SOURCE CODE MANAGEMENT

COURSE CODE: CSE2015

SLOT: L3, L4

NAME: ANEESH PRAKASH DHAVADE

ENROLLMENT NUMBER: A86605224286

FACULTY: DR. MONITH KAPOOR

Lab Session 1: Git Fundamentals

Computer

A **computer** is any device capable of performing calculations, whether they are logical or mathematical.

Program/Code

A **program** (or **code**) is a set of instructions, often organized as an algorithm, that directs a computer to perform a specific task.

Need for Managing Source Code

Modern applications, such as Spotify, consist of multiple programs working together on both the frontend and backend to deliver smooth user experience. Regular updates are essential for:

- **Fixing Bugs:** Quickly resolving errors that may occur.
- **Improving UI/UX:** Enhancing the user interface and overall experience.
- **Optimizing Performance:** Addressing and refining issues for better performance.

For programmers, effective management of source code is crucial because:

- It ensures that all files remain in context throughout the lifecycle of the program.
- It facilitates collaboration, allowing multiple developers to work together on a shared codebase.

Tools for Source Code Management

Git

A version control system that runs locally on your computer. Git helps track changes and manage versions of your project.

GitHub

A global, cloud-based platform that hosts Git repositories, enabling developers to share, collaborate, and contribute to projects from anywhere in the world.

Version

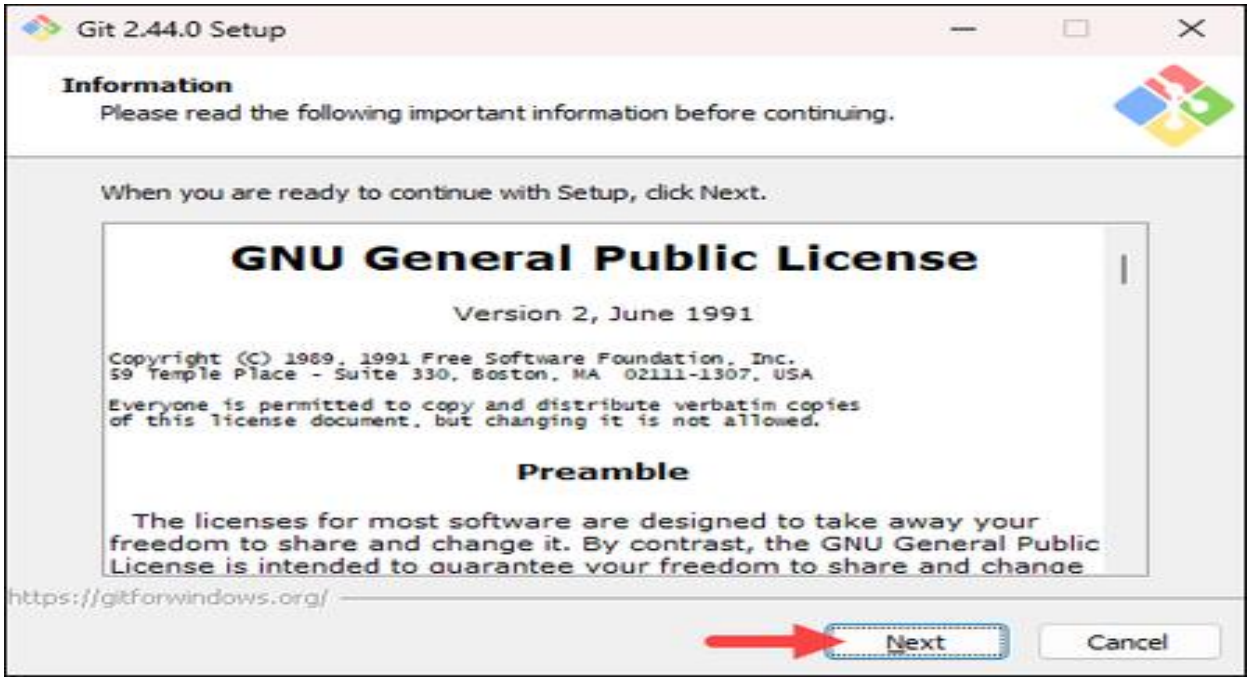
A **version** in version control represents a snapshot of your project at a specific moment in time. This snapshot allows you to review, revert, or compare changes made throughout the development process.

GIT INSTALLATION

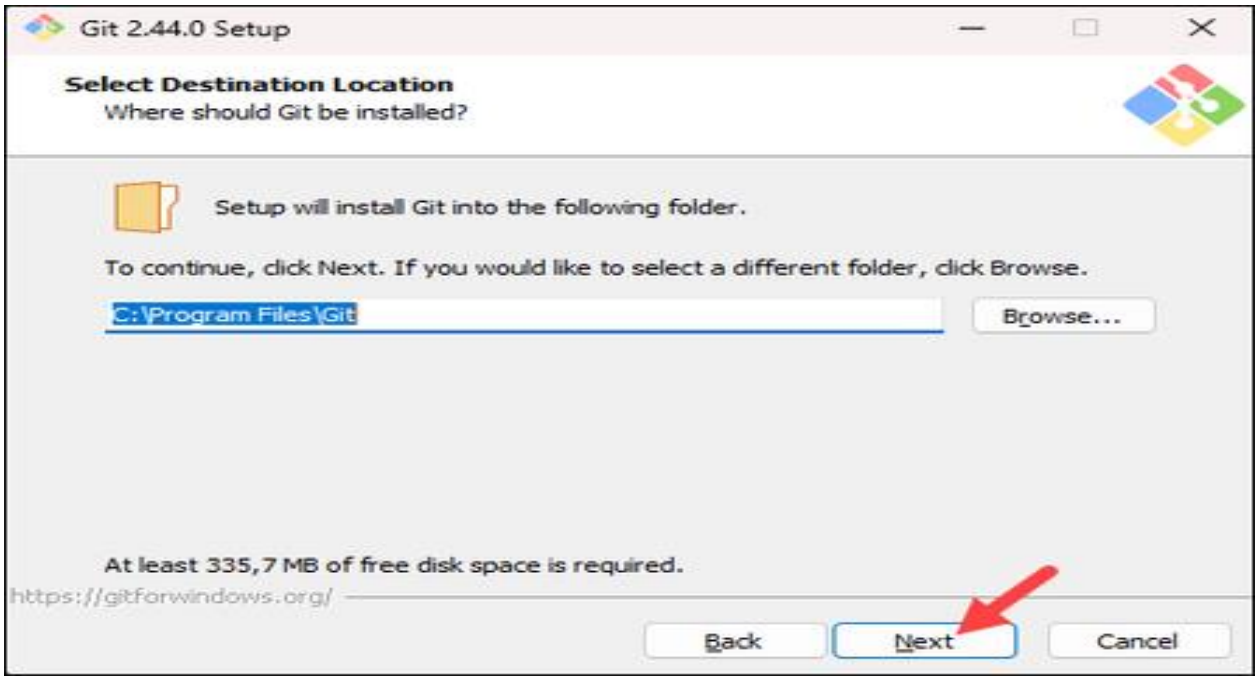
1.wNavigate to the **official Git downloads page** and click the download link for the latest Git version for Windows:

2. Double-click the downloaded **file** to extract and launch the installer

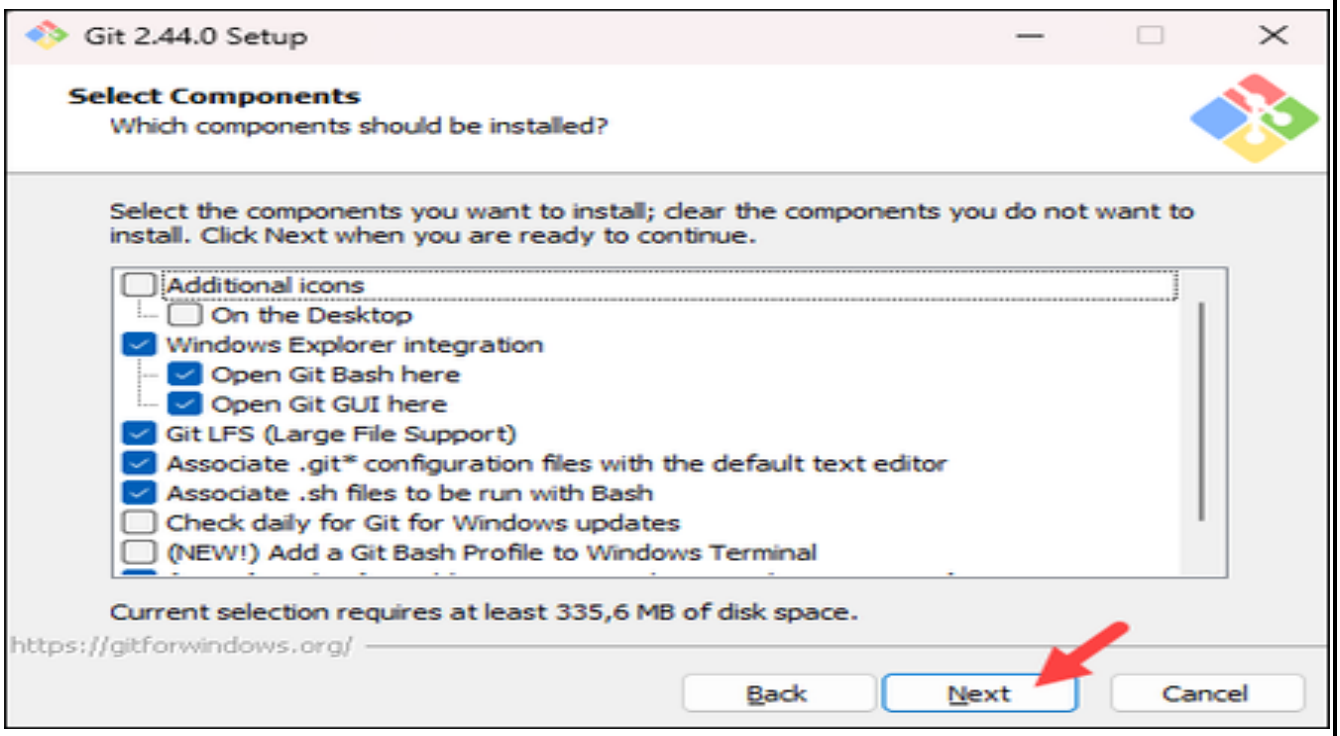
3.Review the **GNU General Public License**, and when you are ready to install, click **Next**.



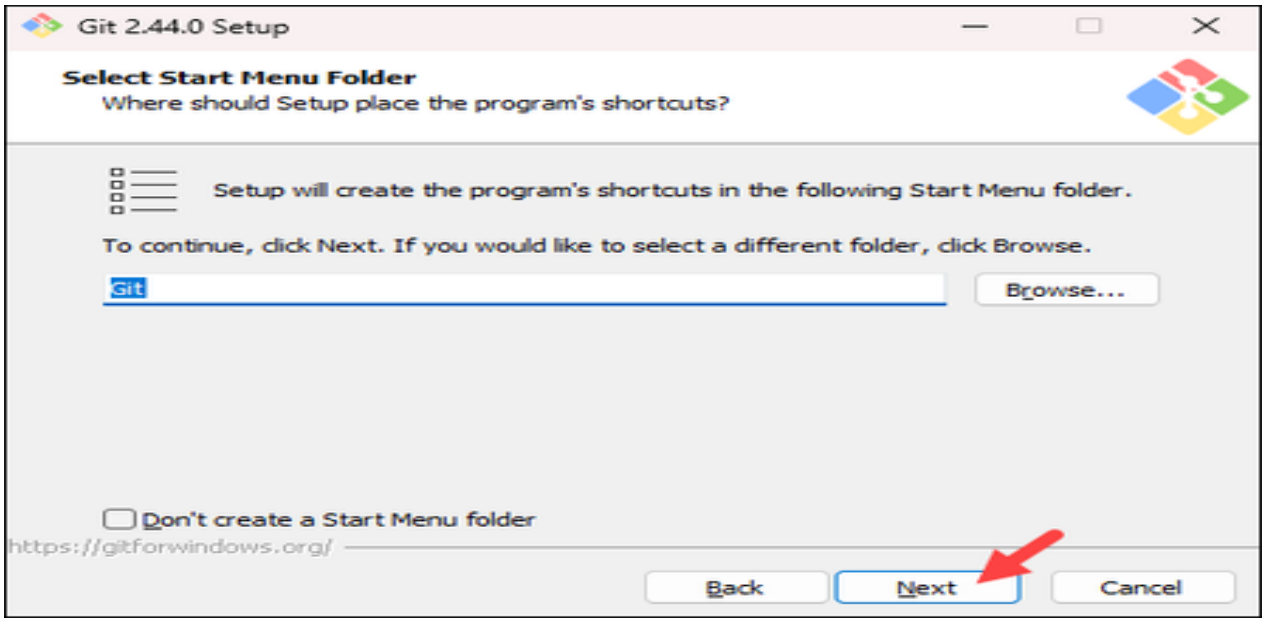
4.The installer prompts you for an installation location. Leave the default one unless you want to change it, and click **Next**.



5.In the component selection screen, leave the defaults unless you need to change them and click **Next**

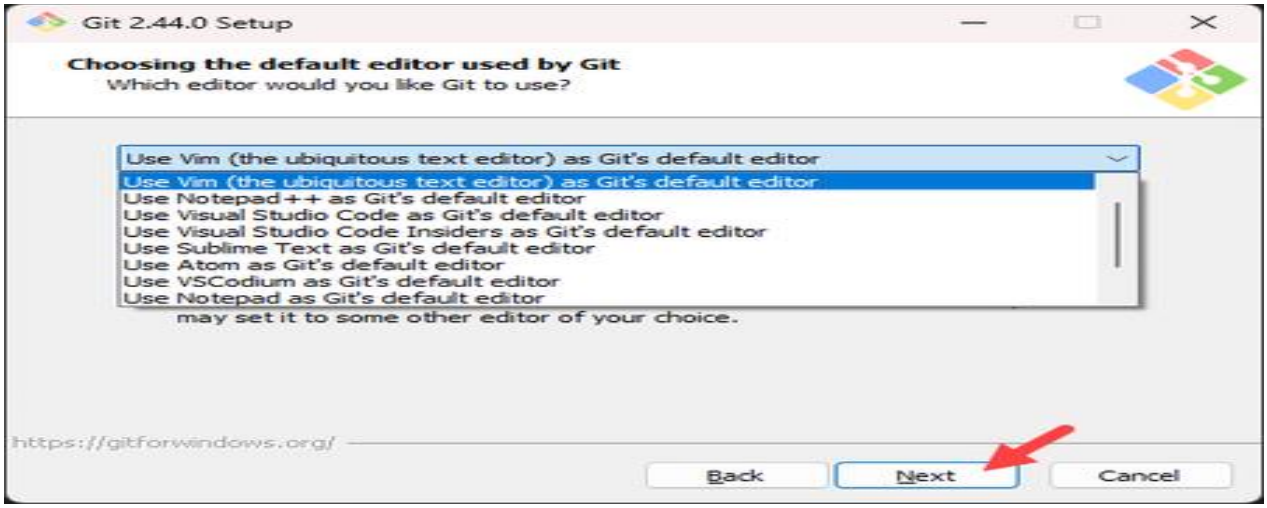


6.The installer offers to create a start menu **folder**. Click **Next** to accept and proceed to the next step.

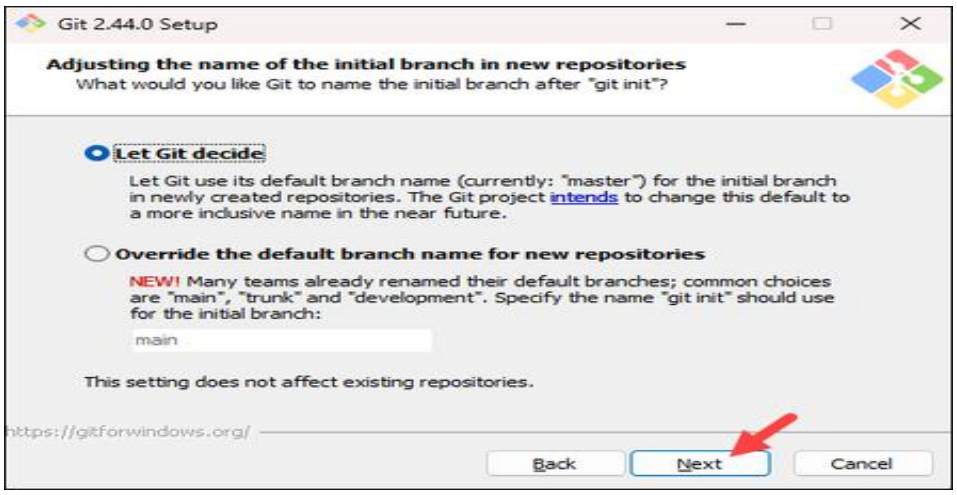


7. Select a text editor you want to use with Git. Use the drop-down menu to select Notepad++ (or whichever text editor you prefer) and click **Next**.

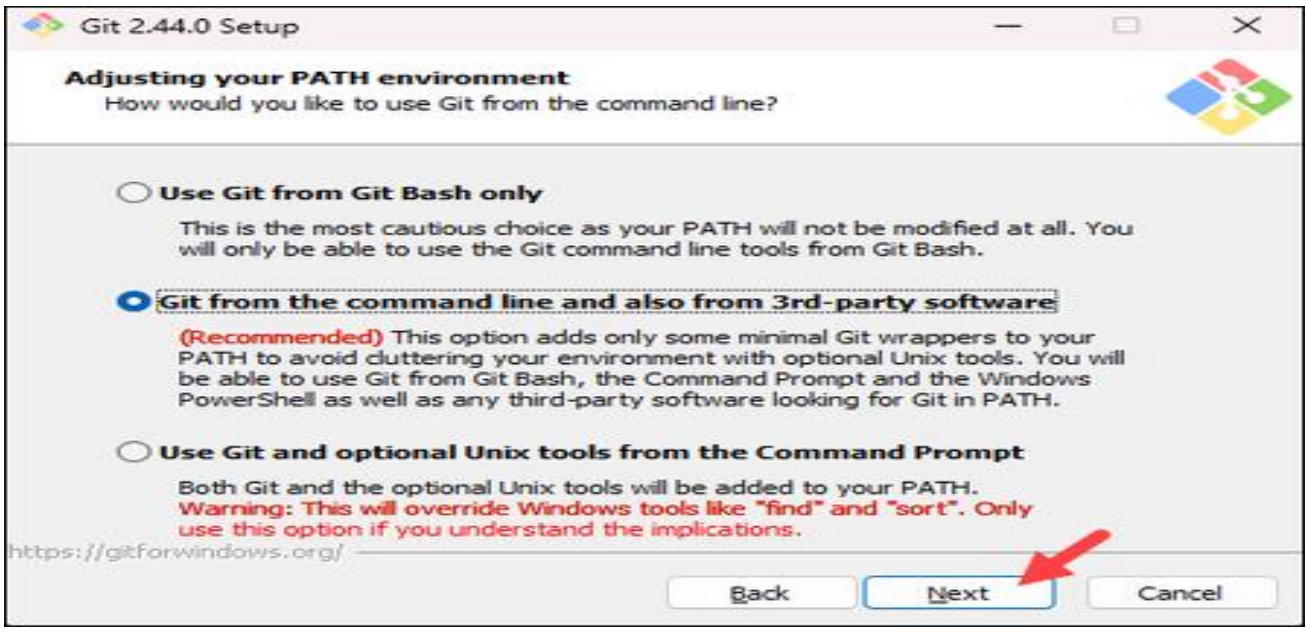
If you prefer to use a CLI text editor in **Git Bash**, select **nano** or **Vim** from the list.



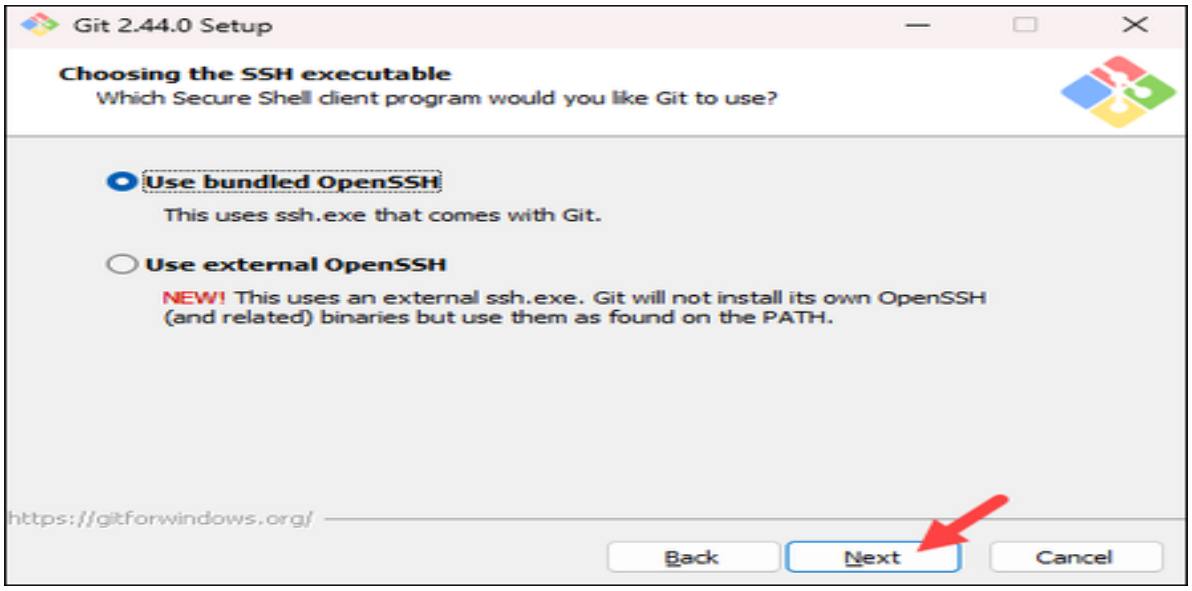
8. The next step allows you to choose a different name for your initial branch. The default is **master**. Unless you are working in a team that requires a different name, leave the default option and click **Next.**



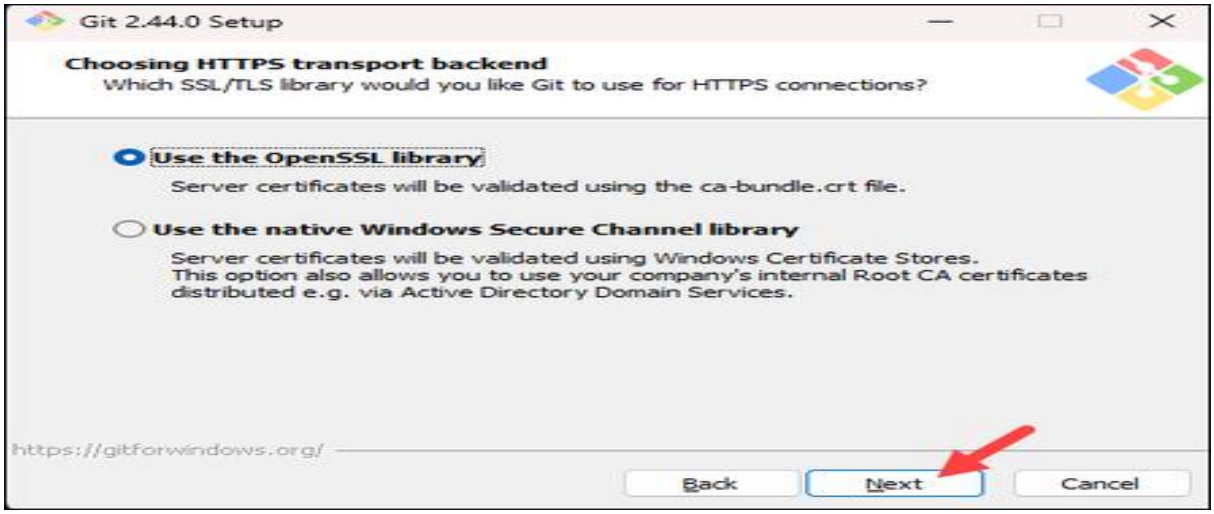
9. The next step allows you to change the **PATH environment**. The **PATH** is the default set of **directories** included when you run a command from the command line. Keep the middle (recommended) selection and click **Next**.



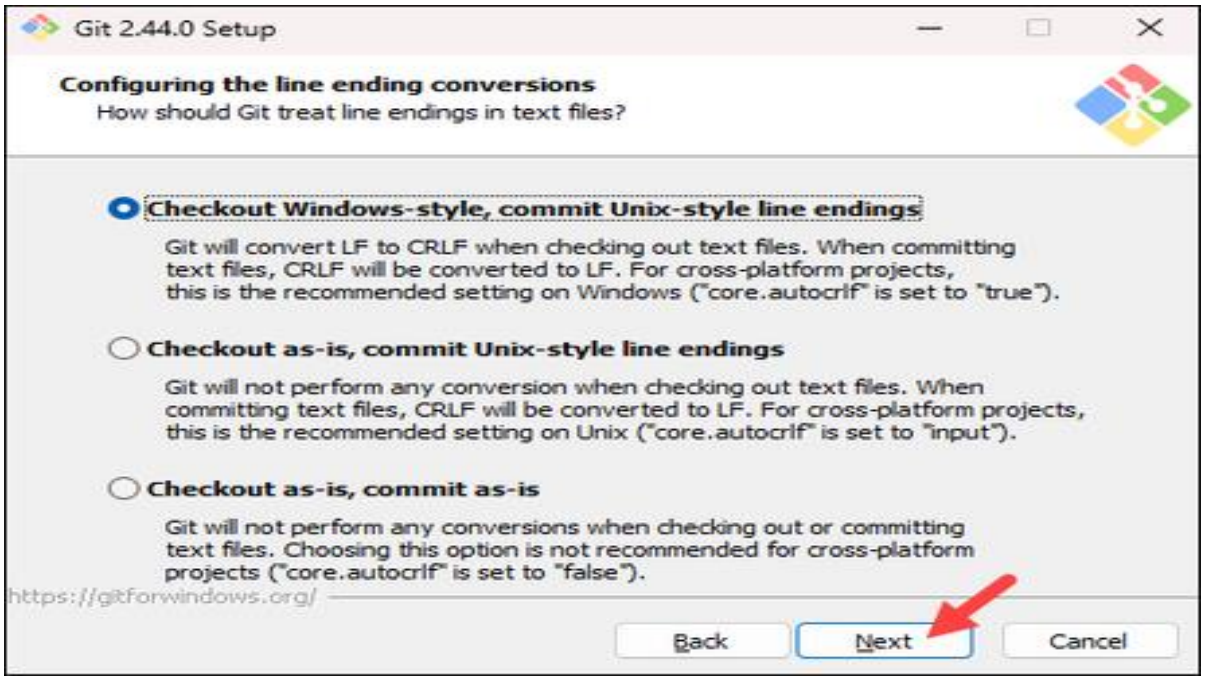
10. The installer prompts you to select the SSH client for Git to use. Git already comes with its own SSH client, so if you don't need a specific one, leave the default option and click **Next.**



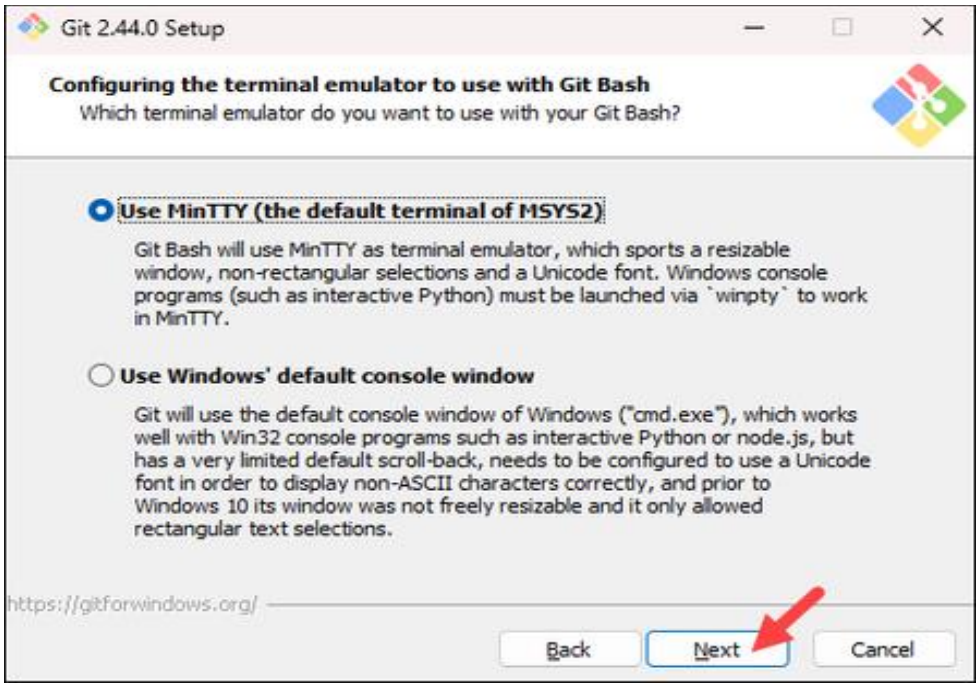
11. The next option relates to server certificates. The default option is recommended for most users. If you work in an Active Directory environment, you may need to switch to Windows Store certificates. Select your preferred option and click **Next**.



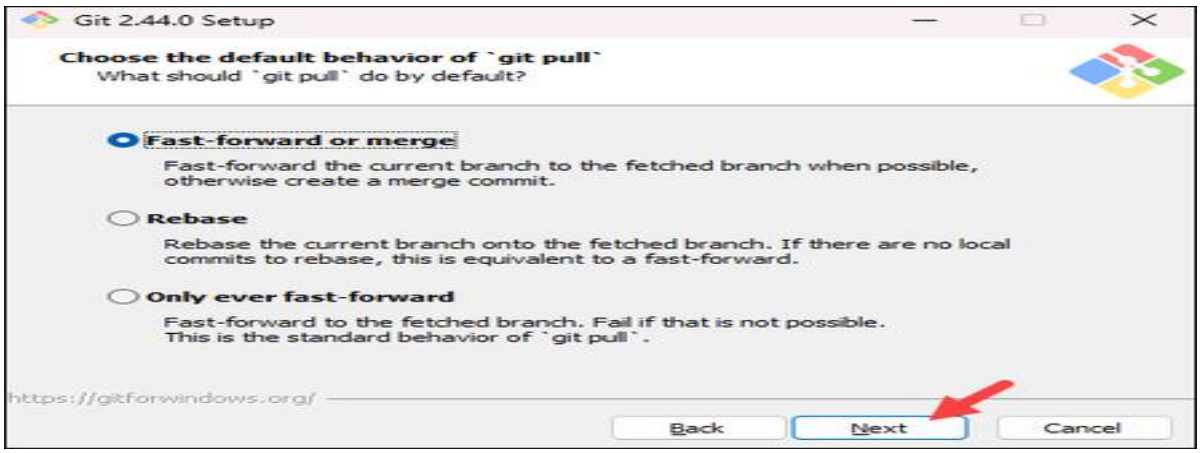
12. The following selection configures line-ending conversion, which relates to the way data is formatted. The default selection is recommended for Windows. Click **Next** to proceed.



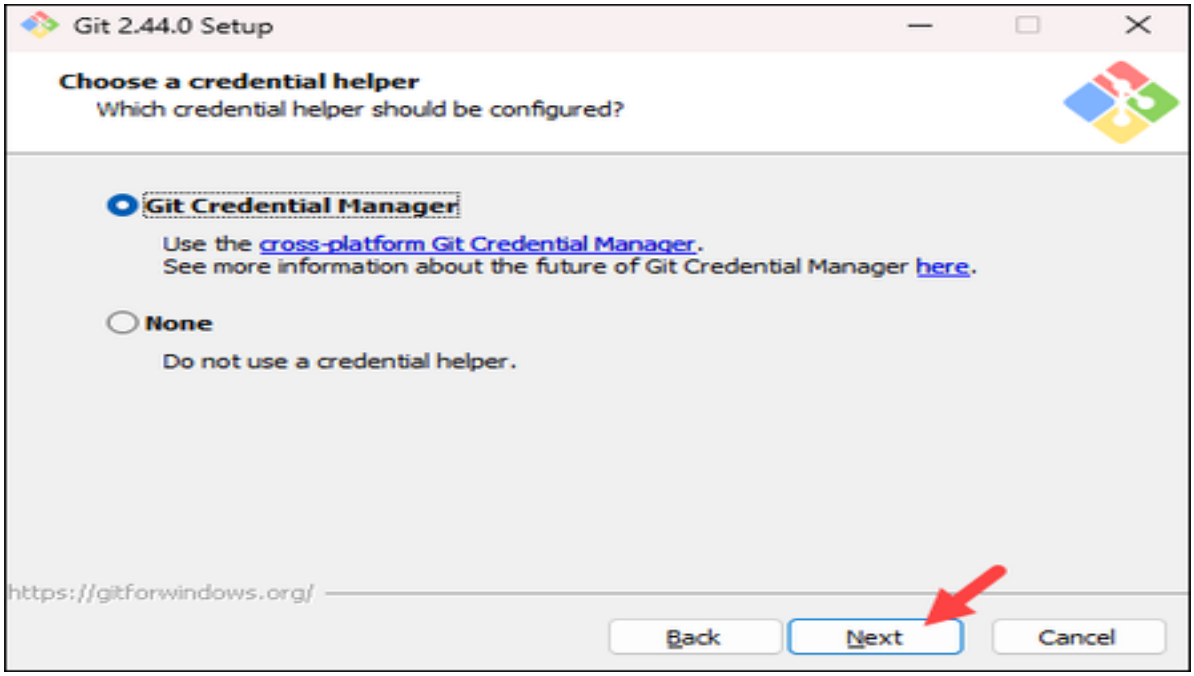
13. Choose the **terminal emulator** you want to use. The default MinTTY is recommended for its features. Click **Next** to continue.



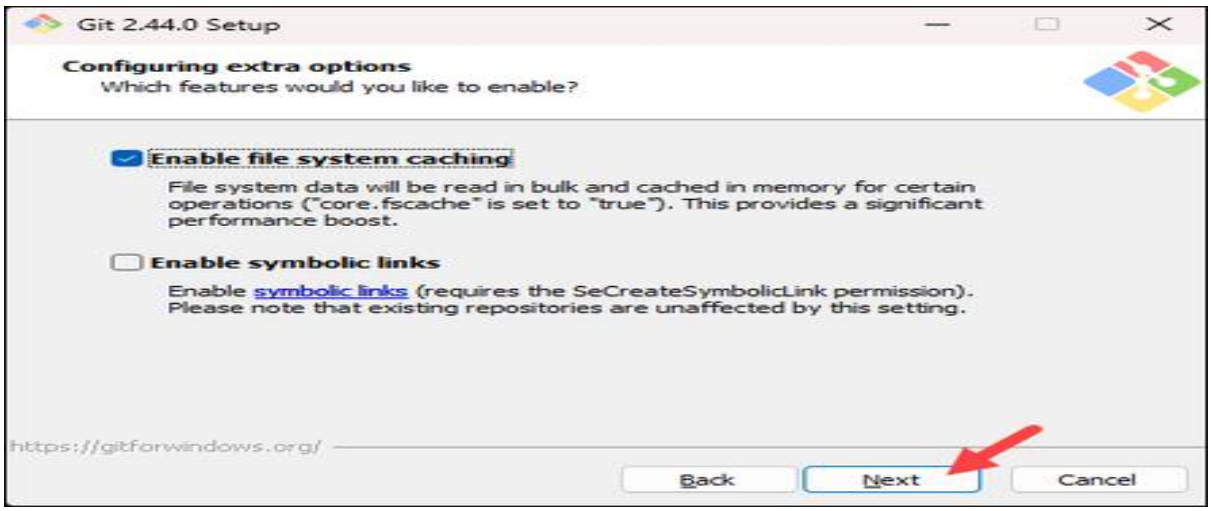
14. The next step allows you to choose what the **git pull** command will do. The default option is recommended unless you specifically need to change its behavior. Click **Next** to continue with the installation.



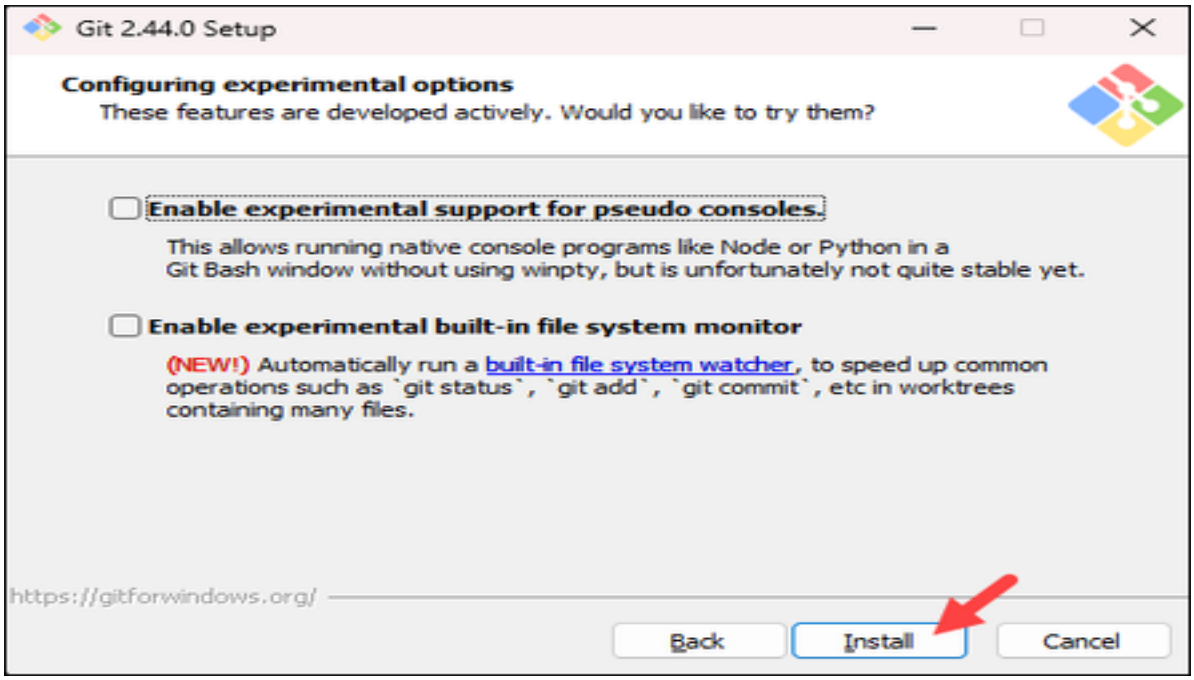
15. The next step is to choose which credential helper to use. Git uses credential helpers to fetch or save credentials. The default option is the most stable one. Select your preferred credential manager and click **Next**.



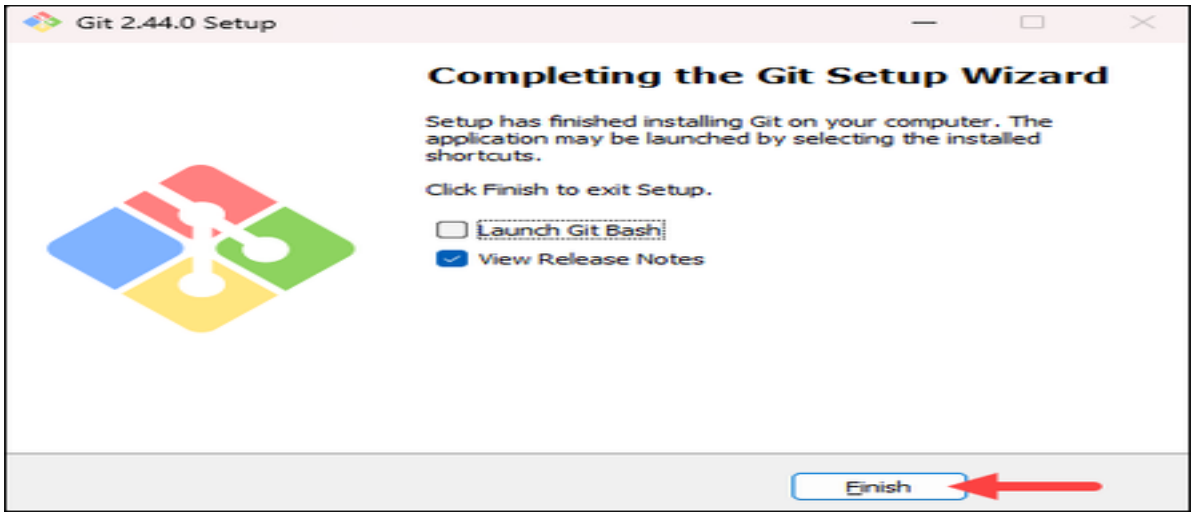
16. The next step lets you decide which extra options to enable. If you use **symbolic links**, which represent shortcuts for the command line, tick the box. Keep **file system** caching checked and click **Next**.



17. Depending on which Git version you are installing, it may offer to install experimental features. At the time this article was written, the installer offered options to include support for pseudo controls and a built-in file system monitor. For the most stable operation, do not install experimental features and click **Install**.



18. Once the installation is complete, tick the boxes to view the Release Notes or launch Git Bash if you want to start using Git right away, and click **Finish**.



COMMNADS

1.pwd

- Presenting work directory
- used to view the current directory working on

2.cd

- change directory
- used to change the directory

3.ls

- used to list all the folders

4.vi

- vim mode/vi editor
- used to create a file, edit it
- Syntax: vi nameoffile (Ex: vi hello.txt)

click the key i to insert or edit your file. then esc to return from insert mode :wq(then enter) to return

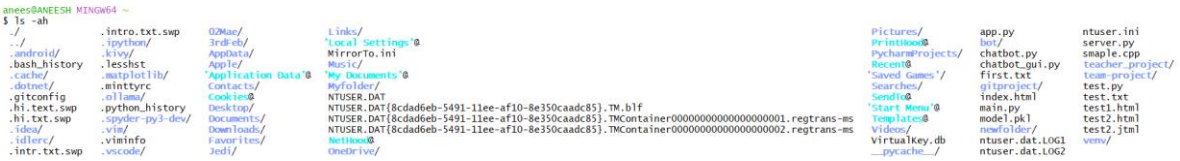
5. cat

- syntax: cat filename (cat hello.txt)
- to view the file contents.

```
anees@ANEESH MINGW64 ~
$ vi test.py

anees@ANEESH MINGW64 ~
$ cat test.py
import speech_recognition
print("SpeechRecognition is installed!")
```

6.ls -ah



7.history

- used to view history of commands

```
451  ls
452  vi styles.css
453  vi styles.css
454  git add .
455  git commit -m "Changes made in the css part"
456  git push origin aneesh-cs
457  cd..
458  cd ..
459  ls
460  cd SCM-Project
461  git checkout Dev
462  git pull origin Dev
463  git checkout main
464  git merge Dev
465  git push orign main
466  git push origin main
467  git push origin Dev
468  ls
469  vi test.py
470  cat test.py
471  ls -ah
472  history
```

9. mkdir

- used to create folder
- Syntax: mkdir nameoffolder

:ls nameoffolder

- confirmation to check if the folder is created

:cd folder/

- to change the present directory as the folder

```
anees@ANEESH MINGW64 ~
$ mkdir project1

anees@ANEESH MINGW64 ~
$ cd project1

anees@ANEESH MINGW64 ~/project1
$ vi new.c
```

- also after this create one new file by using vi (then insert i)

10.git init

- initialize the repository
- working directory to empty repository

```
anees@ANEESH MINGW64 ~/project1
$ git init
Initialized empty Git repository in C:/Users/anees/project1/.git/
```

11.git status

- display the current status of the working directory.

```
anees@ANEESH MINGW64 ~/project1 (master)
$ git status
On branch master

No commits yet

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

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

12.git add file

- to add the file
- Syntax: git add filename
- After this edit the file (using vi, insert i)

```
anees@ANEESH MINGW64 ~/project1 (master)
$ git add .

anees@ANEESH MINGW64 ~/project1 (master)
$ git status
On branch master

No commits yet

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

13. git commit

- Used to commit a change
- Syntax: git commit -m “Message”

```
anees@ANEESH MINGW64 ~/project1 (master)
$ git commit -m "First commit"
[master (root-commit) 2618b9f] First commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 new.c
```

15. git log

- used to view status/entries of commits

```
anees@ANEESH MINGW64 ~/project1 (master)
$ git log
commit 2618b9f7a9edbc4de7636d466aebddd5e51a73a5 (HEAD -> master)
Author: Aneesh1856 <aneesh.dhavade@s.amity.edu>
Date:   Mon Jun 2 09:19:47 2025 +0530

    First commit
```

16. git diff

- shows the changes made in the working directory since the last commit

17.git branch<branch name>

- creates a new branch

```
anees@ANEESH MINGW64 ~/project1 (master)
$ git branch test
```

18. git branch

- lists all the branches in the repository

```
anees@ANEESH MINGW64 ~/project1 (master)
$ git branch
* master
  test
```

19.git log -onelinegit

- view a concise commit history

```
anees@ANEESH MINGW64 ~/project1 (master)
$ git log --oneline
2618b9f (HEAD -> master, test) First commit
```

20. git checkout<branch name>

- switch to the master branch

```
anees@ANEESH MINGW64 ~/project1 (master)
$ git checkout test
Switched to branch 'test'

anees@ANEESH MINGW64 ~/project1 (test)
$
```

SCM Project

The project was to make a repository in GitHub, make 3 branches and merge it with the main branch and access all 4 team-mate's repositories, fork it, clone it, make some changes and merge them.

First, make your own repositories and make 3 branches and add files and merge with the main branch.

1) Go to the directory on your computer

```
anees@ANEESH MINGW64 ~
$ cd "D:\ANEESH DHAVADE GIT & GITHUB\SCM Project"
```

2) Clone the GitHub repositories

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project
$ git clone https://github.com/Aneesh1856/SCM-Project
Cloning into 'SCM-Project'...
warning: You appear to have cloned an empty repository.
```

3) Change to the folder

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project
$ cd SCM-Project
```

4) Make an initial commit as a README.md and commit it

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (main)
$ echo "Initial Commit" > README.md

anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (main)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (main)
$ git commit -m "Initial Commit"
[main (root-commit) d660c0b] Initial Commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

5) Make four branches

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (main)
$ git branch Python

anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (main)
$ git branch Frontend

anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (main)
$ git branch Backend

anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (main)
$ git branch Dev
```

6) Checkout to the first branch, add the files and make two commits

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (main)
$ git checkout Python
Switched to branch 'Python'

anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (Python)
$ vi app.py

anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (Python)
$ git add .
warning: in the working copy of 'app.py', LF will be replaced by CRLF the next time Git touches it

anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (Python)
$ git commit -m "First commit for Python branch"
[Python 4f77074] First commit for Python branch
 1 file changed, 19 insertions(+)
 create mode 100644 app.py

anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (Python)
$ vi app.py

anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (Python)
$ git add .
warning: in the working copy of 'app.py', LF will be replaced by CRLF the next time Git touches it

anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (Python)
$ git commit -m "Second commit for Python branch"
[Python edae8da] Second commit for Python branch
 1 file changed, 25 insertions(+)
```

7) Checkout to the second branch, add the files and make two commits

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (Python)
$ git checkout Frontend
Switched to branch 'Frontend'

anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (Frontend)
$ vi index.html

anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (Frontend)
$ git add .
warning: in the working copy of 'index.html', LF will be replaced by CRLF the next time Git touches it

anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (Frontend)
$ git commit -m "First commit for Frontend branch"
[Frontend 5f2d818] First commit for Frontend branch
 1 file changed, 22 insertions(+)
 create mode 100644 index.html

anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (Frontend)
$ vi style.css

anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (Frontend)
$ git add .
warning: in the working copy of 'style.css', LF will be replaced by CRLF the next time Git touches it

anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (Frontend)
$ git commit -m "Second xommit for Frontend branch"
[Frontend 2c85396] Second xommit for Frontend branch
 1 file changed, 83 insertions(+)
 create mode 100644 stvle.css
```

8) Checkout to the third branch, add the files and make two commits

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (Frontend)
$ git checkout Backend
Switched to branch 'Backend'

anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (Backend)
$ vi script.js

anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (Backend)
$ git add .
warning: in the working copy of 'script.js', LF will be replaced by CRLF the next time Git touches it

anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (Backend)
$ git commit -m "First commit for Backend branch"
[Backend c0eb97d] First commit for Backend branch
 1 file changed, 27 insertions(+)
 create mode 100644 script.js
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (Backend)
$ vi script.js
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (Backend)
$ git add .
warning: in the working copy of 'script.js', LF will be replaced by CRLF the next time Git touches it

anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (Backend)
$ git commit -m "Second commit for Backend branch"
[Backend bd9b3cc] Second commit for Backend branch
 1 file changed, 19 insertions(+)
```

9) Checkout to the main branch

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (Backend)
$ git checkout main
Switched to branch 'main'
Your branch is based on 'origin/main', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)
```

10) Merge all the branches to the main branch

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (main)
$ git merge Python -m "Merging python branch"
Updating d660c0b..edae8da
Fast-forward (no commit created; -m option ignored)
 app.py | 44 ++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 44 insertions(+)
 create mode 100644 app.py

anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (main)
$ git merge Frontend -m "Merging frontend branch"
Merge made by the 'ort' strategy.
 index.html | 22 +++++++++++++++++
 style.css  | 83 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 2 files changed, 105 insertions(+)
 create mode 100644 index.html
 create mode 100644 style.css

anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (main)
$ git merge Backend -m "Merging backend branch"
Merge made by the 'ort' strategy.
 script.js | 46 ++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 46 insertions(+)
 create mode 100644 script.js
```

11) Push all the commits and files in the Github repository

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/SCM-Project (main)
$ git push origin --all
Enumerating objects: 25, done.
Counting objects: 100% (25/25), done.
Delta compression using up to 8 threads
Compressing objects: 100% (23/23), done.
Writing objects: 100% (25/25), 3.93 KiB | 1.31 MiB/s, done.
Total 25 (delta 8), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (8/8), done.
To https://github.com/Aneesh1856/SCM-Project
 * [new branch]      Backend -> Backend
 * [new branch]      Dev -> Dev
 * [new branch]      Frontend -> Frontend
 * [new branch]      Python -> Python
 * [new branch]      main -> main
```

Now we will clone the repository of the rest of my Team Members, make some changes in their repositories, and send pull requests.

Sanin:

1) Go back to the folder where you want to save the folder

```
anees@ANEESH MINGW64 ~
$ cd "D:\ANEESH DHAVADE GIT & GITHUB\SCM Project"
```

2) Clone the forked repository

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project
$ git clone https://github.com/Aneesh1856/sanin-repo
Cloning into 'sanin-repo'...
remote: Enumerating objects: 25, done.
remote: Counting objects: 100% (25/25), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 25 (delta 9), reused 25 (delta 9), pack-reused 0 (from 0)
Receiving objects: 100% (25/25), 5.92 KiB | 866.00 KiB/s, done.
Resolving deltas: 100% (9/9), done.
```

3) Go to the cloned repository

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project
$ cd sanin-repo
```

4) Create a new branch and checkout to it

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/sanin-repo (main)
$ git branch aneesh-html

anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/sanin-repo (main)
$ git checkout aneesh-html
Switched to branch 'aneesh-html'
```

5) Make some changes in the repository

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/sanin-repo (aneesh-html)
$ vi index.html
```

6) Add the edited file

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/sanin-repo (aneesh-html)
$ git add .
warning: in the working copy of 'index.html', LF will be replaced by CRLF the next time Git touches it
```

7) Commit the file

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/sanin-repo (aneesh-html)
$ git commit -m "Changes made in the html file"
[aneesh-html 20b9152] Changes made in the html file
 1 file changed, 124 insertions(+), 103 deletions(-)
```

8) Push it to the forked repository for pull request

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/sanin-repo (aneesh-html)
$ git push origin aneesh-html
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 548 bytes | 548.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'aneesh-html' on GitHub by visiting:
remote:      https://github.com/Aneesh1856/sanin-repo/pull/new/aneesh-html
remote:
To https://github.com/Aneesh1856/sanin-repo
 * [new branch]      aneesh-html -> aneesh-html
```

Dominic:

1) Go back to the folder where you want to save the folder

```
anees@ANEESH MINGW64 ~
$ cd "D:\ANEESH DHAVADE GIT & GITHUB\SCM Project"
```

2) Clone the forked repository

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project
$ git clone https://github.com/Aneesh1856/dominic-repo
Cloning into 'dominic-repo'...
remote: Enumerating objects: 34, done.
remote: Counting objects: 100% (34/34), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 34 (delta 13), reused 31 (delta 13), pack-reused 0 (from 0)
Receiving objects: 100% (34/34), 12.29 KiB | 740.00 KiB/s, done.
Resolving deltas: 100% (13/13), done.
```

3) Go to the cloned repository

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project
$ cd dominic-repo
```

4) Create a new branch and checkout to it

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/dominic-repo (main)
$ git branch aneesh-css

anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/dominic-repo (main)
$ git checkout aneesh-css
Switched to branch 'aneesh-css'
```

5) Make some changes in the repository

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/dominic-repo (aneesh-css)
$ vi styles.css
```

6) Add the edited file

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/dominic-repo (aneesh-css)
$ git add .
warning: in the working copy of 'styles.css', LF will be replaced by CRLF the next time Git touches it
```

7) Commit the file

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/dominic-repo (aneesh-css)
$ git commit -m "Changes made in the css file"
[aneesh-css c10472c] Changes made in the css file
 1 file changed, 108 insertions(+), 221 deletions(-)
```

8) Push it to the forked repository for pull request

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/dominic-repo (aneesh-css)
$ git push origin aneesh-css
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.25 KiB | 1.25 MiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'aneesh-css' on GitHub by visiting:
remote:      https://github.com/Aneesh1856/dominic-repo/pull/new/aneesh-css
remote:
To https://github.com/Aneesh1856/dominic-repo
 * [new branch]      aneesh-css -> aneesh-css
```

Shreyas:

1) Go back to the folder where you want to save the folder

```
anees@ANEESH MINGW64 ~
$ cd "D:\ANEESH DHAVADE GIT & GITHUB\SCM Project"
```

2) Clone the forked repository

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project
$ git clone https://github.com/Aneesh1856/shreyas-repo
Cloning into 'shreyas-repo'...
remote: Enumerating objects: 35, done.
remote: Counting objects: 100% (35/35), done.
remote: Compressing objects: 100% (21/21), done.
remote: Total 35 (delta 12), reused 31 (delta 11), pack-reused 0 (from 0)
Receiving objects: 100% (35/35), 12.46 KiB | 1.25 MiB/s, done.
Resolving deltas: 100% (12/12), done.
```

3) Go to the cloned repository

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project
$ cd shreyas-repo
```

4) Create a new branch and checkout to it

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/shreyas-repo (main)
$ git checkout -b aneesh-cs
Switched to a new branch 'aneesh-cs'
```

5) Make some changes in the repository

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/shreyas-repo (aneesh-cs)
$ vi styles.css
```

6) Add the edited file

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/shreyas-repo (aneesh-cs)
$ git add .
warning: in the working copy of 'styles.css', LF will be replaced by CRLF the next time Git touches it
```

7) Commit the file

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/shreyas-repo (aneesh-cs)
$ git commit -m "Changes made in the css part"
[aneesh-cs 427f14f] Changes made in the css part
 1 file changed, 172 insertions(+), 143 deletions(-)
```

8) Push it to the forked repository for pull request

```
anees@ANEESH MINGW64 /d/ANEESH DHAVADE GIT & GITHUB/SCM Project/shreyas-repo (aneesh-cs)
$ git push origin aneesh-cs
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.54 KiB | 1.54 MiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'aneesh-cs' on GitHub by visiting:
remote:      https://github.com/Aneesh1856/shreyas-repo/pull/new/aneesh-cs
remote:
To https://github.com/Aneesh1856/shreyas-repo
 * [new branch]      aneesh-cs -> aneesh-cs
```