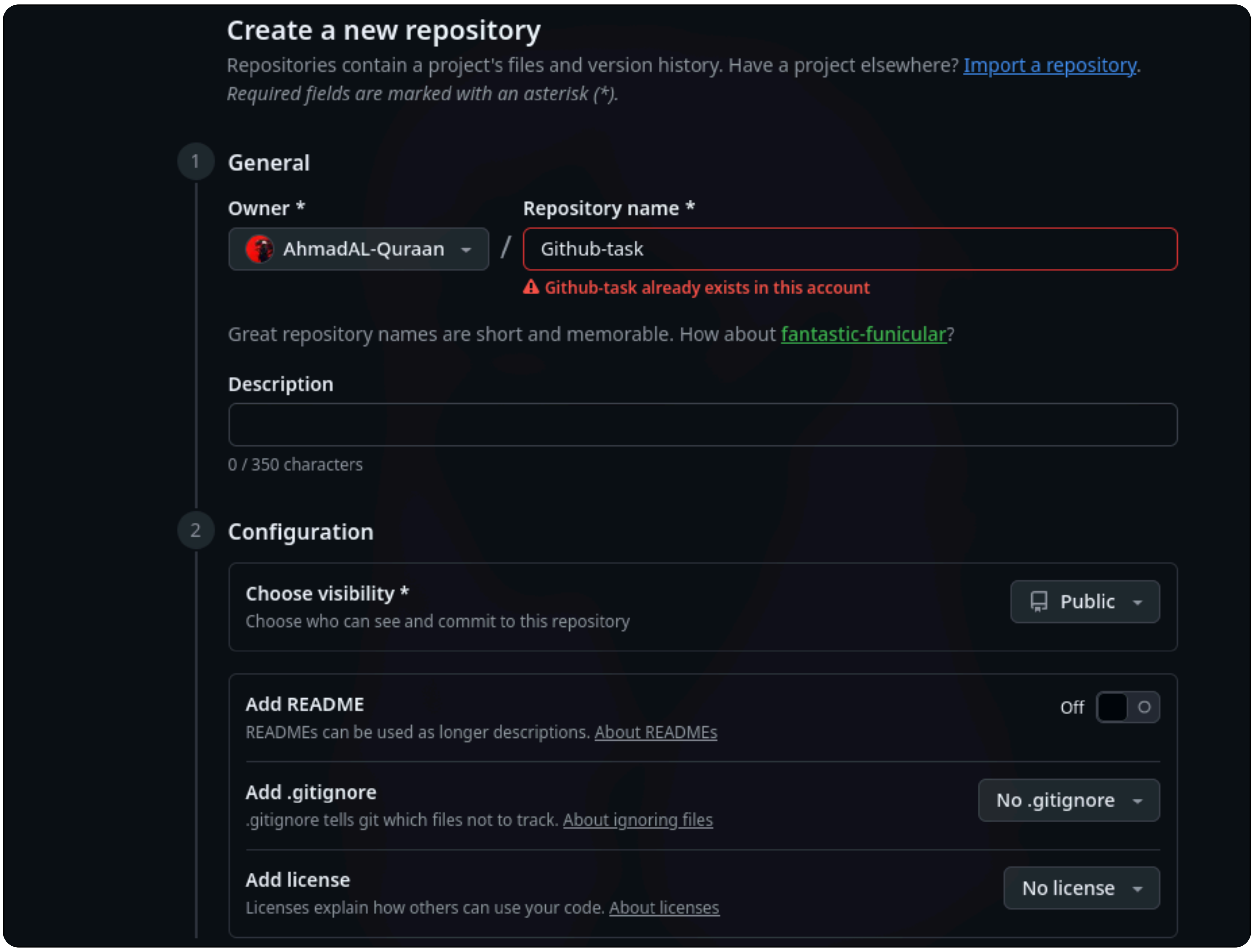
Solution

Phase 1: Project Setup

1. Create the repo



Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

1 General

Owner *: AhmadAL-Quraan / Repository name *: Github-task

⚠ Github-task already exists in this account

Great repository names are short and memorable. How about fantastic-funicular?

Description

0 / 350 characters

2 Configuration

Choose visibility *: Public

Choose who can see and commit to this repository

Add README: Off

READMEs can be used as longer descriptions. About READMEs

Add .gitignore: No .gitignore

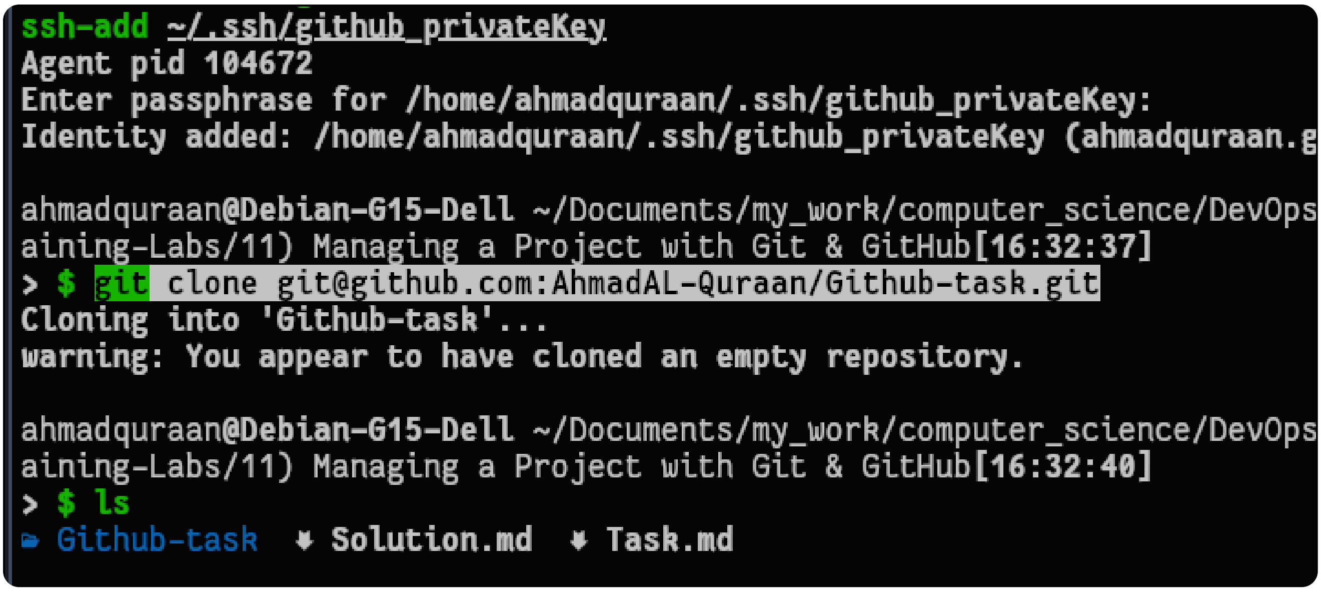
.gitignore tells git which files not to track. About ignoring files

Add license: No license

Licenses explain how others can use your code. About licenses

2. Copy the repository to your local machine.

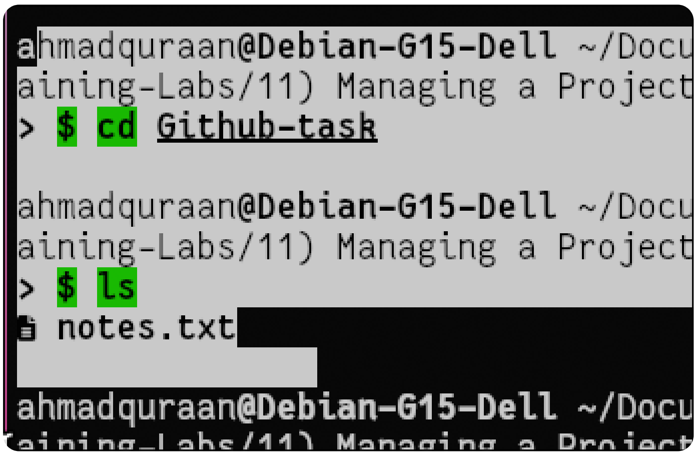
```
git clone git@github.com:AhmadAL-Quraan/Github-task.git
```



```
ssh-add ~/.ssh/github_privateKey
Agent pid 104672
Enter passphrase for /home/ahmadquraan/.ssh/github_privateKey:
Identity added: /home/ahmadquraan/.ssh/github_privateKey (ahmadquraan.g

ahmadquraan@Debian-G15-Dell ~/Documents/my_work/computer_science/DevOps
aining-Labs/11) Managing a Project with Git & GitHub[16:32:37]
> $ git clone git@github.com:AhmadAL-Quraan/Github-task.git
Cloning into 'Github-task'...
warning: You appear to have cloned an empty repository.

ahmadquraan@Debian-G15-Dell ~/Documents/my_work/computer_science/DevOps
aining-Labs/11) Managing a Project with Git & GitHub[16:32:40]
> $ ls
Github-task  Solution.md  Task.md
```

3. Create a text file named notes.txt and write at least one line of text in it.

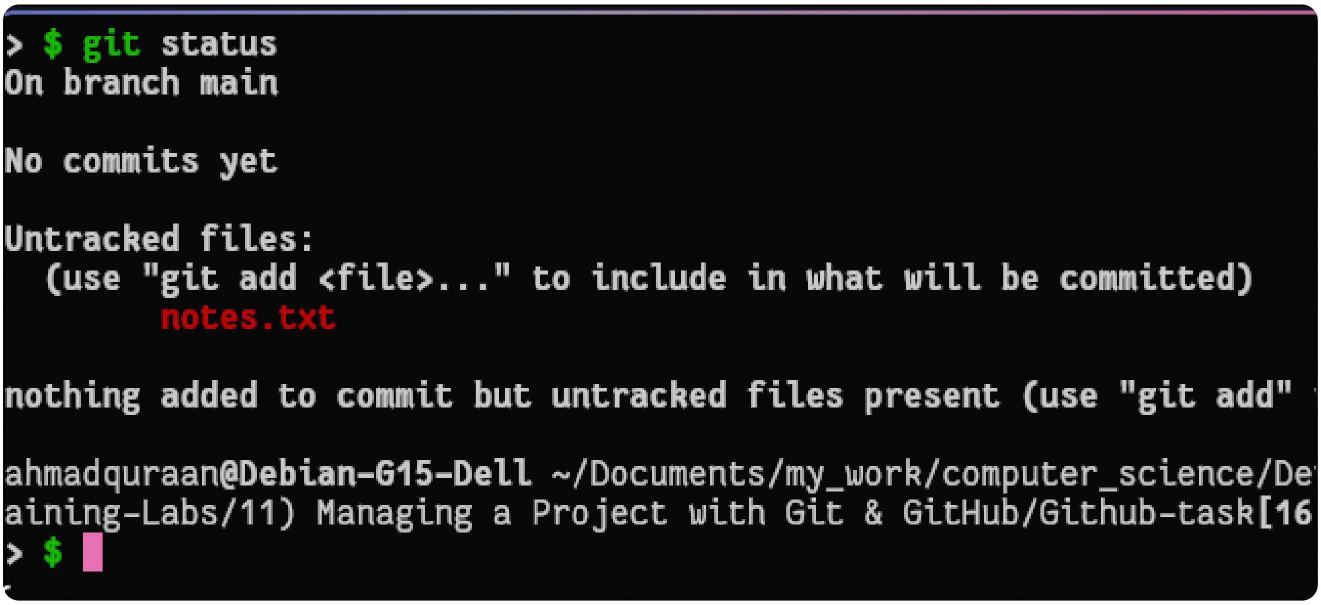
```
cd Github-task
echo "HI" > notes.txt
```



```
ahmadquraan@Debian-G15-Dell ~/Docu
aining-Labs/11) Managing a Project
> $ cd Github-task

ahmadquraan@Debian-G15-Dell ~/Docu
aining-Labs/11) Managing a Project
> $ ls
notes.txt

ahmadquraan@Debian-G15-Dell ~/Docu
aining-Labs/11) Managing a Project
```

4. Check the repository status and verify that the new file is detected.

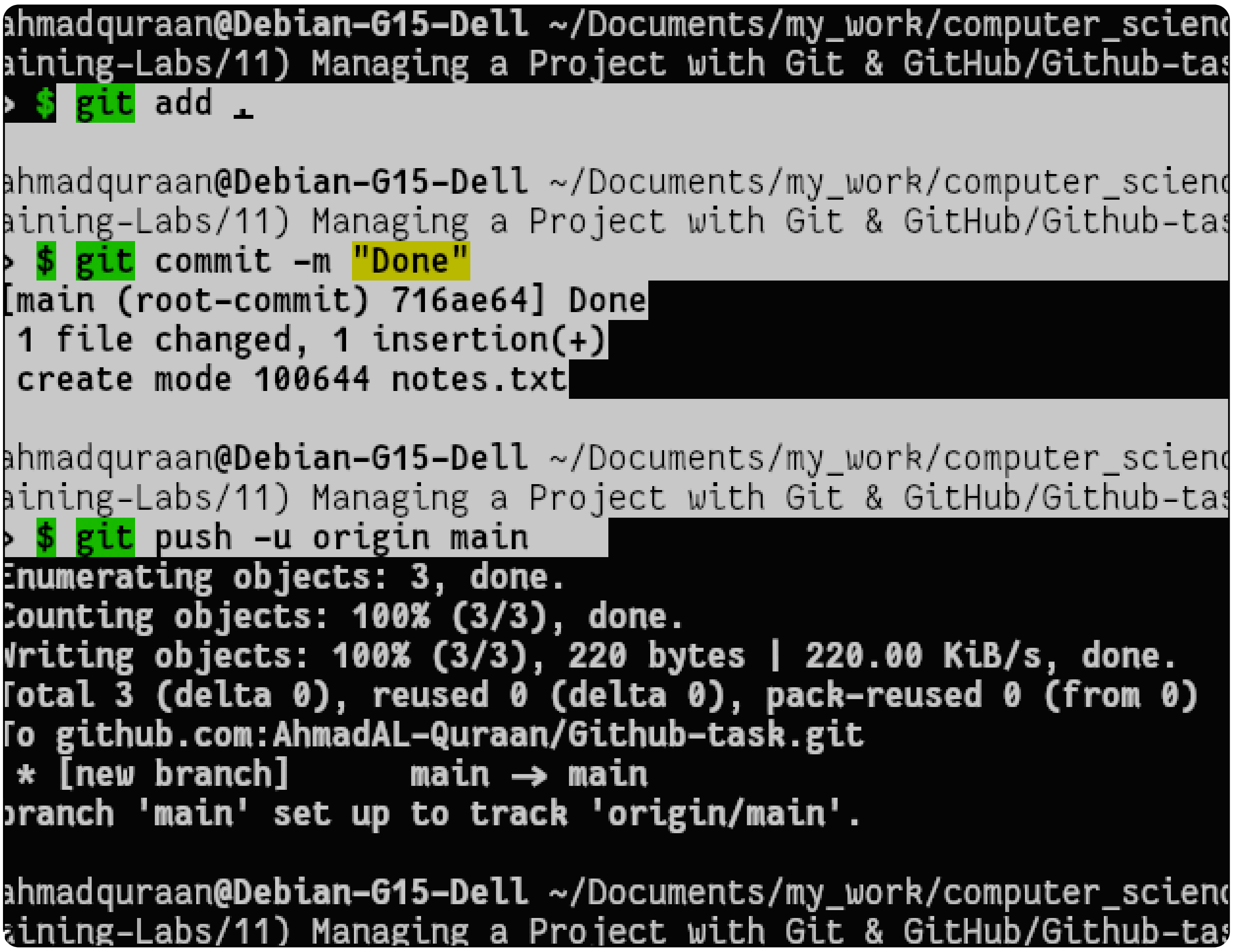
```
git status
```



```
> $ git status
On branch main

No commits yet

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

nothing added to commit but untracked files present (use "git add"

ahmadquraan@Debian-G15-Dell ~/Documents/my_work/computer_science/De
aining-Labs/11) Managing a Project with Git & GitHub/Github-task[16
> $
```

5. Save the initial changes to the repository and upload them to GitHub.

```
git add .
git commit -m "Done"
git push -u origin main
```

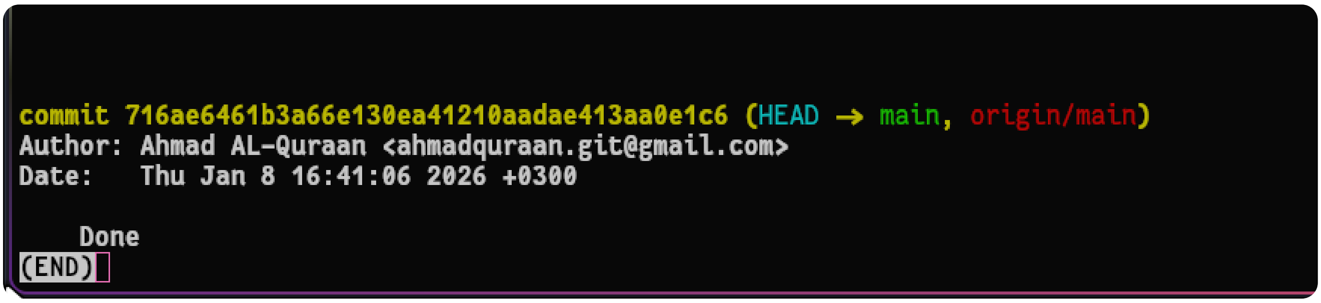


```
ahmadquraan@Debian-G15-Dell ~/Documents/my_work/computer_scien
aining-Labs/11) Managing a Project with Git & GitHub/Github-ta
> $ git add .

ahmadquraan@Debian-G15-Dell ~/Documents/my_work/computer_scien
aining-Labs/11) Managing a Project with Git & GitHub/Github-ta
> $ git commit -m "Done"
[main (root-commit) 716ae64] Done
 1 file changed, 1 insertion(+)
 create mode 100644 notes.txt

ahmadquraan@Debian-G15-Dell ~/Documents/my_work/computer_scien
aining-Labs/11) Managing a Project with Git & GitHub/Github-ta
> $ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 220 bytes | 220.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:AhmadAL-Quraan/Github-task.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

ahmadquraan@Debian-G15-Dell ~/Documents/my_work/computer_scien
aining-Labs/11) Managing a Project with Git & GitHub/Github-ta
```

Phase 2: Viewing Project History

1. Display the full history of saved changes in the repository.

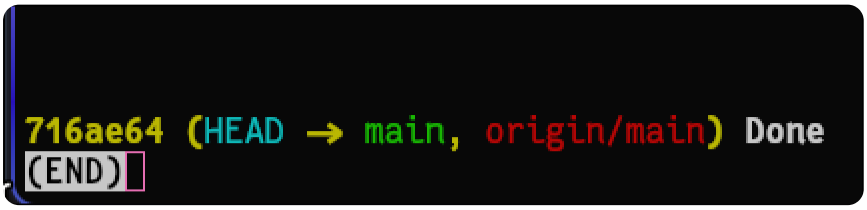
```
git log
```



```
commit 716ae6461b3a66e130ea41210aadae413aa0e1c6 (HEAD -> main, origin/main)
Author: Ahmad AL-Quraan <ahmadquraan.git@gmail.com>
Date:   Thu Jan 8 16:41:06 2026 +0300

    Done
(END)
```

2. Display a shortened version of the history showing one entry per change.

```
git log --oneline
```

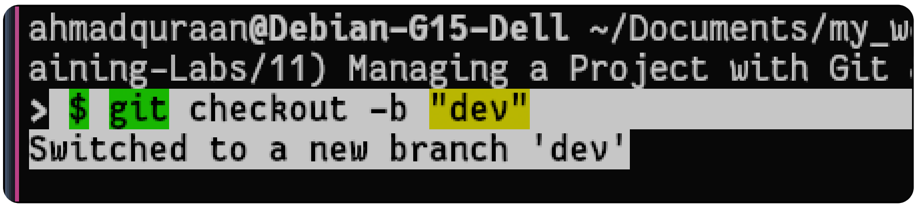


```
716ae64 (HEAD -> main, origin/main) Done
(END)
```

Phase 3: Working with Branches

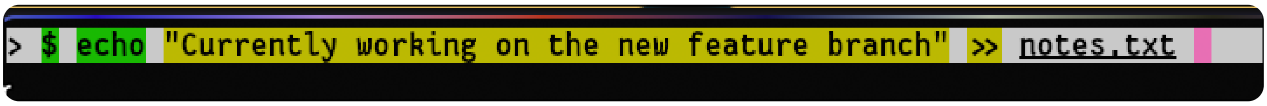
1. Create a new branch for feature development and switch into it

```
git checkout -b "dev"
```



```
ahmadquraan@Debian-G15-Dell ~/Documents/my_w
aining-Labs/11) Managing a Project with Git .
> $ git checkout -b "dev"
Switched to a new branch 'dev'
```

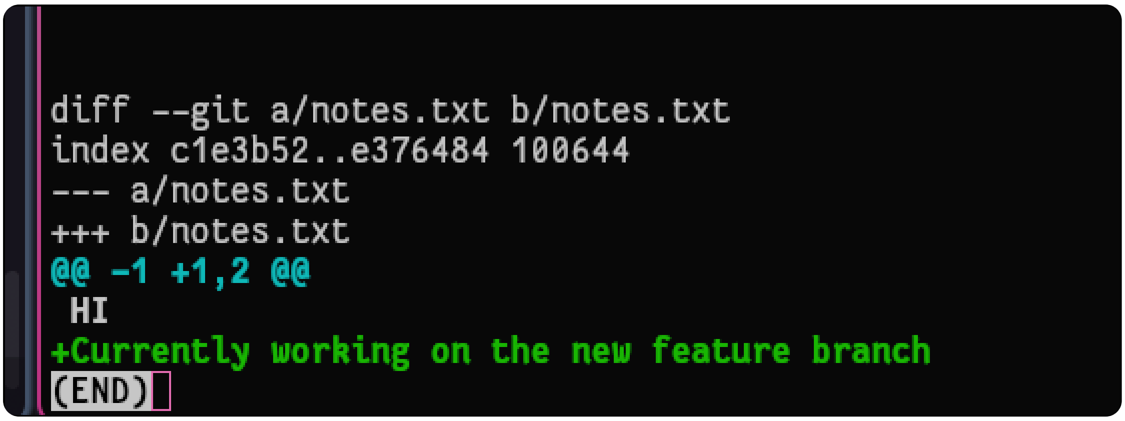
2. Edit the notes.txt file by adding a new line that



```
> $ echo "Currently working on the new feature branch" >> notes.txt
```

3. Compare the current version of the file with the last saved version (last commited one).

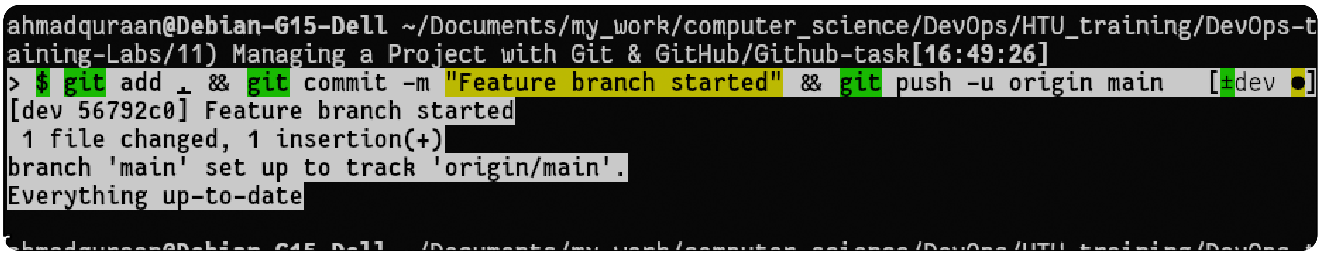
```
git diff notes.txt
```



```
diff --git a/notes.txt b/notes.txt
index c1e3b52..e376484 100644
--- a/notes.txt
+++ b/notes.txt
@@ -1 +1,2 @@
 HI
+Currently working on the new feature branch
(END)
```

4) Save the changes in the feature branch && upload the feature branch to github.

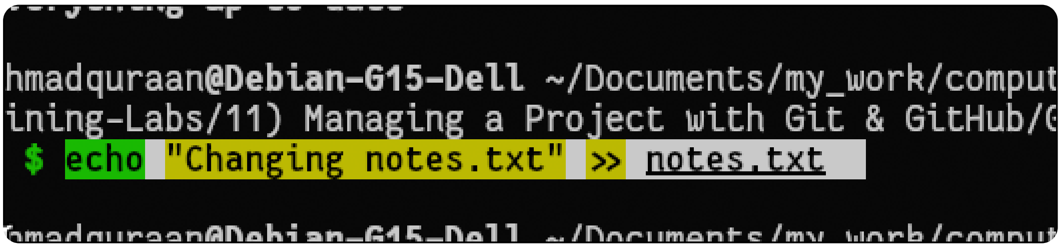
```
git add .
git commit -m "Feature branch started"
git push -u origin dev
```



```
ahmadquraan@Debian-G15-Dell ~/Documents/my_work/computer_science/DevOps/HTU_training/DevOps-t
aining-Labs/11) Managing a Project with Git & GitHub/Github-task[16:49:26]
> $ git add . && git commit -m "Feature branch started" && git push -u origin main    [±dev ●]
[dev 56792c0] Feature branch started
 1 file changed, 1 insertion(+)
branch 'main' set up to track 'origin/main'.
Everything up-to-date
```

Phase 4: Discarding Changes

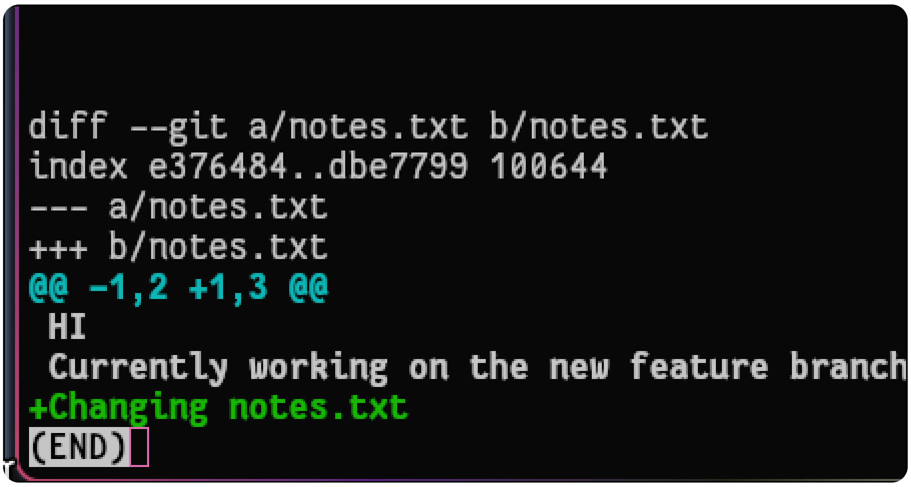
1. Modify the notes.txt file again without saving the change to the repository.



```
hmadquraan@Debian-G15-Dell ~/Documents/my_work/comput
ining-Labs/11) Managing a Project with Git & GitHub/G
 $ echo "Changing notes.txt" >> notes.txt
```

2. View the difference between the modified file and the saved version.

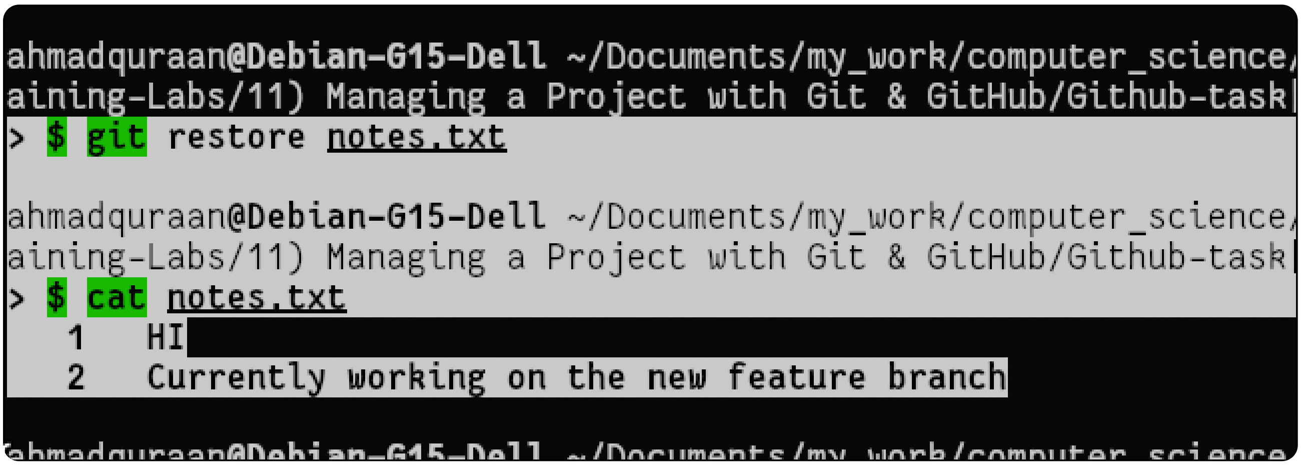
```
git diff notes.txt
```



```
diff --git a/notes.txt b/notes.txt
index e376484..dbe7799 100644
--- a/notes.txt
+++ b/notes.txt
@@ -1,2 +1,3 @@
 HI
 Currently working on the new feature branch
+Changing notes.txt
(END)
```

3. Discard the latest change and restore the file to its last saved state.

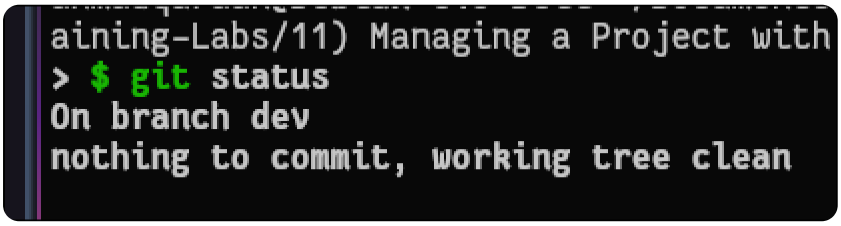
```
git restore notes.txt
```



```
ahmadquraan@Debian-G15-Dell ~/Documents/my_work/computer_science/
aining-Labs/11) Managing a Project with Git & GitHub/Github-task
> $ git restore notes.txt

ahmadquraan@Debian-G15-Dell ~/Documents/my_work/computer_science/
aining-Labs/11) Managing a Project with Git & GitHub/Github-task
> $ cat notes.txt
   1    HI
   2    Currently working on the new feature branch

ahmadquraan@Debian-G15-Dell ~/Documents/my_work/computer_science
```

4. Confirm that the repository has no pending

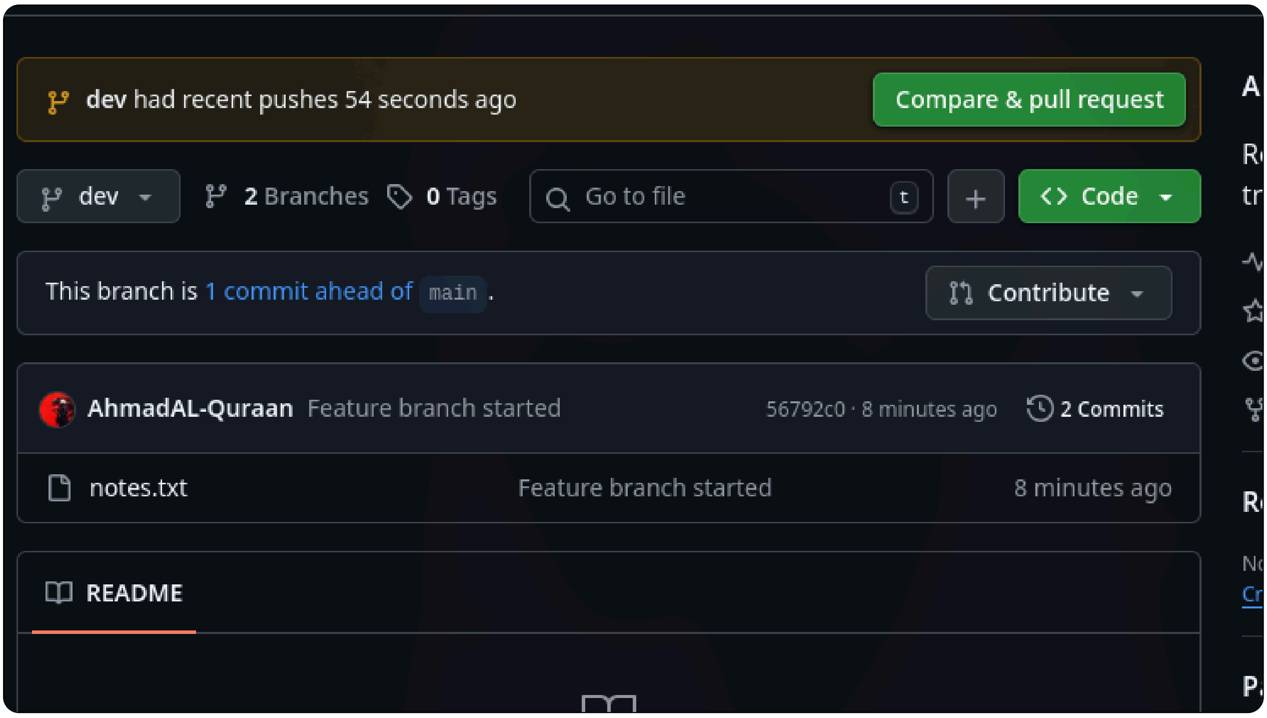
```
git status
```



```
aining-Labs/11) Managing a Project with
> $ git status
On branch dev
nothing to commit, working tree clean
```

Phase 5: Pull Request

1. On GitHub, create a Pull Request to merge the feature branch into the main branch.

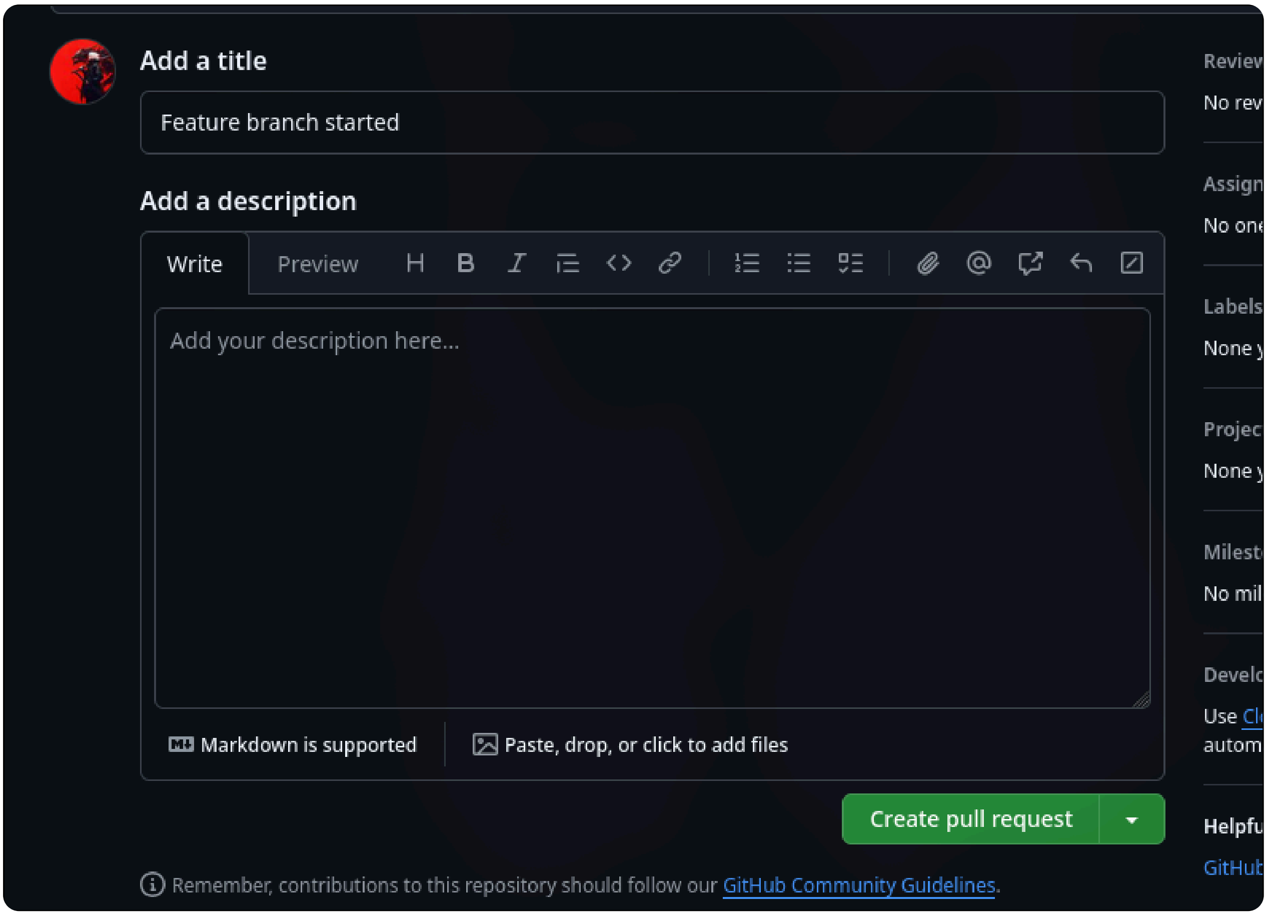


dev had recent pushes 54 seconds ago

This branch is 1 commit ahead of `main`.

AhmadAL-Quraan Feature branch started 56792c0 · 8 minutes ago 2 Commits

notes.txt Feature branch started 8 minutes ago



Add a title

Feature branch started

Add a description

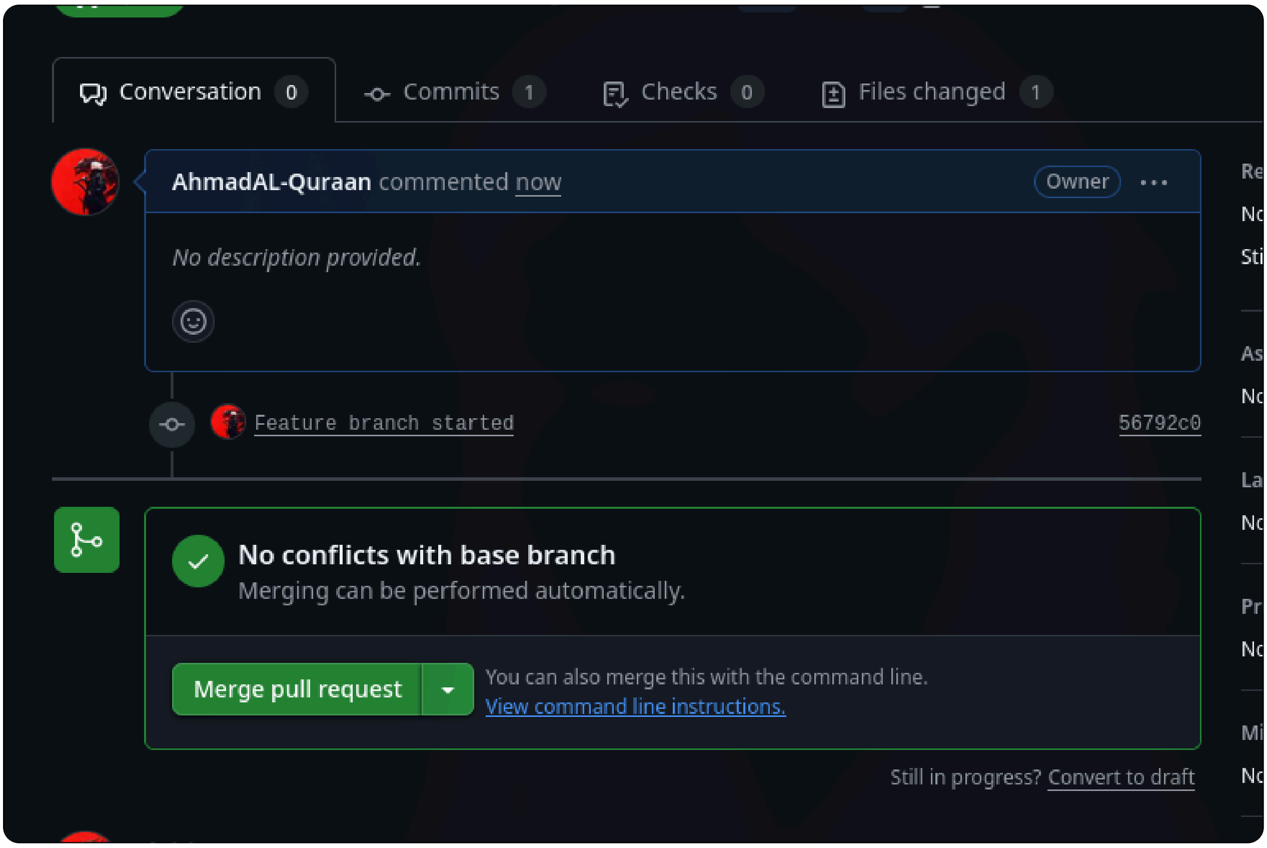
Write Preview

Add your description here...

Markdown is supported | Paste, drop, or click to add files

Create pull request

Remember, contributions to this repository should follow our GitHub Community Guidelines.



Conversation 0 Commits 1 Checks 0 Files changed 1

AhmadAL-Quraan commented now — Owner

No description provided.

Feature branch started 56792c0

No conflicts with base branch

Merging can be performed automatically.

Merge pull request

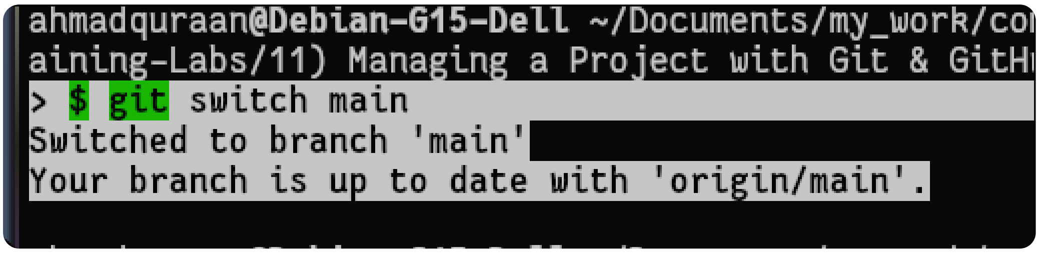
You can also merge this with the command line. View command line instructions.

Still in progress? Convert to draft

Phase 6: Creating a Conflict

1. Return to the main branch locally.

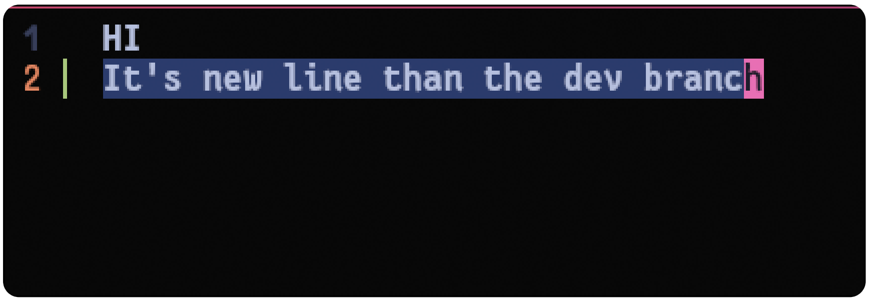
```
git switch main
```



```
ahmadquraan@Debian-G15-Dell ~/Documents/my_work/co
aining-Labs/11) Managing a Project with Git & GitH
> $ git switch main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

2. Modify the same line in notes.txt that was previously changed in the feature branch.

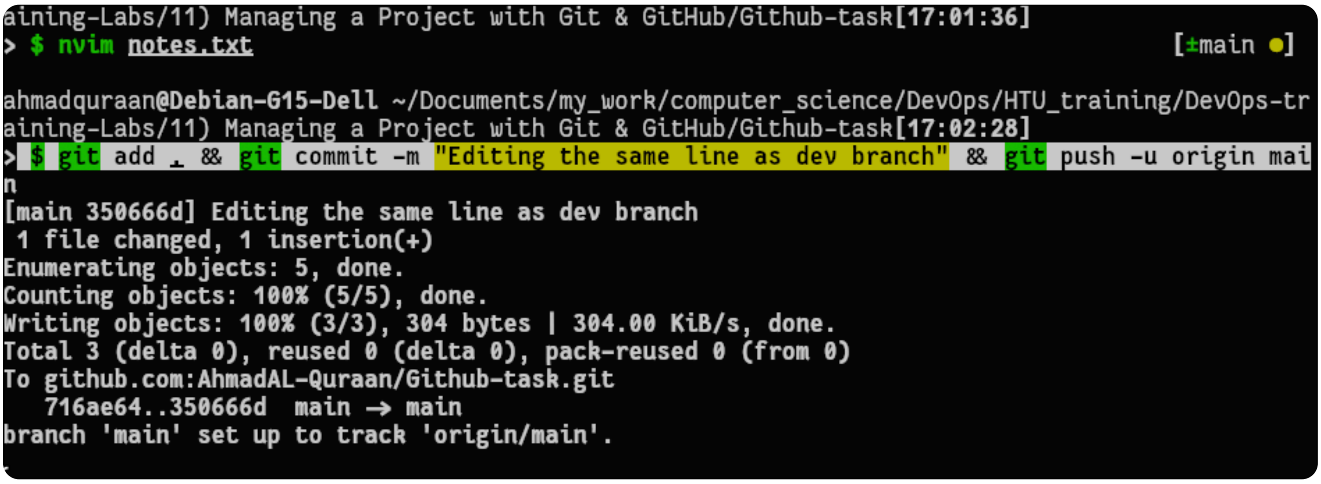
```
nvim notes.txt
```



```
1    HI
2 |  It's new line than the dev branch
```

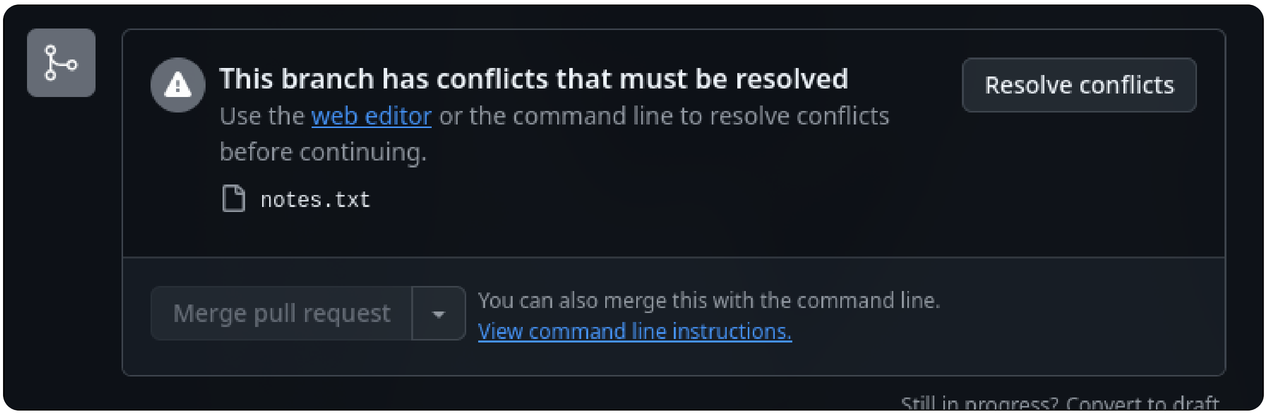
3. Save the change and upload it to GitHub

```
git add .
git commit -m "Editing the same line as dev branch"
git push -u origin main
```



```
aining-Labs/11) Managing a Project with Git & GitHub/Github-task[17:01:36]
> $ nvim notes.txt                                                          [±main ●]

ahmadquraan@Debian-G15-Dell ~/Documents/my_work/computer_science/DevOps/HTU_training/DevOps-tr
aining-Labs/11) Managing a Project with Git & GitHub/Github-task[17:02:28]
> $ git add . && git commit -m "Editing the same line as dev branch" && git push -u origin mai
n
[main 350666d] Editing the same line as dev branch
 1 file changed, 1 insertion(+)
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 304 bytes | 304.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:AhmadAL-Quraan/Github-task.git
   716ae64..350666d  main -> main
branch 'main' set up to track 'origin/main'.
```

5) Attempt to complete the Pull Request and observe the merge conflict

- 

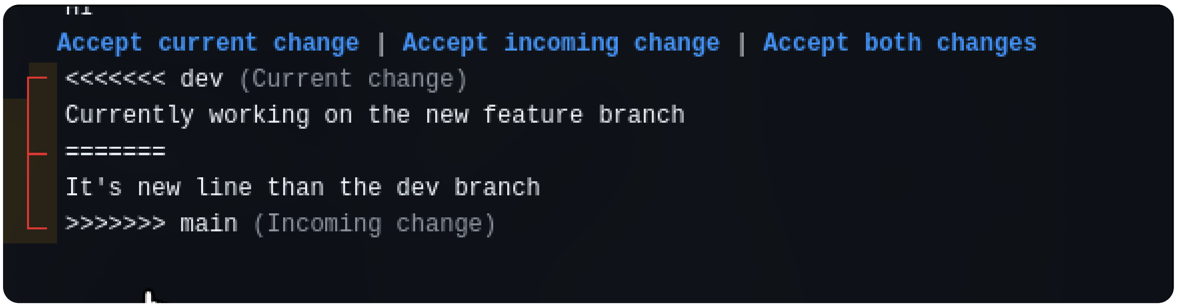
  This branch has conflicts that must be resolved
  Use the web editor or the command line to resolve conflicts before continuing.

  notes.txt

  You can also merge this with the command line. View command line instructions.

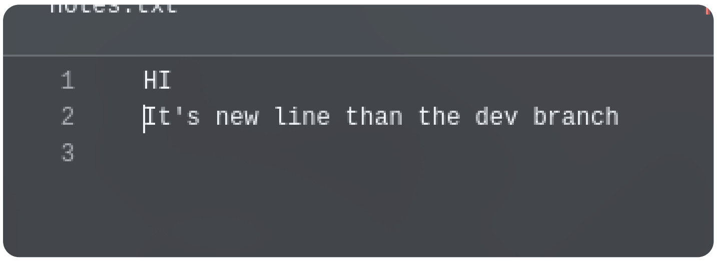
  Still in progress? Convert to draft

5. Resolve the conflict so that the final file content is clear and correct.



```
HI
Accept current change | Accept incoming change | Accept both changes
<<<<<<< dev (Current change)
Currently working on the new feature branch
=======
It's new line than the dev branch
>>>>>>> main (Incoming change)
```

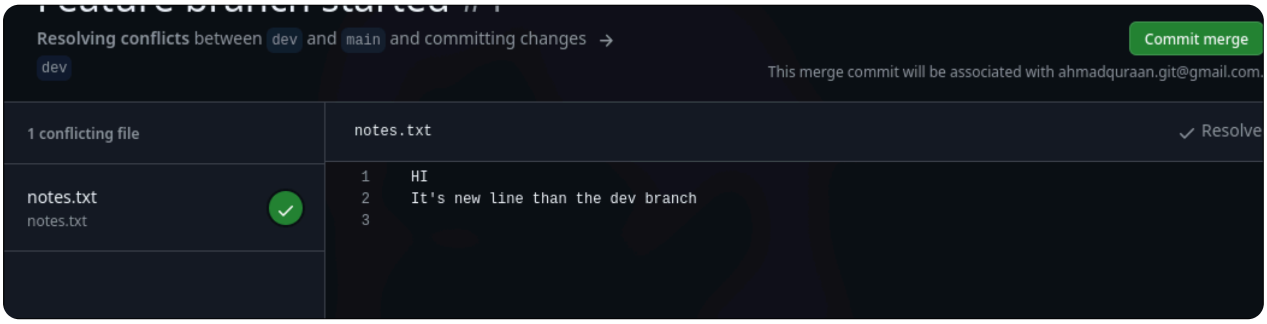
```
Accept incoming change --> choosed this option
```



notes.txt

```
HI
It's new line than the dev branch

```

6. Save the conflict resolution.



Resolving conflicts between `dev` and `main` and committing changes →

dev

This merge commit will be associated with ahmadquraan.git@gmail.com.

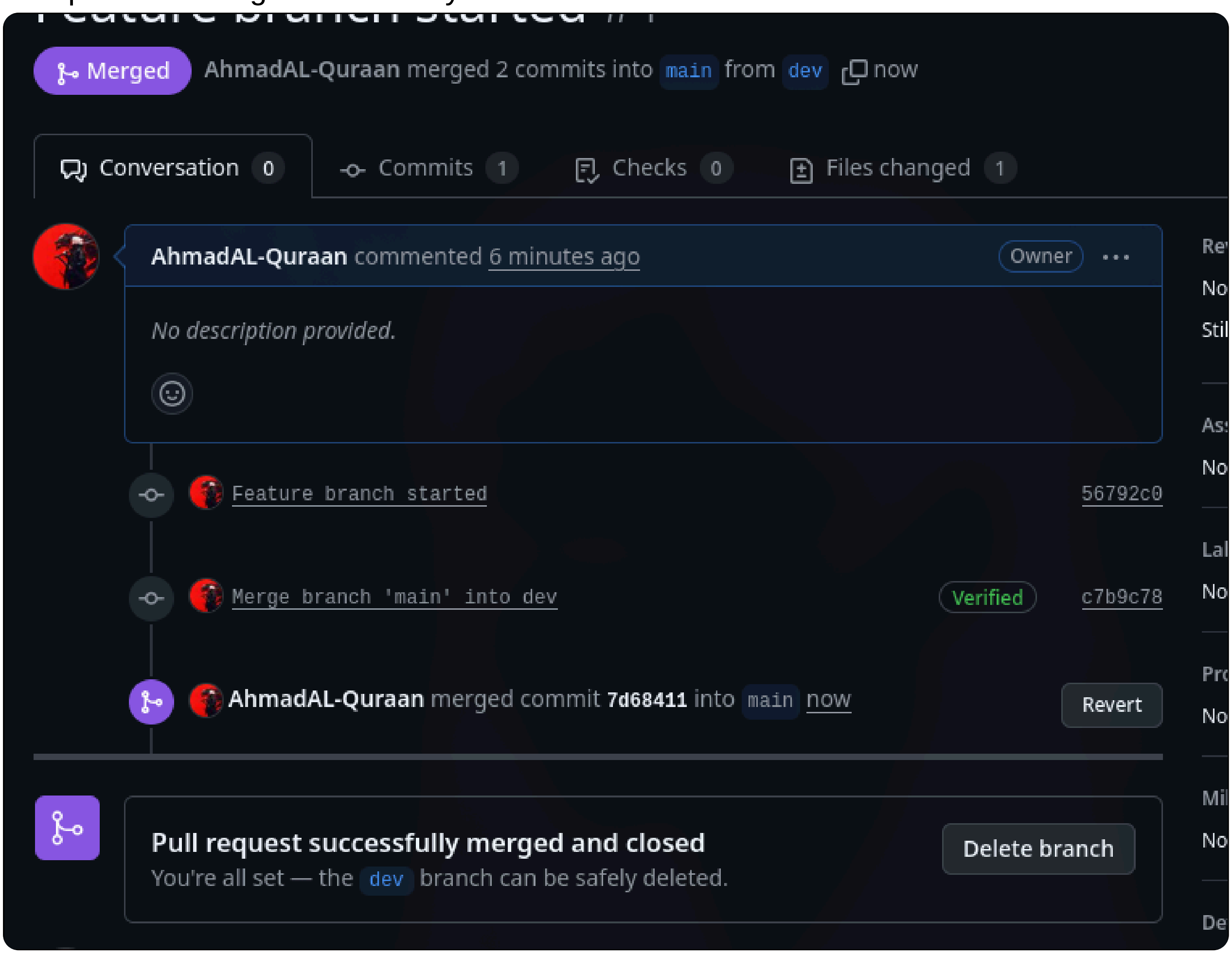
1 conflicting file

notes.txt
notes.txt

notes.txt

```
HI
It's new line than the dev branch

```

7. Complete the merge successfully.



Merged AhmadAL-Quraan merged 2 commits into main from dev now

AhmadAL-Quraan commented 6 minutes ago Owner

No description provided.

Feature branch started 56792c0

Merge branch 'main' into dev Verified c7b9c78

AhmadAL-Quraan merged commit 7d68411 into main now

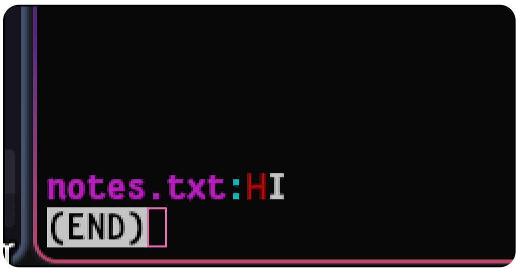
Pull request successfully merged and closed

You're all set — the dev branch can be safely deleted.

Phase 7: Searching the Repository

1. Use Git to search the project files for a word that exists in notes.txt.

```
git grep <word> <fileName>
git grep "H" notes.txt
```

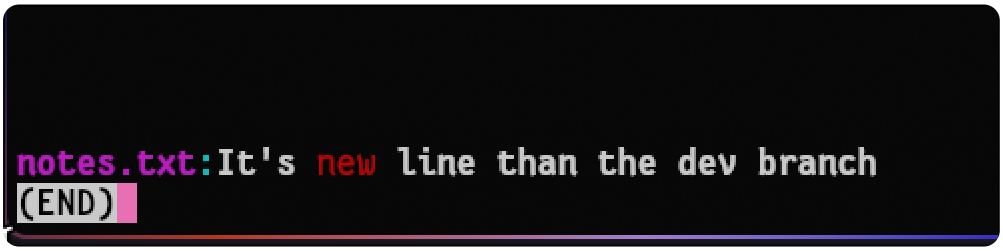


```
notes.txt:HI
(END)
```

2. Perform another search using a different word.

`git grep "new" notes.txt`

Solution

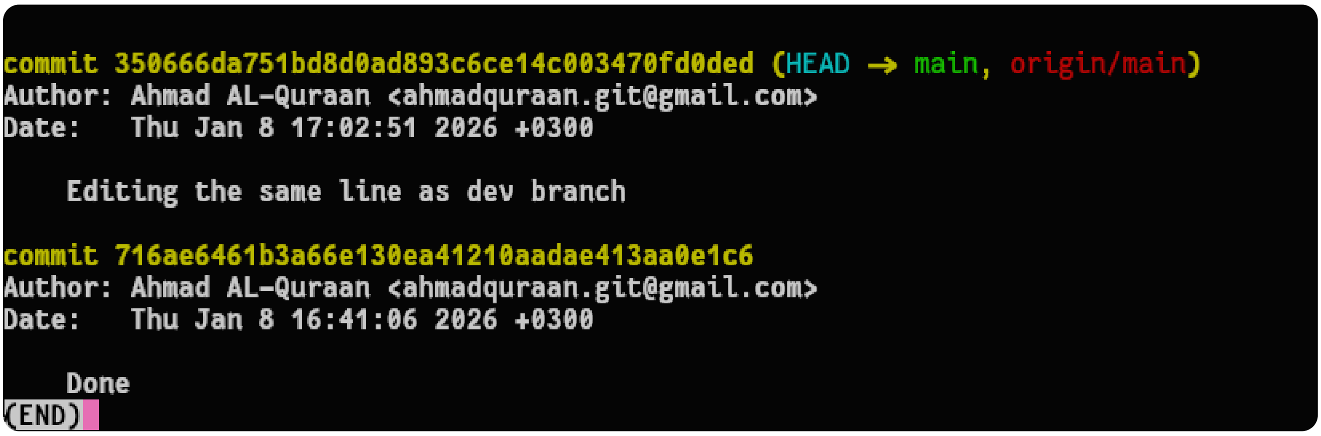


```
notes.txt:It's new line than the dev branch
(END)
```

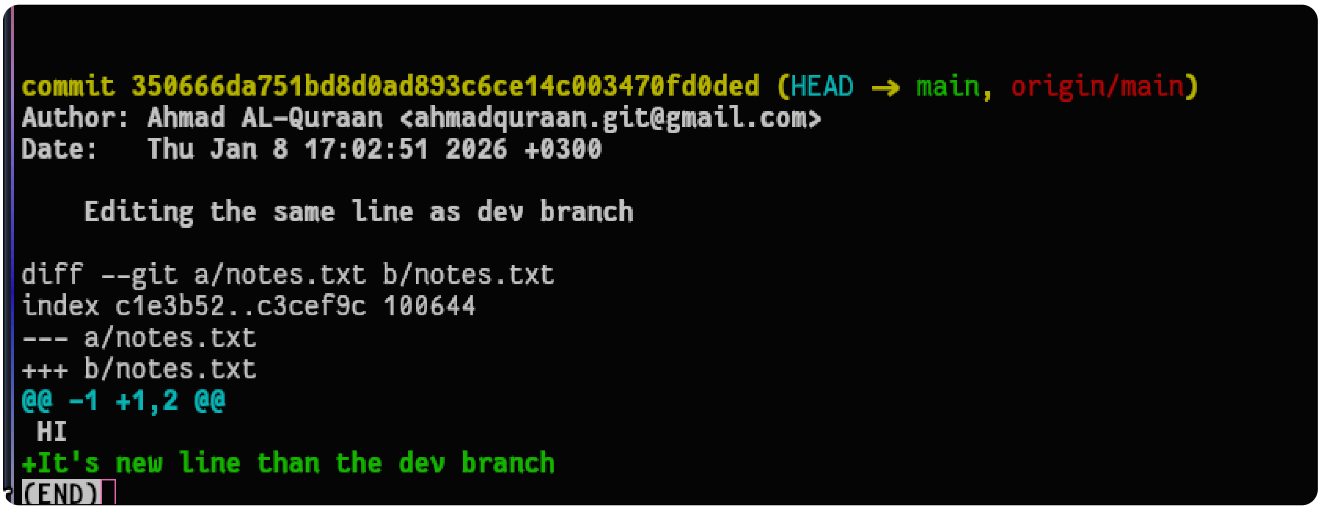
Phase 8: Inspecting Changes

1. Display detailed information about the most recent saved change, including file modifications.

```
git log
git show <commit-id>
```



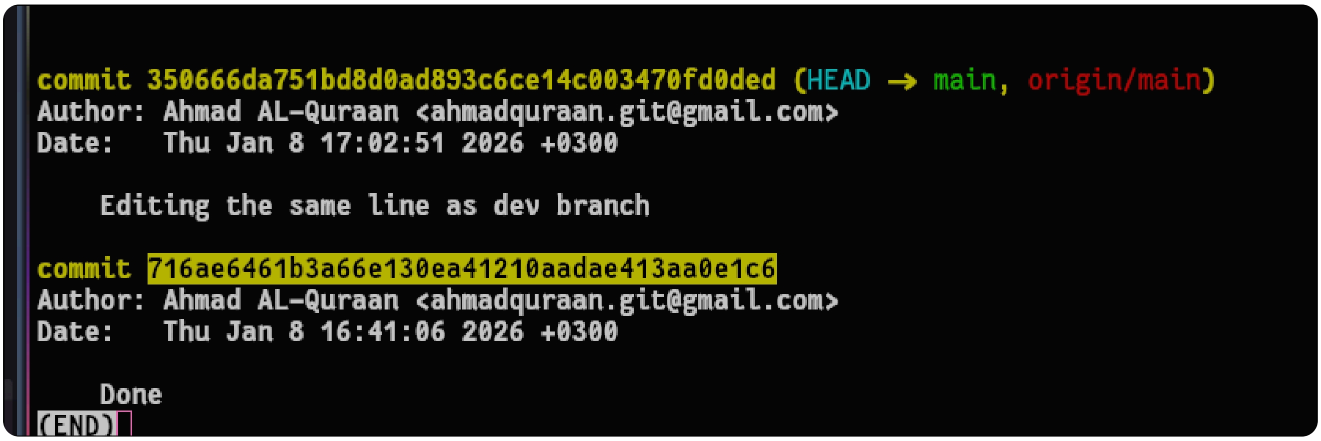
```
commit 350666da751bd8d0ad893c6ce14c003470fd0ded (HEAD -> main, origin/main)
Author: Ahmad AL-Quraan <ahmadquraan.git@gmail.com>
Date:   Thu Jan 8 17:02:51 2026 +0300

    Editing the same line as dev branch

commit 716ae6461b3a66e130ea41210aadae413aa0e1c6
Author: Ahmad AL-Quraan <ahmadquraan.git@gmail.com>
Date:   Thu Jan 8 16:41:06 2026 +0300

    Done
(END)
```



```
commit 350666da751bd8d0ad893c6ce14c003470fd0ded (HEAD -> main, origin/main)
Author: Ahmad AL-Quraan <ahmadquraan.git@gmail.com>
Date:   Thu Jan 8 17:02:51 2026 +0300

    Editing the same line as dev branch

diff --git a/notes.txt b/notes.txt
index c1e3b52..c3cef9c 100644
--- a/notes.txt
+++ b/notes.txt
@@ -1 +1,2 @@
 HI
+It's new line than the dev branch
(END)
```

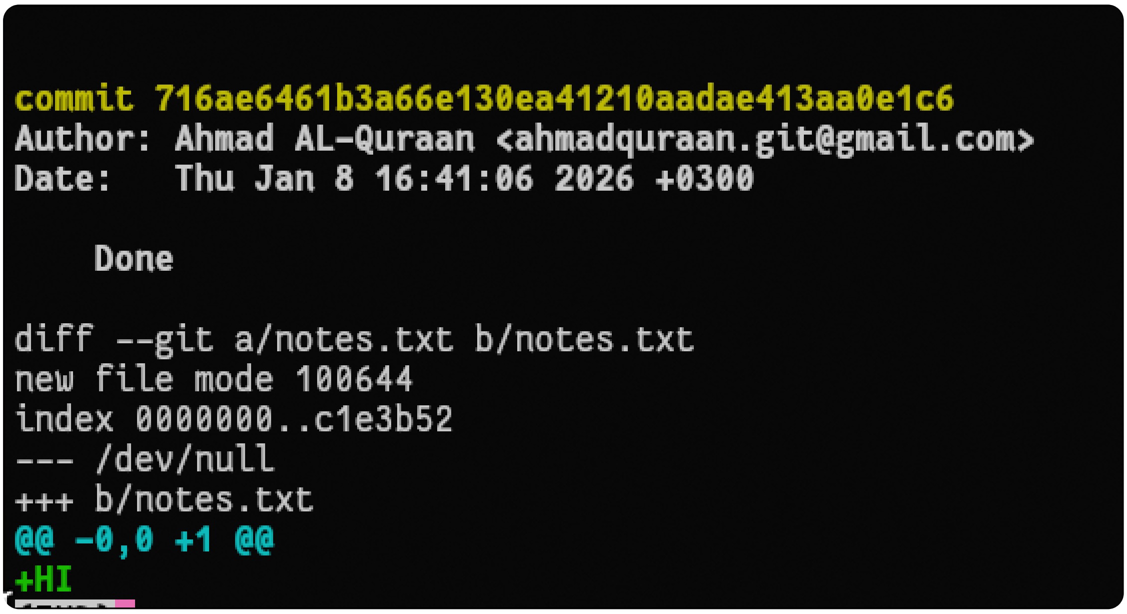
2. Display detailed information for an older saved change using its identifier.

```
git log
git show <commit-id> --> choose the older one
```



```
commit 350666da751bd8d0ad893c6ce14c003470fd0ded (HEAD -> main, origin/main)
Author: Ahmad AL-Quraan <ahmadquraan.git@gmail.com>
Date:   Thu Jan 8 17:02:51 2026 +0300

    Editing the same line as dev branch

commit 716ae6461b3a66e130ea41210aadae413aa0e1c6
Author: Ahmad AL-Quraan <ahmadquraan.git@gmail.com>
Date:   Thu Jan 8 16:41:06 2026 +0300

    Done
(END)
```

`git show <commit-id>`

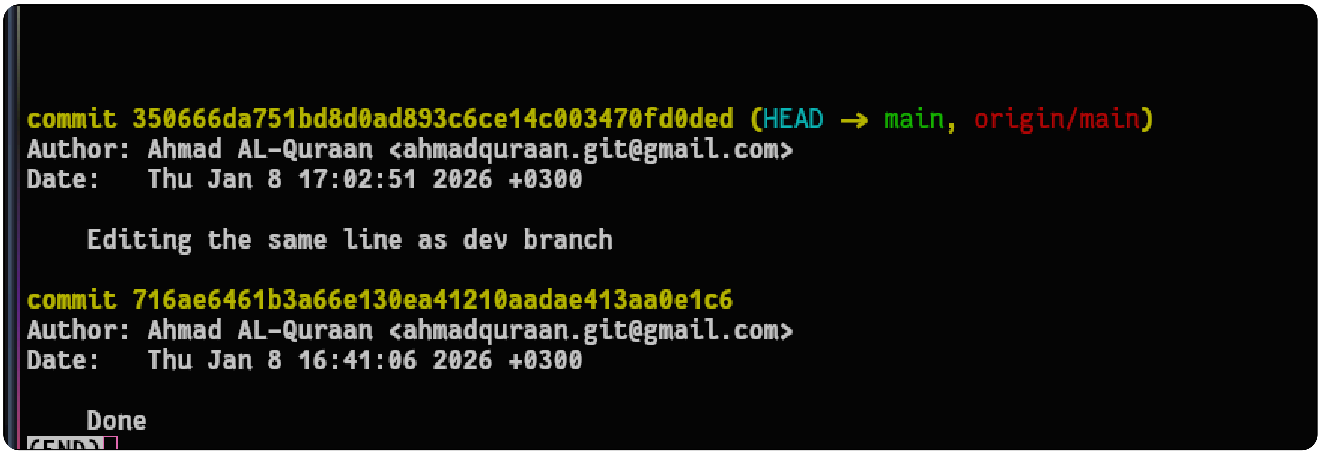


```
commit 716ae6461b3a66e130ea41210aadae413aa0e1c6
Author: Ahmad AL-Quraan <ahmadquraan.git@gmail.com>
Date:   Thu Jan 8 16:41:06 2026 +0300

    Done

diff --git a/notes.txt b/notes.txt
new file mode 100644
index 0000000..c1e3b52
--- /dev/null
+++ b/notes.txt
@@ -0,0 +1 @@
+HI
```

Phase 9: Undoing Commits

1. View the project history again.

```
git log
```

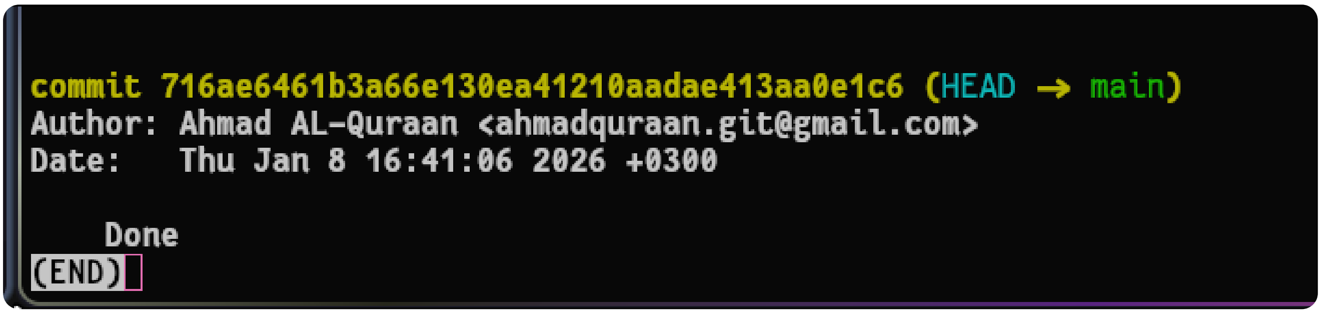


```
commit 350666da751bd8d0ad893c6ce14c003470fd0ded (HEAD -> main, origin/main)
Author: Ahmad AL-Quraan <ahmadquraan.git@gmail.com>
Date:   Thu Jan 8 17:02:51 2026 +0300

    Editing the same line as dev branch

commit 716ae6461b3a66e130ea41210aadae413aa0e1c6
Author: Ahmad AL-Quraan <ahmadquraan.git@gmail.com>
Date:   Thu Jan 8 16:41:06 2026 +0300

    Done
(END)
```

2) Undo the most recent saved change while keeping the file changes locally.

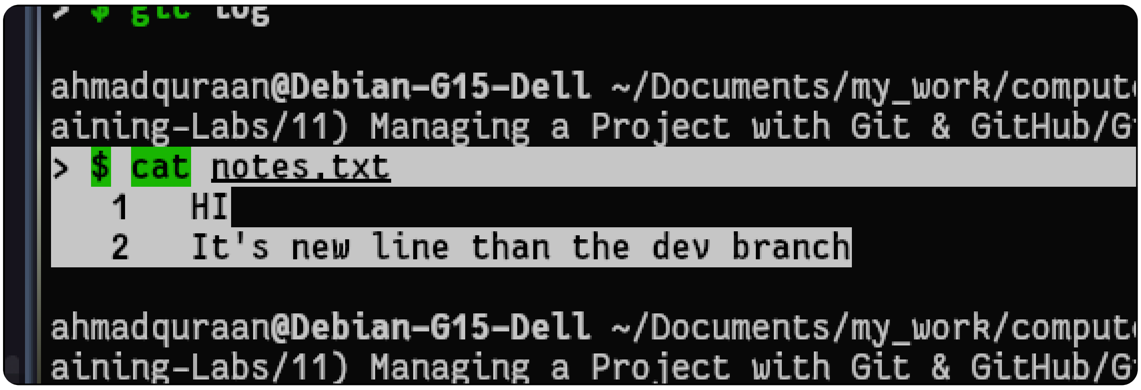
```
git reset --soft HEAD~1
git log
```

``



```
commit 716ae6461b3a66e130ea41210aadae413aa0e1c6 (HEAD -> main)
Author: Ahmad AL-Quraan <ahmadquraan.git@gmail.com>
Date:   Thu Jan 8 16:41:06 2026 +0300

    Done
(END)
```

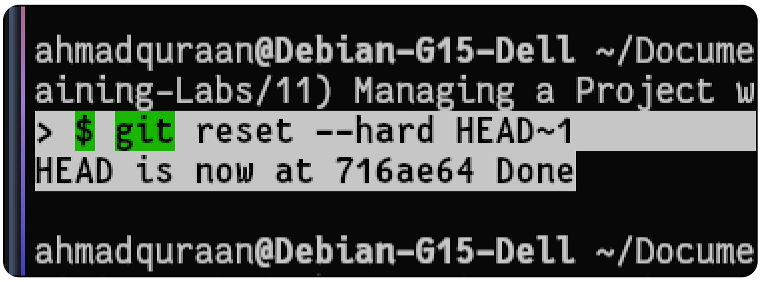
Changes kept as they are in notes.txt



```
> $ git log

ahmadquraan@Debian-G15-Dell ~/Documents/my_work/comput
aining-Labs/11) Managing a Project with Git & GitHub/G
> $ cat notes.txt
   1   HI
   2   It's new line than the dev branch

ahmadquraan@Debian-G15-Dell ~/Documents/my_work/comput
aining-Labs/11) Managing a Project with Git & GitHub/G
```

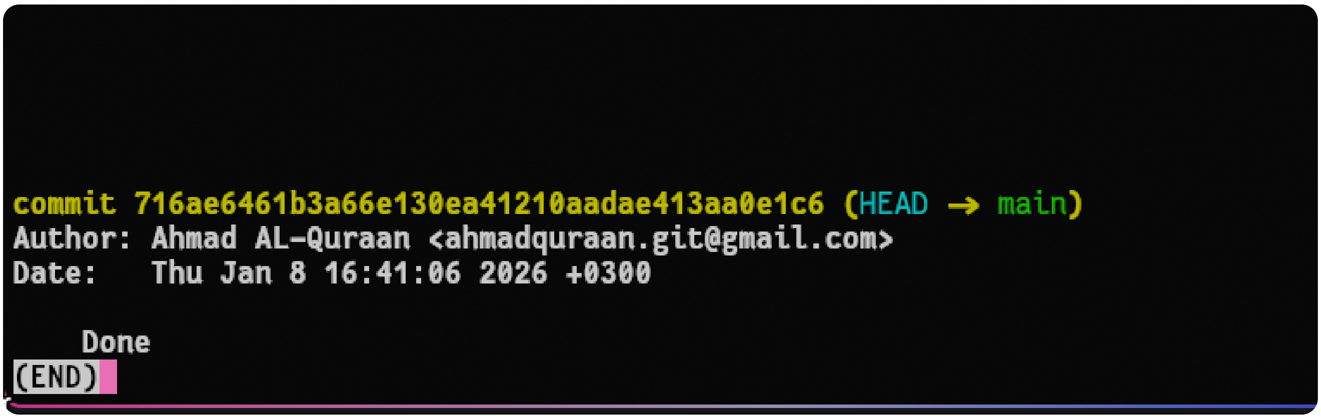
3. Undo another saved change and remove the file changes completely.

```
git reset --hard HEAD~1
```



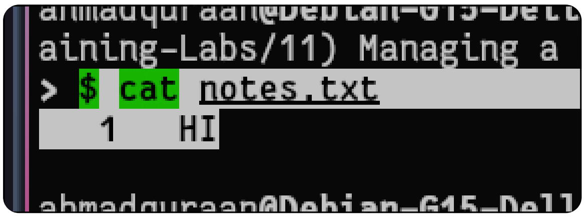
```
ahmadquraan@Debian-G15-Dell ~/Docume
aining-Labs/11) Managing a Project w
> $ git reset --hard HEAD~1
HEAD is now at 716ae64 Done

ahmadquraan@Debian-G15-Dell ~/Docume
```

I made another commit so it has parent and then used the command (because you can't make reset without having a parent commit, like just one commit).



```
commit 716ae6461b3a66e130ea41210aadae413aa0e1c6 (HEAD → main)
Author: Ahmad AL-Quraan <ahmadquraan.git@gmail.com>
Date:   Thu Jan 8 16:41:06 2026 +0300

    Done
(END)
```

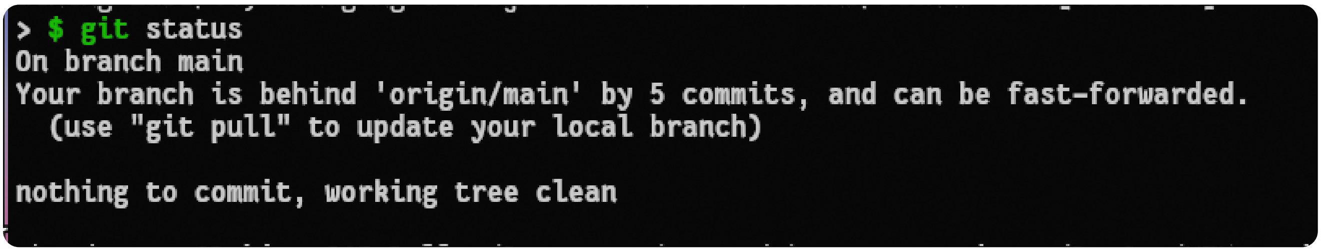
Changes happened in notes.txt



```
ahmadquraan@Debian-G15-Dell
aining-Labs/11) Managing a
> $ cat notes.txt
   1   HI

ahmadquraan@Debian-G15-Dell
```

4. Check the repository status after undoing the changes.

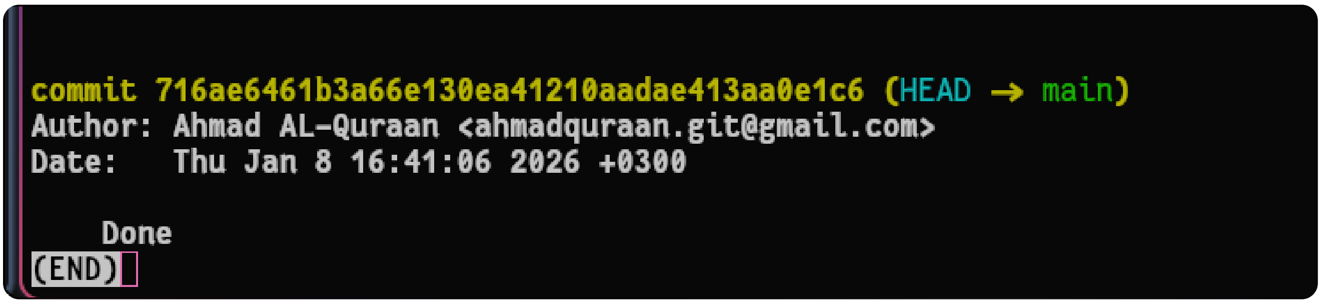
```
git status
```



```
> $ git status
On branch main
Your branch is behind 'origin/main' by 5 commits, and can be fast-forwarded.
  (use "git pull" to update your local branch)

nothing to commit, working tree clean
```

5. Ensure the project history reflects all completed phases.

```
git log
```



```
commit 716ae6461b3a66e130ea41210aadae413aa0e1c6 (HEAD → main)
Author: Ahmad AL-Quraan <ahmadquraan.git@gmail.com>
Date:   Thu Jan 8 16:41:06 2026 +0300

    Done
(END)
```