

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
“Jnana Sangama”, Machhe, Belagavi, Karnataka-590018



Lab Experiment Record
Project Management with Git [BCSL358C]
Submitted in partial fulfillment towards AEC of 3rd semester of
Bachelor of Engineering
in
Computer Science and Engineering
(Artificial Intelligence & Machine Learning)

Submitted by,
Likhitha Shasthry B S
4GW24CIA020



DEPARTMENT OF CSE (Artificial Intelligence & Machine Learning)
GSSS INSTITUTE OF ENGINEERING & TECHNOLOGY FOR WOMEN
(Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi & Govt. of Karnataka)
K.R.S ROAD, METAGALLI, MYSURU-570016, KARNATAKA
(Accredited by NAAC)

2025-26

SL no	CONTENT	PAGE NO
1.	Setting Up and Basic Commands. Initialize a new Git repository in a directory. Create a new file and add it to the staging area and commit the changes with an appropriate commit message.	4
2.	Creating and Meraging Branches. Create a new branch named "feature-branch." Switch to the "master" branch. Merge the "feature-branch" into "master."	4-5
3.	Creating and Managing Branches. Write the commands to stash your changes, switch branches, and then apply the stashed changes.	5-6
4.	Collaboration and Remote Repositories . Clone a remote Git repository to your local machine.	6
5.	Collaboration and Remote Repositories. Fetch the latest changes from a remote repository and rebase your local branch onto the updated remote branch.	6-7
6.	Collaboration and Remote Repositories . Write the command to merge "feature-branch" into "master" while providing a custom commit message for the merge.	7-8
7.	Git Tags and Releases. Write the command to create a lightweight Git tag named "v1.0" for a commit in your local repository.	8-9
8.	Advanced Git Operations Write the command to cherry-pick a range of commits from "source-branch" to the current branch.	9-10
9.	Analysing and Changing Git History. Given a commit ID, how would you use Git to view the details of that specific commit, including the author, date, and commit message?	10-11
10.	Analysing and Changing Git History. Write the command to list all commits made by the author "JohnDoe" between "2023-01-01" and "2023-12-31."	11-12

11.	Analysing and Changing Git History . Write the command to display the last five commits in the repository's history.	13
12.	Analysing and Changing Git History. Write the command to undo the changes introduced by the commit with the ID "abc123".	13-14

APPENDIX

For full reference and verification , Git repository link is given below.

<https://github.com/likhithashasthry/4GW24CI020.git>

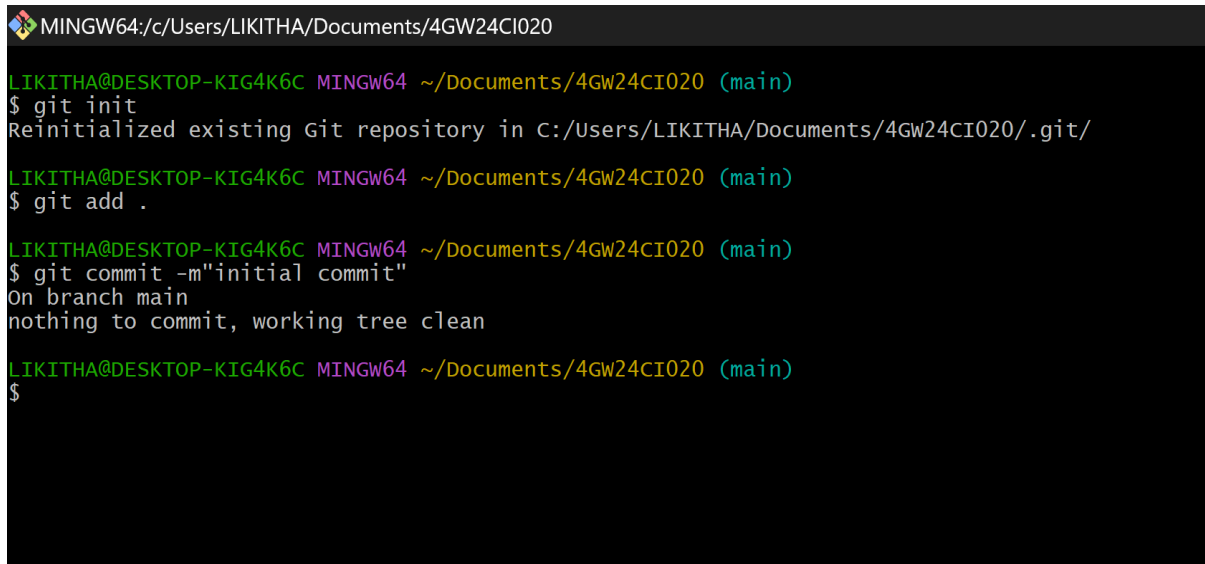
Project Management Using GIT

Lab Manual

1. Setting Up and Basic Commands

Initialize a new Git repository in a directory. Create a new file and add it to the staging area and commit the changes with an appropriate commit message.

- Open the directory where file has to be created.
- Initialize first using the command '**git init**'.
- Then create any file in the directory.
- Then using the command '**git add .**' to add all the files in that directory or '**git add filename.extention**'
- to stage the files/file for the upcoming commit.
- Then using the command '**git commit -m "any message"**' give the appropriate purpose of the changes.



```
MINGW64:/c:/Users/LIKITHA/Documents/4GW24CI020
LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git init
Reinitialized existing Git repository in C:/Users/LIKITHA/Documents/4GW24CI020/.git/
LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git add .
LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git commit -m"initial commit"
On branch main
nothing to commit, working tree clean
LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$
```

2. Creating and Merging Branches.

Create a new branch named "feature-branch." Switch to the "master" branch. Merge the "feature-branch" into "master."

- Create a new branch with name "feature-branch" using the command **git checkout -b feature-branch**.
- Then do so modification into the existing file or create a new file.
- Then add it and commit it using the commands **git add .** and **git commit -m**.

```

MINGW64:/c/Users/LIKITHA/Documents/4GW24CI020
Already on 'main'

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git checkout -b feature branch
fatal: 'branch' is not a commit and a branch 'feature' cannot be created from it

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git checkout -b feature-branch
fatal: a branch named 'feature-branch' already exists

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git checkout -b branch2
Switched to a new branch 'branch2'

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (branch2)
$ echo "git hub manual">README.md

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (branch2)
$ git status
On branch branch2
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:   README.md

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

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (branch2)
$ git add README.md
fatal: pathspec 'README.md' did not match any files

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (branch2)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (branch2)
$ git status
On branch branch2
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (branch2)
$ git commit -m"modified in the new branch"
[branch2 af4cf22] modified in the new branch
 1 file changed, 1 insertion(+), 1 deletion(-)

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (branch2)
$ git checkout main
Switched to branch 'main'

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git merge branch2
Updating 6763464..af4cf22
Fast-forward
 README.md | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ |

```

3. Creating and Managing Branches

Write the commands to stash your changes, switch branches, and then apply the stashed changes.

- Modify any file/files.
- Then save the changes into the stash using the command `git stash save`.
- Then switch to any branch required.
- Then apply these changes.

- Note the stash is like a stack there can be any number of changes , so if there are more than one then while apply specifying which branch to apply.

```

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git stash save "changes saved"
No local changes to save

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ echo "changes made to apply stash">README.md

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git stash save "changes saved"
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it
Saved working directory and index state On main: changes saved

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git checkout branch2
Switched to branch 'branch2'

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (branch2)
$ git stash apply
On branch branch2
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:   README.md

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

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (branch2)
$

```

4. Collaboration and Remote Repositories .

Clone a remote Git repository to your local machine.

- Get the link of your repository.
- Then using the command `git clone <repo url>` a new folder same as repo name will be created.

```

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (branch2)
$ git clone https://github.com/likhithashasthry/gitManual.git
Cloning into 'gitManual'...
warning: You appear to have cloned an empty repository.

```

```

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (branch2)
$

```

5. Collaboration and Remote Repositories.

Fetch the latest changes from a remote repository and rebase your local branch onto the updated remote branch.

- Be in the local branch that has to be rebased.

- Then if necessary to change the branch do it using the command **git checkout <branch name>**
- Then fetch the origin using the command **git fetch origin**
- Then rebase it using the rebase command
- Then push all the files into the repository using the command **git push origin main**
- **Rebase mean** git places takes the commit and places then on the top of the latest commit.

```

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git add README.md

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git commit -m "changes for stash saved"
[main 708ede5] changes for stash saved
1 file changed, 1 insertion(+), 1 deletion(-)

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git rebase main
Current branch main is up to date.

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git rebase --continue
fatal: no rebase in progress

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git push origin main
Enumerating objects: 20, done.
Counting objects: 100% (20/20), done.
Delta compression using up to 12 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (20/20), 199.42 KiB | 5.87 MiB/s, done.
Total 20 (delta 3), reused 3 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/likhithashashtry/gitManual.git
 * [new branch]      main -> main

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)

```

6. Collaboration and Remote Repositories .

Write the command to merge "feature-branch" into "master" while providing a custom commit message for the merge.

- Be in the main branch.
- Then merge with another branch using the command **git merge <new branch>-m "any message"**.

- If it shows already upto date but cannot find it in the github , it means its still in the local repo and not pushed so to push it use the command `git push -u origin <new branch>`.

```

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git checkout main
Already on 'main'

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git merge branch2 -m "mergerd another branch"
Already up to date.

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git checkout branch2
Switched to branch 'branch2'

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (branch2)
$ git push -u origin branch2
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'branch2' on GitHub by visiting:
remote:   https://github.com/likhithashasthry/gitManual/pull/new/branch2
remote:
To https://github.com/likhithashasthry/gitManual.git
 * [new branch]      branch2 -> branch2
branch 'branch2' set up to track 'origin/branch2'.

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (branch2)
$ git checkout main
Switched to branch 'main'

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$

```

7. Git Tags and Releases.

Write the command to create a lightweight Git tag named "v1.0" for a commit in your local repository.

- Create a tag using a **command `git tag <version>`**. This creates tags for the latest commit only.
- If you want to create tag to any other commit then each commit will have its own secure hash algorithm 1 (SHA 1).
- Get that using the command **`git log --oneline`**.
- Then add the tag using the command **`git tag <version tag> SHA 1`**.


```

MINGW64:/c/Users/LIKITHA/Documents/4GW24CI020

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git tag v1.0

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git log --oneline
708ede5 (HEAD -> main, tag: v1.0, origin/main) changes for stash saved
af4cf22 (origin/branch2, branch2) modified in the new branch
6763464 new branch created
5d99ce3 (feature-branch) files pushes
7692577 Create README.md

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git taf v1.1 commit- af4cf22
git: 'taf' is not a git command. See 'git --help'.

The most similar command is
    tag

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git tag v1.1 commit- af4cf22
fatal: too many arguments

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git tag v1.1 af4cf22

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git push origin --tags
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/likhithashasthry/gitManual.git
 * [new tag]          v1.0 -> v1.0
 * [new tag]          v1.1 -> v1.1

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ |

```

8. Advanced Git Operations

Write the command to cherry-pick a range of commits from "source-branch" to the current branch.

- This is used to cherry-pick a range of commits from source branch to current branch.
- Modification has to be made in one branch then it has be committed.
- Then it has to be cherry picked in another branch.
- Note that if that commit is already pushed then cherry pick is not possible and also here for the commits SHA 1 has to be used.

```

fatal: cherry-pick failed

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (branch2)
$ echo "Changes made to use cherry-pick in another branch">README.md

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (branch2)
$ git add .
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (branch2)
$ git commit -m "new changed"
[branch2 85789ec] new changed
1 file changed, 1 insertion(+), 1 deletion(-)

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (branch2)
$ git log --oneline
85789ec (HEAD -> branch2) new changed
af4cf22 (tag: v1.1, origin/branch2) modified in the new branch
6763464 new branch created
5d99ce3 (feature-branch) files pushes
7692577 Create README.md

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (branch2)
$ git cherry-pick 649e108
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
error: could not apply 649e108... changed to get range for cherry-pick
hint: After resolving the conflicts, mark them with
hint: "git add/rm <paths>", then run
hint: "git cherry-pick --continue".
hint: You can instead skip this commit with "git cherry-pick --skip".
hint: To abort and get back to the state before "git cherry-pick",
hint: run "git cherry-pick --abort".
hint: Disable this message with "git config set advice.mergeConflict false"

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (branch2|CHERRY-PICKING)
$

```

- Here cherry is picking there are some errors occurred in the file that modifies , it has to be corrected manually.

```

MINGW64:/c/Users/LIKITHA/Documents/4GW24CI020

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (branch2|CHERRY-PICKING)
$ git log --oneline
85789ec (HEAD -> branch2) new changed
af4cf22 (tag: v1.1, origin/branch2) modified in the new branch
6763464 new branch created
5d99ce3 (feature-branch) files pushes
7692577 Create README.md

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (branch2|CHERRY-PICKING)
$ git checkout main
README.md: needs merge
error: you need to resolve your current index first

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (branch2|CHERRY-PICKING)
$ git add README.md

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (branch2|CHERRY-PICKING)
$ git cherry-pick --continue
hint: Waiting for your editor to close the file...
[branch2 c24c374] changed to get range for cherry-pick
Date: Mon Jan 5 23:04:36 2026 +0530
1 file changed, 3 insertions(+)

```

9. Analysing and Changing Git History.

Given a commit ID, how would you use Git to view the details of that specific commit, including the author, date, and commit message?

- There are 2 command to see the history pf any commit , that is **git show SHA 1** , gives the complete history of that commit .
- Another one is by using **git -n 1 SHA 1**, this displays only of 1 commit.

```

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020
$ git checkout main
Switched to branch 'main'

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git log --oneline
649e108 (HEAD -> main) changed to get range for cherry-pick
053726a changed to use cherry-pick
708ede5 (tag: v1.0, origin/main) changes for stash saved
af4cf22 (tag: v1.1, origin/branch2) modified in the new branch
6763464 new branch created
5d99ce3 (feature-branch) files pushes
7692577 Create README.md

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git show 708ede5
commit 708ede577a890fa082a3e939def9b1f35f78980a (tag: v1.0, origin/main)
Author: likhithashasthry <likhithashasthry@gmail.com>
Date: Mon Jan 5 21:44:49 2026 +0530

    changes for stash saved

diff --git a/README.md b/README.md
index ec63f13..002b3be 100644
--- a/README.md
+++ b/README.md
@@ -1,1 @@
- git hub manual
+ changes made to apply stash

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git log -n 1 708ede5
commit 708ede577a890fa082a3e939def9b1f35f78980a (tag: v1.0, origin/main)
Author: likhithashasthry <likhithashasthry@gmail.com>
Date: Mon Jan 5 21:44:49 2026 +0530

    changes for stash saved

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ |

```

10. Analysing and Changing Git History.

Write the command to list all commits made by the author "JohnDoe" between "2023-01-01" and "2023-12-31."

- Using the author name , since and until date , list of changes can be obtained.
- Note that here both the dated should not be same.

```

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git log --author="likhithashasthry" --since="2026-01-05" --until="2026-01-06"
commit f70464fc3ec50b3f884fc3e7a9f794fb7abfe5c3 (HEAD -> main, origin/main)
Author: likhithashasthry <likhithashasthry@gmail.com>
Date: Tue Jan 6 07:51:06 2026 +0530

    pushed today

commit d6be947131dc210a1b809758ff024af910174075
Author: likhithashasthry <likhithashasthry@gmail.com>
Date: Mon Jan 5 23:39:00 2026 +0530

    Revert "new branch created"

    This reverts commit 6763464e1783d53ed92f32825cbb8479c65a5cb5.

commit 649e108c999e774f25a50ce0aa878eb8b01dd8ab
Author: likhithashasthry <likhithashasthry@gmail.com>
Date: Mon Jan 5 23:04:36 2026 +0530

    changed to get range for cherry-pick

commit 053726aea984937ac6340cca656f3ff12e8910eb
Author: likhithashasthry <likhithashasthry@gmail.com>
Date: Mon Jan 5 23:03:42 2026 +0530

    changed to use cherry-pick

commit 708ede577a890fa082a3e939def9b1f35f78980a (tag: v1.0)
Author: likhithashasthry <likhithashasthry@gmail.com>
Date: Mon Jan 5 21:44:49 2026 +0530

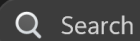
    changes for stash saved

commit af4cf22e487a0e15b951a04dee6aaa1f3fe0b267 (tag: v1.1, origin/branch2)
Author: likhithashasthry <likhithashasthry@gmail.com>
Date: Mon Jan 5 19:27:38 2026 +0530

    modified in the new branch

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ |

```



11. Analysing and Changing Git History .

Write the command to display the last five commits in the repository's history.

- Last 5 commits can be listed using the command `git log -n 5`.

```

MINGW64:/c/Users/LIKITHA/Documents/4GW24CI020

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git log --author="likhithashasthry" --since='2026-01-05' --until='2026-01-05'

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git log -n 5
commit 649e108c999e774f25a50ce0aa878eb8b01dd8ab (HEAD -> main)
Author: likhithashasthry <likhithashasthry@gmail.com>
Date: Mon Jan 5 23:04:36 2026 +0530

    changed to get range for cherry-pick

commit 053726aea984937ac6340cca656f3ff12e8910eb
Author: likhithashasthry <likhithashasthry@gmail.com>
Date: Mon Jan 5 23:03:42 2026 +0530

    changed to use cherry-pick

commit 708ede577a890fa082a3e939def9b1f35f78980a (tag: v1.0, origin/main)
Author: likhithashasthry <likhithashasthry@gmail.com>
Date: Mon Jan 5 21:44:49 2026 +0530

    changes for stash saved

commit af4cf22e487a0e15b951a04dee6aaa1f3fe0b267 (tag: v1.1, origin/branch2)
Author: likhithashasthry <likhithashasthry@gmail.com>
Date: Mon Jan 5 19:27:38 2026 +0530

    modified in the new branch

commit 6763464e1783d53ed92f32825cbb8479c65a5cb5
Author: likhithashasthry <likhithashasthry@gmail.com>
Date: Sat Jan 3 11:40:29 2026 +0530

    new branch created

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ |

```

12. Analysing and Changing Git History.

Write the command to undo the changes introduced by the commit with the ID "abc123".

- Undo of the commit can be done using the command `git revert SHA 1`.

```
MINGW64:/c/Users/LIKITHA/Documents/4GW24CI020

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git log --onelinr
fatal: unrecognized argument: --onelinr

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git log --oneline
649e108 (HEAD -> main) changed to get range for cherry-pick
053726a changed to use cherry-pick
708ede5 (tag: v1.0, origin/main) changes for stash saved
af4cf22 (tag: v1.1, origin/branch2) modified in the new branch
6763464 new branch created
5d99ce3 (feature-branch) files pushes
7692577 Create README.md

LIKITHA@DESKTOP-KIG4K6C MINGW64 ~/Documents/4GW24CI020 (main)
$ git revert 6763464
hint: waiting for your editor to close the file...
[main d6be947] Revert "new branch created"
 1 file changed, 63 deletions(-)
 delete mode 100644 banker.c
```

