INTRODUCTION TO

GIT & GITHUB

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

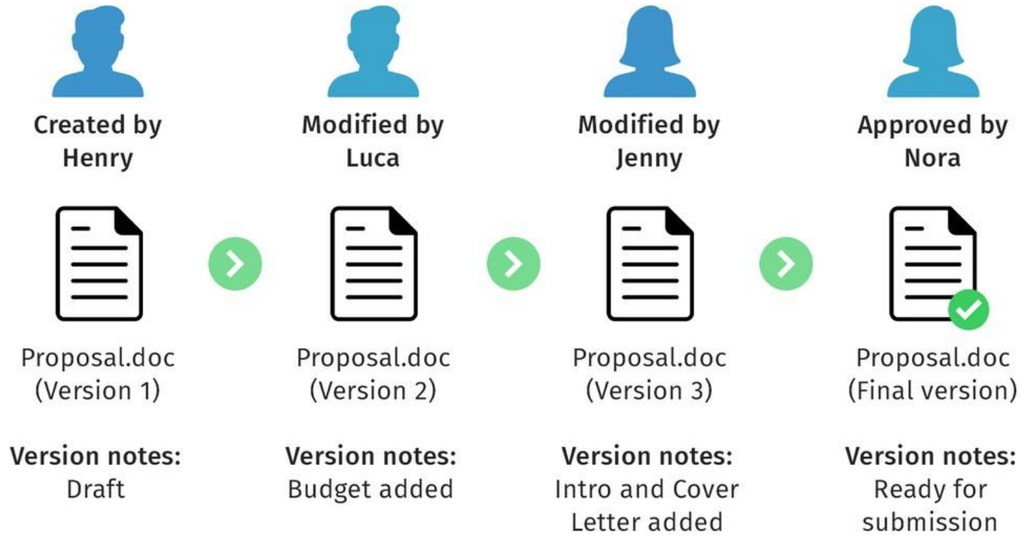
Guided by: Dr. Lalit Purohit Sir and
Mr. Upendra Singh Sir

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

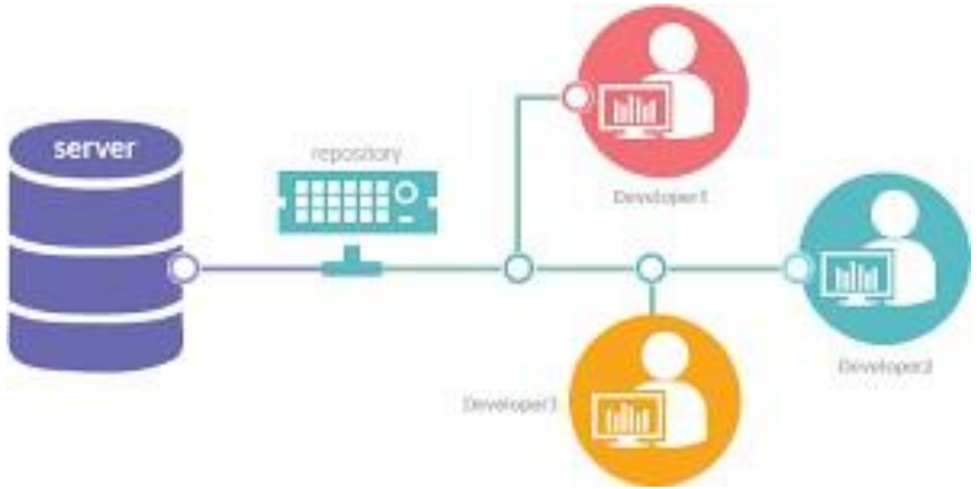
What is Version Control System ?

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:

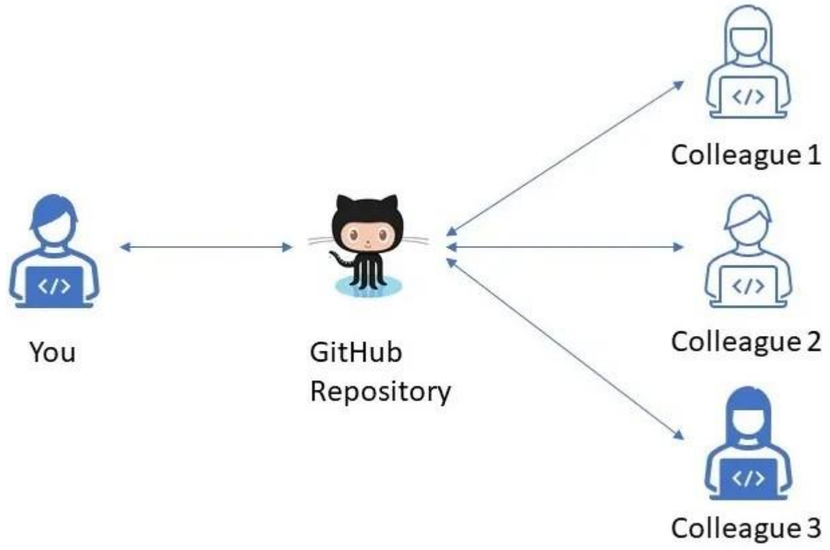
- Git enables easy branching to work on different features or changes independently and allows seamless merging of branches.

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

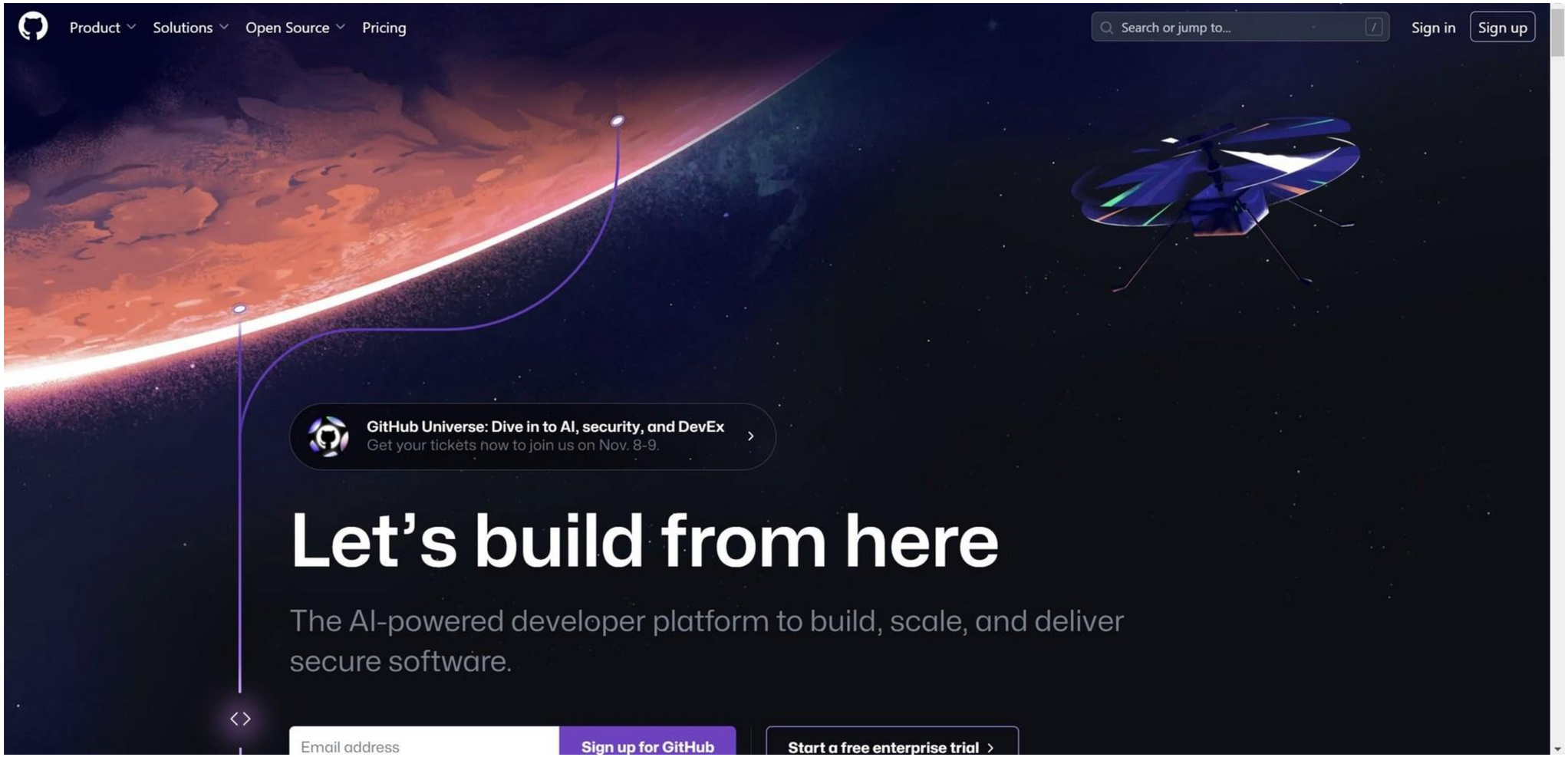
- Distributed version control system.
- Operates on a developer's machine.
- Manages code versions locally.
- No internet dependency.
- No collaboration features
- Command-line tool.

GitHub

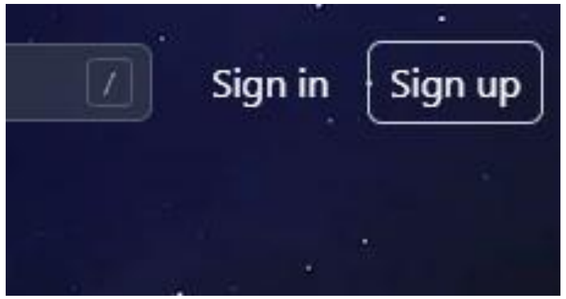
- Web-based platform.
- Hosts repositories in the cloud.
- Provides project management features.
- Requires internet access for usage.
- Facilitates remote collaboration.
- Provides a graphical interface.

Creating account on GitHub:

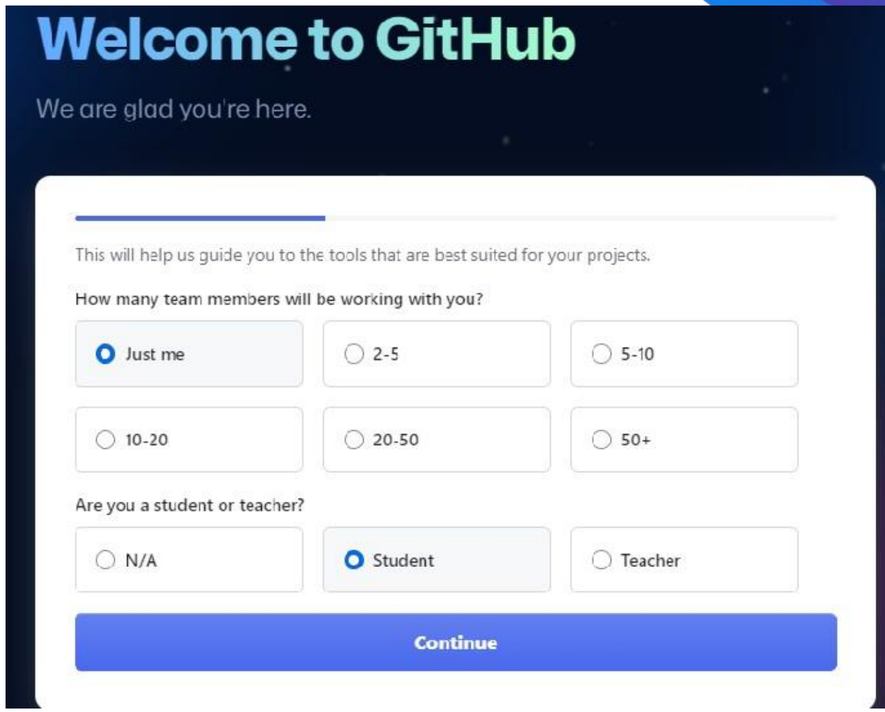
1 Open your web browser and go to the GitHub website at https://github.com.



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



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.



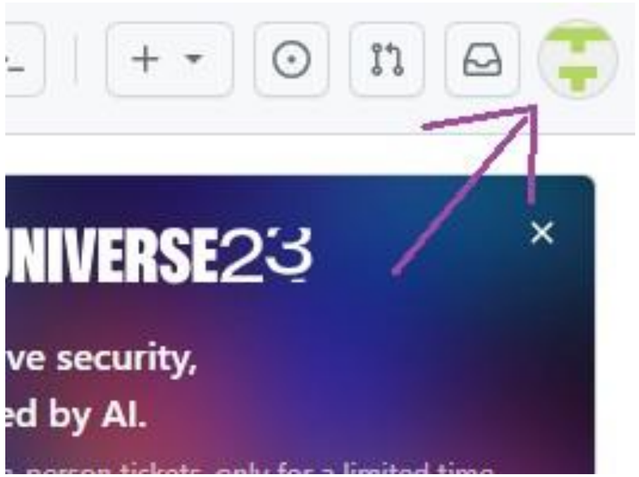
4. You'll be prompted to complete the sign-up process. Optionally, you can:

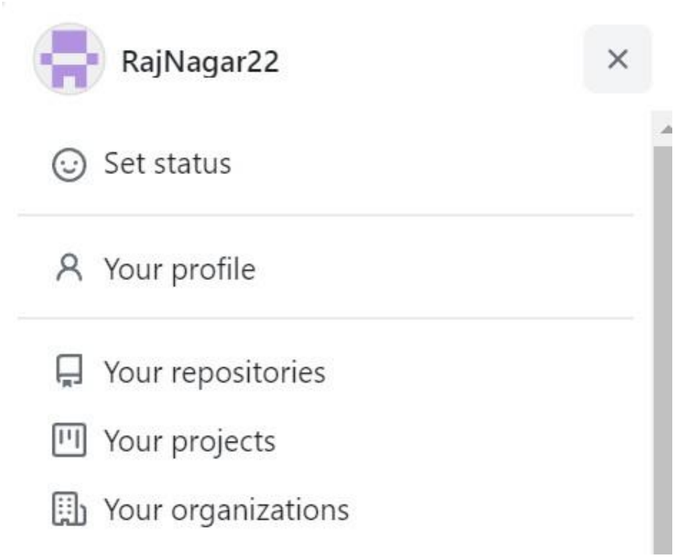
- Customize your experience by selecting your interests.
- Choose to receive updates and newsletters from GitHub.

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

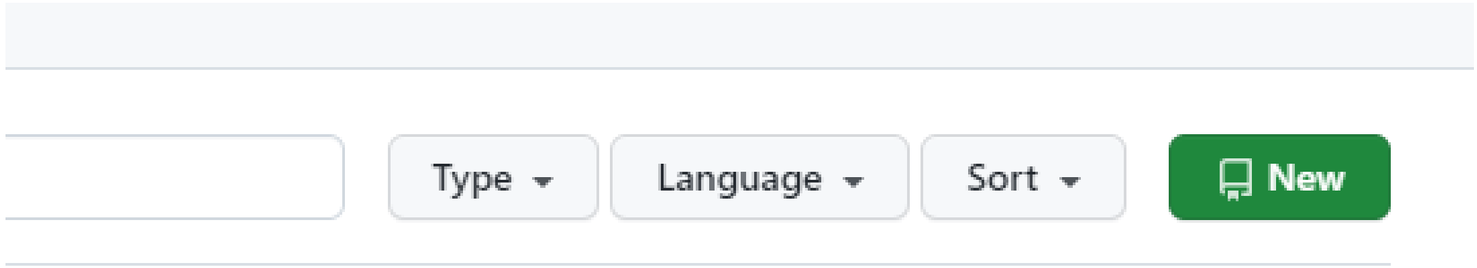
Creating New Repository in GitHub:

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

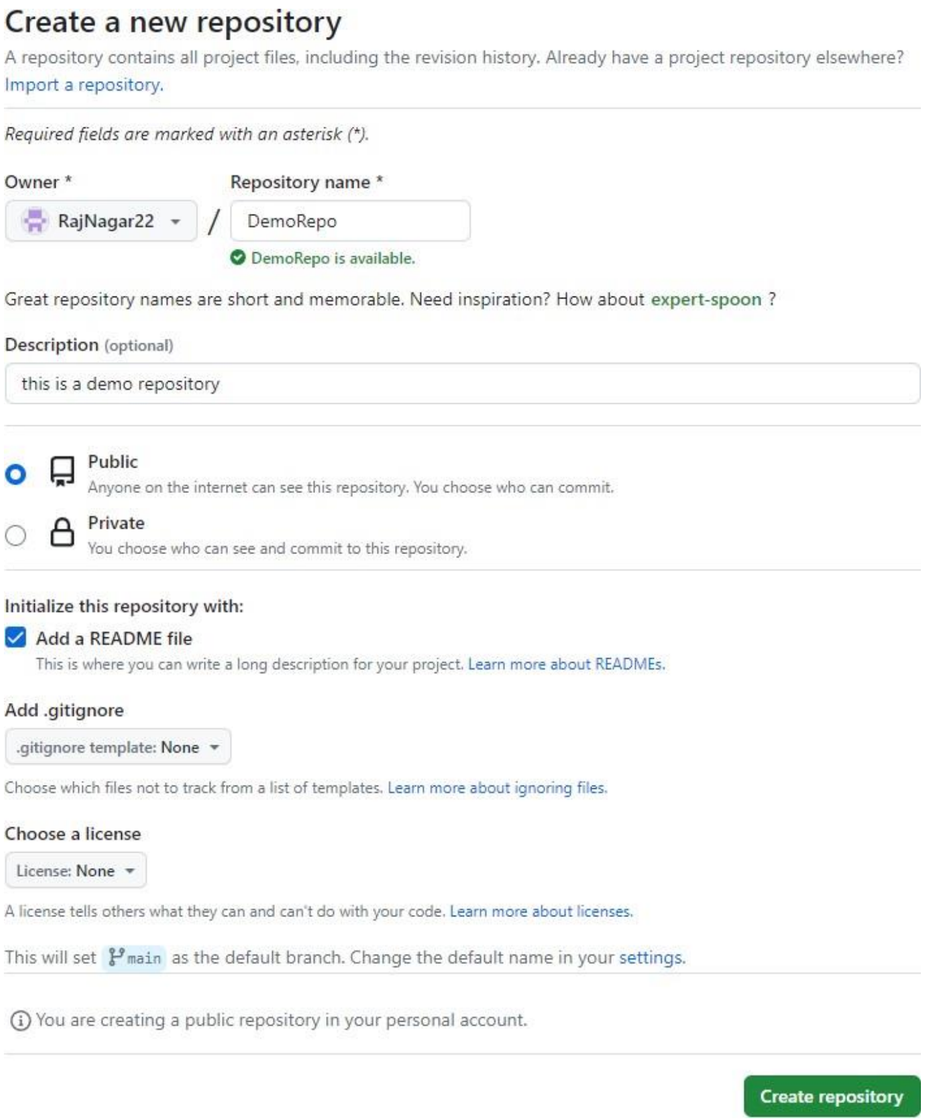




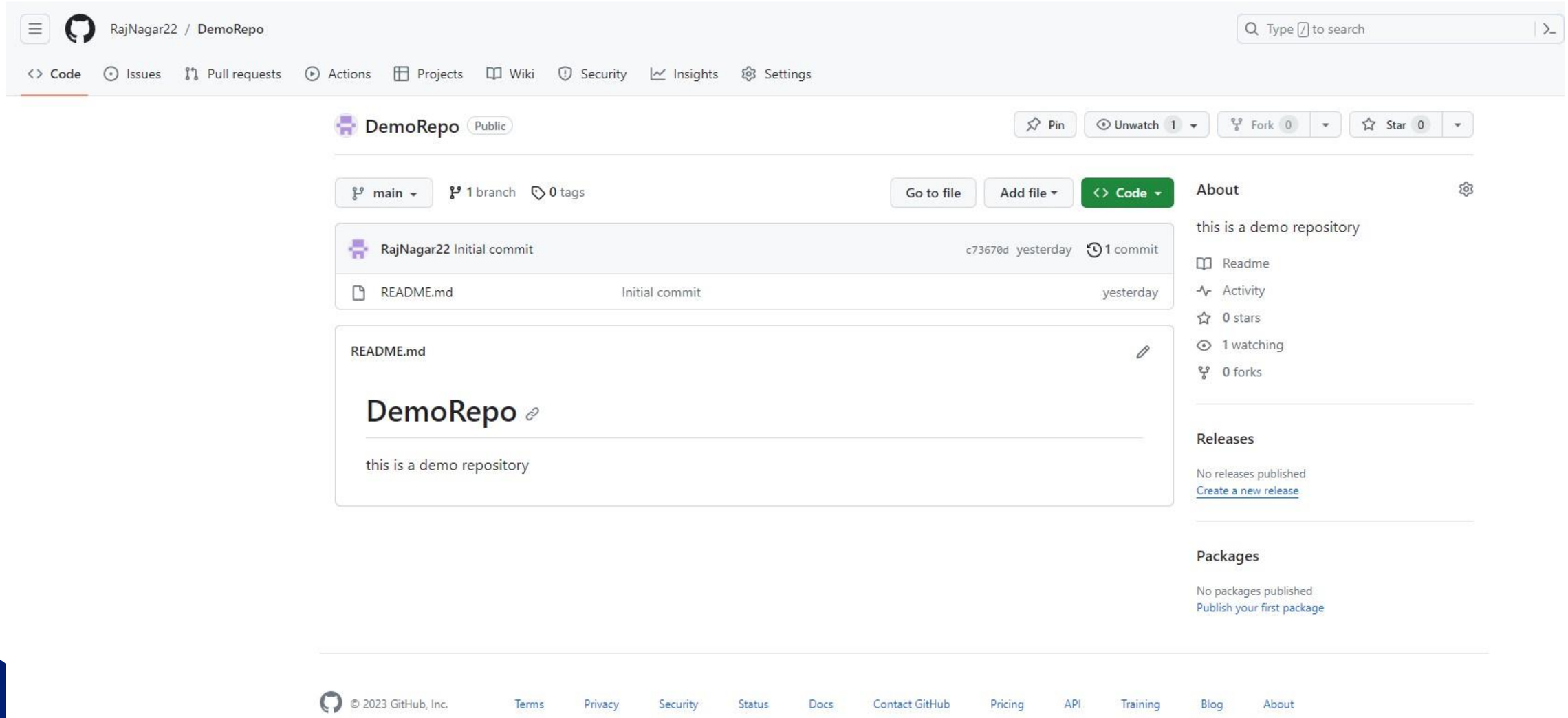
3. Now select “New” button



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.

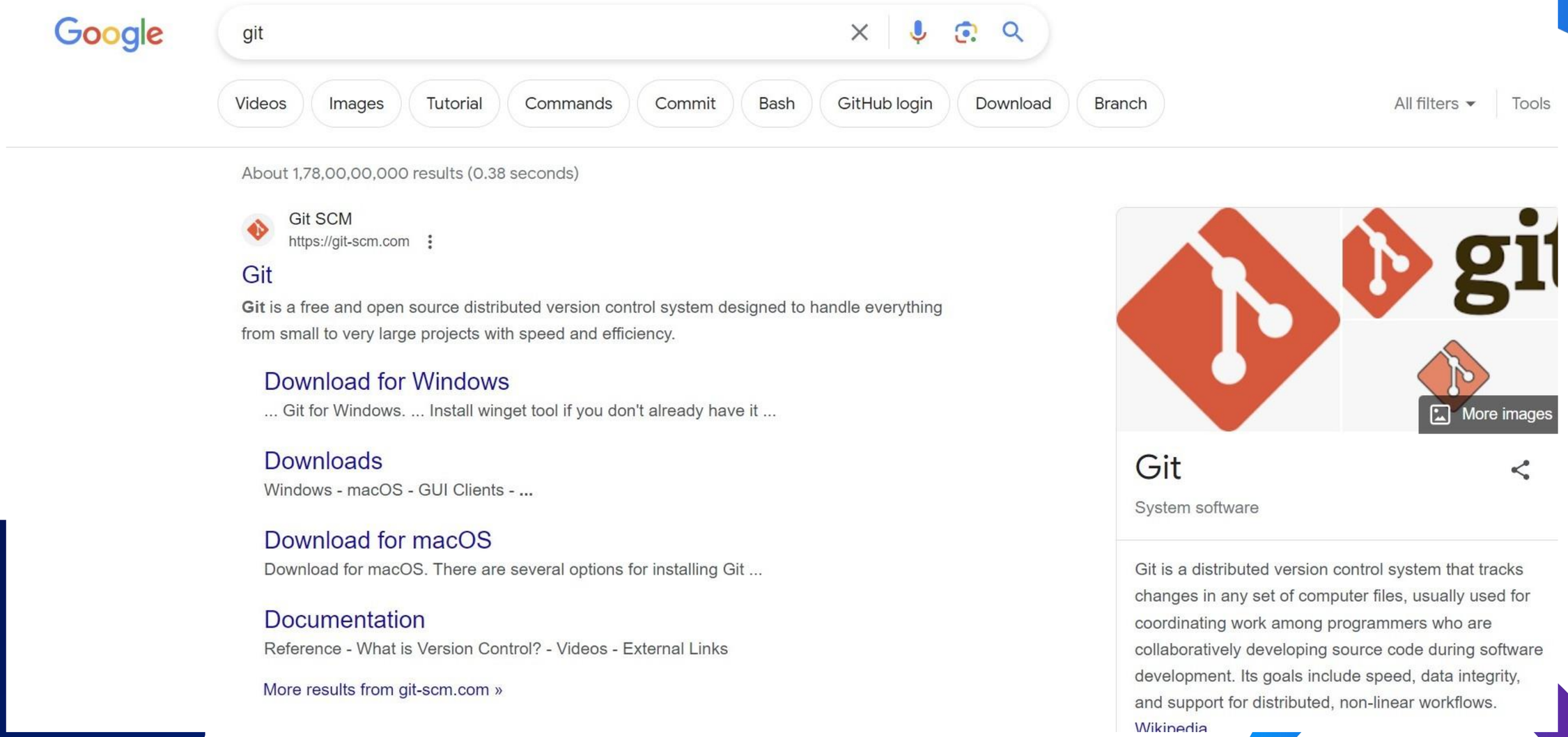


5. Newly created Repository will appear like this :

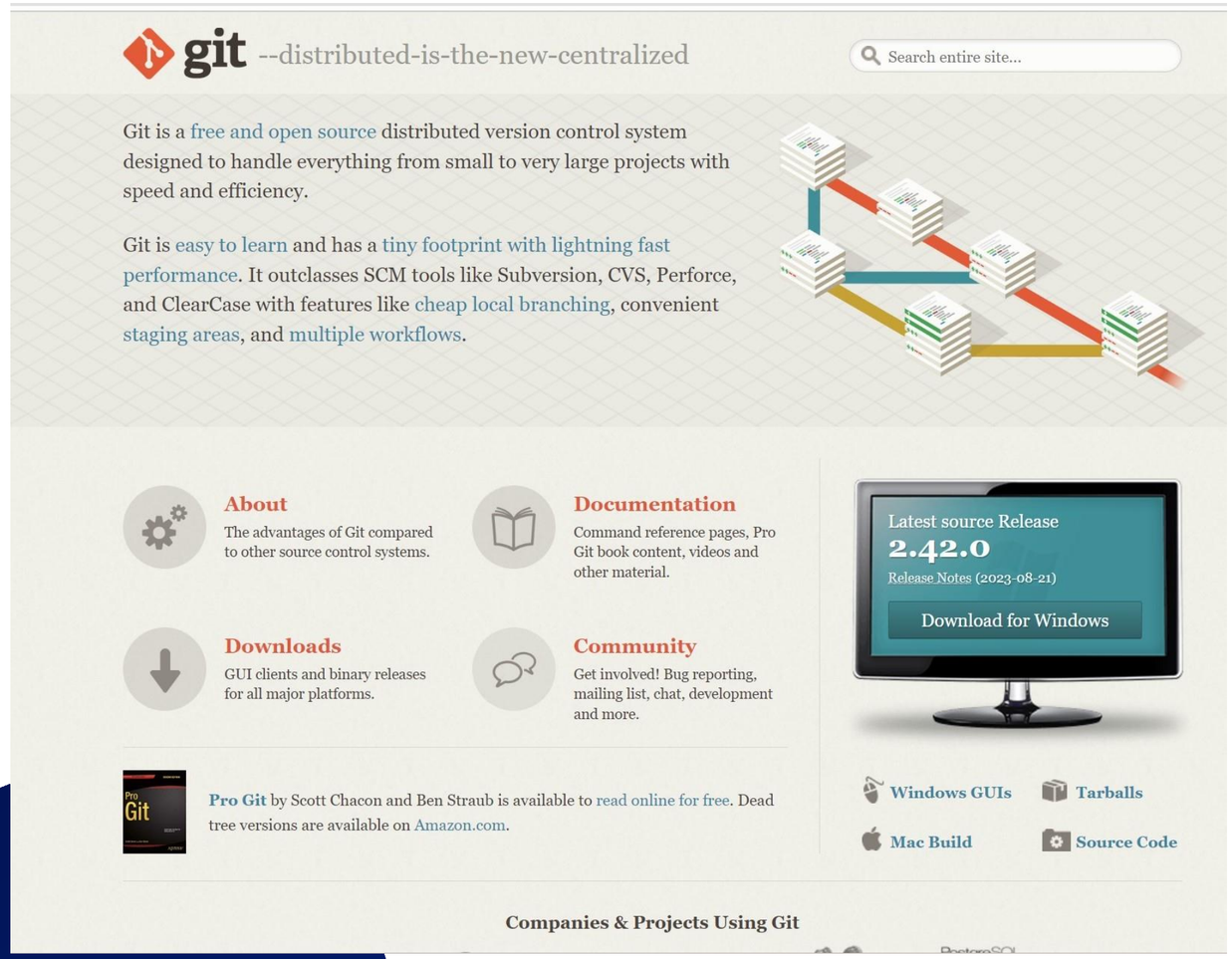


Setting Up Git:

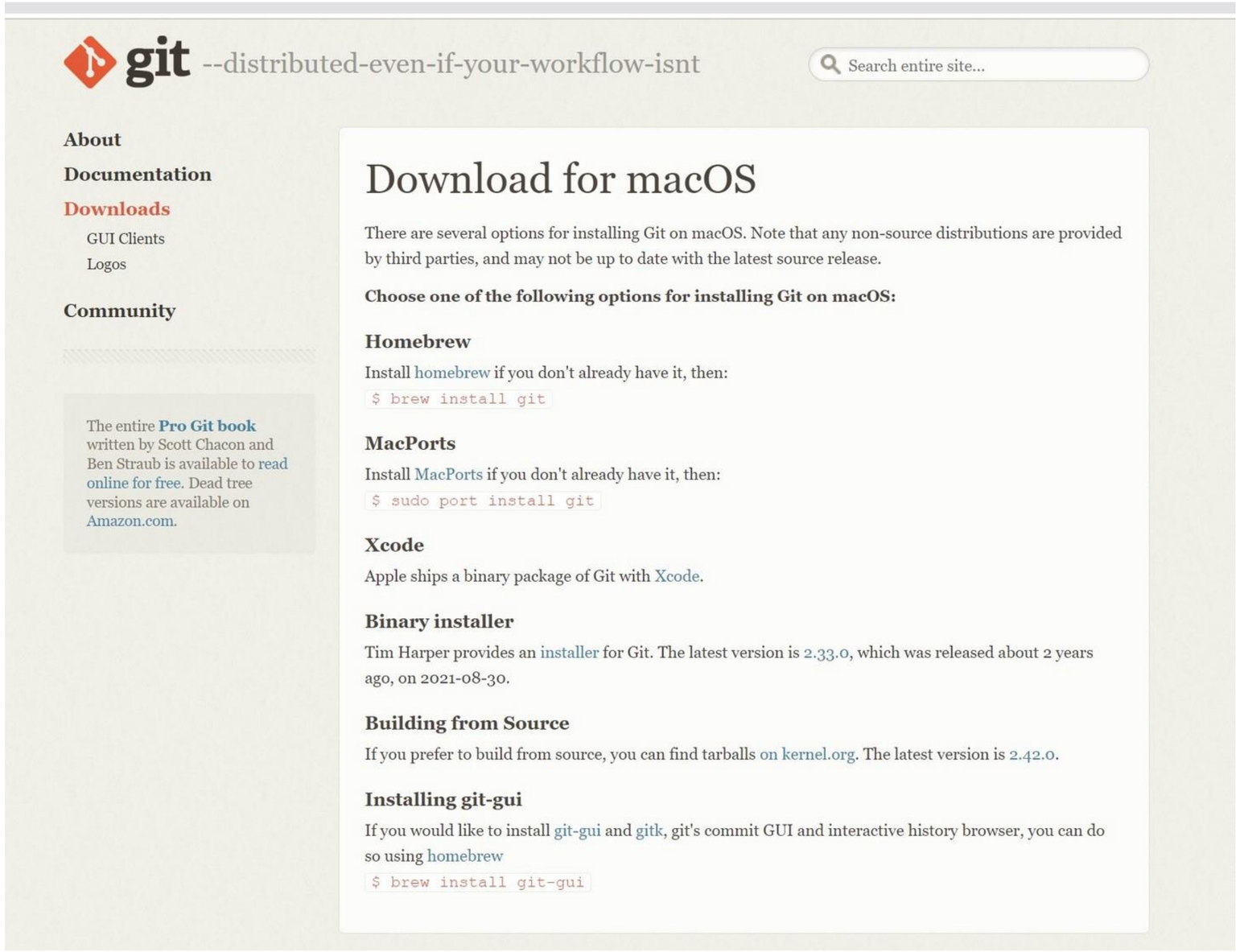
1.Type git in your web browser’s search bar.



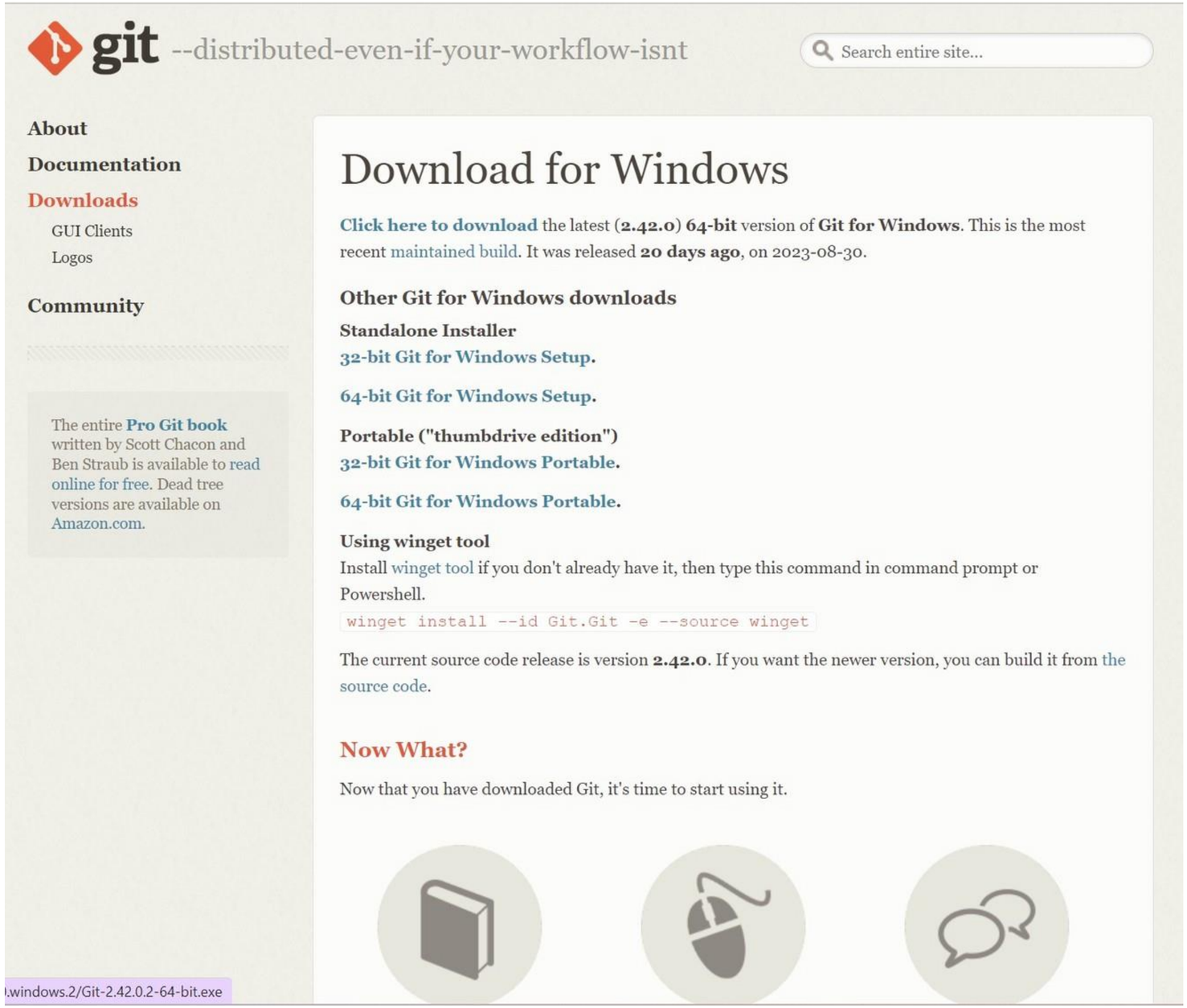
2. If you’re a windows user click download for windows, else click on mac build.



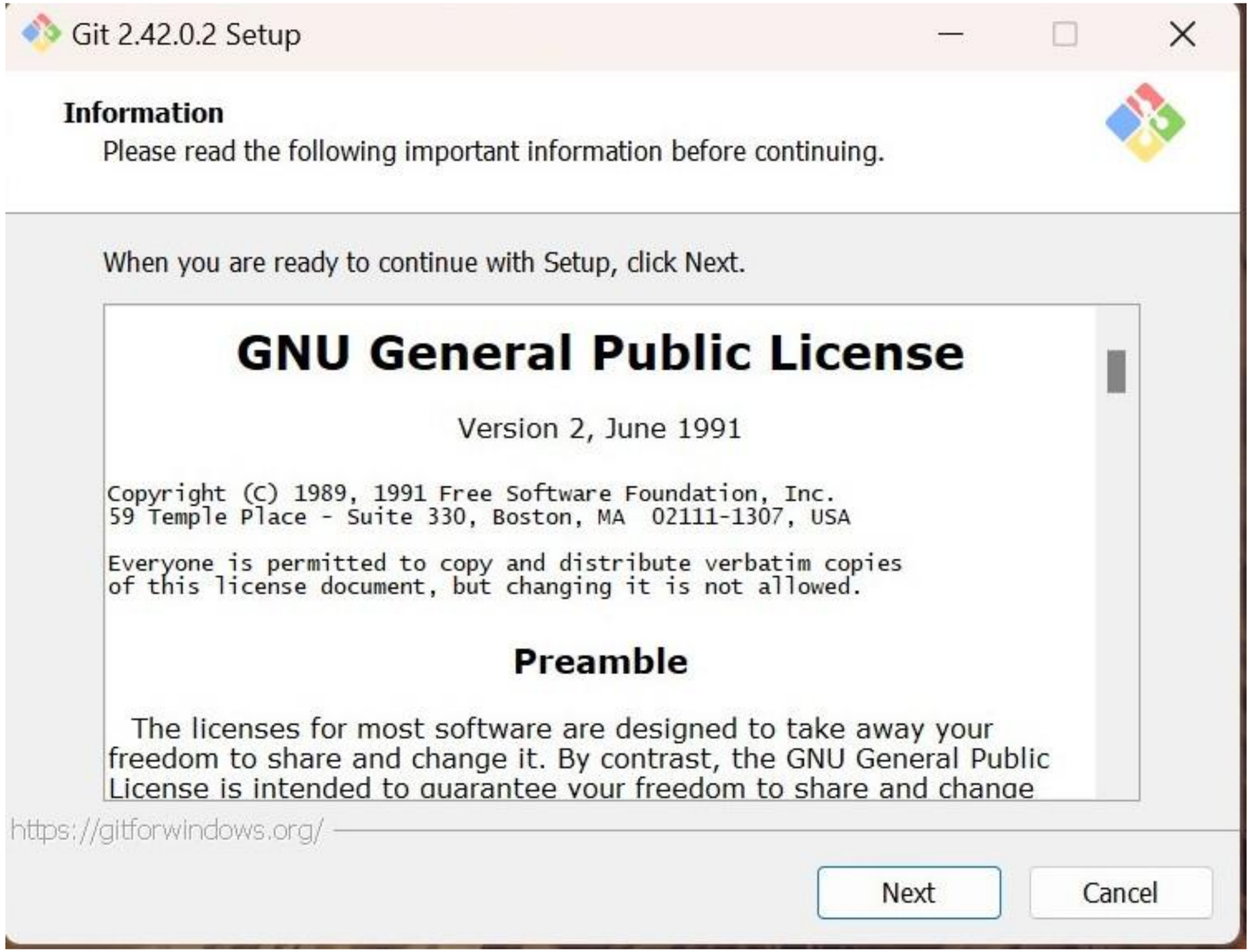
3. Mac users should follow the given procedure:



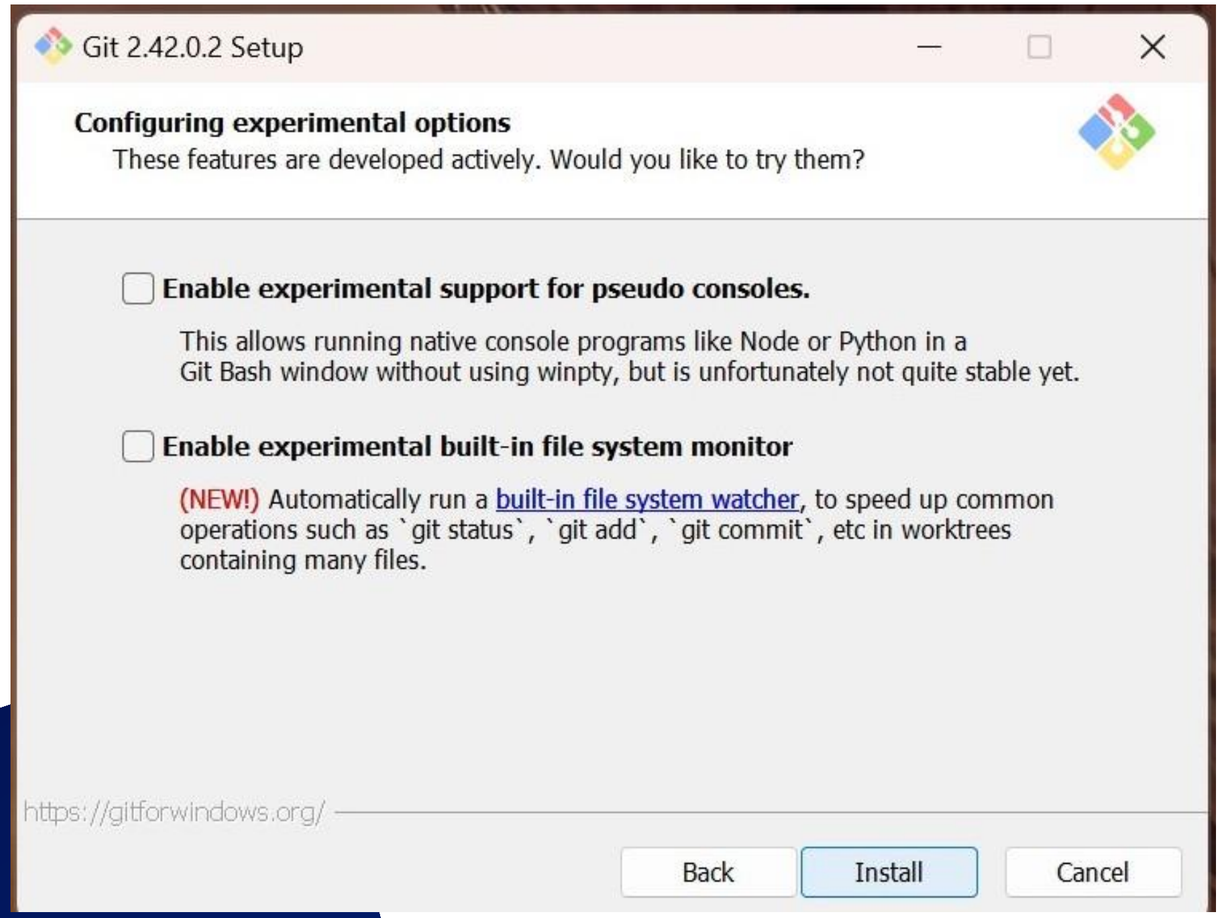
4. Click on click here to download.

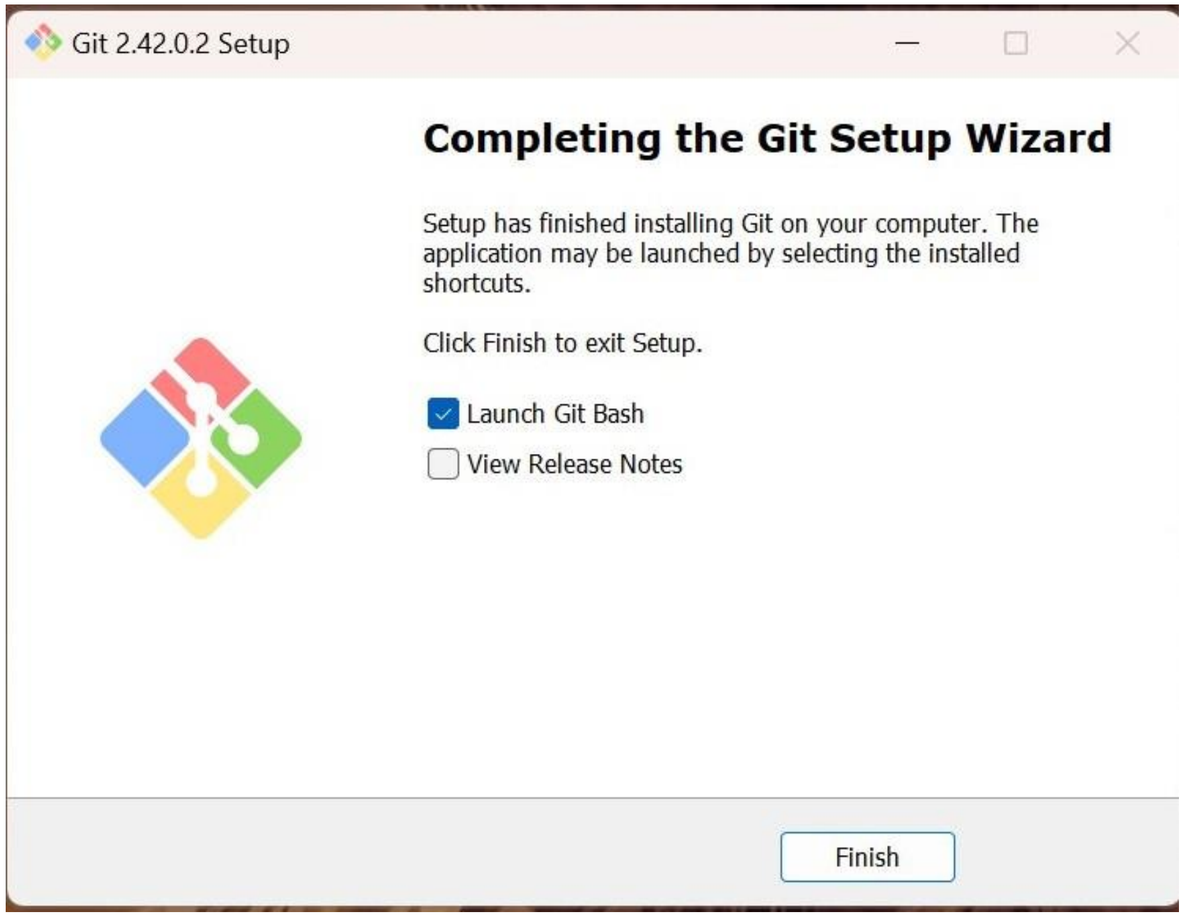


5. After the download is complete, open the downloaded file and the following setup dialog box will appear.

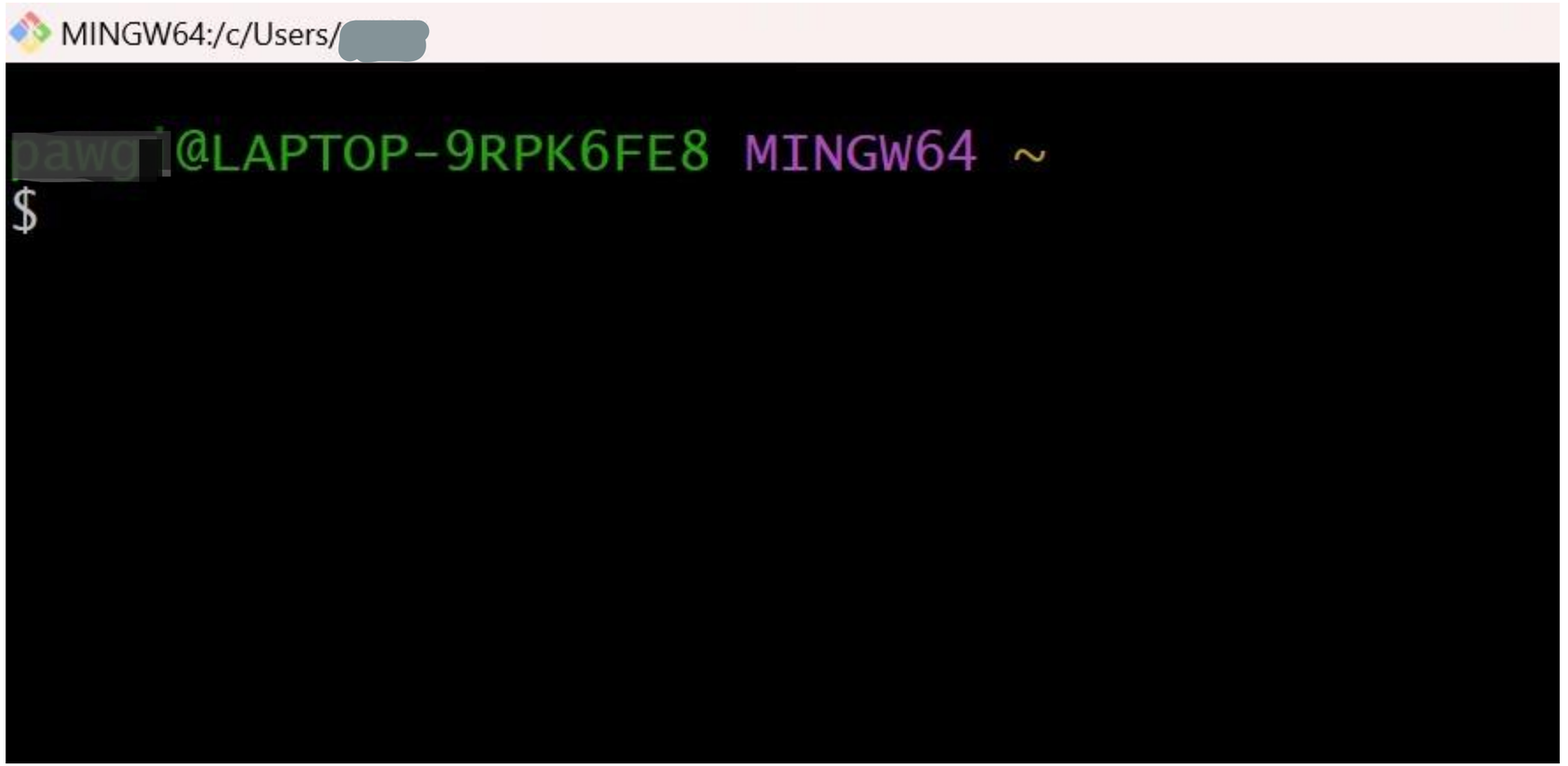


6. Keep clicking “next” until the following dialog box appears. Now click ”install” and git would be installed. Clicking finish would launch Git Bash.





7. This is how the git bash looks like.

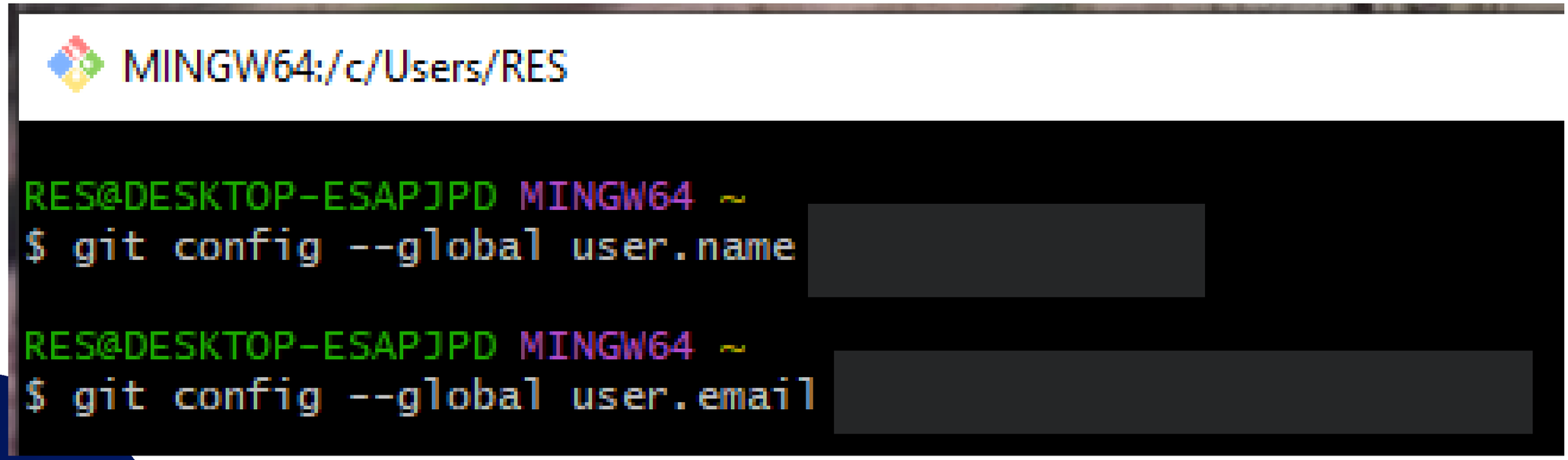


What is Git Bash and why are we using ?

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:



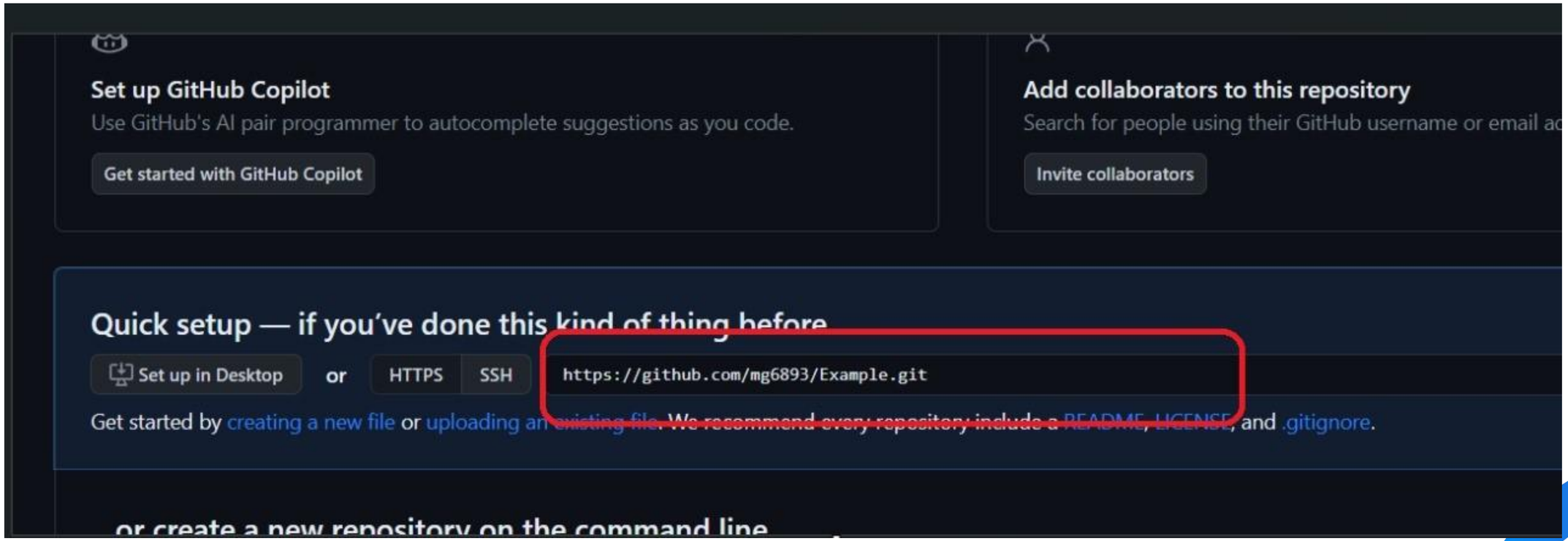
Configure Git:

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

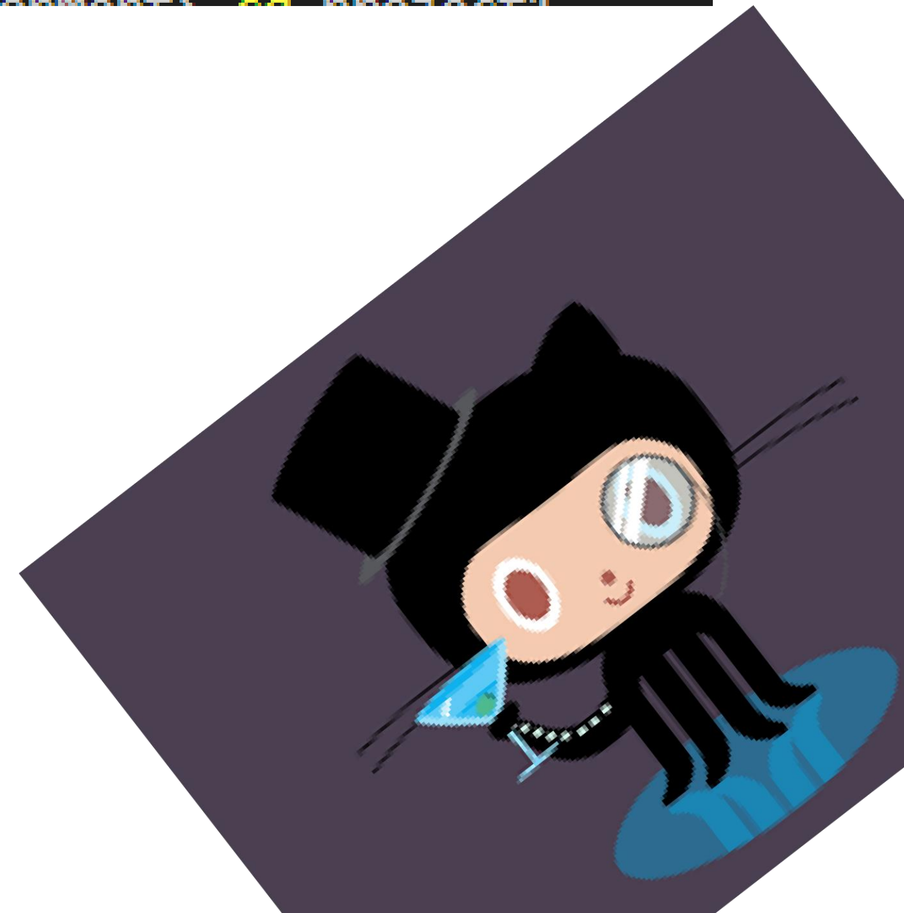
```
RES@DESKTOP-ESAPJPD MINGW64 ~
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
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" --w
ait

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:/
ithub.com/mg6893/Example.git
Cloning into 'Example'...
warning: You appear to have cloned an empty repository.
```

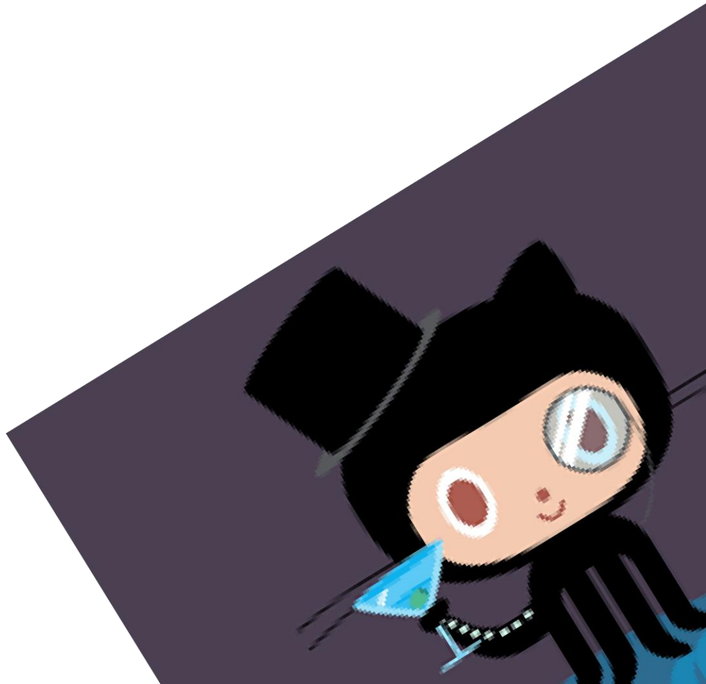


Git Status

- “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

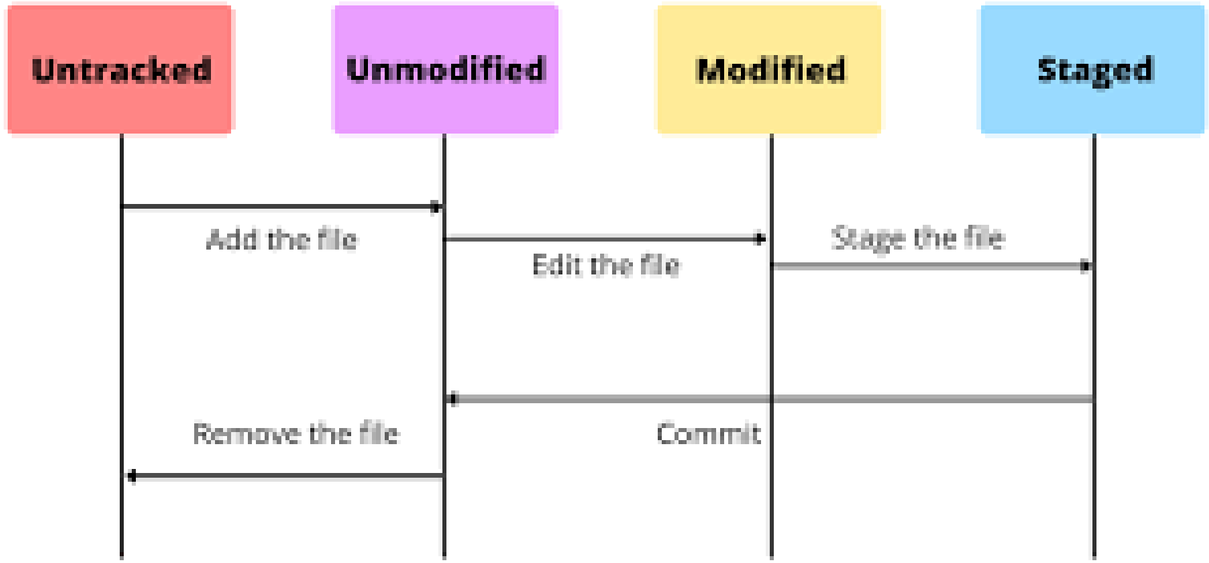
```
delete mode 100644 index.js
PS C:\Users\manas\OneDrive\Desktop\nano.cpp\development\project1> git status
On branch master
nothing to commit, working tree clean
PS C:\Users\manas\OneDrive\Desktop\nano.cpp\development\project1>
```

```
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 “

```
[redacted]-MacBook-Air apnacollege-demo % git add .
```

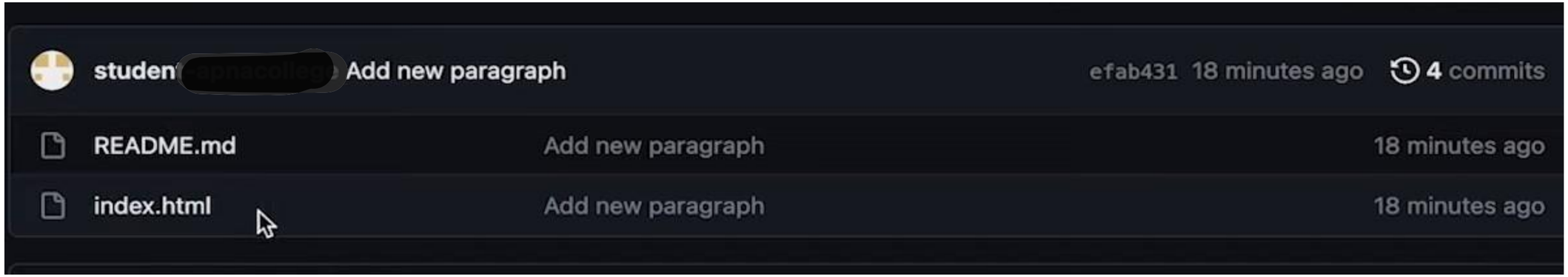
```
[redacted]MacBook-Air [redacted]-demo % git commit -m "Add new paragraph"
1] Add new paragraph
nged, 3 insertions(+), 1 deletion(-)
 100644 index.html
```

```
[redacted]-MacBook-Air [redacted]-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.

```
[redacted]-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[redacted]
   e721017..efab431  main -> main
```

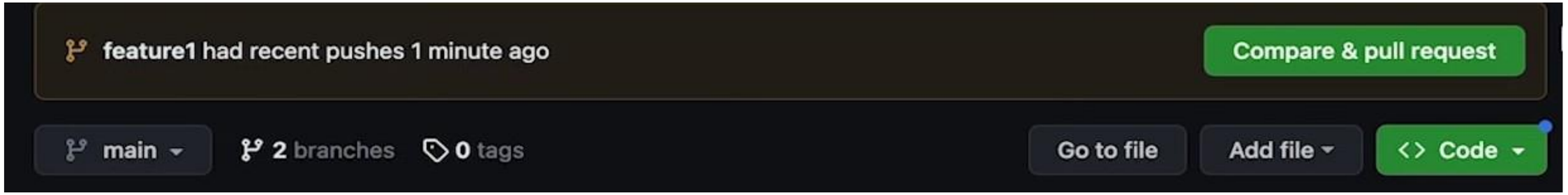


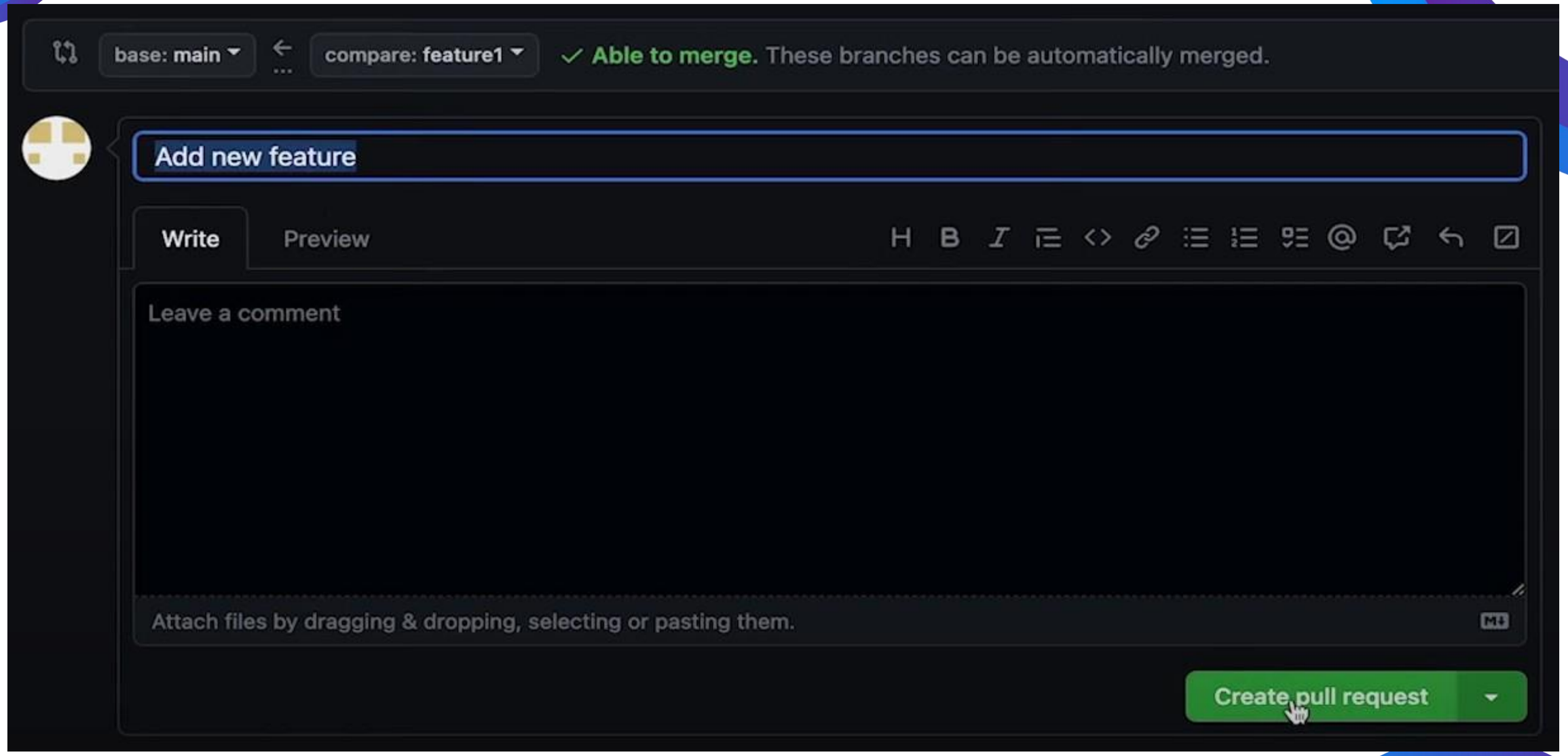
What we have done till now:

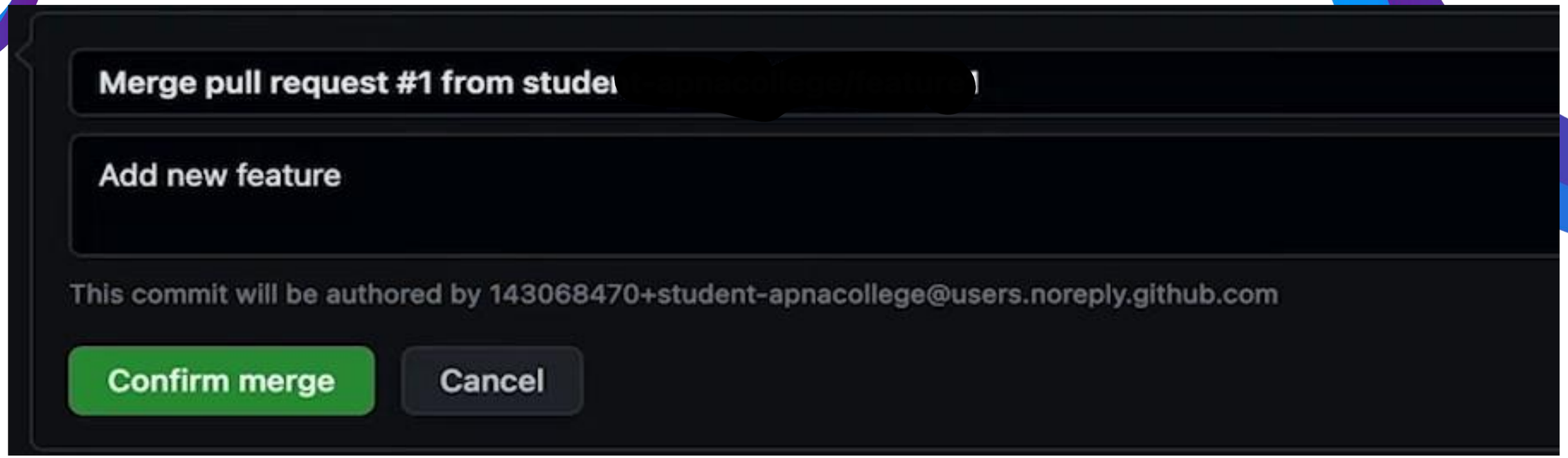
- Created a new repository on github
- Cloned it to VS Code
- Made changes in repo and then added and committed.
- Again pushed back that repo to github.

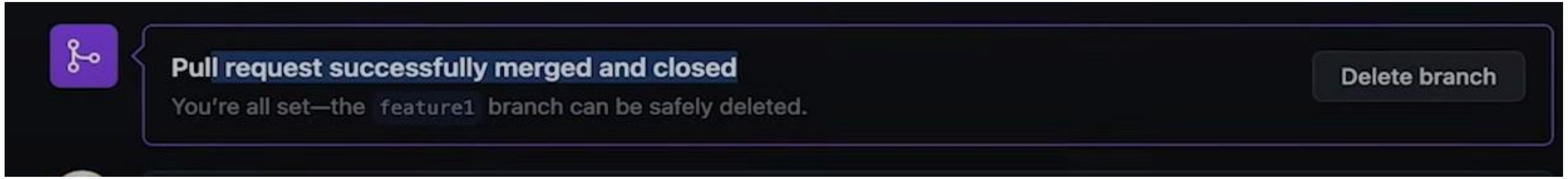
Pull request:

- 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
Leave a comment
Attach files by dragging & dropping, selecting or pasting them.
Create pull request


Merge pull request #1 from stude
Add new feature
This commit will be authored by 143068470+student-apnacollege@users.noreply.github.com
Confirm merge
Cancel

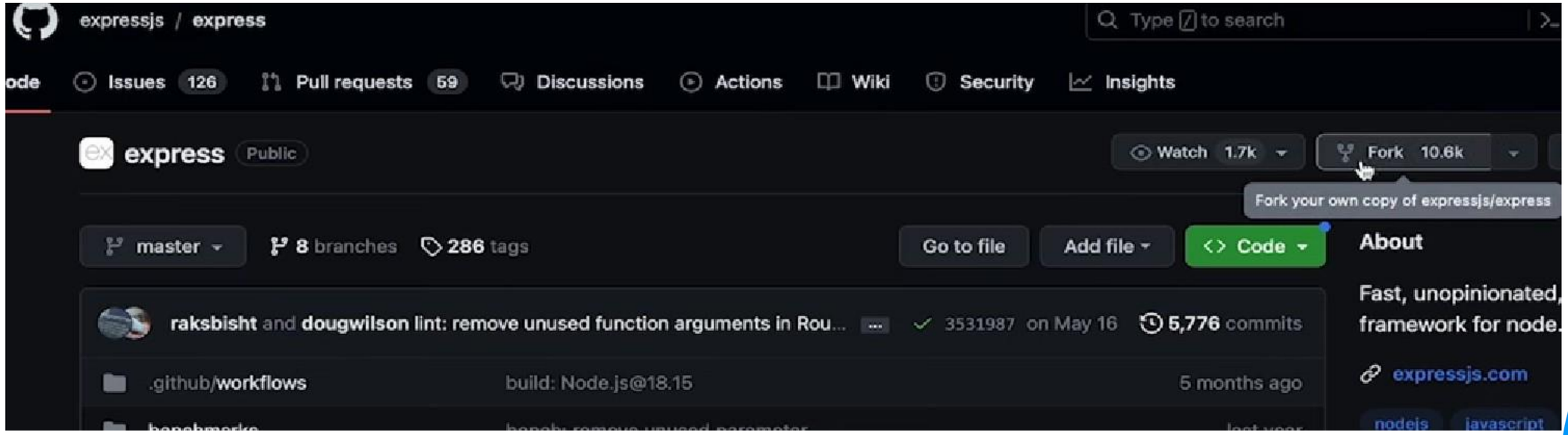

Pull request successfully merged and closed
You're all set—the feature1 branch can be safely deleted.
Delete branch

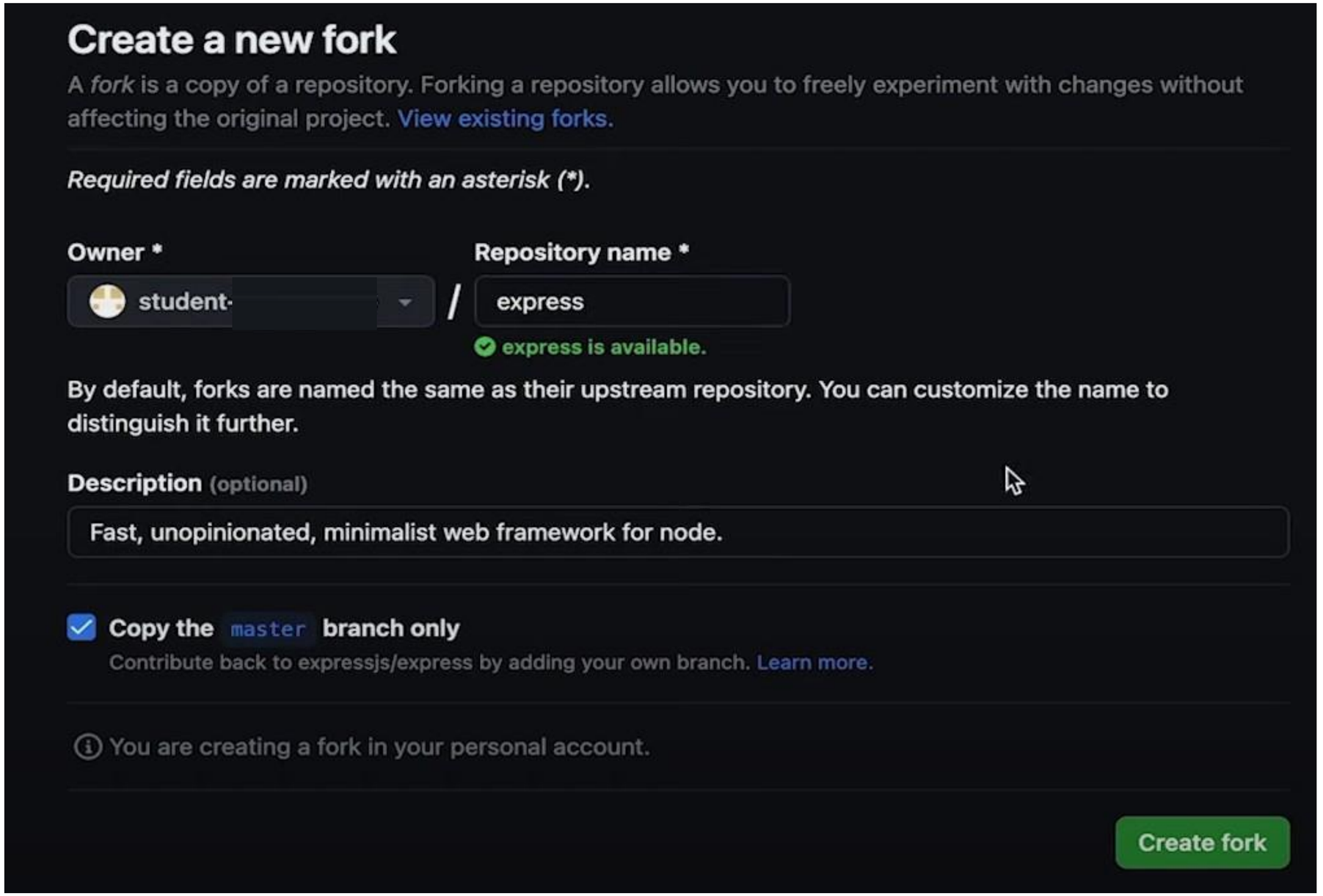
Resolving Merge conflicts:

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

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!