

INTRODUCTION TO GIT & GITHUB

Presented by: Pratyush Jain IT-68
Prince Rajput IT-69
Raj Nagar IT-70

Guided by: Dr. Lalit Purohit Sir and
Mr. Upendra Singh Sir



Overview

- What is Version Control System ?
- About Git & GitHub
- Difference between Git & GitHub
- Setting up Git
- Configure Git
- Account Creation of GitHub
- Branching, Forking , Merging
- Basic commands of Git
- Upload Project on GitHub



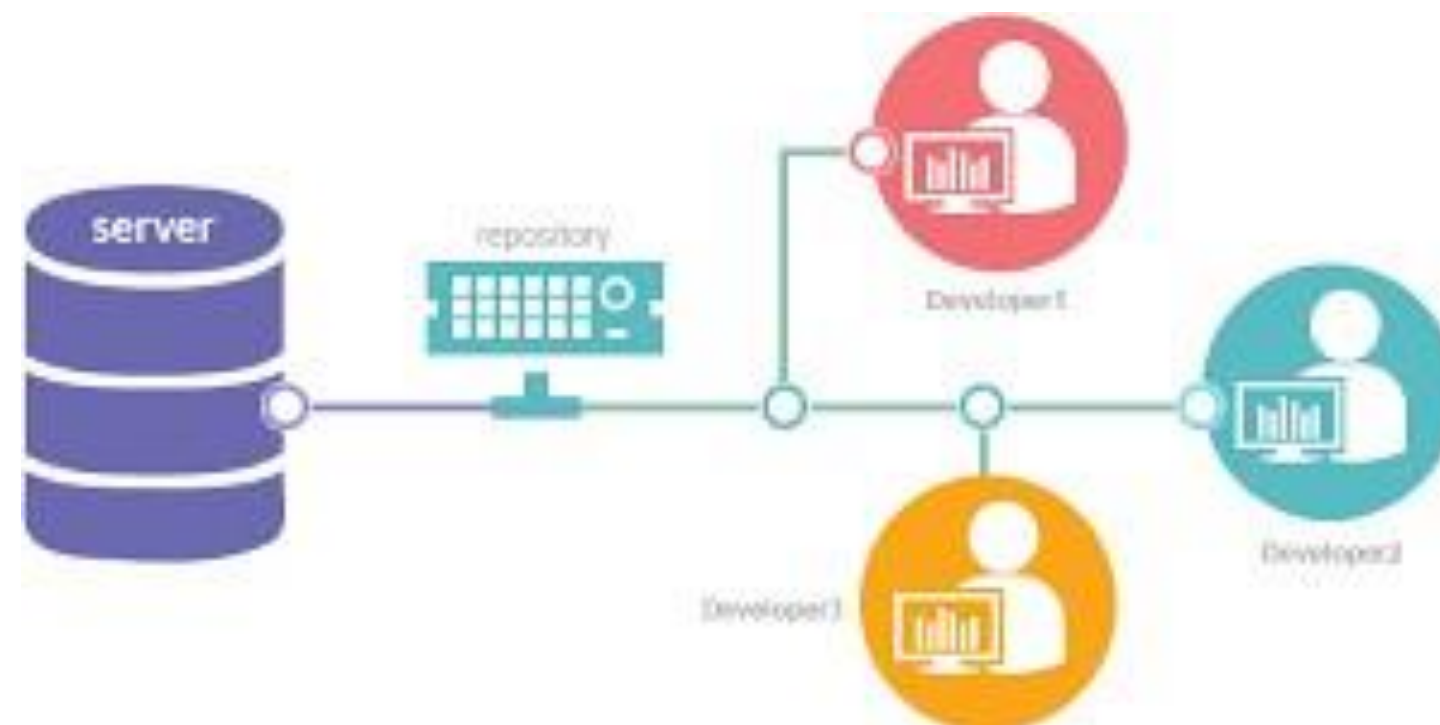
What is Version Control System ?

1. It tracks changes to files and code.
2. Facilitates efficient collaboration for multiple users.
3. Records modifications and maintains a change history.
4. Enables branching and merging for separate development paths.
5. Allows reverting to previous project versions.
6. Ensures structured development and reliable project history.



Git:

1. Git: Tracks changes in computer files.
2. Coordinates work among multiple users.
3. Distributed version control system.
4. Allows simultaneous collaboration on files.
5. Widely used in software development for code collaboration.



Characteristics of Git:

1.Distributed Version Control :

- Git is a distributed version control system, allowing multiple users to work on a project simultaneously.

2. Snapshot-based:

- Git records the entire project at different points in time as snapshots, making it efficient and providing a complete version history.

3.Branching and Merging:

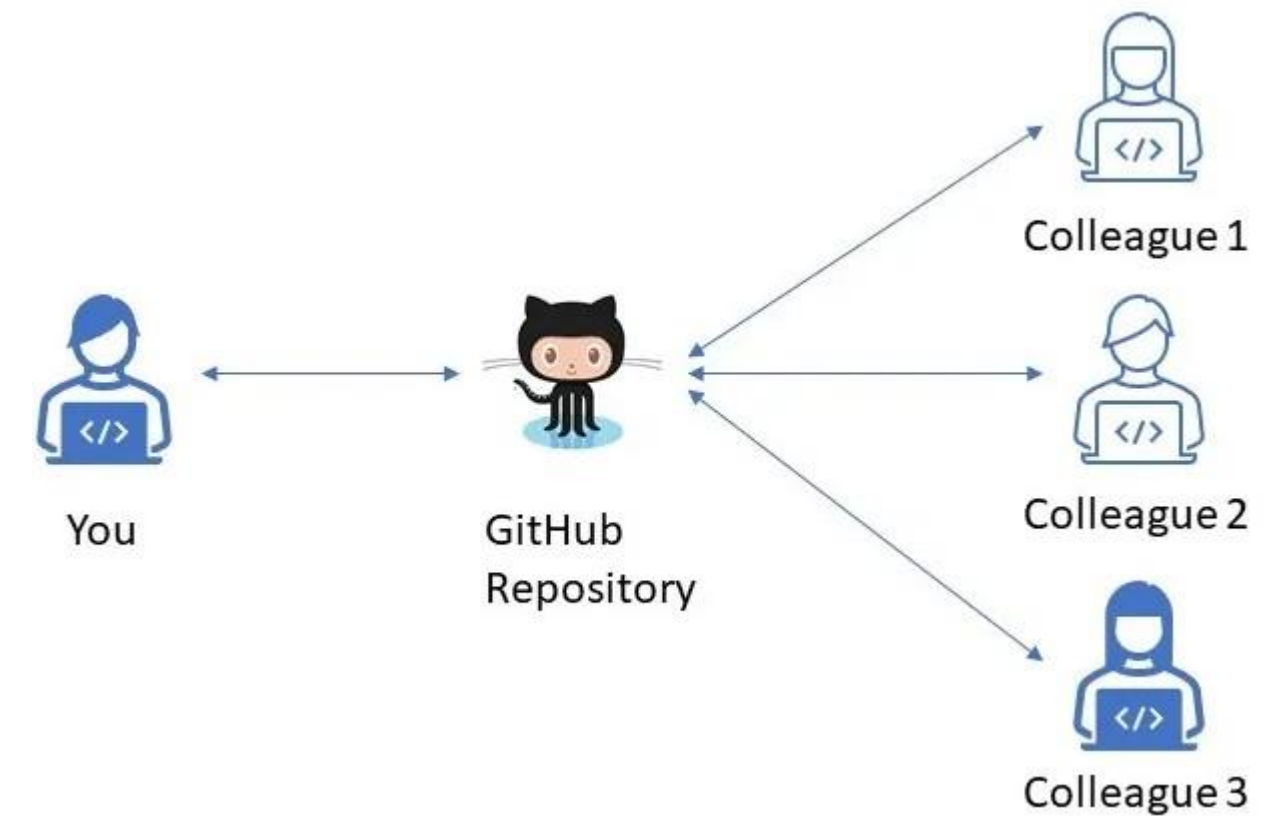
- Git enables easy branching to work on different features or changes independently and allows seamless merging of branches.

4.Local Operations:

- Most Git operations are performed locally, making it fast and efficient, without requiring a constant connection to a central server.

GitHub:

1. GitHub is a web-based platform for collaborative development using Git.
2. It serves as a centralized location for storing, managing, and sharing code.
3. Users create repositories to host projects and track changes through Git version control.
4. Collaboration is facilitated, allowing multiple contributors to work on the same codebase.
5. GitHub provides features like issue tracking, pull requests, and discussions.
6. It is widely used for team-based software development and open-source projects.



Difference

Git

- Distributed version control system.
- Operates on a developer's machine.
- Manages code versions locally.
- No internet dependency.
- No collaboration features
- Command-line tool.

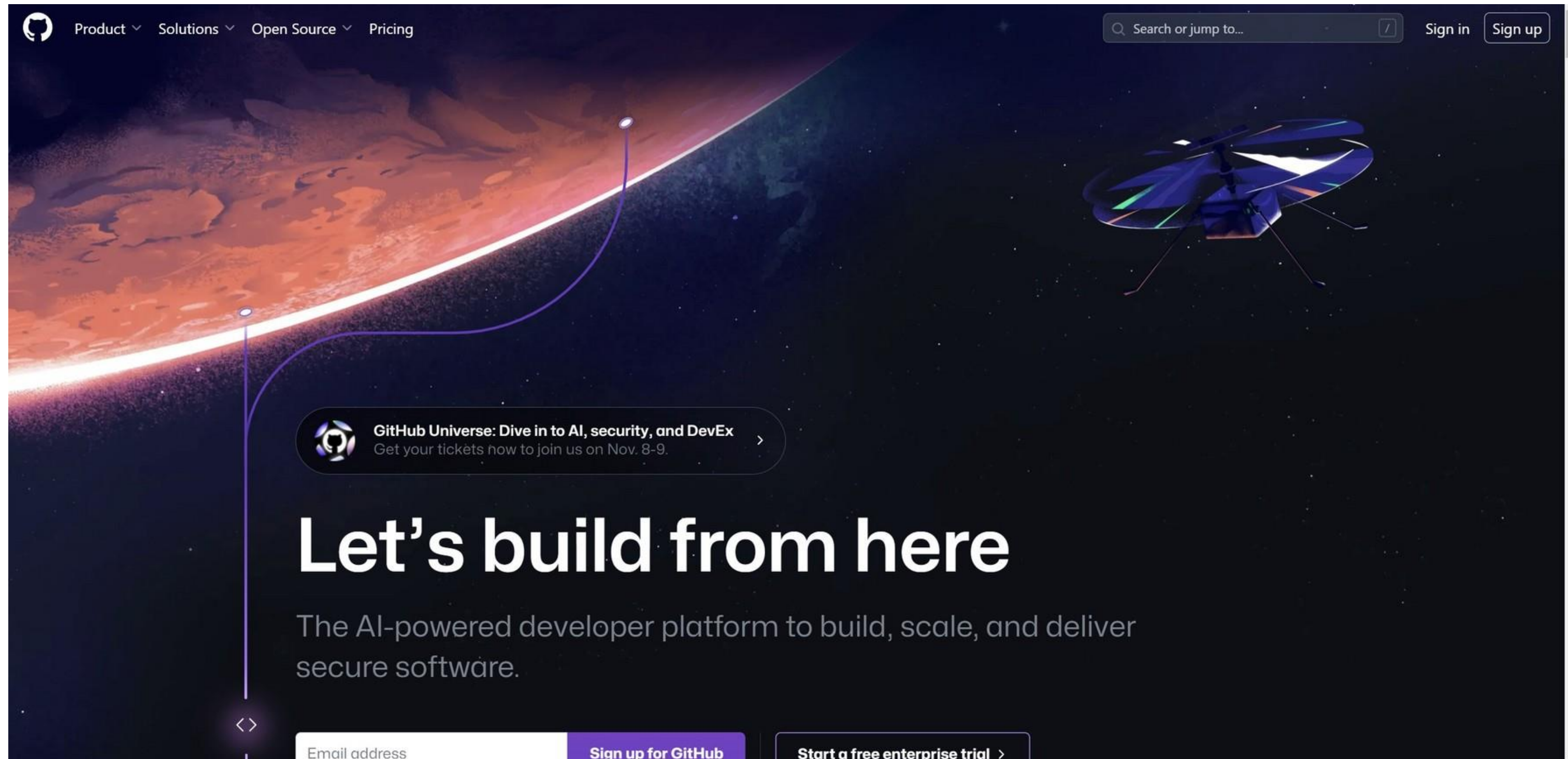
GitHub

- Web-based platform.
- Hosts repositories in the cloud.
- Provides project management features.
- Requires internet access for usage.
- Facilitates remote collaboration.
- Provides a graphical interface.

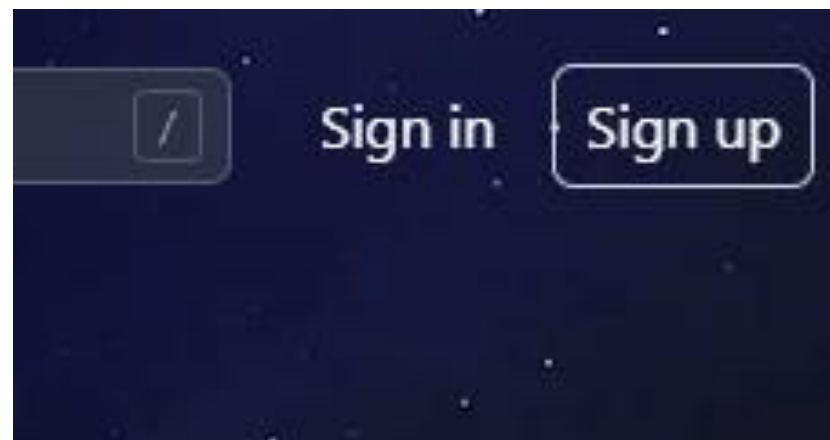


Creating account on GitHub:

1. Open your web browser and go to the GitHub website at <https://github.com>.



2. On the GitHub homepage, you'll see a "Sign up" form. Enter your desired email address and password.



Welcome to GitHub!
Let's begin the adventure

Enter your email*
✓ nrajnagar22@gmail.com

Create a password*
✓

Enter a username*
→ RajNagar22

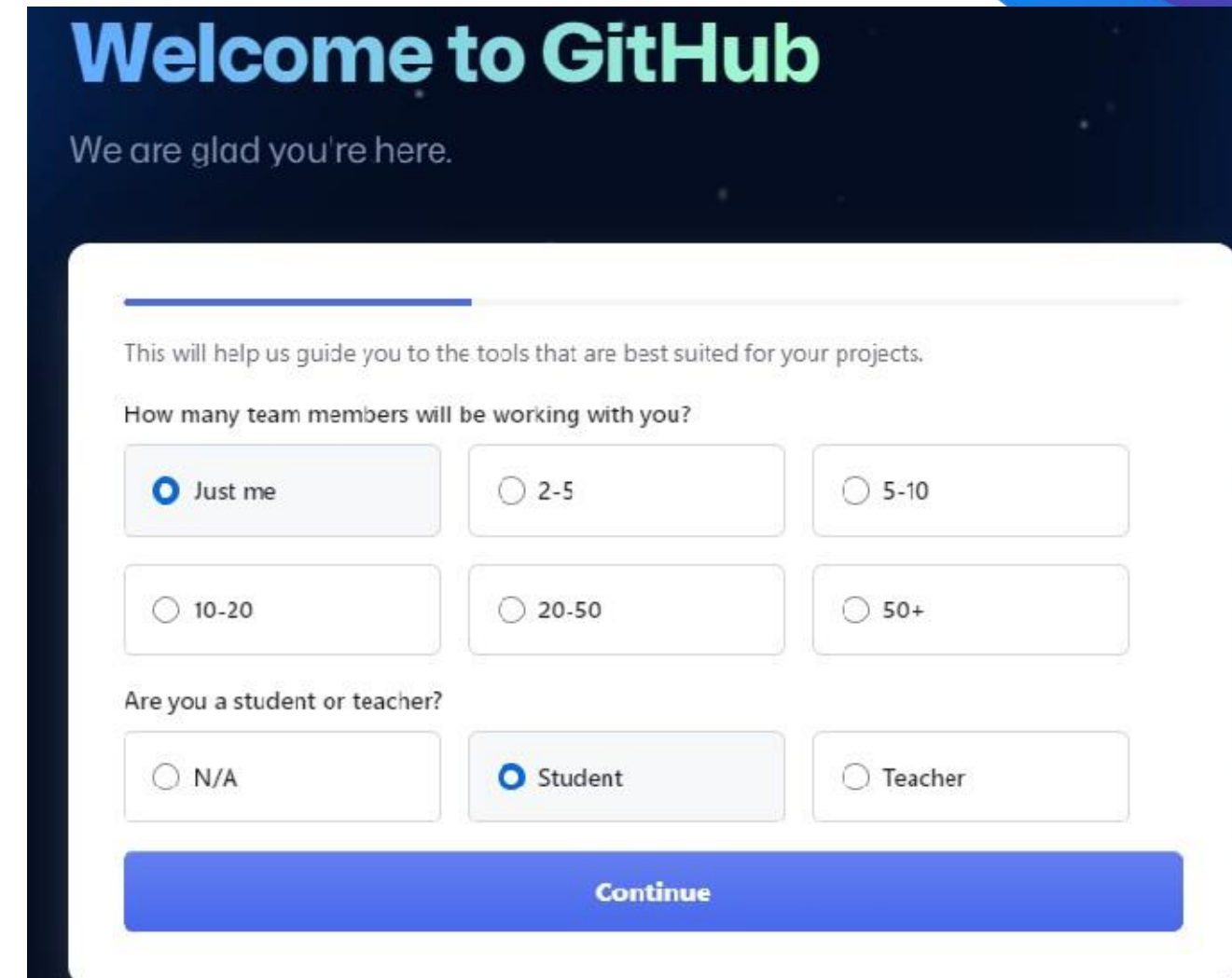
Continue

3. GitHub will send a verification email to the email address you provided. Go to your email inbox, open the email from GitHub, and follow the instructions to verify your email.

4. You'll be prompted to complete the sign-up process.

Optionally, you can:

- Customize your experience by selecting your interests.
- Choose to receive updates and newsletters from GitHub.



The screenshot shows the GitHub 'Welcome to GitHub' screen. At the top, it says 'Welcome to GitHub' in a large, bold, teal font, with 'We are glad you're here.' in a smaller, grey font below it. A progress bar is visible, showing the first step is complete. The main content area is titled 'This will help us guide you to the tools that are best suited for your projects.' and contains two sections of radio button options. The first section is 'How many team members will be working with you?' with options: 'Just me' (selected), '2-5', '5-10', '10-20', '20-50', and '50+'. The second section is 'Are you a student or teacher?' with options: 'N/A', 'Student' (selected), and 'Teacher'. A blue 'Continue' button is at the bottom.

Welcome to GitHub
We are glad you're here.

This will help us guide you to the tools that are best suited for your projects.

How many team members will be working with you?

☒ Just me ☐ 2-5 ☐ 5-10

☐ 10-20 ☐ 20-50 ☐ 50+

Are you a student or teacher?

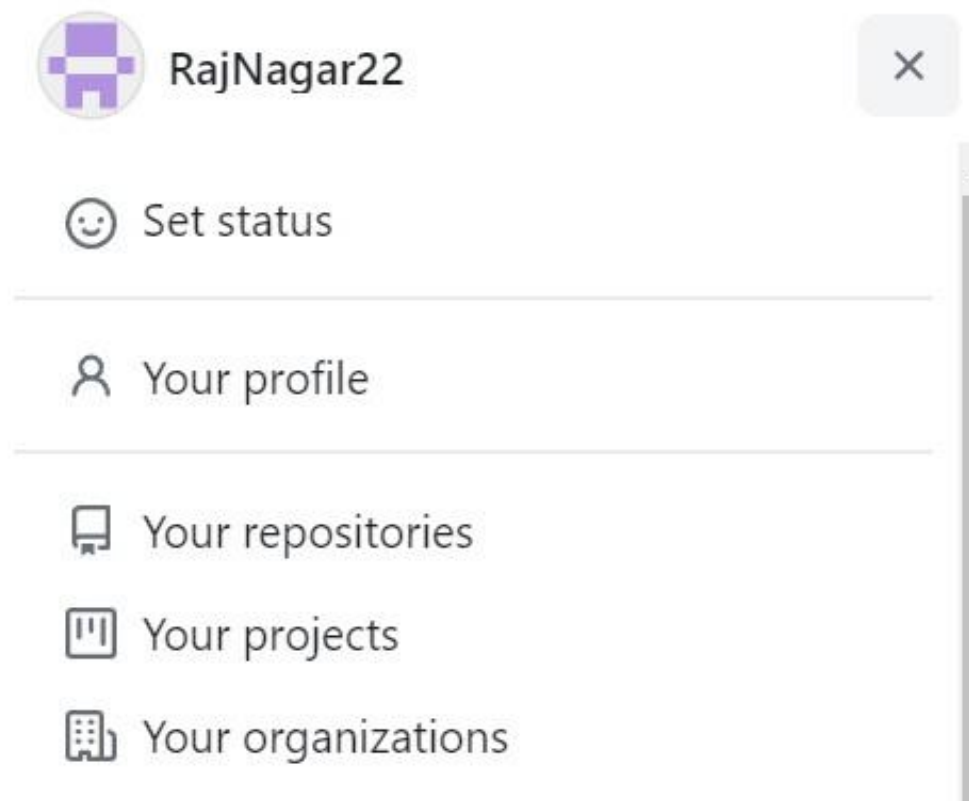
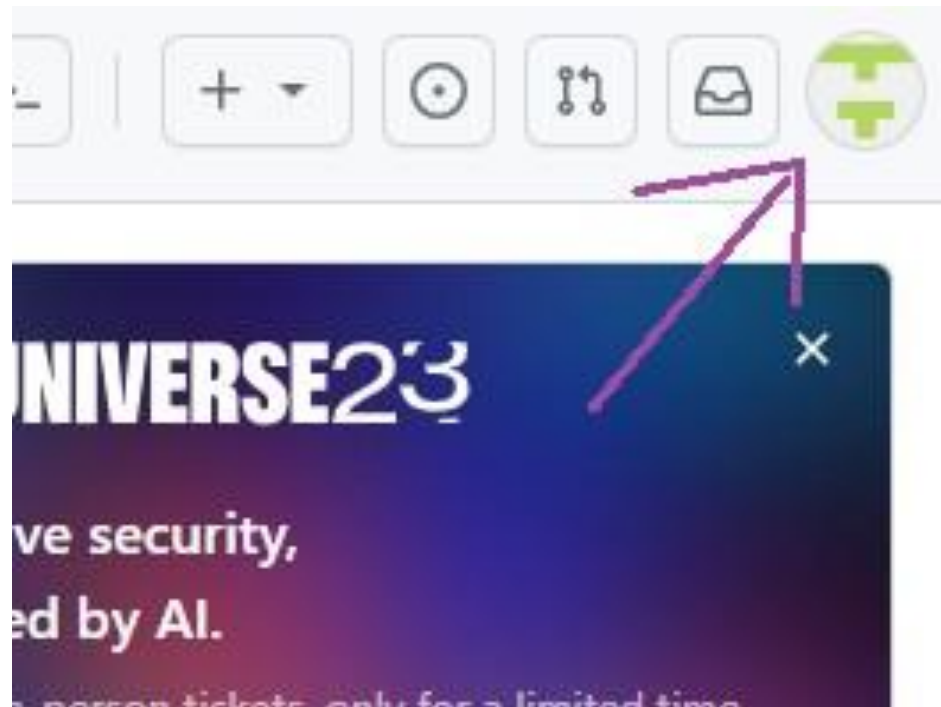
☐ N/A ☒ Student ☐ Teacher

Continue

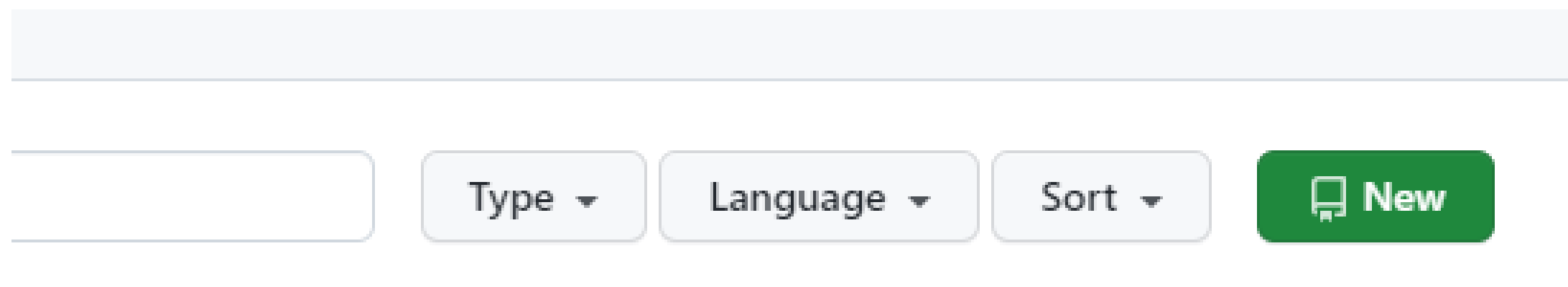
5. Account creation is completed, personalize your account as per your needs.

Creating New Repository in GitHub:

1. Login to your GitHub and Click on the top right corner icon.
2. From the appeared dropdown list select “Your Repositories”.



3. Now select “New” button



4. A new form will appear, type all the required details of the new repository in the form and click on “Create Repository” Button. Then the new repository will be created on GitHub.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *

 RajNagar22 ▾

Repository name *

DemoRepo

✔ DemoRepo is available.

Great repository names are short and memorable. Need inspiration? How about [expert-spoon](#) ?

Description (optional)

this is a demo repository



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:



Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set  main as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.

Create repository

5. Newly created Repository will appear like this :

The screenshot displays the GitHub interface for a newly created repository named "DemoRepo" by the user "RajNagar22". The repository is public and contains a single file, "README.md", which was committed yesterday. The repository has 1 branch (main) and 0 tags. The README content includes the repository name "DemoRepo" and the description "this is a demo repository". The right sidebar shows repository statistics: 0 stars, 1 watching, and 0 forks. The footer of the page includes the GitHub logo, copyright information for 2023, and various links such as Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About.

Repository Header:

- Repo name: DemoRepo (Public)
- Actions: Pin, Unwatch (1), Fork (0), Star (0)

Repository Content:

- Branch: main (1 branch), Tags: 0 tags
- Commit: RajNagar22 Initial commit (c73670d yesterday, 1 commit)
- File: README.md (Initial commit, yesterday)
- README content: DemoRepo, this is a demo repository

Repository Statistics:

- About: this is a demo repository
- Readme
- Activity
- 0 stars
- 1 watching
- 0 forks

Releases:

- No releases published
- [Create a new release](#)

Packages:

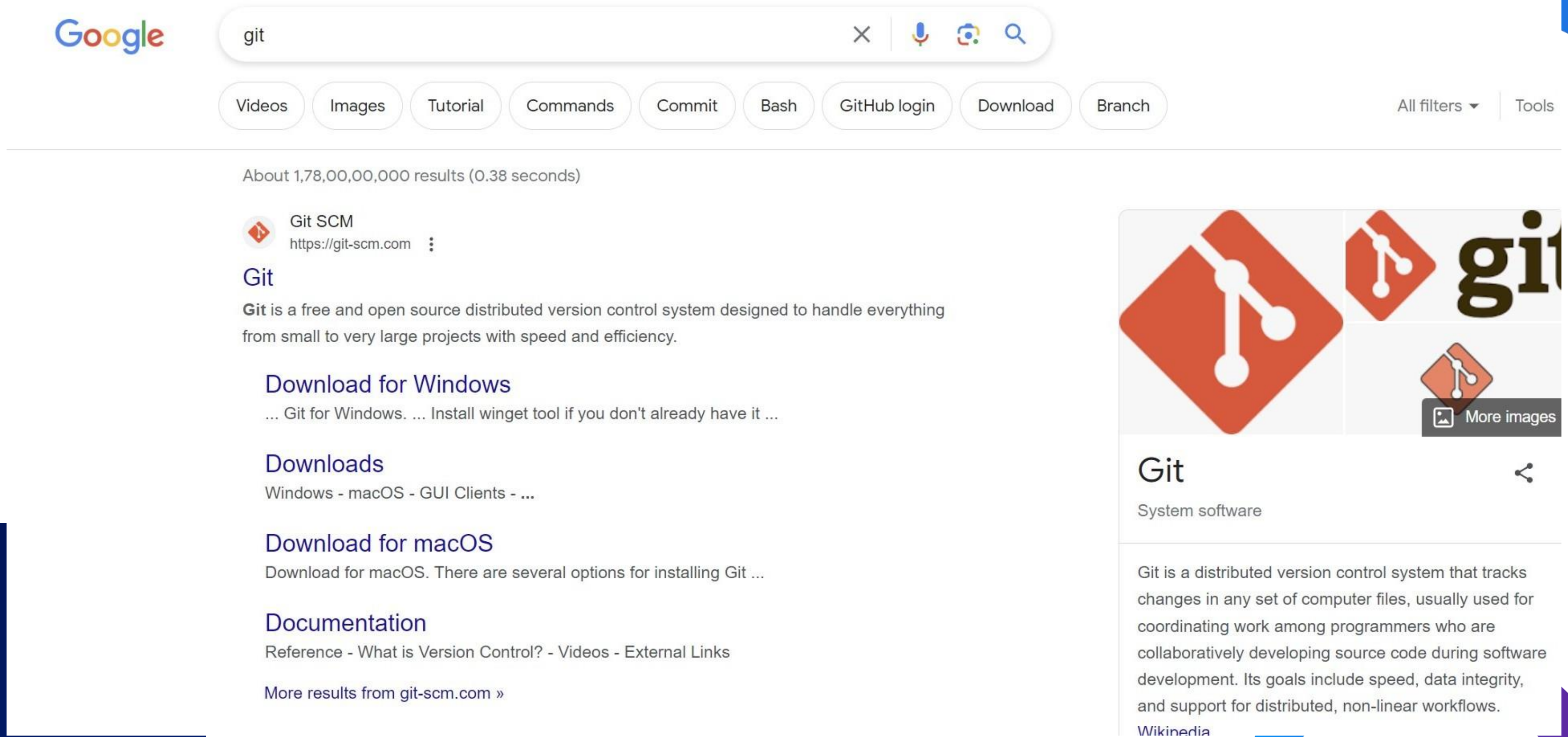
- No packages published
- [Publish your first package](#)

Footer:

- © 2023 GitHub, Inc.
- [Terms](#), [Privacy](#), [Security](#), [Status](#), [Docs](#), [Contact GitHub](#), [Pricing](#), [API](#), [Training](#), [Blog](#), [About](#)

Setting Up Git:

1.Type git in your web browser's search bar.



The image shows a Google search interface with the query 'git'. The search bar is at the top, with the Google logo on the left and search icons on the right. Below the search bar are filter buttons: Videos, Images, Tutorial, Commands, Commit, Bash, GitHub login, Download, and Branch. To the right of these buttons are 'All filters' and 'Tools' links. The search results show 'About 1,78,00,00,000 results (0.38 seconds)'. The first result is 'Git SCM' with the URL 'https://git-scm.com'. Below this is a section titled 'Git' with a description: 'Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.' There are three sub-sections: 'Download for Windows' with a link to 'Git for Windows', 'Downloads' with a link to 'Windows - macOS - GUI Clients', and 'Download for macOS' with a link to 'Download for macOS'. At the bottom of the results is a link to 'Documentation' with a link to 'Reference - What is Version Control?'. To the right of the search results is a sidebar with a large Git logo, the text 'git', and a 'More images' button. Below the logo is the title 'Git' and the subtitle 'System software'. At the bottom of the sidebar is a paragraph of text: 'Git is a distributed version control system that tracks changes in any set of computer files, usually used for coordinating work among programmers who are collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows.' and a link to 'Wikipedia'.


Google

git

Videos Images Tutorial Commands Commit Bash GitHub login Download Branch

All filters Tools

About 1,78,00,00,000 results (0.38 seconds)

 Git SCM
https://git-scm.com

Git

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.



[Download for Windows](#)
... Git for Windows. ... Install winget tool if you don't already have it ...

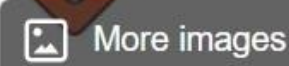
[Downloads](#)
Windows - macOS - GUI Clients - ...

[Download for macOS](#)
Download for macOS. There are several options for installing Git ...

[Documentation](#)
Reference - What is Version Control? - Videos - External Links

[More results from git-scm.com »](#)

  git

 More images

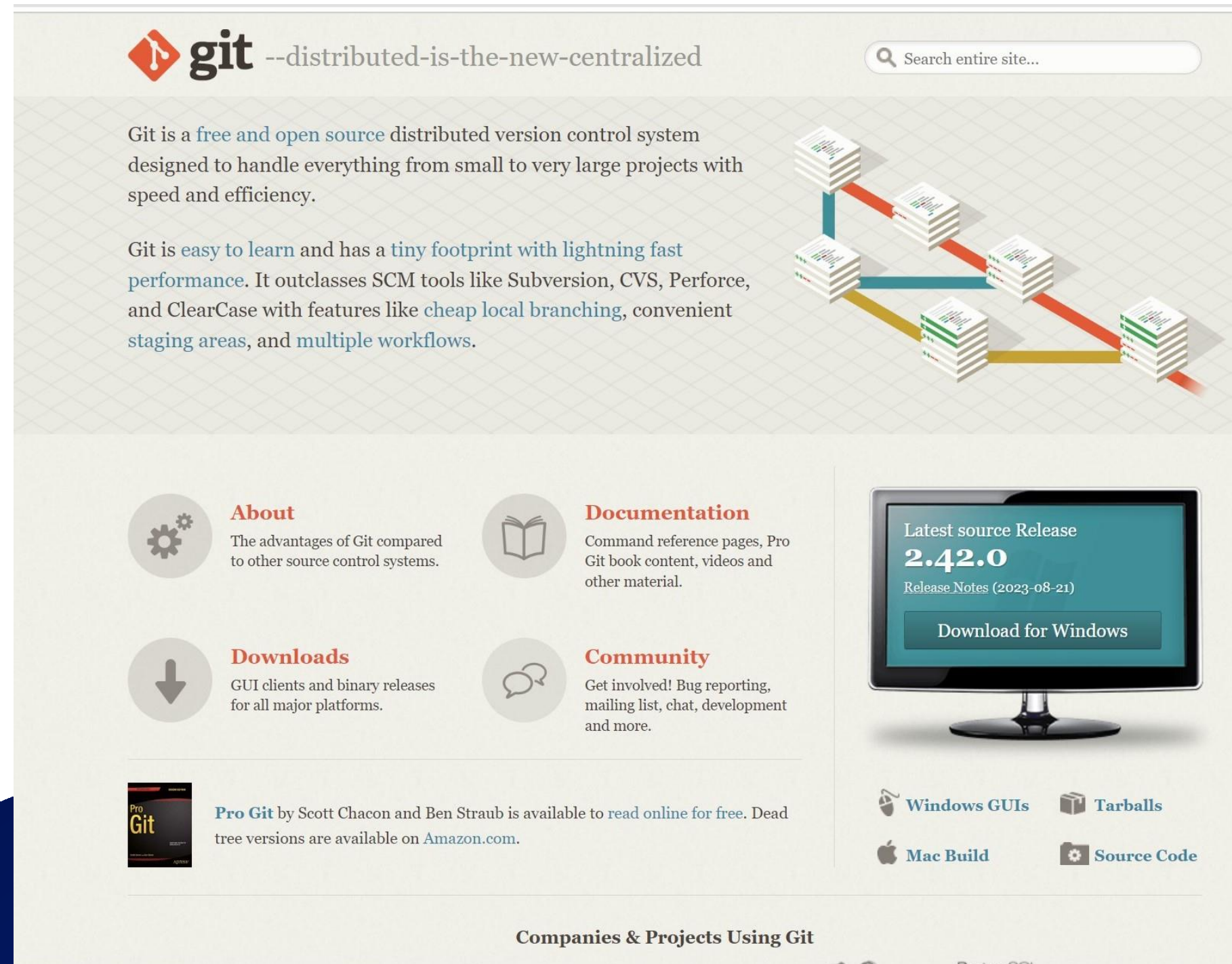
Git

System software

Git is a distributed version control system that tracks changes in any set of computer files, usually used for coordinating work among programmers who are collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows.

[Wikipedia](#)

2. If you're a windows user click download for windows, else click on mac build.



The screenshot shows the Git website homepage. At the top left is the Git logo with the tagline "--distributed-is-the-new-centralized". A search bar is on the top right. The main content area describes Git as a free and open source distributed version control system. Below this, there are four sections: "About" (advantages of Git), "Documentation" (command reference, Pro Git book), "Downloads" (GUI clients, binary releases), and "Community" (bug reporting, mailing list). A central monitor graphic displays the latest source release "2.42.0" and a "Download for Windows" button. At the bottom, there are links for "Windows GUIs", "Tarballs", "Mac Build", and "Source Code".

git --distributed-is-the-new-centralized

Search entire site...

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a **tiny footprint** with **lightning fast** performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient staging areas, and **multiple workflows**.

About
The advantages of Git compared to other source control systems.

Documentation
Command reference pages, Pro Git book content, videos and other material.

Downloads
GUI clients and binary releases for all major platforms.

Community
Get involved! Bug reporting, mailing list, chat, development and more.

Latest source Release
2.42.0
[Release Notes \(2023-08-21\)](#)
[Download for Windows](#)

Pro Git by Scott Chacon and Ben Straub is available to [read online](#) for free. Dead tree versions are available on [Amazon.com](#).

[Windows GUIs](#) [Tarballs](#)
[Mac Build](#) [Source Code](#)

Companies & Projects Using Git

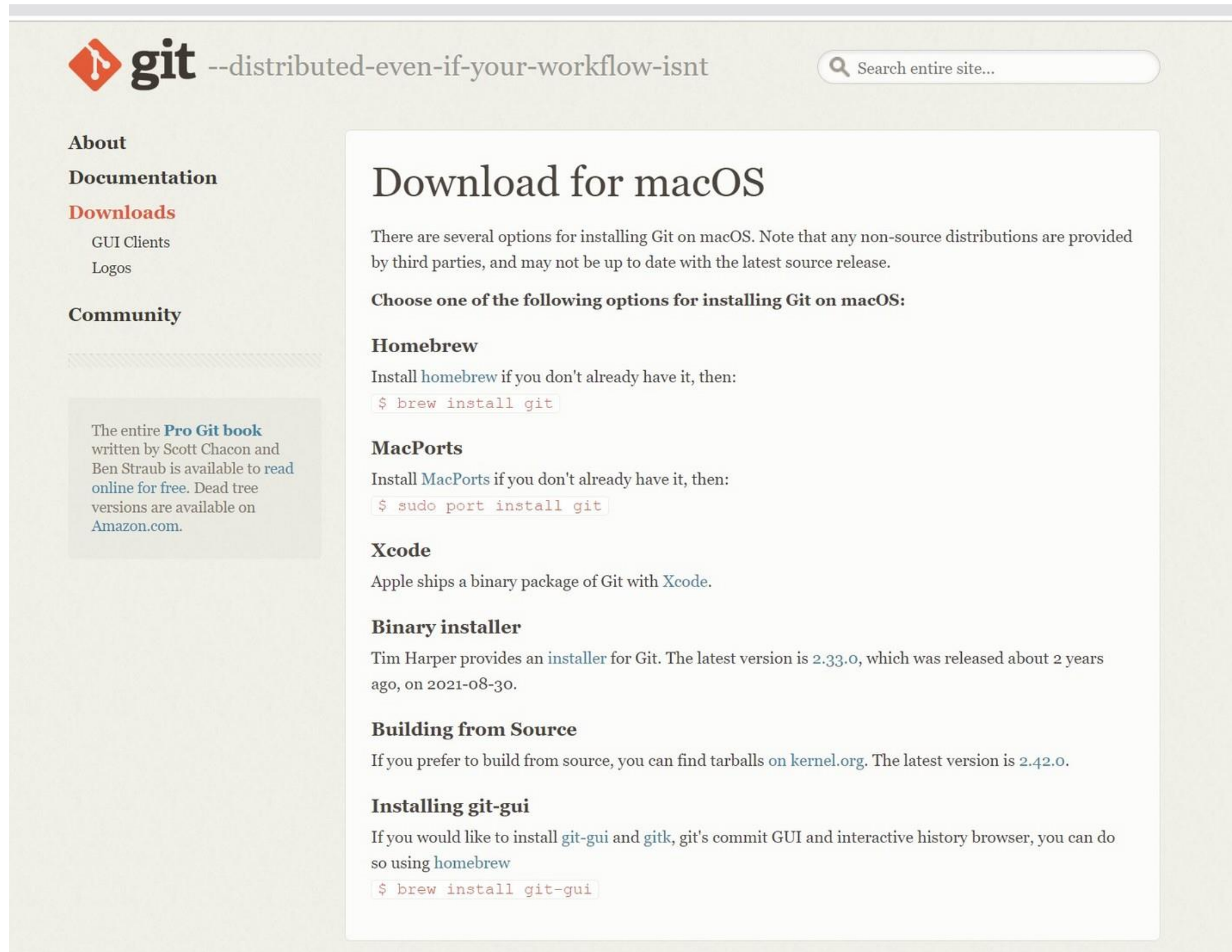


This is a close-up of the download section of the Git website. It features a large monitor graphic displaying the latest source release "2.42.0" and a "Download for Windows" button. Below the monitor, there are four links: "Windows GUIs", "Tarballs", "Mac Build", and "Source Code".

Latest source Release
2.42.0
[Release Notes \(2023-08-21\)](#)
[Download for Windows](#)

[Windows GUIs](#) [Tarballs](#)
[Mac Build](#) [Source Code](#)

3. Mac users should follow the given procedure:



The screenshot shows the Git website's 'Download for macOS' page. The header features the Git logo and the tagline '--distributed-even-if-your-workflow-isnt', along with a search bar. The left sidebar contains navigation links for 'About', 'Documentation', 'Downloads' (highlighted), 'GUI Clients', 'Logos', and 'Community'. A box in the sidebar promotes the 'Pro Git book'. The main content area is titled 'Download for macOS' and provides instructions for installing Git on macOS. It lists several options: Homebrew, MacPorts, Xcode, Binary installer, Building from Source, and Installing git-gui, each with a brief description and a terminal command snippet.

git --distributed-even-if-your-workflow-isnt

Search entire site...

About

Documentation

Downloads

GUI Clients

Logos

Community

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to read [online for free](#). Dead tree versions are available on [Amazon.com](#).

Download for macOS

There are several options for installing Git on macOS. Note that any non-source distributions are provided by third parties, and may not be up to date with the latest source release.

Choose one of the following options for installing Git on macOS:

Homebrew

Install [homebrew](#) if you don't already have it, then:

```
$ brew install git
```

MacPorts

Install [MacPorts](#) if you don't already have it, then:

```
$ sudo port install git
```

Xcode

Apple ships a binary package of Git with [Xcode](#).

Binary installer

Tim Harper provides an [installer](#) for Git. The latest version is [2.33.0](#), which was released about 2 years ago, on 2021-08-30.

Building from Source


If you prefer to build from source, you can find tarballs [on kernel.org](#). The latest version is [2.42.0](#).

Installing git-gui

If you would like to install [git-gui](#) and [gitk](#), git's commit GUI and interactive history browser, you can do so using [homebrew](#)

```
$ brew install git-gui
```


4. Click on click here to download.

 **git** --distributed-even-if-your-workflow-isnt

Search entire site...

About
Documentation
Downloads
GUI Clients
Logos
Community

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on [Amazon.com](#).

Download for Windows

[Click here to download](#) the latest (**2.42.0**) **64-bit** version of **Git for Windows**. This is the most recent [maintained build](#). It was released **20 days ago**, on 2023-08-30.

Other Git for Windows downloads

Standalone Installer
[32-bit Git for Windows Setup.](#)
[64-bit Git for Windows Setup.](#)

Portable ("thumbdrive edition")
[32-bit Git for Windows Portable.](#)
[64-bit Git for Windows Portable.](#)

Using winget tool




Install [winget tool](#) if you don't already have it, then type this command in command prompt or Powershell.

```
winget install --id Git.Git -e --source winget
```

The current source code release is version **2.42.0**. If you want the newer version, you can build it from the [source code](#).

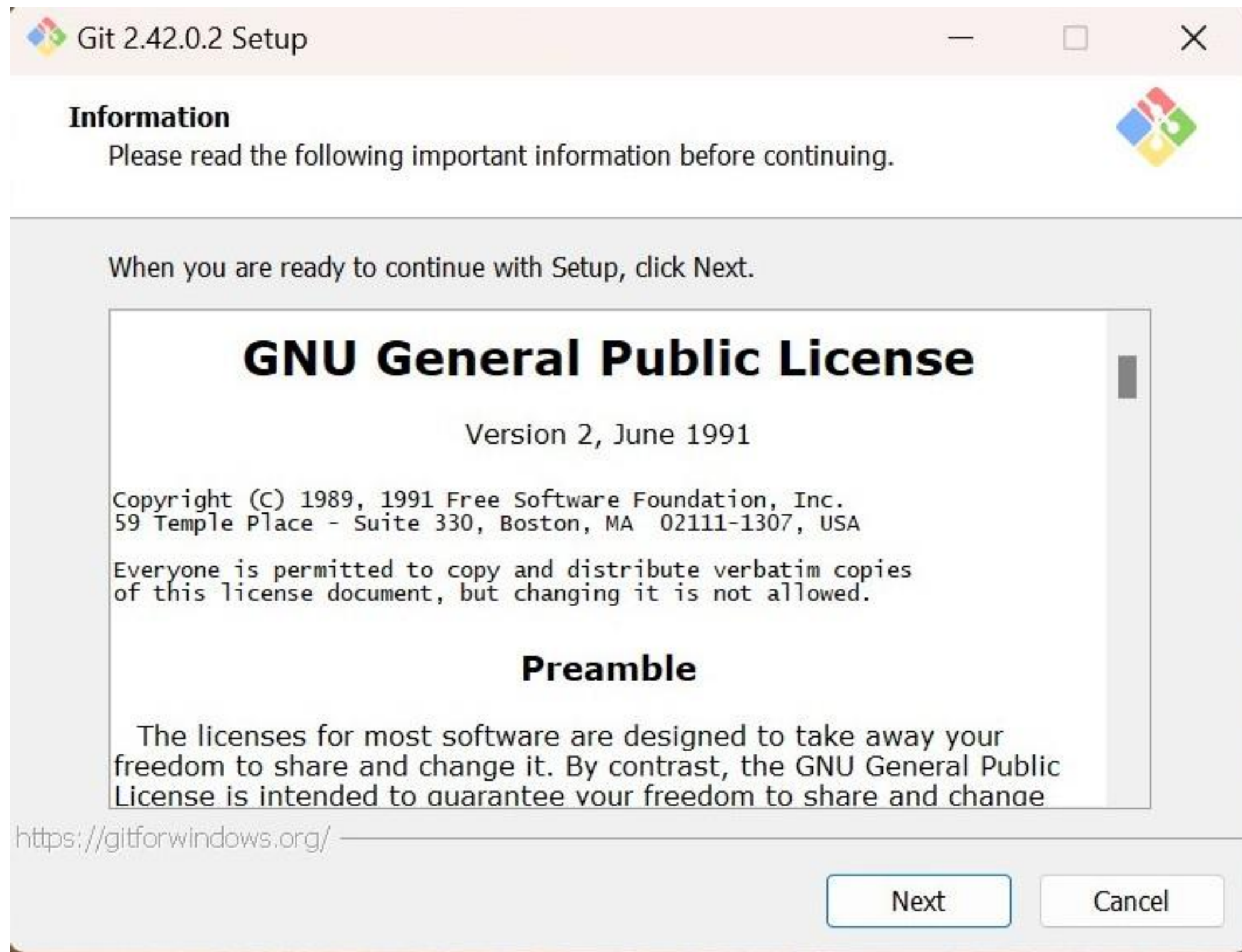
Now What?

Now that you have downloaded Git, it's time to start using it.

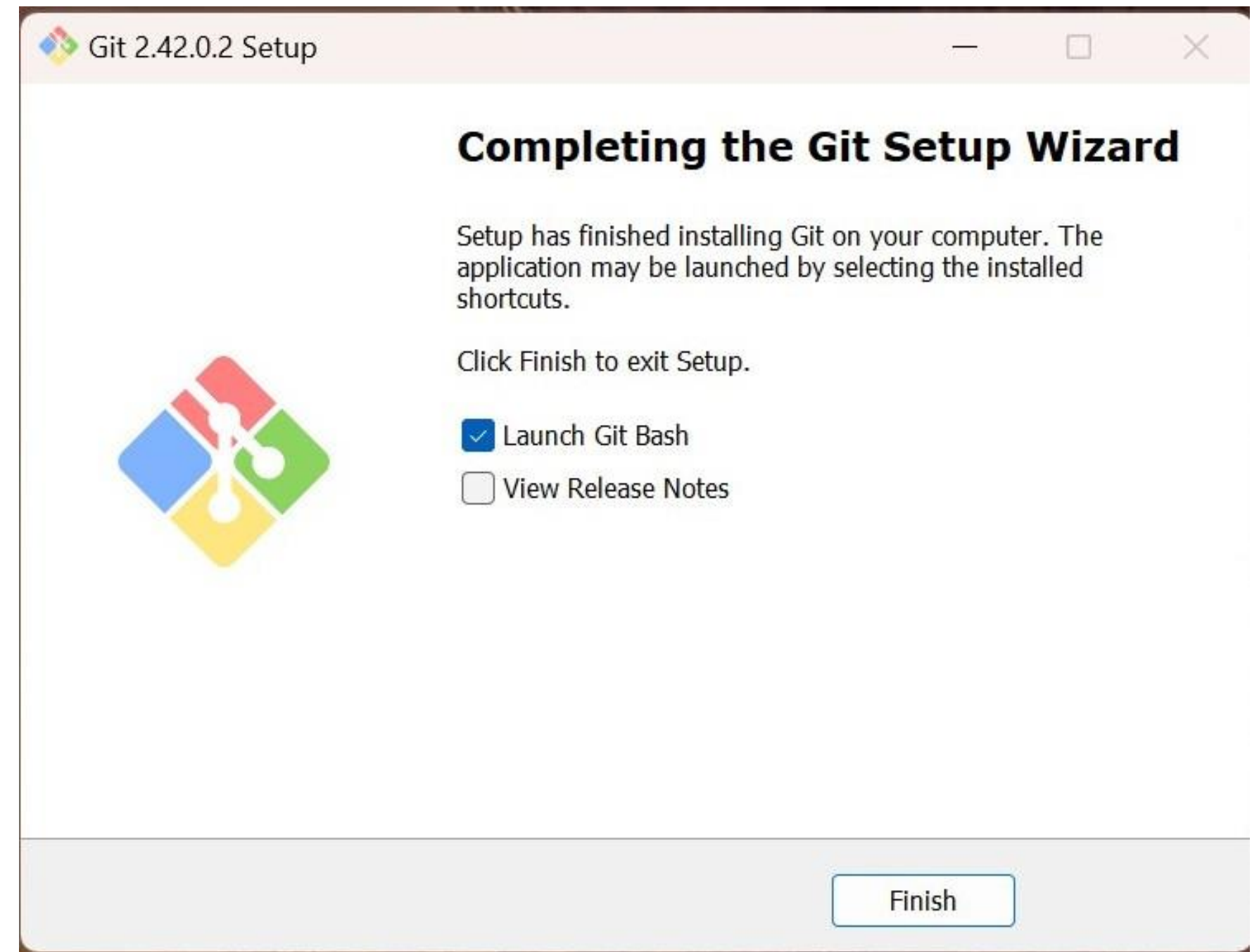
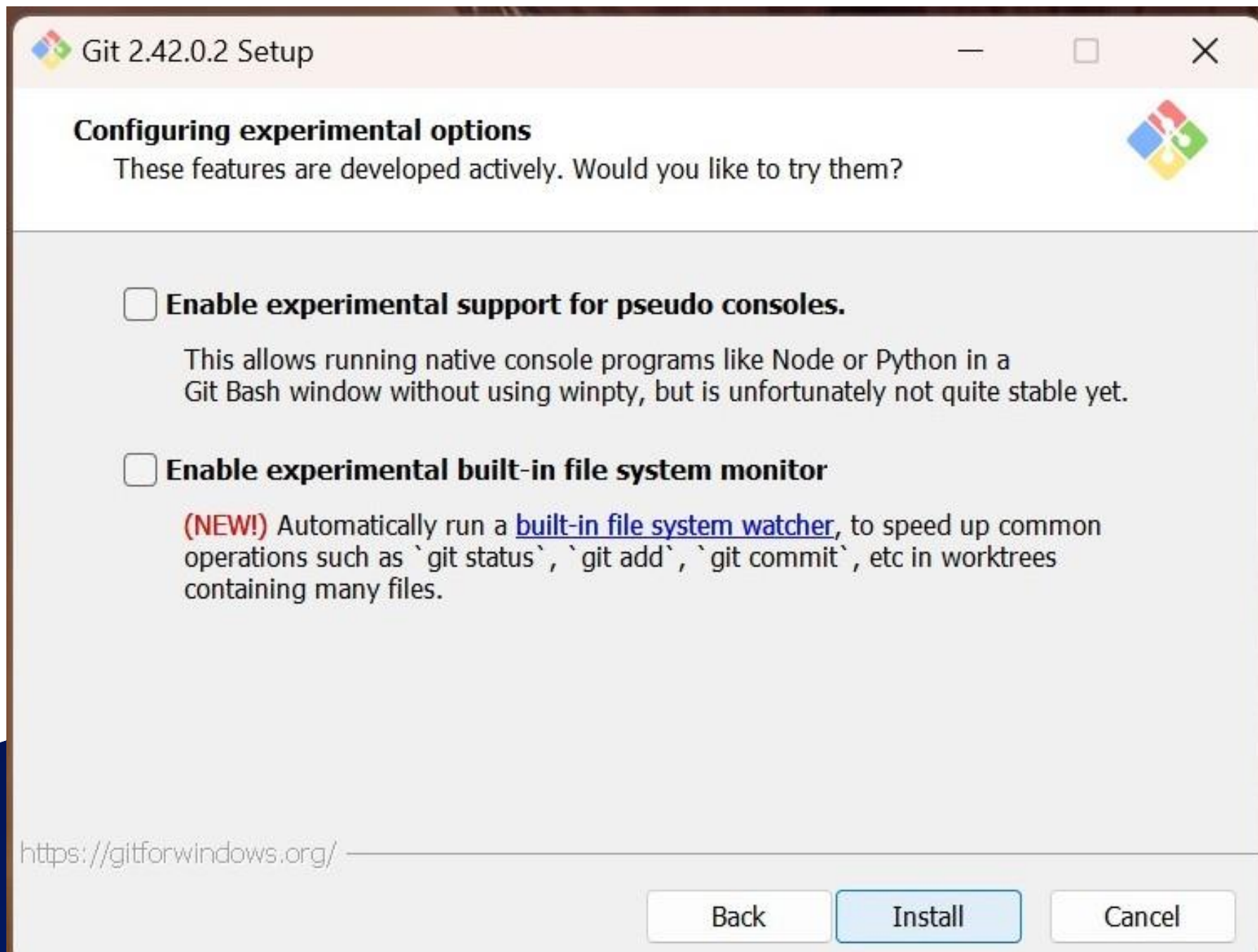


l.windows.2/Git-2.42.0.2-64-bit.exe

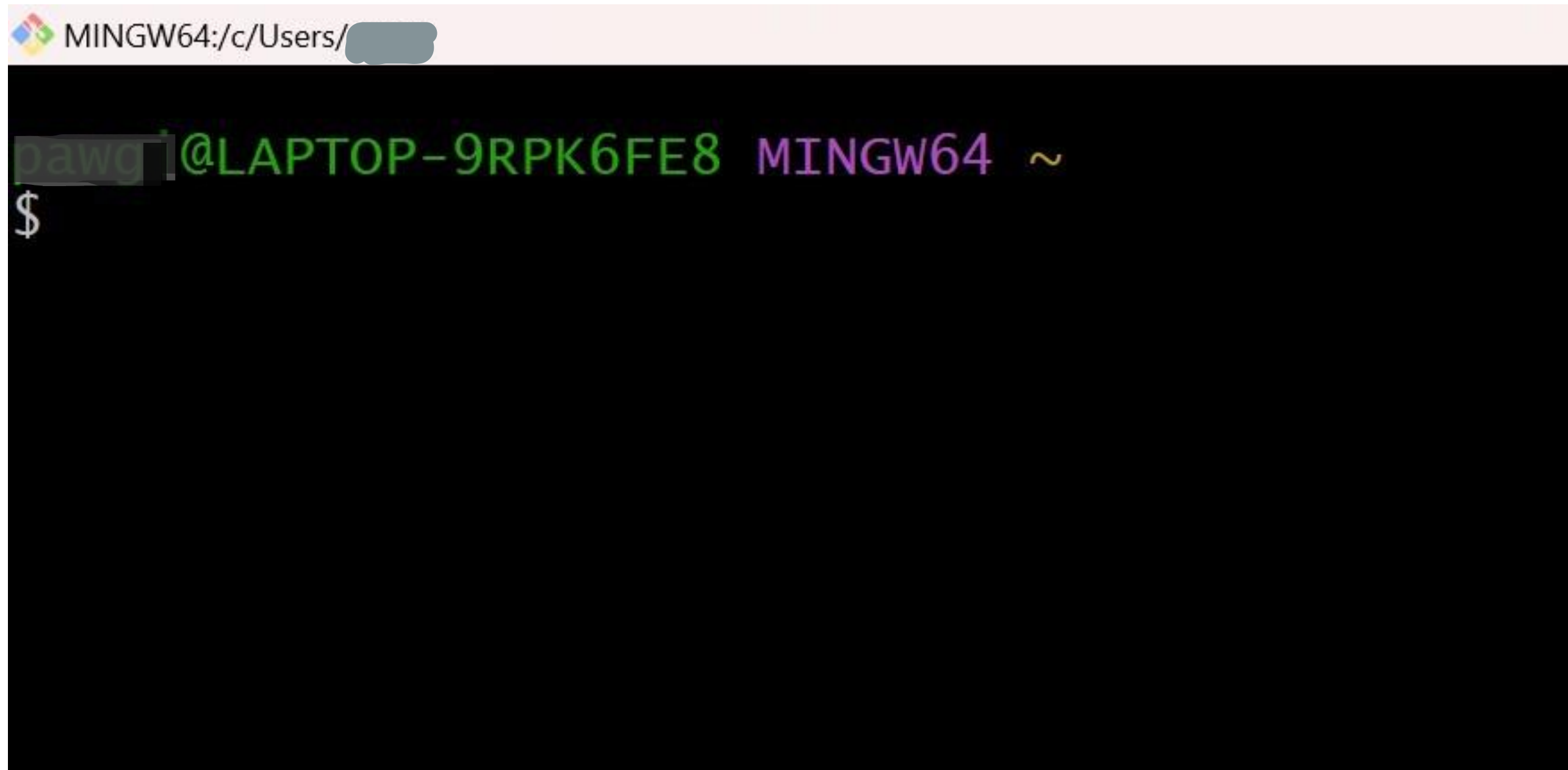
5. After the download is complete, open the downloaded file and the following setup dialog box will appear.



6. Keep clicking "next" until the following dialog box appears. Now click "install" and git would be installed. Clicking finish would launch Git Bash.



7. This is how the git bash looks like.




The image shows a screenshot of a Windows terminal window. The title bar at the top is light pink and contains the text "MINGW64:/c/Users/[redacted]". The terminal itself has a black background. The first line of text is the prompt "pawg@LAPTOP-9RPK6FE8 MINGW64 ~" in a green monospace font. Below this, on the next line, is a white dollar sign "\$" indicating the command prompt.



What is Git Bash and why are we using ?

Git Bash is a Windows application that emulates a Linux-like command line interface, enabling Git usage. It installs Bash, common utilities, and Git, providing a familiar environment for Windows users to work with Git as it resembles Linux and macOS terminal experiences.



Configure Git:

- git config command is used for configuring our username and email address. So for configuration, we can type the following command:

 MINGW64:/c/Users/RES

```
RES@DESKTOP-ESAPJPD MINGW64 ~  
$ git config --global user.name
```

```
RES@DESKTOP-ESAPJPD MINGW64 ~  
$ git config --global user.email
```

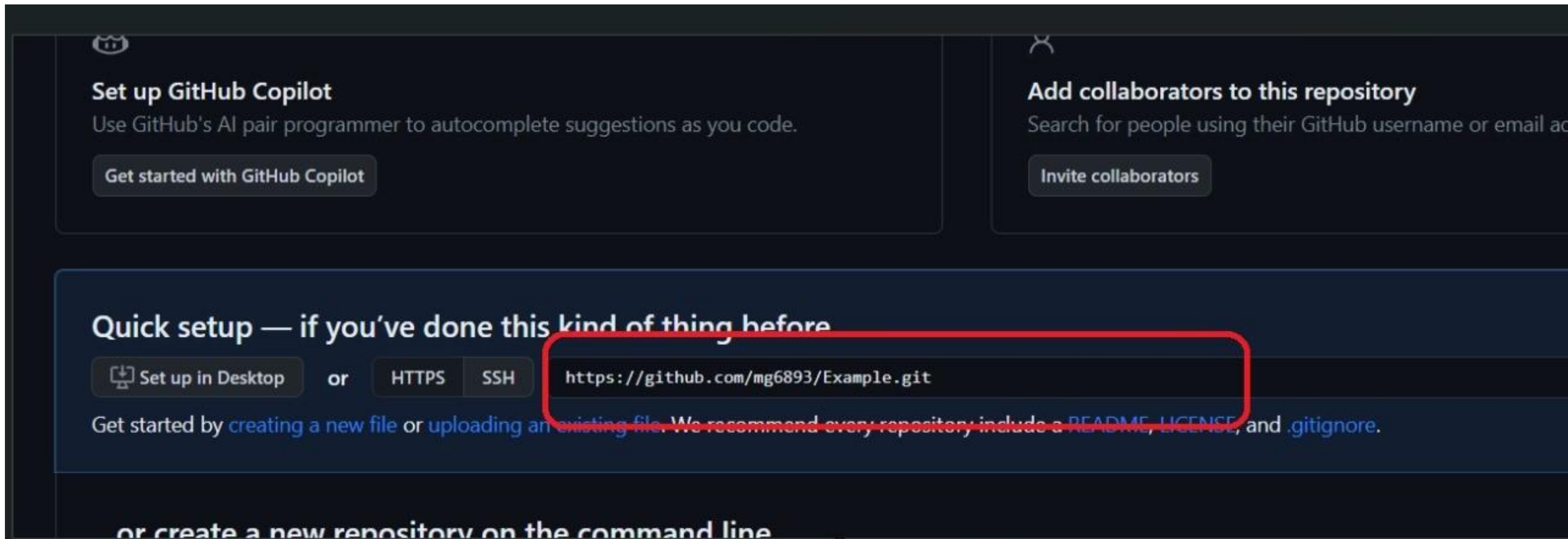
Configure Git:

- git config --list Command. (To verify configured username and email)

```
RES@DESKTOP-ESAPJPD MINGW64 ~  
$ git config --list  
diff.astextplain.textconv=astextplain  
filter.lfs.clean=git-lfs clean -- %f  
filter.lfs.smudge=git-lfs smudge -- %f  
filter.lfs.process=git-lfs filter-process  
filter.lfs.required=true  
http.sslbackend=openssl  
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt  
core.autocrlf=true  
core.fscache=true  
core.symlinks=false  
pull.rebase=false  
credential.helper=manager  
credential.https://dev.azure.com.usehttppath=true  
init.defaultbranch=master  
user.name=  
user.email=  
filter.lfs.clean=git-lfs clean -- %f  
filter.lfs.smudge=git-lfs smudge -- %f  
filter.lfs.process=git-lfs filter-process  
filter.lfs.required=true  
core.editor="C:\Users\RES\AppData\Local\Programs\Microsoft VS Code\bin\code" --wait  
RES@DESKTOP-ESAPJPD MINGW64 ~  
$ |
```

Basic Commands: Clone

- "Clone" in GitHub means making a copy of a repository from the remote server to your local computer for editing.
- syntax: `git clone <link>`




```
PS C:\Users\manas\OneDrive\Desktop\nano.cpp\development> git clone https://github.com/mg6893/Example.git
Cloning into 'Example'...
warning: You appear to have cloned an empty repository.
PS C:\Users\manas\OneDrive\Desktop\nano.cpp\development> cd project1
```

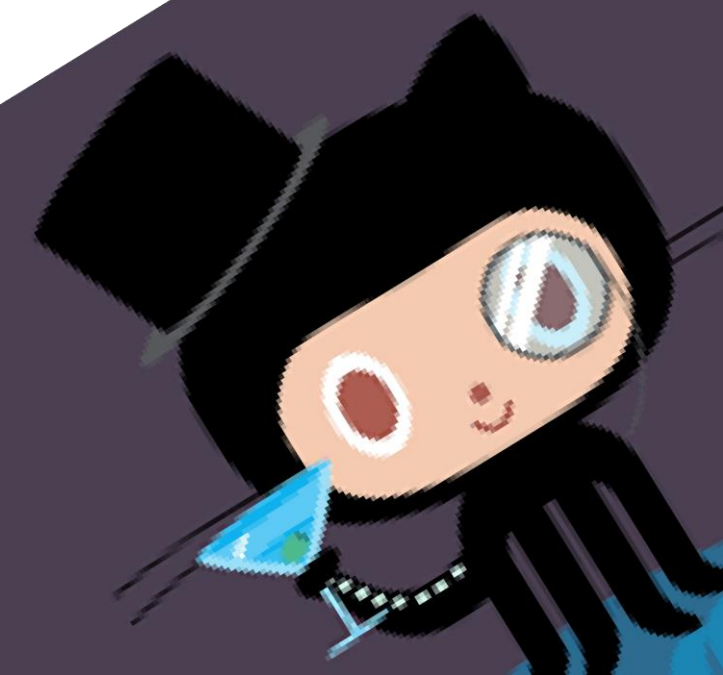


Git Status

- “git status” is a Git command that displays the current state of your local Git repository, showing which files have been modified, added, or deleted, and their status in relation to the last commit. It helps you track changes before committing them.
- syntax: git status

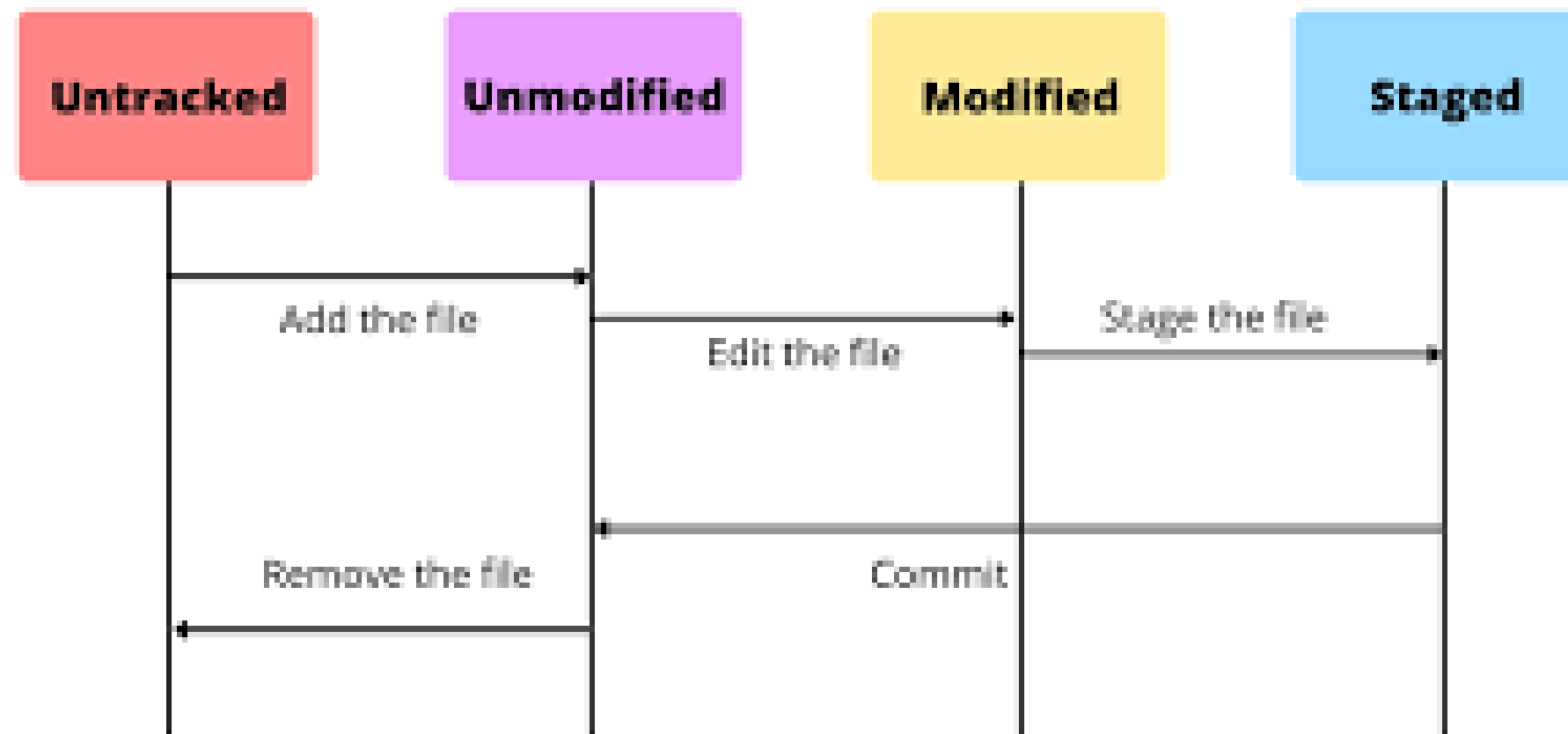
```
delete mode 100044 index.js  
PS C:\Users\manas\OneDrive\Desktop\nano.cpp\development\project1> git status  
On branch master  
nothing to commit, working tree clean  
PS C:\Users\manas\OneDrive\Desktop\nano.cpp\development\project1> |
```

```
PS C:\Users\manas\OneDrive\Desktop\nano.cpp\development\project1> git status  
On branch master  
Changes to be committed:  
  (use "git restore --staged <file>..." to unstage)  
    new file:   cpp/1.cpp  
    new file:   cpp/1.exe  
    modified:   index.html  
    modified:   index.js  
    new file:   new1.html  
  
Changes not staged for commit:  
  (use "git add <file>..." to update what will be committed)  
  (use "git restore <file>..." to discard changes in working directory)  
    modified:   cpp/1.cpp  
    modified:   cpp/1.exe
```



Status of a file/Repo:

- “Untracked”: New files that git doesn't yet track.
- “Modified”: Changed/Changes made in file.
- “Staged”: Files ready to be committed.
- “Unmodified”: Unchanged.



How Git Tracks:

- add: Adds new or changed files in our working directory to git staging areas.

syntax: `git add <file name>`

`git add . - >` to add all the files to staging area.

- commit: It records changes after add commands.

syntax : `git commit -m "message "`



```
MacBook-Air apnacollege-demo % git add .
```

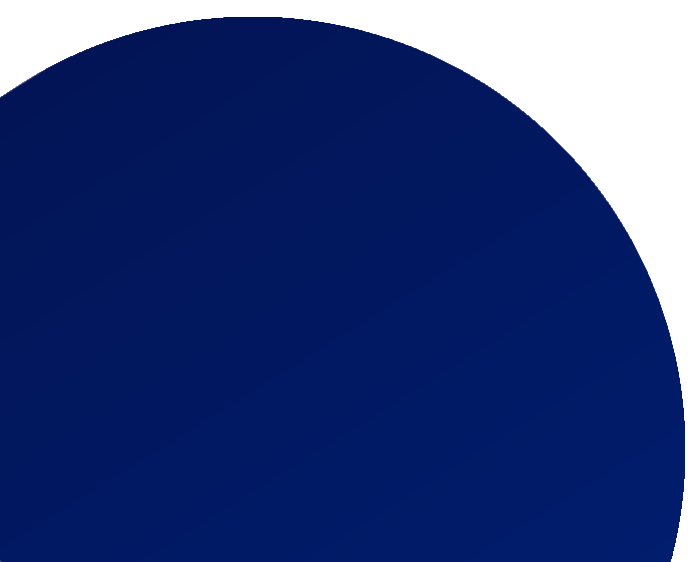
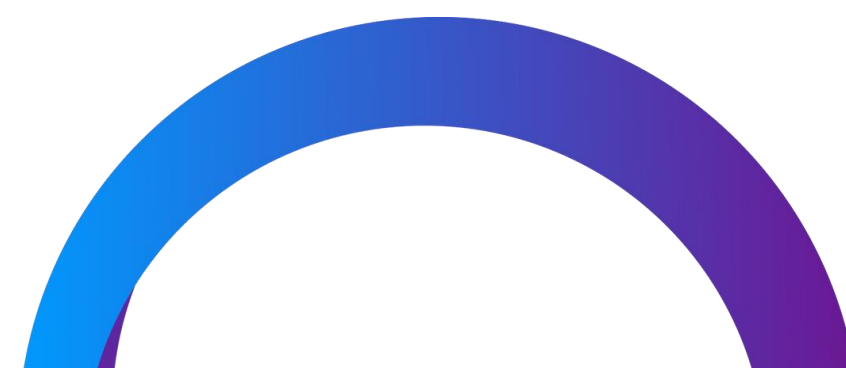
```
MacBook-Air apnacollege-demo % git commit -m "Add new paragraph"
1] Add new paragraph
changed, 3 insertions(+), 1 deletion(-)
100644 index.html
```

```
MacBook-Air apnacollege-demo % git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)


nothing to commit, working tree clean
```





Push:

- Used to upload repo content to remote (on github).
 - The "push" command in Git, commonly used with platforms like GitHub, is the process of uploading your local code changes to a remote repository. It synchronizes your local commits with the remote repository, allowing collaboration and sharing of code changes with others working on the same project.
 - syntax: `git push origin main`.
- 
- 

```
student@apnacollege-MacBook-Air apnacollege-demo % git push origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 401 bytes | 401.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/student-apnacollege/apnacollege-demo.git
e721017..efab431  main -> main
```

 **student-apnacollege** Add new paragraph efab431 18 minutes ago 🕒 4 commits

 README.md	Add new paragraph	18 minutes ago
 index.html	Add new paragraph	18 minutes ago

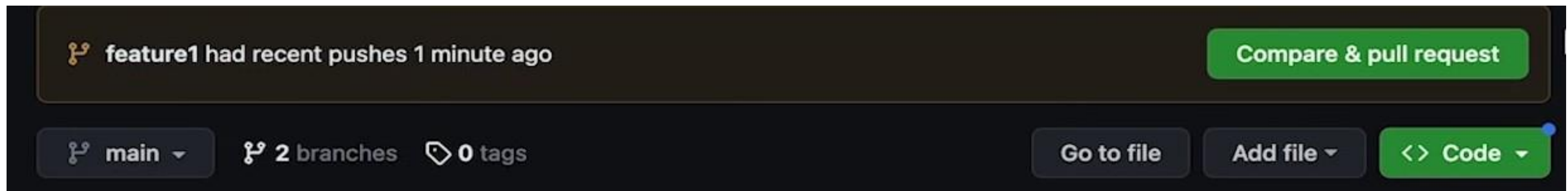
What we have done till now:

- Created a new repository on github
- Cloned it to VS Code
- Made changes in repo and then added and committed.
- Again pushed back that repo to github.



Pull request:

- A "pull request" (PR) is a feature on platforms like GitHub where you propose code changes to a repository. Others review, discuss, and potentially merge your changes into the main codebase. It facilitates collaboration and quality control.
- syntax: `git pull origin main`





base: main ▾



compare: feature1 ▾

✓ **Able to merge.** These branches can be automatically merged.



Add new feature

Write

Preview

H B I @

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.



Create pull request



Merge pull request #1 from student-apnacollege/feature1

Add new feature

This commit will be authored by 143068470+student-apnacollege@users.noreply.github.com

Confirm merge

Cancel



Pull request successfully merged and closed

You're all set—the `feature1` branch can be safely deleted.

Delete branch

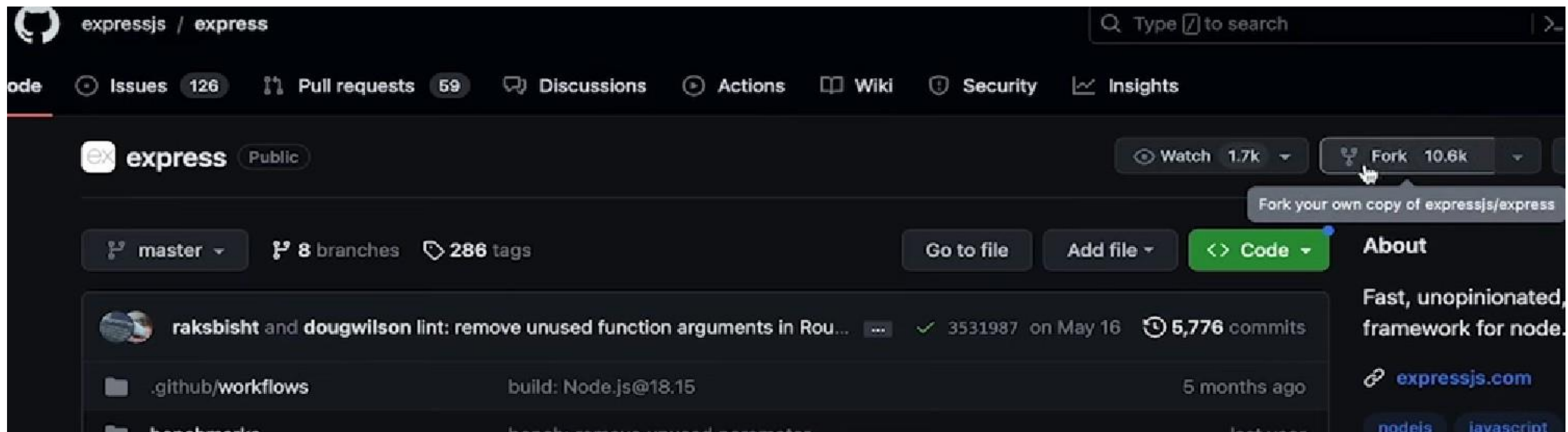
Resolving Merge conflicts:

- An event that takes place when git is unable to automatically resolve differences in code between two commits.
- syntax: `git merge <file_name>`

```
1  <p>This is a new repo.</p>
   Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
2  <<<<<< HEAD (Current Change)
3  <p>This is a new feature (dropdown)</p>
4  ===== I
5  <p>This is a new feature (button)</p>
6  >>>>>> main (Incoming Change)
7
   Resolve in Merge Editor
```

Fork:

- The "fork" command in Git, typically used in platforms like GitHub, creates a personal copy of a repository, allowing you to make independent changes without affecting the original project.
- Fork is like a rough copy



Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Required fields are marked with an asterisk ().*

Owner *

 student-

Repository name *

/ express

✔ express is available.


By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

Fast, unopinionated, minimalist web framework for node.

☒ Copy the master branch only

Contribute back to expressjs/express by adding your own branch. [Learn more.](#)

 You are creating a fork in your personal account.

Create fork



THANK YOU!

