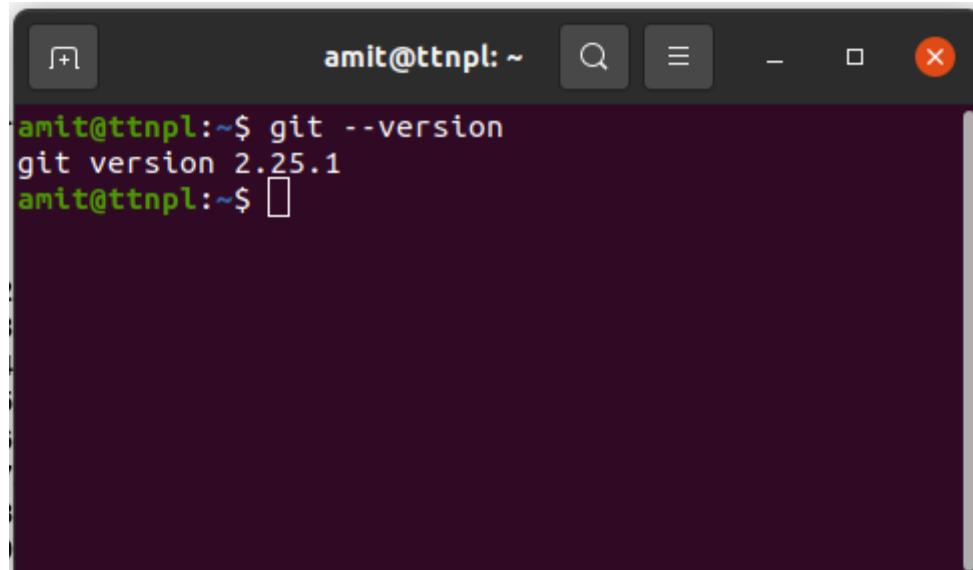
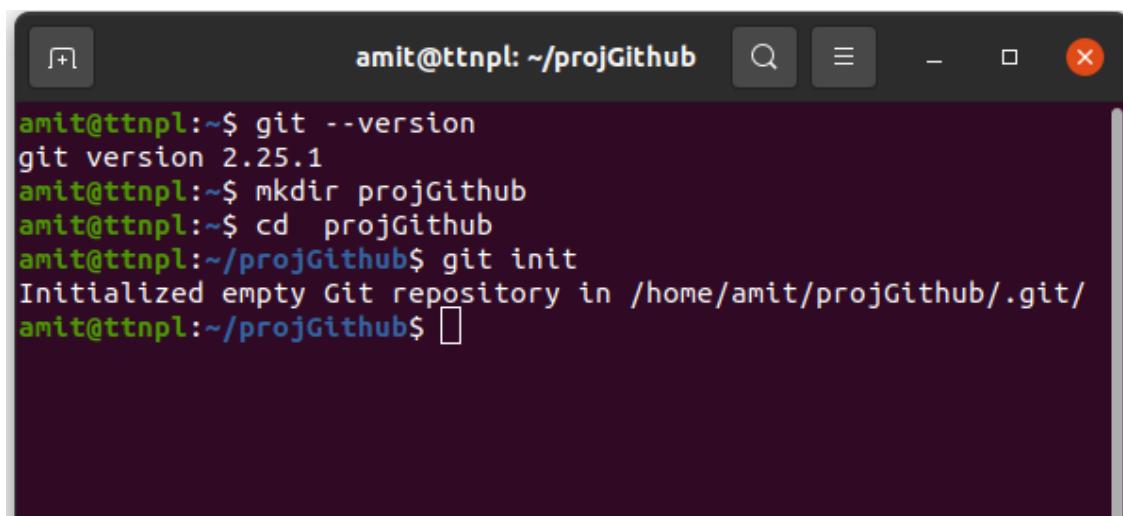


1. Git Setup <https://confluence.atlassian.com/bitbucket/set-up-git-744723531.html>



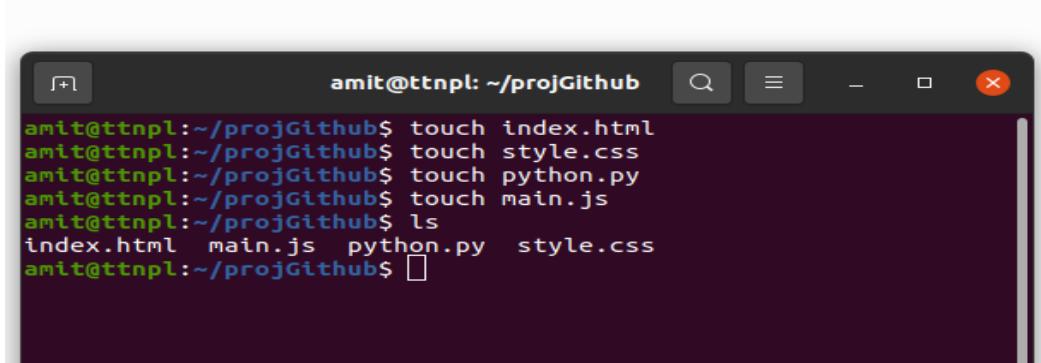
```
amit@ttnpl:~$ git --version
git version 2.25.1
amit@ttnpl:~$ 
```

2. Initialize a Git Repository



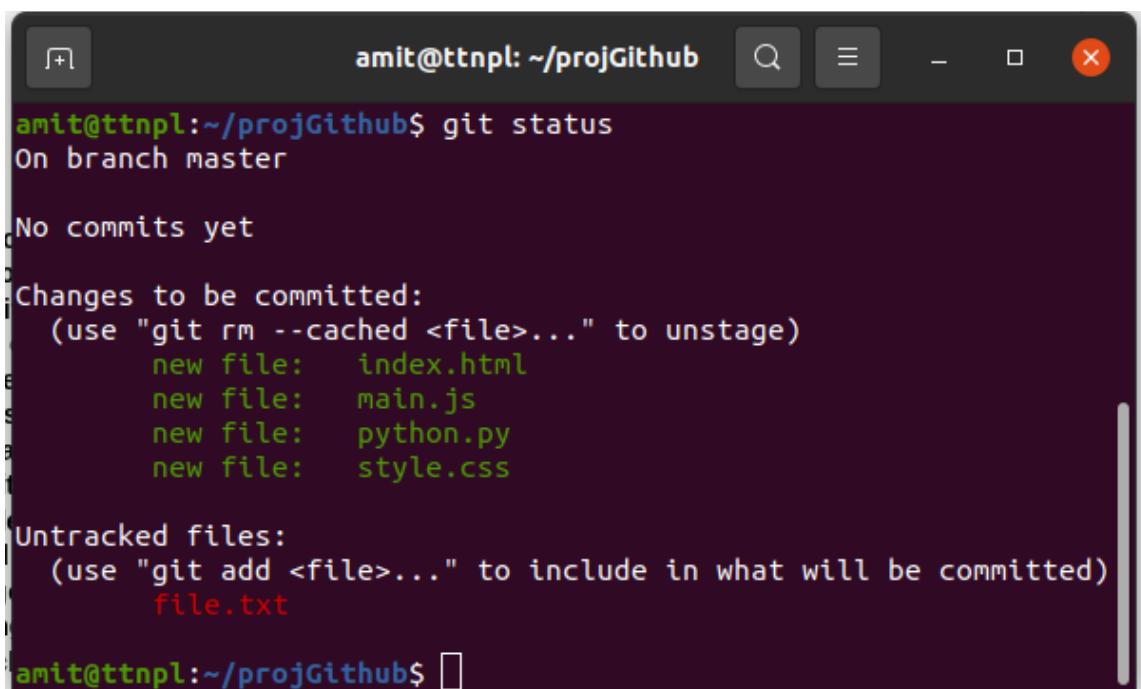
```
amit@ttnpl:~$ git --version
git version 2.25.1
amit@ttnpl:~$ mkdir projGithub
amit@ttnpl:~$ cd projGithub
amit@ttnpl:~/projGithub$ git init
Initialized empty Git repository in /home/amit/projGithub/.git/
amit@ttnpl:~/projGithub$ 
```

### 3. Add files to the repository



```
amit@ttnpl:~/projGithub$ touch index.html
amit@ttnpl:~/projGithub$ touch style.css
amit@ttnpl:~/projGithub$ touch python.py
amit@ttnpl:~/projGithub$ touch main.js
amit@ttnpl:~/projGithub$ ls
index.html  main.js  python.py  style.css
amit@ttnpl:~/projGithub$ 
```

### 4. Unstage 1 file



```
amit@ttnpl:~/projGithub$ git status
On branch master

No commits yet

Changes to be committed:
(use "git rm --cached <file>..." to unstage)
  new file:   index.html
  new file:   main.js
  new file:   python.py
  new file:   style.css

Untracked files:
(use "git add <file>..." to include in what will be committed)
  file.txt

amit@ttnpl:~/projGithub$ 
```

## 5. Commit the file

```
amit@ttnpl: ~/projGithub
new file: index.html
new file: main.js
new file: python.py
new file: style.css

Untracked files:
(use "git add <file>..." to include in what will be committed)
  file.txt

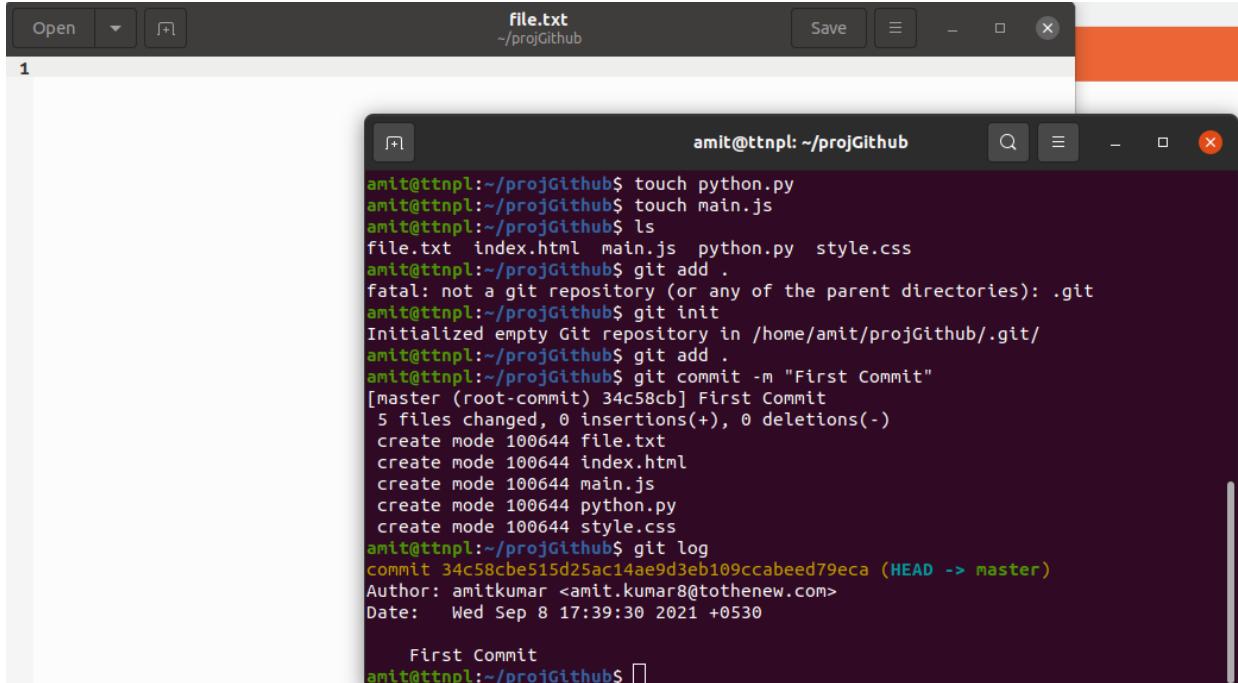
amit@ttnpl:~/projGithub$ git commit -m "Initial commit"
[master (root-commit) 936d534] Initial commit
 4 files changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 index.html
  create mode 100644 main.js
  create mode 100644 python.py
  create mode 100644 style.css
amit@ttnpl:~/projGithub$
```

## 6. Add a remote

```
amit@ttnpl:~/projGithub
amit@ttnpl:~/projGithub$ git remote add origin https://github.com/Amitkumarttn/projGithub.git
amit@ttnpl:~/projGithub$ git remote -v
origin  https://github.com/Amitkumarttn/projGithub.git (fetch)
origin  https://github.com/Amitkumarttn/projGithub.git (push)
amit@ttnpl:~/projGithub$
```

## 7. Undo changes to a particular file

### Initial stage

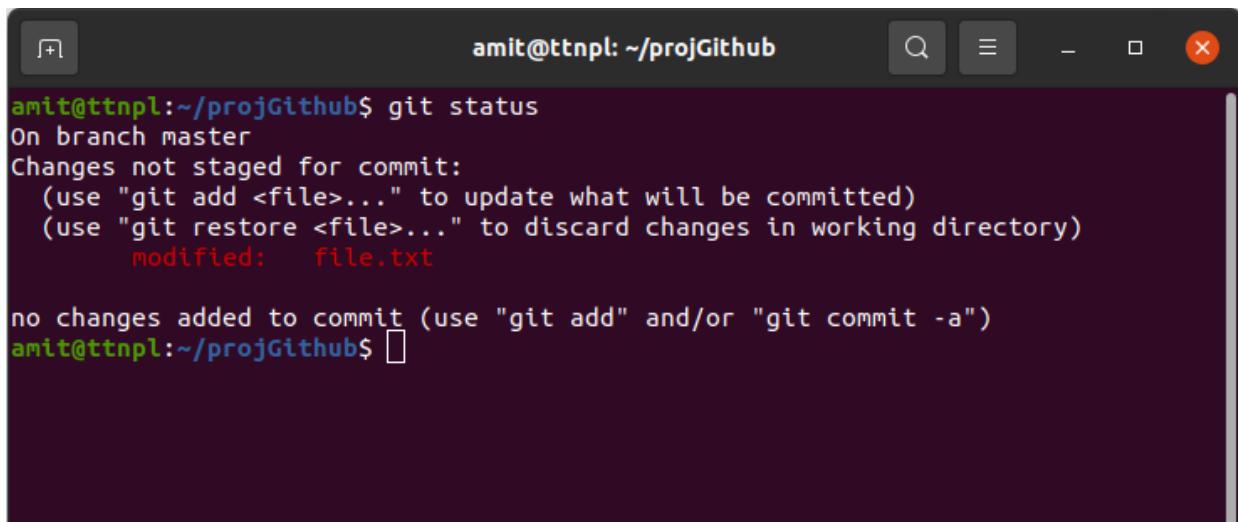


A screenshot of a terminal window titled "file.txt" in a code editor. The terminal shows the following command history:

```
amit@ttnpl:~/projGithub$ touch python.py
amit@ttnpl:~/projGithub$ touch main.js
amit@ttnpl:~/projGithub$ ls
file.txt index.html main.js python.py style.css
amit@ttnpl:~/projGithub$ git add .
fatal: not a git repository (or any of the parent directories): .git
amit@ttnpl:~/projGithub$ git init
Initialized empty Git repository in /home/amit/projGithub/.git/
amit@ttnpl:~/projGithub$ git add .
amit@ttnpl:~/projGithub$ git commit -m "First Commit"
[master (root-commit) 34c58cb] First Commit
 5 files changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 file.txt
  create mode 100644 index.html
  create mode 100644 main.js
  create mode 100644 python.py
  create mode 100644 style.css
amit@ttnpl:~/projGithub$ git log
commit 34c58cbe515d25ac14ae9d3eb109ccabed79eca (HEAD -> master)
Author: amitkumar <amit.kumar8@tothenew.com>
Date:   Wed Sep 8 17:39:30 2021 +0530

  First Commit
amit@ttnpl:~/projGithub$
```

### Git status



A screenshot of a terminal window titled "amit@ttnpl: ~/projGithub". The terminal shows the following "git status" output:

```
amit@ttnpl:~/projGithub$ 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:   file.txt

no changes added to commit (use "git add" and/or "git commit -a")
amit@ttnpl:~/projGithub$
```

## Adding text in file

The screenshot shows a terminal window titled "file.txt" with the path "~/projGitHub". The file contains the text "1 hello world!". Below the terminal window is a command-line interface showing the output of the "git status" command:

```
amit@ttnpl:~/projGitHub$ 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:   file.txt

no changes added to commit (use "git add" and/or "git commit -a")
amit@ttnpl:~/projGitHub$
```

## Undo the changes

The screenshot shows a terminal window titled "file.txt" with the path "~/projGitHub". The file now contains the number "1". Below the terminal window is a command-line interface showing the process of committing changes and then reverting them:

```
amit@ttnpl:~/projGitHub$ git add file.txt
amit@ttnpl:~/projGitHub$ git commit -m "Changes in file.txt"
[master 3bd1c03] Changes in file.txt
 1 file changed, 1 insertion(+)
amit@ttnpl:~/projGitHub$ git log
commit 3bd1c03fb90459cbccbf6834afc8e5ad67c49c0 (HEAD -> master)
Author: amitkumar <amit.kumar8@tothenew.com>
Date:   Wed Sep 8 17:42:27 2021 +0530

    Changes in file.txt

commit 34c58cbe515d25ac14ae9d3eb109ccabed79eca
Author: amitkumar <amit.kumar8@tothenew.com>
Date:   Wed Sep 8 17:39:30 2021 +0530

    First Commit
amit@ttnpl:~/projGitHub$ git revert 4c58cbe515d25ac14ae9d3eb109ccabed79eca
fatal: bad revision '4c58cbe515d25ac14ae9d3eb109ccabed79eca'
amit@ttnpl:~/projGitHub$ git revert 3bd1c03fb90459cbccbf6834afc8e5ad67c49c0
[master e1d3b17] Revert "Changes in file.txt"
 1 file changed, 1 deletion(-)
amit@ttnpl:~/projGitHub$
```

## 8. Push changes to Github

```
amit@tnpl: ~/projGithub
Username for 'https://github.com': amitkumarttn
Password for 'https://amitkumarttn@github.com':
To https://github.com/Amitkumarttn/projGithub.git
 ! [rejected]      main -> main (fetch first)
error: failed to push some refs to 'https://github.com/Amitkumarttn/projGithub.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
amit@tnpl:~/projGithub$ git push -u origin main --force
Username for 'https://github.com': amitkumarttn
Password for 'https://amitkumarttn@github.com':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 610 bytes | 610.00 KiB/s, done.
Total 7 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/Amitkumarttn/projGithub.git
 + d1ffcae...e1d3b17 main -> main (forced update)
Branch 'main' set up to track remote branch 'main' from 'origin'.
amit@tnpl:~/projGithub$ 
```

The screenshot shows a GitHub repository page for 'Amitkumarttn / projGithub'. The 'Code' tab is selected. At the top, it shows 'main' branch, 1 branch, 0 tags, and buttons for 'Go to file', 'Add file', and 'Code'. Below is a table of commits:

Author	Commit Message	Date	Commits
Amitkumarttn	Revert "Changes in file.txt"	e1d3b17 1 hour ago	3 commits
	file.txt	Revert "Changes in file.txt"	1 hour ago
	index.html	<a href="#">First Commit</a>	1 hour ago
	main.js	First Commit	1 hour ago
	python.py	First Commit	1 hour ago
	style.css	First Commit	1 hour ago

## 9. Clone the repository



The screenshot shows a terminal window with a dark background and light-colored text. The title bar indicates the session is running on a machine named 'ttnpl' with the user 'amit' at the path '~/.projGithub'. The window has standard Linux-style window controls (minimize, maximize, close) in the top right corner. The terminal content displays the output of a 'git clone' command:

```
amit@ttnpl:~/projGithub$ git clone https://github.com/AmitkumarTN/projGithub.git
Cloning into 'projGithub'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 7 (delta 2), reused 7 (delta 2), pack-reused 0
Unpacking objects: 100% (7/7), 590 bytes | 295.00 KiB/s, done.
amit@ttnpl:~/projGithub$
```

## 10. Add changes to one of the copies and pull the changes in the other.

```
amit@ttnpl: ~/projGithubCopy
remotes/origin/main
amit@ttnpl:~/projGithubCopy$ git branch dev
amit@ttnpl:~/projGithubCopy$ git branch -a
  dev
* main
  remotes/origin/HEAD -> origin/main
  remotes/origin/main
amit@ttnpl:~/projGithubCopy$ git checkout dev
Switched to branch 'dev'
amit@ttnpl:~/projGithubCopy$ touch version.txt
amit@ttnpl:~/projGithubCopy$ echo '0.1' >> version.txt
amit@ttnpl:~/projGithubCopy$ git status
On branch dev
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    version.txt

nothing added to commit but untracked files present (use "git add" to track)
amit@ttnpl:~/projGithubCopy$ git add .
amit@ttnpl:~/projGithubCopy$ git commit -m "version file added"
[dev b90bad9] version file added
 1 file changed, 1 insertion(+)
 create mode 100644 version.txt
amit@ttnpl:~/projGithubCopy$
```

```
amit@ttnpl: ~/projGithubCopy
nothing to commit, working tree clean
amit@ttnpl:~/projGithubCopy$ git merge dev
Updating f4bd9d7..b90bad9
Fast-forward
  version.txt | 1 +
  1 file changed, 1 insertion(+)
  create mode 100644 version.txt
amit@ttnpl:~/projGithubCopy$ git log
commit b90bad9f74f7838bd00a05ebb63671aa013b9aaf (HEAD -> main, dev)
Author: amitkumar <amit.kumar8@tothenew.com>
Date:   Thu Sep 9 14:18:15 2021 +0530

  version file added

commit f4bd9d76368707964d9c6538063589c61fb45d7
Author: amitkumar <amit.kumar8@tothenew.com>
Date:   Thu Sep 9 14:10:54 2021 +0530

  Change in copy file

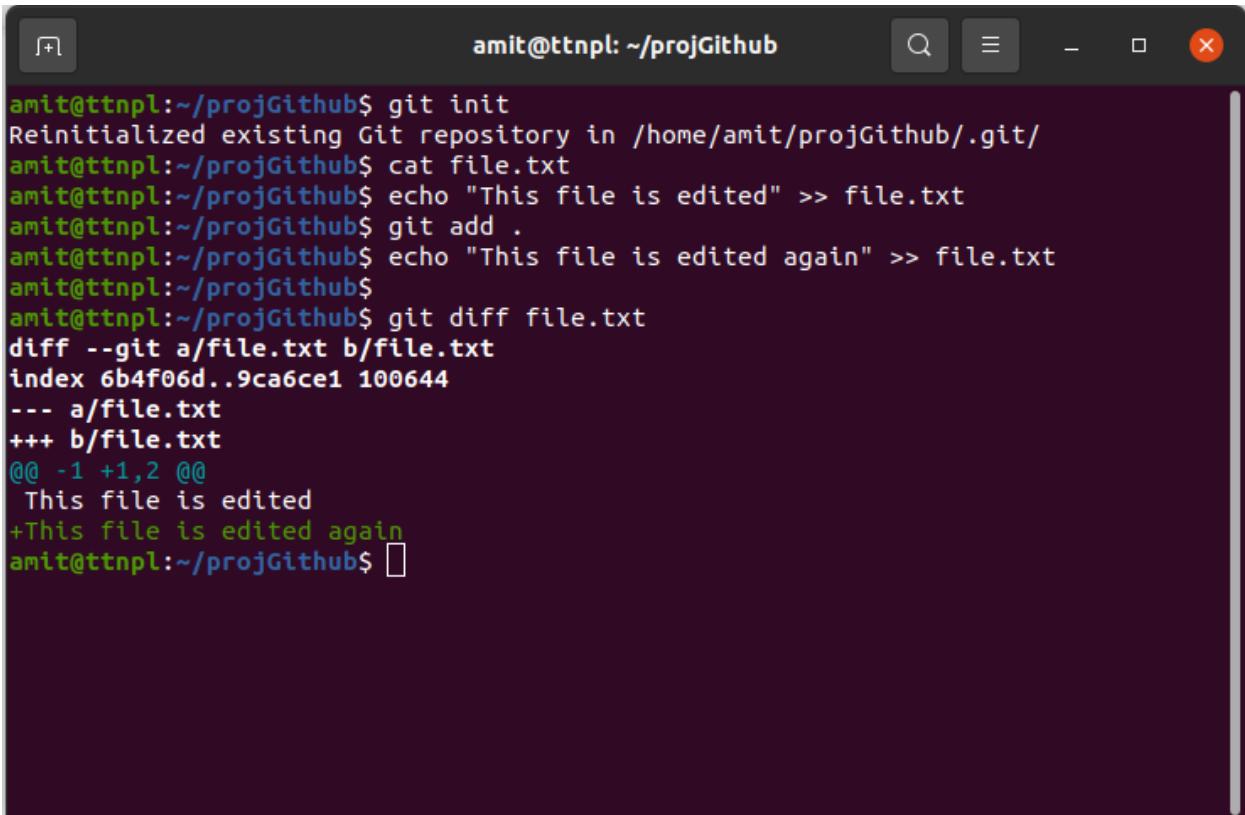
commit e1d3b17bd389381f3215d1bd76e06f7dca0d07ce (origin/main, origin/HEAD)
Author: amitkumar <amit.kumar8@tothenew.com>
Date:   Wed Sep 8 17:43:54 2021 +0530

  Revert "Changes in file.txt"

  This reverts commit 3bd1c03fb90459cbccbf6834afc8e5ad67c49c0.

commit 3bd1c03fb90459cbccbf6834afc8e5ad67c49c0
Author: amitkumar <amit.kumar8@tothenew.com>
Date:   Wed Sep 8 17:42:27 2021 +0530
:...skipping...
```

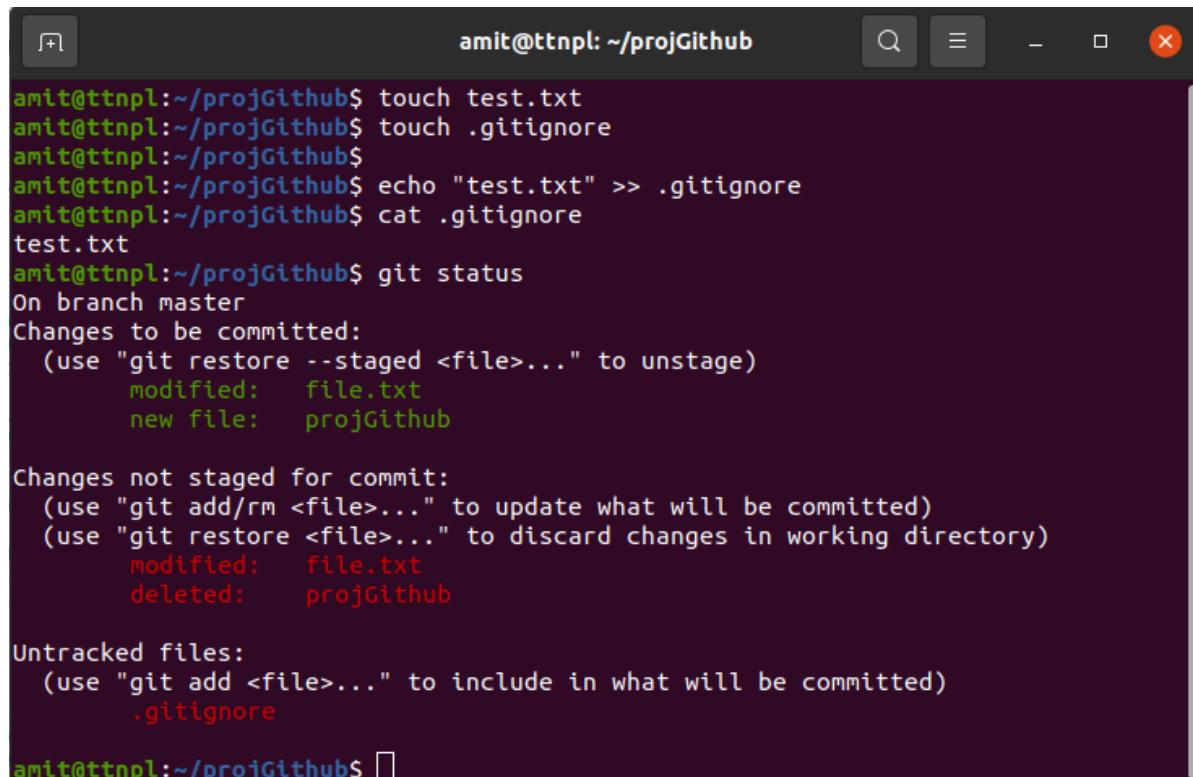
## 11. Check differences between a file and its staged version



The screenshot shows a terminal window with the following session:

```
amit@ttnpl:~/projGithub$ git init
Reinitialized existing Git repository in /home/amit/projGithub/.git/
amit@ttnpl:~/projGithub$ cat file.txt
amit@ttnpl:~/projGithub$ echo "This file is edited" >> file.txt
amit@ttnpl:~/projGithub$ git add .
amit@ttnpl:~/projGithub$ echo "This file is edited again" >> file.txt
amit@ttnpl:~/projGithub$ git diff file.txt
diff --git a/file.txt b/file.txt
index 6b4f06d..9ca6ce1 100644
--- a/file.txt
+++ b/file.txt
@@ -1 +1,2 @@
 This file is edited
+This file is edited again
amit@ttnpl:~/projGithub$
```

## 12. Ignore a few files to be checked in



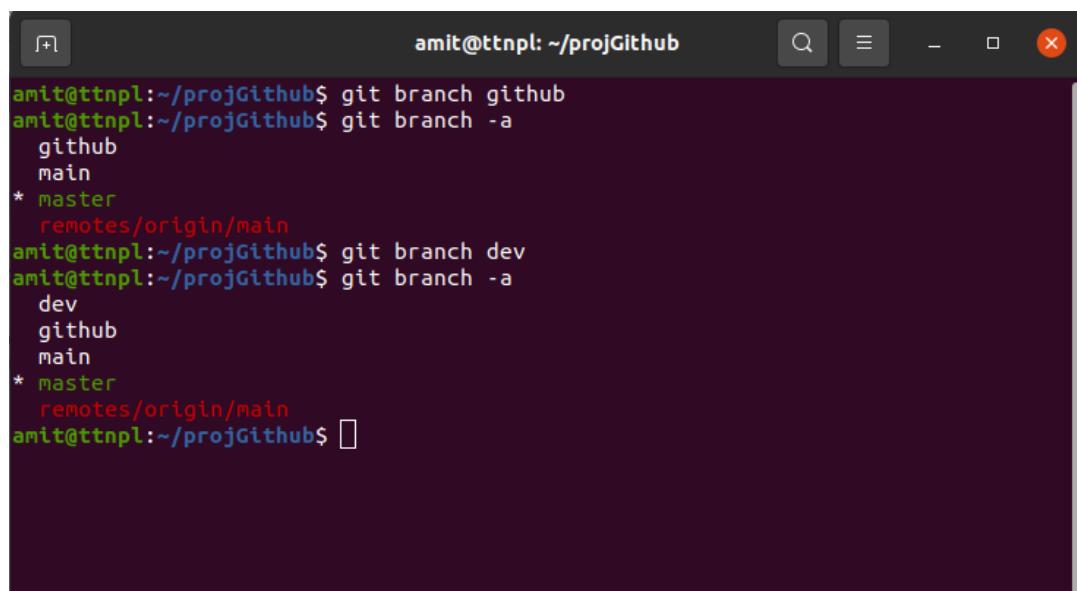
```
amit@ttnpl:~/projGithub$ touch test.txt
amit@ttnpl:~/projGithub$ touch .gitignore
amit@ttnpl:~/projGithub$ echo "test.txt" >> .gitignore
amit@ttnpl:~/projGithub$ cat .gitignore
test.txt
amit@ttnpl:~/projGithub$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   file.txt
    new file:   projGithub

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   file.txt
    deleted:   projGithub

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore

amit@ttnpl:~/projGithub$
```

## 13. Create a new branch.



```
amit@ttnpl:~/projGithub$ git branch github
amit@ttnpl:~/projGithub$ git branch -a
  github
  main
* master
  remotes/origin/main
amit@ttnpl:~/projGithub$ git branch dev
amit@ttnpl:~/projGithub$ git branch -a
  dev
  github
  main
* master
  remotes/origin/main
amit@ttnpl:~/projGithub$
```

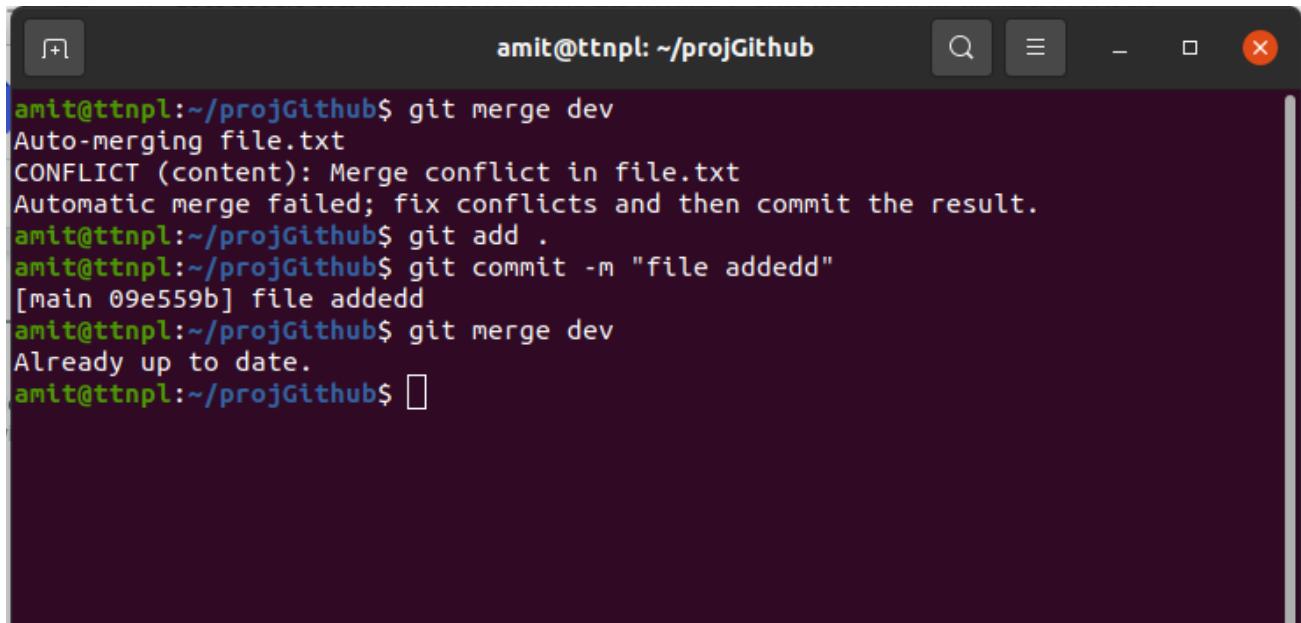
#### 14. Diverge them with commits

```
amit@ttnpl:~/projGithub$ git checkout dev
Switched to branch 'dev'
amit@ttnpl:~/projGithub$ touch amittn.txt
amit@ttnpl:~/projGithub$ 
amit@ttnpl:~/projGithub$ 
amit@ttnpl:~/projGithub$ git add .
amit@ttnpl:~/projGithub$ 
amit@ttnpl:~/projGithub$ 
amit@ttnpl:~/projGithub$ git commit -m "amittn file added"
[dev 48df58a] amittn file added
 2 files changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 amittn.txt
  create mode 100644 test.txt
amit@ttnpl:~/projGithub$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
amit@ttnpl:~/projGithub$ git merge dev
Updating e1d3b17..48df58a
Fast-forward
 amittn.txt | 0
 test.txt   | 0
 2 files changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 amittn.txt
  create mode 100644 test.txt
amit@ttnpl:~/projGithub$ 
```

#### 15. Edit the same file at the same line on both branches and commit

```
amit@ttnpl:~/projGithub$ git checkout dev
Switched to branch 'dev'
amit@ttnpl:~/projGithub$ echo "some change" >> file.txt
amit@ttnpl:~/projGithub$ git add .
amit@ttnpl:~/projGithub$ git commit -m "file.txt changes"
[dev 8095105] file.txt changes
 1 file changed, 1 insertion(+)
amit@ttnpl:~/projGithub$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
 (use "git push" to publish your local commits)
amit@ttnpl:~/projGithub$ echo "some change" >> file.txt
amit@ttnpl:~/projGithub$ git add .
amit@ttnpl:~/projGithub$ git commit -m "file.txt changes"
[main 38e88e9] file.txt changes
 1 file changed, 1 insertion(+)
amit@ttnpl:~/projGithub$ ls
amittn.txt  file.txt  index.html  main.js  python.py  style.css  test.txt
amit@ttnpl:~/projGithub$ git checkout dev
Switched to branch 'dev'
amit@ttnpl:~/projGithub$ ls
amittn.txt  file.txt  index.html  main.js  python.py  style.css  test.txt
amit@ttnpl:~/projGithub$ 
```

## 16. Try merging and resolve merge conflicts



The screenshot shows a terminal window with the following session:

```
amit@ttnpl:~/projGithub$ git merge dev
Auto-merging file.txt
CONFLICT (content): Merge conflict in file.txt
Automatic merge failed; fix conflicts and then commit the result.
amit@ttnpl:~/projGithub$ git add .
amit@ttnpl:~/projGithub$ git commit -m "file addeddd"
[main 09e559b] file addeddd
amit@ttnpl:~/projGithub$ git merge dev
Already up to date.
amit@ttnpl:~/projGithub$
```

## 17. Stash the changes and pop them

```
amit@ttnpl: ~/projGithub
amit@ttnpl:~/projGithub$ git status
On branch main
Your branch is ahead of 'origin/main' by 5 commits.
  (use "git push" to publish your local commits)

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")
amit@ttnpl:~/projGithub$ git stash
Saved working directory and index state WIP on main: 09e559b file addedd
amit@ttnpl:~/projGithub$ git status
On branch main
Your branch is ahead of 'origin/main' by 5 commits.
  (use "git push" to publish your local commits)

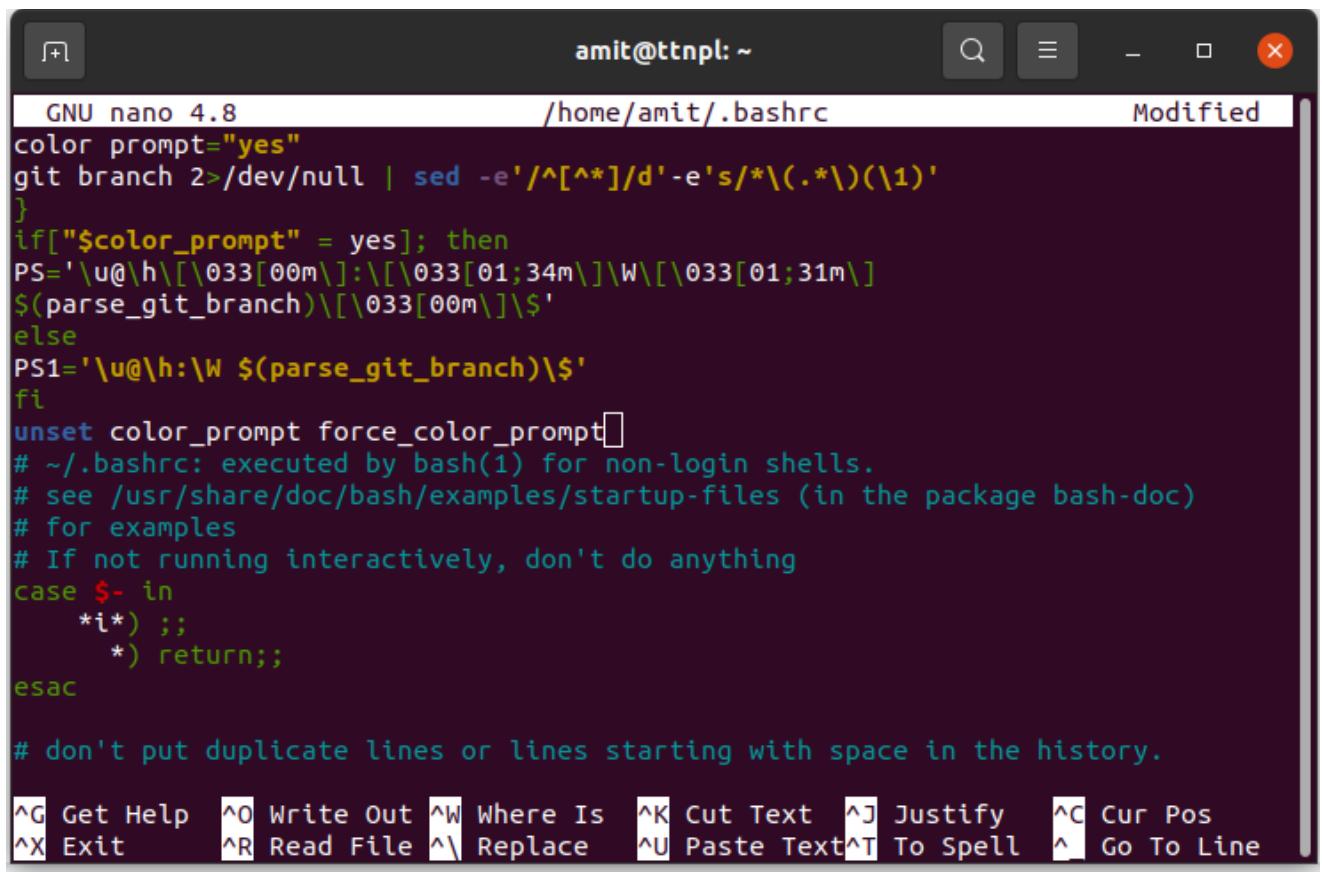
nothing to commit, working tree clean
amit@ttnpl:~/projGithub$ git stash pop
On branch main
Your branch is ahead of 'origin/main' by 5 commits.
  (use "git push" to publish your local commits)

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")
Dropped refs/stash@{0} (5895cbb68e832861feeb1acaac056e378dde64e7)
amit@ttnpl:~/projGithub$ git add .
amit@ttnpl:~/projGithub$ █
```

18. Add the following code to your .bashrc file : color\_prompt="yes"

```
parse_git_branch() {
    git branch 2>/dev/null | sed -e '/^[*]/d' -e 's/* \(.*)/(\1)/'
}
if [ "$color_prompt" = yes ]; then
    PS1='\u@\h[\033[00m]:\[\033[01;34m\]\W\[\033[01;31m\]
$(parse_git_branch)\[\033[00m\]\$ '
else
    PS1='\u@\h:\W $(parse_git_branch)\$ '
fi
unset color_prompt force_color_prompt
```



The screenshot shows a terminal window titled "amit@ttnpl: ~". The window contains the code for the .bashrc file. The code defines a function parse\_git\_branch(), which filters out branches starting with asterisks. It then checks if the environment variable color\_prompt is set to yes. If it is, it sets the PS1 variable to include a color-coded prompt with the current user, host, and working directory, along with the output of the parse\_git\_branch function. If color\_prompt is not set to yes, it uses a standard black-and-white prompt. Finally, it unsets the color\_prompt and force\_color\_prompt variables. The terminal window has a dark theme and includes a status bar at the bottom with various keyboard shortcuts.

```
GNU nano 4.8 /home/amit/.bashrc Modified
color_prompt="yes"
git branch 2>/dev/null | sed -e '/^[*]/d' -e 's/* \(.*)/(\1)'
}
if["$color_prompt" = yes]; then
PS='\u@\h[\033[00m]:\[\033[01;34m\]\W\[\033[01;31m\]
$(parse_git_branch)\[\033[00m\]\$ '
else
PS1='\u@\h:\W $(parse_git_branch)\$ '
fi
unset color_prompt force_color_prompt
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples
# If not running interactively, don't do anything
case $- in
  *i*) ;;
  *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Paste Text ^T To Spell ^_ Go To Line
```