Day 11git stash is used to temporarily save changes in your working directory and index without committing them. This is useful when you want to switch branches or work on something else without committing the changes you're currently working on. Later, you can reapply those changes with git stash apply or git stash pop.

git rebase is a command used to integrate changes from one branch into another by moving or combining a sequence of commits to a new base commit. It's often used to maintain a clean and linear project history, particularly when you want to incorporate changes from a feature branch into the main branch (such as master or main) while keeping the commit history more organized.

Using git rebase, you can alter the commit history, rewrite commit messages, squash or split commits, or even resolve conflicts that might arise during the process of integrating changes from one branch to another. It's a powerful tool but requires care, especially when used on shared branches to avoid disrupting collaboration

Git cherry-pick is a command used to apply a specific commit from one branch onto another. It's handy when you want to include changes from a single commit rather than merging an entire branch. For instance, if you're working on a feature branch and want to include a bug fix or a specific feature commit from another branch, you can use git cherry-pick to do so.

This command applies the changes introduced by the specified commit onto your current branch as a new commit. If there are conflicts between the changes in the commit being cherry-picked and the current state of your branch, Git will pause to allow you to resolve these conflicts manually.

Remember, while cherry-picking is useful, it's essential to be cautious, especially when dealing with shared branches in a collaborative environment, as it can lead to duplicated commits and potential conflicts.

Task 01 :

Create a new branch and make some changes to it.

ubuntu@DevOps\*:~/Devops/Git$ git checkout master\*

Already on 'master'

ubuntu@DevOps\*:~/Devops/Git$\*

ubuntu@DevOps\*:~/Devops/Git$\*

ubuntu@DevOps\*:~/Devops/Git$ git checkout -b new1\*

Switched to a new branch 'new1'

ubuntu@DevOps\*:~/Devops/Git$ vi new\_file1.txt\*

ubuntu@DevOps\*:~/Devops/Git$\*

ubuntu@DevOps\*:~/Devops/Git$ git add .\*

Use git stash to save the changes without committing them.

ubuntu@DevOps\*:~/Devops/Git$ git stash --help\*

ubuntu@DevOps\*:~/Devops/Git$ git stash save\*

Saved working directory and index state WIP on new1: 4b5ada0 New file added in dev

Switch to a different branch, make some changes and commit them.

ubuntu@DevOps\*:~/Devops/Git$ git checkout dev\*

Switched to branch 'dev'

ubuntu@DevOps\*:~/Devops/Git$ ls\*

Day-02.txt file1.txt versio01.txt

ubuntu@DevOps\*:~/Devops/Git$ vi versio01.txt\*

ubuntu@DevOps\*:~/Devops/Git$\*

ubuntu@DevOps\*:~/Devops/Git$\*

ubuntu@DevOps\*:~/Devops/Git$ git add .\*

ubuntu@DevOps\*:~/Devops/Git$\*

ubuntu@DevOps\*:~/Devops/Git$ git commit -m "New line added in version01"\*

[dev 8738869] New line added in version01

1 file changed, 2 insertions(+)

ubuntu@DevOps\*:~/Devops/Git$\*

ubuntu@DevOps\*:~/Devops/Git$ git log --oneline\*

8738869 (HEAD -> dev) New line added in version01

4b5ada0 (new1, master\*) New file added in dev\*

cf77f6e (origin/dev) Reverted to the feature2 change

ed3d381 Added feature4 in development branch

1588013 Added feature3 in development branch

18f6481 Added feature2 in development branch

4ee156e Added new Feature

f0b2ad7 (origin/master) Adding branch, file and content

Use git stash pop to bring the changes back and apply them on top of the new commits.

ubuntu@DevOps\*:~/Devops/Git$ git stash list\*

stash@{0}: WIP on new1: 4b5ada0 New file added in dev

ubuntu@DevOps\*:~/Devops/Git$\*

ubuntu@DevOps\*:~/Devops/Git$ git stash show\*

Git/new\_file1.txt | 1 +

1 file changed, 1 insertion(+)

ubuntu@DevOps\*:~/Devops/Git$ git stash pop\*

On branch new1

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: new\_file1.txt

Dropped refs/stash@{0} (a54729621834d594f6393d7f5c1a4453136b5fdf)

ubuntu@DevOps\*:~/Devops/Git$ ls\*

Day-02.txt file1.txt new\_file1.txt versio01.txt

ubuntu@DevOps\*:~/Devops/Git$ git stash apply\*

No stash entries found.

ubuntu@DevOps\*:~/Devops/Git$ git commit -m "Stashed file new\_file1 is commited"\*

[new1 e8085e2] Stashed file new\_file1 is commited

1 file changed, 1 insertion(+)

create mode 100644 Git/new\_file1.txt

ubuntu@DevOps\*:~/Devops/Git$\*

ubuntu@DevOps\*:~/Devops/Git$\*

ubuntu@DevOps\*:~/Devops/Git$ git log --oneline\*

e8085e2 (HEAD -> new1) Stashed file new\_file1 is commited

4b5ada0 (master) New file added in dev

cf77f6e (origin/dev) Reverted to the feature2 change

ed3d381 Added feature4 in development branch

1588013 Added feature3 in development branch

18f6481 Added feature2 in development branch

4ee156e Added new Feature

f0b2ad7 (origin/master) Adding branch, file and content

TASK 02:

In version01.txt of development branch add below lines after “This is the bug fix in development branch” that you added in Day10 and reverted to this commit.

Line2>> After bug fixing, this is the new feature with minor alteration”

Commit this with message “ Added feature2.1 in development branch”

Line3>> This is the advancement of previous feature

Commit this with message “ Added feature2.2 in development branch”

Line4>> Feature 2 is completed and ready for release

Commit this with message “ Feature2 completed”

All these commits messages should be reflected in Production branch too which will come out from Master branch (Hint: try rebase).

Solution :

ubuntu@DevOps\*:~/Devops/Git$ git checkout dev\*

Switched to branch 'dev'

ubuntu@DevOps\*:~/Devops/Git$\*

ubuntu@DevOps\*:~/Devops/Git$ ls\*

Day-02.txt file1.txt versio01.txt

ubuntu@DevOps\*:~/Devops/Git$ vi versio01.txt\*

ubuntu@DevOps\*:~/Devops/Git$ git add .\*

ubuntu@DevOps\*:~/Devops/Git$ git commit -m "Added feature2.1 in development branch"\*

[dev 5c938ca] Added feature2.1 in development branch

1 file changed, 1 insertion(+)

ubuntu@DevOps\*:~/Devops/Git$ vi versio01.txt\*

ubuntu@DevOps\*:~/Devops/Git$\*

ubuntu@DevOps\*:~/Devops/Git$ git add .\*

ubuntu@DevOps\*:~/Devops/Git$ git commit -m "Added feature2.2 in development branch"\*

[dev 017af1c] Added feature2.2 in development branch

1 file changed, 1 insertion(+)

ubuntu@DevOps\*:~/Devops/Git$ vi versio01.txt\*

ubuntu@DevOps\*:~/Devops/Git$\*

ubuntu@DevOps\*:~/Devops/Git$ git add .\*

ubuntu@DevOps\*:~/Devops/Git$ git commit -m "Feature2 completed"\*

[dev 1115a26] Feature2 completed

1 file changed, 1 insertion(+)

ubuntu@DevOps\*:~/Devops/Git$\*

ubuntu@DevOps\*:~/Devops/Git$\*

ubuntu@DevOps\*:~/Devops/Git$ git log --oneline\*

1115a26 (HEAD -> dev) Feature2 completed

017af1c Added feature2.2 in development branch

5c938ca Added feature2.1 in development branch

8738869 New line added in version01

4b5ada0 (master) New file added in dev

cf77f6e (origin/dev) Reverted to the feature2 change

ed3d381 Added feature4 in development branch

1588013 Added feature3 in development branch

18f6481 Added feature2 in development branch

4ee156e Added new Feature

f0b2ad7 (origin/master) Adding branch, file and content

ubuntu@DevOps\*:~/Devops/Git$ git checkout master\*

Switched to branch 'master'

ubuntu@DevOps\*:~/Devops/Git$ ls\*

Day-02.txt file1.txt versio01.txt

ubuntu@DevOps\*:~/Devops/Git$ cat versio01.txt\*

<<<<<<< HEAD

This is the bug fix in development branch

This is a gadbad code

This feature will be gadbad everything from now

\=======

This is first feature of our application

\>>>>>>> parent of 18f6481 (Added feature2 in development branch)

ubuntu@DevOps\*:~/Devops/Git$ git rebase dev\*

Successfully rebased and updated refs/heads/master.

ubuntu@DevOps\*:~/Devops/Git$ cat versio01.txt\*

<<<<<<< HEAD

This is the bug fix in development branch

After bug fixing, this is the new feature with minor alteration

This is the advancement of previous feature

Feature 2 is completed and ready for release

This is a gadbad code

This feature will be gadbad everything from now

\=======

This is first feature of our application

\>>>>>>> parent of 18f6481 (Added feature2 in development branch)

New line added in text

ubuntu@DevOps\*:~/Devops/Git$\*

TASK-03

In Production branch Cherry pick Commit “Added feature2.2 in development branch” and added below lines in it:

Line to be added after Line3>> This is the advancement of previous feature

Line4>>Added few more changes to make it more optimized.

Commit: Optimized the feature

Solution :

ubuntu@DevOps\*:~/Devops/Git$ git log --oneline\*

1115a26 (HEAD -> master, dev\*) Feature2 completed\*

017af1c Added feature2.2 in development branch

5c938ca Added feature2.1 in development branch

8738869 New line added in version01

4b5ada0 New file added in dev

cf77f6e (origin/dev) Reverted to the feature2 change

ed3d381 Added feature4 in development branch

1588013 Added feature3 in development branch

18f6481 Added feature2 in development branch

4ee156e Added new Feature

f0b2ad7 (origin/master) Adding branch, file and content

ubuntu@DevOps\*:~/Devops/Git$ git cherry-pick 017af1c\*

ubuntu@DevOps\*:~/Devops/Git$ git log --oneline\*

1115a26 (HEAD -> master, dev\*) Feature2 completed\*

017af1c Added feature2.2 in development branch

5c938ca Added feature2.1 in development branch

8738869 New line added in version01

4b5ada0 New file added in dev

cf77f6e (origin/dev) Reverted to the feature2 change

ed3d381 Added feature4 in development branch

1588013 Added feature3 in development branch

18f6481 Added feature2 in development branch

4ee156e Added new Feature

f0b2ad7 (origin/master) Adding branch, file and content

ubuntu@DevOps\*:~/Devops/Git$ git merge dev\*

error: Merging is not possible because you have unmerged files.

hint: Fix them up in the work tree, and then use 'git add/rm <file>'

hint: as appropriate to mark resolution and make a commit.

fatal: Exiting because of an unresolved conflict.

ubuntu@DevOps\*:~/Devops/Git$ git merge --abort\*

fatal: There is no merge to abort (MERGE\_HEAD missing).

ubuntu@DevOps\*:~/Devops/Git$\*

ubuntu@DevOps\*:~/Devops/Git$ git status\*

On branch master

You are currently cherry-picking commit 017af1c.

(fix conflicts and run "git cherry-pick --continue")

(use "git cherry-pick --skip" to skip this patch)

(use "git cherry-pick --abort" to cancel the cherry-pick operation)

Unmerged paths:

(use "git add <file>..." to mark resolution)

both modified: versio01.txt

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

ubuntu@DevOps\*:~/Devops/Git$ vi versio01.txt\*

ubuntu@DevOps\*:~/Devops/Git$\*

ubuntu@DevOps\*:~/Devops/Git$ git add .\*

ubuntu@DevOps\*:~/Devops/Git$ git status\*

On branch master

You are currently cherry-picking commit 017af1c.

(all conflicts fixed: run "git cherry-pick --continue")

(use "git cherry-pick --skip" to skip this patch)

(use "git cherry-pick --abort" to cancel the cherry-pick operation)

Changes to be committed:

modified: versio01.txt

ubuntu@DevOps\*:~/Devops/Git$ git cherry-pick --continue\*

[master 38d8b25] Added feature2.2 in development branch

Date: Thu Nov 16 01:18:47 2023 +0000

1 file changed, 5 insertions(+), 2 deletions(-)

ubuntu@DevOps\*:~/Devops/Git$ cat versio01.txt\*

<<<<<<< HEAD

This is the advancement of previous feature

Added few more changes to make it more optimized.

This is the advancement of previous feature

<<<<<<< HEAD

Feature 2 is completed and ready for release

\=======

\>>>>>>> 017af1c (Added feature2.2 in development branch)

This is a gadbad code

This feature will be gadbad everything from now

\=======

This is first feature of our application

\>>>>>>> parent of 18f6481 (Added feature2 in development branch)

New line added in text