

INDEX

EXPT NO.	DATE	NAME OF THE EXPT	PAGE NO.	SIGN
1	25/11/23	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.	7	
2	09/12/23	Create a new branch named "feature-branch." Switch to the "master" branch. Merge the "feature-branch" into "master".	11	
3	09/12/23	Write the commands to stash your changes, switch branches, and then apply the stashed changes.	13	
4	23/12/2	Clone a remote Git repository to your local machine.	1	
5	3 30/12/2	Fetch the latest changes from a remote repository and rebase your local branch onto the updated remote branch.	5 1	
6	13/01/24	Write the command to merge "feature-branch" into "master" while providing a custom commit message for the merge.	7	
7	27/01/24	Write the command to create a lightweight Git tag named "v1.0" for a commit in your local repository.	21	
8	27/01/24	Write the command to cherry-pick a range of commits from "source-branch" to the current branch.	23	
9	10/01/24	Given a commit ID, how would you use Git to view the details of that specific commit, including the author, date, and commit message?.	27	
10	10/02/24	Write the command to list all commits made by the author "JohnDoe" between "2023-01-01" and "2023-12-31."	29	
11	24/02/24	Write the command to display the last five commits in the repository's history.	29	
12	24/02/24	Write the command to undo the changes introduced by the commit with the ID "abc123".	31	

PROJECT MANAGEMENT WITH GIT

1. Setting Up and Basic Commands :

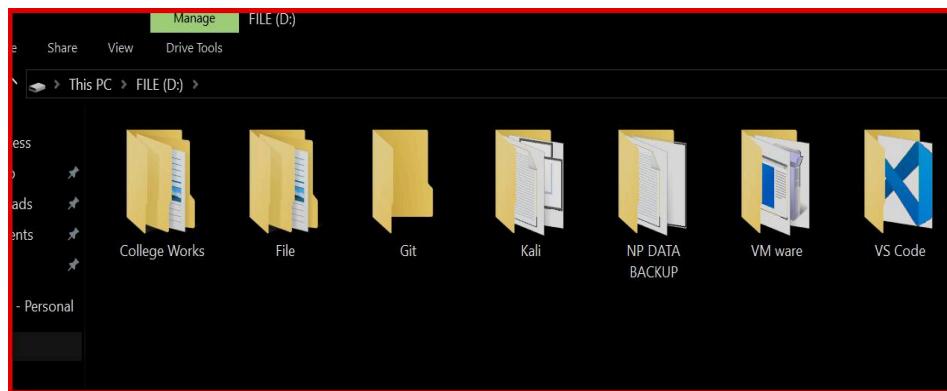
- Initialize a new Git repository in a directory. Create a new file and add it to the staging area and commit the changes with an appropriate commit message.

Step 1:

```
ACER@Kumar MINGW32 ~ (master)
$ mkdir Git

ACER@Kumar MINGW32 ~ (master)
$ cd Git
```

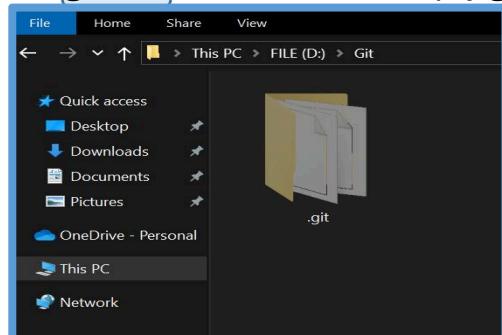
- `mkdir Git` Creates a folder/directory in the present working directory.
- `cd Git` Changes the directory to the git folder which was created.



Step 2:

```
ACER@Kumar MINGW32 ~ (master)
$ git init
Initialized empty Git repository in D:/Git/.git/
```

- `git init` Initializes a empty git repository.



- We can see the .git folder created in the git folder ,in some cases the file is hidden and to see that hidden file we need to click on view the hidden files.

Step 3:

```
ACER@Kumar MINGW32 ~ (master)
$ vim file.txt
```

- vim file.txt □ Creates a empty file of txt extension in the current directory.

```
MINGW64:/D/Git
-----
Hello This is first line of master code
~
```

- Writing or adding content to the file.

```
ACER@Kumar MINGW32 ~ (master)
$ git add .
warning: in the working copy of 'file.txt'
```

- `git add ./ git add <file name>` □ Stage the file with add . or add <file name> add. will stage the whole files in the current directory and are ready to commit.
- Commit the file with appropriate commit message.

```
ACER@Kumar MINGW32 ~ (master)
$ git commit -m "File created(V1)"
[master (root-commit) 990b321] File created(V1)
 1 file changed, 2 insertions(+)
 create mode 100644 file.txt
```

Step 4:

```
ACER@Kumar MINGW32 ~ (master)
$ git log
commit 990b32135d54f774a873d7db2bf520ae7285617b (HEAD -> master)
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:13:19 2024 +0530

  File created(V1)
```

- `git log` Displays all the history of commits with commit messages along with the author name and email.
- Every commits has a unique commit ID.

Step 5:

```
ACER@Kumar MINGW32 ~ (master)
$ vim file.txt
```

```
MINGW64:/D/Git
```

```
Hello This is first line of master code
```

```
Hi This is 2nd line of master code
```

```
~  
~
```

- If we want to add anything else to existing file, then we open it add or write the contents.

```
ACER@Kumar MINGW32 ~ (master)
$ git add file.txt
warning: in the working copy of 'file.txt'
```

```
ACER@Kumar MINGW32 ~ (master)
$ git commit -m "2 lines are added(v2)"
[master 4971195] 2 lines are added(v2)
 1 file changed, 3 insertions(+)
```

- Then again stage the file and commit it with appropriate commit message.

Step 6:

```
ACER@Kumar MINGW32 ~ (master)
$ git status
On branch master
nothing to commit, working tree clean
```

- `git status` □ It checks the status, like on which branch we are and is there any changes made which are not committed .
- If no any other changes has been made after recent commit, it displays the working tree is clean.

Step 7:

```
ACER@Kumar MINGW32 ~ (master)
$ git log
commit 4971195ca5c642240bd461b45cb5db5fef75f076 (HEAD -> master)
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:18:14 2024 +0530

    2 lines are added(v2)

commit 990b32135d54f774a873d7db2bf520ae7285617b
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:13:19 2024 +0530

    File created(v1)
```

- `git log` □ This will display the history of the commits.

2. Creating and Managing Branches :

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

Step 1:

```
ACER@Kumar MINGW32 ~ (master)
$ git branch feature-branch
```

```
ACER@Kumar MINGW32 ~ (master)
$ git branch
  feature-branch
* master
```

- `git branch feature-branch` □ This will create a new branch of the master branch in which the contents and files are copied from the master branch.
- `git branch` □ This command will show all the branches which we have made and the current branch will be in green colour.

Step 2:

```
ACER@Kumar MINGW32 ~ (master)
$ git checkout feature-branch
Switched to branch 'feature-branch'
```

- `git checkout <branch name>` □ It is used to switch one branch to other branch, here we are moving from branch master to the feature-branch.

```
ACER@Kumar MINGW32 ~ (master)
$ vim file.txt
```

- And also we can edited the file (file.txt).

```
MINGW64/D/Git
-----
Hello This is first line of master code
-----
Hi This is 2nd line of master code
-----
Hello This is 1st line of FB code
~
```

Step 3:

```
ACER@Kumar MINGW32 ~ (master)
$ git add .
warning: in the working copy of 'file.txt', LF
```

```
ACER@Kumar MINGW32 ~ (master)
$ git commit -m "1st line of FB(v1)"
[feature-branch 4e73841] 1st line of FB(v1)
 1 file changed, 8 insertions(+)
```

- Here in the feature-branch we staged the file and commit with appropriate message that “1st line of FB (v1)”.
- So here the file has been changed ,but in the master branch it will be as it is until we merge the feature-branch with the master branch.

Step 4:

```
ACER@Kumar MINGW32 ~ (master)
$ git checkout master
Switched to branch 'master'
```

```
ACER@Kumar MINGW32 ~ (master)
$ git merge feature-branch
Updating 4971195..4e73841
Fast-forward
  file.txt | 8 ++++++++
  1 file changed, 8 insertions(+)
```

- For merging the file to the master branch we first need to move to the main/master branch using “git checkout master” command.
- Then we can merge the branch with “git merge feature-branch” command.
- So ,now the files will be merged ,the changes or the edits in the feature-branch will be merged.

Step 5:

```
ACER@Kumar MINGW32 ~ (master)
$ git log
commit 4e7384168d0b6746e0e8621d1d19f36b53231a4e (HEAD -> master, feature-branch)
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:29:03 2024 +0530

  1st line of FB(v1)

commit 4971195ca5c642240bd461b45cb5db5fef75f076
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:18:14 2024 +0530

  2 lines are added(v2)

commit 990b32135d54f774a873d7db2bf520ae7285617b
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:13:19 2024 +0530

  File created(v1)
```

- git log □ Here the history of the commits will be visible.

3. Creating and Managing Branches :

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

Step 1:

```
ACER@Kumar MINGW32 ~ (master)
$ git checkout feature-branch
Switched to branch 'feature-branch'
```

```
ACER@Kumar MINGW32 ~ (master)
$ vim file.txt
```

- Moving to the feature branch and have made changes in the file(file.txt) using the commands git checkout feature-branch and vi file.txt for changing the branch and editing the file respectively.

```
MINGW64:/D/Git
-----
Hello This is first line of master code
-----
Hi This is 2nd line of master code
-----
Hello This is 1st line of FB code
-----
Hi This is 2nd line of FB code and stash code
~
```

- Here we have not staged and committed the changes in the file(file.txt).

Step 2:

```
ACER@Kumar MINGW32 ~ (master)
$ git stash
Saved working directory and index state WIP on feature-branch: 4e73841 1st line of FB(v1)
```

- In this step we have stashed the changes which have been made in the file.txt file in the feature-branch.
- Here we have not added/staged the file and committed the changes.
- The changes will be saved in the branch without committing the changes.
- The command used for stashing the changes is “git stash”

Step 3:

```
ACER@Kumar MINGW32 ~ (master)
$ git stash apply
On branch master
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:   file.txt

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

- Before applying the changes made in the feature-branch, first we need move to the master branch.
- Then in the master branch we applied the stash using “git stash apply” where changes made in the feature-branch.
- After applying the stash to the master. It will give us a message saying that the applied stash is not staged and committed in the master branch.

Step 4:

```
$ git status
On branch master
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:   text.txt

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

- After stashing we check status using “git status”, there it shows that the file(file.txt) is modified but not staged yet.

```
ACER@Kumar MINGW32 ~ (master)
$ git add .
warning: in the working copy of 'file.txt', LF
```

- After staging, if we check the status again it shows that changes to be committed yet

```
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   text.txt
```

```
ACER@Kumar MINGW32 ~ (master)
$ git commit -m "FB code & Stash code"
[master 4c23379] FB code & Stash code
 1 file changed, 4 insertions(+)
```

- Committed with appropriate message that “FB code & Stash code”

```
ACER@Kumar MINGW32 ~ (master)
$ git log
commit 4c23379aa3389cdd8cd071512038120780b805ea (HEAD -> master)
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:50:31 2024 +0530

  FB code & Stash code

commit 4e7384168d0b6746e0e8621d1d19f36b53231a4e (feature-branch)
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:29:03 2024 +0530

  1st line of FB(v1)

commit 4971195ca5c642240bd461b45cb5db5fef75f076
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:18:14 2024 +0530

  2 lines are added(v2)

commit 990b32135d54f774a873d7db2bf520ae7285617b
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:13:19 2024 +0530

  File created(v1)
```

□ After committing the changes we can see the log of the repository.

4. Collaboration and Remote Repositories :

- Clone a remote Git repository to your local machine.

Step1:

A screenshot of a GitHub repository page for 'Kumar-Rathod'. The repository has 1 branch and 0 tags. The README file contains the text 'Kumar-Rathod-' and 'Git hub'. The repository has 1 commit from 'Kumar-Rathod' made 31 minutes ago. The sidebar shows basic repository statistics: 0 stars, 1 watching, and 0 forks. There are sections for Releases and Packages, both of which are currently empty.

A screenshot of the same GitHub repository page, but with the 'Code' dropdown menu open. The 'Clone' section is highlighted, showing three options: Local, Codespaces, and HTTPS. The HTTPS option is selected, displaying the URL 'https://github.com/Kumar-Rathod/Kumar-Rathod-.git'. Below this are links for 'Open with GitHub Desktop' and 'Download ZIP'.

- To clone a remote git repository ,first we need to open the github.com and open any of the account on the github.com
- After that we have chosen the repository which we want to clone into our local machine.
- After choosing the repository , in that repository we clicked on the green button “code” ,which open a drop down list of links in that ,we copied the “HTTPS” link from that. <https://github.com/Kumar-Rathod/Kumar-Rathod-.git>

Step2:

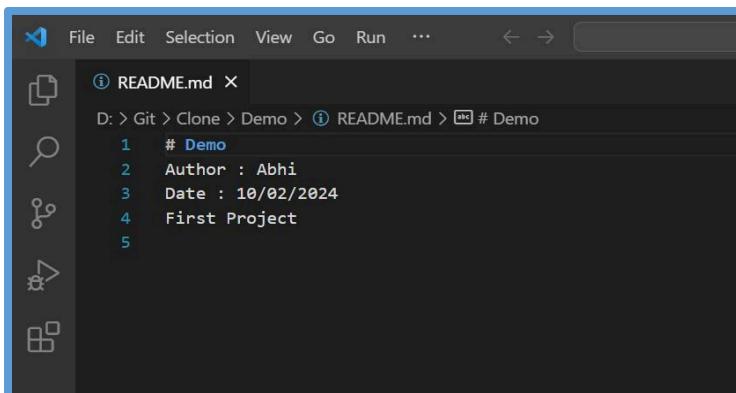
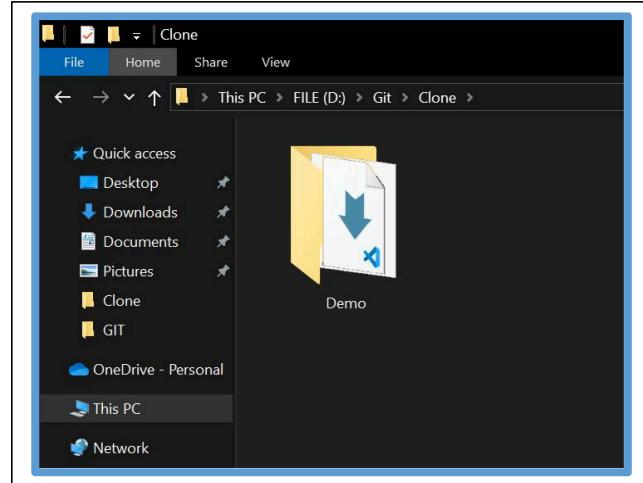
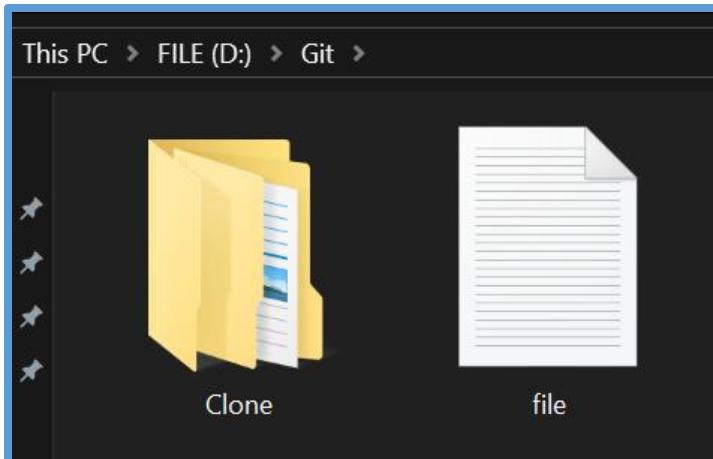
```
ACER@Kumar MINGW32 ~ (master)
$ mkdir Clone

ACER@Kumar MINGW32 ~ (master)
$ cd Clone
```

```

ACER@Kumar MINGW32 ~ (master)
$ git clone https://github.com/Abhishek-4949/Demo.git
Cloning into 'Demo'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.

```



- In the second step we need to open our git bash and in some directory where you need to clone the remote repository we need to move to that location using “cd <path name>” command.
- Here we want to copy the repository to the folder clone in the c so we moved to that location
- `gitclone https://github.com/Kumar-Rathod/Kumar-Rathod-.git` this command will copy the repository from remote to the local machine in the working directory.
- you can see above the “Demo” repository is successfully copied in the clone directory/folder.

5. Collaboration and Remote Repositories :

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

Step 1:

```
ACER@Kumar MINGW32 ~ (master)
$ git log
commit 915e5dbc13380901cd870ca96b34fd1f00e4da62 (HEAD -> main, origin/main, origin/HEAD, local)
Author: Abhishek-4949 <135841591+Abhishek-4949@users.noreply.github.com>
Date:   Mon Feb 26 09:39:18 2024 +0530

HF

commit 4262e0214276a96640fa7b4899eae6d8ad955704
Author: spoorti <spoort@gmail.com>
Date:   Sat Feb 24 12:31:14 2024 +0530

2 h

commit 31b1898d1473814bd5bef61800081cc28d6f29f2
Author: Abhishek-4949 <135841591+Abhishek-4949@users.noreply.github.com>
Date:   Sat Feb 24 12:18:50 2024 +0530

2 hi hello

commit 9e4774f51f4b7bf8af3ca94d5d5cb276ab11aee8
Author: Abhishek-4949 <135841591+Abhishek-4949@users.noreply.github.com>
Date:   Sat Feb 17 11:57:45 2024 +0530

Hi_Hlo

commit 272f7cf90458c67047227cf90ba8c3cf04c6ca6c
Author: Abhishek-4949 <135841591+Abhishek-4949@users.noreply.github.com>
Date:   Sun Feb 11 22:05:39 2024 +0530

1st code

commit 296b1e82fc07838d0f30f4b1558a4d64b8d41f9a
Author: Abhishek-4949 <135841591+Abhishek-4949@users.noreply.github.com>
Date:   Sun Feb 11 22:02:01 2024 +0530

Initial commit
```

- To fetch and rebase the remote repository to local repository ,we will move to the already cloned repo.
- Initially before fetching the changes from the remote repo the last commit was “created onefile”after loggingthecommits

Step 2:

```
ACER@Kumar MINGW32 ~ (master)
$ git fetch
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), 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), 957 bytes | 79.00 KiB/s, done.
From https://github.com/Abhishek-4949/Demo
  915e5db..24d0df1  main      -> origin/main
```

- git fetch origin/git fetch [] this will fetch the latest changes from the remote repo that is the file named “special_note” which was created and committed.
- These changes after fetching will not be available in the working directory.

Step 3:

```

ACER@Kumar MINGW32 ~ (master)
$ git fetch origin

ACER@Kumar MINGW32 ~ (master)
$ git rebase origin
Successfully rebased and updated refs/heads/main.

```

- git rebase origin [this command is used to bring the changes which are fetched and present in the local repo to the working directory]
- after rebasing the remote branch to local branch ,the commit which are made in remote repo that will added to local branch.

```

ACER@Kumar MINGW32 ~ (master)
$ git log
commit 24d0df10dd38916ab335602426e935aca9cdfbda (HEAD -> main, origin/main, origin/HEAD)
Author: Abhishek-4949 <135841591+Abhishek-4949@users.noreply.github.com>
Date:   Mon Feb 26 15:40:09 2024 +0530

    Create New file

commit 915e5dbc13380901cd870ca96b34fd1f00e4da62 (local)
Author: Abhishek-4949 <135841591+Abhishek-4949@users.noreply.github.com>
Date:   Mon Feb 26 09:39:18 2024 +0530

    HF

commit 4262e0214276a96640fa7b4899eae6d8ad955704
Author: spoorti <spoort@gmail.com>
Date:   Sat Feb 24 12:31:14 2024 +0530

    2 h

commit 31b1898d1473814bd5bef61800081cc28d6f29f2
Author: Abhishek-4949 <135841591+Abhishek-4949@users.noreply.github.com>
Date:   Sat Feb 24 12:18:50 2024 +0530

    2 hi hello

commit 9e4774f51f4b7bf8af3ca94d5d5cb276ab11aee8
Author: Abhishek-4949 <135841591+Abhishek-4949@users.noreply.github.com>
Date:   Sat Feb 17 11:57:45 2024 +0530

    Hi_Hlo

commit 272f7cf90458c67047227cf90ba8c3cf04c6ca6c
Author: Abhishek-4949 <135841591+Abhishek-4949@users.noreply.github.com>
Date:   Sun Feb 11 22:05:39 2024 +0530

    1st code

commit 296b1e82fc07838d0f30f4b1558a4d64b8d41f9a
Author: Abhishek-4949 <135841591+Abhishek-4949@users.noreply.github.com>
Date:   Sun Feb 11 22:02:01 2024 +0530

    Initial commit

```

6. Collaboration and Remote Repositories

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

```
ACER@Kumar MINGW32 ~ (master)
```

```
$ git checkout feature-branch  
Switched to branch 'feature-branch'
```

```
ACER@Kumar MINGW32 ~ (master)
```

```
$ vim file.txt
```

```
ACER@Kumar MINGW32 ~ (master)
```

```
$ git add .  
warning: in the working copy of 'file.txt', LF
```

```
ACER@Kumar MINGW32 ~ (master)
```

```
$ git commit -m "1st line of FB(v1)"  
[feature-branch 4e73841] 1st line of FB(v1)  
1 file changed, 8 insertions(+)
```

- Move to the feature branch and make some changes in the that branch ,stage and commit the changes.
- For committing we can use additional/optional □ -m “message” , will commit the state with appropriate message.

```
ACER@Kumar MINGW32 ~ (master)
```

```
$ git checkout master  
Switched to branch 'master'
```

```
ACER@Kumar MINGW32 ~ (master)
```

```
$ git merge feature-branch  
Updating 4971195..4e73841  
Fast-forward  
 file.txt | 8 +++++++  
 1 file changed, 8 insertions(+)
```

- Then checkout to the master branch and merge the branch

```
ACER@Kumar MINGW32 ~ (master)
```

```
$ git log  
commit 4e7384168d0b6746e0e8621d1d19f36b53231a4e (HEAD -> master, feature-branch)  
Author: Sakshi <sakshiburali@gmail.com>  
Date:   Sat Feb 10 19:29:03 2024 +0530  
  
    1st line of FB(v1)  
  
commit 4971195ca5c642240bd461b45cb5db5fef75f076  
Author: Sakshi <sakshiburali@gmail.com>  
Date:   Sat Feb 10 19:18:14 2024 +0530  
  
    2 lines are added(V2)  
  
commit 990b32135d54f774a873d7db2bf520ae7285617b  
Author: Sakshi <sakshiburali@gmail.com>  
Date:   Sat Feb 10 19:13:19 2024 +0530  
  
    File created(V1)
```

7. Git Tags and Releases

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

Step1:

```
ACER@Kumar MINGW32 ~ (master)
$ git tag v1.0
```

```
ACER@Kumar MINGW32 ~ (master)
$ git tag
v1.0
```

```
ACER@Kumar MINGW32 ~ (master)
$ git tag V1.1 4e7384168d0b6746e0e8621d1d19f36b53231a4e
```

```
ACER@Kumar MINGW32 ~ (master)
$ git tag
V1.0
V1.1
```

```
ACER@Kumar MINGW32 ~ (master)
$ git show V1.0
commit 4c23379aa3389cd8cd071512038120780b805ea (HEAD -> master, tag: V1.0)
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:50:31 2024 +0530

    FB code & Stash code

diff --git a/file.txt b/file.txt
index 74cff2c..9445a90 100644
--- a/file.txt
+++ b/file.txt
@@ -11,3 +11,7 @@ Hi This is 2nd line of master code
-----
Hello This is 1st line of FB code
+
+
-----
+Hi This is 2nd line of FB code and stash code
```

- `git tag v1.0` □ this will create a tag of the latest commit (or we can specify the particular commit with commit ID) or we can also add a tag message using `-m "message"`.
- `git tag` □ this command will show all tags made i.e., v1.0 created.
- `git show v1.0` □ this will show details in that tag (v1.0) with full description.

```
ACER@Kumar MINGW32 ~ (master)
$ git show V1.1
commit 4e7384168d0b6746e0e8621d1d19f36b53231a4e (tag: V1.1, feature-branch)
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:29:03 2024 +0530

    1st line of FB(v1)

diff --git a/file.txt b/file.txt
index a3ae407..74cff2c 100644
--- a/file.txt
+++ b/file.txt
@@ -3,3 +3,11 @@ Hello This is first line of master code

-----
Hi This is 2nd line of master code
+
+
+
+
+
+
+Hello This is 1st line of FB code
```

```
ACER@Kumar MINGW32 ~ (master)
$ git log
commit 4c23379aa3389cd8cd071512038120780b805ea (HEAD -> master, tag: V1.0)
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:50:31 2024 +0530

    FB code & Stash code

commit 4e7384168d0b6746e0e8621d1d19f36b53231a4e (tag: V1.1, feature-branch)
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:29:03 2024 +0530

    1st line of FB(v1)

commit 4971195ca5c642240bd461b45cb5db5fef75f076
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:18:14 2024 +0530

    2 lines are added(V2)

commit 990b32135d54f774a873d7db2bf520ae7285617b
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:13:19 2024 +0530

    File created(V1)
```

8. Advanced Git Operations

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

```
ACER@Kumar MINGW32 ~ (master)
$ git branch source-branch

ACER@Kumar MINGW32 ~ (master)
$ git checkout source-branch
Switched to branch 'source-branch'
```

```
ACER@Kumar MINGW32 ~ (master)
$ vim file.txt
```

```
-----
Hello This is first line of master code
-----
Hi This is 2nd line of master code
-----
Hello This is 1st line of FB code
-----
Hi This is 2nd line of FB code and stash code
-----
Source-branch is created
```

- Create a branch named source branch and check out to the source branch.
- And make the first some change in the file.txt file.

```
ACER@Kumar MINGW32 ~ (master)
$ git add file.txt

ACER@Kumar MINGW32 ~ (master)
$ git commit -m "file created"
[source-branch d6c2ffa] file created
 1 file changed, 4 insertions(+)
```

- Stage and commit the changes with commit message saying “first commit in the source branch.

```
ACER@Kumar MINGW32 ~ (master)
$ vim file.txt
```

- Again make some changes in the file or add few more line in the file.txt file.

```
-----
Hello This is first line of master code

-----
Hi This is 2nd line of master code

-----
Hello This is 1st line of FB code

-----
Hi This is 2nd line of FB code and stash code

-----
Source-branch is created

-----
Source-branch file edited|
~
```

```
ACER@Kumar MINGW32 ~ (master)
$ git add file.txt

ACER@Kumar MINGW32 ~ (master)
$ git commit -m "SB edited"
[source-branch 467b312] SB edited
 1 file changed, 3 insertions(+)
```

- Stage and commit the changes with message saying “SB edited”.

```
ACER@Kumar MINGW32 ~ (master)
$ git log
commit 467b312409f713aa405b5bd0fed67d0c9a4e1eba (HEAD -> source-branch)
Author: Abhishek-4949 <abhishekhiremath4949@gmail.com>
Date:   Mon Feb 26 18:20:06 2024 +0530

    SB edited

commit d6c2ffa931df5e1f9cc9cedcccd9ff8d01da45b43
Author: Abhishek-4949 <abhishekhiremath4949@gmail.com>
Date:   Mon Feb 26 18:17:25 2024 +0530

    file created

commit 4c23379aa3389cd8cd071512038120780b805ea (tag: V1.0)
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:50:31 2024 +0530

    FB code & Stash code
```

- Here we want copy the commit ID to cherry-pick the specific state from the git log

```
ACER@Kumar MINGW32 ~ (master)
$ git checkout master
Switched to branch 'master'
```

```
ACER@Kumar MINGW32 ~ (master)
$ git cherry-pick d6c2ffa931df5e1f9cc9cedccd9ff8d01da45b43
[master 60b5560] file created
Date: Mon Feb 26 18:17:25 2024 +0530
1 file changed, 4 insertions(+)
```

- Now move to the master branch.
- Git cherry-pick <commit ID>[] this will take the mentioned commit ID stage and merge to the master branch.
- Main advantage of using cherry pick is we can pick the required snapshot from the branches and add to the master branch.

```
MINGW64:/D/Git
-----
Hello This is first line of master code
-----
Hi This is 2nd line of master code
-----
Hello This is 1st line of FB code
-----
Hi This is 2nd line of FB code and stash code
-----
Source-branch is created
~
```

- Here we can see the content of the file.txt file at that snapshot is added to the master.

9. Analyzing 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?

Step1:

```
ACER@Kumar MINGW32 ~ (master)
$ git show 60b5560e3275e1a53e0e281ab34edb/42/625ed1
commit 60b5560e3275e1a53e0e281ab34edb7427625ed1 (HEAD -> master)
Author: Abhishek-4949 <abhishekhiremath4949@gmail.com>
Date:   Mon Feb 26 18:17:25 2024 +0530

    file created

diff --git a/file.txt b/file.txt
index 9445a90..a94ee49 100644
--- a/file.txt
+++ b/file.txt
@@ -15,3 +15,7 @@ Hello This is 1st line of FB code
-----
Hi This is 2nd line of FB code and stash code
+
+
+-----+
+Source-branch is created
```

- To view the details of the specific commit including author, date and commit message we should copy the specific commit which you want to view in detail.
- git show <commit ID> [] this will show the full detail of the commit ID mentioned ,added changes will be shown in green colour and deleted changes will be shown in red colour.

10. Analysing and Changing Git History

- Write the command to list all commits made by the author "JohnDoe" between "2024-01-27" and "2023-01-28."

```
ACER@Kumar MINGW32 ~ (master)
$ git log --author="Sakshi" --since="2024-02-09" --until="2024-02-11"
commit 4c23379aa3389cd8cd071512038120780b805ea (tag: v1.0)
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:50:31 2024 +0530

    FB code & Stash code

commit 4e7384168d0b6746e0e8621d1d19f36b53231a4e (tag: v1.1)
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:29:03 2024 +0530

    1st line of FB(v1)

commit 4971195ca5c642240bd461b45cb5db5fef75f076
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:18:14 2024 +0530

    2 lines are added(v2)

commit 990b32135d54f774a873d7db2bf520ae7285617b
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:13:19 2024 +0530

    File created(V1)
```

- git log --author="Sakshi" --since="2024-02-09" --until="2024-02-11" □ this will show all the commits made by the author "Sakshi" b/w dated "2024-02-09" and "2024-02-11"

11. Analyzing and Changing Git History

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

```
ACER@Kumar MINGW32 ~ (master)
$ git log -5
commit 60b5560e3275e1a53e0e281ab34edb7427625ed1 (HEAD -> master)
Author: Abhishek-4949 <abhishekhiremath4949@gmail.com>
Date:   Mon Feb 26 18:17:25 2024 +0530

    file created

commit 4c23379aa3389cd8cd071512038120780b805ea (tag: v1.0)
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:50:31 2024 +0530

    FB code & Stash code

commit 4e7384168d0b6746e0e8621d1d19f36b53231a4e (tag: v1.1)
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:29:03 2024 +0530

    1st line of FB(v1)

commit 4971195ca5c642240bd461b45cb5db5fef75f076
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:18:14 2024 +0530

    2 lines are added(v2)

commit 990b32135d54f774a873d7db2bf520ae7285617b
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:13:19 2024 +0530

    File created(V1)
```

- git log -n □ this will display last n no.of commits. Here n is 5.



12. Analyzing and Changing Git History

- Write the command to undo the changes introduced by the commit with

Step 1:

```
git log --oneline  
Hello This is first line of master code  
-----  
Hi This is 2nd line of master code  
-----  
Hello This is 1st line of FB code  
-----  
Hi This is 2nd line of FB code and stash code  
-----  
Source-branch is created
```

- The above image is before reverting

```
ACER@Kumar MINGW32 ~ (master)  
$ git revert 60b5560e3275e1a53e0e281ab34edb7427625ed1  
[master 8f3882a] Revert "file created"  
1 file changed, 4 deletions(-)
```

- git revert <commit ID> this will revert to the that stage of commit

```
MINGW64:/D/Git  
-----  
Hello This is first line of master code  
-----  
Hi This is 2nd line of master code  
-----  
Hello This is 1st line of FB code  
-----  
Hi This is 2nd line of FB code and stash code  
~  
~
```

- The above image is after reverting

```
ACER@Kumar MINGW32 ~ (master)
$ git log
commit 8f3882ad1810d0a9ba17fb279f267a4c8694cbb6 (HEAD -> master)
Author: Abhishek-4949 <abhishekhiremath4949@gmail.com>
Date:   Mon Feb 26 20:03:20 2024 +0530

    Revert "file created"

    This reverts commit 60b5560e3275e1a53e0e281ab34edb7427625ed1.

commit 60b5560e3275e1a53e0e281ab34edb7427625ed1
Author: Abhishek-4949 <abhishekhiremath4949@gmail.com>
Date:   Mon Feb 26 18:17:25 2024 +0530

    file created

commit 4c23379aa3389cdd8cd071512038120780b805ea (tag: v1.0)
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:50:31 2024 +0530

    FB code & Stash code

commit 4e7384168d0b6746e0e8621d1d19f36b53231a4e (tag: v1.1)
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:29:03 2024 +0530

    1st line of FB(v1)

commit 4971195ca5c642240bd461b45cb5db5fef75f076
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:18:14 2024 +0530

    2 lines are added(V2)

commit 990b32135d54f774a873d7db2bf520ae7285617b
Author: Sakshi <sakshiburali@gmail.com>
Date:   Sat Feb 10 19:13:19 2024 +0530

    File created(V1)
```