Git Hub Terminal Command

1) Git config Command

Why is Git configuration important

When you push code to GitHub:

- Git uses your **name & email** to identify the author
- GitHub matches that email with your account
- Your commits appear under **your profile**

Without configuration:

- Commits may show as **unknown author**
- GitHub may not link commits to your account

2) Clone & Status Command

1) What is `git clone:-`

Command :--

`git clone https://github.com/username/project-name.git`

git clone is used to copy an existing remote repository (usually from GitHub) to your local computer.
It downloads the entire project with its files, history, and branches.

2) git status Command:-

Command :- **git status**

git status shows the current state of your working directory and staging area.

It tells you:

- Which files are modified
- Which files are staged
- Which files are untracked
- Whether your branch is up to date

3) Cd Command :--

cd stands for Change Directory. It is used to move from one folder (directory) to another in the command line (Terminal / Git Bash / CMD).

i) Go inside a folder

cd "Folder Name"

Ex : cd "git titorial"

ii) Go back one directory (Maghachya file Madhie janya sathie)

cd ..

iii) Go to home directory cd ~ (Use home directoryC:\Users\kamb)

~ represents the home folder of the current user.

Command : **cd ~**

iii) Go to root directory

Command : **cd /**

Example :

PS C:\Users\kambl\Desktop\git titorial> cd /

PS C:\>

PS C:\> cd "C:\Users\kambl\Desktop\git titorial"

PS C:\Users\kambl\Desktop\git titorial>

4) **ls** Command

Purpose :-

The l**s** command is used to display files and folders present in the current directory.

ls -Force Command (Show all files (including hidden files**)**

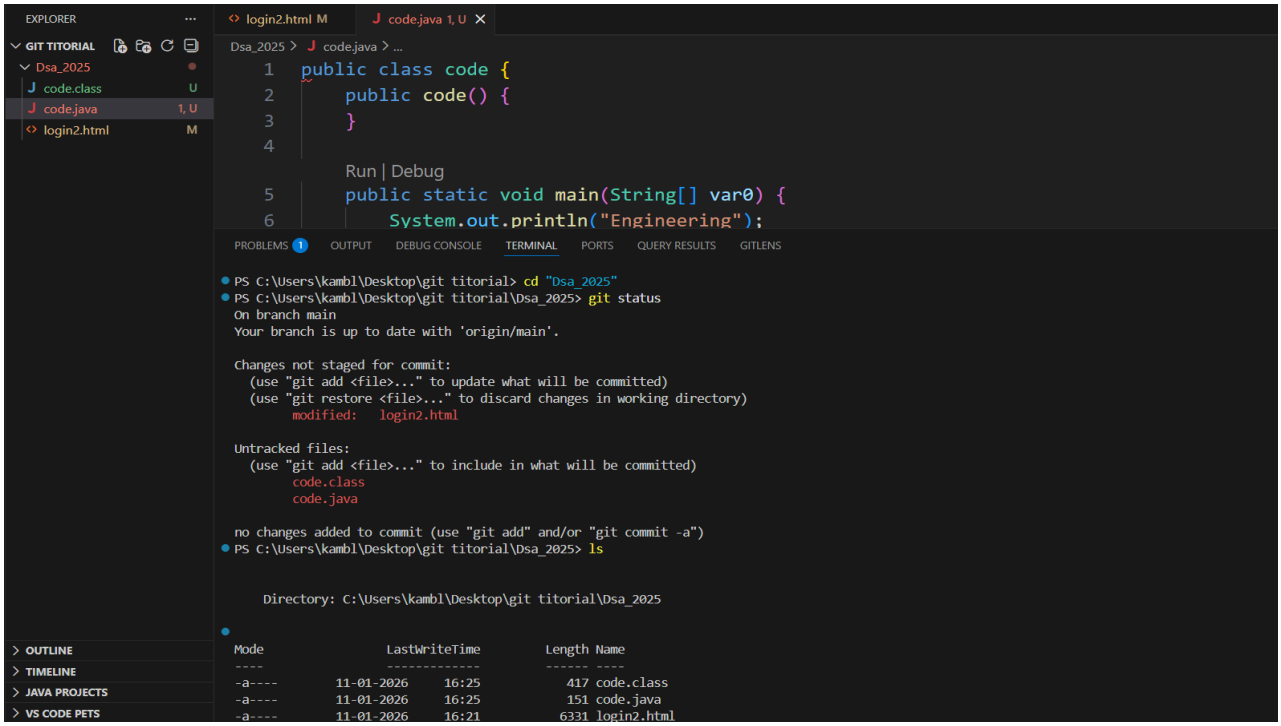
Step After the File Modify (Clone from the Git Hub)

3)

I) git status (shows the current state of your working directory)

II) git add .

git add . is used to add all new and modified files from the current directory and its sub-directories to the staging area.



4) Add & Commit

1) What is **`git add .`**

`git add.` is used to add all new and modified files from the current directory and its sub-directories to the staging area.

I) If You Want to Add the single File only give command

git add File name (Ex: git add code.java)

II) If you want to add All Files from folder give command

git add .

2) What is a commit

A commit is a record **(**snapshot**)** of changes you have made to your project.
It saves the current state of staged files into Git history

Command **:** **git commit -m "some message"**

- commit → saves the changes
- -m → message
- "some message" → description of what you changed

5) Push Command Final Command (From Vs code to Git Hub)

What is push Command

Push means uploading your local commits to the remote repository (GitHub).

Simple meaning:
Local → GitHub

Command syntax **:-**

git push origin main

- git push → send changes to remote
- origin → remote repository name (default)
- main → branch name

6) Step To Remove the Git Hub credentials Authentication

FINAL & GUARANTEED FIX (do exactly this)

STEP 1: Remove cached GitHub credentials (MOST IMPORTANT)

Method A (Best – GUI way)

1. Press Windows + R
2. Type:
3. control keymgr.dll
4. Open Windows Credential

5. Find entries starting with:
    - git:https://github.com
6. Delete ALL GitHub-related entries
7. Close everything

7) Steps To Verify Git identity / configuration (just to confirm)

git config --global user.name (Enter in the git Bash Shell)

git config --global user.email

8) I Push One File to One Git Hub Repository(Name is -Regression File (Time : 1.47))

```
PROBLEMS 1   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   QUERY RESULTS   GITLENS

PS C:\Users\kambl\Desktop\git titorial> cd "dsa_2025"
PS C:\Users\kambl\Desktop\git titorial\dsa_2025> ls


    Directory: C:\Users\kambl\Desktop\git titorial\dsa_2025


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        11-01-2026     18:00            417 code.class
-a----        11-01-2026     16:25            151 code.java
-a----        11-01-2026     21:36           6327 login2.html


PS C:\Users\kambl\Desktop\git titorial\dsa_2025> ls


    Directory: C:\Users\kambl\Desktop\git titorial\dsa_2025


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        11-01-2026     18:00            417 code.class
-a----        11-01-2026     16:25            151 code.java
-a----        11-01-2026     21:36           6327 login2.html
-a----        12-01-2026     13:42           1051 regression.py
```

```
PS C:\Users\kambl\Desktop\git titorial\dsa_2025> git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        regression.py

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\kambl\Desktop\git titorial\dsa_2025> git add .
PS C:\Users\kambl\Desktop\git titorial\dsa_2025> git commit -m "ml model"
PS C:\Users\kambl\Desktop\git titorial\dsa_2025> git add .
PS C:\Users\kambl\Desktop\git titorial\dsa_2025> git add .
PS C:\Users\kambl\Desktop\git titorial\dsa_2025> git add .
PS C:\Users\kambl\Desktop\git titorial\dsa_2025> git commit -m "ml model"
[main d37e46f] ml model
 1 file changed, 41 insertions(+)
 create mode 100644 regression.py
PS C:\Users\kambl\Desktop\git titorial\dsa_2025> git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 798 bytes | 399.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/RITESHKAMBLE7793/Dsa_2025.git
   df83ffe..d37e46f  main -> main
PS C:\Users\kambl\Desktop\git titorial\dsa_2025>
```

9) If You Want To Create Folder with Terminal In Vs_code Terminal (go To Vs terminal & Enter Command)

I) Create one folder

mkdir foldername

Example: **mkdir** regression

II) Create folder with space in name

mkdir "File Name"

Ex : **mkdir** "Java Programs"

III) Create multiple folders at once

mkdir src test docs

10) Step To Make Local Repository To Git Repository (For Push New Foder/Files From Vs To Git Hub website)

- Check Hidden Files With Command : **ls -Force**
- If File/folder Are Not Git repo Then use Command To Make Git Repo : **git init** (Start git for Push Files)

11) Caution While Push the File From Vs To Git Repository

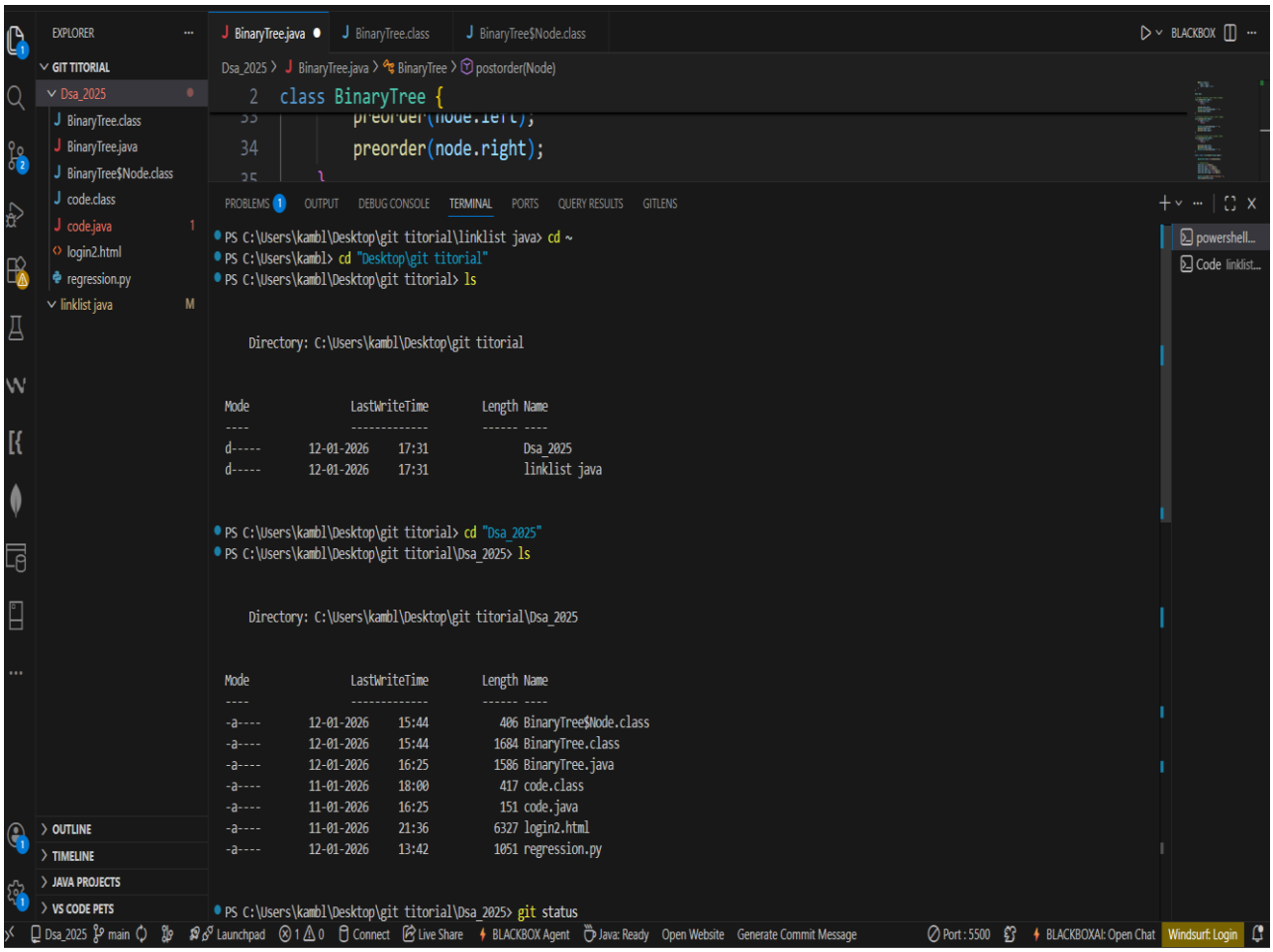
If th Git Hub repo url Already Exist In Local File (Vs Code) &You try to Push Saperate New Folder in same git remote repository that alredy exist (That Not Exist In folder of git repository)
Then New Folder Past into folder of git & then push new file.

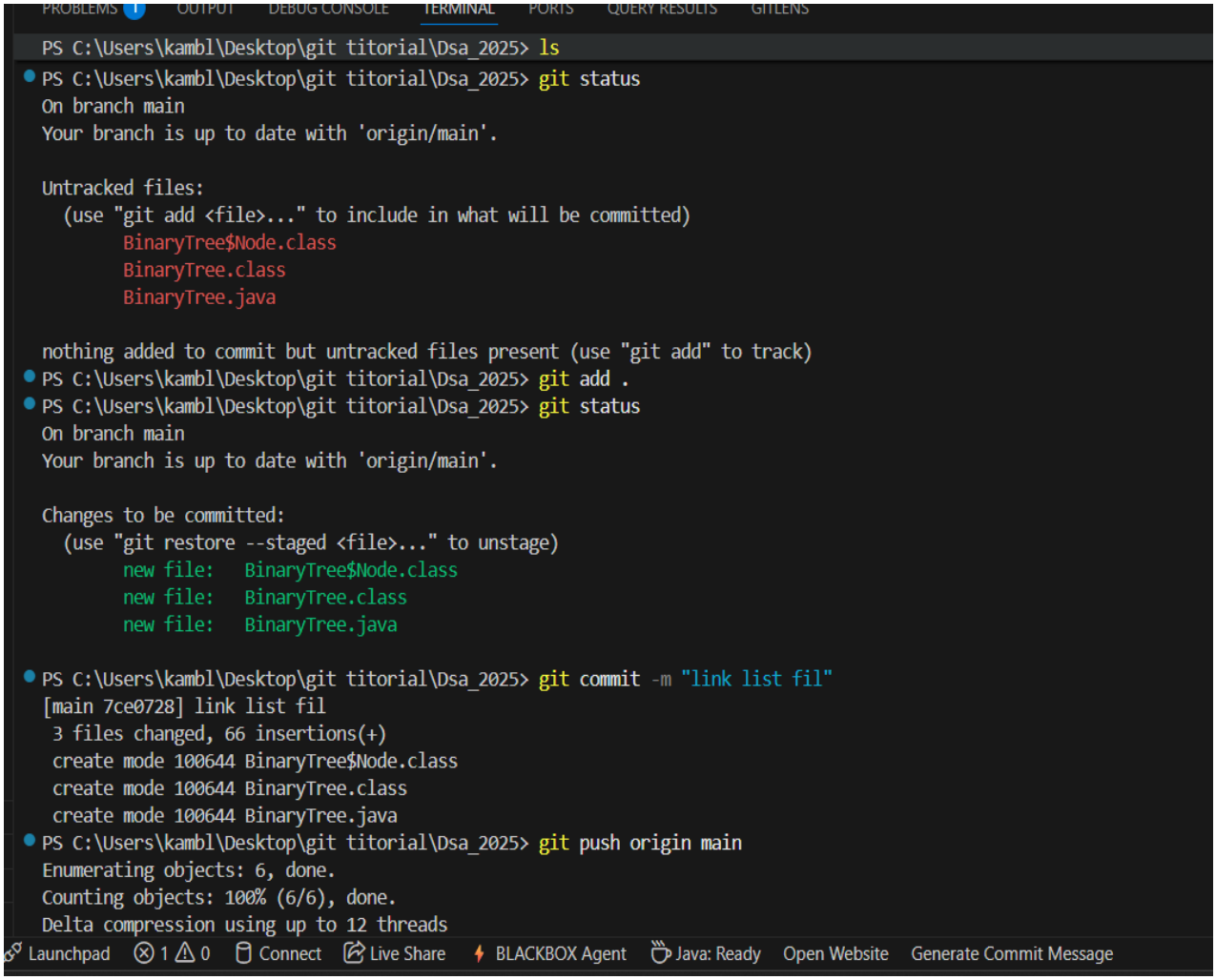
12) Rules While Push the File From Vs to Git Hub

THE CORE RULE YOU MISSED (Very Important)

ONE GitHub repository = ONE .git folder (ONLY at root)

You CANNOT push two separate folders independently to the same repo.




PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS GITLENS
PS C:\Users\kambl\Desktop\git titorial\Dsa_2025> ls
PS C:\Users\kambl\Desktop\git titorial\Dsa_2025> git status
On branch main
Your branch is up to date with 'origin/main'.
Untracked files:
(use "git add <file>..." to include in what will be committed)
BinaryTree$Node.class
BinaryTree.class
BinaryTree.java
nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\kambl\Desktop\git titorial\Dsa_2025> git add .
PS C:\Users\kambl\Desktop\git titorial\Dsa_2025> git status
On branch main
Your branch is up to date with 'origin/main'.
Changes to be committed:
(use "git restore --staged <file>..." to unstage)
new file: BinaryTree$Node.class
new file: BinaryTree.class
new file: BinaryTree.java
PS C:\Users\kambl\Desktop\git titorial\Dsa_2025> git commit -m "link list fil"
[main 7ce0728] link list fil
3 files changed, 66 insertions(+)
create mode 100644 BinaryTree$Node.class
create mode 100644 BinaryTree.class
create mode 100644 BinaryTree.java
PS C:\Users\kambl\Desktop\git titorial\Dsa_2025> git push origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Launchpad 1 0 Connect Live Share BLACKBOX Agent Java: Ready Open Website Generate Commit Message

13) Init Command (While Use When Push File From Vs to Git hub)

I) git init command

git init is used to create a new Git repository in a folder.

Git creates a hidden .git folder

That folder stores all version history, commits, branches, config

After this command, the folder becomes a Git-tracked project**.**

II) git remote add origin <link>

It connects your local Git repository to a remote repository on GitHub.

III) git remote -v

What it do:

Verifies the remote connection

Example output

origin https://github.com/username/repo.git (fetch)

origin https://github.com/username/repo.git (push)

IV) git branch

Shows all local branches

Highlights the current branch

Example output :

* master

* main

V) git branch -M main

Renames the current branch to main

Why this is needed

GitHub's default branch is **main**

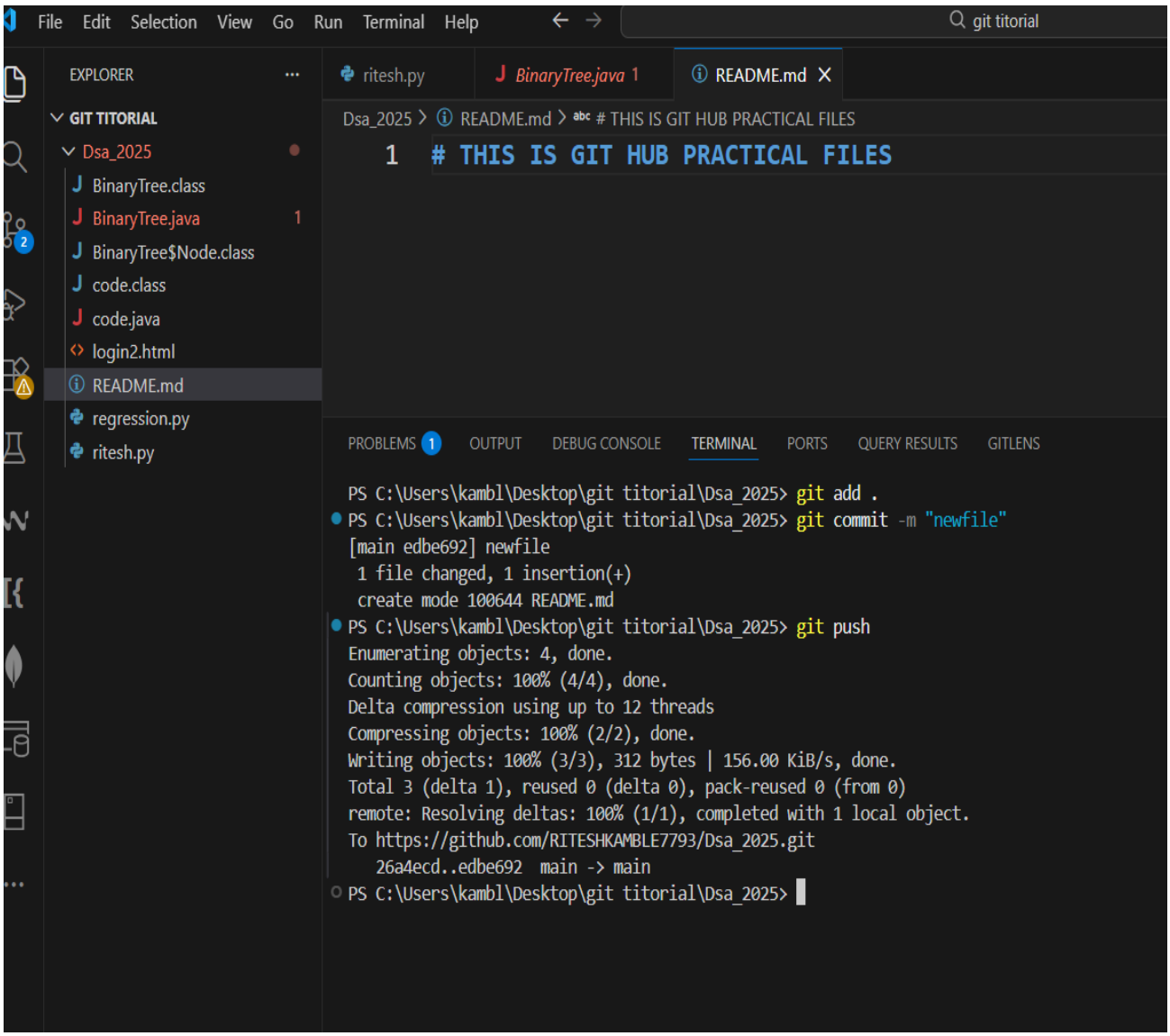
Older Git versions use **master**

Vi) git push origin main Or

git push -u origin main (Final Step)

Uploads your local commits to GitHub

14) Step To Add README.md File For " massage "



15)Summary: Difference between Git & GitHub

- Git is a version control system used to track changes in files on your local machine.
  It works offline and manages code history using the .git folder.
- GitHub is a cloud-based platform that hosts Git repositories online.
  It helps with backup, sharing, and team collaboration.

👉 Git = tool

👉 GitHub = service using Git

16) Branch Commands .

I) git branch

Purpose: Check branches

- Lists all local branches
- shows the current branch

II) git branch -M main

Purpose: Rename branch

- Renames the current branch to main
- Used to match GitHub's default branch
- -M forces rename if needed

III) git checkout <branch-name>

Purpose: Navigate (switch) branch

- Moves you from one branch to another
- Does not create a new branch

IV) git checkout -b <new-branch-name>

Purpose: Create + switch branch

- Creates a new branch
- Immediately switches to it
- Most commonly used command

V) git branch -d <branch-name>

Purpose: Delete branch

- Deletes a local branch
- Works only if the branch is already merged

17) Pull Command

Commnad **: git pull origin main**

git pull downloads changes from a remote repository and merges them into your current branch**.**

(this command used when we merge the two branch file &we want to download from remote to local)

18) Undo changes Command

1) You edited a file but want to discard changes.

git restore <file>

Example : git restore ritesh.html

2) Undo ALL unstaged changes

Resets all modified (unstaged) files

git restore .

3) Undo the LAST commit (keep changes)

You committed by mistake but want to edit and recommit.

git reset --soft HEAD~1