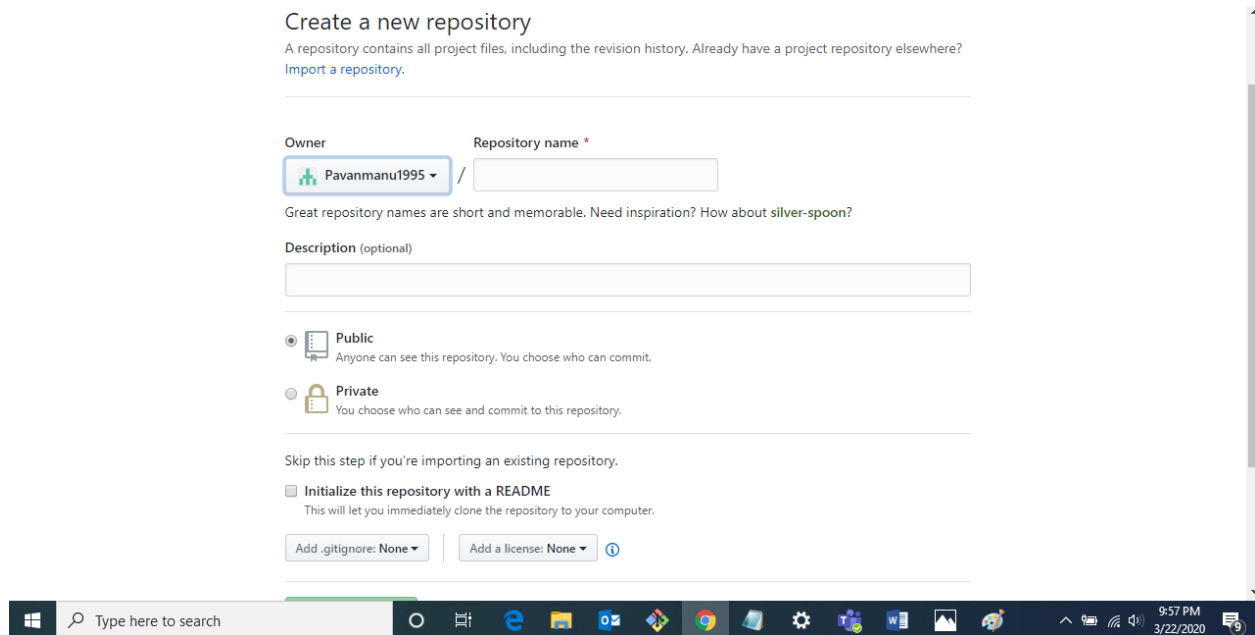


# Git Assignment

GitHub is a cloud-based service that helps developers store and manage their code, share codes as well as track and control changes to their code.

- Create a remote repository in the github.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner: Pavanmanu1995 / Repository name \*

Great repository names are short and memorable. Need inspiration? How about [silver-spoon](#)?

Description (optional)

☒ Public  
Anyone can see this repository. You choose who can commit.

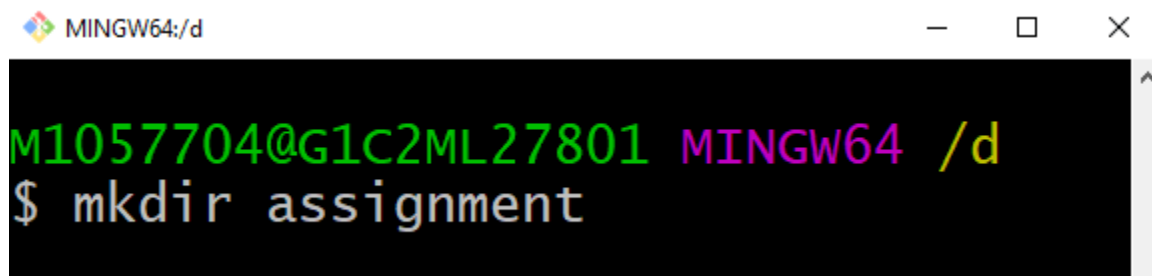
☐ Private  
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ Initialize this repository with a README  
This will let you immediately clone the repository to your computer.

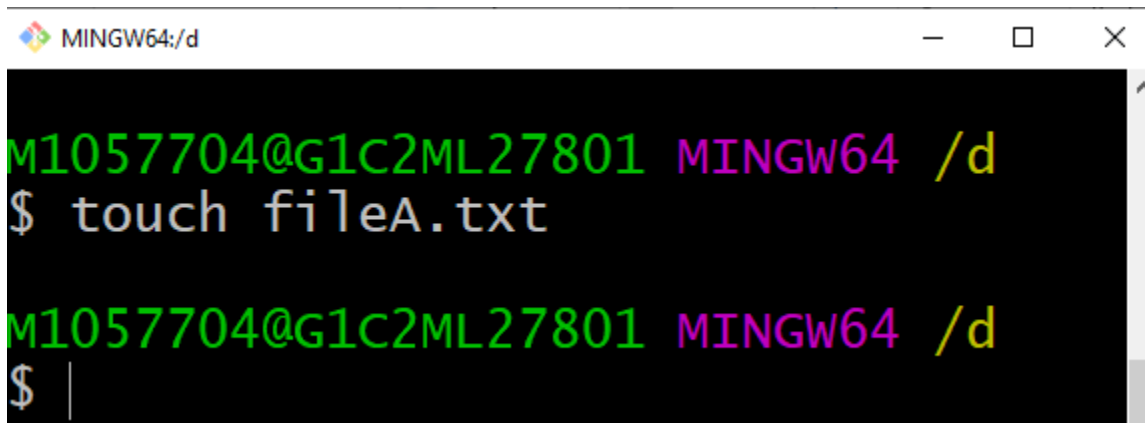
Add .gitignore: None | Add a license: None ⓘ

- In git bash create a directory using **mkdir** command.



```
M1057704@G1C2ML27801 MINGW64 /d
$ mkdir assignment
```

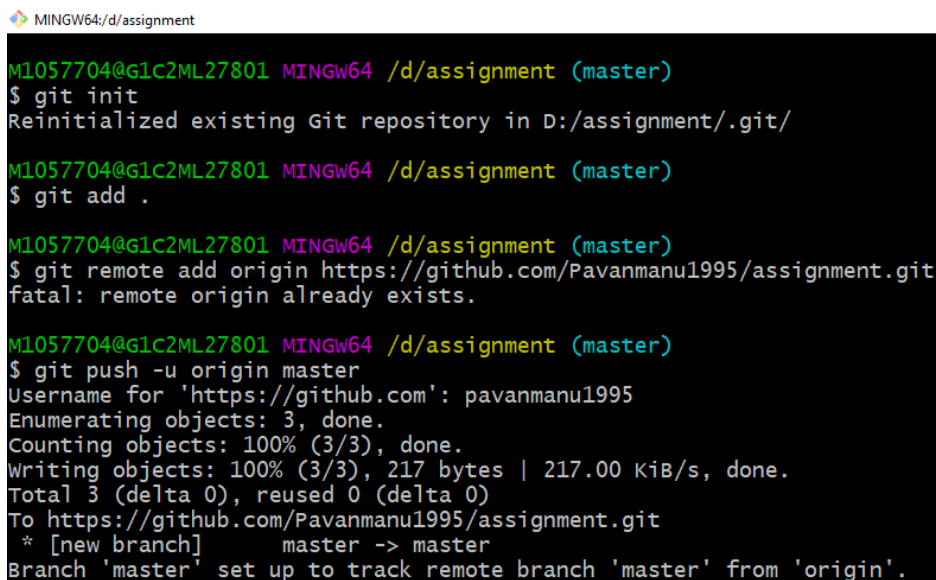
- Using git bash create a file inside the directory using **touch** command.



```
MINGW64:/d
M1057704@G1C2ML27801 MINGW64 /d
$ touch fileA.txt
M1057704@G1C2ML27801 MINGW64 /d
$ |
```

## Commit A:

- By default the master branch will be created which is the main branch.
- **Init**: is used to initialize the local repository.
- **Add .** : it is used to add the file to the staging.
- **Push** : it is used to send the files from the local repository to the remote repository.



```
MINGW64:/d/assignment
M1057704@G1C2ML27801 MINGW64 /d/assignment (master)
$ git init
Reinitialized existing Git repository in D:/assignment/.git/
M1057704@G1C2ML27801 MINGW64 /d/assignment (master)
$ git add .
M1057704@G1C2ML27801 MINGW64 /d/assignment (master)
$ git remote add origin https://github.com/Pavanmanu1995/assignment.git
fatal: remote origin already exists.
M1057704@G1C2ML27801 MINGW64 /d/assignment (master)
$ git push -u origin master
Username for 'https://github.com': pavanmanu1995
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 217 bytes | 217.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/Pavanmanu1995/assignment.git
* [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

- **Clone** :it is used to create a copy of files that are present in the remote repository.

```
M1057704@G1C2ML27801 MINGW64 /d/assignment (master)
$ git clone https://github.com/Pavanmanu1995/assignment.git
Cloning into 'assignment'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
```

## Commit B:

- **Branch** : it is used to create a branch from the master.
- Develop is a branch name.
- **Checkout**: it points to the branch to which the branch name is given.
- **Ls** : it is used to list the directories and the files.
- **Echo** : it is used to create a file and write the content to the file.
- **Cat** : it is used to create a file and displays the content in the file.
- The develop branch is created.
- An empty string is written in the fileA.txt in the develop branch.
- Using **add** . the fileA.txt is added to the staging area.

```
MINGW64/d/assignment

M1057704@G1C2ML27801 MINGW64 /d/assignment (master)
$ git branch develop

M1057704@G1C2ML27801 MINGW64 /d/assignment (master)
$ git checkout develop
Switched to branch 'develop'

M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ ls
assignment/  fileA.txt

M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ echo ""(empty string) >> fileA.txt
bash: syntax error near unexpected token `('

M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ echo "" >> fileA.txt

M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ cat fileA.txt

M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ git add .
warning: LF will be replaced by CRLF in fileA.txt.
The file will have its original line endings in your working directory.
warning: adding embedded git repository: assignment
hint: You've added another git repository inside your current repository.
hint: Clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:   git submodule add <url> assignment
hint:
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
```

No description, website, or topics provided.

Edit

Manage topics

2 commits

3 branches

0 packages

0 releases

0 contributors

Your recently pushed branches:

develop (31 minutes ago)

Compare & pull request

feature1 (1 minute ago)

Compare & pull request

Branch: develop

New pull request

Create new file

Upload files

Find file

Clone or download

This branch is 1 commit ahead of master.

Pull request

Compare

M1057704 and M1057704 add fileA.txt

Latest commit 8c067db 35 minutes ago

assignment

add fileA.txt

35 minutes ago

fileA.txt

add fileA.txt

35 minutes ago

## Commit C:

- Feature 1 wip is written into the fileA.txt using **echo** command.
- Using **cat** content is displayed from the fileA.txt
- Feature1 branch is created using **branch** command.
- Using **checkout** command the branch is pointed.
- Using **add .** command the fileA.txt is added to the staging.
- **Commit:** is used to send the fileA.txt from staging area to the local repository.
- **-m :** is the message.
- The content in the fileA.txt is pushed to the remote repository from the local repository.

```
MINGW64:/d/assignment
M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ echo feature 1 wip >> fileA.txt
M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ cat fileA.txt
feature 1 wip
M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ git branch feature1
M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ git checkout feature1
Switched to branch 'feature1'
M      fileA.txt
M1057704@G1C2ML27801 MINGW64 /d/assignment (feature1)
$ git add .
warning: LF will be replaced by CRLF in fileA.txt.
The file will have its original line endings in your working directory.
M1057704@G1C2ML27801 MINGW64 /d/assignment (feature1)
$ git add .
M1057704@G1C2ML27801 MINGW64 /d/assignment (feature1)
$ git commit -m "feature1 wip"
[feature1 a850b38] feature1 wip
1 file changed, 1 insertion(+)
M1057704@G1C2ML27801 MINGW64 /d/assignment (feature1)
$ git push -u origin feature1
Username for 'https://github.com': pavanmanu1995
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 296 bytes | 296.00 KiB/s, done.
```

## Commit D:

- using **echo** “feature 1 with 2 bugs” is written into the fileA.txt
- the fileA.txt is added to the staging.
- Using **commit** command the fileA.txt is sent from staging area to the local repository.

MINGW64: /d/assignment

```
M1057704@G1C2ML27801 MINGW64 /d/assignment (feature1)
$ echo feature 1 with 2 bugs >> fileA.txt

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature1)
$ cat fileA.txt

feature 1 wip
feature 1 with 2 bugs

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature1)
$ git add .
warning: LF will be replaced by CRLF in fileA.txt.
The file will have its original line endings in your working directory.

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature1)
$ git commit -m "add feature1"
[feature1 7e97f8a] add feature1
1 file changed, 1 insertion(+)

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature1)
$ git push -u origin feature1
Username for 'https://github.com': pavanmanu1995
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 309 bytes | 154.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/Pavanmanu1995/assignment.git
 a850b38..7e97f8a feature1 -> feature1
Branch 'feature1' set up to track remote branch 'feature1' from 'origin'.
```

- Feature1 branch is created in the github.

The screenshot shows a GitHub repository interface. At the top, there are statistics: 2 commits, 3 branches, 0 packages, 0 releases, and 0 contributors. Below this, a section titled 'Your recently pushed branches:' lists two branches: 'develop' (pushed 31 minutes ago) and 'feature1' (pushed 1 minute ago). Each branch has a 'Compare & pull request' button. Below the branches list, there are buttons for 'Branch: develop', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. A dropdown menu for 'Switch branches/tags' is open, showing a search bar 'Find or create a branch...' and a list of branches: 'master' (marked as default), 'develop' (checked with a green checkmark), and 'feature1'. To the right of the dropdown, there is a 'Pull request' button and a 'Compare' button. Below the dropdown, a table shows the commit history. The latest commit is '8c067db' from 35 minutes ago, with the message 'add fileA.txt'. Below the commit history, there is a button 'Add a README'.

## Commit E:

- Using **checkout** the branch is changed from feature1 branch to the develop branch.
- **Merge** : it is used to merge the content of feature1 branch to the develop branch.
- After merging it is added to the staging area using **add** .
- Using **commit** command it is sent to the local repository from the staging area.
- Then it is pushed to the remote repository.

```
MINGW64:/d/assignment
M1057704@G1C2ML27801 MINGW64 /d/assignment (feature1)
$ git checkout develop
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.

M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ git merge feature1
Updating 8c067db..7e97f8a
Fast-forward
 fileA.txt | 2 ++
 1 file changed, 2 insertions(+)

M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ git add .

M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ git commit -m "Merge branch 'feature1' into develop"
On branch develop
Your branch is ahead of 'origin/develop' by 2 commits.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ git push -u origin develop
Username for 'https://github.com': pavanmanu1995
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/Pavanmanu1995/assignment.git
 8c067db..7e97f8a develop -> develop
Branch 'develop' set up to track remote branch 'develop' from 'origin'.
```

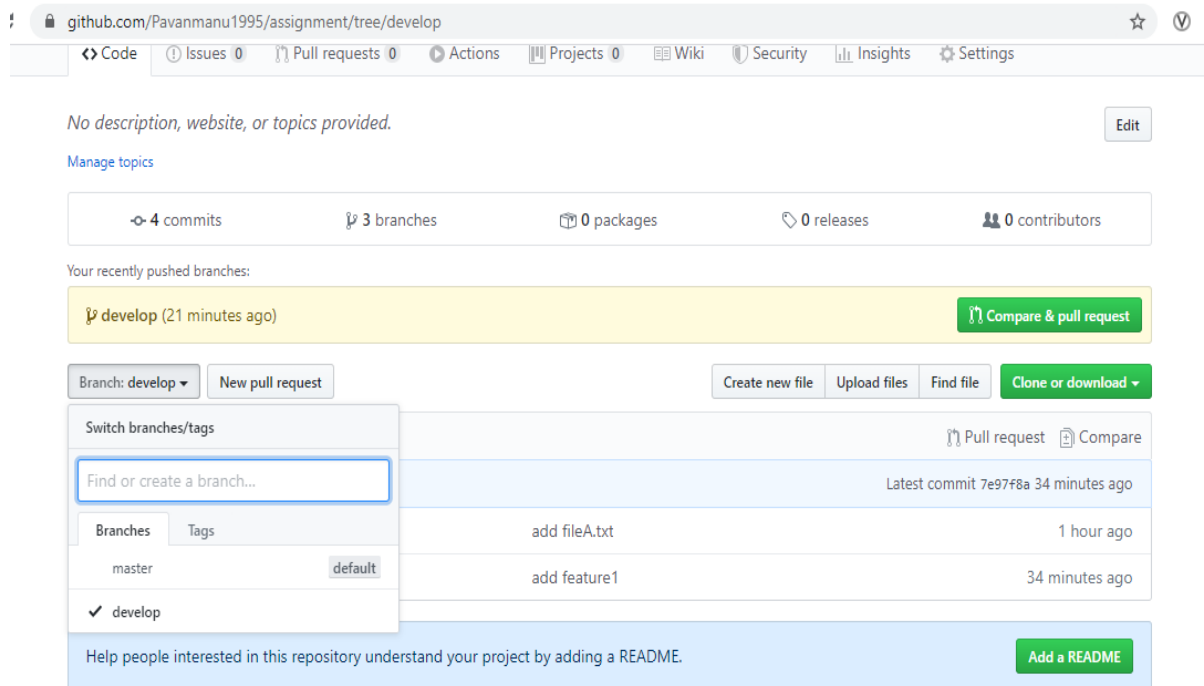
- **-d** : it is used to delete the branch.
- This command will not delete the branch in the github.

```
M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ git branch -d feature1
Deleted branch feature1 (was 7e97f8a).
```

- **push origin :branchName** will delete the branch in github.

```
M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ git push origin :feature1
Username for 'https://github.com': pavanmanu1995
To https://github.com/Pavanmanu1995/assignment.git
 - [deleted]          feature1
```

- The feature1 is deleted in the github.
- So there are only two branches that is master and develop branch.



## Commit F:

- Branch feature2 is created using **branch** command.
- “Feature 1 with 2 bugs feature 2 wip” is written to the fileA.txt.
- Then it is added to the staging area using **add** .
- Using **commit** command it is sent to the local repository from the staging area.
- Then it is pushed to the remote repository.

```
MINGW64:/d/assignment

M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ git branch feature2

M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ git checkout feature2
Switched to branch 'feature2'

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2)
$ echo feature 1 with 2 bugs feature 2 wip >> fileA.txt

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2)
$ cat fileA.txt

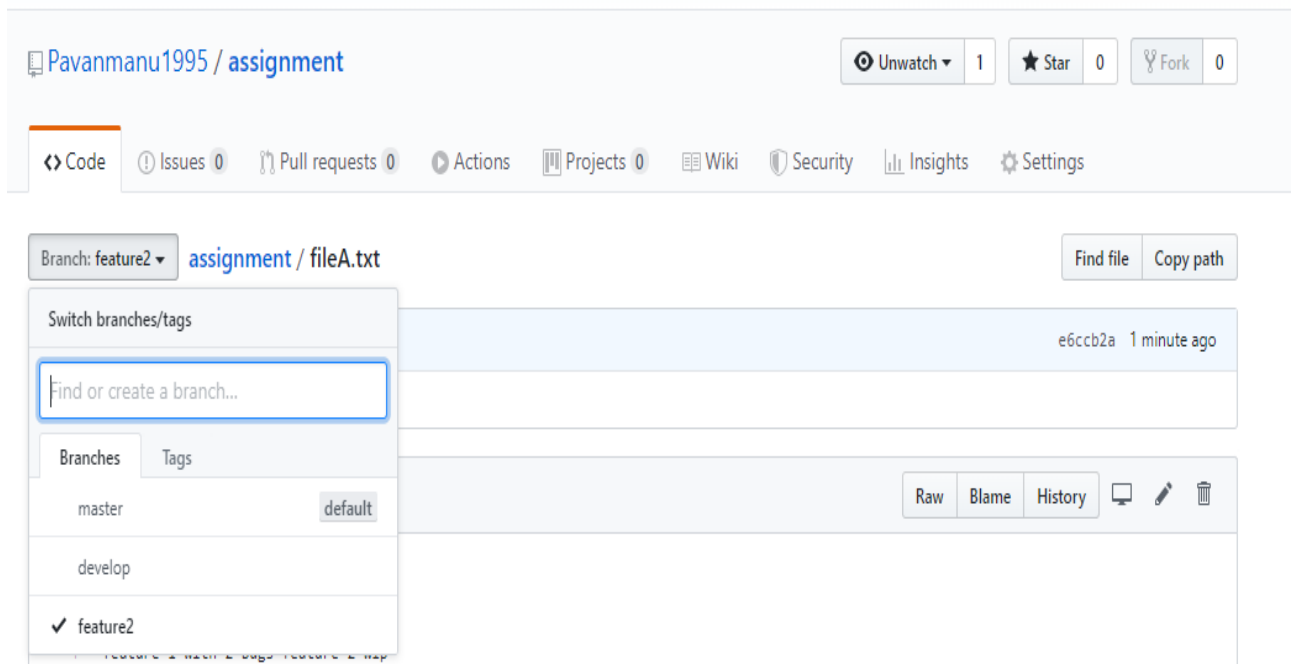
feature 1 wip
feature 1 with 2 bugs
feature 1 with 2 bugs feature 2 wip

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2)
$ git add
warning: LF will be replaced by CRLF in fileA.txt.
The file will have its original line endings in your working directory.

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2)
$ git commit -m "feature 2 wip"
[feature2 e6ccb2a] feature 2 wip
1 file changed, 1 insertion(+)

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2)
$ git push -u origin feature2
Username for 'https://github.com': pavanmanu1995
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), 316 bytes | 316.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
```

- Feature2 branch is created in github.





## Commit G:

- Release branch is created using **branch** command.
- Using **echo** command “feature 1 with 1 bug” is written to the fileA.txt.

```
MINGW64:/d/assignment
M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2)
$ git branch release1
M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2)
$ git checkout release1
Switched to branch 'release1'
M1057704@G1C2ML27801 MINGW64 /d/assignment (release1)
$ echo feature 1 with 1 bug >> fileA.txt
M1057704@G1C2ML27801 MINGW64 /d/assignment (release1)
$ cat fileA.txt
feature 1 wip
feature 1 with 2 bugs
feature 1 with 2 bugs feature 2 wip
feature 1 with 1 bug
M1057704@G1C2ML27801 MINGW64 /d/assignment (release1)
$ git add .
warning: LF will be replaced by CRLF in fileA.txt.
The file will have its original line endings in your working directory.
M1057704@G1C2ML27801 MINGW64 /d/assignment (release1)
$ git commit -m "fix feature 1 bug X"
[release1 d290758] fix feature 1 bug X
1 file changed, 1 insertion(+)
M1057704@G1C2ML27801 MINGW64 /d/assignment (release1)
$ git push -u origin release1
Username for 'https://github.com': pavanmanu1995
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), 332 bytes | 166.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
```

## Commit H:

- Using **merge** command release1 is merged with the master branch.
- Then it is added to the staging area using add .
- Using commit command it is sent to the local repository from the staging area.
- Then it is pushed to the remote repository.

```
MINGW64:/d/assignment
M1057704@G1C2ML27801 MINGW64 /d/assignment (release1)
$ git checkout master
warning: unable to rmdir 'assignment': Directory not empty
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
M1057704@G1C2ML27801 MINGW64 /d/assignment (master)
$ git merge release1
Updating 2f2210d..d290758
Fast-forward
 assignment | 1 +
 fileA.txt | 5 +++++
 2 files changed, 6 insertions(+)
 create mode 160000 assignment
M1057704@G1C2ML27801 MINGW64 /d/assignment (master)
$ echo results in feature 1 with 1 bug >> fileA.txt
M1057704@G1C2ML27801 MINGW64 /d/assignment (master)
$ git add .
warning: LF will be replaced by CRLF in fileA.txt.
The file will have its original line endings in your working directory.
M1057704@G1C2ML27801 MINGW64 /d/assignment (master)
$ git commit -m "Merge branch realese1"
[master 82ff66d] Merge branch realese1
1 file changed, 1 insertion(+)
M1057704@G1C2ML27801 MINGW64 /d/assignment (master)
$ git push -u origin master
Username for 'https://github.com': pavanmanu1995
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), 342 bytes | 342.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
```

## Commit I:

- Using **merge** command release1 is merged with the develop branch.
- Then it is added to the staging area using **add**.
- Using **commit** command it is sent to the local repository from the staging area.
- Then it is pushed to the remote repository.
- After merging the release1 branch is deleted

```
MINGW64:/d/assignment
M1057704@G1C2ML27801 MINGW64 /d/assignment (master)
$ git checkout develop
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ git merge release1
Updating 7e97f8a..d290758
Fast-forward
 fileA.txt | 2 ++
 1 file changed, 2 insertions(+)
M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ git add .
M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ git commit -m "Merge branch release1 into develop"
On branch develop
Your branch is ahead of 'origin/develop' by 2 commits.
(use "git push" to publish your local commits)

nothing to commit, working tree clean
M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ git push -u origin develop
Username for 'https://github.com': pavanmanu1995
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/Pavanmanu1995/assignment.git
 7e97f8a..d290758  develop -> develop
Branch 'develop' set up to track remote branch 'develop' from 'origin'.
M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ git branch -d release1
Deleted branch release1 (was d290758).
M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ git push origin :release1
```

- Merging of release1 branch in github

Pavanmanu1995 / assignment Unwatch 1 Star 0 Fork 0

[Code](#) [Issues 0](#) [Pull requests 0](#) [Actions](#) [Projects 0](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

No description, website, or topics provided. [Edit](#)

[Manage topics](#)

7 commits 4 branches 0 packages 0 releases 0 contributors

Branch: master [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#) [Clone or download](#)

M1057704 and M1057704 Merge branch reale1		Latest commit 82ff66d 2 minutes ago
assignment	add fileA.txt	1 hour ago
fileA.txt	Merge branch reale1	2 minutes ago

Help people interested in this repository understand your project by adding a README. [Add a README](#)

## Commit F1:

- Initially the branch is pointing to the develop branch.
- Using **checkout** command the branch is pointing to the feature2 branch.
- Using **echo** command “feature 1 with 1 bug feature 2 wip” is written to the fileA.txt.
- Then it is added to the staging area using **add .** command.
- Using **commit** command it is sent to the local repository from the staging area.
- Only after commit rebase can be done.
- Rebase:** it is similar to merge but the difference is merge creates a new commit to merge master and sub-branch, but rebase is used merge a feature branch onto master, you move the base of the feature branch to master branch’s ending point.
- when rebase is done conflict arises, that conflict need to be resolved using **mergetool** command, it will take to a new window which gives a option to change or keep both the things then is saved using :wq .
- conflict arises when two people changed the same lines in a file, or if one developer deleted a file while another developer was modifying it.

```
MINGW64:/d/assignment
M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ git checkout feature2
Switched to branch 'feature2'
Your branch is up to date with 'origin/feature2'.

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2)
$ echo feature 1 with 1 bug feature 2 wip >> fileA.txt

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2)
$ git rebase master
Cannot rebase: You have unstaged changes.
Please commit or stash them.


M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2)
$ git add .
warning: LF will be replaced by CRLF in fileA.txt.
The file will have its original line endings in your working directory.

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2)
$ git commit -m "feature 2 wip"
[feature2 e240bd8] feature 2 wip
1 file changed, 1 insertion(+)

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2)
$ git rebase master
First, rewinding head to replay your work on top of it...
Applying: feature 2 wip
Using index info to reconstruct a base tree...
M       fileA.txt
Falling back to patching base and 3-way merge...
Auto-merging fileA.txt
CONFLICT (content): Merge conflict in fileA.txt
error: Failed to merge in the changes.
Patch failed at 0001 feature 2 wip
Use 'git am --show-current-patch' to see the failed patch

Resolve all conflicts manually, mark them as resolved with
```

- After resolving the conflict **rebase --continue** command is used to continue the rebase to merge the feature2 branch to the master branch.
- Then it is pushed to the remote repository.

 MINGW64:/d/assignment

```
Resolve all conflicts manually, mark them as resolved with
M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2|REBASE 1/1)
$ git rebase --continue
Applying: feature 2 wip

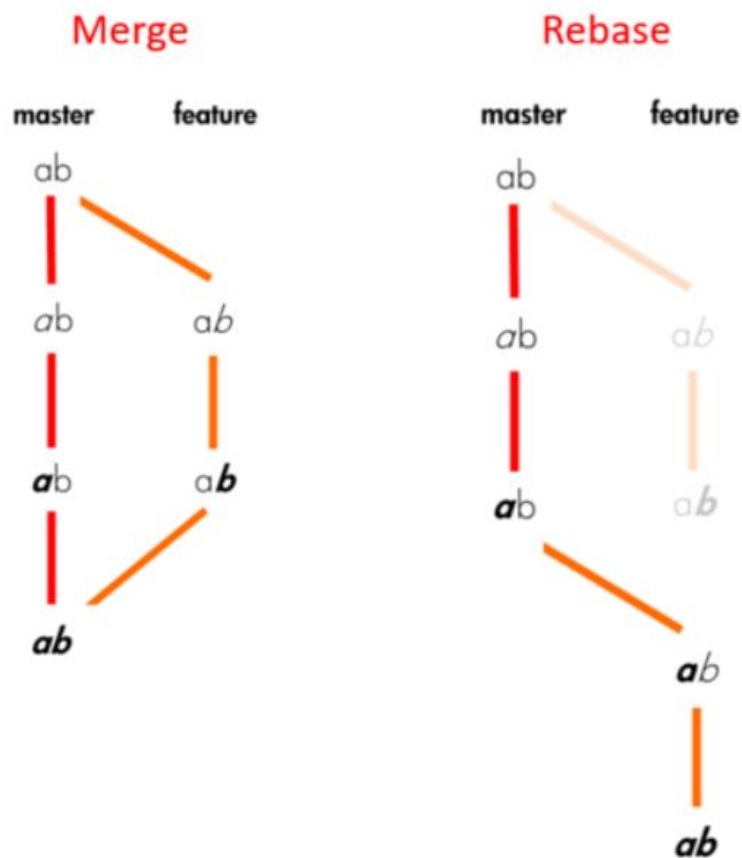
M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2)
$ git add .

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2)
$ git commit -m "feature 2 wip"
[feature2 22656bc] feature 2 wip
1 file changed, 10 insertions(+)
create mode 100644 fileA.txt.orig

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2)
$ git push -u origin feature2
Username for 'https://github.com': pavanmanu1995
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 542 bytes | 542.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Pavanmanu1995/assignment.git
e6ccb2a..22656bc feature2 -> feature2
Branch 'feature2' set up to track remote branch 'feature2' from 'origin'.
```

**Merge:** merging commits makes changes to another branch. It takes the contents of a source branch and integrates it with a target branch. In this process, only the target branch is changed. The source branch history remains.

**Rebase:** compresses all the changes into a single “patch.” Then it integrates the patch onto the target branch. It transfers the completed work from one branch to another. In the process, history is eliminated.



## Commit K:

- Create a new branch “hotfix1” using **branch** command.
- Initially the branch is pointing to the feature2 branch using **checkout** command it is changed to hotfix1 branch.
- Using **echo** command “feature 1” is written to the fileA.txt.
- Then it is added to the staging area using **add .** command.
- Using **commit** command it is sent to the local repository from the staging area.
- Then it is pushed to the remote repository.

```
MINGW64:/d/assignment

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2)
$ git branch hotfix1

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2)
$ git checkout hotfix1
Switched to branch 'hotfix1'

M1057704@G1C2ML27801 MINGW64 /d/assignment (hotfix1)
$ echo feature1 >> fileA.txt

M1057704@G1C2ML27801 MINGW64 /d/assignment (hotfix1)
$ git add .
warning: LF will be replaced by CRLF in fileA.txt.
The file will have its original line endings in your working directory.

M1057704@G1C2ML27801 MINGW64 /d/assignment (hotfix1)
$ git commit -m "fix feature 1 bug Y"
[hotfix1 7def4dc] fix feature 1 bug Y
1 file changed, 1 insertion(+)

M1057704@G1C2ML27801 MINGW64 /d/assignment (hotfix1)
$ git push -u origin hotfix1
Username for 'https://github.com': pavanmanu1995
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), 356 bytes | 356.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'hotfix1' on GitHub by visiting:
remote:   https://github.com/Pavanmanu1995/assignment/pull/new/hotfix1
remote:
To https://github.com/Pavanmanu1995/assignment.git
 * [new branch]      hotfix1 -> hotfix1
Branch 'hotfix1' set up to track remote branch 'hotfix1' from 'origin'.
```

## Commit L:

- Initially the branch is pointing to the hotfix1 branch using **checkout** command it is changed to master branch.
- Using **echo** command “feature 1” is written to the fileA.txt.
- Then it is added to the staging area using **add .** command.
- Using **commit** command it is sent to the local repository from the staging area.
- Hotfix1 branch is merged using merge command to the master branch.
- Then it is pushed to the remote repository.

```
MINGW64:/d/assignment
M1057704@G1C2ML27801 MINGW64 /d/assignment (hotfix1)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
M1057704@G1C2ML27801 MINGW64 /d/assignment (master)
$ echo feature1 >> fileA.txt
M1057704@G1C2ML27801 MINGW64 /d/assignment (master)
$ git merge hotfix1
error: Your local changes to the following files would be overwritten by merge:
       fileA.txt
Please commit your changes or stash them before you merge.
Aborting
Updating 82ff66d..7def4dc
M1057704@G1C2ML27801 MINGW64 /d/assignment (master)
$ git add .
warning: LF will be replaced by CRLF in fileA.txt.
The file will have its original line endings in your working directory.
M1057704@G1C2ML27801 MINGW64 /d/assignment (master)
$ git commit -m "Merge branch hotfix1"
[master 41fb1ea] Merge branch hotfix1
1 file changed, 1 insertion(+)
M1057704@G1C2ML27801 MINGW64 /d/assignment (master)
$ git merge hotfix1
Auto-merging fileA.txt
CONFLICT (content): Merge conflict in fileA.txt
Automatic merge failed; fix conflicts and then commit the result.
M1057704@G1C2ML27801 MINGW64 /d/assignment (master|MERGING)
$ git mergetool

This message is displayed because 'merge.tool' is not configured.
See 'git mergetool --tool-help' or 'git help config' for more details.
```

## Commit M:

- Initially the branch is pointing to the master.
- Using **checkout** command change the pointing of the branch to develop branch.
- Using **echo** command “feature 1” is written to the fileA.txt.
- Then it is added to the staging area using **add .** command.
- Using **commit** command it is sent to the local repository from the staging area.
- Hotfix1 branch is merged using **merge** command to the develop branch.
- Then it is pushed to the remote repository.


```
M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ git merge hotfix1
Already up to date.

M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ git push -u origin develop
Username for 'https://github.com': pavanmanu1995
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 718 bytes | 718.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/Pavanmanu1995/assignment.git
    f02a7fe..a553f11  develop -> develop
Branch 'develop' set up to track remote branch 'develop' from 'origin'.
```

- Hotfix1 branch is deleted using **push origin :hotfix1** command.

```
M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2|REBASE 1/1)
$ git push origin :hotfix1
Username for 'https://github.com': pavanmanu1995
To https://github.com/Pavanmanu1995/assignment.git
- [deleted]          hotfix1

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2|REBASE 1/1)
$
```

A screenshot of the Windows taskbar at the bottom of the terminal window. It includes the Start button, a search bar with the text "Type here to search", and several pinned application icons: a web browser, a file explorer, and a communication app.



## Commit F2:

- Initially the branch is pointing to the develop.
- Using **checkout** command change the pointing of the branch to feature2 branch.
- Using **echo** command “feature 1 feature 2 wip” is written to the fileA.txt.
- Then it is added to the staging area using **add .** command.
- Using **commit** command it is sent to the local repository from the staging area.
- Hotfix1 branch is merged using **merge** command to the develop branch.

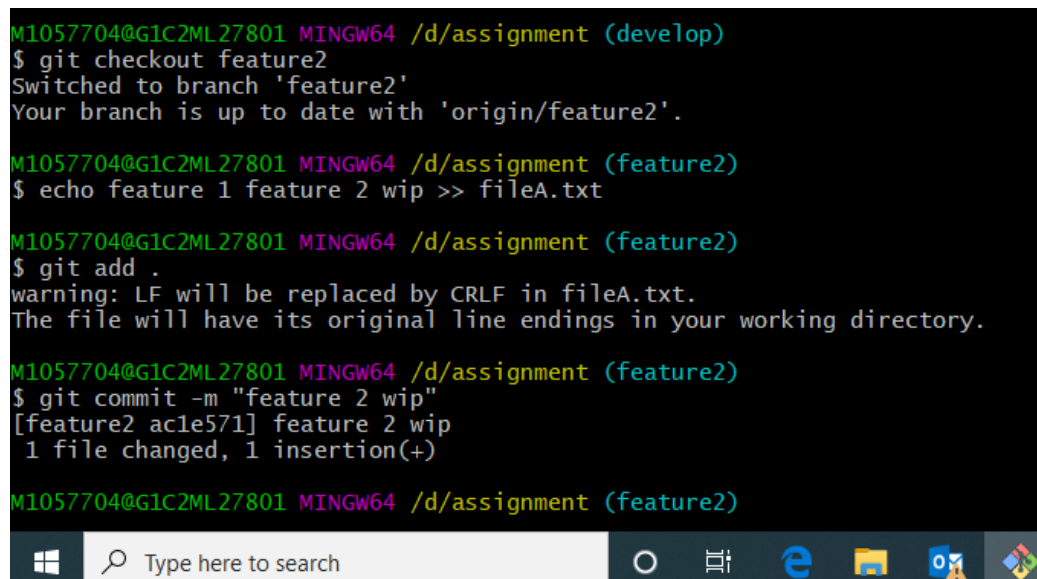
```
M1057704@G1C2ML27801 MINGW64 /d/assignment (develop)
$ git checkout feature2
Switched to branch 'feature2'
Your branch is up to date with 'origin/feature2'.

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2)
$ echo feature 1 feature 2 wip >> fileA.txt

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2)
$ git add .
warning: LF will be replaced by CRLF in fileA.txt.
The file will have its original line endings in your working directory.

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2)
$ git commit -m "feature 2 wip"
[feature2 ac1e571] feature 2 wip
1 file changed, 1 insertion(+)

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2)
```



- After commit rebase command is used to marge the feature2 branch to the master branch.
- Conflict arises it should be resolved using **mergetool** command then it takes to the new window conflict is resolved and :wq is used to save and quit.

```

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2)
$ git rebase master
First, rewinding head to replay your work on top of it...
Applying: feature 2 wip
Using index info to reconstruct a base tree...
M       fileA.txt
Falling back to patching base and 3-way merge...
Auto-merging fileA.txt
CONFLICT (content): Merge conflict in fileA.txt
error: Failed to merge in the changes.
Patch failed at 0001 feature 2 wip
Use 'git am --show-current-patch' to see the failed patch

Resolve all conflicts manually, mark them as resolved with
"git add/rm <conflicted_files>", then run "git rebase --continue".
You can instead skip this commit: run "git rebase --skip".
To abort and get back to the state before "git rebase", run "git rebase --abort".

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2|REBASE 1/1)
$

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2|REBASE 1/1)
$ git mergetool

This message is displayed because 'merge.tool' is not configured.
See 'git mergetool --tool-help' or 'git help config' for more details.
'git mergetool' will now attempt to use one of the following tools:
opendiff kdiff3 tkdiff xxdiff meld tortoisemerge gvimdiff diffuze diffmerge ecmerge p4merge araxis bc codecompare emerge vimdiff
Merging:
fileA.txt

Normal merge conflict for 'fileA.txt':
{local}: modified file
{remote}: modified file
Hit return to start merge resolution tool (vimdiff):
4 files to edit

```

- After saving rebase --continue command is used to continue rebase to merge to the master branch.
- Then it is pushed to the remote repository.

```

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2|REBASE 1/1)
$ git rebase --continue
fileA.txt.orig: needs update
You must edit all merge conflicts and then
mark them as resolved using git add

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2|REBASE 1/1)
$ git add .

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2|REBASE 1/1)
$ git commit -m "feature 2 wip"
[detached HEAD b42f2e8] feature 2 wip
2 files changed, 8 insertions(+)

M1057704@G1C2ML27801 MINGW64 /d/assignment (feature2|REBASE 1/1)
$ git push -u origin feature2
Username for 'https://github.com': pavanmanu1995
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), 358 bytes | 358.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Pavanmanu1995/assignment.git
22656bc..ac1e571 feature2 -> feature2
Branch 'feature2' set up to track remote branch 'feature2' from 'origin'.

```