ವಿಶ್ವೇಶ್ವರಯ್ಯ ತಾಂತ್ರಿಕ ವಿಶ್ವವಿದ್ಯಾಲಯ, ಬೆಳಗಾವಿ

VISVESVARAYA TECHNOLOGICAL UNIVERSITY - BELAGAVI

Department of Computer Science and Engineering

"Jnana Sangama", VTU-Campus, Belagavi-590018

LAB-MANUAL

Academic Year: 2024-25

Name : HEMANTH B K

USN : 2VX23CS012

Sem : III Semester

Subject : Project Management with Git

Code : BCS358C

Course : B.tech

Programme : Computer Science and Engineering

Certificate

This is certify that Mr./~~Mrs.~~ HEMANTH B K with USN 2VX23CS012 has satisfactorily completed all the Laboratory Assignment of Subject Project Management with Git having Subject Code BCS358C during the academic year 2024-25.

Faculty in-charge **Programmer Co-ordinator**

Signature of the Examiners

INDEX

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 the appropriate commit message.

Step 1:

```
File  Edit  View  Search  Terminal  Help
vtu@vtu-Vostro-3888:~$ mkdir Hi
vtu@vtu-Vostro-3888:~$ cd Hi
```

- cd c; ->changes the current working directory into c disk
- mkdir Hi -> creates a folder/directory in the present working directory.
- Cd Hi -> changes the directory to the git folder which was created.

Step 2:

```
vtu@vtu-Vostro-3888:~/Hi$ git init
Initialized empty Git repository in /home/vtu/Hi/.git/
```

- git init -> initializes a empty git repository.
- We can see the .git folder created in thegit folder, in some cases the file is hidden and to see the hidden file we need to click on view the hidden files.

Step 3:

```
vtu@vtu-Vostro-3888:~/Hi$ touch dot.txt
vtu@vtu-Vostro-3888:~/Hi$ git add .
vtu@vtu-Vostro-3888:~/Hi$ git commit -m "file created"
[master (root-commit) 6f8a313] file created
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 dot.txt
```

- touch text.txt -> creates a empty file pf txt extension in the current directory.
- git add ./git add text.txt-> stages the file in the case of specific file add ,or add . will stage the whole files in the current directory and are ready to the commited.

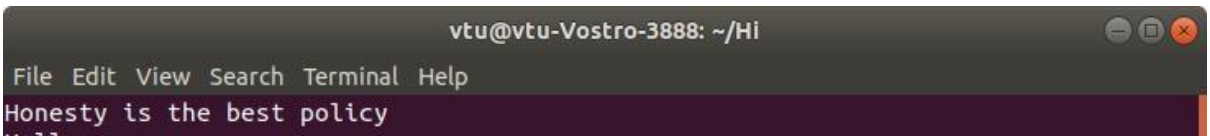
Step 4:

```
vtu@vtu-Vostro-3888:~/Hi$ git log
commit 6f8a3131becf98dd480190b5dfe3e593685ec1d9 (HEAD -> master)
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:02:09 2024 +0530

    file created
```

- git log -> displays all the history of commits with commit message along with the author name and email.
- Every commits has unique commit ID.

Step 5:

```
vtu@vtu-Vostro-3888:~/Hi$ vi pk.txt
vtu@vtu-Vostro-3888:~/Hi$ git add .
vtu@vtu-Vostro-3888:~/Hi$ git commit -m "added a file"
[master b21e83d] added a file
 1 file changed, 1 insertion(+)
 create mode 100644 pk.txt
```



- Here the file content is added using vi text.txt and then file is staged and commited with appropriate message.

Step 6:

```
vtu@vtu-Vostro-3888:~/Hi$ git status
On branch master
nothing to commit, working tree clean
```

- git status -> it checks the status, like on which branchwe are and is there any changes made which are not commited.
- If no any other changes has been made after recent commit the it displays the working tree is clean.

Step 7:

```
vtu@vtu-Vostro-3888:~/Hi$ git log
commit b21e83d601ed5207dd50848882707b4a2efec320 (HEAD -> master)
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:03:04 2024 +0530

    added a file

commit 6f8a3131becf98dd480190b5dfe3e593685ec1d9
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:02:09 2024 +0530

    file created
```

- git log -> this will display the history of the commits.

2)Creating and Managing Branches:

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

Step 1:

```
vtu@vtu-Vostro-3888:~/Hi$ git branch new
vtu@vtu-Vostro-3888:~/Hi$ git branch
* master
  new
```

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

Step 2:

```
vtu@vtu-Vostro-3888:~/Hi$ git checkout new
Switched to branch 'new'
vtu@vtu-Vostro-3888:~/Hi$ vi pk.txt
```

- git checkout -> it is used to switch one branch to other branch ,here we are moving from branch master to the feature-branch.
- And also we have edited the file text.txt.

Step 3:

```
File  Edit  View  Search  Terminal  Help
Switched to branch 'new'
vtu@vtu-Vostro-3888:~/Hi$ vi pk.txt
vtu@vtu-Vostro-3888:~/Hi$ git add .
vtu@vtu-Vostro-3888:~/Hi$ git commit -m "edited a file"
[new 7a8a0c0] edited a file
 1 file changed, 1 insertion(+)
```

- Here in the feature-branch we staged the file and commited with message saying “edited file in feature branch”.
- 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:

```
vtu@vtu-Vostro-3888:~/Hi$ git checkout master
Switched to branch 'master'
vtu@vtu-Vostro-3888:~/Hi$ git merge new
Updating b21e83d..7a8a0c0
Fast-forward
 pk.txt | 1 +
 1 file changed, 1 insertion(+)
```

- 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:

```
vtu@vtu-Vostro-3888:~/Hi$ git log
commit 7a8a0c0c823d013b2e8f3d6bb2a2e010d2283659 (HEAD -> master, new)
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:04:37 2024 +0530

    edited a file

commit b21e83d601ed5207dd50848882707b4a2efec320
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:03:04 2024 +0530

    added a file

commit 6f8a3131becf98dd480190b5dfe3e593685ec1d9
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:02:09 2024 +0530

    file created
```

- 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:

```
vtu@vtu-Vostro-3888:~/Hi$ git checkout new
Switched to branch 'new'
vtu@vtu-Vostro-3888:~/Hi$ vi pk.txt
```

- Moving to the feature branch and have made changes in the text.txt file using the commands git checkout feature-branch and vi text.txt for changing the branch and editing the file respectively.
- Here we have not staged and commited the changes in the text.txt file.

Step 2:

```
vtu@vtu-Vostro-3888:~/Hi$ git stash
Saved working directory and index state WIP on new: 7a8a0c0 edited a file
```

- In this step we have stashed the changes which have been made in the text.txt file in the feature-branch.
- Here we have not added/staged the file and commite dthe changes.
- The changes will be saved in the branch without the commiting the changes.
- The commited used fir stashing the changes ->git stash

Step 3:

```
vtu@vtu-Vostro-3888:~/Hi$ git stash apply
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   pk.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 moved to the master branch.
- Then, in the master branch we applied the stashed changes made in the feature-branch.
- After applying the stash to the master. It will give us a message saying the applied stash is not staged and committed in the master branch.

Step 4:

```
vtu@vtu-Vostro-3888:~/Hi$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   pk.txt
```

```
vtu@vtu-Vostro-3888:~/Hi$ git add .
vtu@vtu-Vostro-3888:~/Hi$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   pk.txt

vtu@vtu-Vostro-3888:~/Hi$ git commit -m "stash changed"
[master c6dbc82] stash changed
 1 file changed, 1 insertion(+)
```

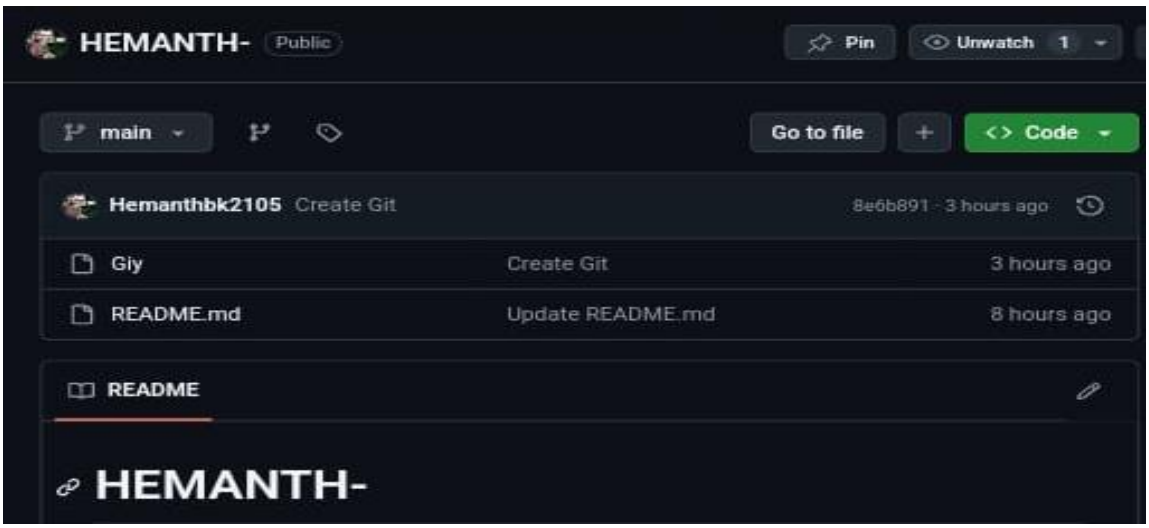
```
File  Edit  View  Search  Terminal  Help
vtu@vtu-Vostro-3888:~/Hi$ git log
commit c6dbc8201d12e15d33d0bde99525329770bf310c (HEAD -> master)
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:08:57 2024 +0530

    stash changed

commit 7a8a0c0c823d013b2e8f3d6bb2a2e010d2283659 (new)
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:04:37 2024 +0530

    edited a file

commit b21e83d601ed5207dd50848882707b4a2efec320
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:03:04 2024 +0530

    added a file

commit 6f8a3131becf98dd480190b5dfe3e593685ec1d9
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:02:09 2024 +0530

    file created
```

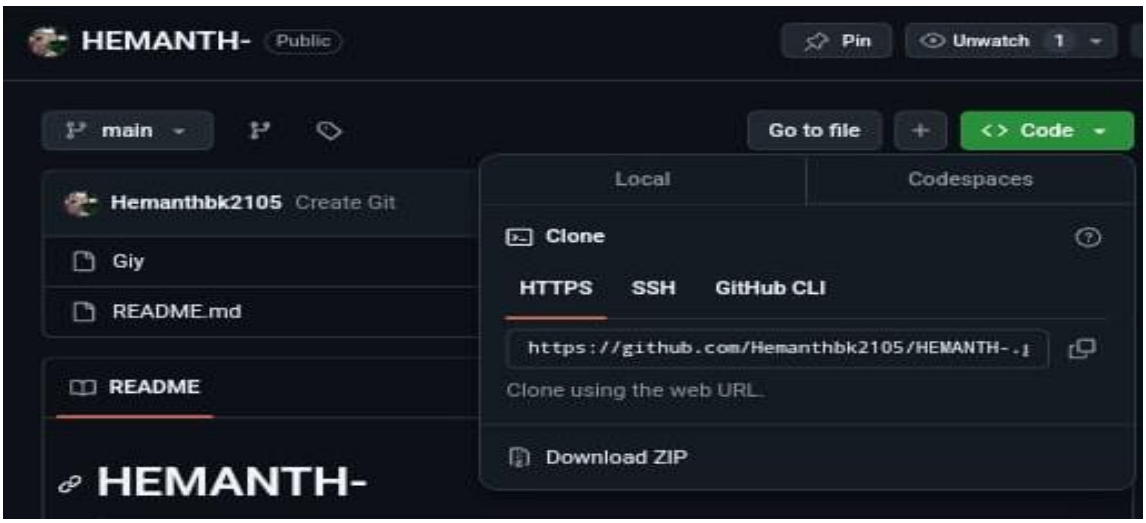
- After stashing we can see the status ,here it showed that the file is modified but not yet commited.
- If we watch git status after adding on the stage but not commiting , then it will “say to be commited.”
- After commiting the changes we can see the log of the repository.

4)Collabration and Remote Repositories :

* Clone a remote Git repositoryto your local machine.

Step 1:

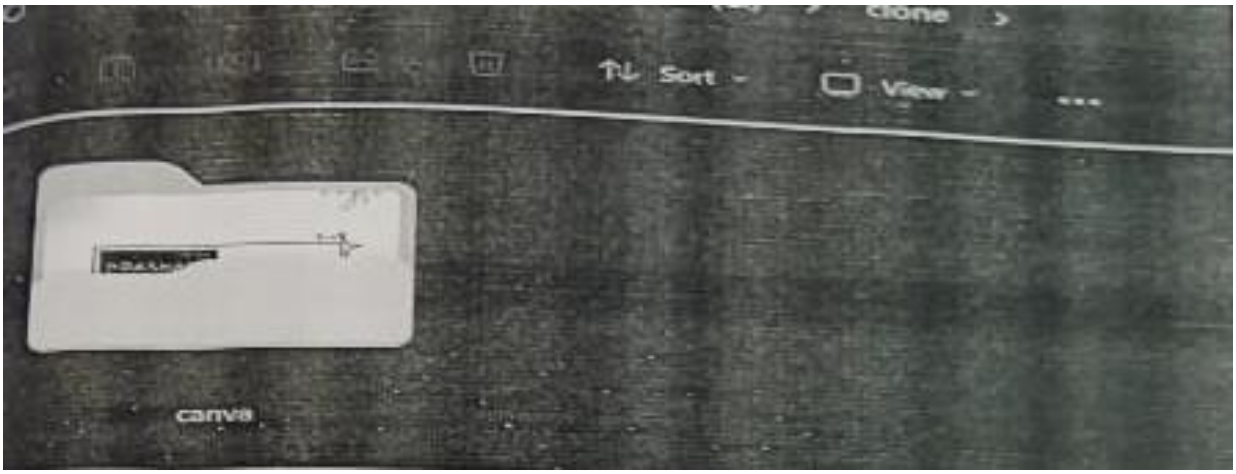




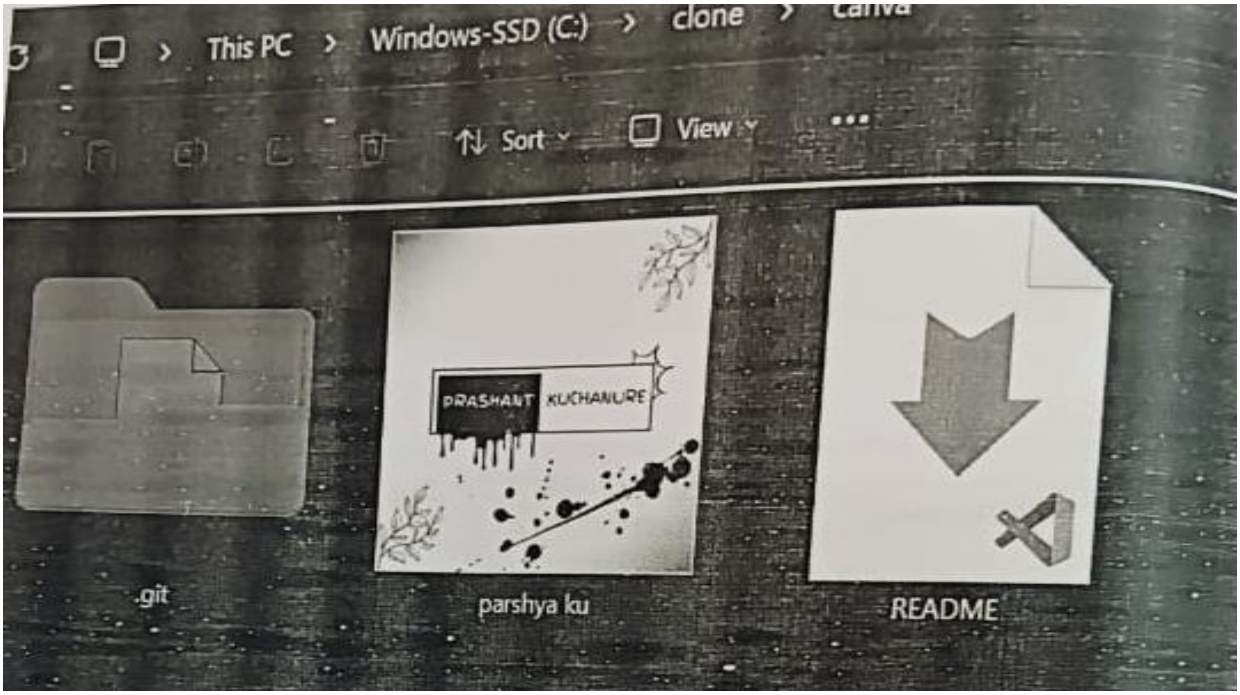
- To clone a remote repository ,first we need to open the github.com and open any of the account on the github.com
- After that we have chosen the repository which we want to clone into our local machine
- After choosing the repository , in that repository we clicked on the green button “code” ,which open a dropdown lists links in that , we copied the “HTTPS” link from that.
- https://github.com/Hemanthbk2105/HEMANTH-.git

Step 2:

```
vtu@vtu-Vostro-3888:~/Hi$ mkdir clone
vtu@vtu-Vostro-3888:~/Hi$ cd clone
vtu@vtu-Vostro-3888:~/Hi/clone$ git clone https://github.com/Hemanthbk2105/HEMAN
TH-.git
Cloning into 'HEMANTH-'...
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 (from 0)
Unpacking objects: 100% (6/6), done.
```



- In the second step we need to open our git bash in some directory where you need clone the remote repository we need to move to that location using “cd <path>” command.
- Here we want to copy the repository to the folder clone in the c so we moved to that location
- git clone https://github.com/Hemanthbk2105/HEMANTH-.git -> this command will copy the repository from remote to the local machine in the working directory
- you can see above the “flipkart_clone” repository is successfully copied in the clone directory/folder.



- We know that whole repository have been cloned to the local machine so the files and “png” files can be seen in that repo.

5)Collabration and Remote Repositories :

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

Step 1:

```
vtu@vtu-Vostro-3888:~/Hi/clone$ cd HEMANTH-
```

- 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 logging the commits.

Step 2:

- Move to the remote repo and make some changes/add new file and commit it.

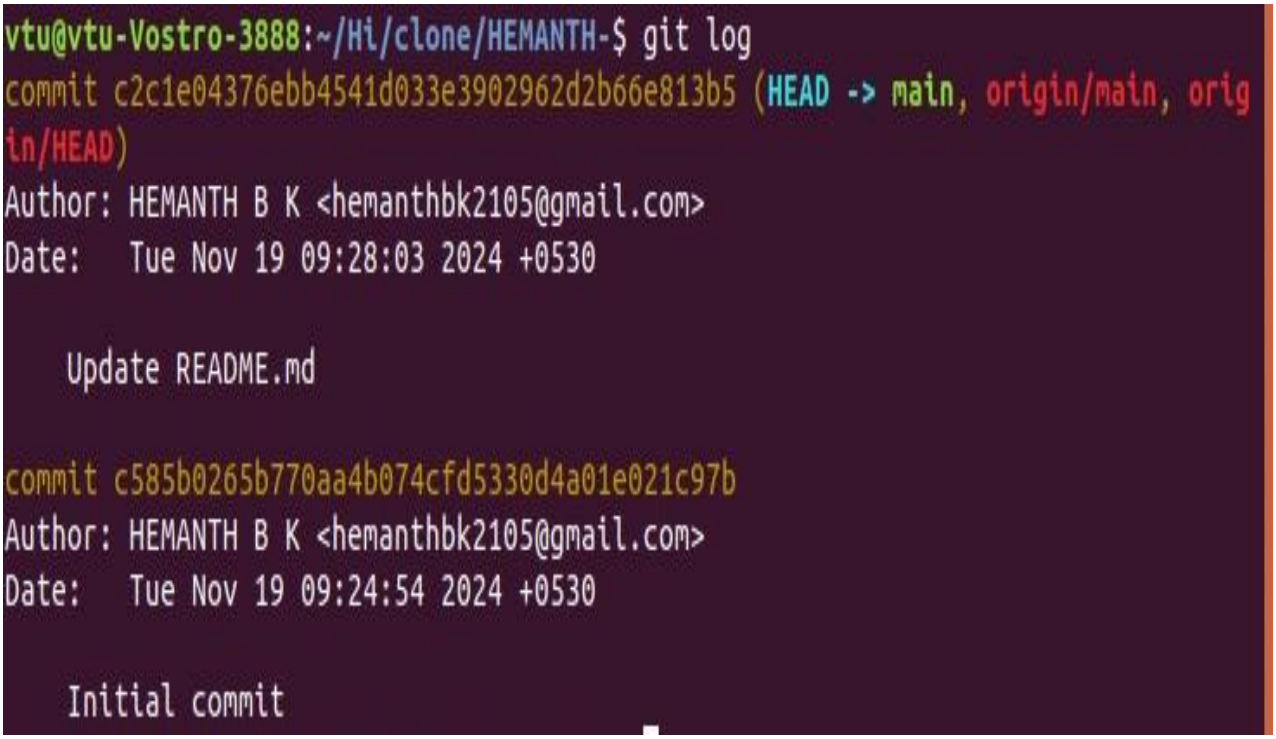
Step 3:

```
vtu@vtu-Vostro-3888:~/Hi/clone/HEMANTH-$ git fetch origin
```

- git fetch origin -> 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 4:

```
vtu@vtu-Vostro-3888:~/Hi/clone/HEMANTH-$ git rebase origin
```



- 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.

6)Collabration and Remote Repositories :

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

Step 1:

```
vtu@vtu-Vostro-3888:~/Hi$ git checkout new
Switched to branch 'new'
vtu@vtu-Vostro-3888:~/Hi$ vi pk.txt
vtu@vtu-Vostro-3888:~/Hi$ git add .
```

```
vtu@vtu-Vostro-3888:~/Hi$ git commit -m "edited a file"
[new d73b32b] edited a file
 2 files changed, 3 insertions(+)
 create mode 160000 clone/HEMANTH-
```

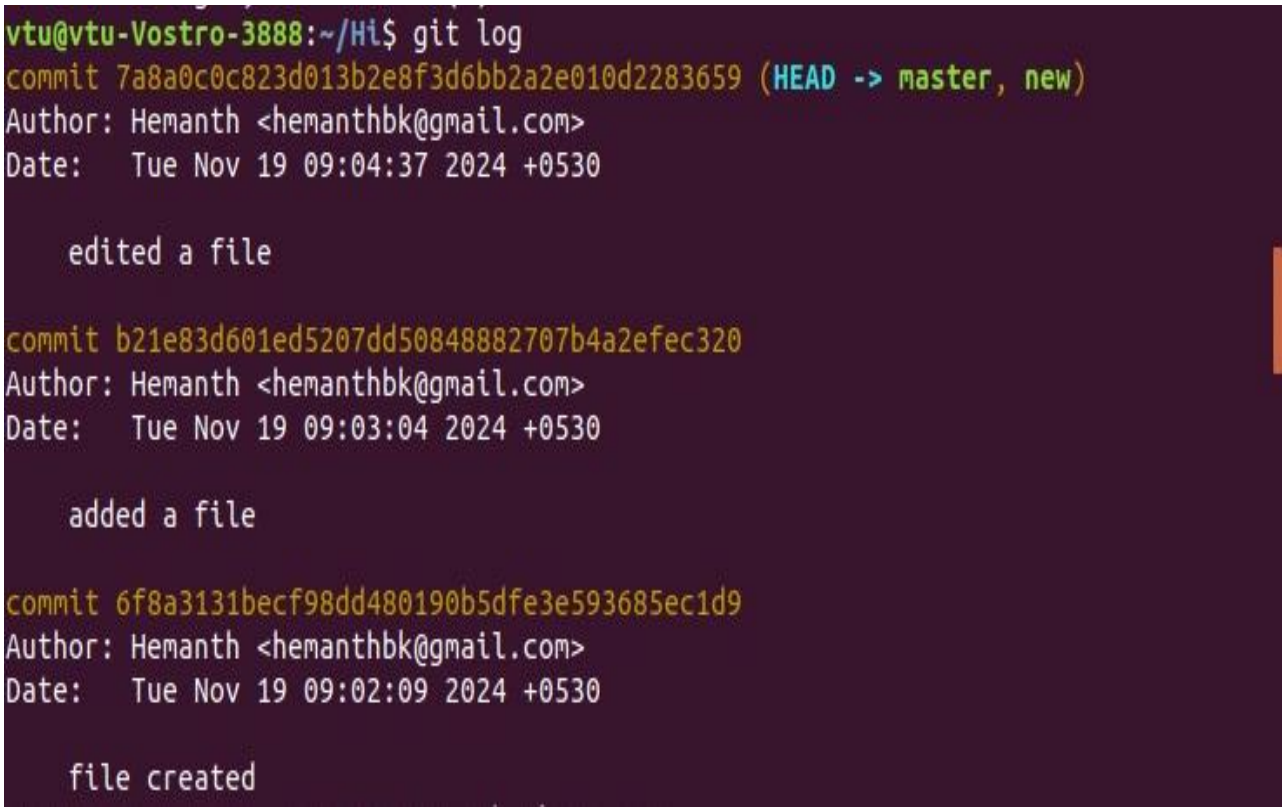
- Move to the feature branch and make some changes in the that branch ,stage and commit the changes.
- For committing we can use additional/optional -> -m”message” ,will commit the state with appropriate message.

Step 2:

```
vtu@vtu-Vostro-3888: ~/Hi
File  Edit  View  Search  Terminal  Help
vtu@vtu-Vostro-3888:~/Hi$ git checkout master
warning: unable to rmdir 'clone/HEMANTH-': Directory not empty
Switched to branch 'master'
vtu@vtu-Vostro-3888:~/Hi$ git merge new
Auto-merging pk.txt
CONFLICT (content): Merge conflict in pk.txt
```

- Then checkout to the master branch and merge the branch

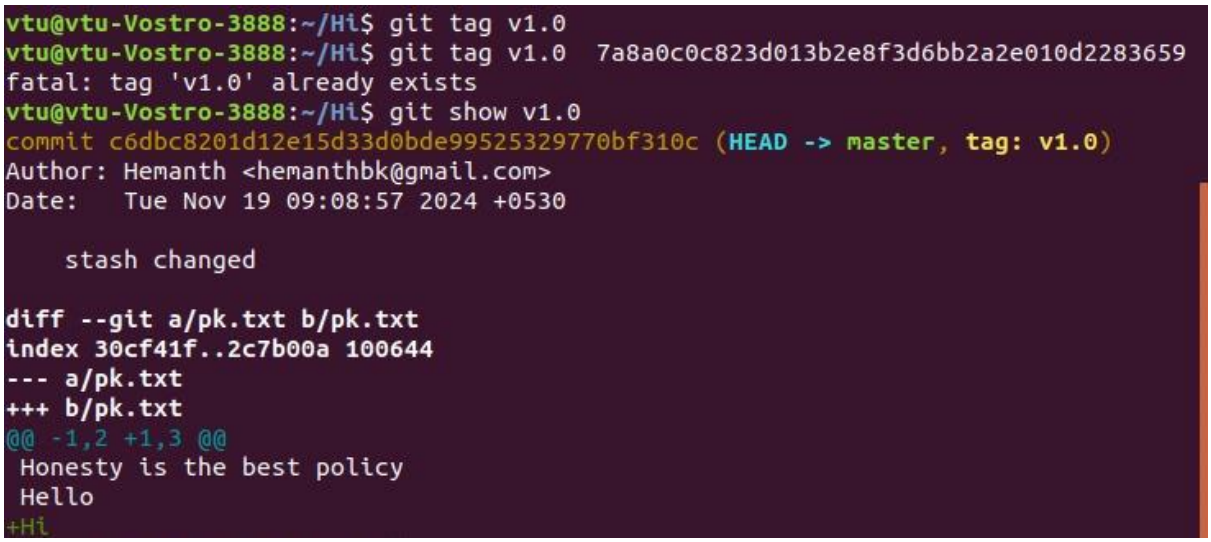
Step 3:


vtu@vtu-Vostro-3888:~/Hi$ git log
commit 7a8a0c0c823d013b2e8f3d6bb2a2e010d2283659 (HEAD -> master, new)
Author: Hemanth <hemanthbk@gmail.com>
Date: Tue Nov 19 09:04:37 2024 +0530
edited a file
commit b21e83d601ed5207dd50848882707b4a2efec320
Author: Hemanth <hemanthbk@gmail.com>
Date: Tue Nov 19 09:03:04 2024 +0530
added a file
commit 6f8a3131becf98dd480190b5dfe3e593685ec1d9
Author: Hemanth <hemanthbk@gmail.com>
Date: Tue Nov 19 09:02:09 2024 +0530
file created

7)Git Tags and Releases :

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

Step 1:



- 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 the all tags made i.e, v1.0 created.
- git show v1.0 -> this will show details in that tag(v1.0) with full description.

8)Advanced Git Operations :

- write the command to cherry-pick a range of commits from “source-branch” to the current branch.

Step 1:

```
vtu@vtu-Vostro-3888:~/Hi$ git branch
* master
  new
  source-branch
vtu@vtu-Vostro-3888:~/Hi$ git checkout source-branch
pk.txt: needs merge
error: you need to resolve your current index first
vtu@vtu-Vostro-3888:~/Hi$ vi pk.txt
```

```
Honesty is the best policy
Hello
<<<<<<< HEAD
Hi
=======
Try not to become a man of success, but rather try to become a man of value.
```

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

Step 2:

```
vtu@vtu-Vostro-3888:~/Hi$ git add .
vtu@vtu-Vostro-3888:~/Hi$ git commit -m "this is edited in source branch"
[master 0eba498] this is edited in source branch
```

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

Step 3:

```
vtu@vtu-Vostro-3888:~/Hi$ vi pk.txt
```

```
Honesty is the best policy
Hello
<<<<<<< HEAD
Hi
=======
Try not to become a man of success, but rather try to become a man of value.

>>>>>>> new
You must be the change you wish to see in the world.
~
```

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

Step 4:

```
vtu@vtu-Vostro-3888:~/Hi$ git add .
vtu@vtu-Vostro-3888:~/Hi$ git commit -m "this is edited in source branch"
[master 9b28ac1] this is edited in source branch
 1 file changed, 2 insertions(+), 1 deletion(-)
```

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

Step 5:

```
commit 9b28ac191edbca4e705fb1c94c4b4cbe1a42da80 (HEAD -> master)
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:55:14 2024 +0530

    this is edited in source branch

commit 0eba4986930aa04e26d2959cf93f072fde145c1f
Merge: c6dbc82 d73b32b
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:54:15 2024 +0530

    this is edited in source branch

commit d73b32bf25d31713c1dd42cf4f30e35313bb845e (new)
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:45:18 2024 +0530

    edited a file
```

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

Step 6:

```
vtu@vtu-Vostro-3888:~/Hi$ git checkout master
```

```
vtu@vtu-Vostro-3888:~/Hi$ git cherry-pick 9b28ac191edbca4e705fb1c94c4b4cbe1a42da
80
On branch master
You are currently cherry-picking commit 9b28ac1.
```

- Now move to the master branch.
- Git cheery-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.

```
Honesty is the best policy
Hello
<<<<<<< HEAD
Hi
=======
Try not to become a man of success, but rather try to become a man of value.
```

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

9)Analysing 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 ?

Step 1:

```
vtu@vtu-Vostro-3888:~/Hi$ git show d73b32bf25d31713c1dd42cf4f30e35313bb845e
commit d73b32bf25d31713c1dd42cf4f30e35313bb845e (new)
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:45:18 2024 +0530

    edited a file

diff --git a/clone/HEMANTH- b/clone/HEMANTH-
new file mode 160000
index 0000000..482d6fd
--- /dev/null
+++ b/clone/HEMANTH-
@@ -0,0 +1 @@
+Subproject commit 482d6fd3f48418bb7b6f2cbea3be1a802ca85b40
diff --git a/pk.txt b/pk.txt
index 30cf41f..b9dab10 100644
--- a/pk.txt
+++ b/pk.txt
@@ -1,2 +1,4 @@
 Honesty is the best policy
 Hello
```

- 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 reed colour.

10)Analysing and Changing Git History:

Step 1:

- Write the command to list all commits made by the author “JohnDoe” between “2024-11-17” and “2024-11-19.”

```
vtu@vtu-Vostro-3888:~/Hi$ git log --author="Hemanth" --since"2024-11-17"  --unti
l="2024-11-19"
```

```
commit c6dbc8201d12e15d33d0bde99525329770bf310c (HEAD -> master)
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:08:57 2024 +0530

    stash changed

commit 7a8a0c0c823d013b2e8f3d6bb2a2e010d2283659
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:04:37 2024 +0530

    edited a file

commit b21e83d601ed5207dd50848882707b4a2efec320
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:03:04 2024 +0530

    added a file

commit 6f8a3131becf98dd480190b5dfe3e593685ec1d9
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:02:09 2024 +0530
```

- git log –author =”Hemanth” –since=”2024-11-17” –until—“2024-11-19” ->this will show all the commits made by the author “Hemanth” b/w dated “2024-11-17” an d “2024-11-19”.

11)Analysing and Changing Git History:

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

Step 1:

```
vtu@vtu-Vostro-3888:~/Hi$ git log -s
commit 9b28ac191edbca4e705fb1c94c4b4cbe1a42da80 (HEAD -> master)
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:55:14 2024 +0530

    this is edited in source branch

commit 0eba4986930aa04e26d2959cf93f072fde145c1f
Merge: c6dbc82 d73b32b
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:54:15 2024 +0530

    this is edited in source branch

commit d73b32bf25d31713c1dd42cf4f30e35313bb845e (new)
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:45:18 2024 +0530

    edited a file
```

```
commit c6dbc8201d12e15d33d0bde99525329770bf310c (HEAD -> master)
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:08:57 2024 +0530

    stash changed

commit 7a8a0c0c823d013b2e8f3d6bb2a2e010d2283659 (new)
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:04:37 2024 +0530

    edited a file
```

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

12)Analysing and Changing Git History:

- Write the command to undothe changes introduced by the commit with the ID “abc123” .

Step 1:

```
Honesty is the best policy
Hello
<<<<<<< HEAD
Hi
=======
Try not to become a man of success, but rather try to become a man of value.
```

- The above image is before reverting

```
vtu@vtu-Vostro-3888:~/Hi$ git revert 9b28ac191edbca4e705fb1c94c4b4cbe1a42da80
Use "fg" to return to nano.or to close the file...

[1]+  Stopped                 git revert 9b28ac191edbca4e705fb1c94c4b4cbe1a42da8
0
```

- Git revert<commit ID> -> this will revert to the that stage of commit
- In case of failed of auto conflict ,conflict will arise and we should solve conflict.

```
Honesty is the best policy
Hello
<<<<<<< HEAD
Hi
```

- The above image is after reverting

```
vtu@vtu-Vostro-3888:~/Hi$ git log
commit 9b28ac191edbca4e705fb1c94c4b4cbe1a42da80 (HEAD -> master)
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:55:14 2024 +0530

    this is edited in source branch

commit 0eba4986930aa04e26d2959cf93f072fde145c1f
Merge: c6dbc82 d73b32b
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:54:15 2024 +0530

    this is edited in source branch

commit d73b32bf25d31713c1dd42cf4f30e35313bb845e (new)
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:45:18 2024 +0530

    edited a file

commit 9b28ac191edbca4e705fb1c94c4b4cbe1a42da80 (HEAD -> master)
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:55:14 2024 +0530

    this is edited in source branch

commit 0eba4986930aa04e26d2959cf93f072fde145c1f
Merge: c6dbc82 d73b32b
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:54:15 2024 +0530

    this is edited in source branch

commit d73b32bf25d31713c1dd42cf4f30e35313bb845e (new)
Author: Hemanth <hemanthbk@gmail.com>
Date:   Tue Nov 19 09:45:18 2024 +0530
```