**GIT/GITHUB/GITBASH:**

**1. Git is a version controller .**

**2. GitHub is a platform to share the projects & repositories with one or more people.**

**3. Git Bash (CLI) is a command line.**

**THERE ARE 2 WAYS TO CREATE:**

**1.locally to remote**

**2.remote to locally**

LAB-1:

**A screen shot of a computer

AI-generated content may be incorrect.**

**Commands and Their Purpose**

1. **ls  
   ➤ Lists all files and directories in the current directory.**
2. **touch mm.java  
   ➤ Creates an empty file named mm.java.**
3. **git init  
   ➤ Initializes a new Git repository in the current directory.**
4. **git add mm.java  
   ➤ Stages the file mm.java for the next commit.**

A computer screen with text on it

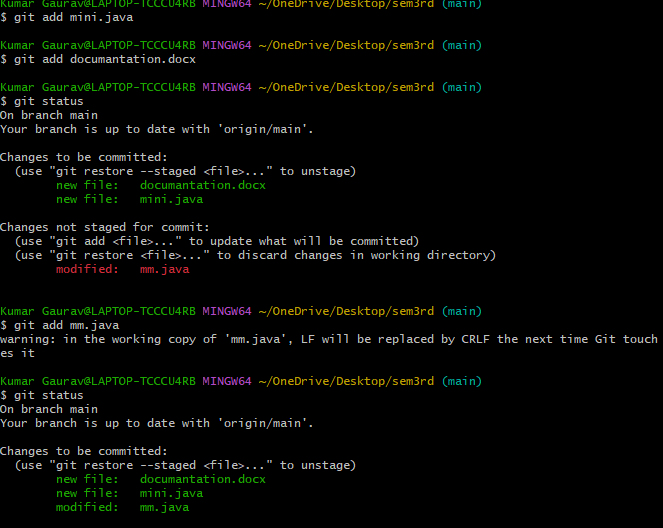
AI-generated content may be incorrect.

1. git status  
   ➤ Shows that mm.java is **staged** and ready to be committed.
2. git commit -m "first commit"  
   ➤ Saves the staged changes to the Git repo with the message **"first commit"**.

A computer screen shot of a computer code

AI-generated content may be incorrect.

1. **git branch -m main**  
   ➤ Renames the current branch from master to main.
2. **git remote add origin https://github.com/mitalisingh5/sem3rd.git**  
   ➤ Links your local repo to the remote repo on GitHub named origin.
3. **git push -u origin main**  
   ➤ Pushes your local main branch to the origin (GitHub), and sets up tracking so future git push/pull knows where to go.



1. **git add mini.java**  
   ➤ Stages the new file mini.java for commit.
2. **git add documantation.docx**  
   ➤ Stages the file documantation.docx.
3. **git status**  
   ➤ Shows:

* Files staged for commit: mini.java, documantation.docx
* File modified but not yet staged: mm.java

1. **git add mm.java**  
   ➤ Stages the **modified** mm.java file for commit.

A screenshot of a computer

AI-generated content may be incorrect.

1. **git commit -m**  
   ➤ (First attempt failed) — missing commit message; Git requires a string after -m.
2. **git commit -m "second commit"**  
   ➤ Creates a new commit with the message **"second commit"**, including all staged changes.
3. **git push -u origin main**  
   ➤ Pushes your **main** branch to GitHub and sets upstream tracking with **origin/main**.

LAB-2:

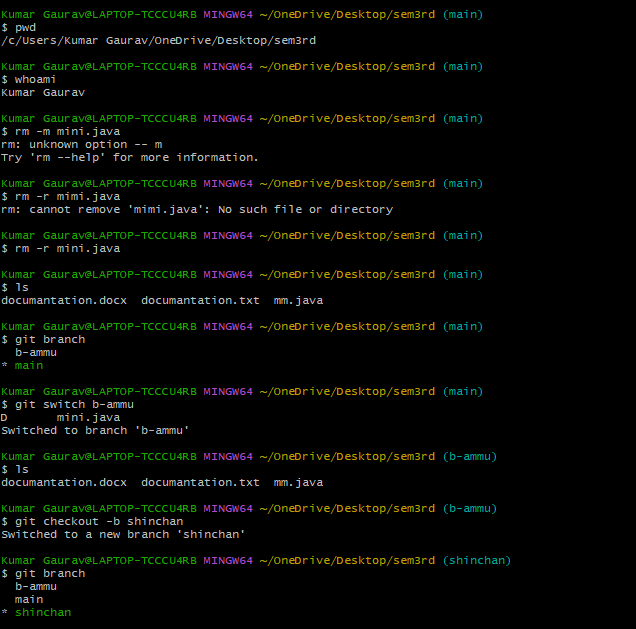
A screen shot of a computer

AI-generated content may be incorrect.

19. **git branch b-ammu**  
➤ Creates a new local branch named b-ammu.

20. **git branch**  
➤ Lists all branches; the one with \* (main) is the currently active branch.

21. **ls -a**  
➤ Lists **all files** in the directory, including **hidden files** like .git.



22. **pwd**  
➤ Prints the current working directory (/sem3rd folder in this case).

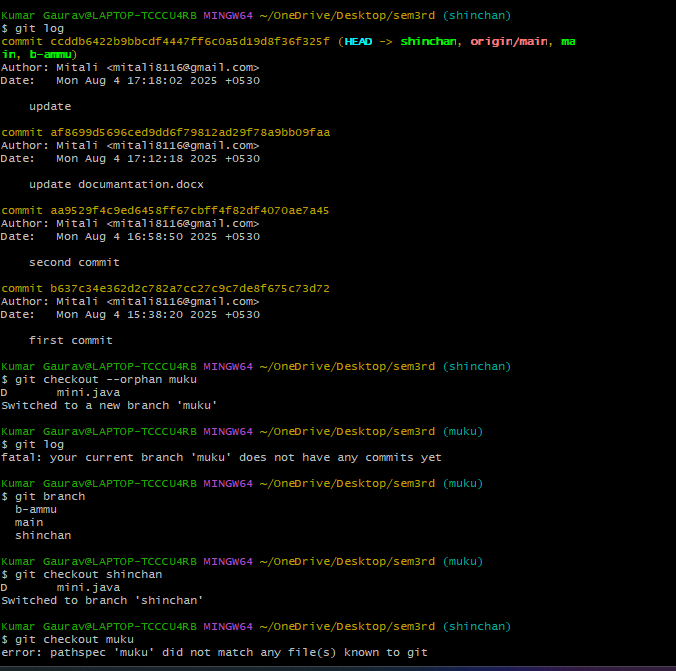
23. **whoami**  
➤ Shows your current system username (Kumar Gaurav).

24**. rm -r mini.java**  
➤ Removes the file mini.java.

25. **git switch b-ammu**  
➤ Switches from main to b-ammu branch.

26. **git checkout -b shinchan**  
➤ Creates and switches to a new branch named shinchan.

27. **git branch**  
➤ Shows all 3 branches now: main, b-ammu, shinchan (you’re now on shinchan branch).

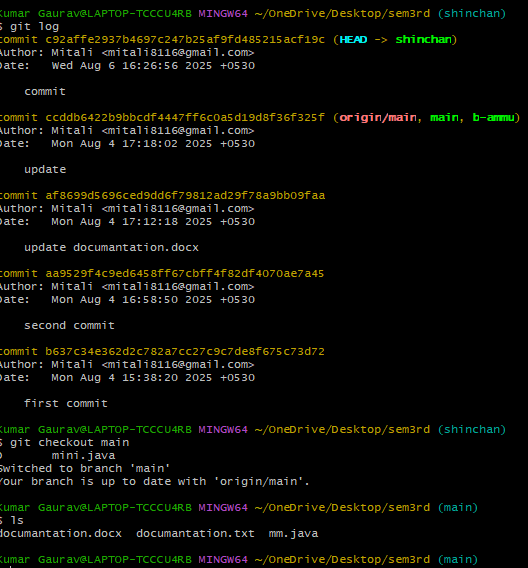


28. **git log**  
➤ Shows the commit history (author, date, message, commit ID).

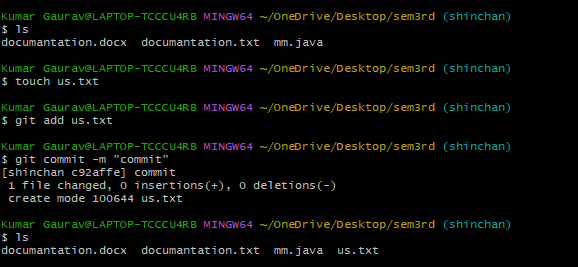
29. **git checkout --orphan muku**  
➤ Creates a new branch muku without any commit history (clean slate).

30. **git checkout shinchan**  
➤ Switches back to the existing branch named shinchan.

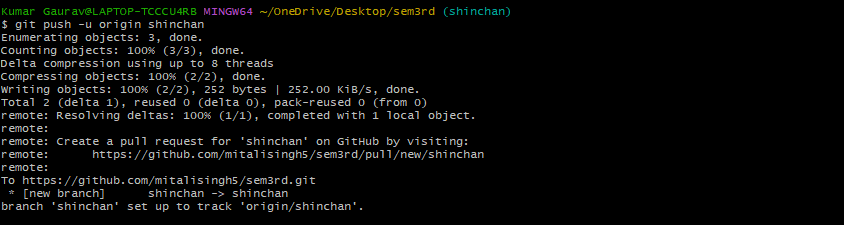
31. **git checkout muku**  
➤ Tries to switch to branch muku, but fails since it wasn't committed properly.



We switch to branch(Shinchan)next:-[git checkout Shinchan]



\*We make a new file(us.txt),commit it & push it.\*

NOW, switch to branch main:-

A computer screen shot of text

AI-generated content may be incorrect.

32. git branch -d b-ammu  
 ➤ Deletes the local branch b-ammu.

33. git branch  
➤ Again lists local branches to confirm deletion.

34.git push origin --delete shinchan  
➤ Deletes the shinchan branch from the **remote GitHub repository**.

35. git branch -a  
➤ Lists **all branches**, including local and remote (remotes/origin/main)