



**ANJUMAN HAMI-E-MUSLIMEEN'S  
ANJUMAN INSTITUTE OF TECHNOLOGY & MANAGEMENT  
ANJUMANABAD, BHATKAL - 581 320**

**INDEX**

SI.No.	Date	DESCRIPTION	Page No.	Marks	Signature
1	18/12/2024	Selling up and Basic Commands	1	10	
2	18/12/2024	creating and managing branches	2	10	
3	18/12/2024	creating and managing Branches	3	09	
4	18/12/2024	collaboration and Remote Repositories	4	10	
5	18/12/2024	collaboration and Remote Repositories	5	10	
6	18/12/2024	collaboration and Remote Repositories	6	10	
7	18/12/2024	Git Tags and Releases	7	09	
8	18/12/2024	Advanced Git Operations	8	10	
9	18/12/2024	Analyzing and changing Git History	9	10	
10	18/12/2024	Analyzing and changing Git History	10	10	



**ANJUMAN HAMI-E-MUSLIMEEN'S  
ANJUMAN INSTITUTE OF TECHNOLOGY & MANAGEMENT  
ANJUMANABAD, BHATKAL - 581 320**

## INDEX

Sl.No.	Date	DESCRIPTION	Page No.	Marks	Signature
11	18/12/2024	Analyzing and changing Grt History	11	10	SL
12	18/12/2024	Analyzing and changing Grt History	12	10	SL



TITLE

Setting up and Basic Commands.

Step 1: Initialise a new Git repository.

mkdir your-project-directory.

cd your-project-directory.

git init

Step 2: Create a new file.

echo this is a new file > your-new-file.txt

Step 3: Add the file to the staging area:

git add your-new-file.txt

git add.

Step 4: Commit the changes with a message.

git commit -m "Initial commit - adding a new file"



TITLE

Creating and managing Branches.

Step 1 : create a new branch named "feature-branch".

git branch feature-branch

git checkout -b feature-branch.

Step 2 : Switch to the "master" branch.

git checkout master.

Step 3 : merge "feature-branch" into "master".

git merge feature-branch

git checkout master

git merge feature-branch.

Step 4 : Resolve any conflicts and commit the merge.

git add

git commit -m "merge feature-branch into master"

git branch -d feature-branch.

git branch -D feature-branch



TITLE

Creating and managing Branches

Step 1. stash your changes:

git stash save "your stash message"

Step 2. Switch to another branch:

git checkout your-desired-branch

Step 3. Apply the stashed changes.

git stash apply

git stash apply stash@{1}

Step 4. Remove the applied stash

git stash drop

git stash drop stash@{1}

git stash clear

git stash pop

git stash pop stash@{1}



TITLE

Collaboration and Remote Repositories

- clone a remote Git repository to your local machine.

git clone <repository\_url>

git clone https://github.com/user/repository.git

For SSH:

git clone git@github.com:example/repo.git.



TITLE

## Collaboration and Remote Repositories

Step 1. fetch the latest changes from the remote repository:

git fetch

Step 2. Rebase your local branch onto the updated remote branch:

git rebase origin/your-branch

git rebase -i origin/your-branch

Step 3. Continue the rebase or resolve conflicts:

git rebase --continue

git rebase --abort

Step 4. push the rebased branch to the remote repository:

git push origin your-branch --force

git push origin your-branch --force-with-leas



TITLE

Collaboration and Remote Repositories

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

git merge feature-branch

git commit --amend -m "your custom commit - message for the merge"

git merge --continue

git merge feature-branch --no-ff > e



ANJUMAN INSTITUTE OF TECHNOLOGY  
& MANAGEMENT  
ANJUMANABAD, BHATKAL - 581 320

PAGE NO. : 87  
DATE : 18/12/2024  
EXP. NO. : 7

TITLE

Git Tags and Releases

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

git tag v1.0 <commit-hash>

git tag v1.0 HEAD

git push origin v1.0



TITLE
<u>Advanced Git Operations</u>

- write the command to cherry-pick a range of commits from "Source-branch" to the current branch.

git cherry-pick <start-commit>^..<end-commit>  
git cherry-pick A^..B  
git cherry-pick --continue  
git cherry-pick --abort



ANJUMAN INSTITUTE OF TECHNOLOGY  
& MANAGEMENT  
ANJUMANABAD, BHATKAL - 581 320

PAGE NO. : ..... 9 .....  
DATE : ..... 18/12/2024 .....  
EXP. NO. : ..... 9 .....

TITLE

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?

git show <commit-id>

git show abc123

git log -n 1 --pretty=format:"%h - %an, %ar : %s" <commit-id>



ANJUMAN INSTITUTE OF TECHNOLOGY  
& MANAGEMENT  
ANJUMANABAD, BHATKAL - 581 320

PAGE NO. .... 10  
DATE : 18/12/2024  
EXP. NO. : 10

TITLE

Analyzing and changing Git History

- write the command to list all commits made by the author "John Doe" b/w "2023-01-01" and "2023 - 12 - 31."

git log --author = "John Doe" --since = "2023-01-01"  
--until = "2023 - 12 - 31"

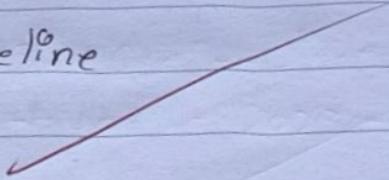


TITLE

Analyzing and changing Git History

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

git log -n 5  
git log -n 5 --oneline





ANJUMAN INSTITUTE OF TECHNOLOGY  
& MANAGEMENT  
ANJUMANABAD, BHATKAL - 581 320

PAGE NO. .... 12  
DATE : 18/12/2024  
EXP. NO. .... 12

TITLE

Analyzing and changing Git History

- write the command to undo the changes introduced by the commit with the ID "abc123".

git revert abc123

git reset --hard abc123

~~Suman  
10/12/2021~~