

Git & GitHub

- By Tejal Pathak

Topics to study:

- Version control system
- Installation of Git
- Introduction to Git
- Git Basics
- Branching
- Merging
- Rebasing
- Reflog
- Stashing
- GitHub
- Extra Tools

VCS: What is VCS ?

- VCS stands for version control system.
- Version control is the practice of managing and tracking changes in software codes. It is also known as source control.
- Version control system is simply a software which helps us managing a version control.
- It allows to maintain code efficiently.

Types of VCS

- Local VCS
- Centralized VCS
- Distributed VCS

Local VCS

- All changes done in the file will be stored locally and will be only available on user's machine.
- Changes are stored with time so as to keep the track of when exactly changes were made.

Disadvantages of Local VCS

- Since all the changes are stored locally, If the hard disk gets corrupted then the project can be lost.
- No other person can access your changes.

Centralized VCS

- In Centralized VCS there is only one central repository where all changes are made by multiple people.
- Multiple people can get the code from central repo and then push their changes into same repo.

Disadvantages of Centralized VCS

- In case central repo is not available or down for some time no one will be able to get the code from the repo and push the changes.
- If the hard disk where central repo is changed get's corrupted then the entire history of the project will get lost.

Distributed VCS

- In Distributed VCS user has their own copy of repository from central repository.
- Even if central repo goes down user still has their own copy of repo and they can continue working with that.

Installation of Git on Windows

- Go to this URL - <https://git-scm.com/>



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.

Downloads
GUI clients and binary releases for all major platforms.

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

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

Latest source Release
2.44.0
Release Notes (2024-02-23)
[Download for Windows](#)

Windows GUIs **Tarballs**

Pro Git by Scott Chacon and Ben Straub is available to [read online for free](#). Dead



[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](#).

Downloads



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 →](#)

Latest source Release

2.44.0

[Release Notes \(2024-02-23\)](#)

[Download for Windows](#)



Logos

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

[View Logos →](#)

- Your git is installed – to check if it's installed open command prompt and check: **git --version**

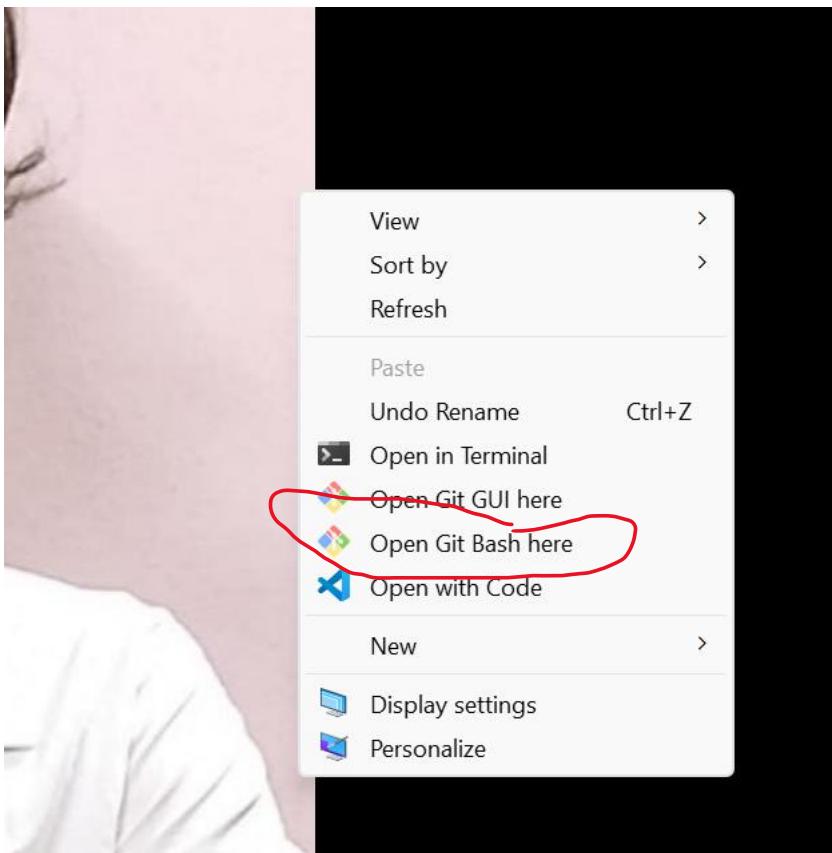
```
Microsoft Windows [Version 10.0.22631.3235]
(c) Microsoft Corporation. All rights reserved.
```

```
C:\Users\hp>git --version
git version 2.44.0.windows.1
```

```
C:\Users\hp>
```

To configure Git

- Go to desktop and right click -> git bash



MINGW64:/c/Users/hp/Desktop

hp@LAPTOP-6SAE1LLS MINGW64 ~/Desktop
\$ |

n
n
nel



Tejal -
Chrome



Visual Studio
Code



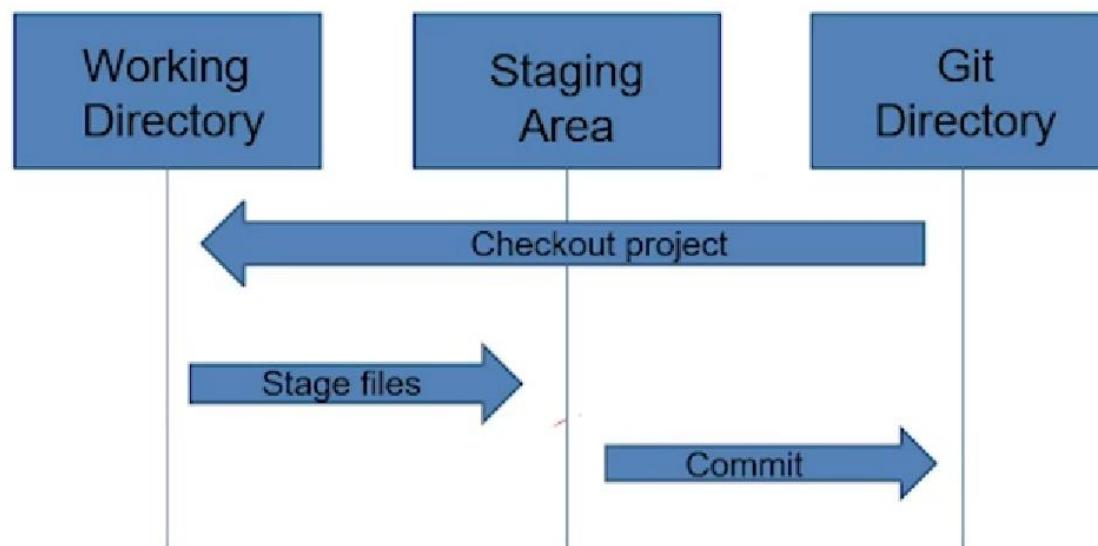
```
hp@LAPTOP-6SAE1LLS MINGW64 ~/Desktop
$ git config --global user.name "Tejal Pathak"

hp@LAPTOP-6SAE1LLS MINGW64 ~/Desktop
$ git config --global user.email "pathak23tejal@gmail.com"

hp@LAPTOP-6SAE1LLS MINGW64 ~/Desktop
$ 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=Tejal Pathak
user.email=pathak23tejal@gmail.com
```

```
hp@LAPTOP-6SAE1LLS MINGW64 ~/Desktop
$ |
```

Introduction to Git – Three stage architecture



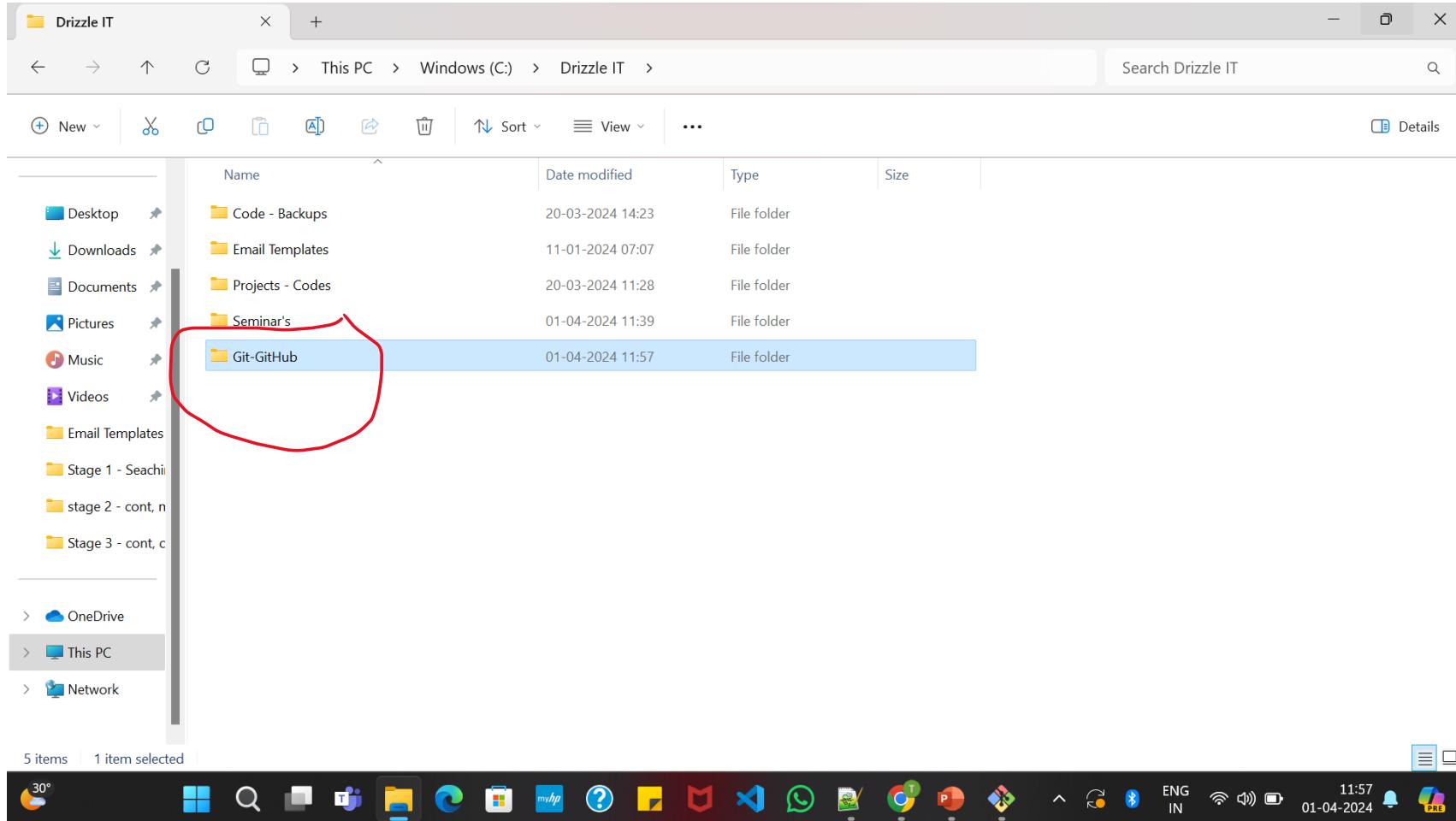
How Git differs from other VCS?

- Git – Distributed VCS
- Git – Open source
- Git – Command Line / GUI
- Cloud Hosting
- GitHub

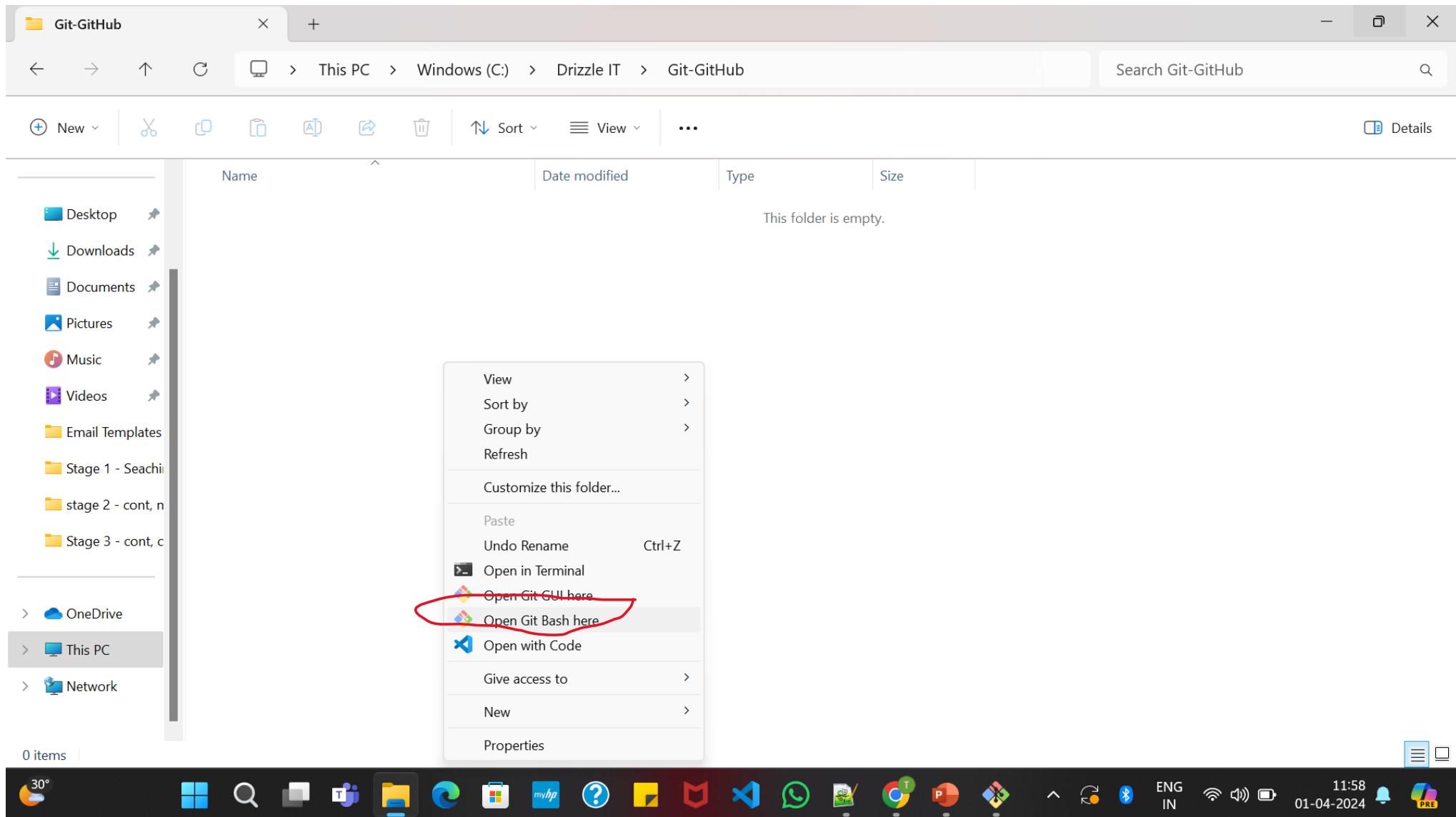
Git Basics – Initializing a Git repository

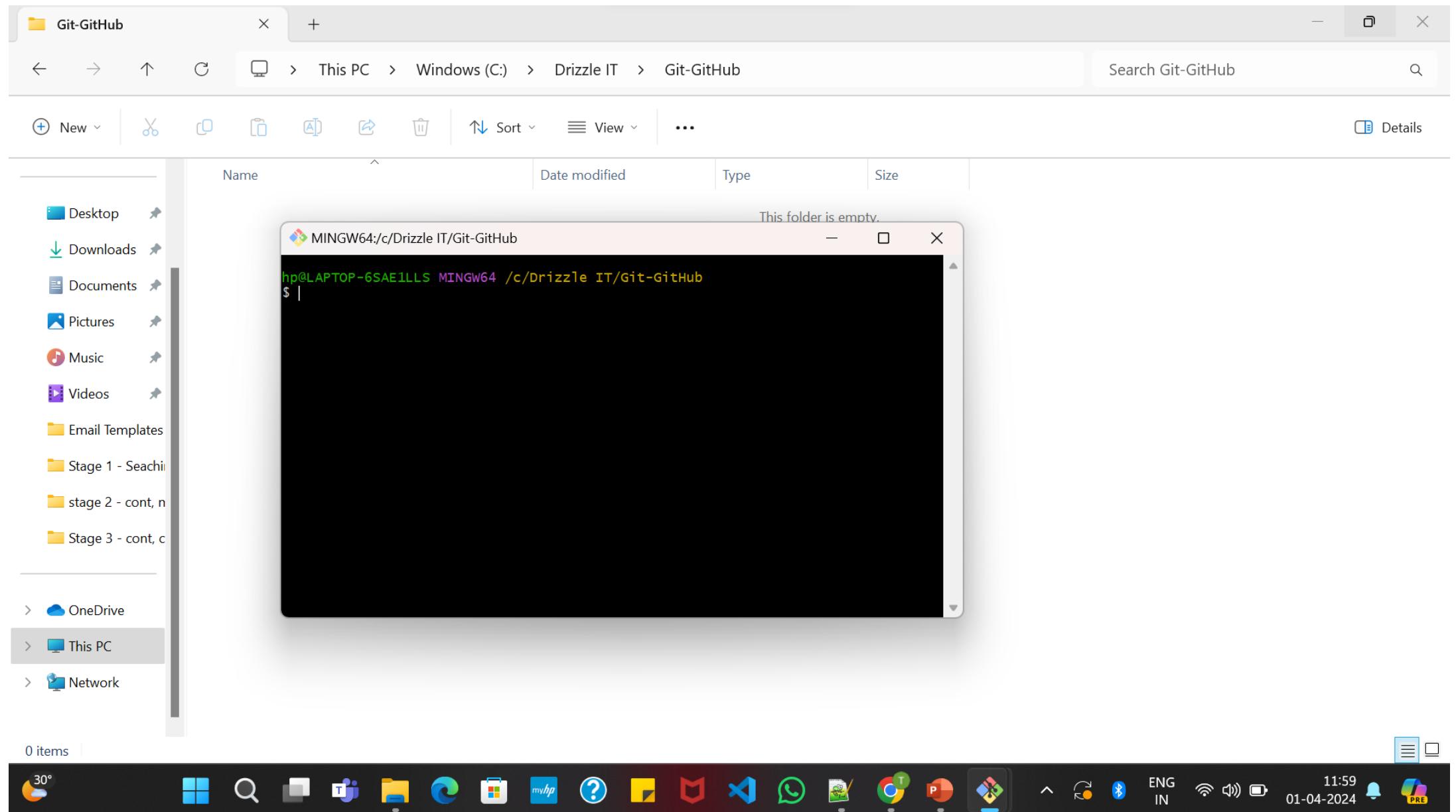
- Command – `git init`
 - This is usually the very first command you will ever run when starting with a new project.
 - Without this command no git command would ever work.
 - The `git init` command creates a new Git repository for you.
- 1] Convert an existing, unversioned project to git repository.
 - 2] Initialise a new empty git repository.

- Executing git init creates a .git subdirectory in the current working directory, which contains all the necessary git metadata from new repository.
- For this create a new folder :

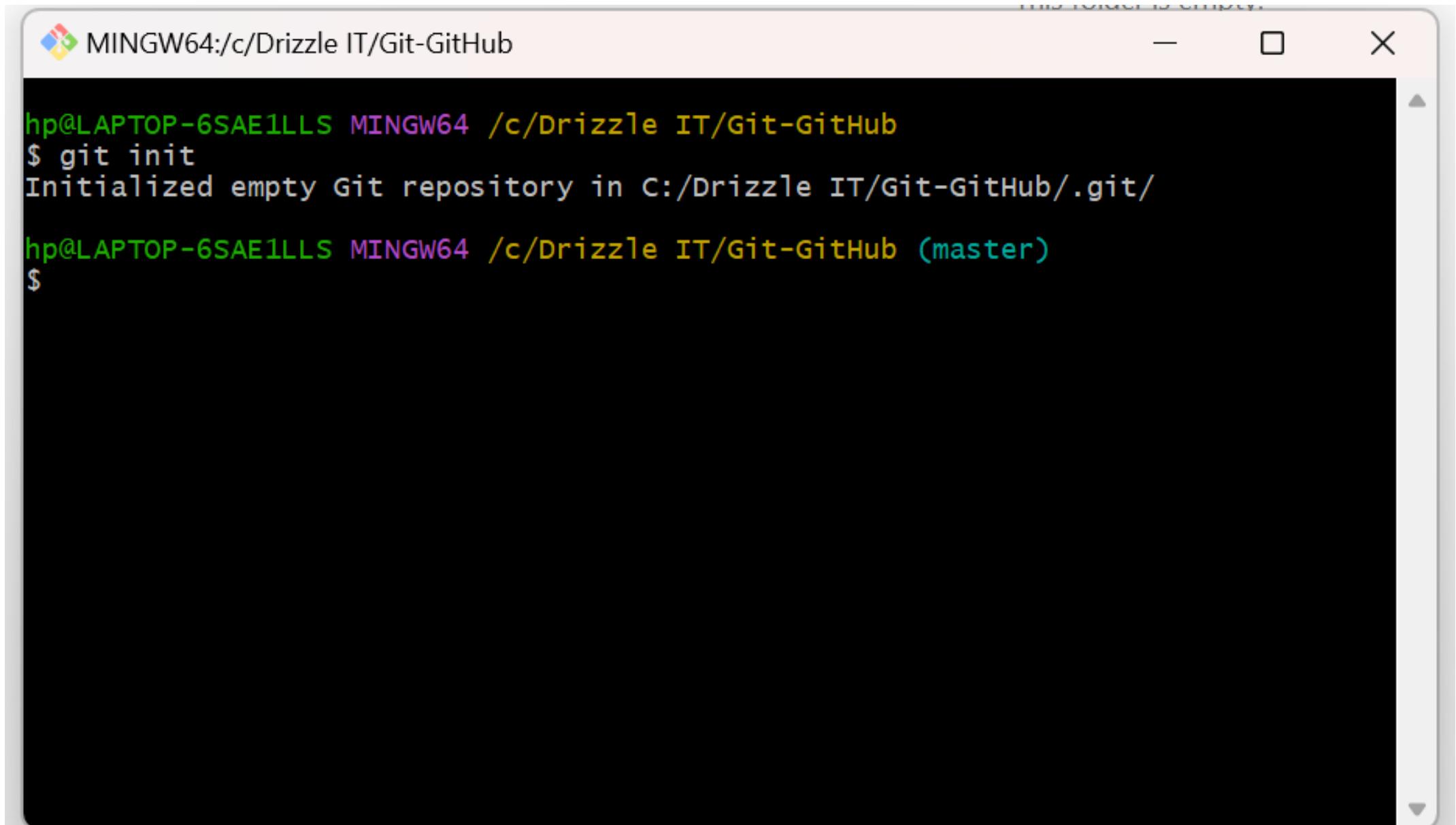


- Open gitbash in the current directory.





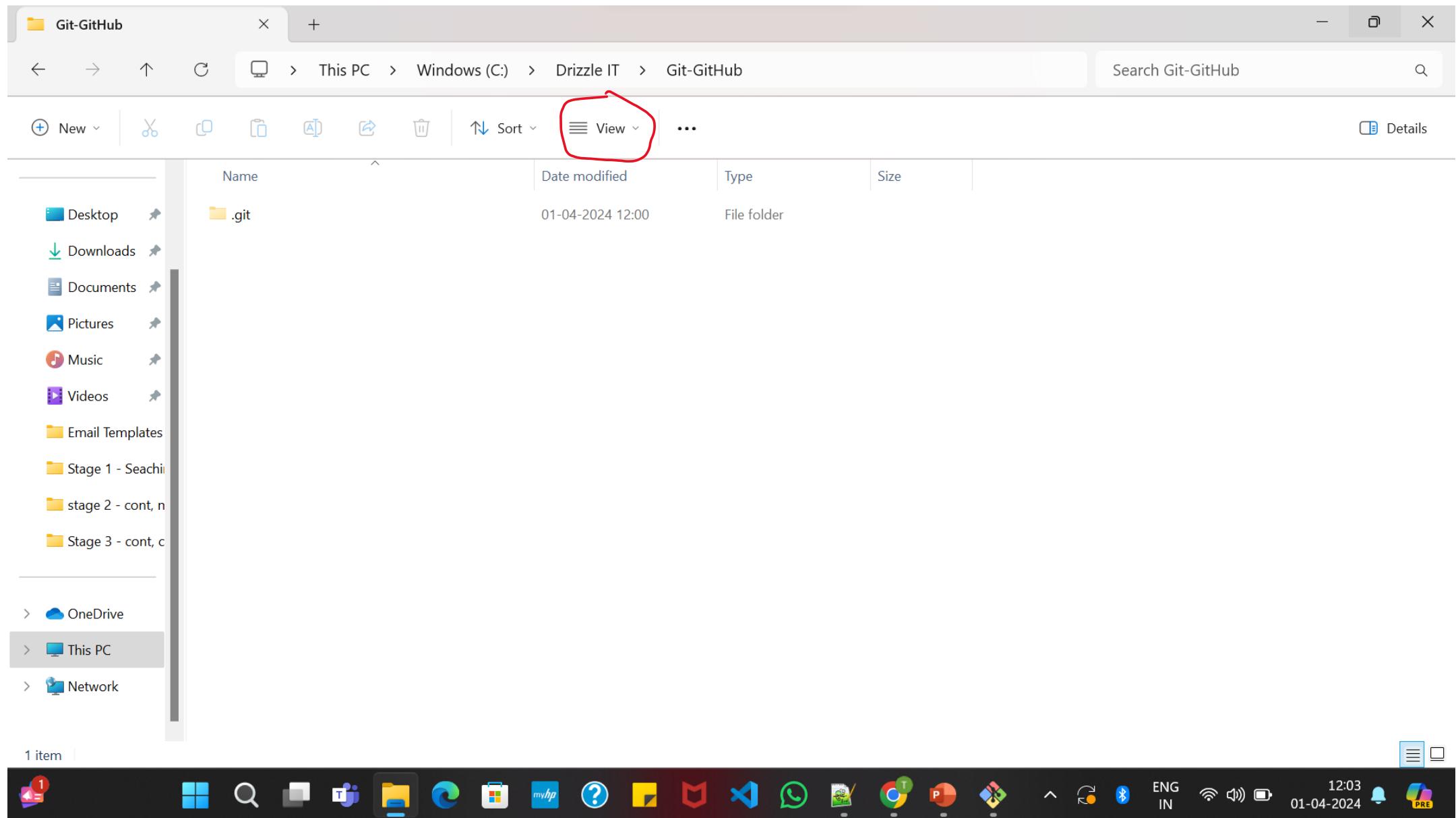
- New empty repo is been initialized.



The screenshot shows a terminal window titled "MINGW64:/c/Drizzle IT/Git-GitHub". The window contains the following text:

```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub
$ git init
Initialized empty Git repository in C:/Drizzle IT/Git-GitHub/.git/
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$
```

- View → Untick show hidden → Now we will be able to see the file



Adding files and committing to the repository

- Git Add – Adds file folder to staging area.
- Command – `git add filename`
- Git Add –All files:
 - 1] `git add -all`
 - 2] `git add -A`
 - 3] `git add .`

- Git Commit – Adds staged changes to local repository
- Command:

`git commit`

`git commit -m "message"`

```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub
$ git init
Initialized empty Git repository in C:/Drizzle IT/Git-GitHub/.git/
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ touch 1.txt

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ git add 1.txt

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ touch 2.txt 3.py

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ git add .

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ git commit -m "Add initial files"
[master (root-commit) 8824fc2] Add initial files
 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 1.txt
 create mode 100644 2.txt
 create mode 100644 3.py

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$
```

Repository – History & Status

- Status – Status of your staged areas. (`git status`)
- History – History of commits. (`git log`)
- Open working directory -> open Git bash -> git init -> add files -> status -> commit

```
MINGW64:/c/Drizzle IT/Git-GitHub
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub
$ git init
Initialized empty Git repository in C:/Drizzle IT/Git-GitHub/.git/
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ touch 1.txt 2.cpp 3.py
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ git add .
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ git status
On branch master
No commits yet
Changes to be committed:
(use "git rm --cached <file>..." to unstage)
  new file: 1.txt
  new file: 2.cpp
  new file: 3.py

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ git commit -m "2nd stage"
[master (root-commit) abe265d] 2nd stage
 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 1.txt
 create mode 100644 2.cpp
 create mode 100644 3.py
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ git status
On branch master
nothing to commit, working tree clean
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ git log
commit abe265da4682dce2a57f7e651d88d4a656a76666 (HEAD -> master)
Author: Tejal Pathak <pathak23tejal@gmail.com>
Date: Mon Apr 1 12:24:17 2024 +0530

 2nd stage
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ |
```



12:25
01-04-2024



Branching – What is branching and why do we need branching ?

- Default branch of Git is **Master Branch**.
- How dose Branching help ?
 - Try a new feature
 - Collaboration

Creating and switching between branches

- Create branch : `git branch <branch name>`
- Switching branch : `git checkout <branch name>`
- Note – we cannot create a new branch unless the stage is all cleared.

```
MINGW64:/c/Drizzle IT/Git-GitHub
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub
$ git init
Initialized empty Git repository in C:/Drizzle IT/Git-GitHub/.git/
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ touch main.txt

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ git branch new_branch
fatal: not a valid object name: 'master'

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ git add .

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ git commit -m "branching"
[master (root-commit) 8b1726e] branching
 1 file changed, 1 insertion(+)
 create mode 100644 main.txt

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ git branch new_branch

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ git checkout new_branch
Switched to branch 'new_branch'

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (new_branch)
$ touch main2.txt

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (new_branch)
$ git add .

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (new_branch)
$ git commit -m "newBranch"
[new_branch 3515f4c] newBranch
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 main2.txt

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (new_branch)
$ git checkout master
Switched to branch 'master'

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ |
```



Branch logging and deleting a branch

- Delete branch :
 - `git branch -d <branch name>`
 - Error if branch is not fully merged
- Force delete branch:
 - `git branch -D <branch name>`

```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ git log
commit 8b1726e7ba508e0eb7de26b2b0c23666c229b63b (HEAD -> master)
Author: Tejal Pathak <pathak23tejal@gmail.com>
Date:   Mon Apr 1 12:43:03 2024 +0530
```

branching

```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ git branch -d new_branch
error: the branch 'new_branch' is not fully merged
hint: If you are sure you want to delete it, run 'git branch -D new_branch'
hint: Disable this message with "git config advice.forceDeleteBranch false"
```

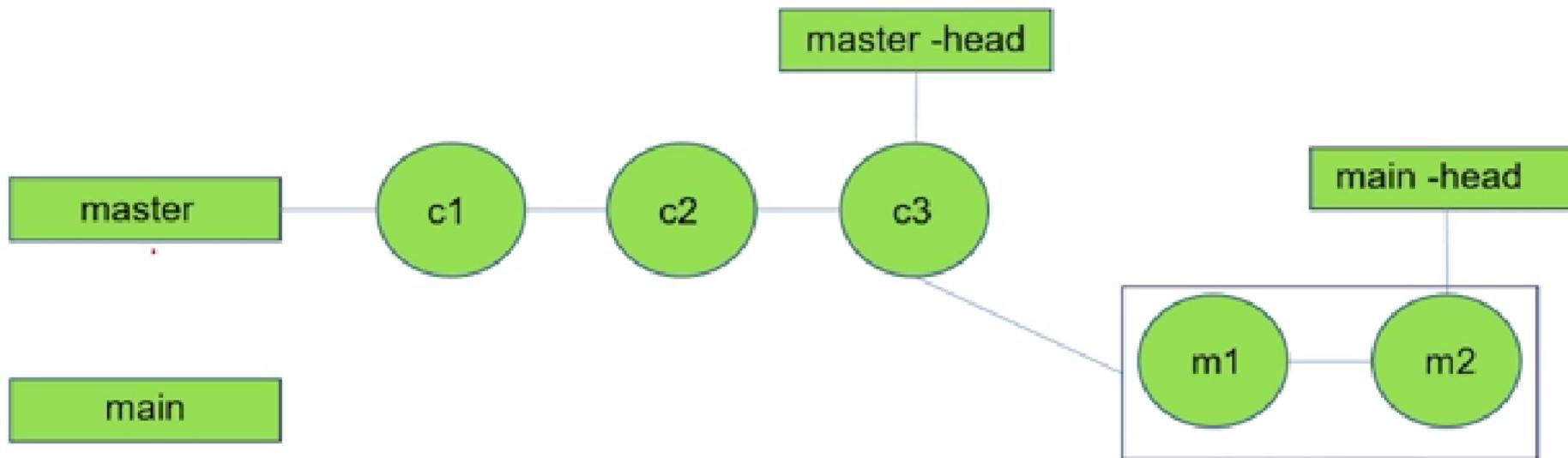
```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ git branch -D new_branch
Deleted branch new_branch (was 3515f4c).
```

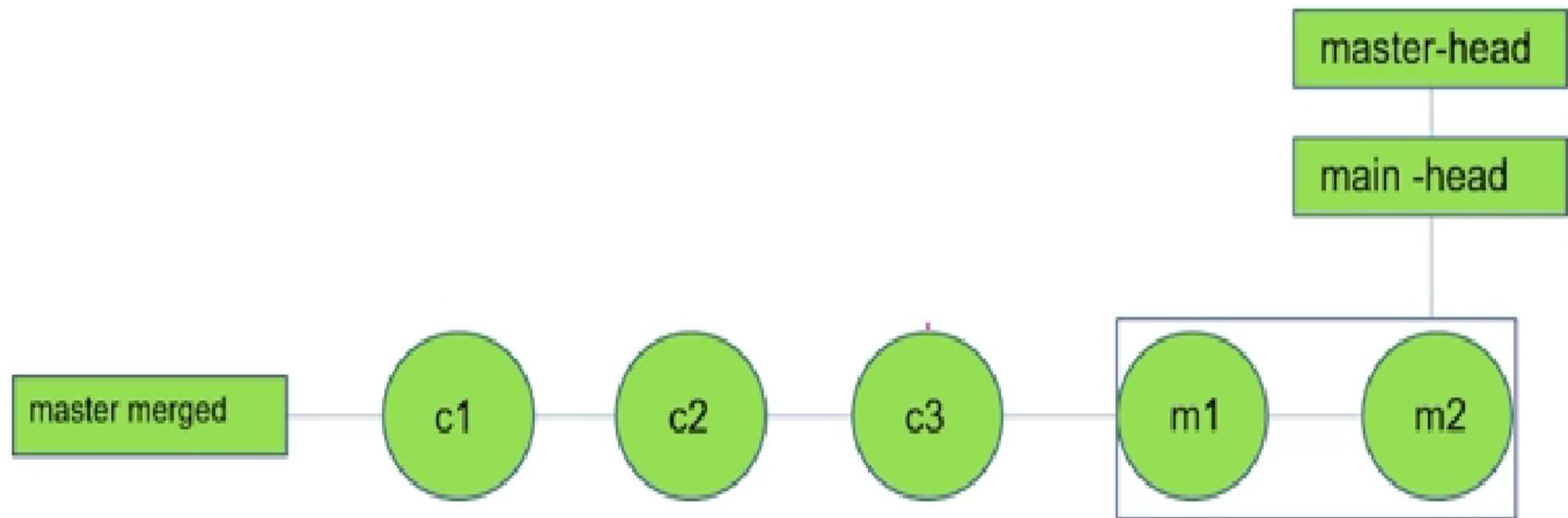
```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ |
```

Merging: Basic merging

Basic Merging

æG





MINGW64:/c/Drizzle IT/Git-GitHub

```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub
$ git init
Initialized empty Git repository in C:/Drizzle IT/Git-GitHub/.git/
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ touch index.html

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ git add .

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ git commit -m "First Commit"
[master (root-commit) 3c256c9] First Commit
 1 file changed, 1 insertion(+)
 create mode 100644 index.html

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ git checkout -b main
Switched to a new branch 'main'

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (main)
$ touch contact.html

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (main)
$ git add .

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (main)
$ git commit -m "First in contact"
[main 180abd6] First in contact
 1 file changed, 1 insertion(+)
 create mode 100644 contact.html

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (main)
$ git checkout master
Switched to branch 'master'

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ git merge main
Updating 3c256c9..180abd6
Fast-forward
 contact.html | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 contact.html

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Git-GitHub (master)
$ git log --oneline
180abd6 (HEAD -> master, main) First in contact
3c256c9 First Commit
```



13:14
01-04-2024

Merge Conflicts

- A merge conflict is an event that occurs when git is unable to automatically resolve differences in code between two commits.

Rebasing – What is rebasing and how it is different from merging.

- Rebasing is changing the base of your branch from one commit to another making it appear as if you'd created your branch from a different commit.
- The output which we get from rebasing and merging is the same, but the process is quite different.
- Rebasing cuts all the previous commits and add it to the last commit of the master branch.
- But, in merging last commit of the main branch is merged to the last commit of the master branch.
- Final product of both is same, but how they are working internally is different.

Rebasing a branch

- Command : `git rebase <branch_name>`

```
PS C:\Drizzle IT\rebasing> git checkout -b main
Switched to a new branch 'main'
PS C:\Drizzle IT\rebasing> touch contact.html
PS C:\Drizzle IT\rebasing> git add .
PS C:\Drizzle IT\rebasing> git commit -m "m1"
[main 8310ddd] m1
 1 file changed, 1 insertion(+)
  create mode 100644 contact.html
PS C:\Drizzle IT\rebasing> git add .
PS C:\Drizzle IT\rebasing> git commit -m "m2"
[main 44db893] m2
 1 file changed, 1 insertion(+)
PS C:\Drizzle IT\rebasing> git checkout master
Switched to branch 'master'
PS C:\Drizzle IT\rebasing> git add .
PS C:\Drizzle IT\rebasing> git commit -m "C3"
[master 09a3bf0] C3
 1 file changed, 2 insertions(+), 1 deletion(-)
PS C:\Drizzle IT\rebasing> git rebase main
Successfully rebased and updated refs/heads/master.
PS C:\Drizzle IT\rebasing> git log
```

Advantages and disadvantages of rebasing

- Advantages :
 - The log history looks more clean as every commit even the commit in the other branch will be added together. So you can have a look at all the commits of all branches together.
- Disadvantages :
 - Never rebase the commits that exists outside your repository.

Stashing

- Stashing takes modifies tracked files, stage changes, and saves them on stack of unfinished changes that you can reapply at any time.
- Command : `git stash`
- Before your using `git stash` you need to put all the files into staging area. (`git add .`)
- So the concept is before actually doing the commit in second file we want to commit new changes in 1st file without even loosing the data, we use stashing.

```
PS C:\Drizzle IT\Code - Backups\Github Seminar Demo\Stashing> touch index.html
PS C:\Drizzle IT\Code - Backups\Github Seminar Demo\Stashing> git add .
PS C:\Drizzle IT\Code - Backups\Github Seminar Demo\Stashing> git commit -m "first"
[master ee67d7f] first
 1 file changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 contact.html
PS C:\Drizzle IT\Code - Backups\Github Seminar Demo\Stashing> touch contact.html
PS C:\Drizzle IT\Code - Backups\Github Seminar Demo\Stashing> git add .
PS C:\Drizzle IT\Code - Backups\Github Seminar Demo\Stashing> git stash
Saved working directory and index state WIP on master: ee67d7f first
PS C:\Drizzle IT\Code - Backups\Github Seminar Demo\Stashing> git add .
PS C:\Drizzle IT\Code - Backups\Github Seminar Demo\Stashing> git commit -m "second"
[master cae9ccc] second
 1 file changed, 2 insertions(+), 1 deletion(-)
PS C:\Drizzle IT\Code - Backups\Github Seminar Demo\Stashing>
```

- git stash list – gives list of all stashes
- git stash apply stash@{}
- git stash drop stash{}

E.g.

```
PS C:\Drizzle IT\Code - Backups\Github Seminar Demo\Stashing> git stash list
stash@{0}: WIP on master: ee67d7f first
PS C:\Drizzle IT\Code - Backups\Github Seminar Demo\Stashing>
```

Reflog

- Something similar to git log but not exactly git log.
- Command : **git reflog**
- The reflog is basically the list of commits that HEAD has previously pointed to or you can say undo history of repository.

```
PS C:\Drizzle IT\Code - Backups\Github Seminar Demo\Stashing> git reflog
cae9ccc (HEAD -> master) HEAD@{0}: commit: second
ee67d7f HEAD@{1}: reset: moving to HEAD
ee67d7f HEAD@{2}: commit: first
56631e1 HEAD@{3}: commit: c2
e3f591a HEAD@{4}: commit (initial): c1
PS C:\Drizzle IT\Code - Backups\Github Seminar Demo\Stashing>
```

GitHub: What is GitHub and how it is different from git

- GitHub is cloud based hosting services that lets you manage the git repositories.
- Git vs. GitHub

Git	GitHub
Software	Service
Locally installed	Cloud based
Command line interface	Graphical user interface

Creating a GitHub repository

- Open your GitHub account.

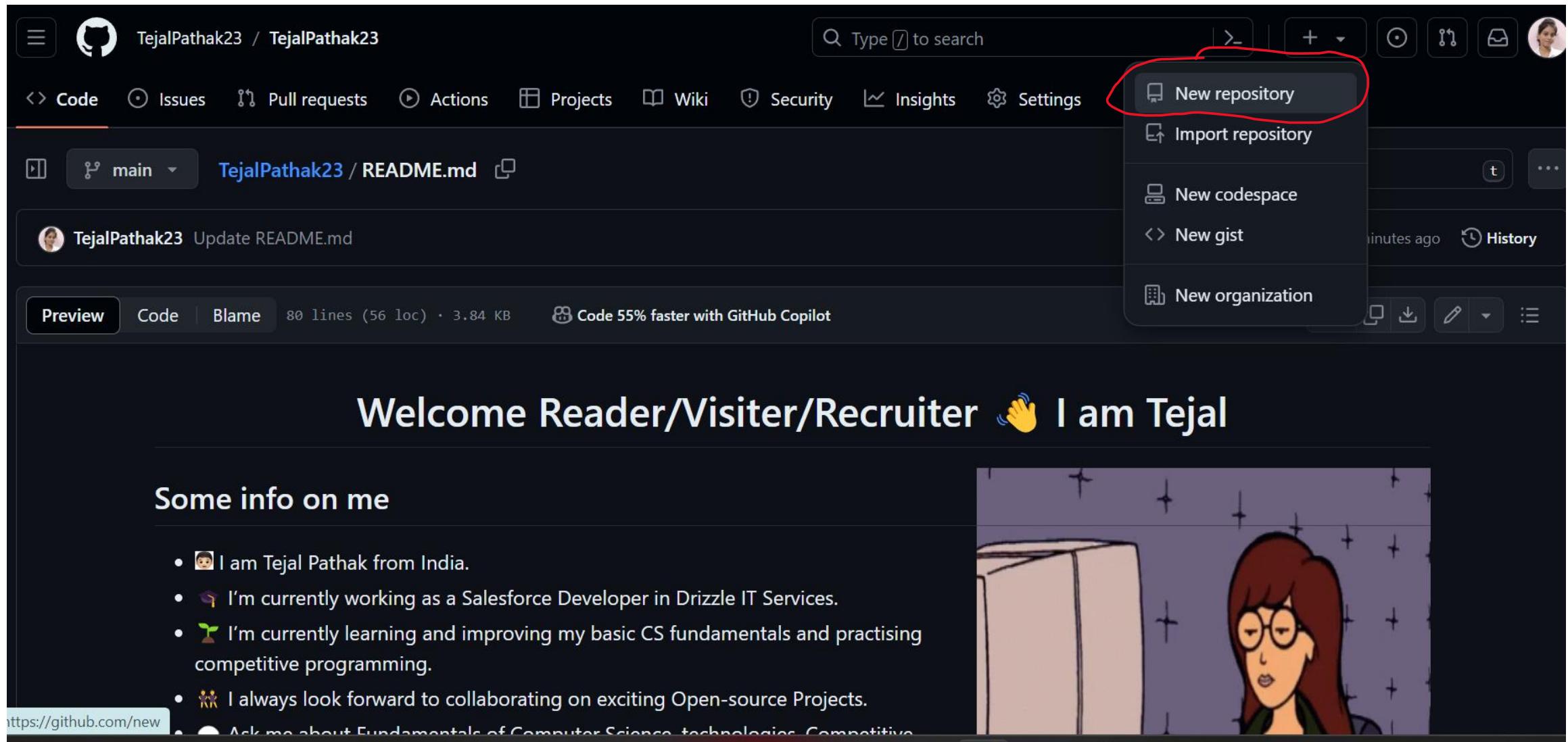
The screenshot shows a GitHub repository page for the user 'TejalPathak23'. The repository name is 'TejalPathak23 / TejalPathak23'. The 'Code' tab is selected, showing the 'main' branch. The README.md file contains the following content:

```
Welcome Reader/Visiter/Recruiter 🤝 I am Tejal

Some info on me
• 📸 I am Tejal Pathak from India.
• 💻 I'm currently working as a Salesforce Developer in Drizzle IT Services.
• 🚀 I'm currently learning and improving my basic CS fundamentals and practising competitive programming.
• 🌟 I always look forward to collaborating on exciting Open-source Projects.
• 💬 Ask me about Fundamentals of Computer Science, technologies, Competitive programming and life in general.
• 📧 How to reach me: Linkedin.
```

Below the README, there is a section titled 'Skills' with a 'Coding Languages' section. To the right of the text, there is a cartoon illustration of a woman with brown hair and glasses sitting at a desk, working on a computer.

- Click on new repository to create new repository.



The screenshot shows a GitHub repository page for 'TejalPathak23 / TejalPathak23'. The 'Code' tab is selected. In the top right corner, there is a dropdown menu with several options: 'New repository' (which is circled in red), 'Import repository', 'New codespace', 'New gist', and 'New organization'. Below the dropdown, there is a message: 'Welcome Reader/Visiter/Recruiter 🙌 I am Tejal'. On the left, there is a section titled 'Some info on me' with a bulleted list about the user. On the right, there is a cartoon illustration of a woman with brown hair and glasses.

TejalPathak23 / TejalPathak23

Type / to search

New repository

Import repository

New codespace

New gist

New organization

Welcome Reader/Visiter/Recruiter 🙌 I am Tejal

Some info on me

- I am Tejal Pathak from India.
- I'm currently working as a Salesforce Developer in Drizzle IT Services.
- I'm currently learning and improving my basic CS fundamentals and practising competitive programming.
- I always look forward to collaborating on exciting Open-source Projects.

https://github.com/new

Ask me about Fundamentals of Computer Science, technologies, Competitive

- Hence, this type of repository is been created.

The screenshot shows a GitHub repository page for 'Demo_About_Github'. The repository is private, as indicated by the 'Private' button next to the repository name. A red oval highlights the repository name 'Demo_About_Github'. The page displays basic repository statistics: 1 branch, 0 tags, 1 commit, and a README file. The README content is identical to the repository title. The 'About' section contains a brief description: 'This repo is made as a demo to git and github.' It also lists repository metrics: 0 stars, 1 watching, 0 forks, and 0 releases published. There is a link to 'Create a new release'.

TejalPathak23 / Demo_About_Github

Type ⌘ to search

Code Issues Pull requests Actions Projects Security Insights Settings

Demo_About_Github Private

Unwatch 1 Fork 0 Star 0

main 1 Branch 0 Tags

Go to file Add file Code

TejalPathak23 Initial commit 787e3b9 · now 1 Commits

README.md Initial commit now

README

Demo_About_Github

This repo is made as a demo to git and github.

About

This repo is made as a demo to git and github.

Readme Activity 0 stars 1 watching 0 forks

Releases

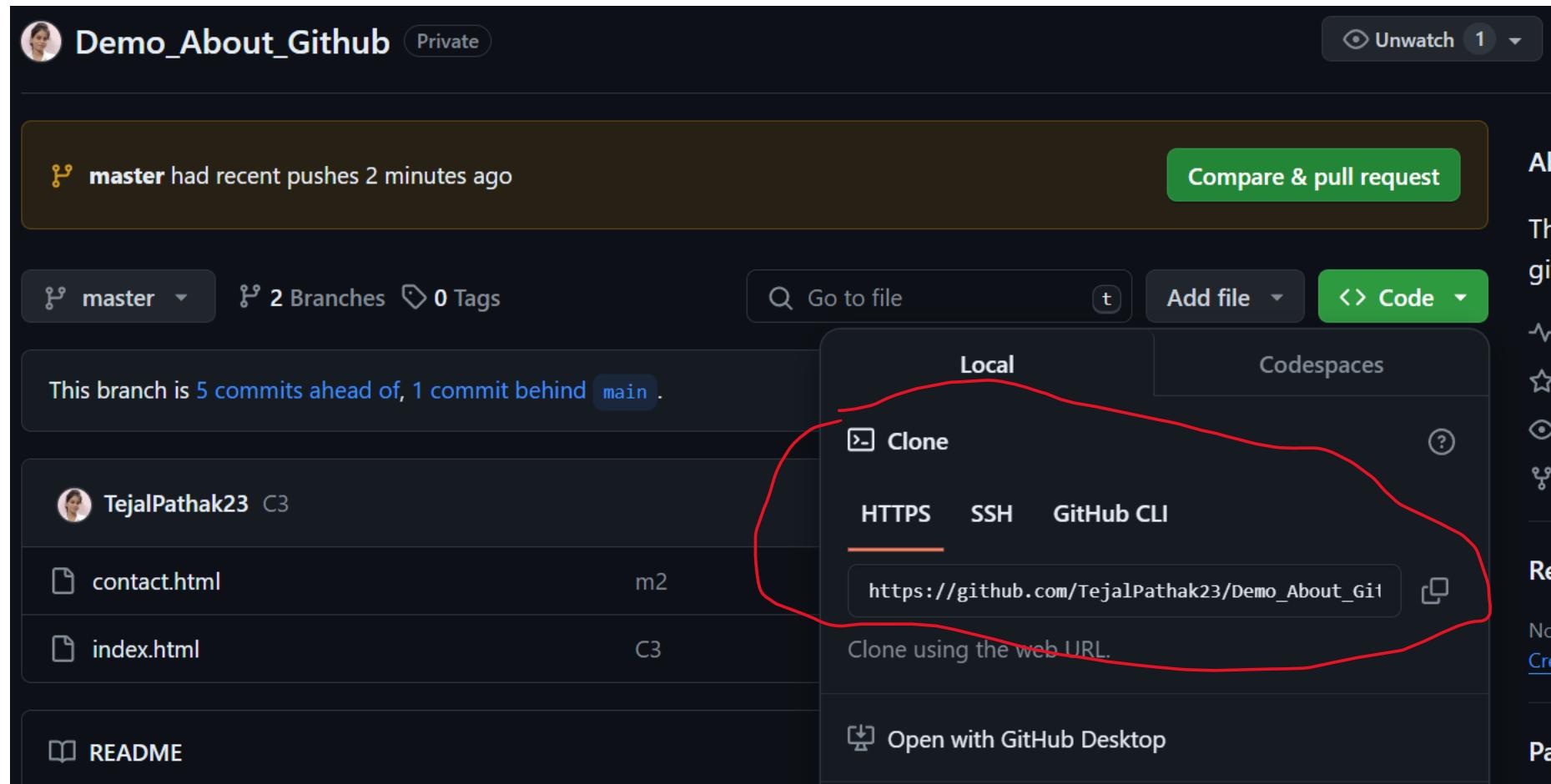
No releases published Create a new release

Uploading existing project on GitHub

git-ignore

- Commands
 - git init
 - git add .
 - git commit -m "message"
 - git remote add origin <remote URL>
 - git push origin master

- You will get that remote URL from : code -> Local -> HTTPS



```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/reba
sing (master)
$ git remote add origin https://github.com/TejalPathak23/Demo_About_Github.git

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/reba
sing (master)
$ git push origin master
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 8 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (15/15), 1.27 KiB | 434.00 KiB/s, done.
Total 15 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:     https://github.com/TejalPathak23/Demo_About_Github/pull/new/master
remote:
To https://github.com/TejalPathak23/Demo_About_Github.git
 * [new branch]      master -> master

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/reba
sing (master)
$ |
```

Working with remote repositories

```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/rebasing (master)
$ git remote
newOrigin
origin

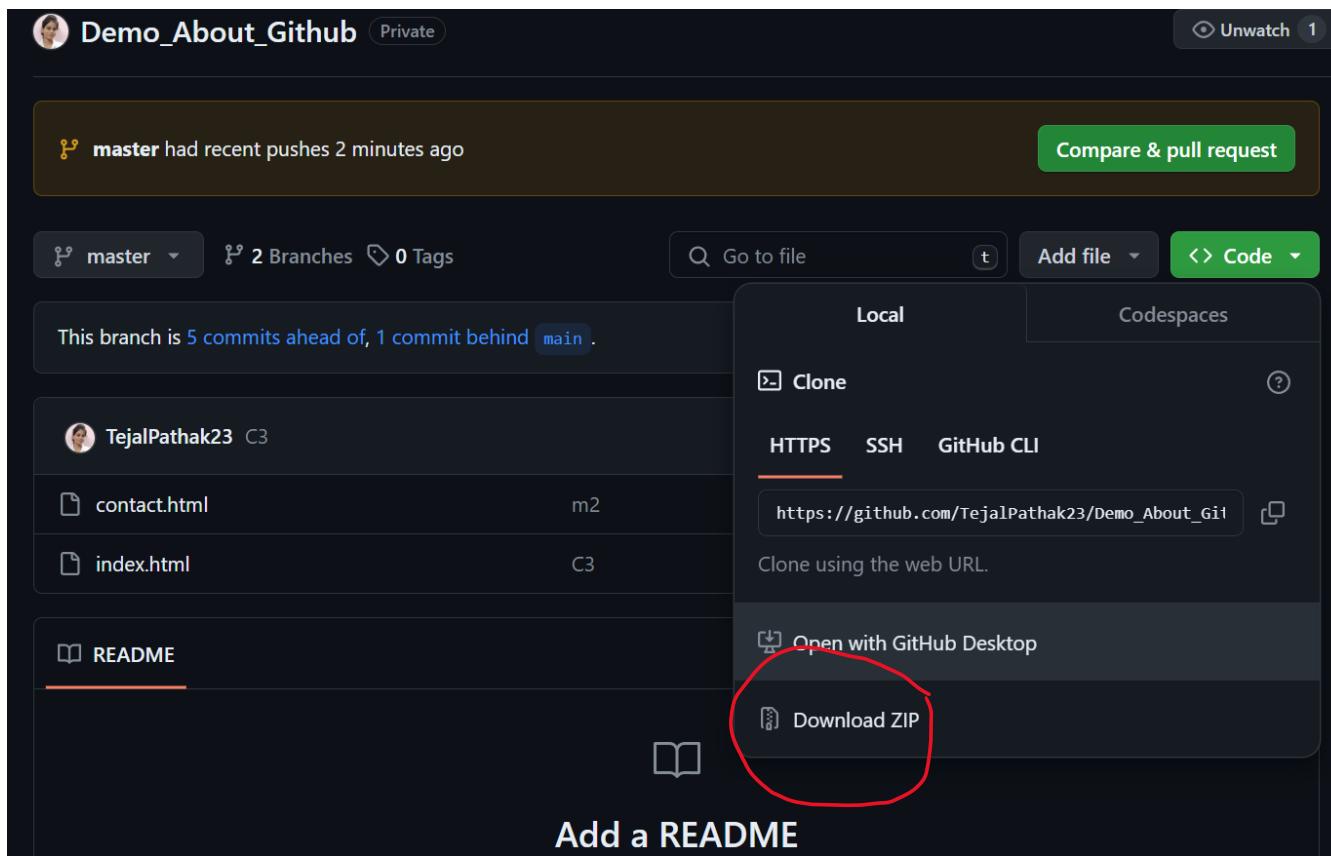
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/rebasing (master)
$ git remote -v
newOrigin      https://github.com/TejalPathak23/Demo_About_Github.git (fetch)
newOrigin      https://github.com/TejalPathak23/Demo_About_Github.git (push)
origin        https://github.com/TejalPathak23/Demo_About_Github.git (fetch)
origin        https://github.com/TejalPathak23/Demo_About_Github.git (push)

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/rebasing (master)
$ git push newOrigin master
Everything up-to-date

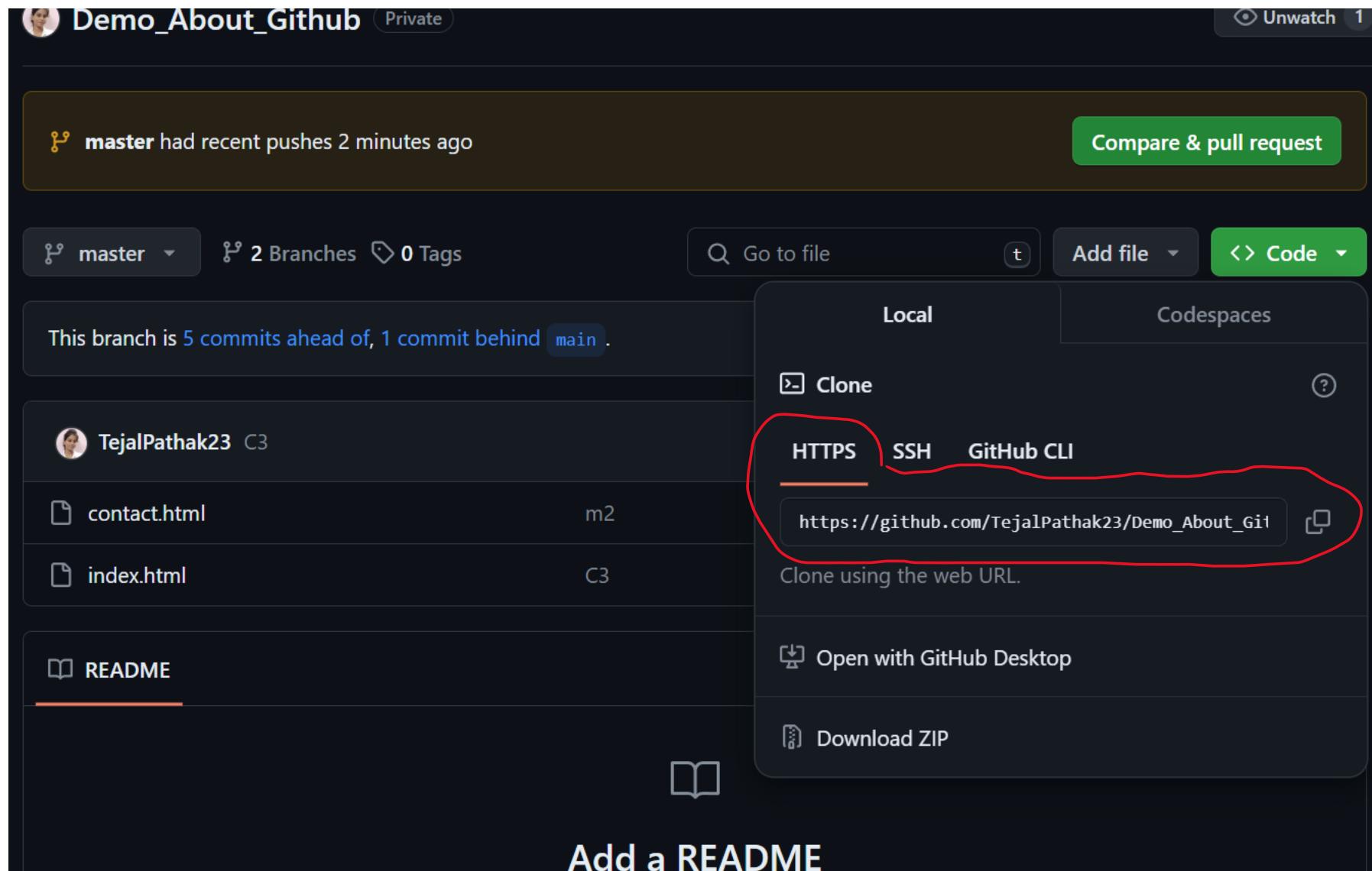
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/rebasing (master)
$ |
```

Cloning a repository

- Cloning helps us to work on anyone's code locally on our system.
- Option 1 : Download zip file.



- Option 2: Copy this URL

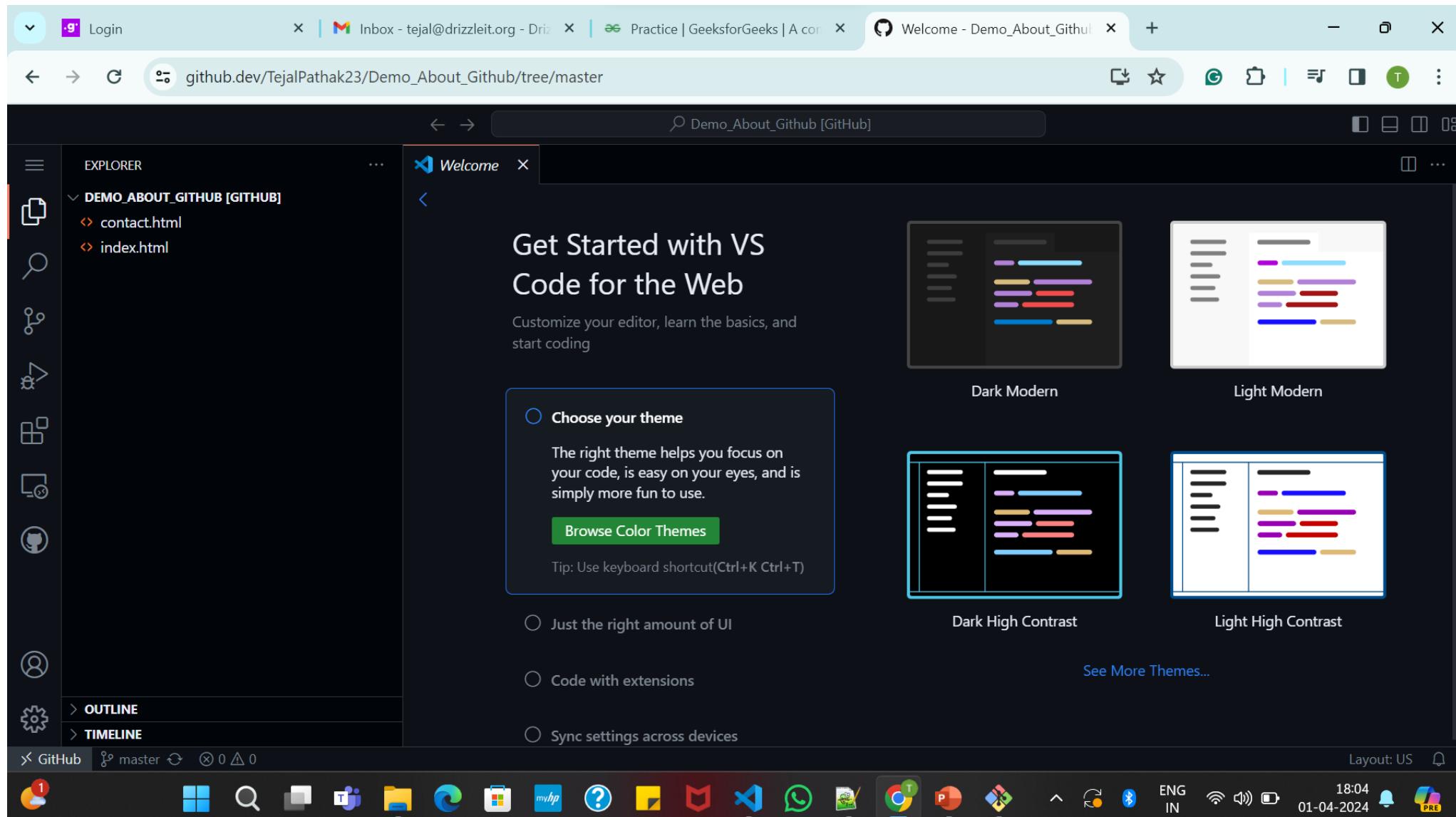


- Open gitbash -> type this command: **git clone <remote url>**

```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/rebasing (master)
$ git clone https://github.com/TejalPathak23/Demo_About_Github.git
Cloning into 'Demo_About_Github'...
remote: Enumerating objects: 18, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 18 (delta 1), reused 15 (delta 1), pack-reused 0
Receiving objects: 100% (18/18), done.
Resolving deltas: 100% (1/1), done.

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/rebasing (master)
$
```

- Option 3: go to that repository page -> click . -> automatically it will open a code editor similar to vs code.

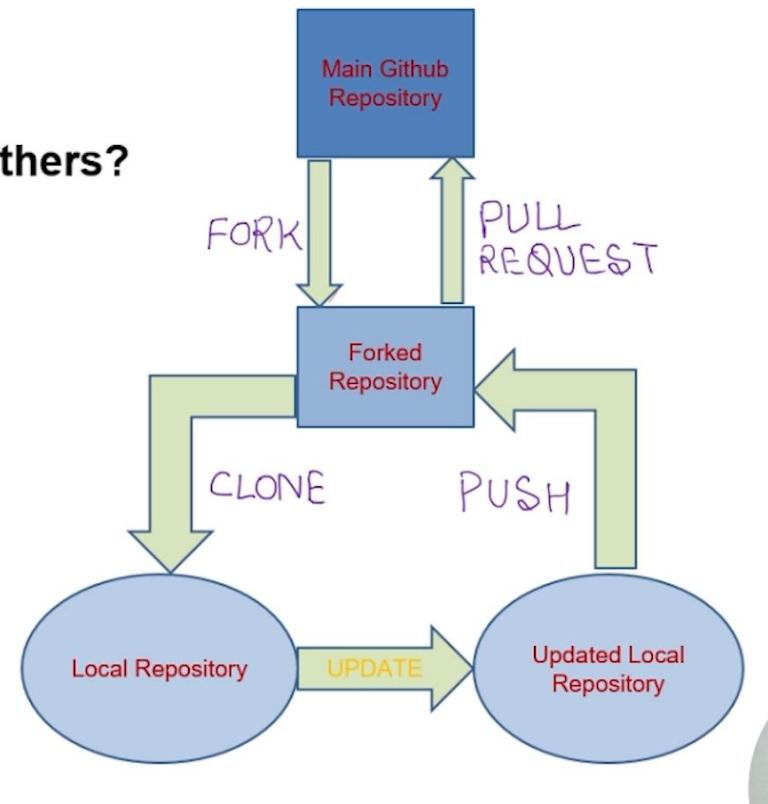


Forking a repository

- Forking is making a distinct separate copy of source code.



How to Collaborate with others?



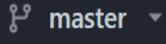
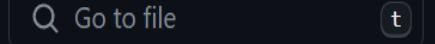
kevinohara80 / sfdc-trigger-framework

Type  to search

 Code  Issues 15  Pull requests 4  Actions  Projects  Wiki  Security  Insights

 sfdc-trigger-framework 

 Watch 101  Fork 496  Star 892

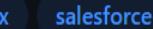
 master  3 Branches  5 Tags  Go to file  Add file  Code

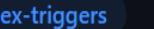
 kevinohara80 Merge pull request #39 from shindegirish/master  b7e36c7 · last year  70 Commits

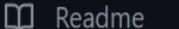
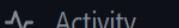
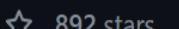
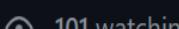
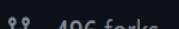
 src	updated api version	2 years ago
 .codeclimate.yml	fixing hash	6 years ago
 .gitignore	updating gitignore	8 years ago
 LICENSE	initial commit	11 years ago
 README.md	add codeclimate maintainability	6 years ago
 package.json	1.3.0	8 years ago
 readme.md	removing deprecated rule	6 years ago

 About

A minimal trigger framework for your Salesforce Apex Triggers

 salesforce  apex  salesforce-developers

 salesforce-api  apex-triggers

 Readme
 MIT license
 Activity
 892 stars
 101 watching
 496 forks
 Report repository

- After we forked this repository -> our version of the forked repository get's created.

The screenshot shows a GitHub repository page for the forked repository `sfdc-trigger-framework`. The repository was forked from `kevinohara80/sfdc-trigger-framework`.

Repository Details:

- Name:** sfdc-trigger-framework
- Owner:** TejalPathak23
- Status:** Public
- Forked From:** kevinohara80/sfdc-trigger-framework
- Branches:** master (1 Branch)
- Tags:** 0 Tags
- Last Commit:** b7e36c7 · last year
- Commits:** 70 Commits

Commit History:

File	Message	Date
src	updated api version	2 years ago
.codeclimate.yml	fixing hash	6 years ago
.gitignore	updating gitignore	8 years ago
LICENSE	initial commit	11 years ago

Repository Statistics:

- Readme
- MIT license
- Activity
- 0 stars
- 0 watching
- 0 forks

Releases:

- You can edit this repo, you can add & upload files, commit changes

The screenshot shows a GitHub repository page for 'sfdc-trigger-framework' owned by 'TejalPathak23'. The 'Code' tab is selected. A context menu is open over the repository name, with the 'Add file' option highlighted and circled in red. The menu also includes 'Create new file' and 'Upload files' options.

Code Pull requests Actions Projects Wiki Security Insights Settings

sfdc-trigger-framework Public
forked from [kevino'hara80/sfdc-trigger-framework](#)

master 1 Branch 0 Tags

This branch is up to date with [kevino'hara80/sfdc-trigger-framework:master](#).

Add file Create new file Upload files

About

A minimal trigger framework for your Salesforce Apex Triggers

Readme MIT license Activity 0 stars 0 watching 0 forks

Releases

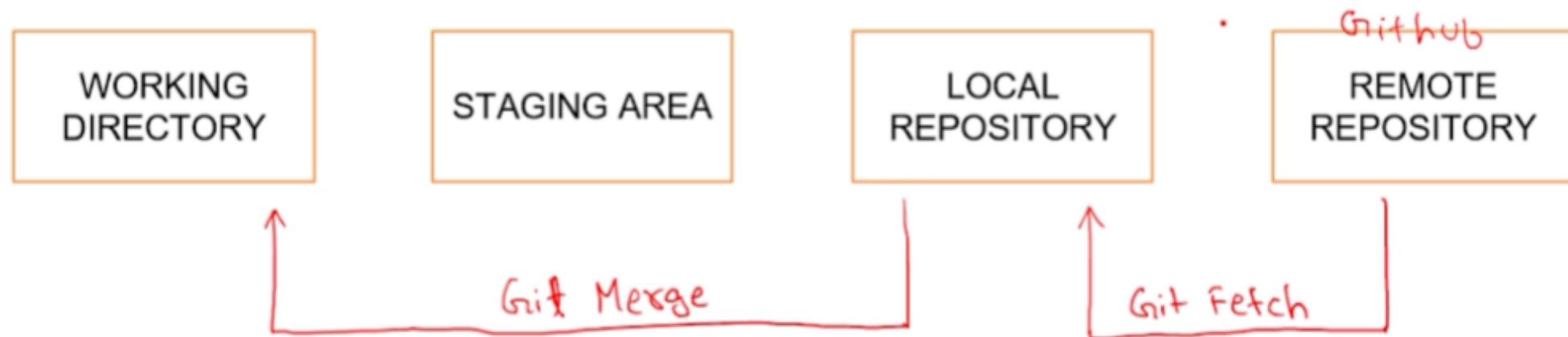
No releases published

https://github.com/TejalPathak23/sfdc-trigger-framework/new/master

1 18:15 01-04-2024 PRE

Fetching changes from GitHub

- To fetch the latest changes made into the repository.
- Command : **git fetch**
- With the help of this command it downloads, commits, files and refs from a remote repo into your local repo.



- Command : **git merge**

```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/reba
sing/Demo_About_Github (main)
$ git fetch
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 959 bytes | 137.00 KiB/s, done.
From https://github.com/TejalPathak23/Demo_About_Github
    4d1382d..65432a4  main      -> origin/main
```

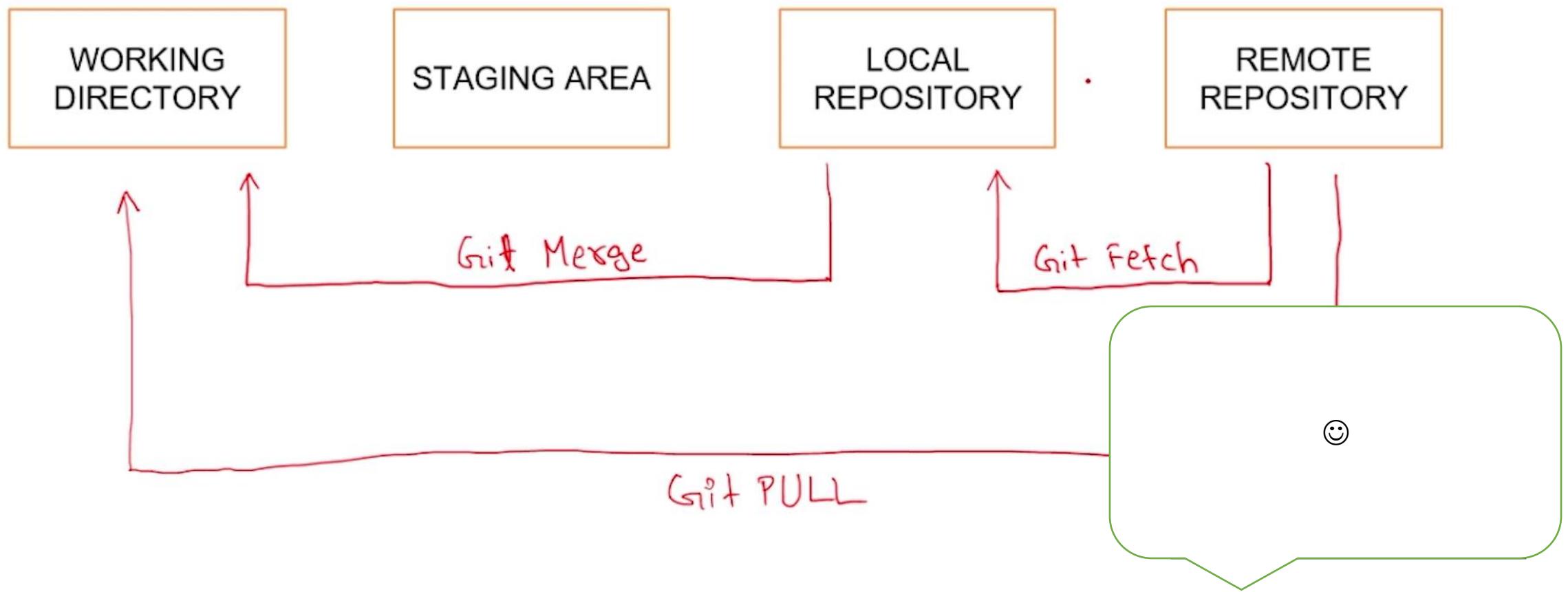
```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/reba
sing/Demo_About_Github (main)
$ git merge origin/main →
Updating 787e3b9..65432a4
Fast-forward
 1.cpp    | 1 +
 trial.py | 1 +
 2 files changed, 2 insertions(+)
 create mode 100644 1.cpp
 create mode 100644 trial.py
```

Instead of main sometimes u may need to use master, if u haven't include readme file

```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/reba
sing/Demo_About_Github (main)
$
```

Git fetch vs. git pull

- Git pull = git fetch + git merge



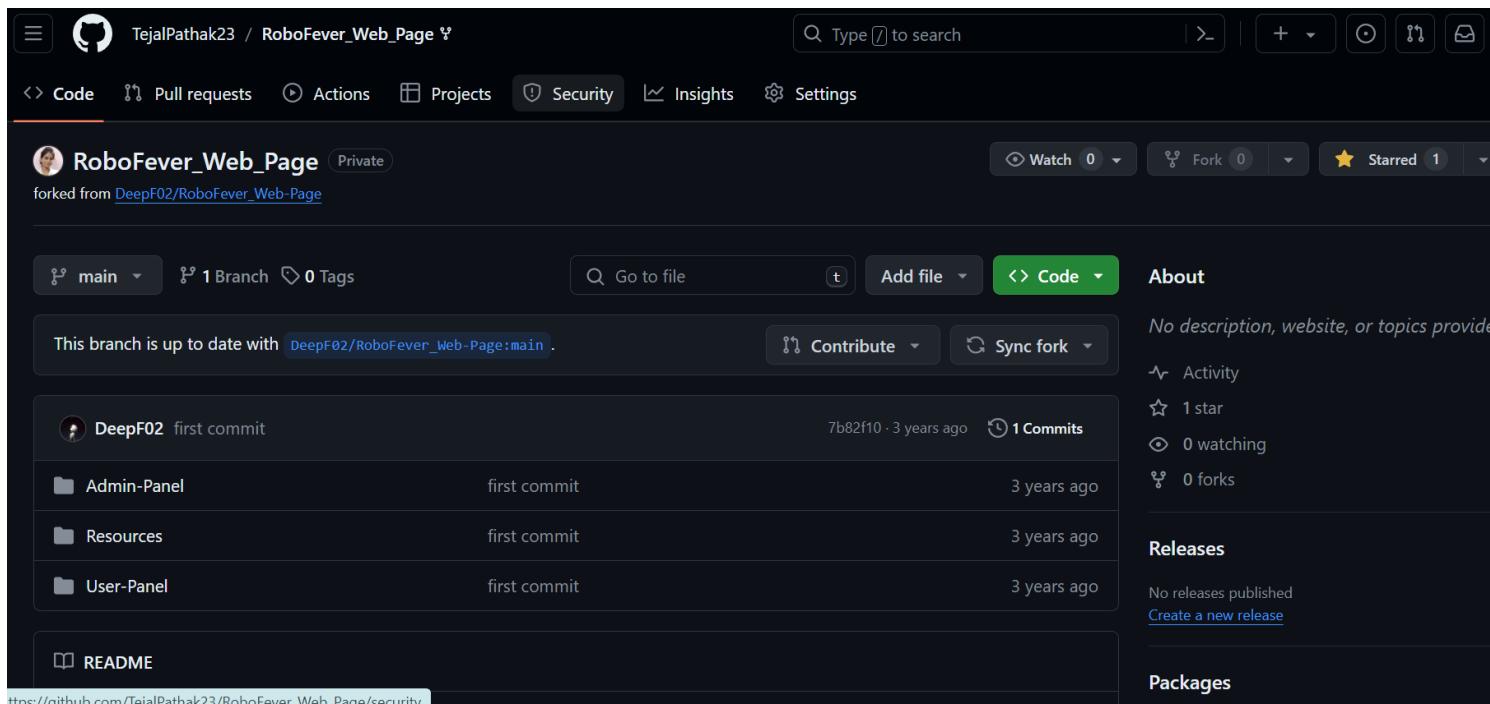
MINGW64:/c/Drizzle IT/Code - Backups/Github Seminar Demo/rebasing/Demo_About_Github

```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/rebasing/Demo_About_Github
$ git pull origin main
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (2/2), 879 bytes | 175.00 KiB/s, done.
From https://github.com/TejalPathak23/Demo_About_Github
 * branch           main      -> FETCH_HEAD
   65432a4..b0e309b  main      -> origin/main
Updating 65432a4..b0e309b
Fast-forward
 8.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 8.txt
```

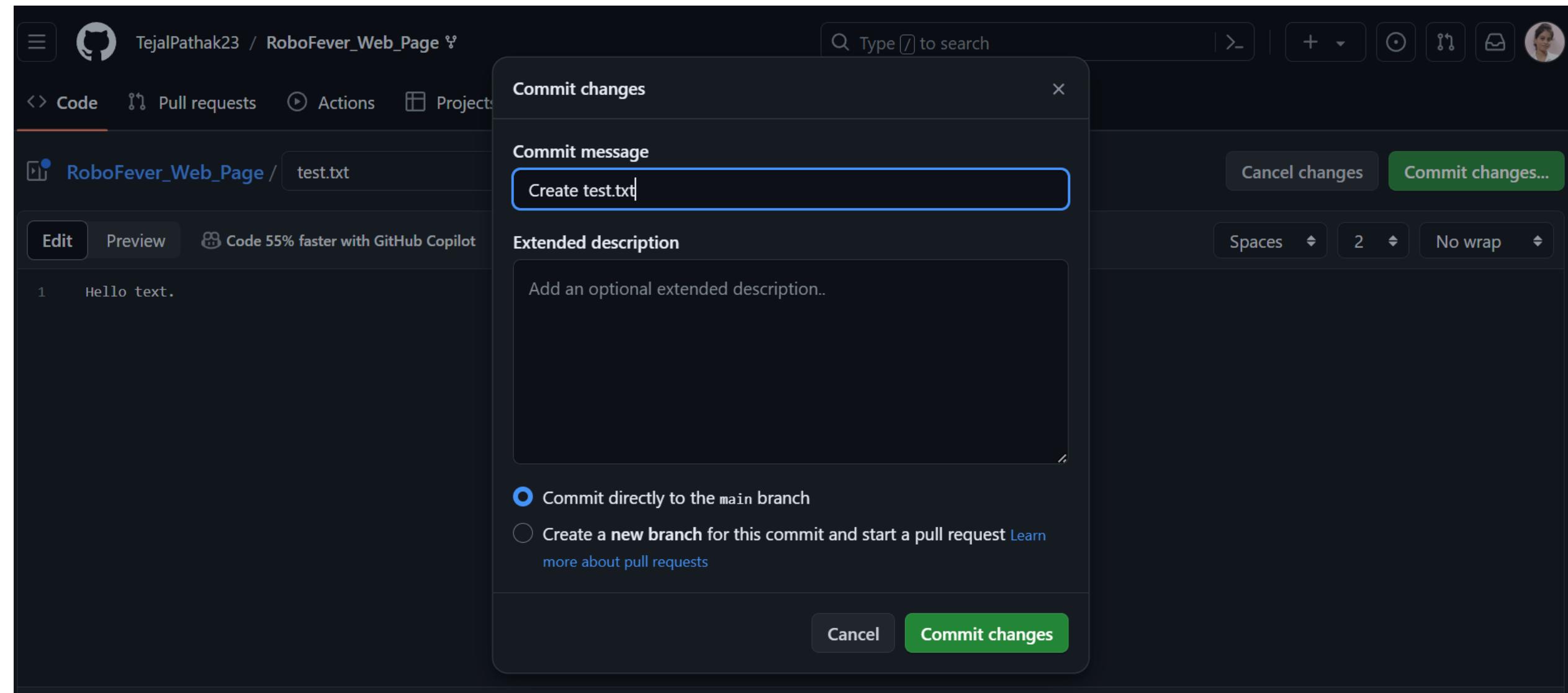
```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/rebasing/Demo_About_Github
$ |
```

Creating a Pull request

- Pull requests let you tell others about changes you've pushed to a branch in a repository on GitHub.
- Step 1 : Fork a repository.



- Step 2: Add new file and commit the changes.



A screenshot of a GitHub repository interface showing a commit dialog. The repository is named 'RoboFever_Web_Page' and contains a single file 'test.txt' with the content 'Hello text.' The commit message field contains 'Create test.txt'. The extended description field is empty. The 'Commit directly to the main branch' option is selected. The 'Commit changes...' button is highlighted in green.

TejalPathak23 / RoboFever_Web_Page

Code Pull requests Actions Projects

RoboFever_Web_Page / test.txt

Edit Preview Code 55% faster with GitHub Copilot

1 Hello text.

Commit changes

Commit message

Create test.txt

Extended description

Add an optional extended description..

Commit directly to the `main` branch

Create a **new branch** for this commit and start a pull request Learn more about pull requests

Cancel Commit changes

TejalPathak23 / RoboFever_Web_Page

Type ⌘ to search | > | + | ○ | ⓘ | 📁 | 🚪 | 🧑

Code Pull requests Actions Projects Security Insights Settings

main RoboFever_Web_Page / Go to file Add file ...

TejalPathak23 Create test.txt 15291c3 · 1 minute ago History

This branch is 1 commit ahead of DeepF02/RoboFever_Web-Page:main.

Contribute Sync fork

Name	Last commit message	Last commit date
Admin-Panel	first commit	3 years ago
Resources	first commit	3 years ago
User-Panel	first commit	3 years ago
test.txt	Create test.txt	1 minute ago

- Step 3: Click contribute -> Open pull request.

The screenshot shows a GitHub repository page for 'RoboFever_Web_Page' at the 'main' branch. The repository belongs to 'TejalPathak23'. The main navigation bar includes links for Code, Pull requests, Actions, Projects, Security, Insights, and Settings. A search bar is at the top right. Below the header, there's a breadcrumb trail showing 'main / RoboFever_Web_Page /'. A commit history is visible, with the most recent commit being 'Create test.txt' by 'TejalPathak23' (15291c3) just 3 minutes ago. A message indicates the branch is 1 commit ahead of 'DeepF02/RoboFever_Web-Page:main'. A prominent 'Contribute' button with a pull request icon is displayed. A tooltip-like overlay provides instructions: 'This branch is 1 commit ahead of DeepF02/RoboFever_Web-Page:main.' It suggests opening a pull request to contribute changes upstream. A large green 'Open pull request' button is at the bottom of this overlay. The bottom of the screen shows the Windows taskbar with various pinned icons and system status indicators.

greyTHR

Drizzle IT Services Mail

Practice | GeeksforGeeks | A com

RoboFever_Web_Page/ at main

github.com/TejalPathak23/RoboFever_Web_Page/tree/main

TejalPathak23 / RoboFever_Web_Page

Type ⌘ to search

Code Pull requests Actions Projects Security Insights Settings

main / RoboFever_Web_Page /

Go to file Add file ...

TejalPathak23 Create test.txt 15291c3 · 3 minutes ago History

This branch is 1 commit ahead of DeepF02/RoboFever_Web-Page:main.

Contribute Sync fork

Name	Last commit message	Last commit date
Admin-Panel	first commit	3 years ago
Resources	first commit	3 years ago
User-Panel	first commit	3 years ago
test.txt	Create test.txt	3 minutes ago

This branch is 1 commit ahead of DeepF02/RoboFever_Web-Page:main.

Open a pull request to contribute your changes upstream.

Open pull request

29°

https://github.com/TejalPathak23/RoboFever_Web_Page/pull/new/main

11:18 05-04-2024 ENG IN

PRE

greyTHR Drizzle IT Services Mail Practice | GeeksforGeeks | A cor Comparing DeepF02:main...Teja

github.com/DeepF02/RoboFever_Web-Page/compare/main...TejalPathak23:RoboFever_Web_Page:main

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#). Learn more about diff comparisons [here](#).

base repository: DeepF02/RoboFever_Web-Page base: main head repository: TejalPathak23/RoboFever_Web... compare: main

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

Add a title
Create test.txt

Add a description

Write Preview

Testing

Reviewers
No reviews

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone



11:20 05-04-2024 ENG IN

- Last Step: After we do the needful, we'll get this window.

The screenshot shows a GitHub pull request window for a repository named "RoboFever_Web-Page". The pull request has been merged, as indicated by the green "Open" button which is circled in red. The PR summary shows "TejalPathak23 wants to merge 1 commit into DeepF02:main from TejalPathak23:main". The commit details show a single commit from "Create test.txt" with a "Verified" status and hash "15291c3". A comment from "TejalPathak23" says "Testing" with a smiley face emoji. The merge status message at the bottom left says "This branch has no conflicts with the base branch" and "Merging can be performed automatically." It includes a "Merge pull request" button and links to "GitHub Desktop" and "command line instructions". On the right side, there are sections for "Reviewers" (No reviews), "Assignees" (No one—assign yourself), "Labels" (None yet), "Projects" (None yet), "Milestone" (No milestone), and "Development" (No development details shown). The bottom of the screen shows a Windows taskbar with various icons and system status.

greytHR Drizzle IT Services Mail Practice | GeeksforGeeks Create test.txt by TejalPat ChatGPT

github.com/DeepF02/RoboFever_Web-Page/pull/1

Create test.txt

Open TejalPathak23 wants to merge 1 commit into DeepF02:main from TejalPathak23:main

Conversation 0 Commits 1 Checks 0 Files changed 1 +1 -0

TejalPathak23 commented 6 minutes ago

Testing

15291c3 Verified

Add more commits by pushing to the main branch on TejalPathak23/RoboFever_Web_Page.

This branch has no conflicts with the base branch
Merging can be performed automatically.

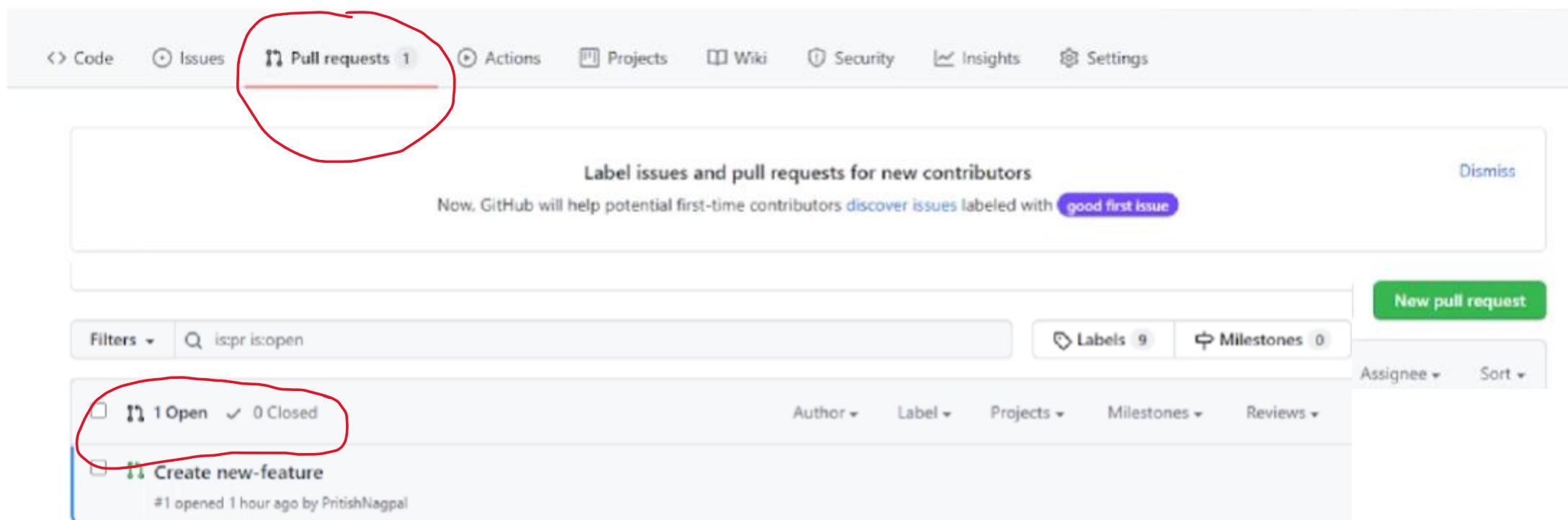
Merge pull request You can also open this in GitHub Desktop or view command line instructions.

Add a comment

Write Preview H B I E < > C M 15 ENG IN 11:26 05-04-2024 PRE

Merging a Pull request

- In the original repo we get this option.



- Then click-> Merge Pull request option , also you can give some feedbacks.

The screenshot shows a GitHub pull request interface for a repository named "Create new-feature". The pull request has been verified and has the commit hash "a96d116". It is assigned to "No one".

Annotations highlight two areas:

- A red circle surrounds the "Merge pull request" button, which is located in a green box containing status information. The text in the box says:
 - Continuous integration has not been set up (GitHub Actions and several other apps can be used to automatically catch bugs and enforce style).
 - This branch has no conflicts with the base branch (Merging can be performed automatically).
- A larger red circle surrounds the "Write" text input field, which contains the placeholder text "there is s|". Below the input field, there is a note: "Attach files by dragging & dropping, selecting or pasting them."

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

- The last step would be to confirm the merge, you have the option of closing the merge request, If you do so then the request cannot be merged.

Add more commits by pushing to the `master` branch on [PritishNagpal/learning-github](#).

Merge pull request #1 from [PritishNagpal/master](#)

Create new-feature

[Confirm merge](#) [Cancel](#)

Write Preview

H B I

Leave a comment

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

[Close pull request](#) [Comment](#)

① Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Creating issues on GitHub

- GitHub Issues are items you can create in a repository to plan, discuss and track work. Issues are simple to create and flexible to suit a variety of scenarios. You can use issues to track work, give or receive feedback, collaborate on ideas or tasks, and efficiently communicate with others.

- When you open any repository, you will see an issues tab in it.

The screenshot shows a GitHub repository page for 'Demo_About_Github'. The top navigation bar includes links for Code, Issues (which is circled in red), Pull requests, Actions, Projects, Security, Insights, and Settings. Below the navigation is the repository name 'Demo_About_Github' and its status as 'Private'. On the right, there are buttons for Unwatch (1), Fork (0), and Star (0). The main content area displays the repository's history with commits from 'TejalPathak23' and files like '1.cpp', '8.txt', 'README.md', and 'trial.py'. A sidebar on the right contains sections for 'About', 'Readme', 'Activity', '0 stars', '1 watching', '0 forks', and 'Releases'.

TejalPathak23 / Demo_About_Github

Type / to search

Issues

Code Pull requests Actions Projects Security Insights Settings

Demo_About_Github Private

Unwatch 1 Fork 0 Star 0

main 2 Branches 0 Tags

Go to file Add file

Code

About

This repo is made as a demo to git and github.

Readme

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

README

- Create new issue by clicking on new issues button.

The screenshot shows a GitHub repository named "TejalPathak23 / Demo_About_Github". The "Issues" tab is selected. A search bar at the top right contains the placeholder "Type ⌘ to search". Below the navigation bar are several links: Code, Issues (which is underlined), Pull requests, Actions, Projects, Security, Insights, and Settings. A "Filters" dropdown is set to "is:issue is:open". To the right of the filters are buttons for Labels (9) and Milestones (0). A prominent green button labeled "New issue" is located on the far right, with a red oval highlighting it. The main content area displays a single circular icon with a dot. Below the icon, the text "Welcome to issues!" is displayed in bold. A descriptive paragraph follows: "Issues are used to track todos, bugs, feature requests, and more. As issues are created, they'll appear here in a searchable and filterable list. To get started, you should [create an issue](#)".

- Write your issue -> Describe it -> Add suitable labels -> Submit

The screenshot shows the GitHub Issues interface. At the top, the repository is set to "Demo_About_Github". The "Issues" tab is selected. In the main area, there's a title input field with "Testing issues" and a description input field containing "Hey! I am learning the use of issues." Below the description is a rich text editor toolbar. A note says "Markdown is supported" and there's a file upload section. A reminder at the bottom links to "GitHub Community Guidelines". On the right, there are sections for "Assignees" (set to "No one—assign yourself") and "Labels". A red circle highlights the "Labels" section, which contains a list of predefined labels: bug, documentation, duplicate, enhancement, good first issue, help wanted, invalid, and question. A red circle also highlights the "Submit new issue" button at the bottom right.

Add a title

Testing issues

Add a description

Write Preview

Hey! I am learning the use of issues.

Markdown is supported

Paste, drop, or click to add files

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Assignees
No one—assign yourself

Labels

Apply labels to this issue

Filter labels

- bug Something isn't working
- documentation Improvements or additions to documentat...
- duplicate This issue or pull request already exists
- enhancement New feature or request
- good first issue Good for newcomers
- help wanted Extra attention is needed
- invalid This doesn't seem right
- question

Submit new issue

- The next time you will visit the repository, you will see open issues.

The screenshot shows a GitHub repository named "Demo_About_Github". The "Issues" tab is selected, displaying 1 open issue. A red circle highlights the "Filters" dropdown and the search bar containing "is:issue is:open". Below the filters, there are two items: "1 Open" and "Testing issues". The "Testing issues" item is associated with labels "enhancement" and "good first issue". A pro tip at the bottom suggests adding "no:label" to show everything without a label.

Code Issues 1 Pull requests Actions Projects Security Insights Settings

Filters Labels 9 Milestones 0 New issue

1 Open ✓ 0 Closed

Testing issues enhancement good first issue

#1 opened now by TejalPathak23

Author Label Projects Milestones Assignee Sort

💡 ProTip! Adding [no:label](#) will show everything without a label.

Deploying static sites on GitHub

- Static web pages/Static websites are those sites which do not need a server to run.
- *Note : The repository must be public in order to use this feature.*
- GitHub pages :
 - Deploy static websites.
 - Share working demo of your code.
 - Showcase your portfolio.

- Note: If you want to deploy your site using GitHub pages, then your main file should be named as **index.html**, otherwise that GitHub pages would not even work.

The screenshot shows a GitHub repository interface. At the top, the repository name is `Demo_About_Github`. The `Code` tab is selected. Below the tabs, there's a search bar and a navigation bar with buttons for `main`, `Add file`, and more. A commit history is displayed, showing the creation of various files: `1.cpp`, `8.txt`, `README.md`, `index.html` (which is circled in red), and `trial.py`. The `index.html` file was created just now. The repository description at the bottom states: `Demo_About_Github` and `This repo is made as a demo to git and github.`

Name	Last commit message	Last commit date
<code>1.cpp</code>	Create 1.cpp	4 days ago
<code>8.txt</code>	Create 8.txt	4 days ago
<code>README.md</code>	Initial commit	4 days ago
<code>index.html</code>	Create index.html	now
<code>trial.py</code>	python code	4 days ago

- Go to settings -> pages

The screenshot shows the GitHub repository settings page for 'Demo_About_Github'. The 'General' tab is selected. A red circle highlights the 'Settings' tab in the top navigation bar. Another red circle highlights the 'Pages' tab in the left sidebar.

General

Repository name: Demo_About_Github [Rename](#)

Template repository
Template repositories let users generate new repositories with the same directory structure and files. [Learn more about template repositories.](#)

Require contributors to sign off on web-based commits
Enabling this setting will require contributors to sign off on commits made through GitHub's web interface. Signing off is a way for contributors to affirm that their commit complies with the repository's terms, commonly the [Developer Certificate of Origin \(DCO\)](#). [Learn more about signing off on commits.](#)

Default branch

The default branch is considered the "base" branch in your repository, against which all pull requests and code commits are automatically made, unless you specify a different branch.

main [Edit](#) [Compare](#)

Features

Wikis
Wikis host documentation for your repository.

https://github.com/TejalPathak23/Demo_About_Github/settings/pages

- Select the branch as main -> click Save

The screenshot shows the GitHub Pages settings page for a repository. The left sidebar has a 'Pages' tab selected. The main area is titled 'GitHub Pages' and contains the following sections:

- Build and deployment**:
 - Source**: A dropdown menu is open, showing 'Deploy from a branch'. Below it, a 'Branch' section indicates the site is built from the `main` branch. There are two buttons: a dropdown menu set to `main` and a `Save` button.
 - A note below says: 'Learn how to [add a Jekyll theme](#) to your site.'
- Custom domain**: A section about custom domains with a 'Save' and 'Remove' button.
- Integrations**: A section with a checked checkbox for 'Enforce HTTPS'.

Two buttons, 'main' and 'Save', are circled in red at the bottom center of the 'Source' section.

- Reload the page , you will get the URL where the site is been hosted.

The screenshot shows the GitHub Pages settings page for a repository named 'TejalPathak23 / portfolio'. The 'Settings' tab is selected. On the left, a sidebar lists various repository settings like General, Access, Collaborators, and Pages (which is currently selected). The main content area is titled 'GitHub Pages' and contains a message: 'GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.' Below this, a box highlights the live site URL: 'Your site is live at <https://tejalpathak23.github.io/portfolio/>' with a red oval around it. A timestamp indicates it was 'Last deployed by TejalPathak23 2 minutes ago'. There are 'Visit site' and '...' buttons. The right side of the page includes sections for 'Build and deployment' (with a dropdown for 'Deploy from a branch'), 'Branch' (set to 'main'), and 'Custom domain'.

TejalPathak23 / portfolio

Type to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security and analysis

Deploy keys

Secrets and variables

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is live at <https://tejalpathak23.github.io/portfolio/>

Last deployed by TejalPathak23 2 minutes ago

Visit site ...

Build and deployment

Source

Deploy from a branch

Branch

Your GitHub Pages site is currently being built from the `main` branch. [Learn more about configuring the publishing source for your site.](#)

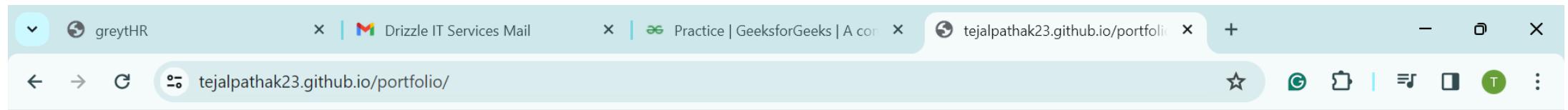
`main` / (root) Save

Learn how to [add a Jekyll theme](#) to your site.

Your site was last deployed to the `github-pages` environment by the `pages build and deployment` workflow. [Learn more about deploying to GitHub Pages using custom workflows](#)

Custom domain

- Output : Your hosted site.



Learning - How to deploy sites using github

Host - Tejal Pathak



Exploring network graphs on GitHub

- The network graph displays the branch history of the entire repository network, including fork branches. This graph is a timeline of the most recent commits, and shows up to 100 of the most recently pushed-to branches

- When you click any repository you will get “Insights” option. Which shows the count of pull request, etc , but to read this is not handy so we have a network option which shows graph.

The screenshot shows the GitHub Insights page for the repository `facebook/react`. The top navigation bar includes links for Code, Issues (1.2k), Pull requests (411), Actions, Projects, Wiki, Security, and Insights. A red oval highlights the **Insights** link. On the left sidebar, there are links for Pulse, Contributors, Community Standards, Commits, Code frequency, Dependency graph, Network (which is highlighted with a red oval), and Forks. The main content area displays a timeline from March 29, 2024 – April 5, 2024, with a "Period: 1 week" dropdown. A large red oval encloses the "Overview" section, which contains the following data:

Metric	Value
Active pull requests	72
Merged pull requests	48
Open pull requests	24
Closed issues	12
New issues	3
Active issues	15

Below the overview, a text block states: "Excluding merges, 15 authors have pushed 48 commits to main and 82 commits to all branches. On main, 191 files have changed and there have been 3,143 additions and 4,207 deletions." To the right is a bar chart showing commit counts for 13 different users, each accompanied by a small profile picture. A red oval highlights the first two bars of the chart. At the bottom, a specific pull request is highlighted: "#28739 Remove _owner field from JSX elements in prod if string refs are disabled" (merged 15 hours ago).

- It will show data into graphical format.

Screenshot of the GitHub Insights "Network graph" page for the repository `facebook/react`.

The page displays a timeline of commits and their relationships across different branches and forks. The main repository is owned by `facebook`, and the `main` branch is highlighted in blue. Two other branches are shown: `builds/www` and `dependabot/npm_and_yarn_fixtures_might_be_included-5.28.4`. A commit from the `builds/www` branch is connected to the `main` branch. The network graph also shows connections between the `facebook` repository and two forks: `UbuntuEvangelist` and `zys-contrib`.

Pulse

Contributors

Community Standards

Commits

Code frequency

Dependency graph

Network

Forks

Owners

facebook

UbuntuEvangelist

zys-contrib

Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.

The repository network shows the 100 most recently pushed forks.

```
graph LR; facebook[facebook] --> buildsw[builds/www]; facebook --> dependabot[dependabot/npm_and_yarn_fixtures_might_be_included-5.28.4]; buildsw --> main[main]; dependabot --> main; UbuntuEvangelist[UbuntuEvangelist] --- facebook; zyscontrib[zys-contrib] --- facebook;
```

Extra Tools: Cleaning the working repository

- We could forcefully delete files from the working directory if they are not yet into staging area.
- Command : `git clean -f -d`
- Here, f = force & d = any present directory

```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitA  
ndGitHub (master)  
$ touch t1.txt
```

```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitA  
ndGitHub (master)  
$ touch index.html
```

```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitA  
ndGitHub (master)  
$ touch about.html
```

```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitA  
ndGitHub (master)  
$ git clean -f -d  
Removing about.html  
Removing index.html  
Removing t1.txt
```

```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitA  
ndGitHub (master)  
$ |
```

Changing commit messages and changes

- Suppose in case you commit a file previously with a wrong message, now you want to update that message in this case the command is helpful.
- *Note: This can be done only for previous commits*
- Command : **git commit – amend**
- When you write this command then a “vim editor” will open on your screen, where
 - I = Insert
 - To exit = esc -> :wq

```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitAndGitHub (master)
$ touch index.html
```

```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitAndGitHub (master)
$ git add .
```

```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitAndGitHub (master)
$ git commit -m "index commit"
[master 238a812] index commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 index.html
```

```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitAndGitHub (master)
$ git commit --amend
```

The file was inserted

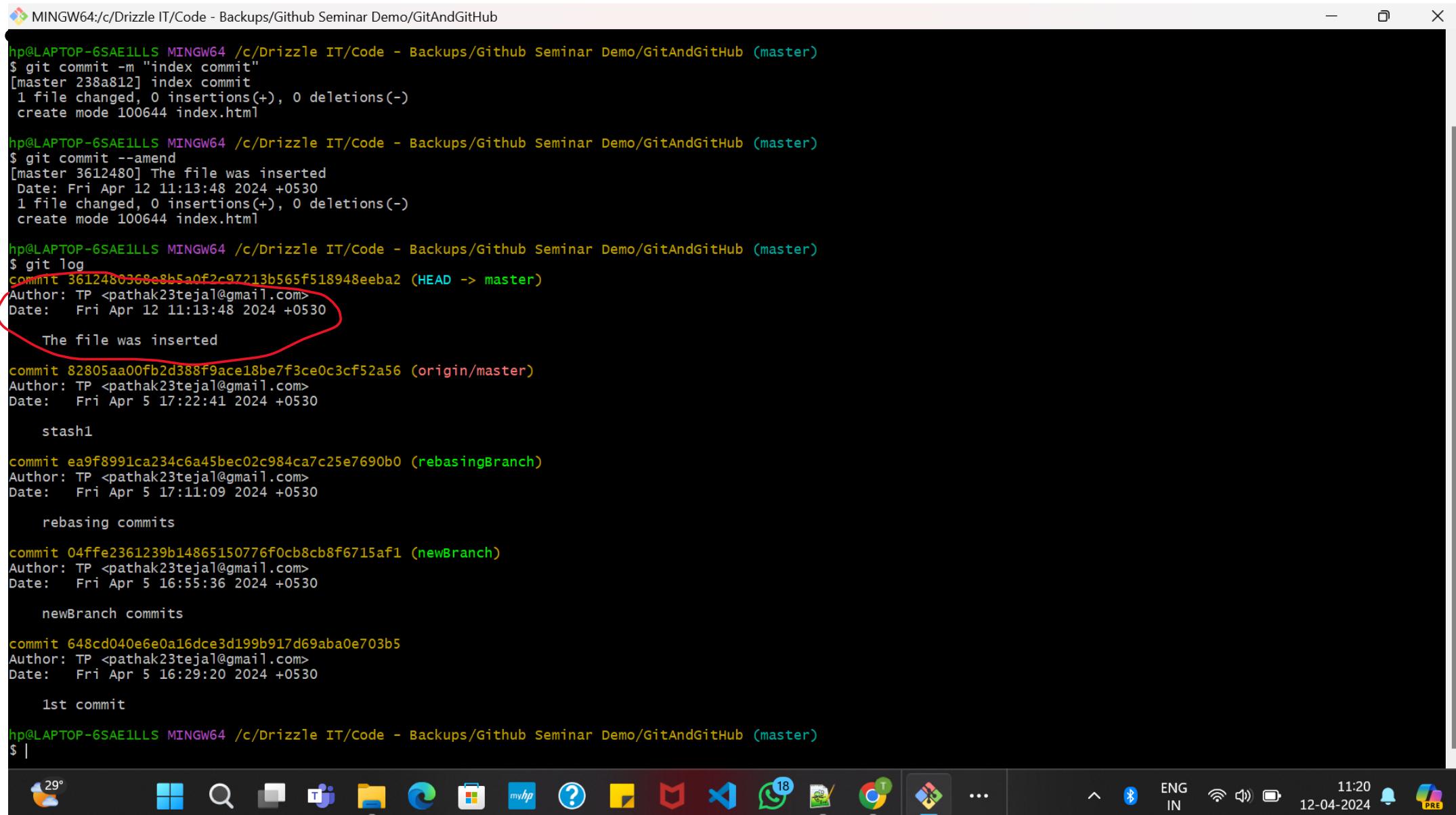
```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Fri Apr 12 11:13:48 2024 +0530
#
# On branch master
# Changes to be committed:
#   new file:   index.html
#
```

.git/COMMIT_EDITMSG[+][unix] (11:16 12/04/2024)
:wq

1,1 All



- Now by using git log, you can see the commit message is been changed.



```
MINGW64:/c/Drizzle IT/Code - Backups/Github Seminar Demo/GitAndGitHub
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitAndGitHub (master)
$ git commit -m "index commit"
[master 238a812] index commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 index.html

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitAndGitHub (master)
$ git commit --amend
[master 3612480] The file was inserted
Date: Fri Apr 12 11:13:48 2024 +0530
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 index.html

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitAndGitHub (master)
$ git log
commit 3612480368e8b5a0f2c97213b565f518948eeba2 (HEAD -> master)
Author: TP <pathak23tejal@gmail.com>
Date: Fri Apr 12 11:13:48 2024 +0530

  The file was inserted

commit 82805aa00fb2d388f9ace18be7f3ce0c3cf52a56 (origin/master)
Author: TP <pathak23tejal@gmail.com>
Date: Fri Apr 5 17:22:41 2024 +0530

  stash1

commit ea9f8991ca234c6a45bec02c984ca7c25e7690b0 (rebasingBranch)
Author: TP <pathak23tejal@gmail.com>
Date: Fri Apr 5 17:11:09 2024 +0530

  rebasing commits

commit 04ffe2361239b14865150776f0cb8cb8f6715af1 (newBranch)
Author: TP <pathak23tejal@gmail.com>
Date: Fri Apr 5 16:55:36 2024 +0530

  newBranch commits

commit 648cd040e6e0a16dce3d199b917d69aba0e703b5
Author: TP <pathak23tejal@gmail.com>
Date: Fri Apr 5 16:29:20 2024 +0530

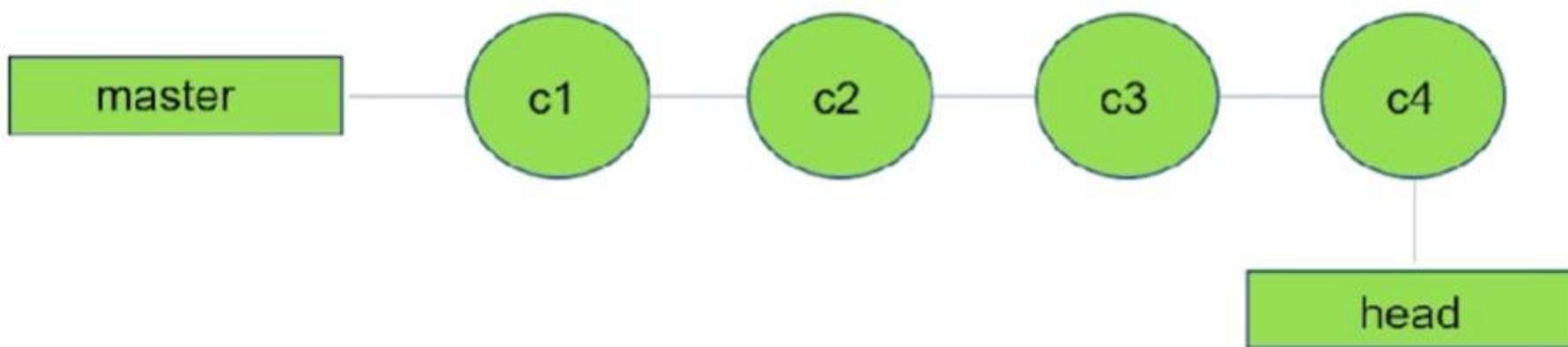
  1st commit

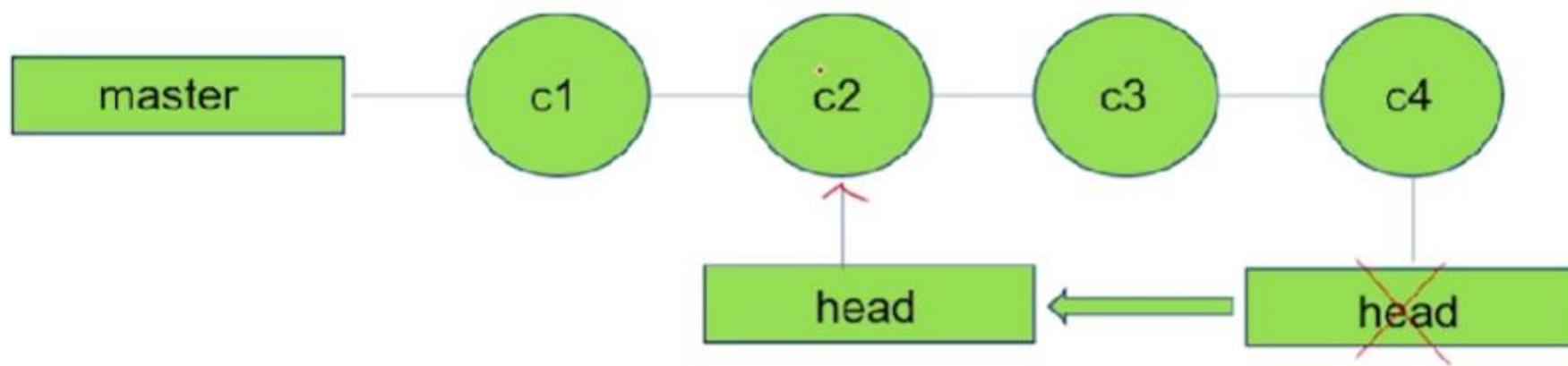
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitAndGitHub (master)
$ |
```

The screenshot shows a terminal window on a Windows system (MINGW64) displaying a git log output. The terminal title is 'MINGW64:/c/Drizzle IT/Code - Backups/Github Seminar Demo/GitAndGitHub'. The log output shows several commits. A specific commit is highlighted with a red circle around its message: 'The file was inserted'. This commit is the result of an 'git commit --amend' operation, as indicated by the date and author information matching the previous commit. The terminal also shows other commits, including ones for a 'rebase' branch and a 'newBranch'.

Checking past commits

Checking Past Commits



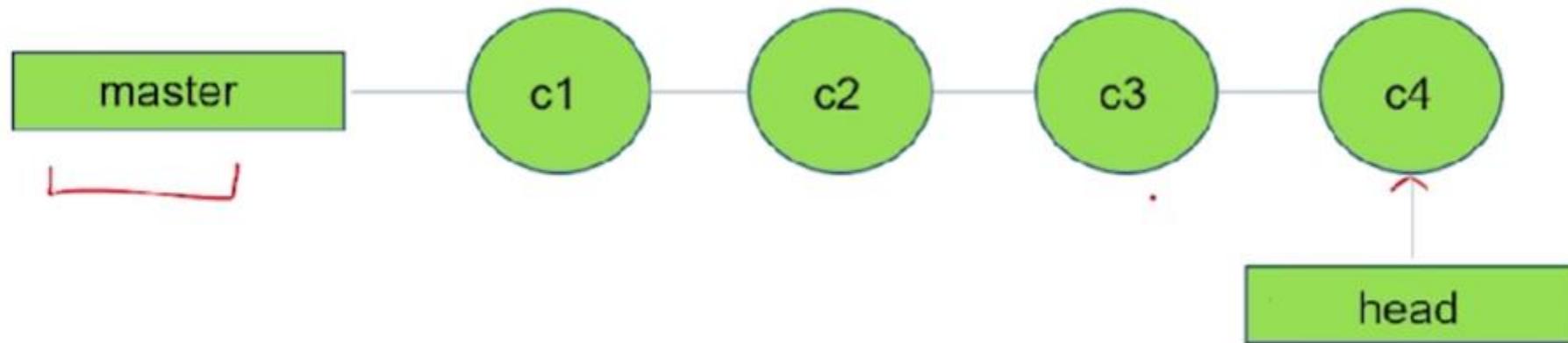


- Checking Past commits:
 - `git checkout <commit_id>`

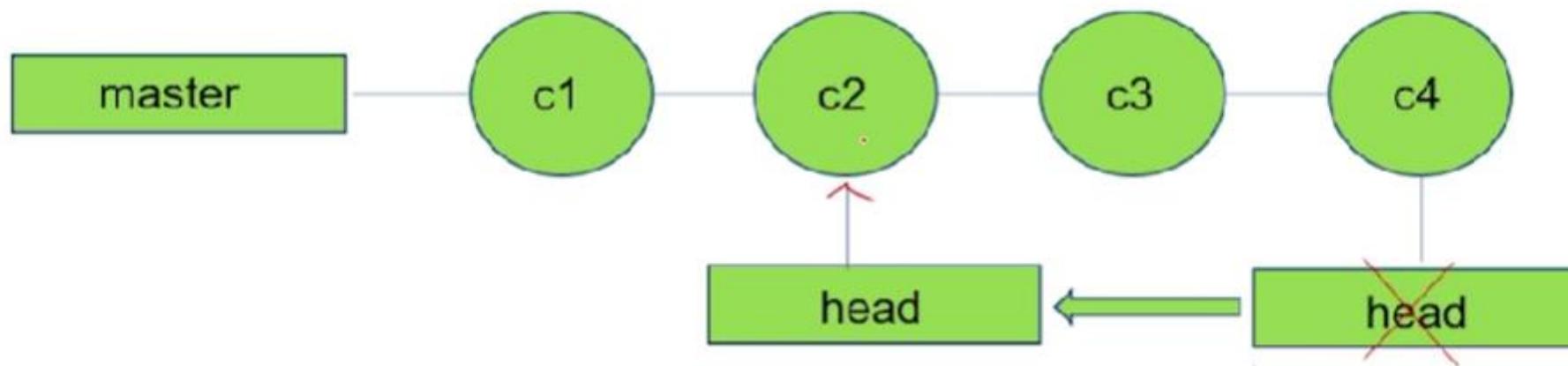
We could get this commit Id by using this command: `git log --oneline`

- Head will be pointing to the most recent commit.

Checking Past Commits



- To change the previous commit the head must be pointed to that specific commit.



```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitAndGitHub ((3dbdc0...))
$ touch past1.txt

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitAndGitHub ((3dbdc0...))
$ git add .

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitAndGitHub ((3dbdc0...))
$ git commit -m "first"
[detached HEAD 9040a60] first
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 past1.txt

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitAndGitHub ((9040a60...))
$ touch past2.txt

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitAndGitHub ((9040a60...))
$ git add .

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitAndGitHub ((9040a60...))
$ git commit -m "past2 in past1"
[detached HEAD ef40cb2] past2 in past1
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 past2.txt

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitAndGitHub ((ef40cb2...))
$ rm past2.txt

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitAndGitHub ((ef40cb2...))
$ git status
HEAD detached from 3dbdc0
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:   past2.txt

no changes added to commit (use "git add" and/or "git commit -a")

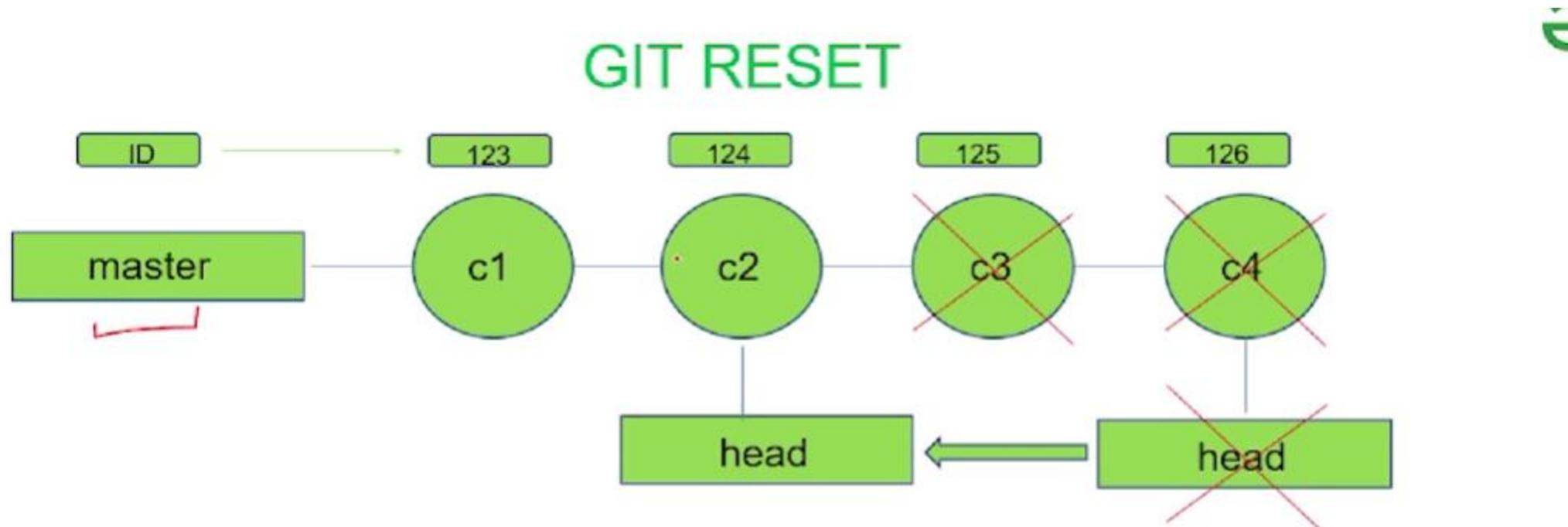
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitAndGitHub ((ef40cb2...))
$ git add .
```

ENG
IN12:15
12-04-2024

```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitA  
ndGitHub ((ef40cb2...))  
$ git add .  
  
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitA  
ndGitHub ((ef40cb2...))  
$ git commit -m "Deleted past2"  
[detached HEAD c82f909] Deleted past2  
 1 file changed, 0 insertions(+), 0 deletions(-)  
 delete mode 100644 past2.txt  
  
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitA  
ndGitHub ((c82f909...))  
$ git log --oneline  
c82f909 (HEAD) Deleted past2  
ef40cb2 past2 in past1  
9040a60 first  
3dbcd0 second commit  
33c8086 first commit  
3612480 The file was inserted  
82805aa (origin/master) stash1  
ea9f899 (rebasingBranch) rebasing commits  
04ffe23 (newBranch) newBranch commits  
648cd04 1st commit  
  
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitA  
ndGitHub ((c82f909...))  
$ git checkout ef40cb2  
Warning: you are leaving 1 commit behind, not connected to  
any of your branches:  
  
      c82f909 Deleted past2  
  
If you want to keep it by creating a new branch, this may be a good time  
to do so with:  
  
  git branch <new-branch-name> c82f909  
  
HEAD is now at ef40cb2 past2 in past1
```

Git Reset

- Deletes the commit from the top and resets head to the given id.



- **git reset <commit_id>** : By default it is soft, Deletes the commit but you can still get back the changes introduced in those deleted commits.
- **git reset <commit_id> -- hard** : Deletes the commit and even the changes introduced in that commit.

```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitAndGitHub (master)
$ git log --oneline
854745c (HEAD -> master) deleted p2
3dbcd0 second commit
33c8086 first commit
3612480 The file was inserted
82805aa (origin/master) stash1
ea9f899 (rebasingBranch) rebasing commits
04ffe23 (newBranch) newBranch commits
648cd04 1st commit
```

```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitAndGitHub (master)
$ git reset 33c8086
```

```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitAndGitHub (master)
$ git commit -am "first commit"
On branch master
nothing to commit, working tree clean
```

```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitAndGitHub (master)
$ git reset 3dbcd0 --hard
HEAD is now at 3dbcd0 second commit
```

Reverting Commit

- The changes introduced by the commit can be reversed using revert command.
- **git revert <commit_id>** : reverts the commit and again commits the result.
- **git revert -n <commit_id>** : reverts the commit but do not again commit the result. You have to explicitly commit the files.
- **git revert –abort** : Aborts the current revert operation

Deleting commits from github

- Command: `git reset <commit_id>`

```
hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitAndGitHub (main)
$ git log --oneline
27ac0f6 (HEAD -> main, master) Third
d090c2a second
8ffa84c First
3dbcd0 second commit
33c8086 first commit
3612480 The file was inserted
82805aa (origin/master) stash1
ea9f899 (rebasingBranch) rebasing commits
04ffe23 (newBranch) newBranch commits
648cd04 1st commit

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitAndGitHub (main)
$ git reset 27ac0f6
Unstaged changes after reset:
M      index.txt

hp@LAPTOP-6SAE1LLS MINGW64 /c/Drizzle IT/Code - Backups/Github Seminar Demo/GitAndGitHub (main)
$ git status
On branch main
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:   index.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

