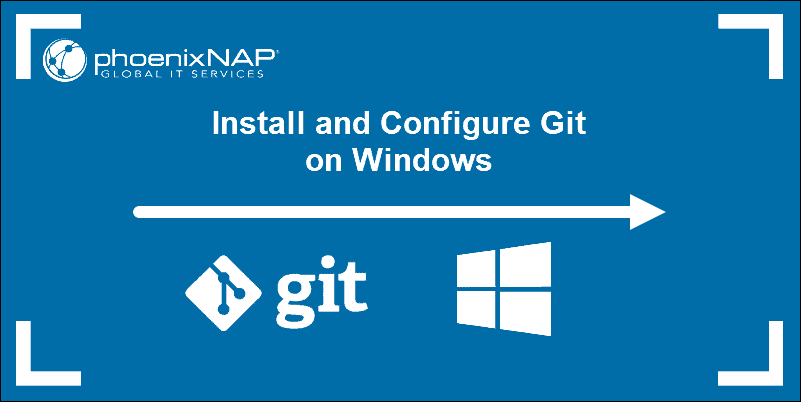
**Introduction**

Git is a widely used open-source software tracking application used to track projects across different teams and revision levels.

**This guide will show you how to install Git on Windows.**

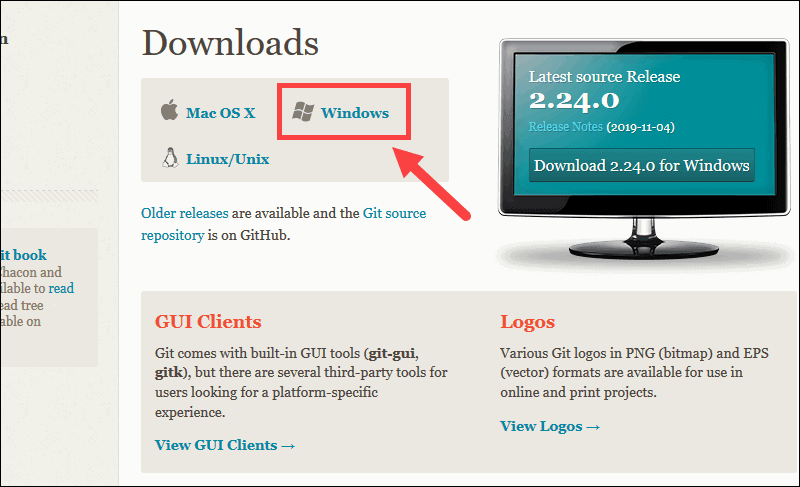


**Prerequisites**

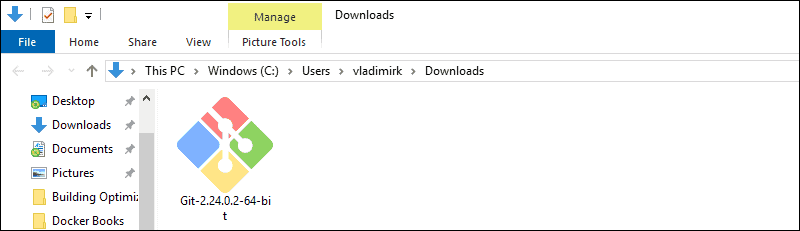
* Administrator privileges
* Access to a command-line
* Your favourite coding text editor
* Username and password for the Github website (optional)

### Download Git for Windows

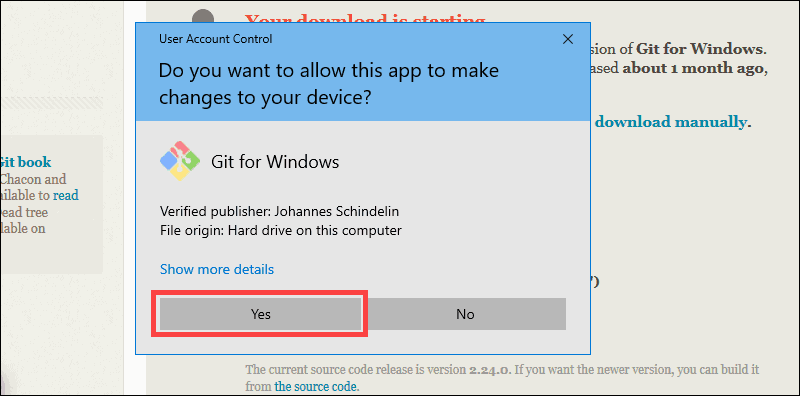
1. Browse to the official Git website:   
2. Click the download link for Windows and allow the download to complete.



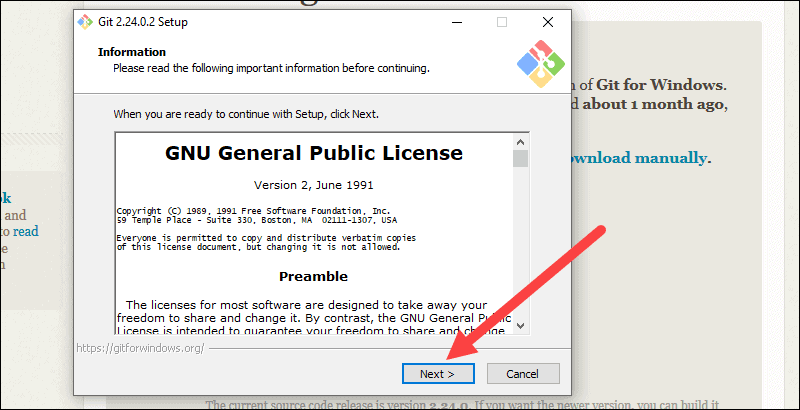
3. Browse to the download location (or use the download shortcut in your browser). Double-click the file to extract and launch the installer.



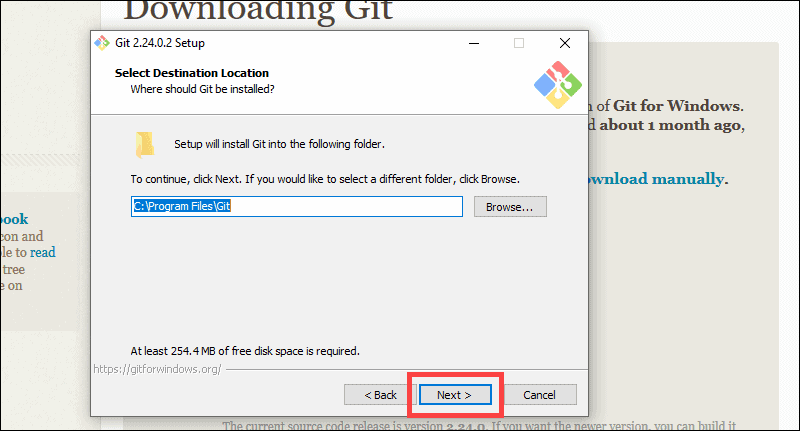
4. Allow the app to make changes to your device by clicking **Yes** on the User Account Control dialog that opens.



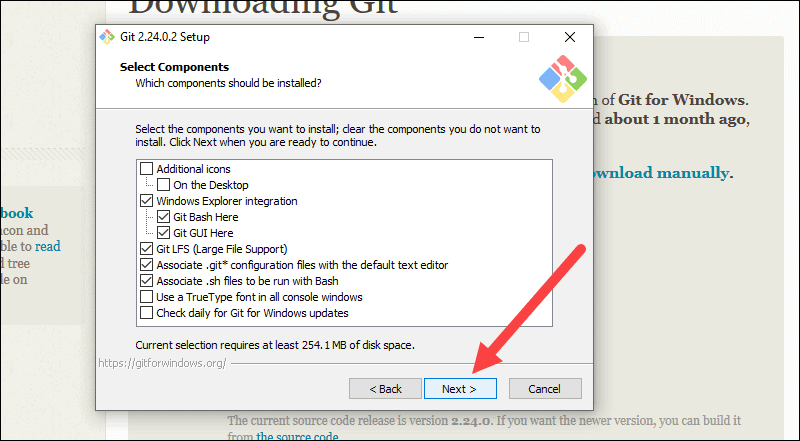
5. Review the GNU General Public License, and when you’re ready to install, click **next**.



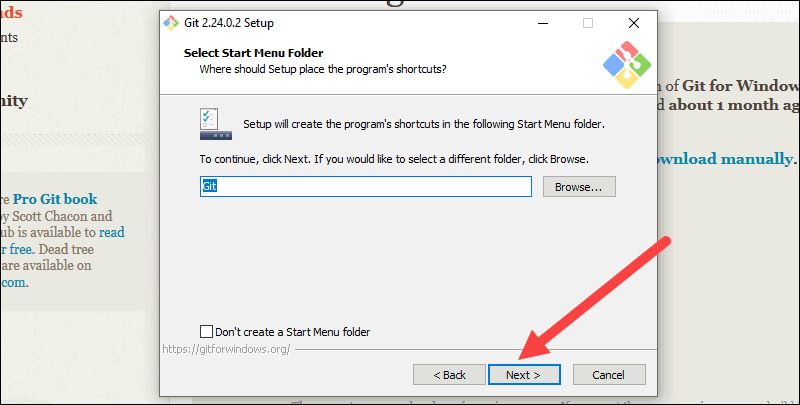
6. The installer will ask you for an installation location. Leave the default, unless you have reason to change it, and click **next**.



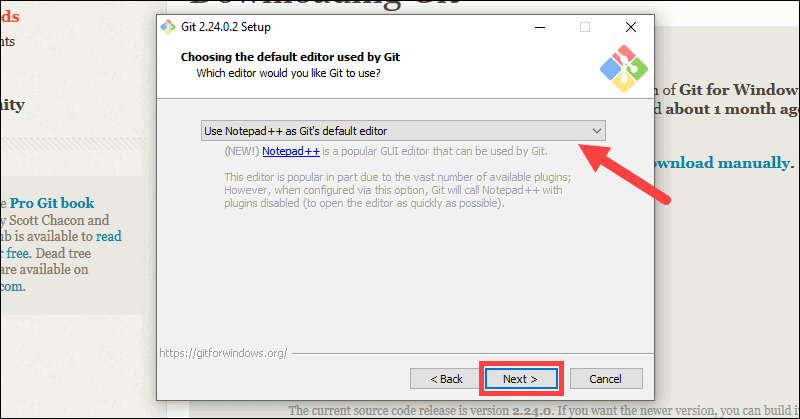
7. A component selection screen will appear. Leave the defaults unless you have a specific need to change them and click **next**.



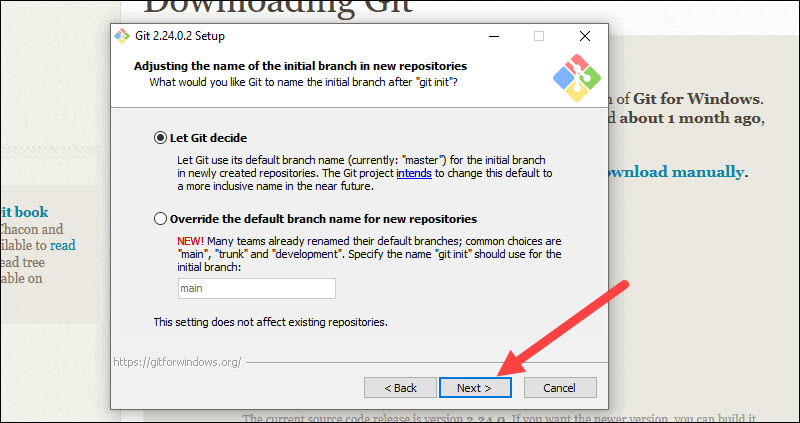
8. The installer will offer to create a start menu folder. Simply click **next**.



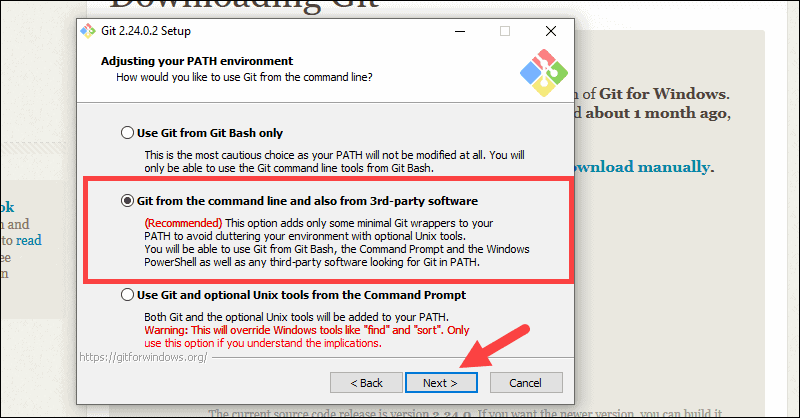
9. Select a text editor you’d like to use with Git. Use the drop-down menu to select Notepad++ (or whichever text editor you prefer) and click **Next**.



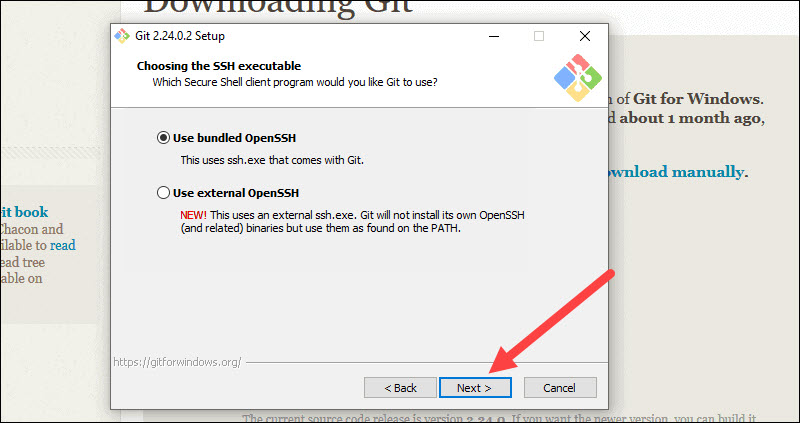
10. The next step allows you to choose a different name for your initial branch. The default is 'master.' Unless you're working in a team that requires a different name, leave the default option and click **next.**



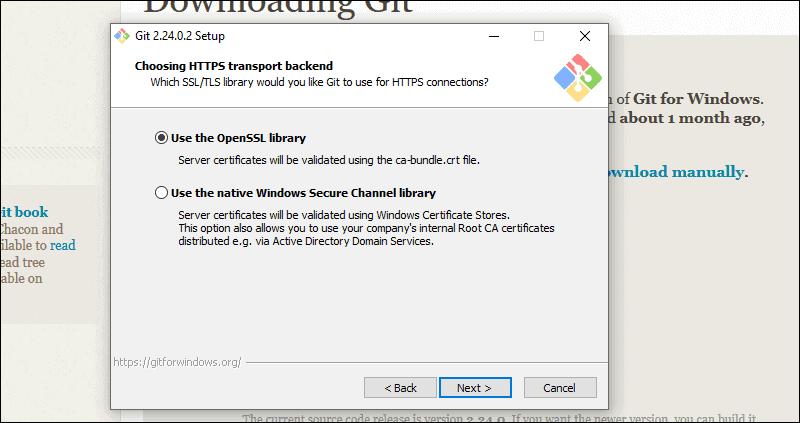
11. This installation step allows you to change the **PATH environment**. The **PATH**is the default set of directories included when you run a command from the command line. Leave this on the middle (recommended) selection and click **next**.



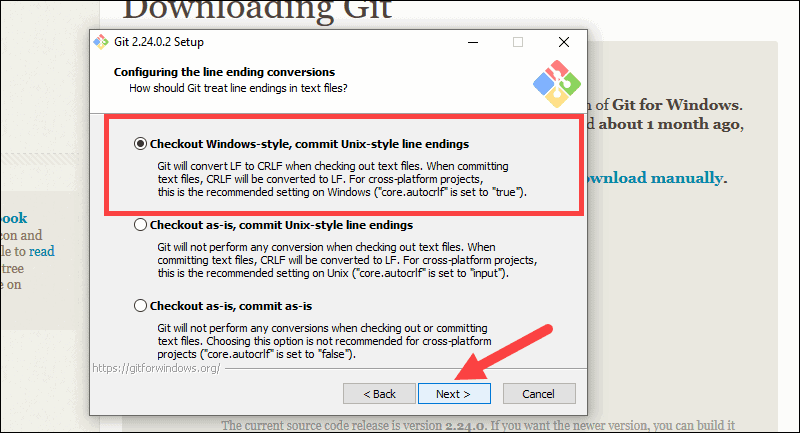
12. The installer now asks which SSH client you want Git to use. Git already comes with its own SSH client, so if you don't need a specific one, leave the default option and click **next.**



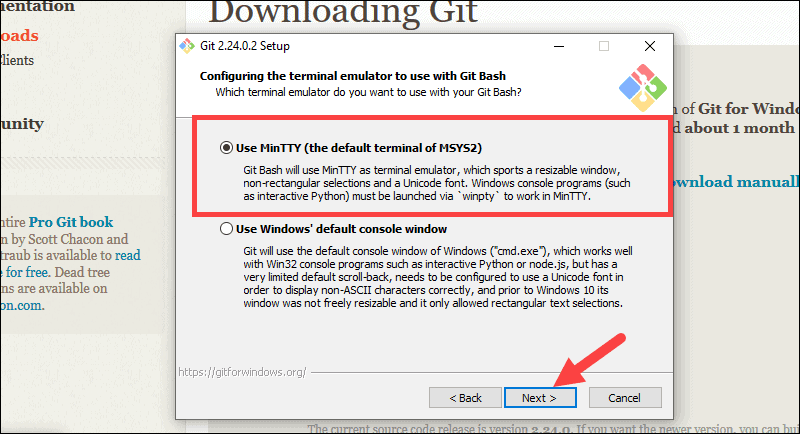
13. The next option relates to server certificates. Most users should use the default. If you’re working in an Active Directory environment, you may need to switch to Windows Store certificates. Click **Next**.



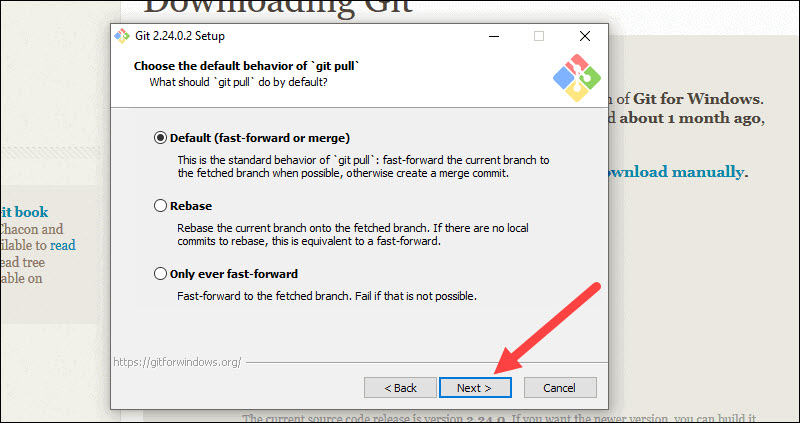
14. The next selection converts line endings. It is recommended that you leave the default selection. This relates to the way data is formatted and changing this option may cause problems. Click **Next**.



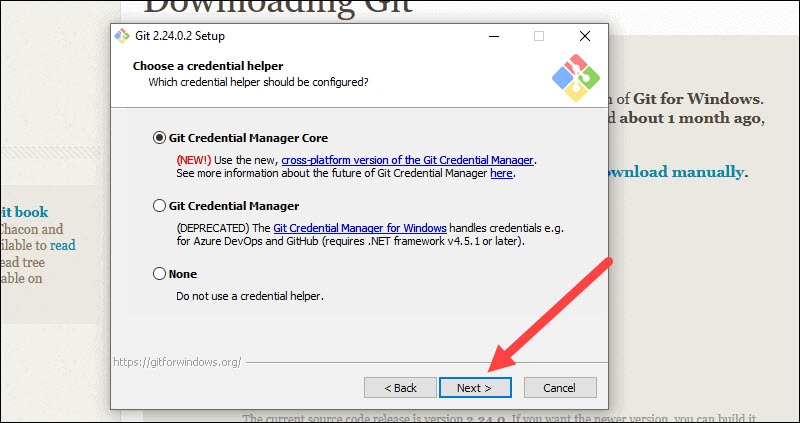
15. Choose the [terminal emulator](https://phoenixnap.com/glossary/terminal-emulation) you want to use. The default MinTTY is recommended, for its features. Click **Next**.



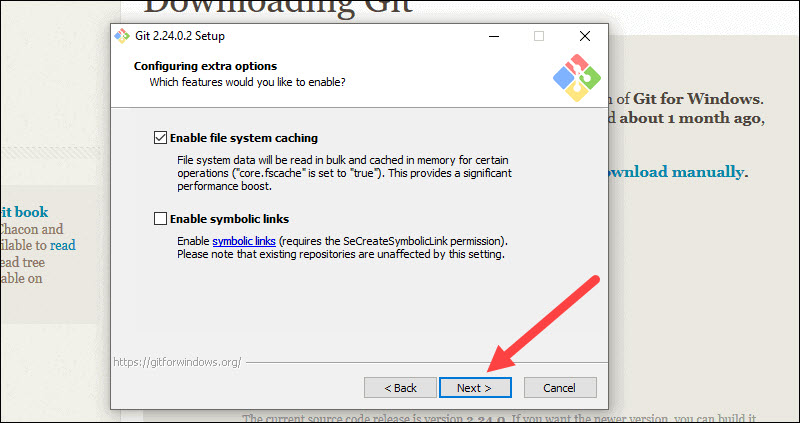
16. The installer now asks what the **git pull** command should do. The default option is recommended unless you specifically need to change its behavior. Click **Next**to continue with the installation.



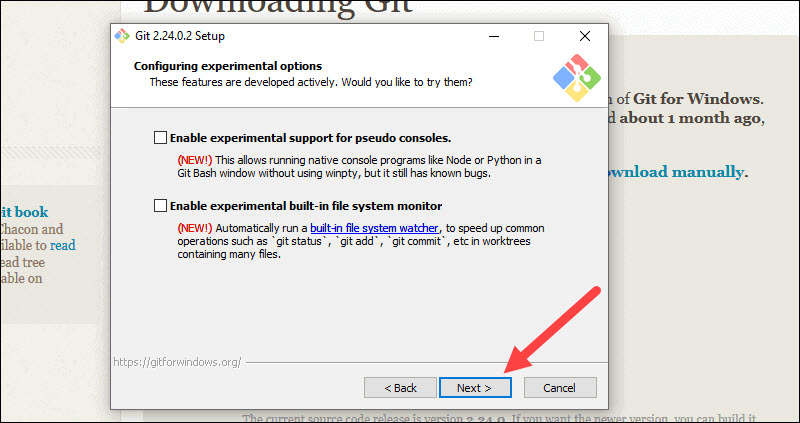
17. Next you should choose which credential helper to use. Git uses credential helpers to fetch or save credentials. Leave the default option as it is the most stable one, and click **Next**.



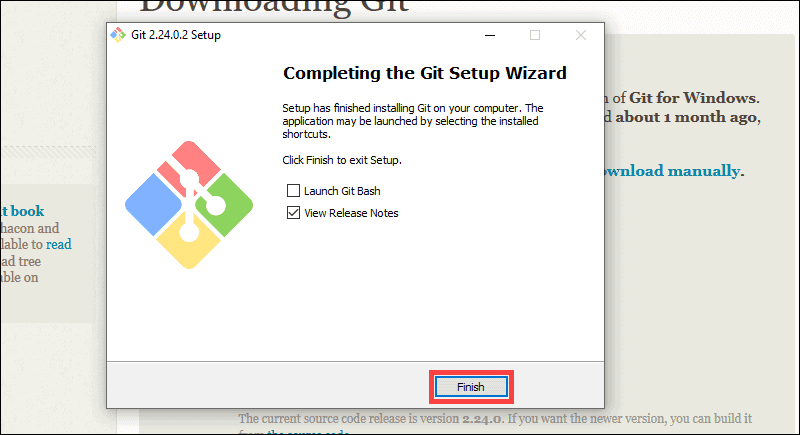
18. The default options are recommended, however this step allows you to decide which extra option you would like to enable. If you use symbolic links, which are like shortcuts for the command line, tick the box. Click **Next**.



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**.

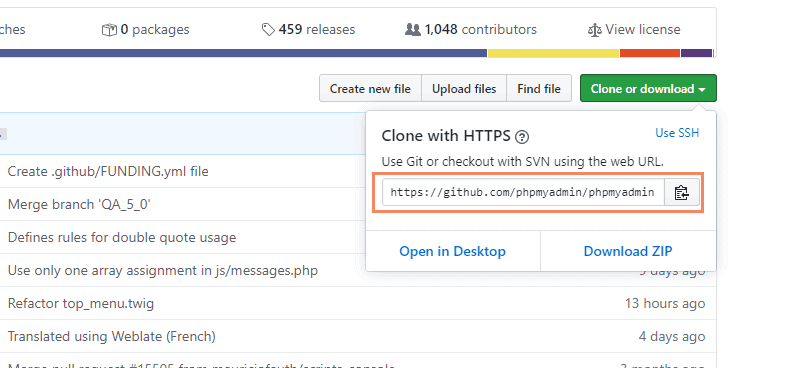


20. Once the installation is complete, tick the boxes to view the Release Notes or Launch Git Bash, Then click **Finish**.



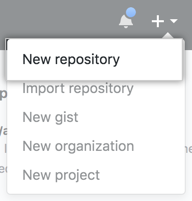
**Clone a GitHub Repository**

Go to your repository on GitHub. In the top right above the list of files, open the **Clone or download** drop-down menu. Copy the **URL for cloning over HTTPS**.



**step1**: In the upper-right corner of any page, use the drop-down menu, and select **new repository.**

After creating your account in gilt, go to your Projects page and click new project. From the Blank project tab, give the project a name and add a description. If you want it to be a public repository, click the Public option. Make sure the Initialize repository with README option is left unchecked.



**Step 2:** Type a short, memorable name for your repository. for example, "mini project ".

Sign in to Gilt with at least the Maintainer role.

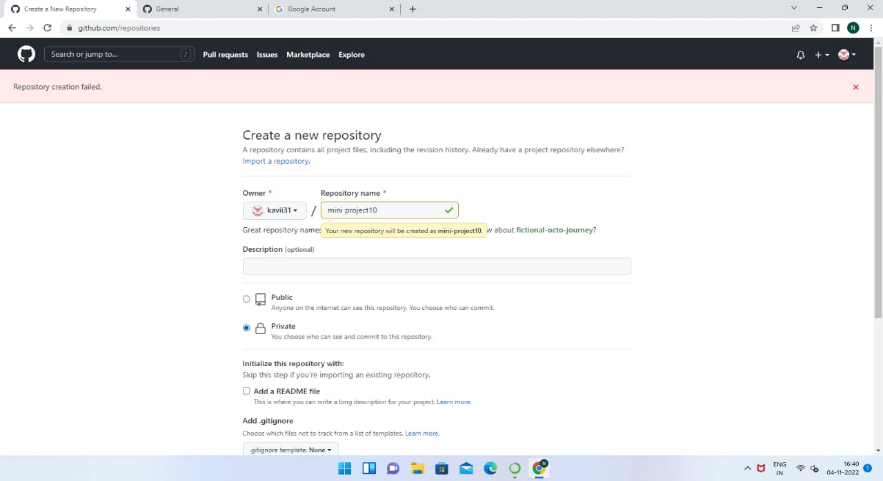
On the top bar, select Main menu Projects and find your project.

On the left sidebar, select Settings General.

In the Project name text box, enter your project name.

In the Project description text box, enter your project description.

Under Project avatar, to change your project avatar, select Choose file

****

**Step 3 :** Choose a repository visibility. For more information, see "[About repositories](https://docs.github.com/en/repositories/creating-and-managing-repositories/about-repositories#about-repository-visibility)."

Git lad allows users with the Owner role to set a project’s or group’s visibility as:

Public

Internal

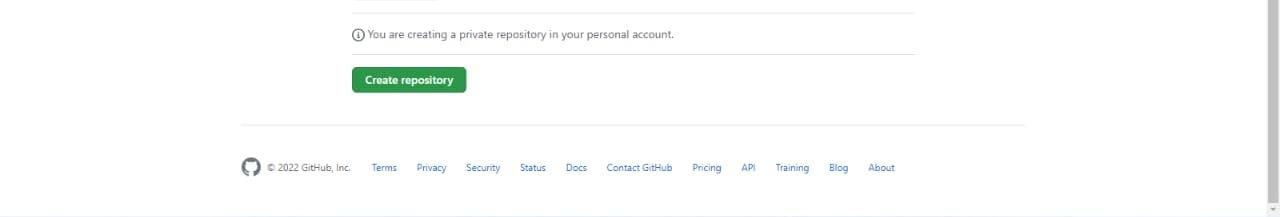
Private

These visibility levels affect who can see the project in the public access directory (/public for your git Lab instance). You can control the visibility of individual features with project feature settings.

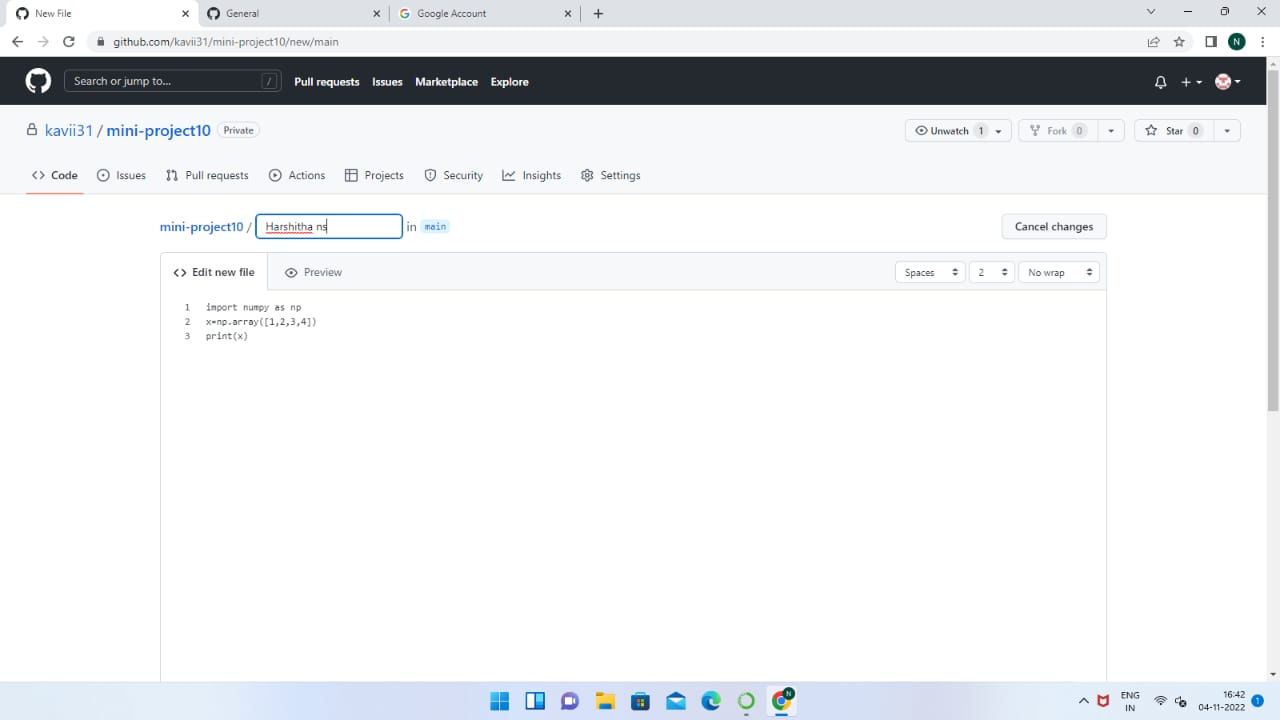
****

**Step 4:** Click **Create repository**.

* In the upper-right corner of any page, use the drop-down menu, and select New repository.
* Type a short, memorable name for your repository
* Optionally, add a description of your repository.
* Choose a repository visibility.
* Select initialize this repository with a README.
* Click Create repository.

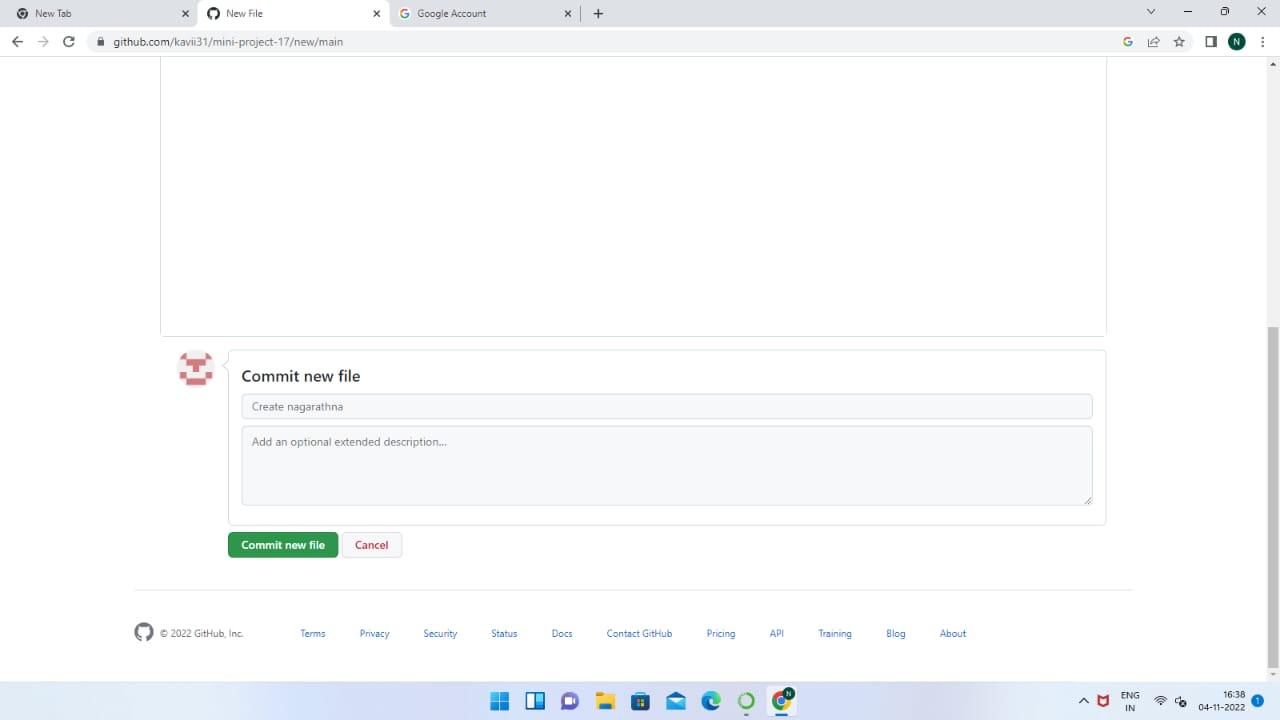
****

**Step5 : S**On the **Edit file** tab, type small one python program.

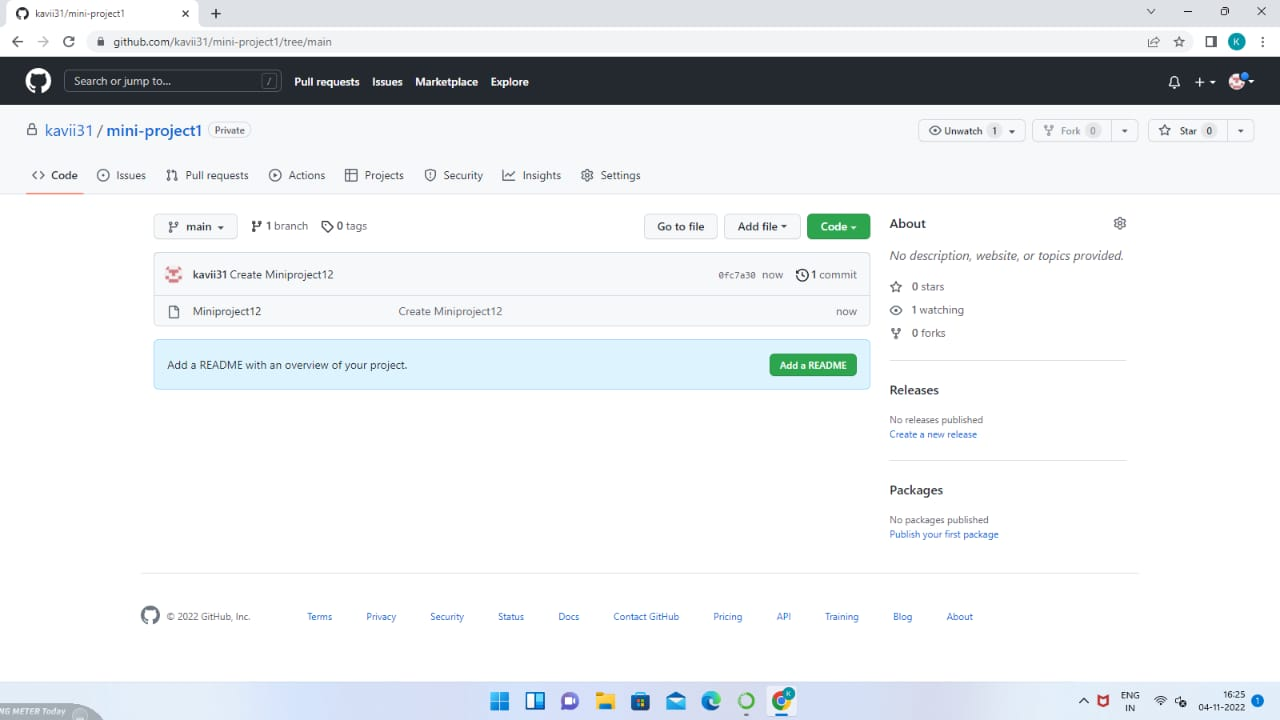


**Step 6:** click commit new file

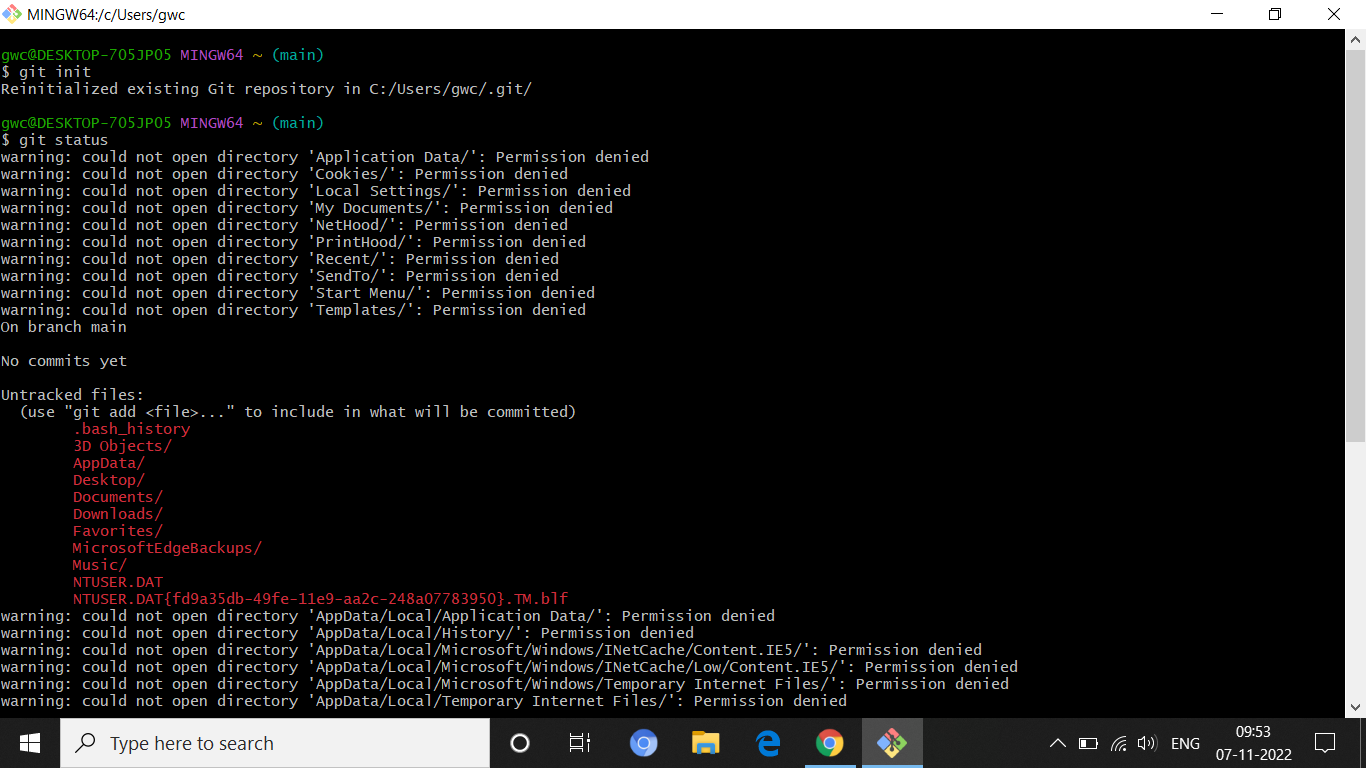
The git commit command will save all staged changes, along with a brief description from the user, in a “commit” to the local repository. Commits are at the heart of Git usage. You can think of a commit as a snapshot of your project, where a new version of that project is created in the current repository.

****

**Step7 :**  get result of this.

s

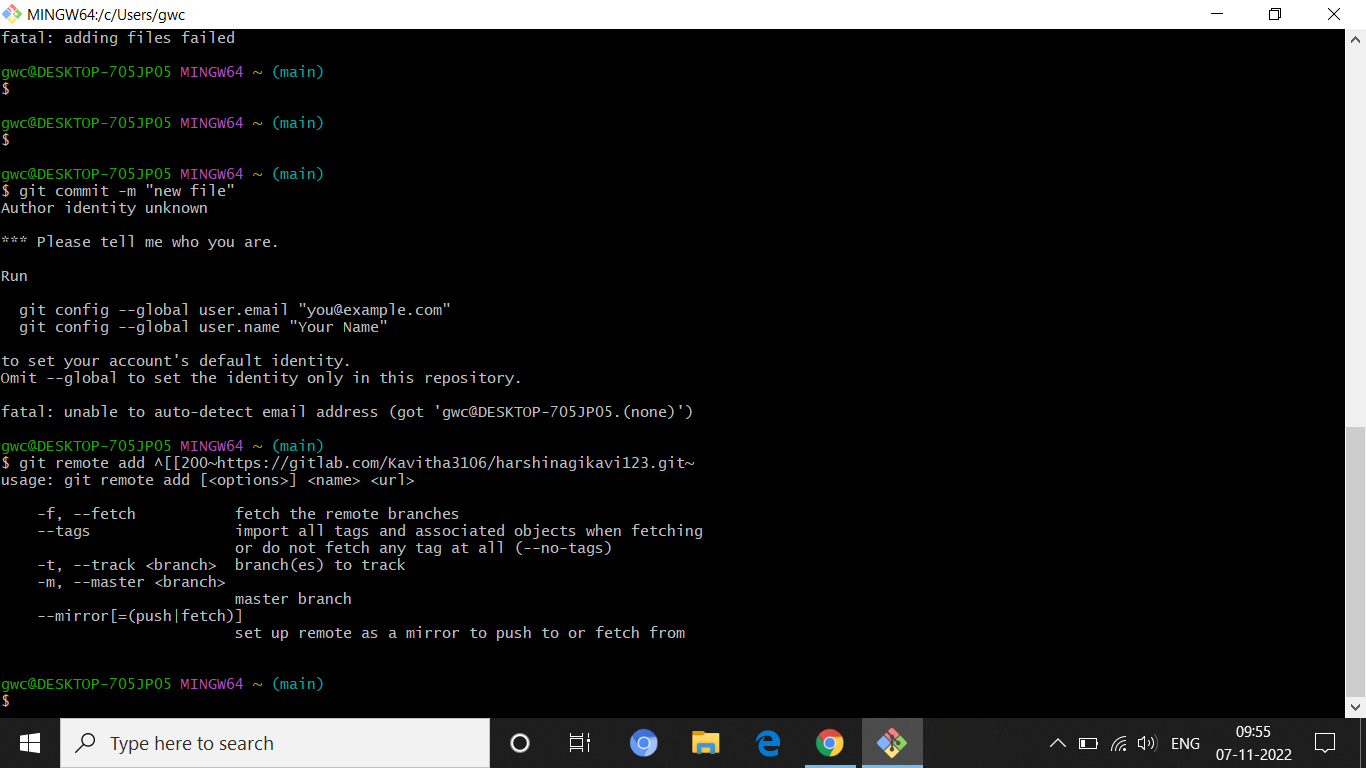
**Step8 : $git init**



**$git commit -m “new file”**

**&**

**$git remote add [git lab htts copy link]**



**$git push -u -f gitlab master**

**&**

**$git pull**

