

ASSIGNMENT 2 DevOps

IT_BE_41_Himanshu Shukla

Perform the following operations on Git and **Gitlab**.

- **Create a Repository**
- **Perform Add, Commit, Push, Pull**
- **Create Branches**
- **Fork a project, Open and merge Pull request**
- **Merge and Rebase**
- **Squashing Commits**
- **Delete local and remote branches**
- **Undo commits**

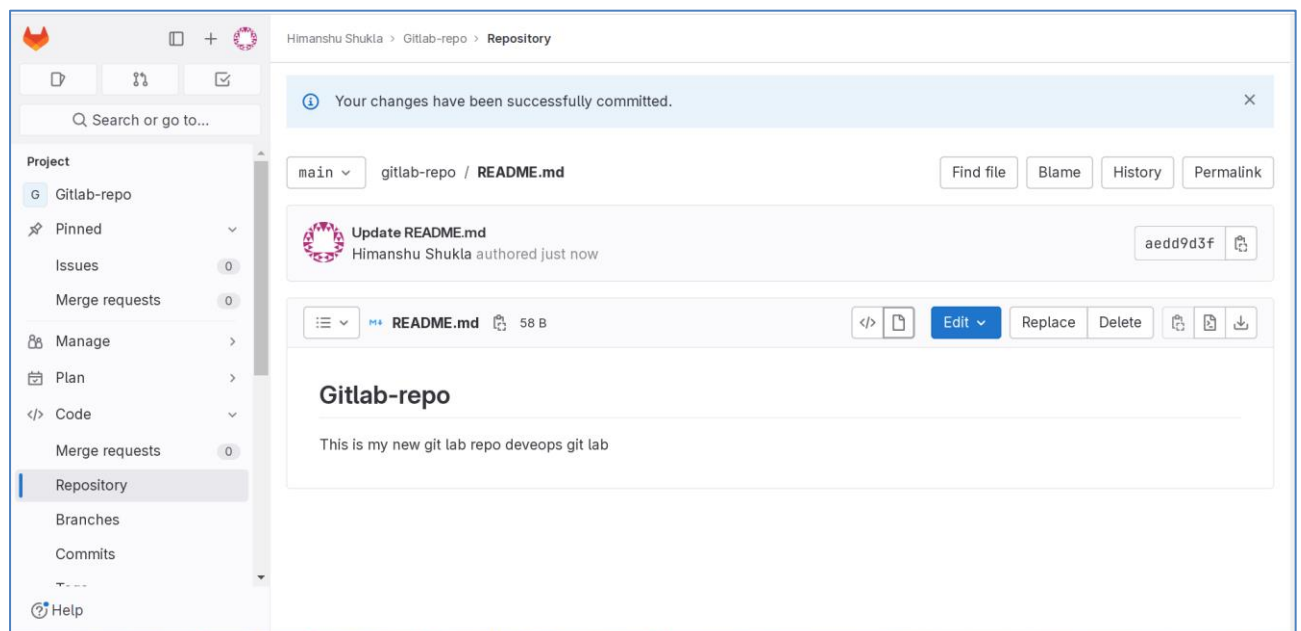
Prerequisite: git-scm, **git-lab account(gitlab.com)**, VS code

For performing git operation Git lab account is must

1)Creating a repository:

creating a git-lab repository on **gitlab** and accessing it from gitlab to local machine

a)create a git repository on gitlab

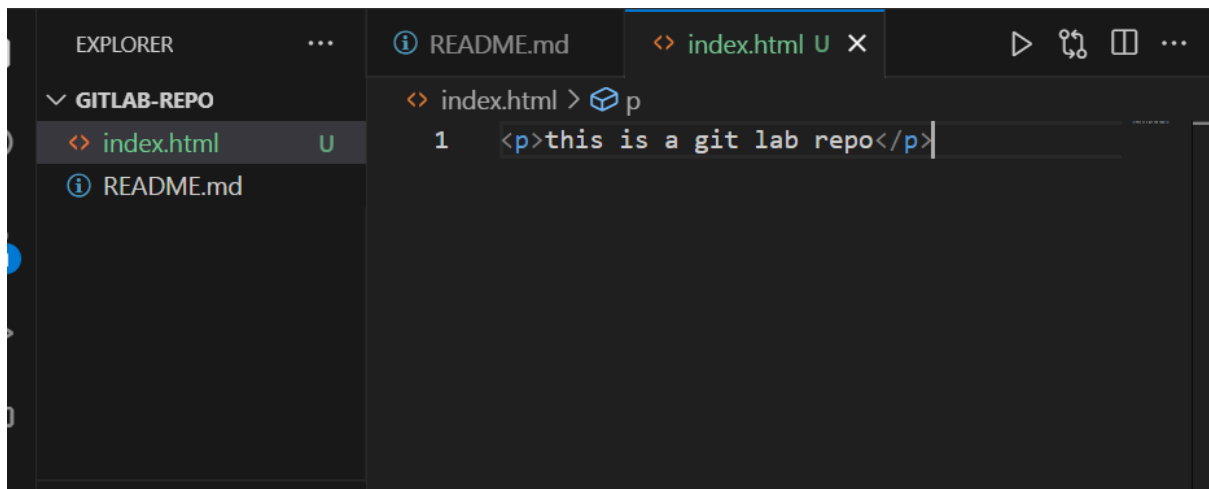


Now you can clone the git lab-repo with https password by clicking on gitlab clone using visual studio

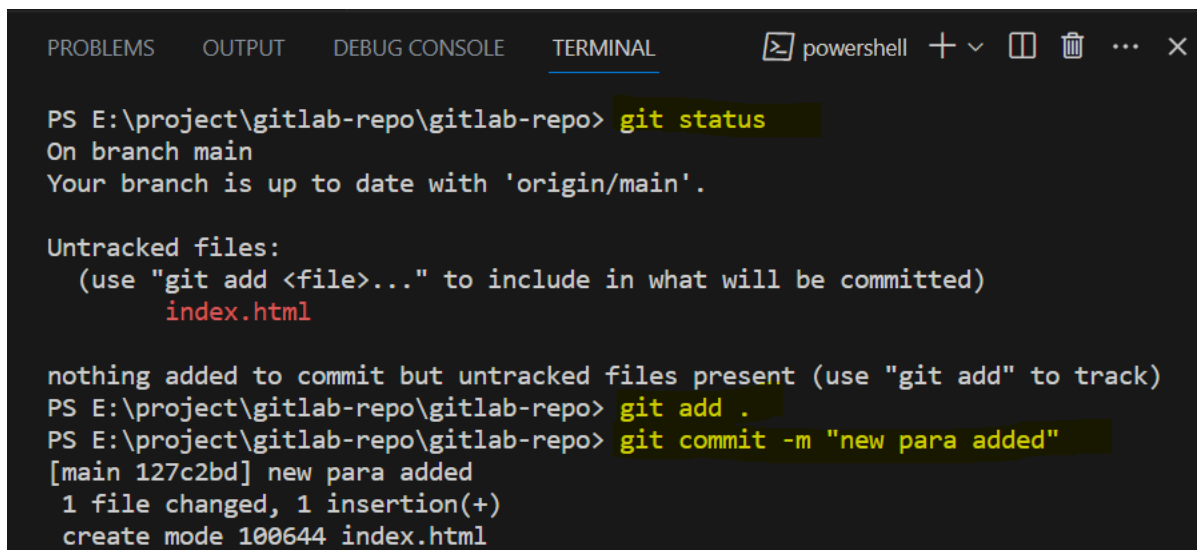


2)Performing ADD COMMIT PUSH & PULL

create a new file named index.html in same folder of git lab–repo



a)add, commit: This operation will help in saving the modified file on local machine




b)push: This operation will reflect the changes on gitlab repo(remote) since uptill commit it was saved only on local machine

```
PS E:\project\gitlab-repo\gitlab-repo> git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 617 bytes | 617.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://gitlab.com/himanshu.hsk02/gitlab-repo.git
aedd9d3..f6caa0f  main -> main
PS E:\project\gitlab-repo\gitlab-repo>
```

After this we can see that our newly created file `index.html` has been pushed on gitlab


Himanshu Shukla > Gitlab-repo > Repository



Merge branch 'main' of https://gitlab.com/himanshu.hsk02/gitlab-repo

Himanshu Shukla authored just now

f6caa0fb



main ▾


gitlab-repo /

+ ▾



History


Find file

Edit ▾

 ▾

Clone ▾

Name	Last commit	Last update
 README.md	Update README.md	16 minutes ago
 index.html	new para added	5 minutes ago

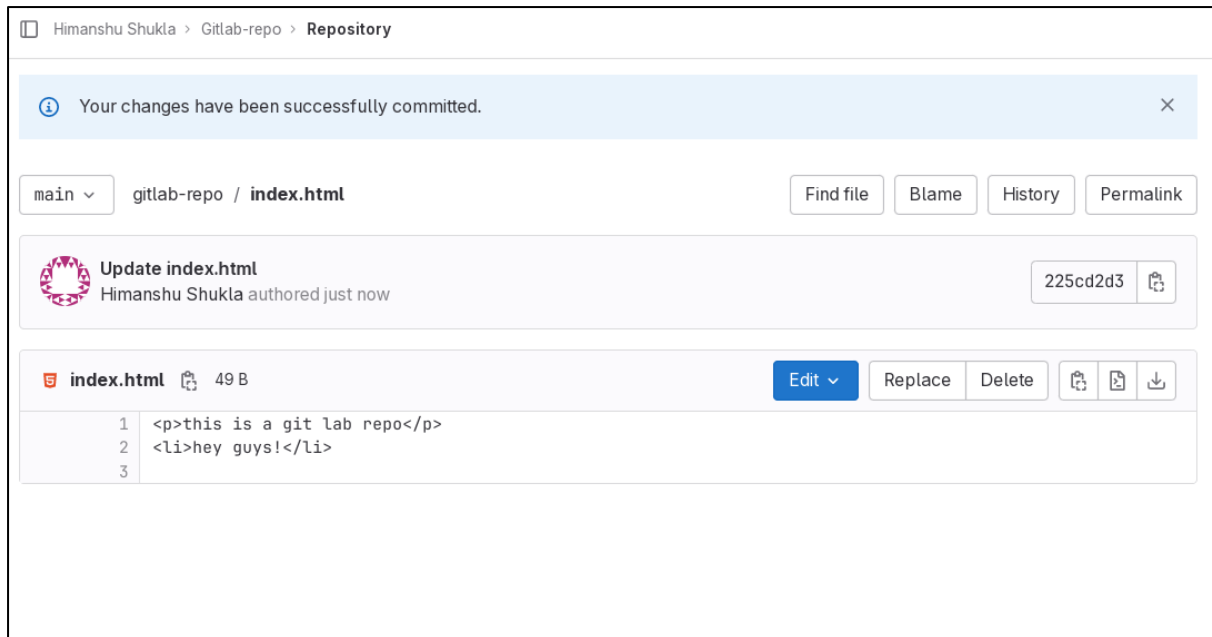
 README.md

Gitlab-repo

This is my new git lab repo deveops git lab

c)PULL: It is used to immediately update the content on local machine repo from remote gitlab repo

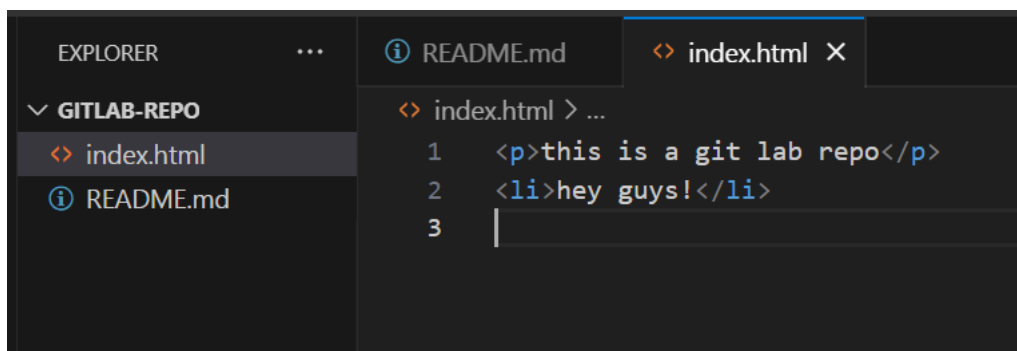
now suppose I have modified the index file by adding few more lines so let see it ,



Now again open terminal and write the command

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL powershell + - [ ] [X] ... X
PS E:\project\gitlab-repo\gitlab-repo> git pull origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 316 bytes | 11.00 KiB/s, done.
From https://gitlab.com/himanshu.hsk02/gitlab-repo
* branch main -> FETCH_HEAD
f6caa0f..225cd2d main -> origin/main
Updating f6caa0f..225cd2d
Fast-forward
 index.html | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
PS E:\project\gitlab-repo\gitlab-repo>
```

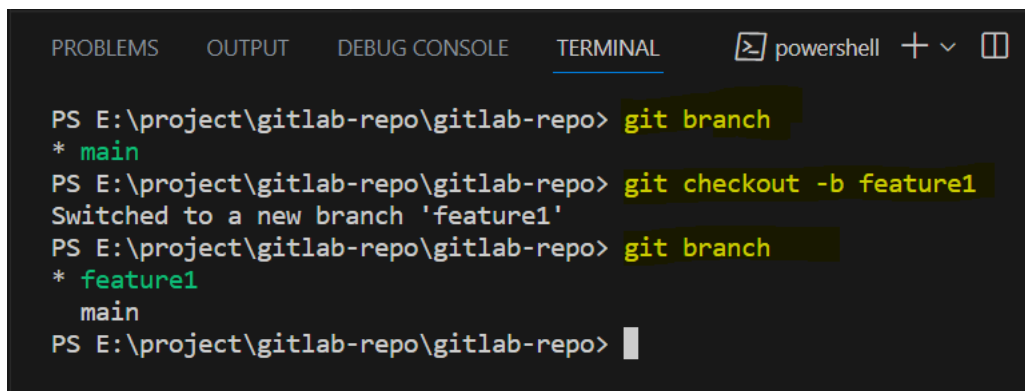
Now the update can be clearly visible in local machine(your desktop) repo



3) Create Branches

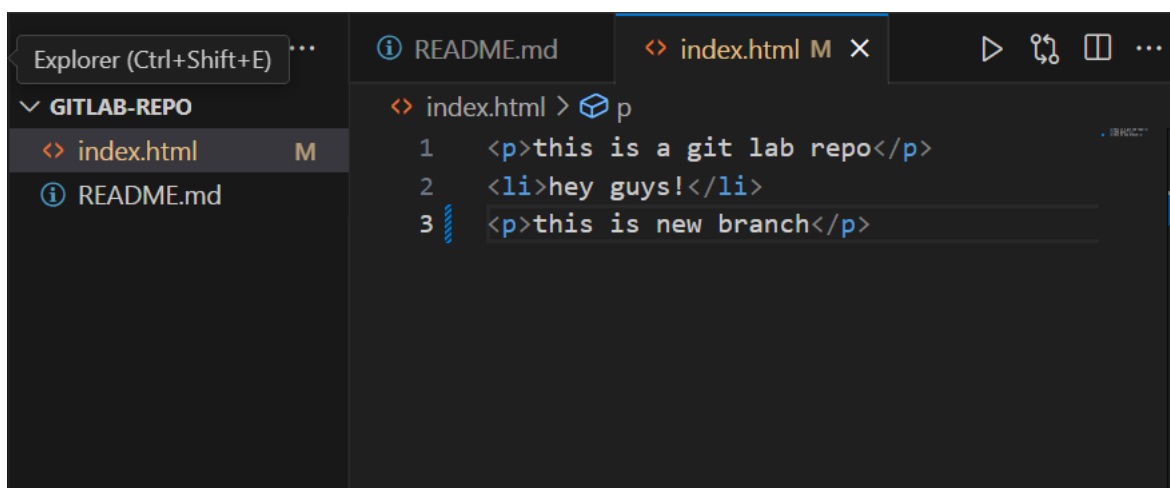
Before creating branch check that the branch name is “main” if not change it to “main” using command `git branch -M main`

a) create a new branch named feature1



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL powershell + -
PS E:\project\gitlab-repo\gitlab-repo> git branch
* main
PS E:\project\gitlab-repo\gitlab-repo> git checkout -b feature1
Switched to a new branch 'feature1'
PS E:\project\gitlab-repo\gitlab-repo> git branch
* feature1
  main
PS E:\project\gitlab-repo\gitlab-repo>
```

b) Modify branch(feature1) of main



The Explorer view on the left shows the file structure of the 'GITLAB-REPO' with 'index.html' and 'README.md'. The Editor view on the right shows the content of 'index.html' with three lines of HTML code:

```
<p>this is a git lab repo</p>
<li>hey guys!</li>
<p>this is new branch</p>
```

After this, push it to gitlab repo using git commands where you can see the two branches now

```

PS E:\project\gitlab-repo\gitlab-repo> git add .
PS E:\project\gitlab-repo\gitlab-repo> git commit -m "new branch"
[feature1 e7e33c8] new branch
1 file changed, 1 insertion(+)
PS E:\project\gitlab-repo\gitlab-repo> git push origin feature1
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 346 bytes | 346.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: To create a merge request for feature1, visit:
remote:   https://gitlab.com/himanshu.hsk02/gitlab-repo/-/merge_requests/new?merge_request%5Bsource_branch%5D=feature1
remote:
To https://gitlab.com/himanshu.hsk02/gitlab-repo.git
* [new branch]      feature1 -> feature1
PS E:\project\gitlab-repo\gitlab-repo>

```

Now we can see that the two branches has been formed

The screenshot shows the GitLab web interface. At the top, there are tabs for 'Overview', 'Active', 'Stale', and 'All'. Below these is a search bar labeled 'Filter by branch name' and buttons for 'View branch rules' and 'New branch'. A blue notification banner states: 'See all branch-related settings together with branch rules. You can now find an overview of settings for protected branches, merge request approvals, status checks, and security approvals conveniently in one spot. View branch rules Dismiss'. Below the notification, the 'Active branches' section lists two branches: 'feature1' (commit e7e33c8a, new branch 2 minutes ago) and 'main' (commit 225cd2d3, default, protected, update index.html 8 minutes ago). Each branch entry has icons for cloning, creating a new branch, and downloading the code.

4) Fork a project, Open and merge Pull request

a) forking a random project of joshnh “Git-Commands”

This is mehmetkozan account, now we will fork it to our account

integration_test

Free

Project ID: 4924443

Star

0

Forks

2

48,569 Commits

2 Branches

0 Tags

399.1 MiB Project Storage

Not for end user usage, for internal test of my package.

ci test output added

Mehmet Kozan authored just now

7f862418

master

_integration_test /

+

History

Find file

Edit

Clone

README

MIT License

CHANGELOG

CONTRIBUTING

Name	Last commit	Last update
.circleci	circle-ci updated.	5 years ago
.vscode	initial commit.	5 years ago
lib	travis-ci updated.	5 years ago
outputs	ci test output added	just now
test	ci updated.	5 years ago
tests	readme added.	5 years ago

Now after clicking on create fork this whole branch will be copied to our account

autokent

>

_integration_test

>

Fork project

Fork project

A fork is a copy of a project. Forking a repository allows you to make changes without affecting the original project.

Project name

_integration_test

Must start with a lowercase or uppercase letter, digit, emoji, or underscore. Can also contain dots, pluses, dashes, or spaces.

Project URL

https://gitlab.com/

Select a namespace

Project slug

_integration_test

Want to organize several dependent projects under the same namespace? [Create a group](#)

Project description (optional)

Not for end user usage, for internal test of my package.

Visibility level

☐
Private
Project access must be granted explicitly to each user. If this project is part of a group, access will be granted to members of the group.

☐
Internal
The project can be accessed by any logged in user.

☒
Public
The project can be accessed without any authentication.

Fork project

Cancel

b) Open and merge pull request

now we will merge our modified feature1 branch with main branch using pull request and then will pull it to our local machine

Now go to gitlab account click on [create merge request](#) on top

Himanshu Shukla > Gitlab-repo > Merge requests > New

New merge request

From

feature1

 into

main

[Change branches](#)

Title (required)

new branch







☐ Mark as draft

Drafts cannot be merged until marked ready.

Description

Preview **B** *I* U ~~ABC~~ `</>` [Link](#)

-
-
-

Describe the goal of the changes and what reviewers should be aware of.

Switch to rich text editing

IM3

Add [description templates](#) to help your contributors to communicate effectively!

Assignee

Unassigned

▼

[Assign to me](#)

Reviewer

Unassigned

▼

Milestone

Select milestone

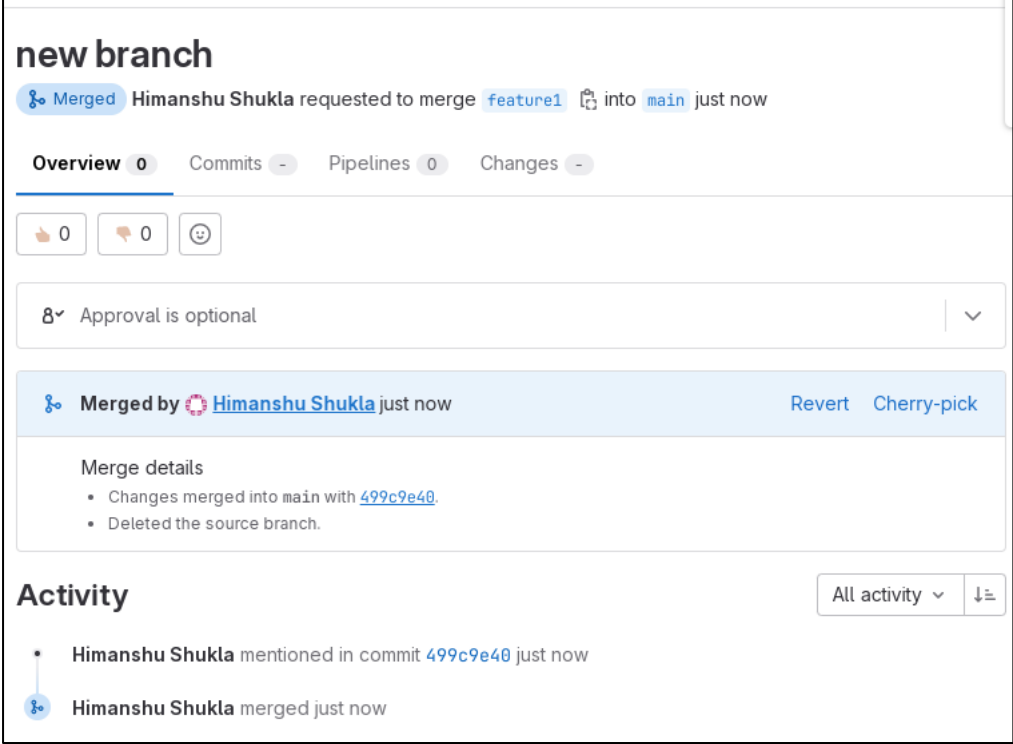
▼

Labels

Select label

▼

Now the branch has been successfully merged



Pull it to local machine using command `git pull origin main`

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL powershell + ▢ 🗑
PS E:\project\gitlab-repo\gitlab-repo> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS E:\project\gitlab-repo\gitlab-repo> git pull origin main
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (1/1), 252 bytes | 14.00 KiB/s, done.
From https://gitlab.com/himanshu.hsk02/gitlab-repo
* branch          main          -> FETCH_HEAD
   225cd2d..499c9e4 main        -> origin/main
Updating 225cd2d..499c9e4
Fast-forward
 index.html | 1 +
 1 file changed, 1 insertion(+)

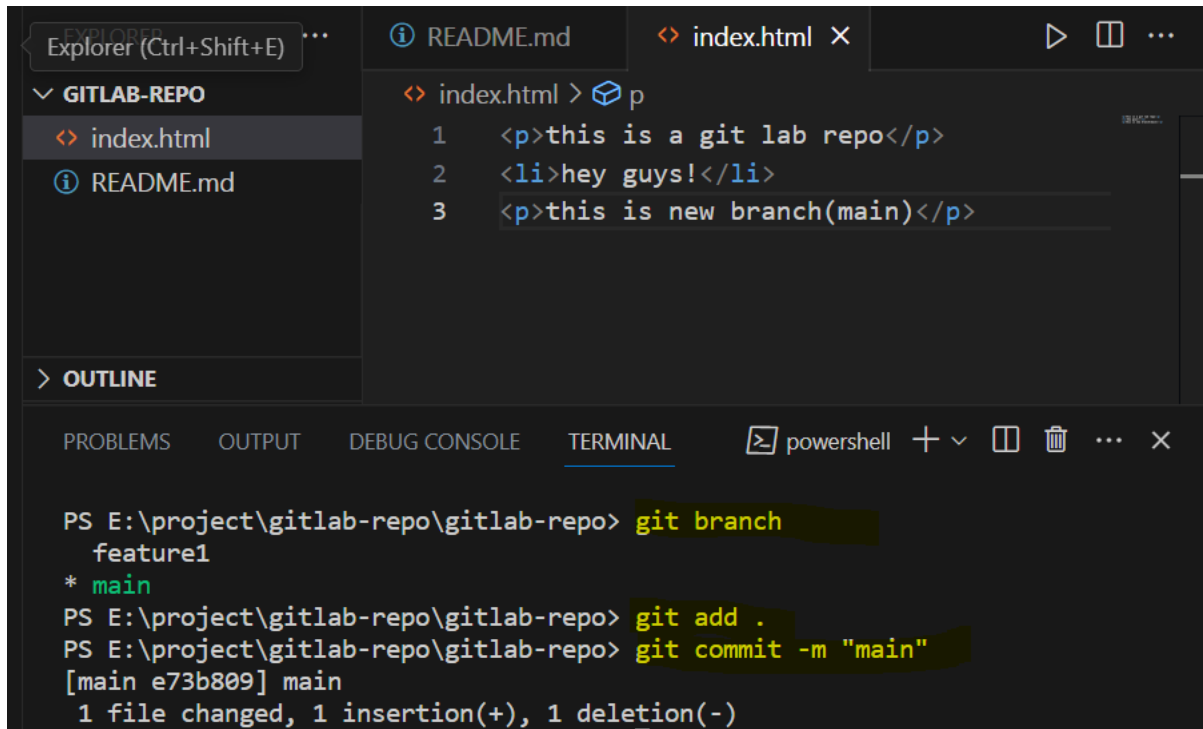
```

Now after this , the merging can be seen in local machine repo too.

5) Merge and Rebase

Now some time there will be a conflict merge so let see how to merge the conflict content

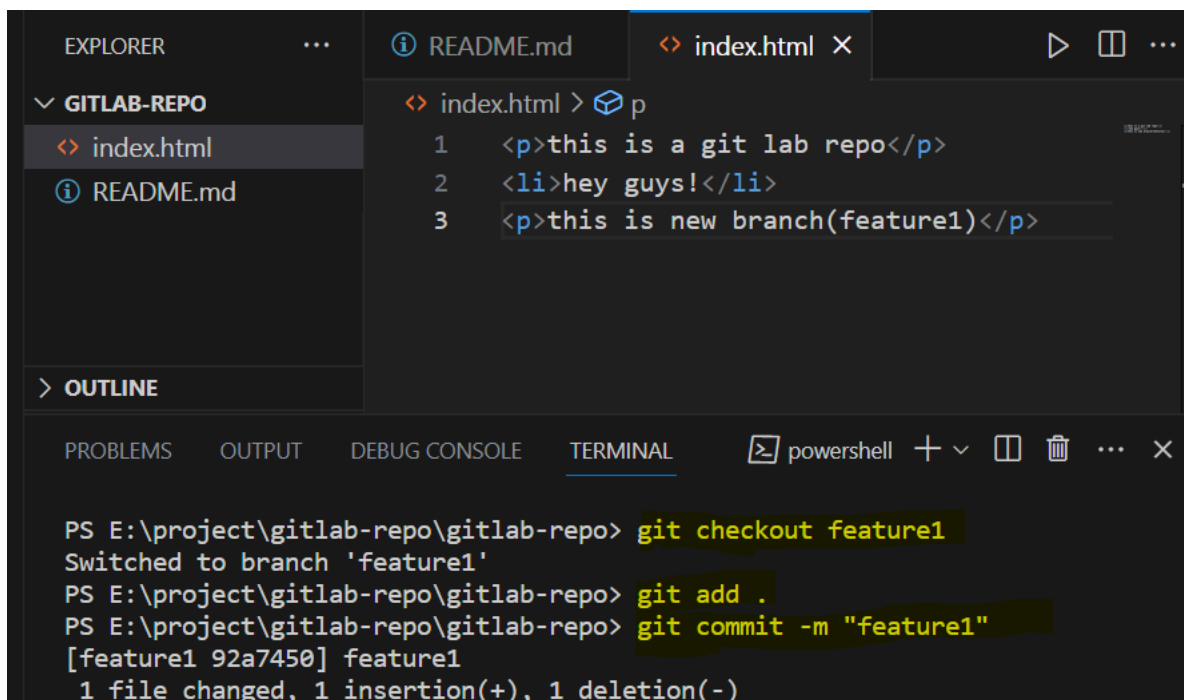
Modify the same line on both branches main and feature1



```
EXPLORER (Ctrl+Shift+E) ...
  GITLAB-REPO
    index.html
    README.md
  OUTLINE

index.html > p
1 <p>this is a git lab repo</p>
2 <li>hey guys!</li>
3 <p>this is new branch(main)</p>

TERMINAL
PS E:\project\gitlab-repo\gitlab-repo> git branch
feature1
* main
PS E:\project\gitlab-repo\gitlab-repo> git add .
PS E:\project\gitlab-repo\gitlab-repo> git commit -m "main"
[main e73b809] main
1 file changed, 1 insertion(+), 1 deletion(-)
```



```
EXPLORER ...
  GITLAB-REPO
    index.html
    README.md
  OUTLINE

index.html > p
1 <p>this is a git lab repo</p>
2 <li>hey guys!</li>
3 <p>this is new branch(feature1)</p>

TERMINAL
PS E:\project\gitlab-repo\gitlab-repo> git checkout feature1
Switched to branch 'feature1'
PS E:\project\gitlab-repo\gitlab-repo> git add .
PS E:\project\gitlab-repo\gitlab-repo> git commit -m "feature1"
[feature1 92a7450] feature1
1 file changed, 1 insertion(+), 1 deletion(-)
```

Now will merge this conflict and will vanish one of line of branches

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL powershell + - [ ] [X] ... X

PS E:\project\gitlab-repo\gitlab-repo> git branch
* feature1
  main
PS E:\project\gitlab-repo\gitlab-repo> git diff main
diff --git a/index.html b/index.html
index d9bd73d..eb280e0 100644
--- a/index.html
+++ b/index.html
@@ -1,3 +1,3 @@
  <p>this is a git lab repo</p>
  <li>hey guys!</li>
- <p>this is new branch(main)</p>
\ No newline at end of file
+ <p>this is new branch(feature1)</p>
\ No newline at end of file
PS E:\project\gitlab-repo\gitlab-repo> git merge main
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```

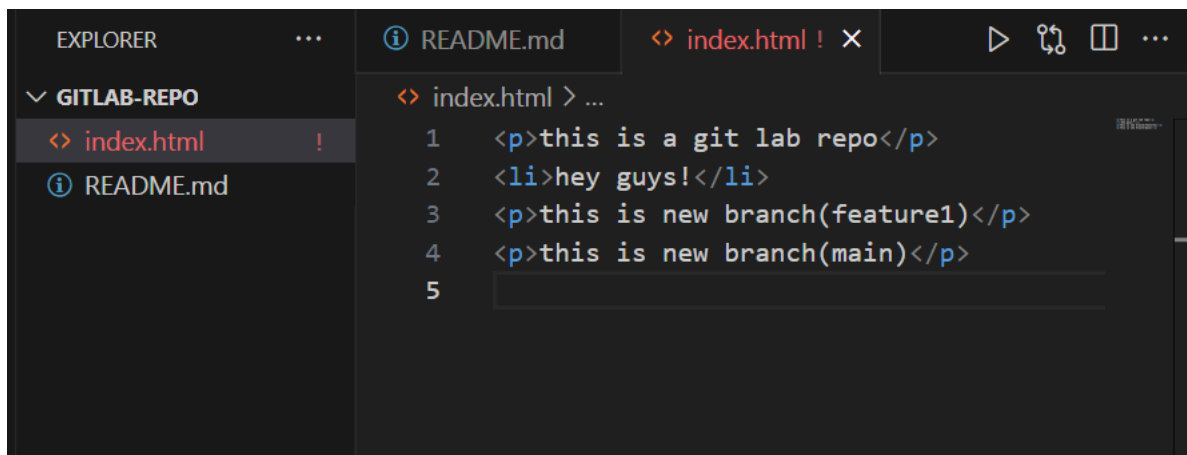
Now just delete the unwanted thing and merge it after this push to main gitlab page

```
Explorer (Ctrl+Shift+E) ... README.md index.html ! X [ ] [X] ...

v GITLAB-REPO
  < index.html !
  i README.md

< index.html > ?
1 <p>this is a git lab repo</p>
2 <li>hey guys!</li>
  Accept Current Change | Accept Incoming Change | Accep
3 <<<<<<< HEAD (Current Change)
4 <p>this is new branch(feature1)</p>
5 =====
6 <p>this is new branch(main)</p>
7 >>>>>> main (Incoming Change)
8
```

After this it can be seen like this



now write the commad

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL powershell + v [] [] ... X
PS E:\project\gitlab-repo\gitlab-repo> git add .
PS E:\project\gitlab-repo\gitlab-repo> git commit -m "add both feature"
[feature1 66d6a09] add both feature
PS E:\project\gitlab-repo\gitlab-repo> git push
fatal: The current branch feature1 has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin feature1

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.

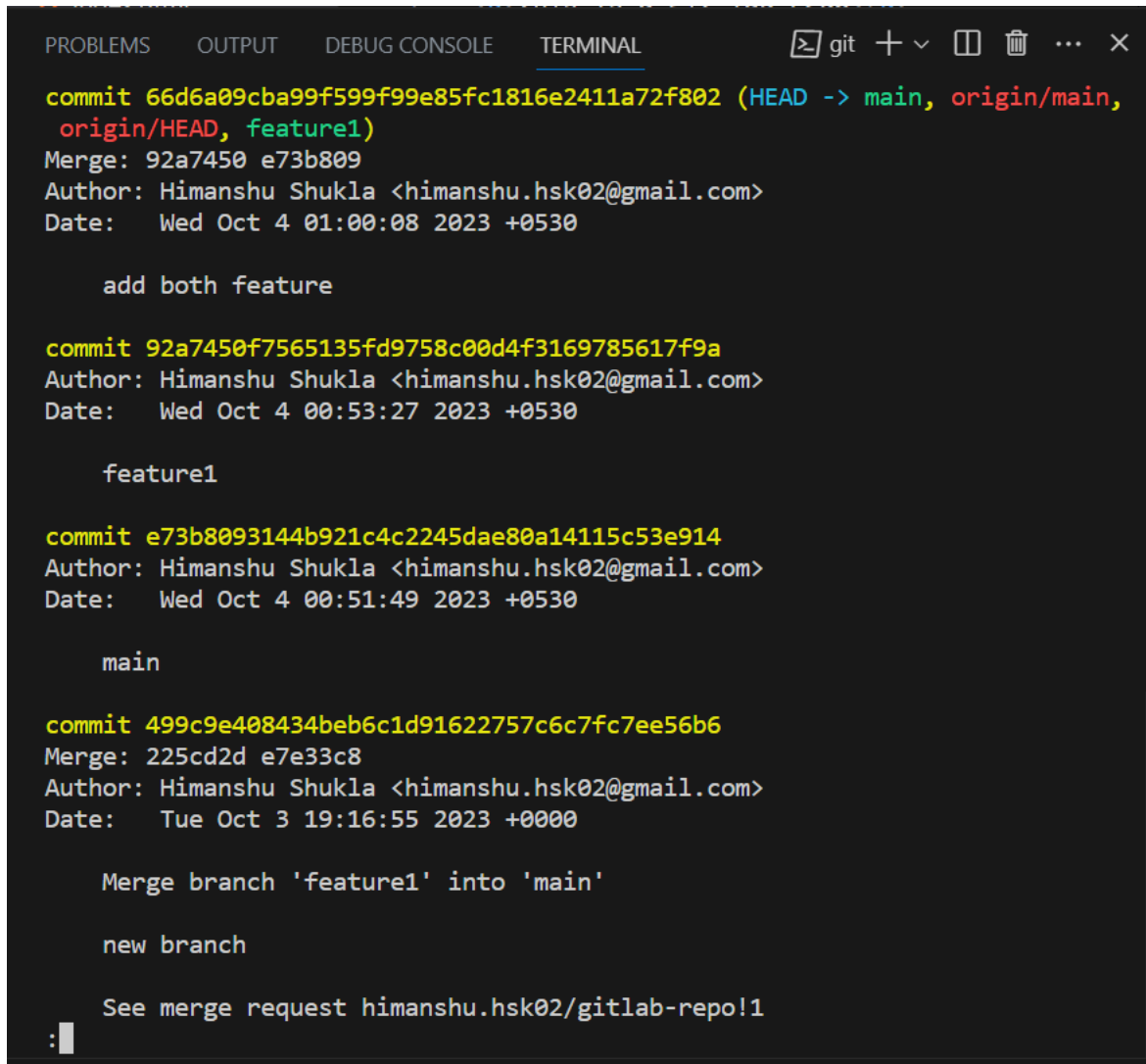
PS E:\project\gitlab-repo\gitlab-repo> git branch
* feature1
  main
PS E:\project\gitlab-repo\gitlab-repo> git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)
PS E:\project\gitlab-repo\gitlab-repo> git merge feature1
Updating e73b809..66d6a09
Fast-forward
 index.html | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
PS E:\project\gitlab-repo\gitlab-repo> git push
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 4 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 902 bytes | 451.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0
```

Now after refreshing it will be reflected on main gitlab repo

6) Squashing commits

Squash will retain the commits

Check the commits using [git log](#)



```
commit 66d6a09cba99f599f99e85fc1816e2411a72f802 (HEAD -> main, origin/main, origin/HEAD, feature1)
Merge: 92a7450 e73b809
Author: Himanshu Shukla <himanshu.hsk02@gmail.com>
Date: Wed Oct 4 01:00:08 2023 +0530

    add both feature

commit 92a7450f7565135fd9758c00d4f3169785617f9a
Author: Himanshu Shukla <himanshu.hsk02@gmail.com>
Date: Wed Oct 4 00:53:27 2023 +0530

    feature1

commit e73b8093144b921c4c2245dae80a14115c53e914
Author: Himanshu Shukla <himanshu.hsk02@gmail.com>
Date: Wed Oct 4 00:51:49 2023 +0530

    main

commit 499c9e408434beb6c1d91622757c6c7fc7ee56b6
Merge: 225cd2d e7e33c8
Author: Himanshu Shukla <himanshu.hsk02@gmail.com>
Date: Tue Oct 3 19:16:55 2023 +0000

    Merge branch 'feature1' into 'main'

    new branch

    See merge request himanshu.hsk02/gitlab-repo!1
```

Enter q to quite the log

After write the command `git rebase -i HEAD~3` and hit insert key change the pick to s

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
pick 225cd2d Update index.html
pick 356cf3c # This is a combination of 2 commits. # This is the 1st commit
message:
pick 5206da9 add both

# Rebase f6caa0f..5206da9 onto f6caa0f (3 commands)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup [-C | -c] <commit> = like "squash" but keep only the previous
#      commit's log message, unless -C is used, in which case
#      keep only this commit's message; -c is same as -C but
#      opens the editor
# x, exec <command> = run command (the rest of the line) using shell
# b, break = stop here (continue rebase later with 'git rebase --continue')
# d, drop <commit> = remove commit
# l, label <label> = label current HEAD with a name
# t, reset <label> = reset HEAD to a label
# m, merge [-C <commit> | -c <commit>] <label> [# <oneline>]
#      create a merge commit using the original merge commit's
#      message (or the oneline, if no original merge commit was
#      specified); use -c <commit> to reword the commit message
# u, update-ref <ref> = track a placeholder for the <ref> to be updated
#      to this position in the new commits. The <ref> is
#      updated at the end of the rebase
.git/rebase-merge/git-rebase-todo [unix] (01:23 04/10/2023) 1,1 Top
```

After this you can see that the two commits have been merged

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
commit 5206da98e863aab379b5f5648c2139b914d3c19d (HEAD)
Author: Himanshu Shukla <himanshu.hsk02@gmail.com>
Date: Wed Oct 4 01:22:01 2023 +0530

    add both

commit 356cf3cba3e575507a2c561e896ffc19b5b08b2e
Author: Himanshu Shukla <himanshu.hsk02@gmail.com>
Date: Wed Oct 4 00:32:26 2023 +0530

    # This is a combination of 2 commits.
    # This is the 1st commit message:

    new branch

    # This is the commit message #2:

    feature1

commit 225cd2d359138590bd609920847b27e357e497fb
Author: Himanshu Shukla <himanshu.hsk02@gmail.com>
Date: Tue Oct 3 18:56:31 2023 +0000

    Update index.html

commit f6caa0fb1a01299cd621d196a1d67e32480c664f
Merge: 127c2bd aedd9d3
Author: Himanshu Shukla <himanshu.hsk02@gmail.com>
Date: Wed Oct 4 00:19:40 2023 +0530
:
```

7) Delete branch

Create a new branch and delete it

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL powershell + v [ ] [ ] ... X
PS E:\project\gitlab-repo\gitlab-repo> git checkout -b feature
Switched to a new branch 'feature'
PS E:\project\gitlab-repo\gitlab-repo> git branch
* feature
  feature1
  main
PS E:\project\gitlab-repo\gitlab-repo> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS E:\project\gitlab-repo\gitlab-repo> git branch -d feature
Deleted branch feature (was 66d6a09).
PS E:\project\gitlab-repo\gitlab-repo> git branch
* main
feature1
PS E:\project\gitlab-repo\gitlab-repo> 
```

8) undo commits

suppose I have deleted the last line and perform the commit operation but now I have to undo it . let us see ,

```
EXPLORER ... [i] README.md <> index.html M X [ ] [ ] [ ] ...
v GITLAB-REPO
<> index.html M
[i] README.md

<> index.html > ...
1 <p>this is a git lab repo</p>
2 <li>hey guys!</li>
3 <p>this is new branch(feature1)</p>
4 |
5 
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  powershell + - [ ] [X] ... X

PS E:\project\gitlab-repo\gitlab-repo> git add .
PS E:\project\gitlab-repo\gitlab-repo> git commit -m "new edited"
[main b1dc50d] new edited
 1 file changed, 1 insertion(+), 1 deletion(-)
PS E:\project\gitlab-repo\gitlab-repo> git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
PS E:\project\gitlab-repo\gitlab-repo> git reset HEAD~1
Unstaged changes after reset:
M       index.html
PS E:\project\gitlab-repo\gitlab-repo> git status
On branch main
Your branch is up to date with 'origin/main'.

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:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
PS E:\project\gitlab-repo\gitlab-repo> git restore index.html
PS E:\project\gitlab-repo\gitlab-repo>
```

After restoring,... it get undo,below we can see

```
EXPLORER  ...  README.md  index.html X  [ ] ...

v GITLAB-REPO
  <> index.html
  ⓘ README.md

  <> index.html > ...
  1  <p>this is a git lab repo</p>
  2  <li>hey guys!</li>
  3  <p>this is new branch(feature1)</p>
  4  <p>this is new branch(main)</p>
  5
```

(git reset <filename> can be used for undo after the add operation has been performed)