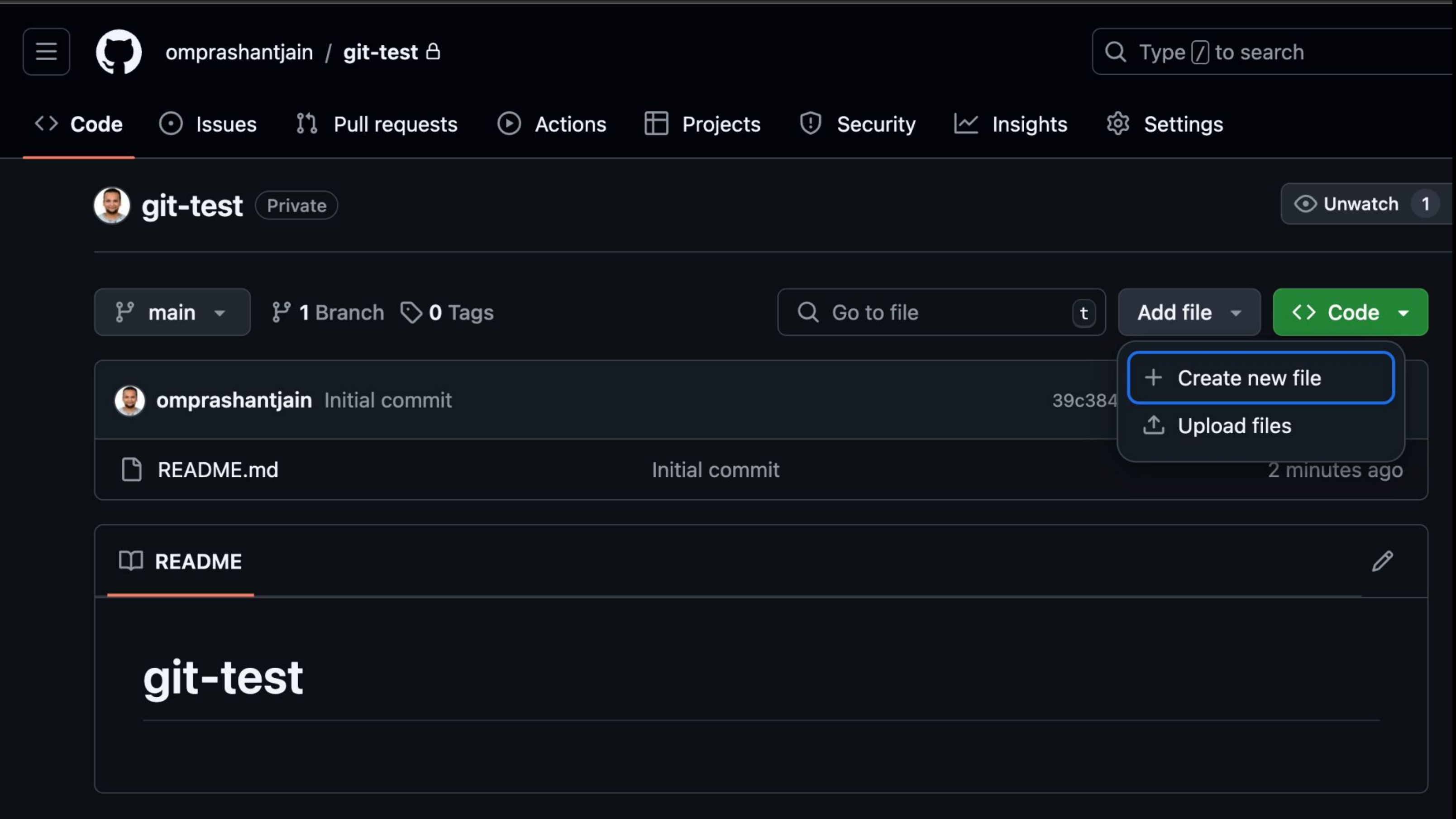
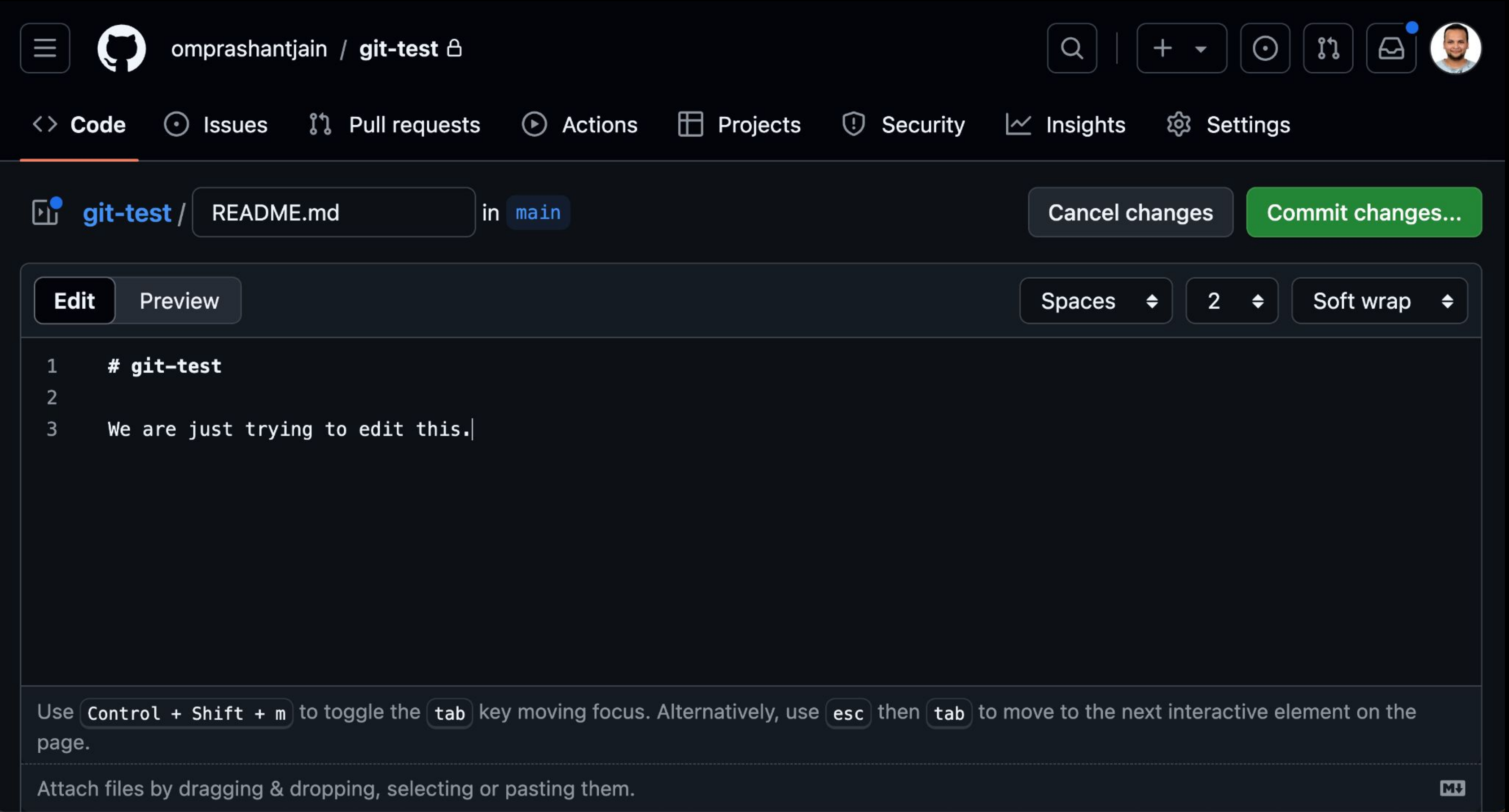
4. Working with GitHub UI

1. Creating or Uploading files on GitHub
2. Editing Files
3. Committing changes
4. Viewing commit history
5. Forking of Other Repos
6. Working with Github Desktop
7. Git inside VSCode

4.1 Creating or Uploading files on GitHub



4.2 Editing Files



git-test / README.md in main

Spaces 2 Soft wrap

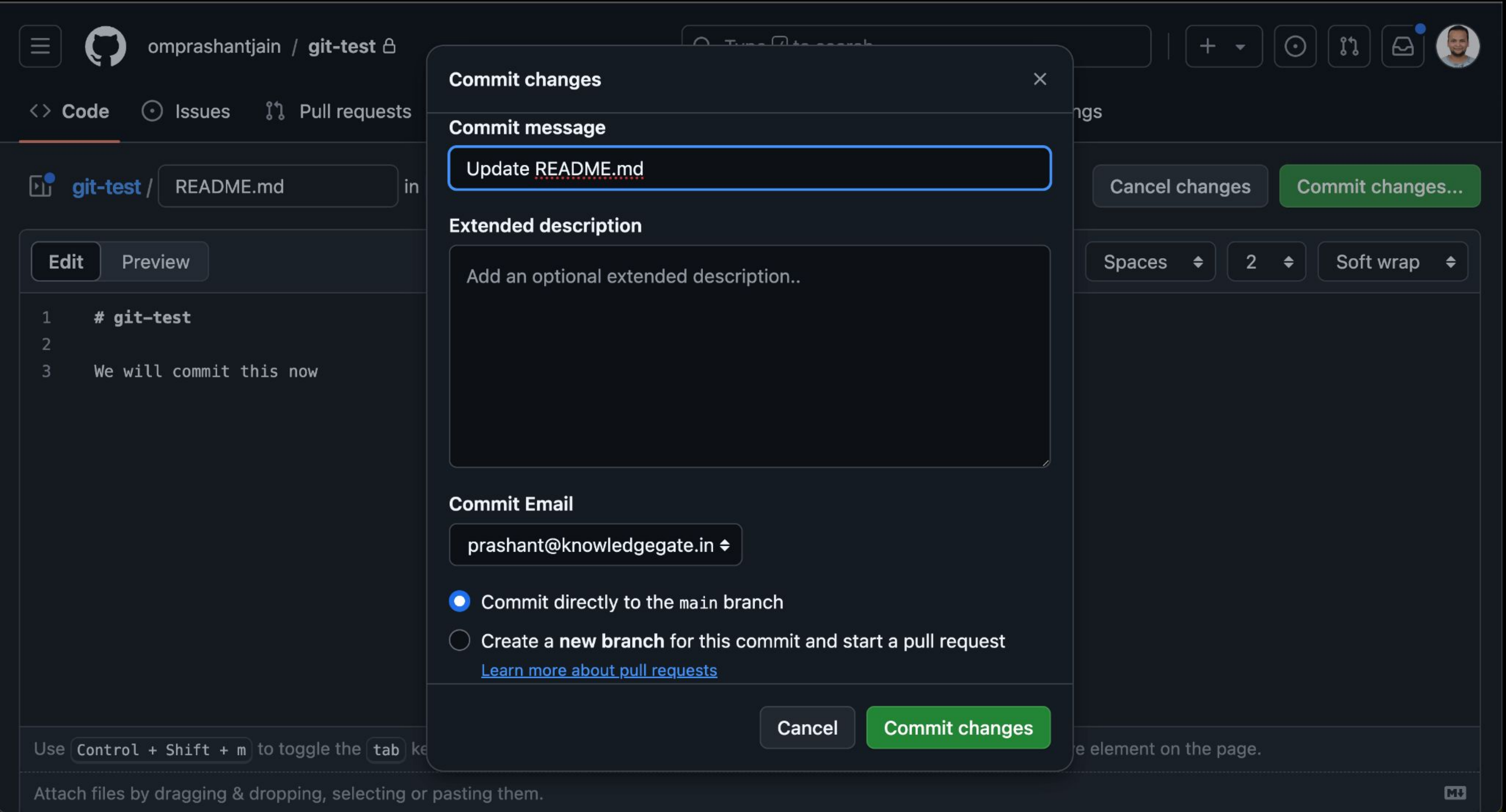
```
# git-test

We are just trying to edit this.
```

Use `Control + Shift + m` to toggle the `tab` key moving focus. Alternatively, use `esc` then `tab` to move to the next interactive element on the page.

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

4.3 Committing changes



```
# git-test

We will commit this now
```

Commit changes

Commit message: Update README.md

Extended description: Add an optional extended description..

Commit Email: prashant@knowledgegate.in

● Commit directly to the `main` branch

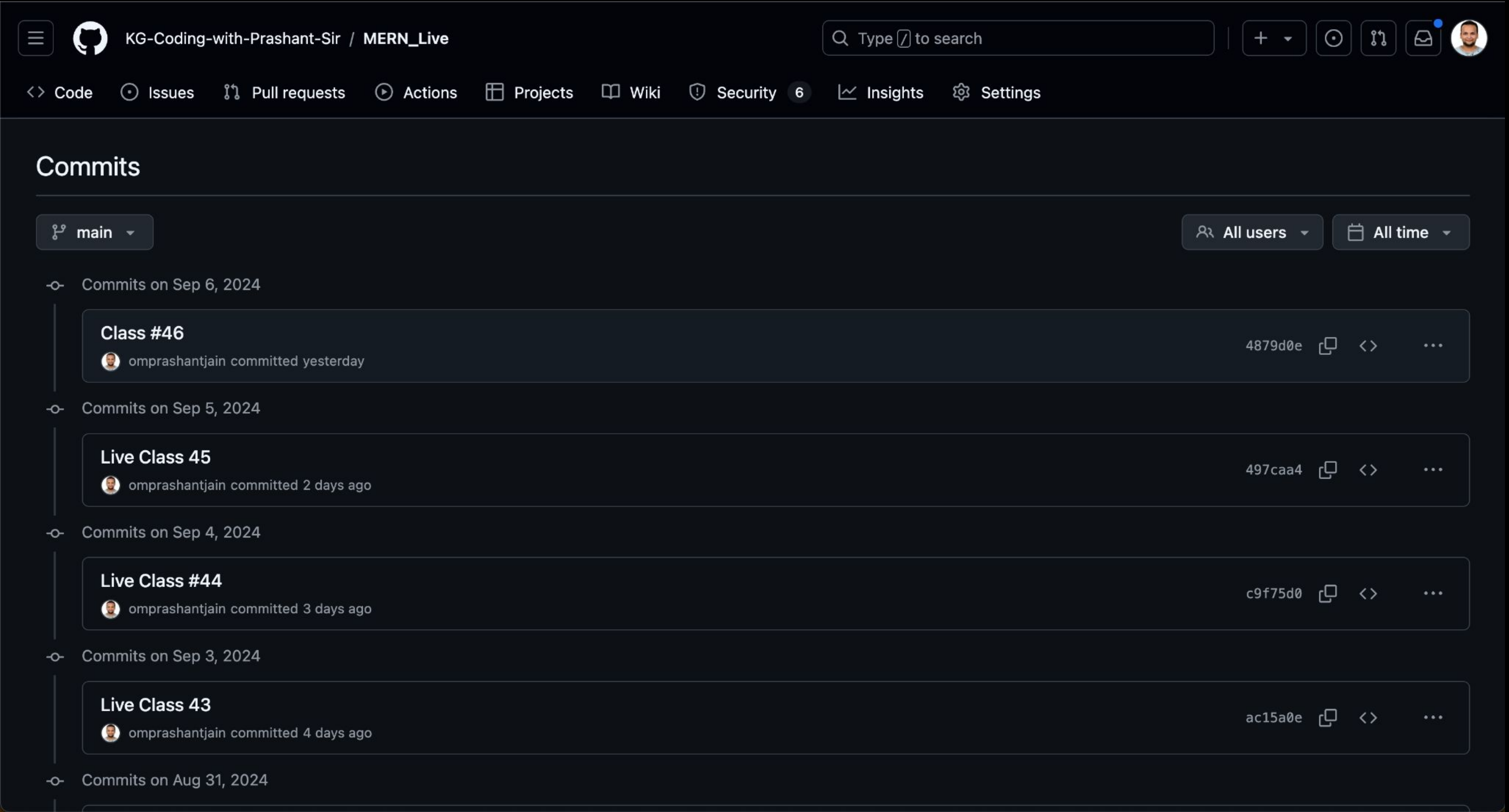
○ Create a **new branch** for this commit and start a pull request

Learn more about pull requests

Use `Control + Shift + m` to toggle the `tab` ke... e element on the page.

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

4.4 Viewing commit history



Commits

main · All users · All time

Commits on Sep 6, 2024

Class #46
omprashantjain committed yesterday · 4879d0e

Commits on Sep 5, 2024

Live Class 45
omprashantjain committed 2 days ago · 497caa4

Commits on Sep 4, 2024

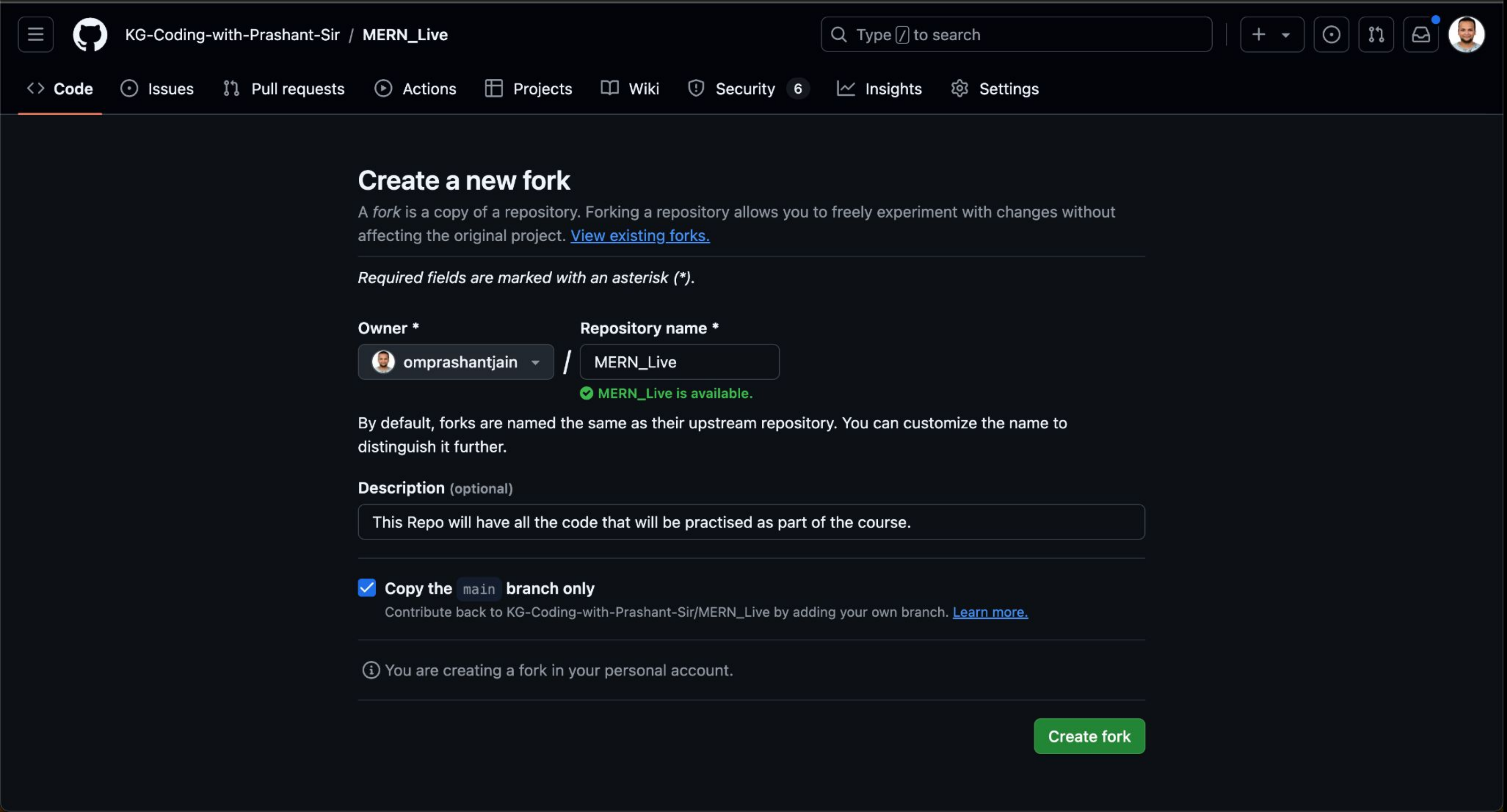
Live Class #44
omprashantjain committed 3 days ago · c9f75d0

Commits on Sep 3, 2024

Live Class 43
omprashantjain committed 4 days ago · ac15a0e

Commits on Aug 31, 2024

4.5 Forking of Other Repos



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 * : omprashantjain / Repository name * : MERN_Live

✓ MERN_Live is available.

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

Description (optional)

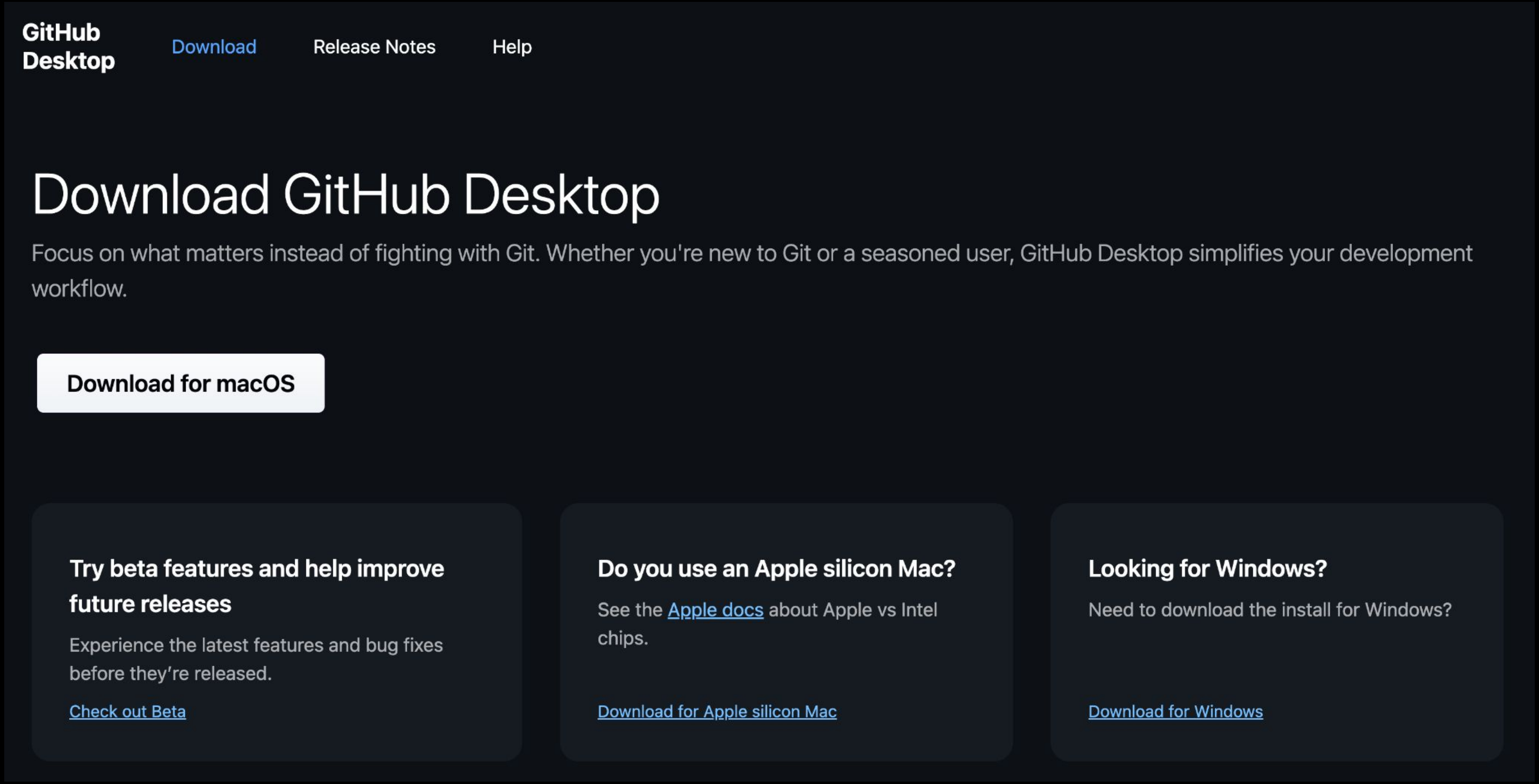
This Repo will have all the code that will be practised as part of the course.

☑ **Copy the `main` branch only**

Contribute back to KG-Coding-with-Prashant-Sir/MERN_Live by adding your own branch. Learn more.

ⓘ You are creating a fork in your personal account.

4.6 Working with Github Desktop



GitHub Desktop

Download GitHub Desktop

Focus on what matters instead of fighting with Git. Whether you're new to Git or a seasoned user, GitHub Desktop simplifies your development workflow.

Try beta features and help improve future releases

Experience the latest features and bug fixes before they're released.

Check out Beta

Do you use an Apple silicon Mac?

See the Apple docs about Apple vs Intel chips.

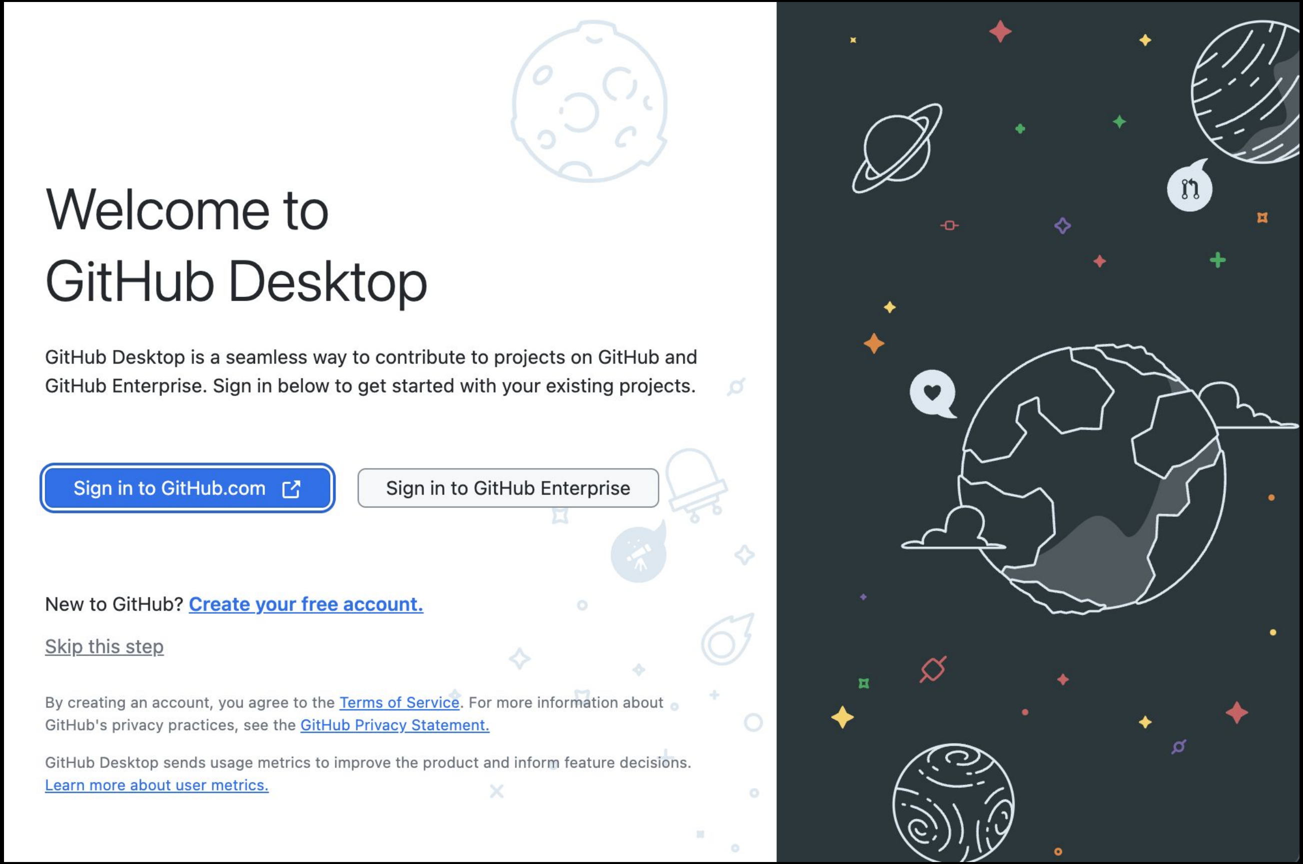
Download for Apple silicon Mac

Looking for Windows?

Need to download the install for Windows?

Download for Windows

4.6 Working with Github Desktop



Welcome to GitHub Desktop

GitHub Desktop is a seamless way to contribute to projects on GitHub and GitHub Enterprise. Sign in below to get started with your existing projects.

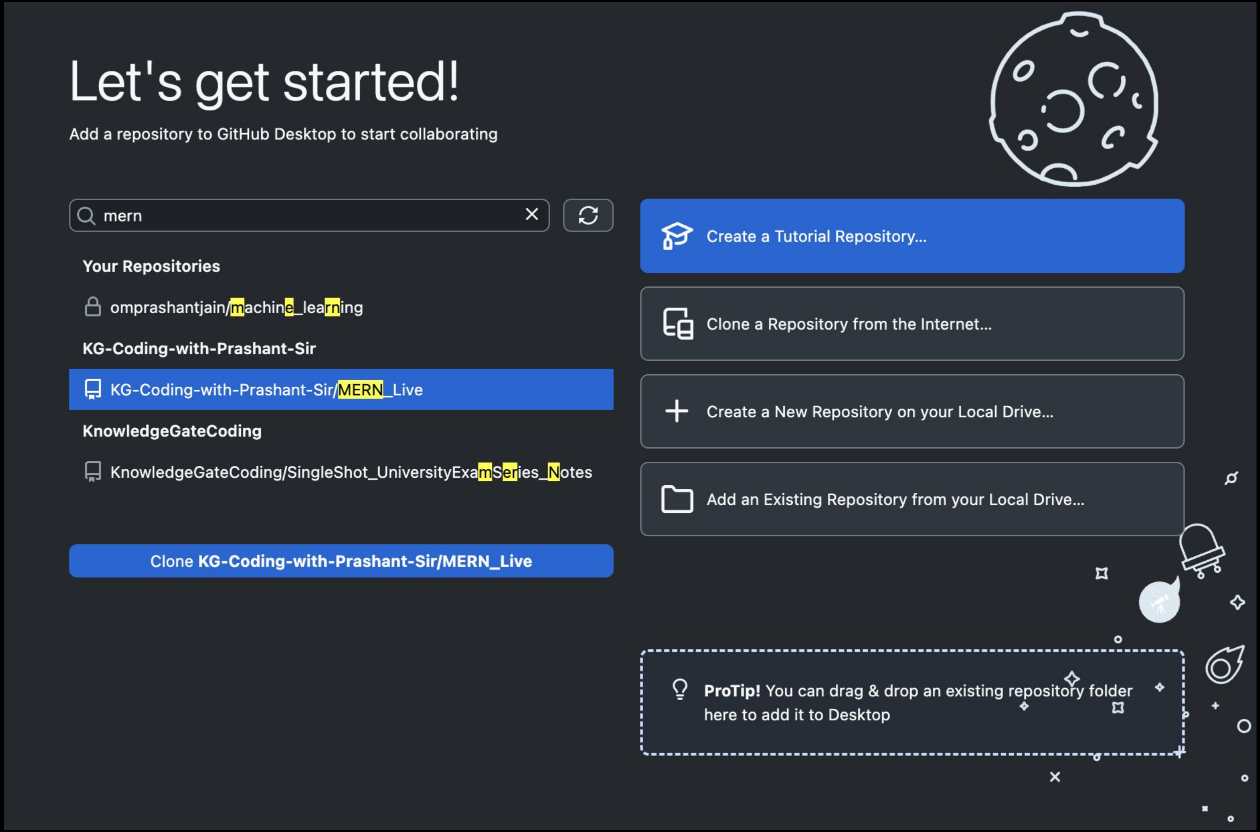
New to GitHub? Create your free account.

Skip this step

By creating an account, you agree to the Terms of Service. For more information about GitHub's privacy practices, see the GitHub Privacy Statement.

GitHub Desktop sends usage metrics to improve the product and inform feature decisions. Learn more about user metrics.

4.6 Working with Github Desktop



Let's get started!

Add a repository to GitHub Desktop to start collaborating

mern

Your Repositories

omprashantjain/machine_learning

KG-Coding-with-Prashant-Sir

KG-Coding-with-Prashant-Sir/MERN_Live

KnowledgeGateCoding

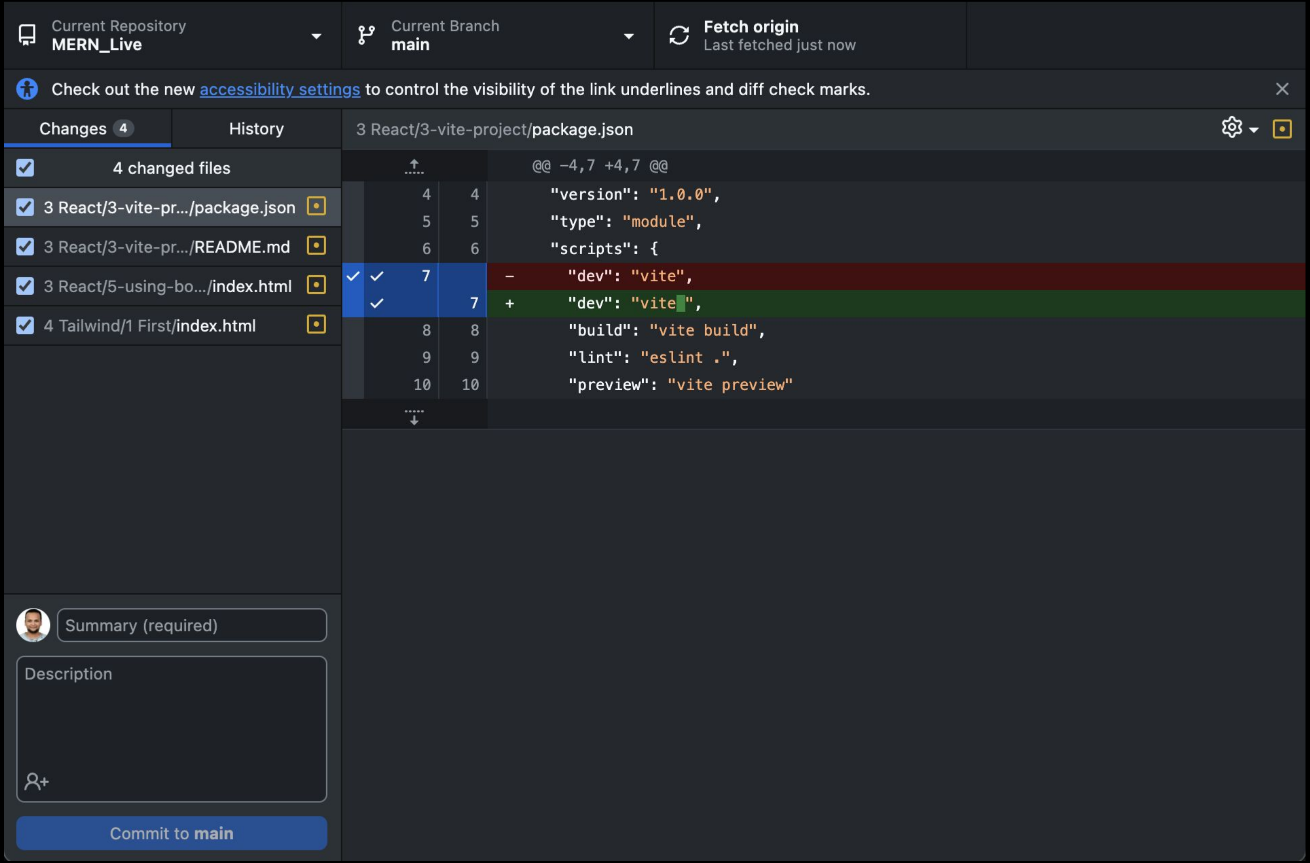
KnowledgeGateCoding/SingleShot_UniversityExamSeries_Notes

Clone **KG-Coding-with-Prashant-Sir/MERN_Live**

Create a Tutorial Repository...
Clone a Repository from the Internet...
Create a New Repository on your Local Drive...
Add an Existing Repository from your Local Drive...

ProTip! You can drag & drop an existing repository folder here to add it to Desktop

4.6 Working with Github Desktop



Check out the new accessibility settings to control the visibility of the link underlines and diff check marks.

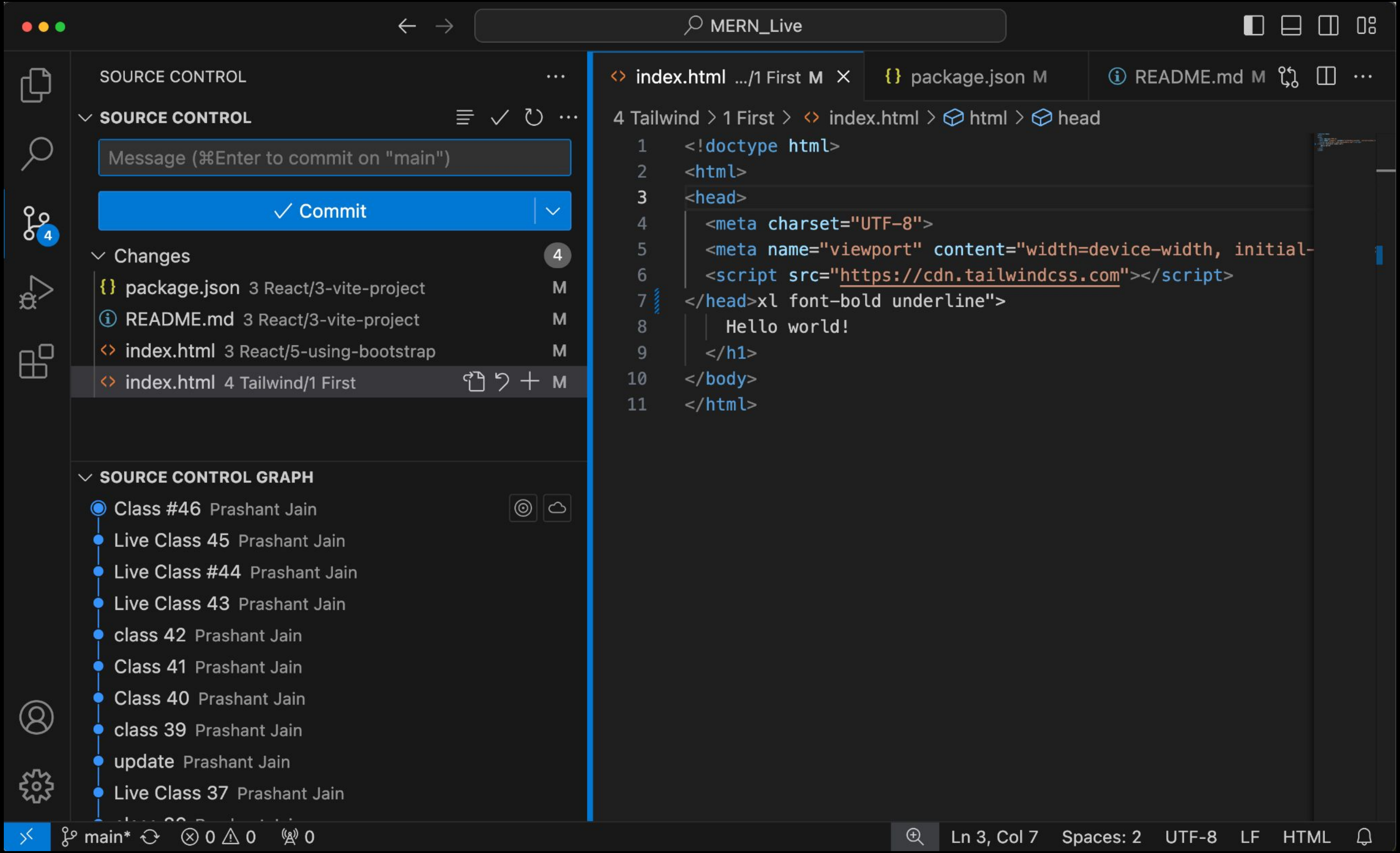
☑ 4 changed files

- ☑ 3 React/3-vite-pr.../package.json
- ☑ 3 React/3-vite-pr.../README.md
- ☑ 3 React/5-using-bo.../index.html
- ☑ 4 Tailwind/1 First/index.html

3 React/3-vite-project/package.json

```
@@ -4,7 +4,7 @@
    "version": "1.0.0",
    "type": "module",
    "scripts": {
-     "dev": "vite",
+     "dev": "vite ",
      "build": "vite build",
      "lint": "eslint .",
      "preview": "vite preview"
```

4.7 Git inside VSCode



Changes 4

package.json	3 React/3-vite-project	M
README.md	3 React/3-vite-project	M
index.html	3 React/5-using-bootstrap	M
index.html	4 Tailwind/1 First	M

SOURCE CONTROL GRAPH

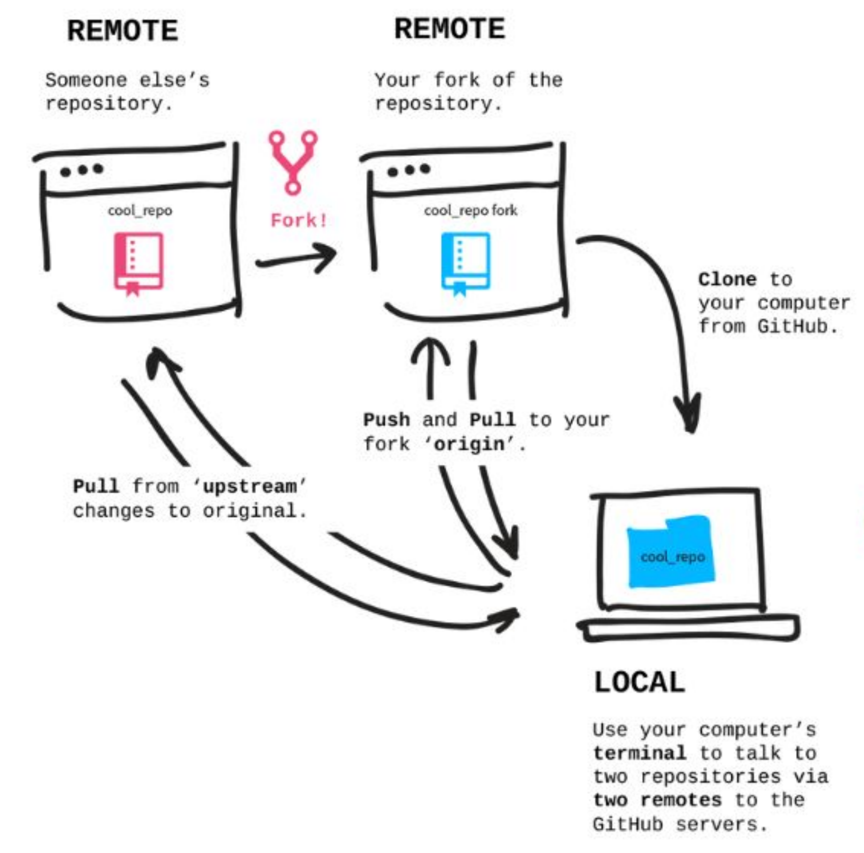
- Class #46 Prashant Jain
- Live Class 45 Prashant Jain
- Live Class #44 Prashant Jain
- Live Class 43 Prashant Jain
- class 42 Prashant Jain
- Class 41 Prashant Jain
- Class 40 Prashant Jain
- class 39 Prashant Jain
- update Prashant Jain
- Live Class 37 Prashant Jain

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
  <script src="https://cdn.tailwindcss.com"></script>
</head>xl font-bold underline">
    Hello world!
  </h1>
</body>
</html>
```

5. Essential Git Commands

1. git clone
2. What is Working Directory
3. git status
4. What is Staging Area
5. git diff
6. git add & git checkout
7. git commit & git reset
8. git log
9. git push & git pull

5.1 git clone



REMOTE

Someone else's repository.

REMOTE

Your fork of the repository.

Clone to your computer from GitHub.

Push and **Pull** to your fork '**origin**'.

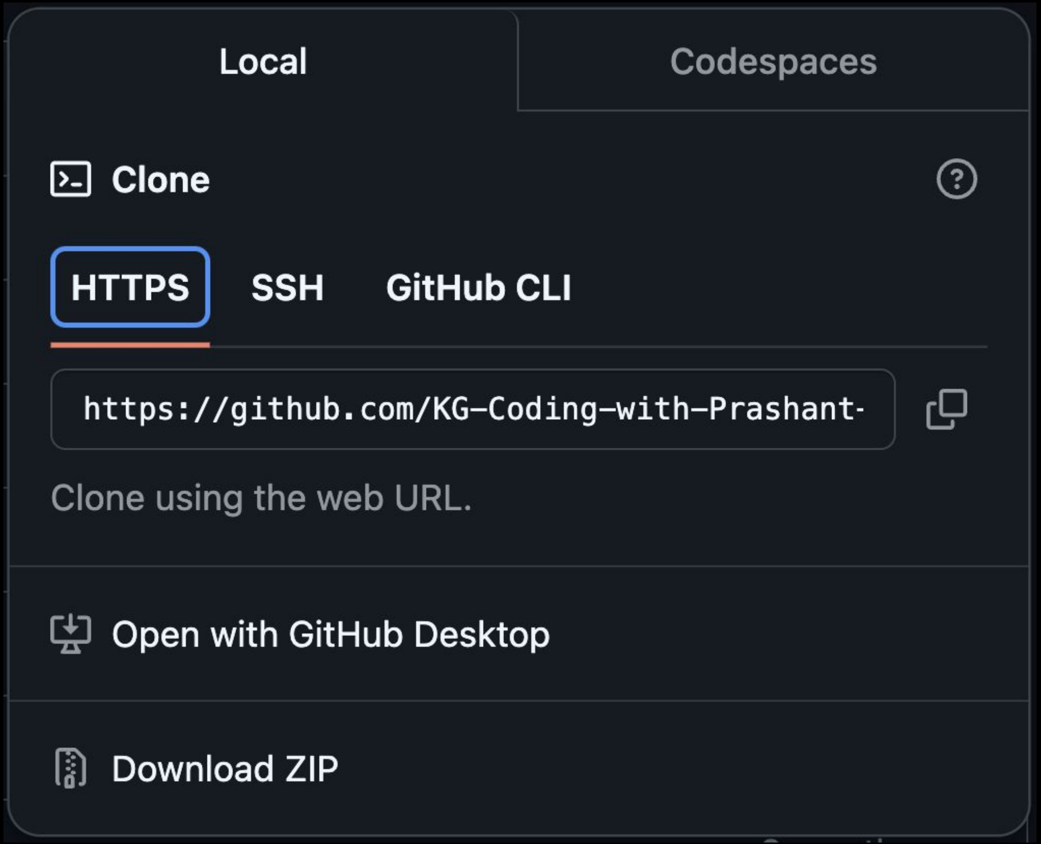
Pull from '**upstream**' changes to original.

LOCAL

Use your computer's **terminal** to talk to two repositories via **two remotes** to the GitHub servers.

- Creates a local copy of a remote repository.
- Downloads all files, branches, and commit history.

5.1 git clone



```
https://github.com/KG-Coding-with-Prashant-
```

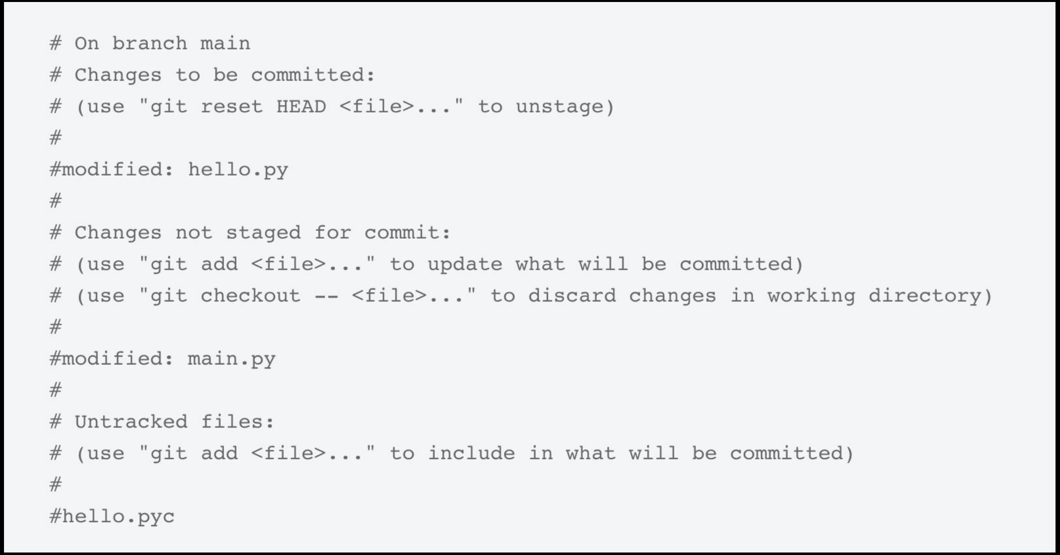
Clone using the web URL.

5.2 What is Working Directory

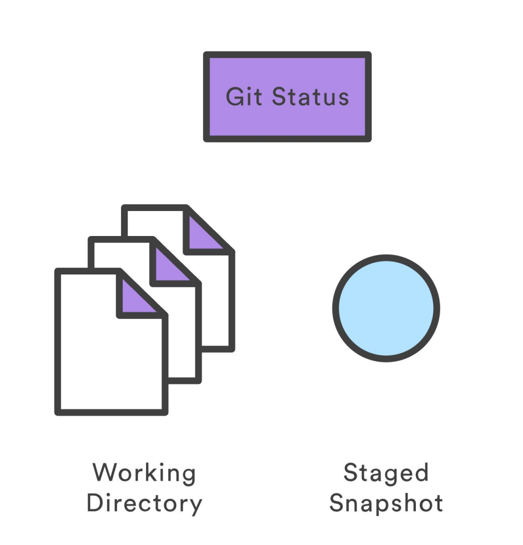
The **desk** where you're actively working on a draft of a book or project. The drafts and notes on your desk are not yet published or saved in the library.

- The current state of your project files on your local machine.
- Shows the files you're actively working on.
- Changes here are not tracked until added to the staging area.
- It reflects a specific snapshot or branch of the repository.

5.3 git status



```
# On branch main
# Changes to be committed:
# (use "git reset HEAD <file>..." to unstage)
#
#modified: hello.py
#
# Changes not staged for commit:
# (use "git add <file>..." to update what will be committed)
# (use "git checkout -- <file>..." to discard changes in working directory)
#
#modified: main.py
#
# Untracked files:
# (use "git add <file>..." to include in what will be committed)
#
#hello.pyc
```

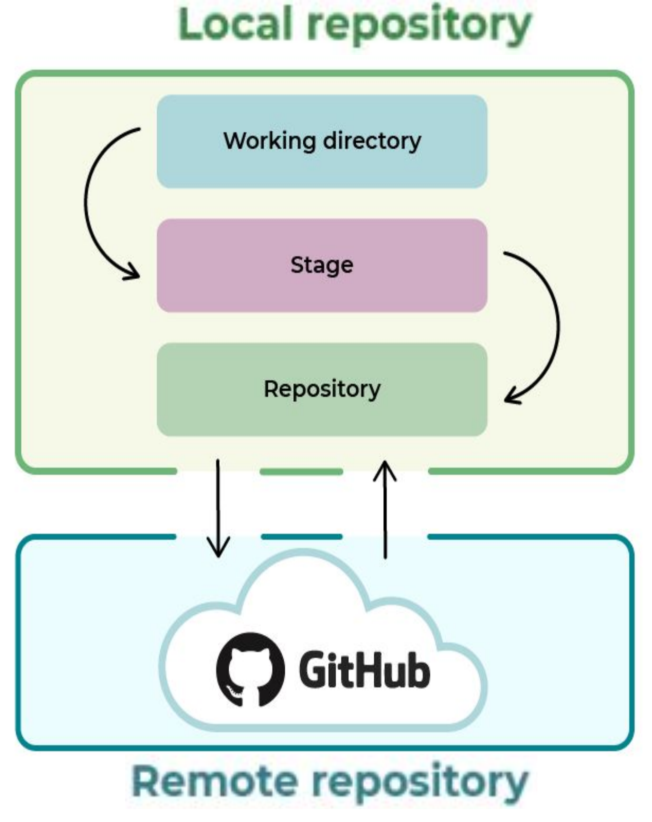


- Displays the current state of the working directory.
- Shows changes that have been staged, unstaged, or not tracked.

5.4 What is Staging Area

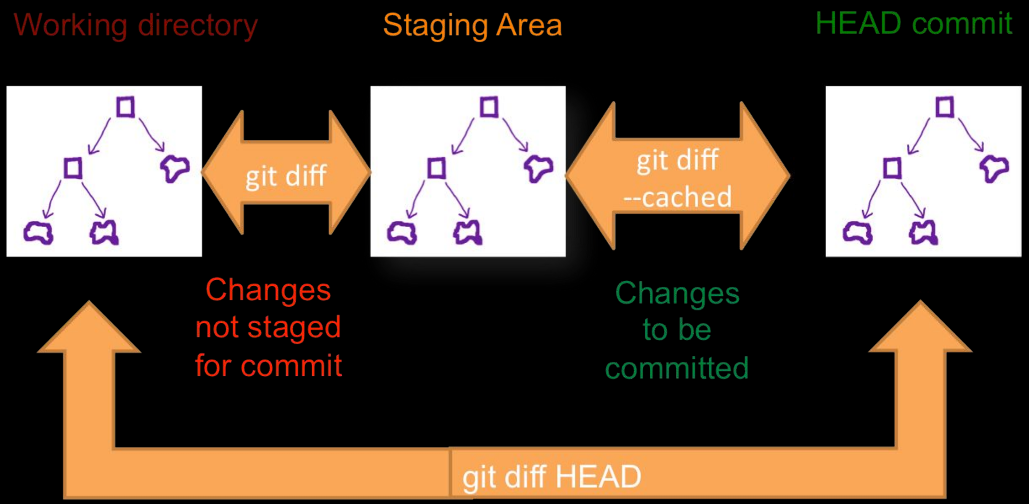
A **clipboard** where you place specific pages you want to review before finalizing a document. You might choose only a few changes to add for the next version of the book.

5.4 What is Staging Area



- Also known as the "index" or "cache"
- A intermediate area between the working directory and repository
- Holds changes that are ready to be committed
- Allows you to selectively choose which changes to include in the next commit
- Acts as a preview of your next commit

5.5 git diff



Working directory

Staging Area

HEAD commit

git diff

git diff --cached

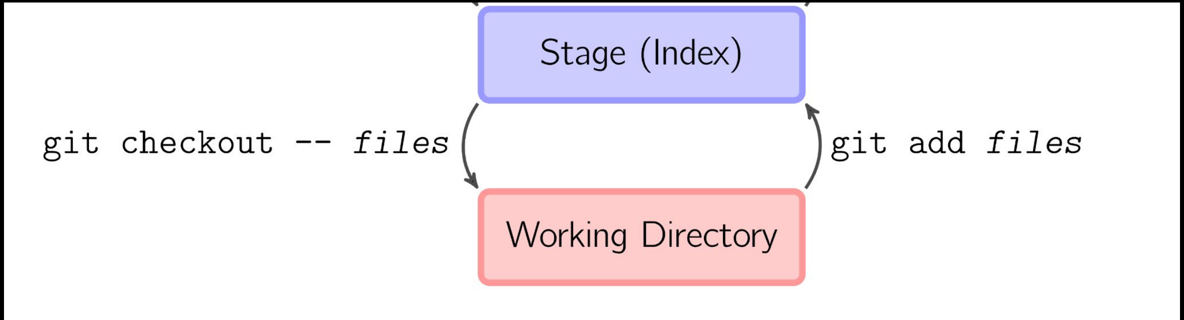
Changes not staged for commit

Changes to be committed

git diff HEAD

- Shows the differences between two states of a repository.
- Can be used to view changes between:
  - Working directory and staging area (uncommitted changes).
  - Staging area and last commit (staged changes).
  - Two different commits or branches.
- Useful for reviewing changes before committing or pushing them.

5.6 git add & git checkout



Stage (Index)

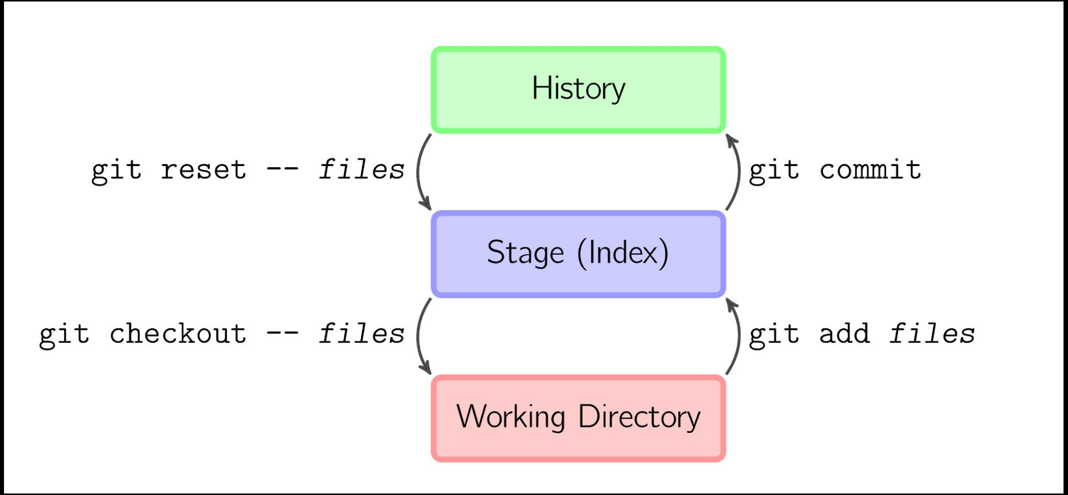
git checkout -- *files*

git add *files*

Working Directory

- git add
  - Stages changes to be committed.
  - Can add individual files or all changes with `git add`
- `git checkout`
  - Replaces the contents of the working directory with the version from the target branch or commit.

5.7 git commit & git reset



History

git reset -- *files*

git commit

Stage (Index)

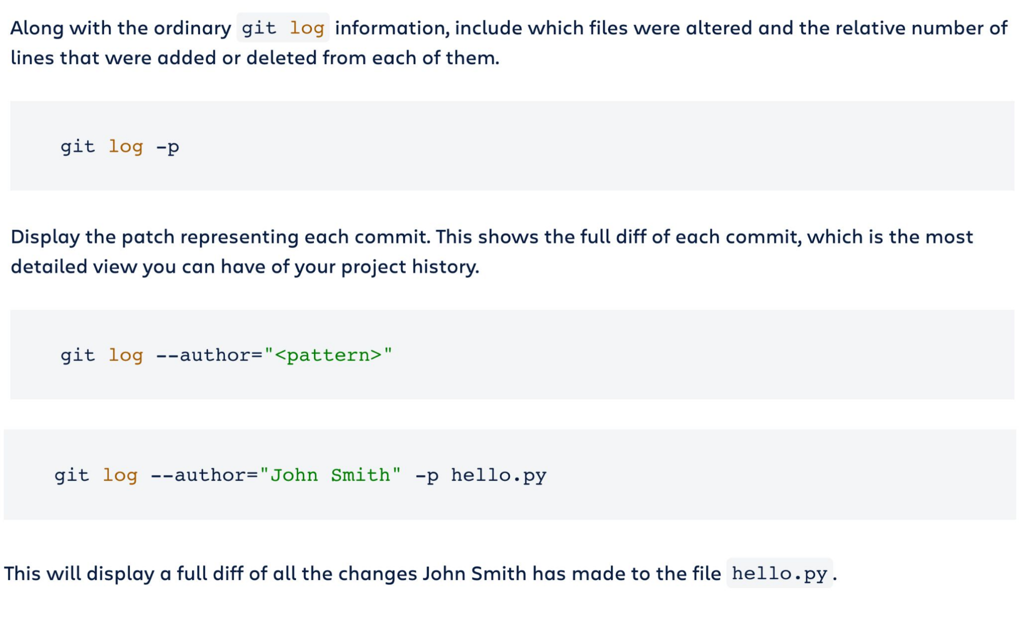
git checkout -- *files*

git add *files*

Working Directory

- git commit
  - Records the staged changes in the repository's history.
  - Requires a commit message describing the changes.
- git reset
  - Undoes changes by moving the current branch pointer to a previous commit. Modes (--soft, –mixed, –hard)

5.8 git log



Along with the ordinary `git log` information, include which files were altered and the relative number of lines that were added or deleted from each of them.

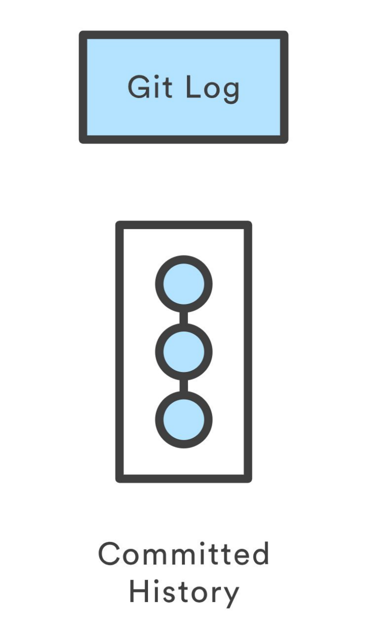
```
git log -p
```

Display the patch representing each commit. This shows the full diff of each commit, which is the most detailed view you can have of your project history.

```
git log --author="<pattern>"
```

```
git log --author="John Smith" -p hello.py
```

This will display a full diff of all the changes John Smith has made to the file `hello.py`.



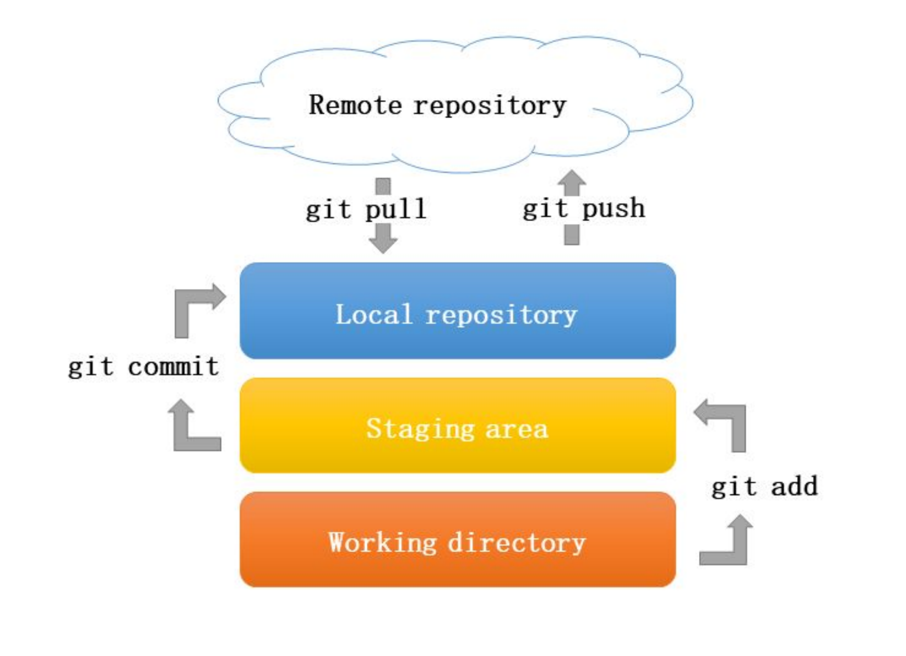
Git Log

Committed History

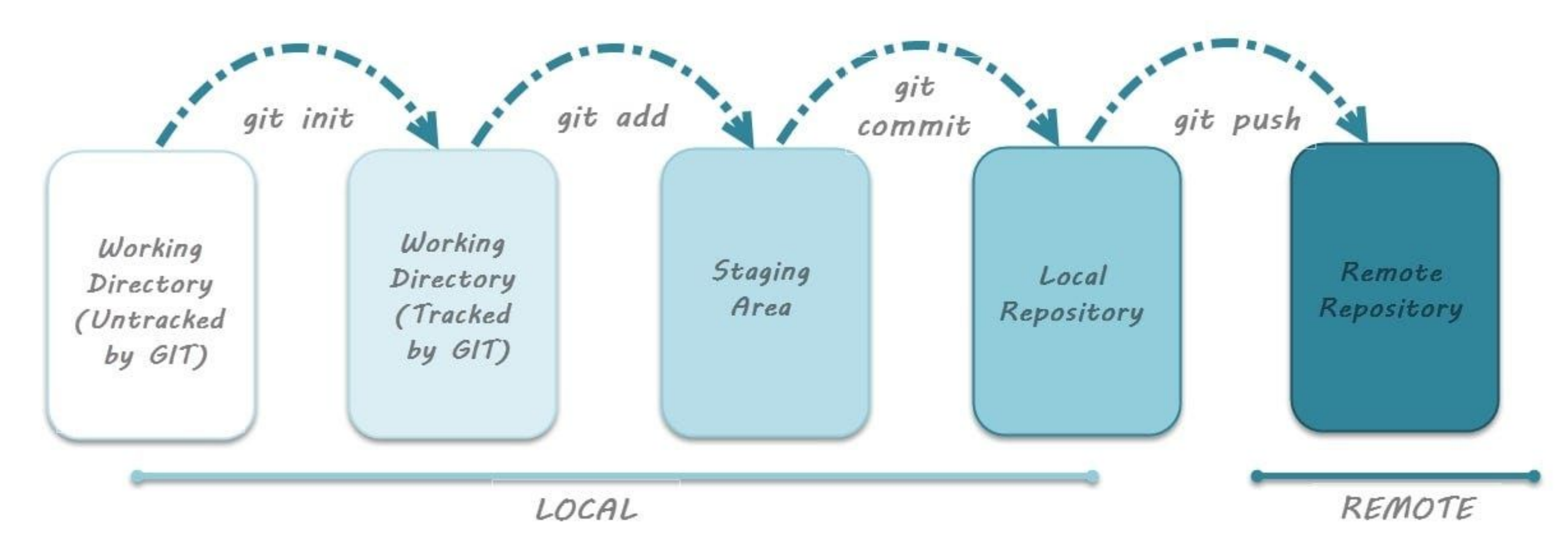
- Displays the commit history of the repository.
- Shows commit messages, authors, and timestamps.

5.9 `git push & git pull`

- git push
  - Uploads local commits to a remote repository.
  - Typically pushes to a specific branch.
- git pull
  - Fetches and merges changes from a remote repository into the local branch.
  - Combines `git fetch` and `git merge`.



Summary


git init
git add
git commit
git push
Working Directory (Untracked by GIT)
Working Directory (Tracked by GIT)
Staging Area
Local Repository
Remote Repository
LOCAL
REMOTE