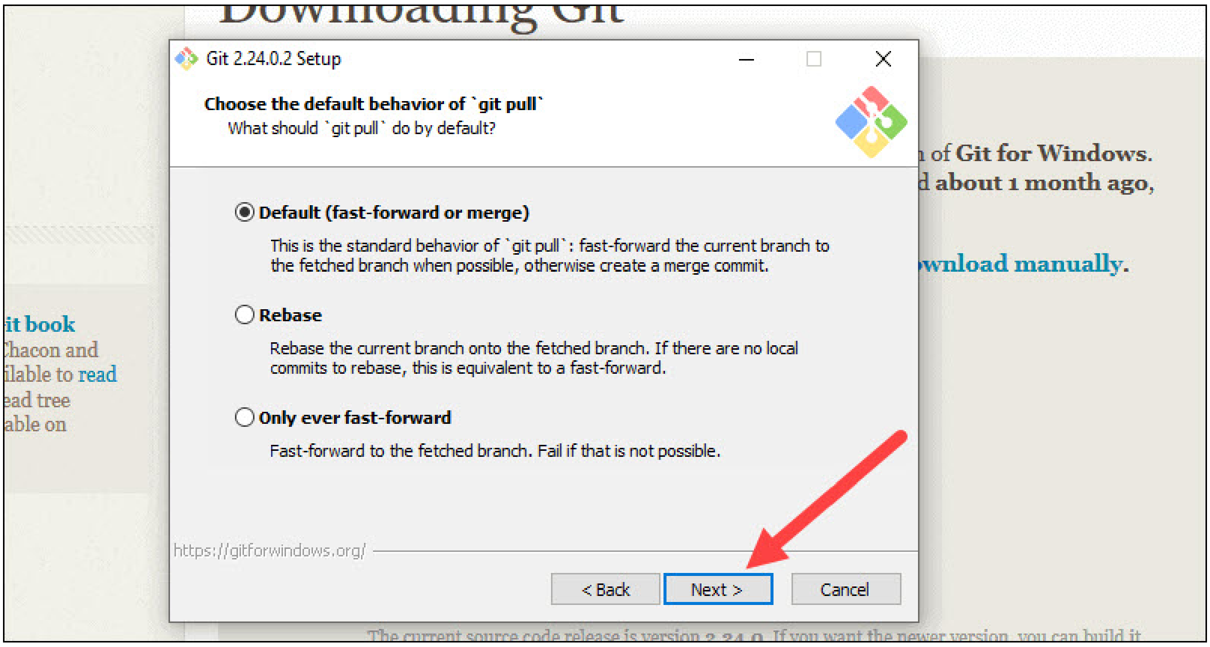
● **Use MinTTY (the default terminal of MSYS2)**

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.

16. The installer now asks what the `git pull` command should do. The default option is recommended unless you specifically need to change its behavior. Click **Next** to continue with the installation.



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.

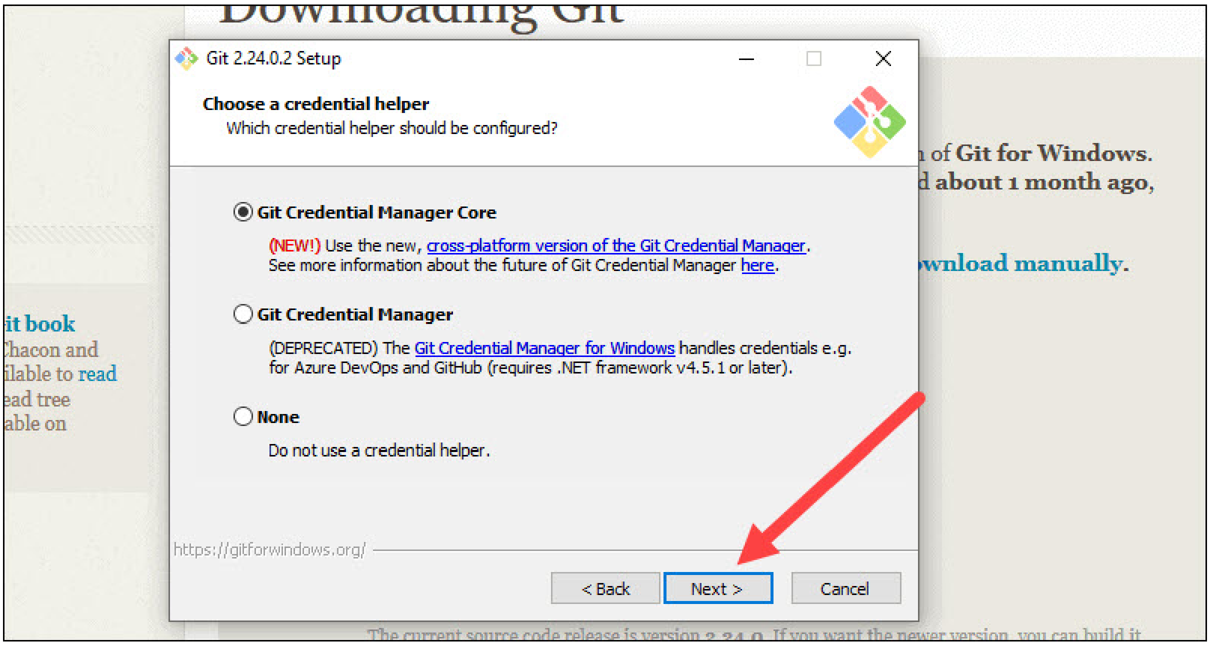
○ **Rebase**

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.

17. Next you should choose which credential helper to use. Git uses credential helpers to fetch or save credentials. Leave the default option as it is the most stable one, and click **Next**.



Git 2.24.0.2 Setup

Choose a credential helper

Which credential helper should be configured?

● **Git Credential Manager Core**

(NEW!) Use the new, cross-platform version of the Git Credential Manager. See more information about the future of Git Credential Manager here.

○ **Git Credential Manager**

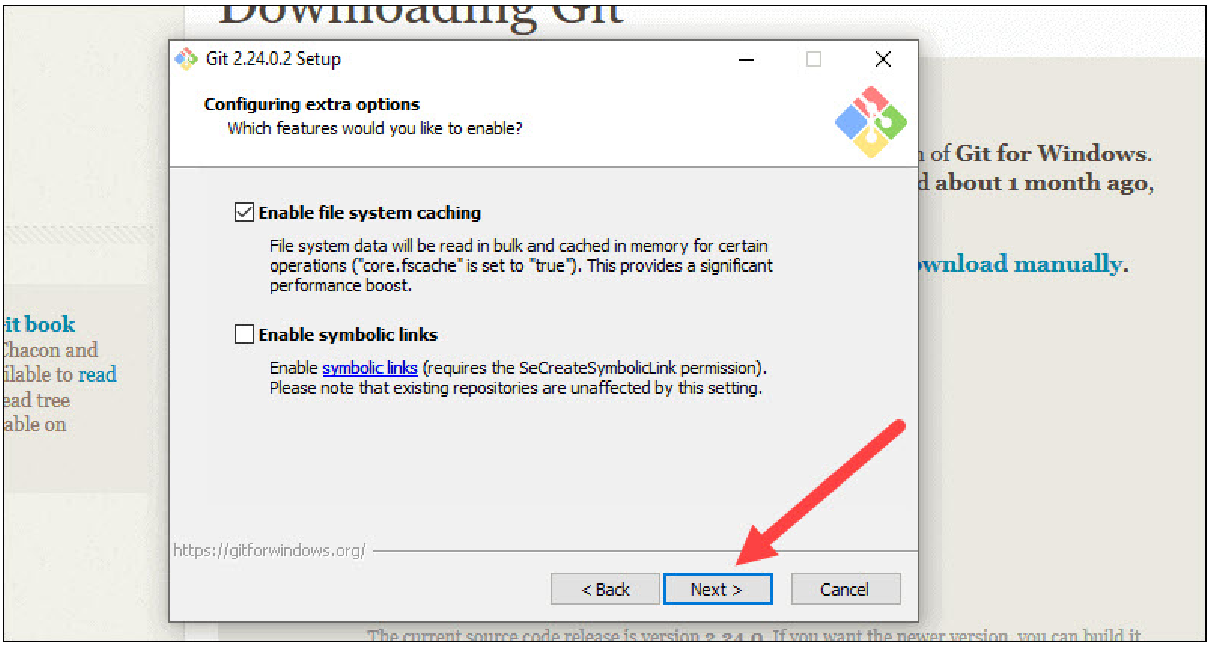
(DEPRECATED) The Git Credential Manager for Windows handles credentials e.g. for Azure DevOps and GitHub (requires .NET framework v4.5.1 or later).

○ **None**

Do not use a credential helper.

Additional Customization Options

18. The default options are recommended, however this step allows you to decide which extra option you would like to enable. If you use symbolic links, which are like shortcuts for the command line, tick the box. Click **Next**.



Git 2.24.0.2 Setup

Configuring extra options

Which features would you like to enable?

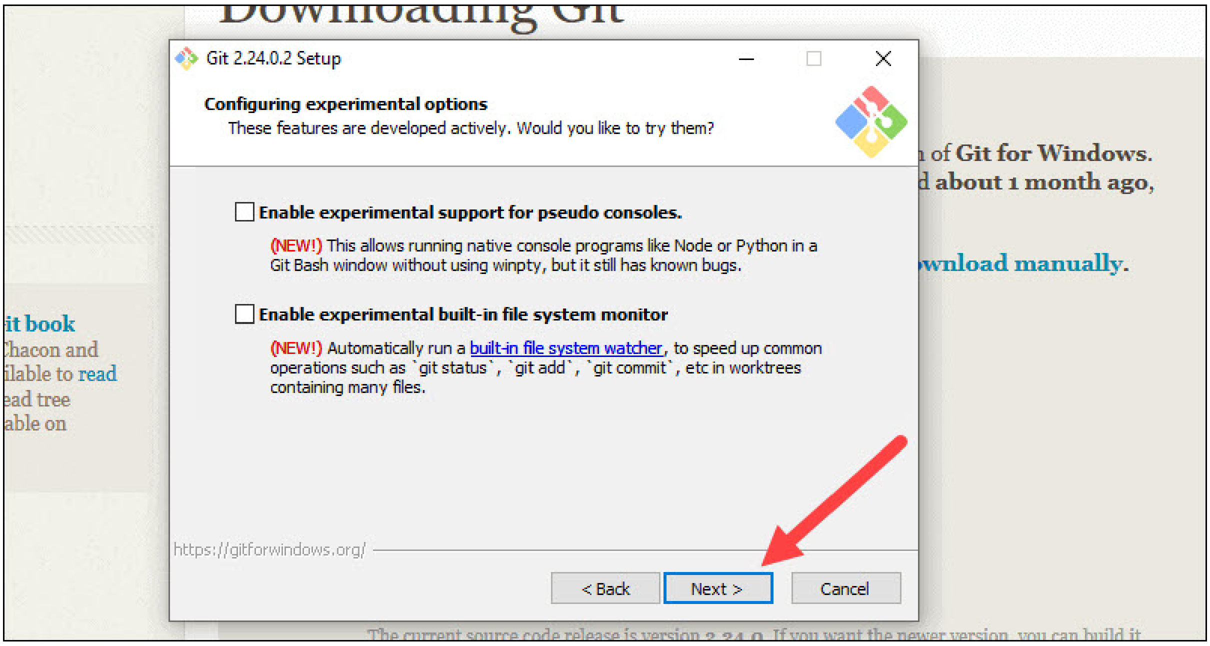
☑ **Enable file system caching**

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.

19. Depending on the version of Git you're installing, it may offer to install experimental features. At the time this article was written, the options to include support for pseudo controls and a built-in file system monitor were offered. Unless you are feeling adventurous, leave them unchecked and click **Install**.



Configuring experimental options

These features are developed actively. Would you like to try them?

□ **Enable experimental support for pseudo consoles.**

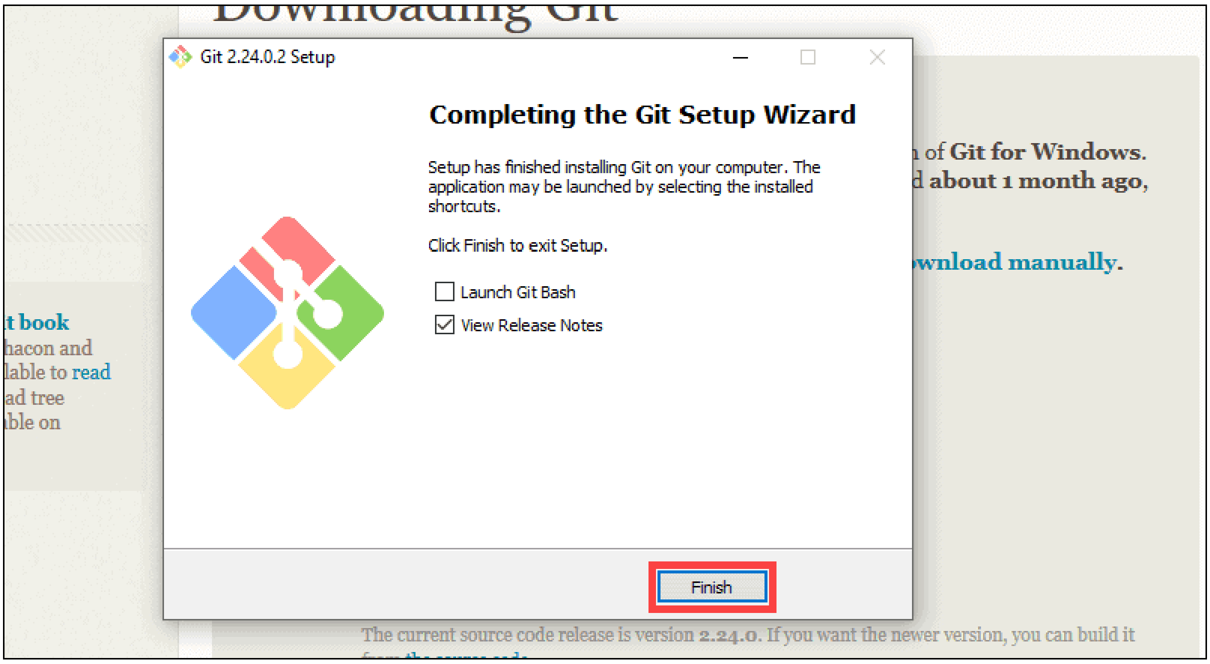
(NEW!) This allows running native console programs like Node or Python in a Git Bash window without using winpty, but it still has known bugs.

□ **Enable experimental built-in file system monitor**

(NEW!) Automatically run a built-in file system watcher, to speed up common operations such as `git status`, `git add`, `git commit`, etc in worktrees containing many files.

Complete Git Installation Process

20. Once the installation is complete, tick the boxes to view the Release Notes or Launch Git Bash, then click **Finish**.



Completing the Git Setup Wizard

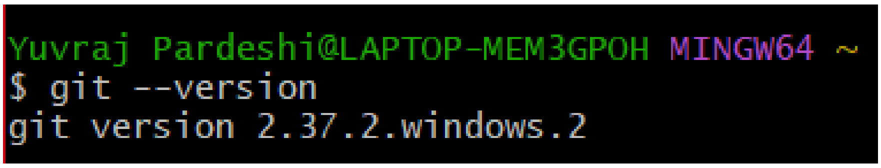
Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

□ Launch Git Bash

☑ View Release Notes

21. Know open the git bash and check if it is installed properly or not using git version command.



```
Yuvraj Pardeshi@LAPTOP-MEM3GPOH MINGW64 ~
$ git --version
git version 2.37.2.windows.2
```

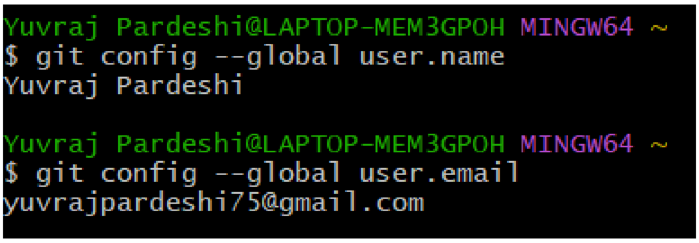
If the output comes like this then your git is installed successfully.

Create Repository using git commands –

Git Set up commands –

1. The first thing you should do when you install Git is to set your user name and email address. This is important because every Git commit uses this information, and it’s immutably baked into the commits you start creating:

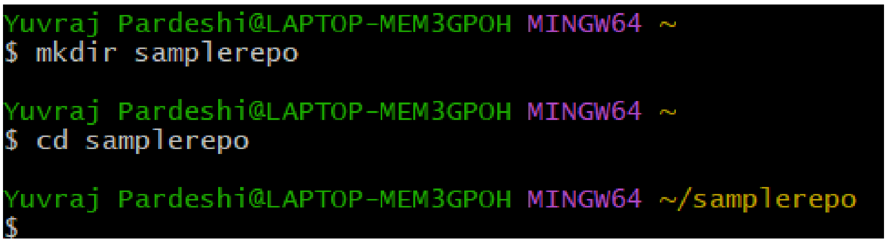
```
$ git config --global user.name "Yuvraj Pardeshi"

$ git config --global user.email yuvrajpardeshi75@gmail.com
```



```
Yuvraj Pardeshi@LAPTOP-MEM3GPOH MINGW64 ~
$ git config --global user.name
Yuvraj Pardeshi

Yuvraj Pardeshi@LAPTOP-MEM3GPOH MINGW64 ~
$ git config --global user.email
yuvrajpardeshi75@gmail.com
```

Create a new project –

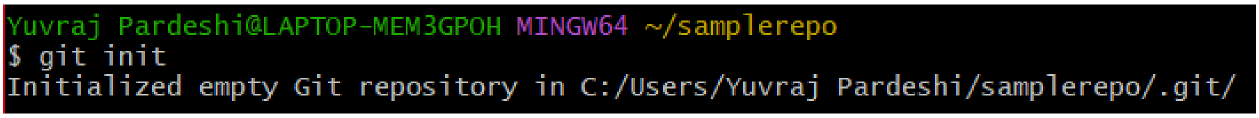
2. Create an folder using command mkdir <dir name>
and to enter into this create folder use command cd <dir name>



```
Yuvraj Pardeshi@LAPTOP-MEM3GPOH MINGW64 ~
$ mkdir samplerepo

Yuvraj Pardeshi@LAPTOP-MEM3GPOH MINGW64 ~
$ cd samplerepo

Yuvraj Pardeshi@LAPTOP-MEM3GPOH MINGW64 ~/samplerepo
$
```

Create Repository –

3. To create your current folder into repository use command git init



```
Yuvraj Pardeshi@LAPTOP-MEM3GPOH MINGW64 ~/samplerepo
$ git init
Initialized empty Git repository in C:/Users/Yuvraj Pardeshi/samplerepo/.git/
```

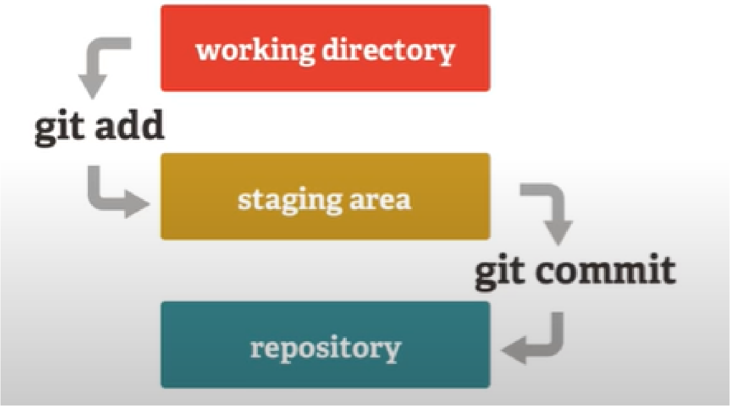
as shown in above figure your empty repo is created on the path shown in the figure.

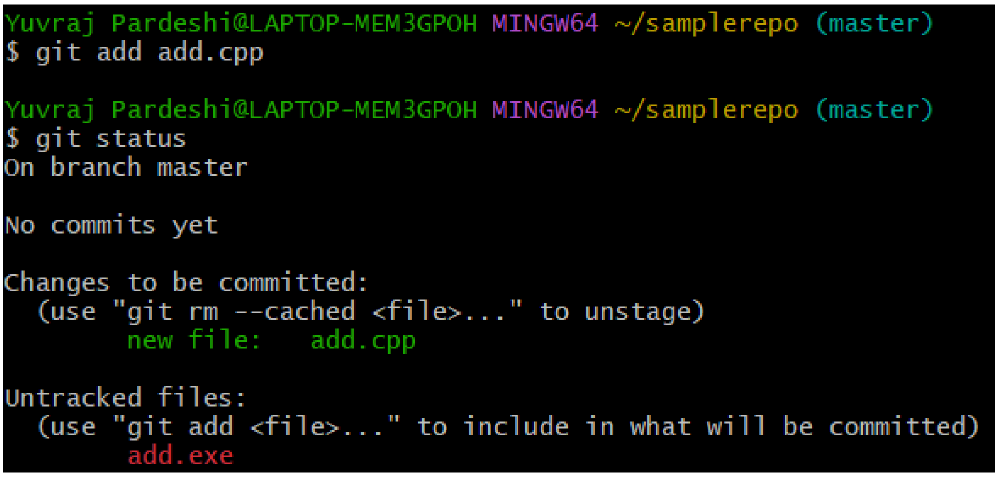
Adding files to Repository -

4. Know here comes the concept of staging area and committing a file.

   Staging area – Initially when you add files to the git folder they are not tracked by git, these files are also referred to as "untracked files". Staging area is files that are going to be a part of next commit, which lets the git know what changes in the file are going to occur for the next commit. The command use to add the file to staging area is (git add).

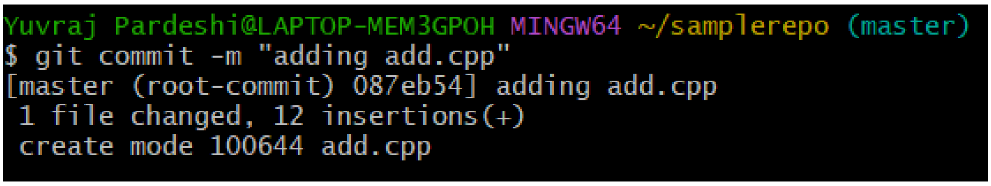
   Committing file - The git commit command captures a snapshot of the project's currently staged changes. Committed snapshots can be thought of as "safe" versions of a project—Git will never change them unless you explicitly ask it to. Prior to the execution of git commit, The git add command is used to promote or 'stage' changes to the project that will be stored in a commit. These two commands "git commit" and "git add" are two of the most frequently used.





```
Yuvraj Pardeshi@LAPTOP-MEM3GPOH MINGW64 ~/samplerepo (master)
$ git add add.cpp

Yuvraj Pardeshi@LAPTOP-MEM3GPOH MINGW64 ~/samplerepo (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   add.cpp

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

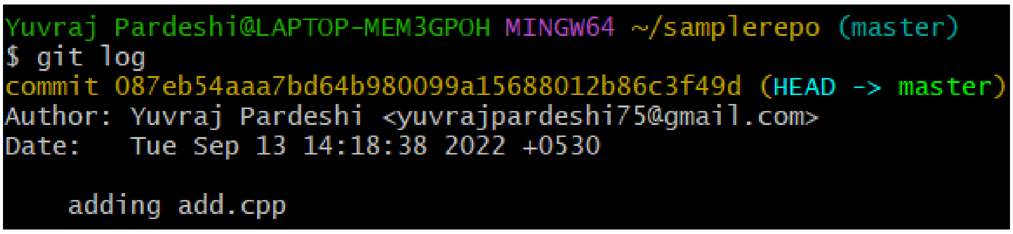
(Added file to staging area)



```
Yuvraj Pardeshi@LAPTOP-MEM3GPOH MINGW64 ~/samplerepo (master)
$ git commit -m "adding add.cpp"
[master (root-commit) 087eb54] adding add.cpp
 1 file changed, 12 insertions(+)
 create mode 100644 add.cpp
```

(Committed the file)

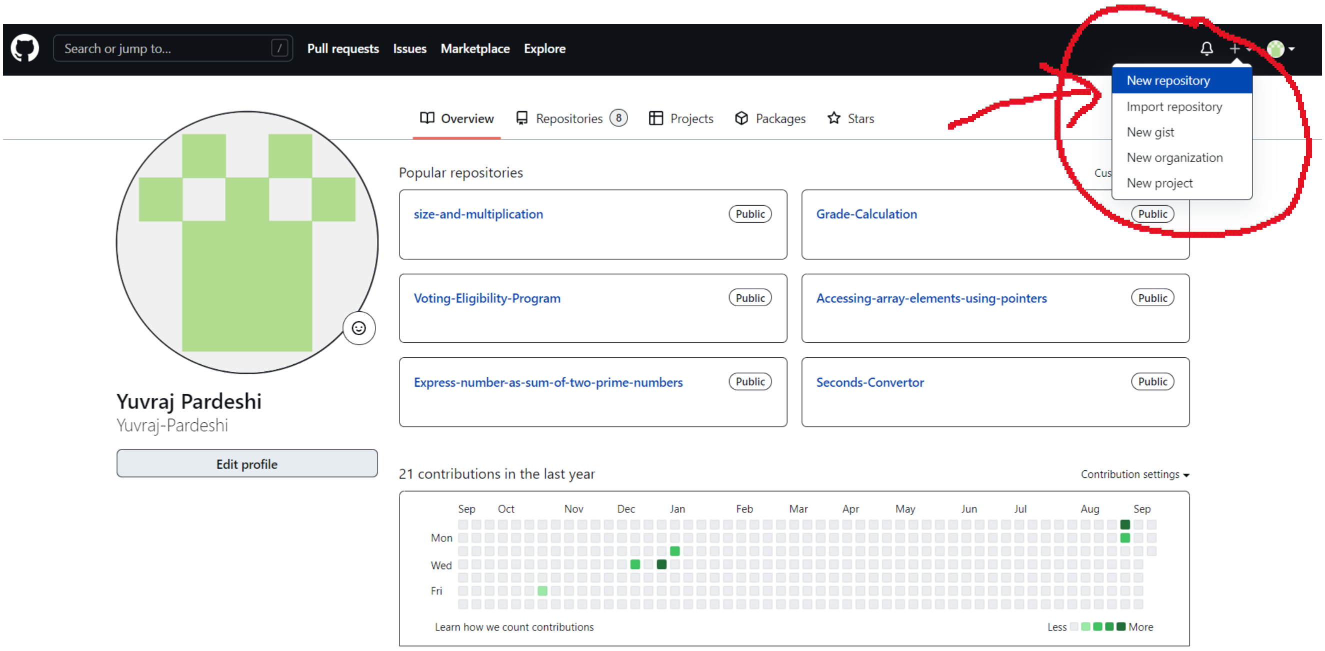
To check the commit status use git log command –

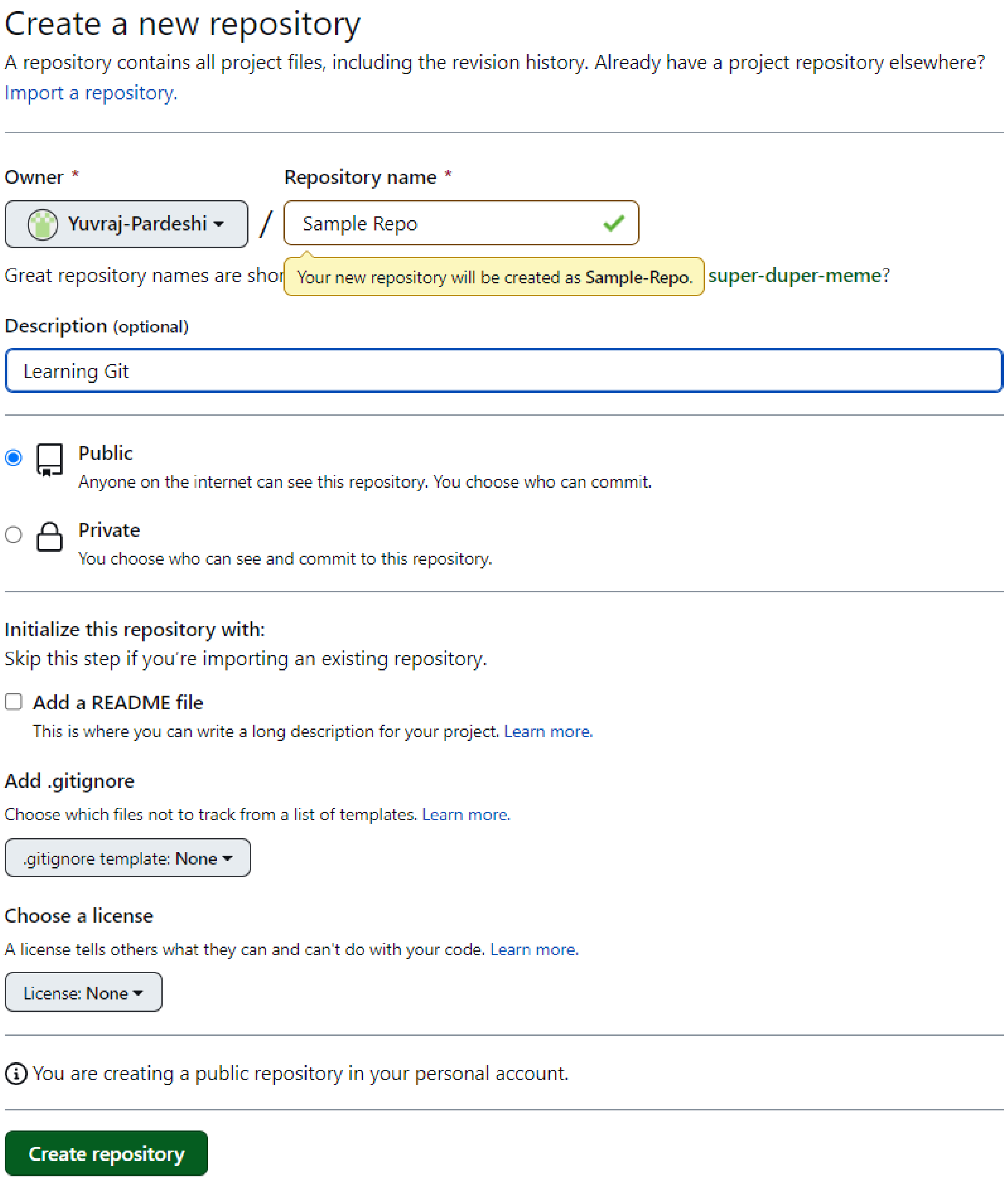


```
Yuvraj Pardeshi@LAPTOP-MEM3GPOH MINGW64 ~/samplerepo (master)
$ git log
commit 087eb54aaa7bd64b980099a15688012b86c3f49d (HEAD -> master)
Author: Yuvraj Pardeshi <yuvrajpardeshi75@gmail.com>
Date:   Tue Sep 13 14:18:38 2022 +0530

    adding add.cpp
```

Pushing your files to GitHub –

5 . Login your github account and create a new repository.





Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner *: Yuvraj-Pardeshi / Repository name *: Sample Repo ✓

Great repository names are shor Your new repository will be created as **Sample-Repo.** super-duper-meme?

Description (optional): Learning Git

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

□ Add a README file
This is where you can write a long description for your project. Learn more.

Add .gitignore
Choose which files not to track from a list of templates. Learn more.

.gitignore template: None

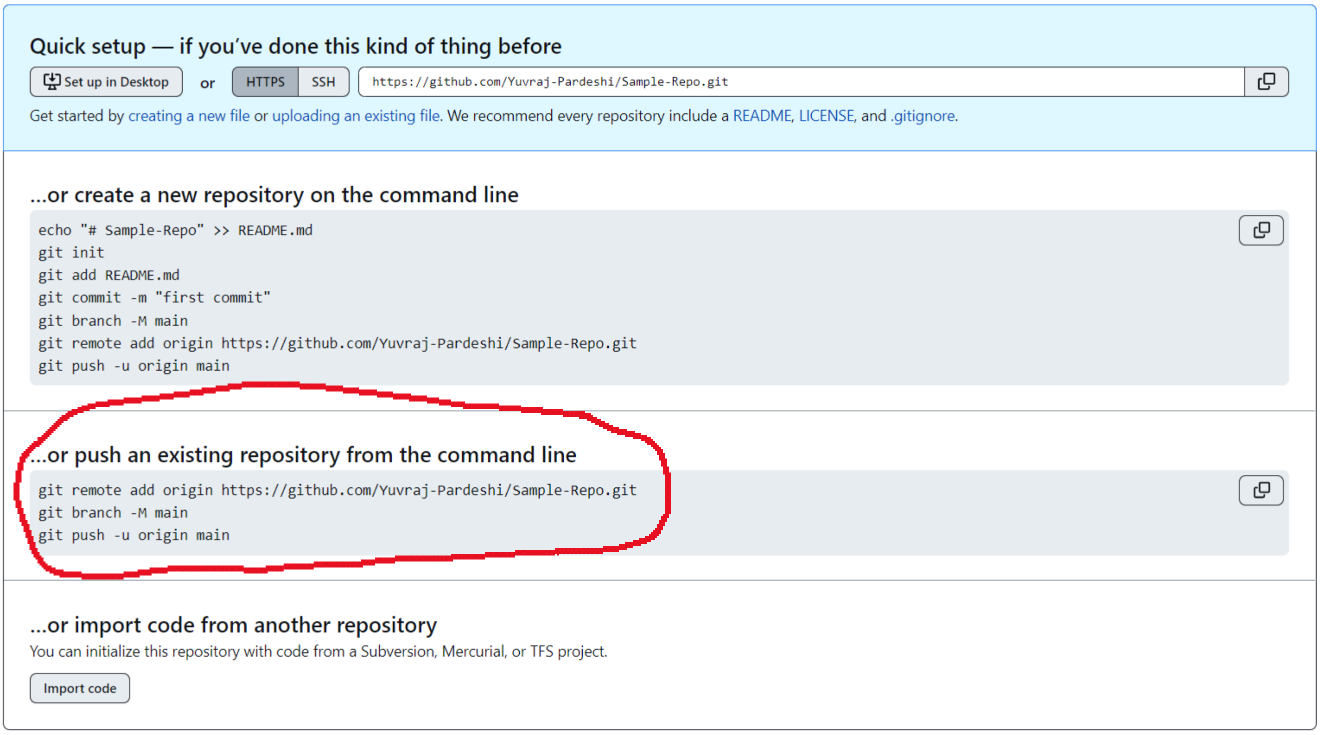
Choose a license
A license tells others what they can and can't do with your code. Learn more.

License: None

ⓘ You are creating a public repository in your personal account.

After creating the repository in github, there will the option to create your repo from scratch or if there is some local repo already created then push it. We are going to push our local repo created using git.

Pushing the local repo –



Quick setup — if you've done this kind of thing before

```
https://github.com/Yuvraj-Pardeshi/Sample-Repo.git
```

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# Sample-Repo" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Yuvraj-Pardeshi/Sample-Repo.git
git push -u origin main
```

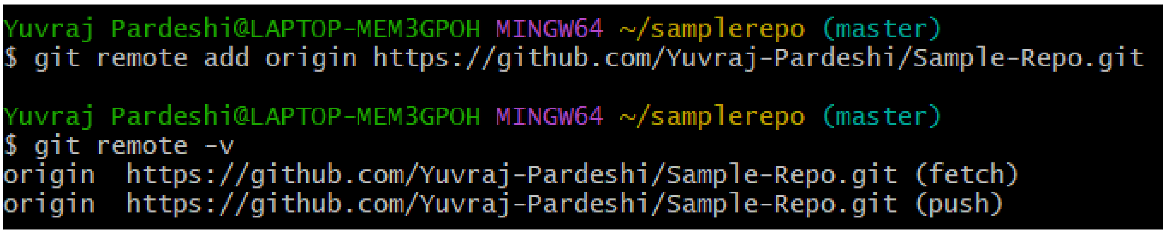
...or push an existing repository from the command line

```
git remote add origin https://github.com/Yuvraj-Pardeshi/Sample-Repo.git
git branch -M main
git push -u origin main
```

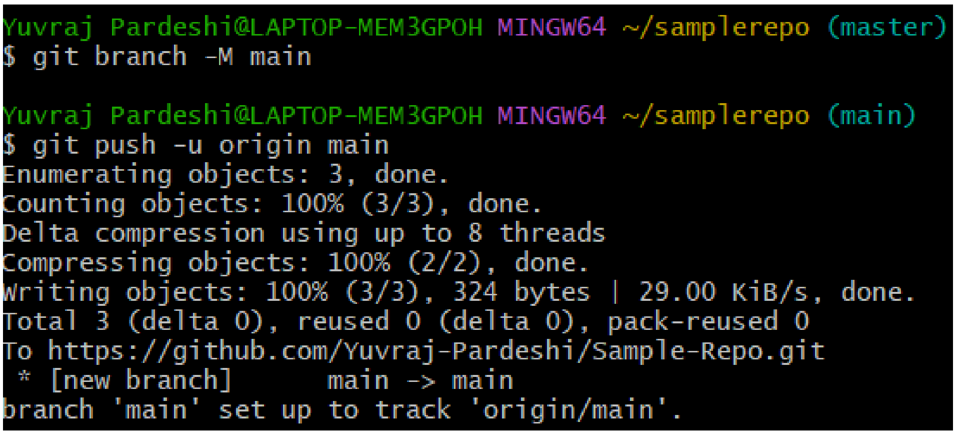
...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

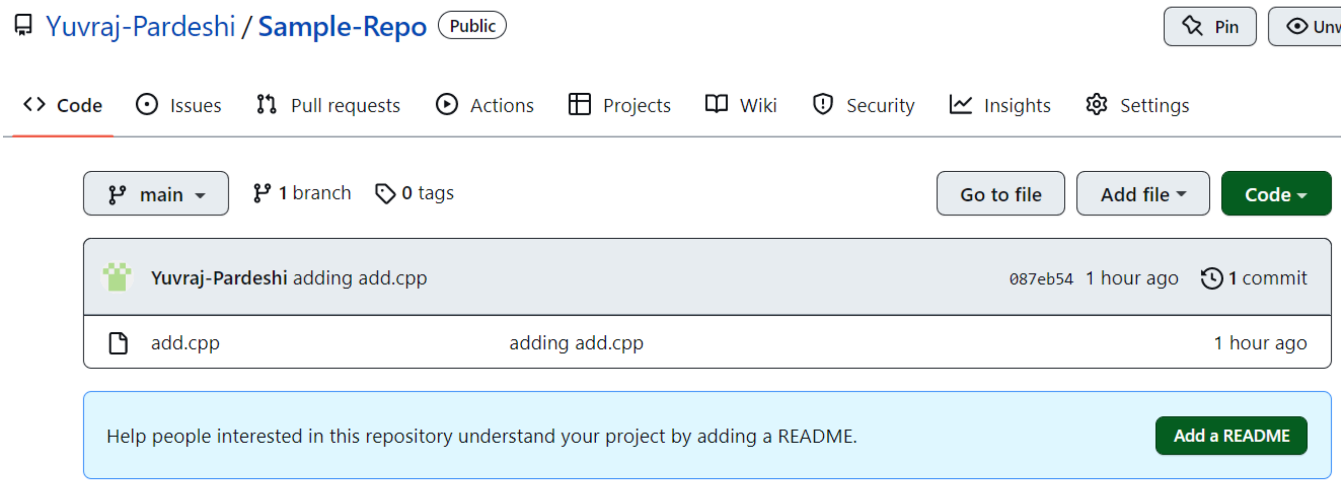
(Use the above commands to push your local repo)



```
Yuvraj Pardeshi@LAPTOP-MEM3GPOH MINGW64 ~/samplerepo (master)
$ git remote add origin https://github.com/Yuvraj-Pardeshi/Sample-Repo.git

Yuvraj Pardeshi@LAPTOP-MEM3GPOH MINGW64 ~/samplerepo (master)
$ git remote -v
origin  https://github.com/Yuvraj-Pardeshi/Sample-Repo.git (fetch)
origin  https://github.com/Yuvraj-Pardeshi/Sample-Repo.git (push)
```



```
Yuvraj Pardeshi@LAPTOP-MEM3GPOH MINGW64 ~/samplerepo (master)
$ git branch -M main

Yuvraj Pardeshi@LAPTOP-MEM3GPOH MINGW64 ~/samplerepo (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 324 bytes | 29.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Yuvraj-Pardeshi/Sample-Repo.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```



Yuvraj-Pardeshi adding add.cpp 087eb54 1 hour ago 1 commit

add.cpp adding add.cpp 1 hour ago

Help people interested in this repository understand your project by adding a README.

Hence we have create your repo on github and added file to the repo successfully