



Git Practical Exam (LMS)

Parth Shah

Simform Solutions

Overview

This exercise demonstrates practical Git workflows using a combination of git-flow, feature branching, and advanced operations like **squash**, **rebase**, **reset**, and **cherry-pick**.

Goals

- Learn to **initialize**, **clone** and use **Git-flow** in a repository.
- Practice creating **features** and **sub-branches**.
- Apply advanced Git commands: **squash**, **rebase**, **reset**, **cherry-pick** and keep the **pull requests (PRs)** clean and concise.

- Understand **branch management** when previous PRs are under review.
- Maintain a **version history** by tagging builds.
- Perform **commit message** updates and manage **branch deletions** safely.

Specifications

Repository setup

- Created a new repository using a template.

Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).
Required fields are marked with an asterisk (*).

1 General

Owner * parthshahsimform / **Repository name *** LMS_Practical_Exam
✓ LMS_Practical_Exam is available.

Great repository names are short and memorable. How about [animated-doodle?](#)

Description
Demonstrate practical Git skills including repository setup, branching strategies, commit management, pul
251 / 350 characters

2 Configuration

Choose visibility * Public
Choose who can see and commit to this repository

Add README On ☒
READMEs can be used as longer descriptions. [About READMEs](#)

Add .gitignore No .gitignore
.gitignore tells git which files not to track. [About ignoring files](#)

Add license No license
Licenses explain how others can use your code. [About licenses](#)

Create repository

- Clone the Repo and Initialize the Git-flow.

```
C:\Users\parthk.shah>cd Documents

C:\Users\parthk.shah\Documents>git clone https://github.com/parthshah02/LMS_Basics_of_Git.git
Cloning into 'LMS_Basics_of_Git'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 15 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (15/15), 5.43 KiB | 2.71 MiB/s, done.
Resolving deltas: 100% (1/1), done.
```

```
MINGW64:/c/Users/parthk.shah

parthk.shah@SF-CPU-821 MINGW64 ~
$ git flow init
Initialized empty Git repository in C:/Users/parthk.shah/.git/
No branches exist yet. Base branches must be created now.
Branch name for production releases: [master]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/] Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
```

Feature branch workflow

Create a feature branch from develop for project setup:

```
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> git checkout -b TE-T84_Project
Switched to a new branch 'TE-T84_Project'
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> git commit -m "created a proper zoho-id feature branch"
On branch TE-T84_Project
```

LMS_PRAC_EXAM Public

feature/TE-T84_another had recent pushes 32 seconds ago [Compare & pull request](#)

main 5 Branches 1 Tag [Go to file](#) [Add file](#) [Code](#)

parthshahsimform Merge pull request #1 from parthshahsimform/feature/login 704dd45 · 17 hours ago 3 Commits

File	Commit	Time
README.md	Initial commit	18 hours ago
abc.txt	abc	17 hours ago

README

LMS_PRAC_EXAM

Demonstrate practical Git skills including repository setup, branching strategies, commit management, pull request handling, and version tagging. You will practice using Git commands like rebase, cherry-pick, reset, and squash in real-world scenarios.

About

Demonstrate practical Git skills including repository setup, branching strategies, commit management, pull request handling, and version tagging. You will practice using Git commands like rebase, cherry-pick, reset, and squash in real-world scenarios.

Releases

1 tags [Create a new release](#)

Packages

No packages published [Publish your first package](#)

// feature branch to develop

```
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> git add bug.txt
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> git commit -m "merging to develop"
[feature/TE-T84_fix e449d41] merging to develop
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 bug.txt
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> git status
On branch feature/TE-T84_fix
nothing to commit, working directory clean
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> git push -u origin feature/TE-T84_fix
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 334 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'feature/TE-T84_fix' on GitHub by visiting:
remote:   https://github.com/parthshahsimform/LMS_PRAC_EXAM/pull/new/feature/TE-T84_fix
remote:
To git@github.com:parthshahsimform/LMS_PRAC_EXAM.git
 * [new branch]      feature/TE-T84_fix -> feature/TE-T84_fix
Branch feature/TE-T84_fix set up to track remote branch feature/TE-T84_fix from origin.
```

Sub-branch for practice (squash, reset, rebase, cherry-pick)

RESET

In Git, reset moves your current branch (HEAD) to a different commit, and depending on the option you use

Soft reset (keeps changes staged)

Mixed reset (keeps changes unstaged)

Hard reset (discards changes)

```

PS C:\Users\parthk.shah\Documents\LMS_Basics_of_Git> git log --oneline
ca8dbc9 docs: add contributing section
77d1d7b Merge pull request #1 from parthshah02/parthshah02-test-1
3d945b9 Add initial content to test_change.txt
fcf856d Add git pull command explanation
f73e8c4 Document Git commands and their usage
ec4ade7 Create basics.txt
e3090ff Updated README.md
274d6cb Initial commit
PS C:\Users\parthk.shah\Documents\LMS_Basics_of_Git>
>> git reset --soft HEAD~1
>> git commit -m "docs: contributing + standards (squashed)"
>>
[TE-T84_practical_lms 6881ffa] docs: contributing + standards (squashed)
1 file changed, 0 insertions(+), 0 deletions(-)
rewrite README.md (100%)
PS C:\Users\parthk.shah\Documents\LMS_Basics_of_Git> git log --oneline
6881ffa docs: contributing + standards (squashed)
77d1d7b Merge pull request #1 from parthshah02/parthshah02-test-1
3d945b9 Add initial content to test_change.txt
fcf856d Add git pull command explanation
f73e8c4 Document Git commands and their usage
ec4ade7 Create basics.txt
e3090ff Updated README.md
274d6cb Initial commit

```

SQUASH

Squash combines multiple commits into a single commit.

Mark the first commit as pick, and the rest as squash (or s) as per the requirements .

```

pick e5c6a38 abc
squash 6f1d2db add line a
squash 812d20a line c

# Rebase 3a3ae71..812d20a onto 3a3ae71 (3 command(s))
#
# Commands:
# p, pick = use commit
# r, reword = use commit, but edit the commit message
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
# f, fixup = like "squash", but discard this commit's log message
# x, exec = run command (the rest of the line) using shell
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
#
# Note that empty commits are commented out
~
~
~
~
~
~
~/Documents/LMS_PRAC_EXAM/.git/rebase-merge/git-rebase-todo[+] [unix] (18:07 28/01/2026)

```

```

# This is a combination of 3 commits.
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> echo "L
line A" >> NOTES.md
git add
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> git com
mit -m "line c"
[feature/docs 812d20a] line c
1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> git reb
ase -i HEAD~3
[detached HEAD f6ba7bd] abc
Date: Wed Jan 28 17:57:27 2026 +0530
2 files changed, 1 insertion(+)
create mode 100644 NOTES.md
create mode 100644 abc.txt
Successfully rebased and updated refs/heads/feature/docs.
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM>

```

REBASE

Rebase moves or reapplies commits from one branch onto another, creating a linear history.

```
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> git checkout develop
Switched to branch 'develop'
Your branch is up-to-date with 'origin/develop'.
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> git pull --ff-only
Already up-to-date.
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> git checkout -b another
Switched to a new branch 'another'
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> echo 'change' >> second.txt
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> git add second.txt
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> git commit -m 'another change'
[another ce8cd7e] another change
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 second.txt
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> git push -u origin another
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 298 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'another' on GitHub by visiting:
remote:   https://github.com/parthshahsimform/LMS_PRAC_EXAM/pull/new/another
remote:
To git@github.com:parthshahsimform/LMS_PRAC_EXAM.git
 * [new branch]      another -> another
Branch another set up to track remote branch another from origin.
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> git fetch origin
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> git rebase origin/develop
Current branch another is up to date.
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> git fetch origin
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (1/1), done.
From github.com:parthshahsimform/LMS_PRAC_EXAM
 3a3ae71..704dd45  main      -> origin/main
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> git push -f origin another
Everything up-to-date
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> █
```

CHERRY-PICK

Cherry-pick lets you **apply a specific commit from one branch onto another**. Instead of merging or rebasing an entire branch, you can “pick” just one commit and replay it in your current branch.

```
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM>
>> git checkout develop && git pull
>> git checkout -b feature/TE-T84
>> echo "Minor README CHANGE" >> README.md
>> git add README.md
>> git commit -m "TE-T84: Update README"
>> git push -u origin feature/TE-T84
>>
>>
>> git checkout develop
>> git checkout -b feature/TE-T84
>>
>>
>> git log --oneline origin/feature/
>> git cherry-pick 7a7579d
>>
>>
>> git commit --amend -m "cherry-pick tset"
>>
>> echo "C1" >> misc.txt && git add misc.txt && git commit -m "TE-T84: C1"
>> echo "C2" >> misc.txt && git add misc.txt && git commit -m "TE-T84: C2"
>> echo "C3" >> misc.txt && git add misc.txt && git commit -m "TE-T84: C3"
>> git reset --soft HEAD~1
>> git revert HEAD
```

REVERT and DELETE

Revert - Creates a new commit that undoes the changes of a previous commit.

Delete - In Git, delete usually refers to removing a branch or a tag. It doesn't delete commits themselves (unless they're unreachable), but it removes the reference (pointer) to them.


```

PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> git push -u origin feature/TE-T84_another
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 329 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'feature/TE-T84_another' on GitHub by visiting:
remote:   https://github.com/parthshahsimform/LMS_PRAC_EXAM/pull/new/feature/TE-T84_another
remote:
To git@github.com:parthshahsimform/LMS_PRAC_EXAM.git
 * [new branch]      feature/TE-T84_another -> feature/TE-T84_another
Branch feature/TE-T84_another set up to track remote branch feature/TE-T84_another from origin.
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> git push origin --delete another
To git@github.com:parthshahsimform/LMS_PRAC_EXAM.git
 - [deleted]         another

```

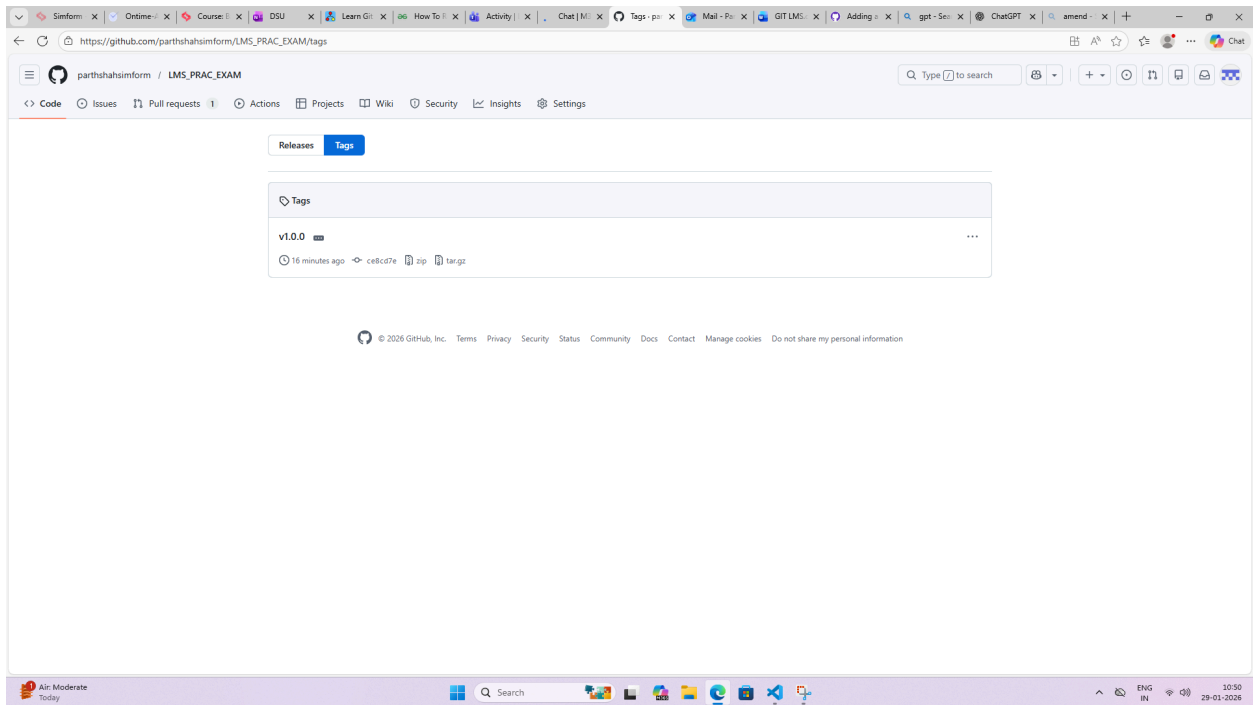
VERSION TAGS

- **Version tags** are tags used to label commits that represent official release versions of your project.
- They act like permanent bookmarks in history, so you can always return to or reference that exact state of the code.

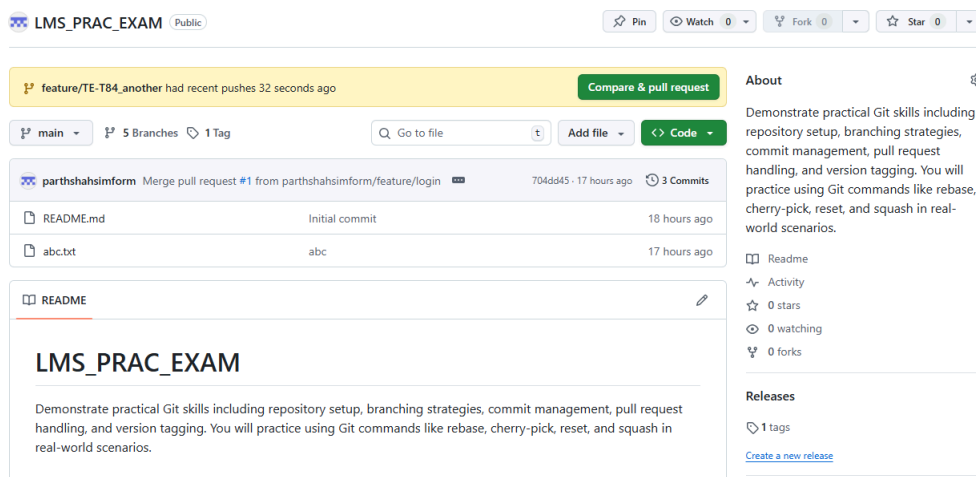
```

PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> git tag -a v1.0.0 -m "Release v1.0.0"
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> git push origin v1.0.0
Counting objects: 1, done.
Writing objects: 100% (1/1), 166 bytes | 0 bytes/s, done.
Total 1 (delta 0), reused 0 (delta 0)
remote: To git@github.com:parthshahsimform/LMS_PRAC_EXAM.git
 * [new tag]         v1.0.0 -> v1.0.0
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM>

```




PRACTICE SCENARIO



Revert "abc" #6


 parthshahsimform merged 3 commits into `develop` from `revert-1-feature/login` 1 hour ago


 Conversation 0  Commits 3  Checks 0  Files changed 2




parthshahsimform commented 1 hour ago

Owner ...

Reverts #1





parthshah02 and others added 3 commits 3 hours ago




merging to develop

e449d41




Merge pull request #5 from parthshahsimform/feature/TE-T84_fix

Verified 6d04883



Revert "abc"

Verified 49f4af5



parthshahsimform changed the base branch from `main` to `develop` 1 hour ago

COMMITTING MESSAGES

```
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> echo Hello > file1.txt
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> git add file1.txt
PS C:\Users\parthk.shah\Documents\LMS_PRAC_EXAM> git commit -m "added file1.txt"
[another 26f787c] added file1.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file1.txt
```