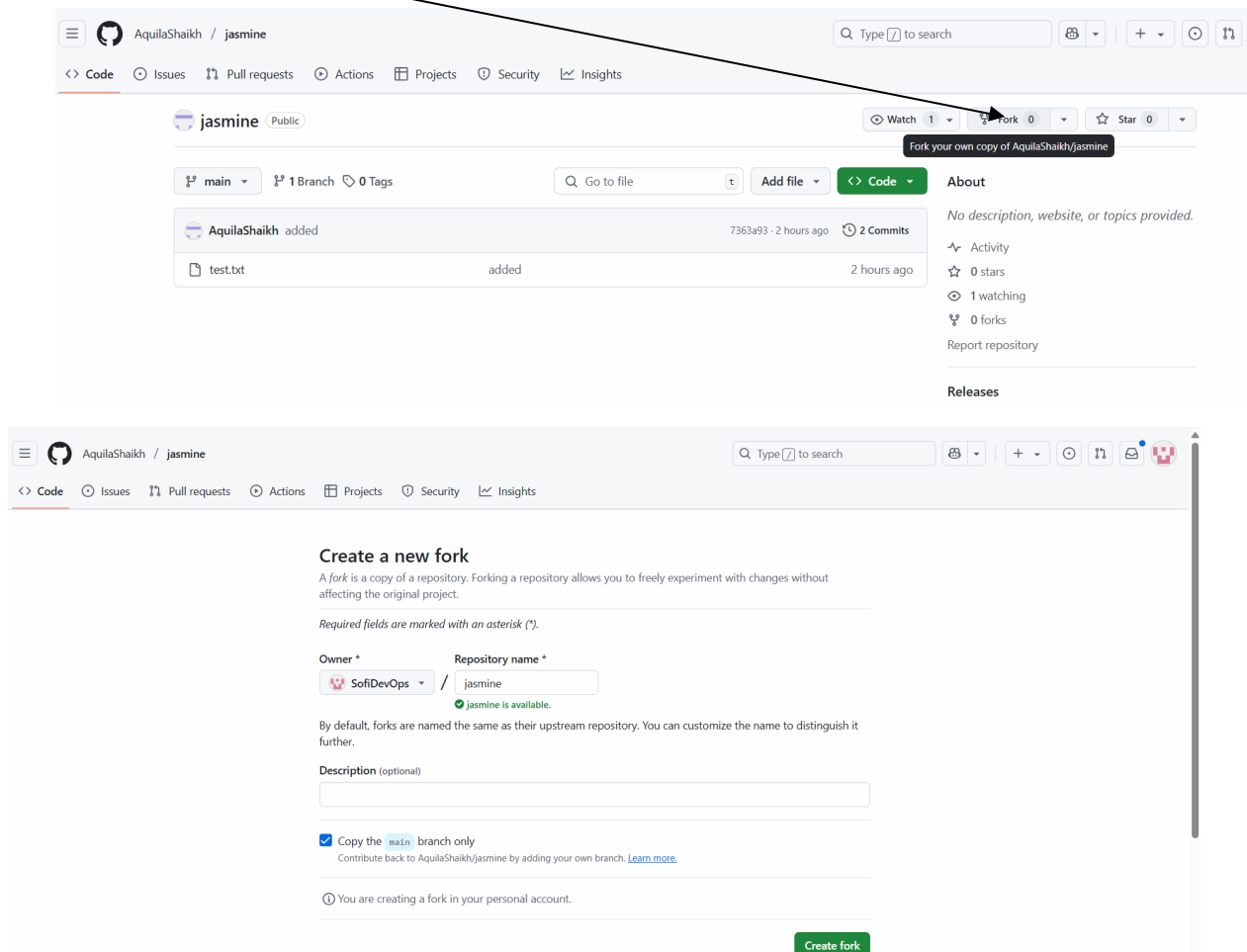


1. Go to First account and select the repository you want to fork and click on the fork option.



2. Click on the fork option.
3. Now you will be able to see that forked repository in your second account.
4. Clone that forked repository in any folder, say "DemoFork"

```
aquis@LAPTOP-CVDR41EG MINGW64 ~ (master)
$ cd C:\\Users\\aquis\\Desktop\\GitNew

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitNew (master)
$ git clone https://github.com/SofiDevOps/jasmine.git
Cloning into 'jasmine'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (6/6), done.

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitNew (master)
$ ls
jasmine/
```

5. You will be able to see the repository in your folder.
6. Check the file present in your repository.

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitNew/jasmine (main)
$ ls
test.txt
```

And content of your file.

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitNew/jasmine (main)
$ cat test.txt
hello
hi
hi
hi
```

7. Modify your file and check the status.

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitNew/jasmine (main)
$ vim test.txt

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitNew/jasmine (main)
$ cat test.txt
hello
hi
hi
hi
hello
hello

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitNew/jasmine (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

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:   test.txt
```

8. Check origin and remove it.

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitNew/jasmine (main)
$ git remote
origin

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitNew/jasmine (main)
$ git remote rm origin

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitNew/jasmine (main)
$ git remote
```

9. Perform add and commit. Check the status.

```

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitNew/jasmine (main)
$ git add .

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitNew/jasmine (main)
$ git status
on branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   test.txt

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitNew/jasmine (main)
$ git commit -m "adding updation"
[main baca4aa] adding updation
1 file changed, 2 insertions(+)

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitNew/jasmine (main)
$ git status
on branch main
nothing to commit, working tree clean

```

10. Add your repository to the remote origin.

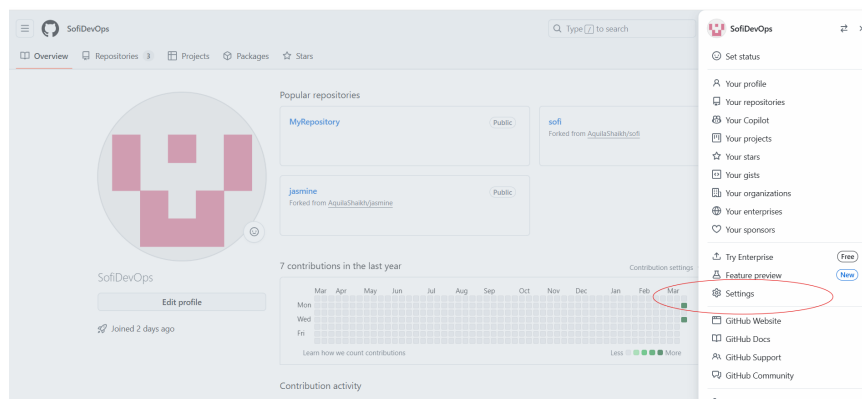
```

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitNew/jasmine (main)
$ git remote add origin https://github.com/SofiDevOps/jasmine.git

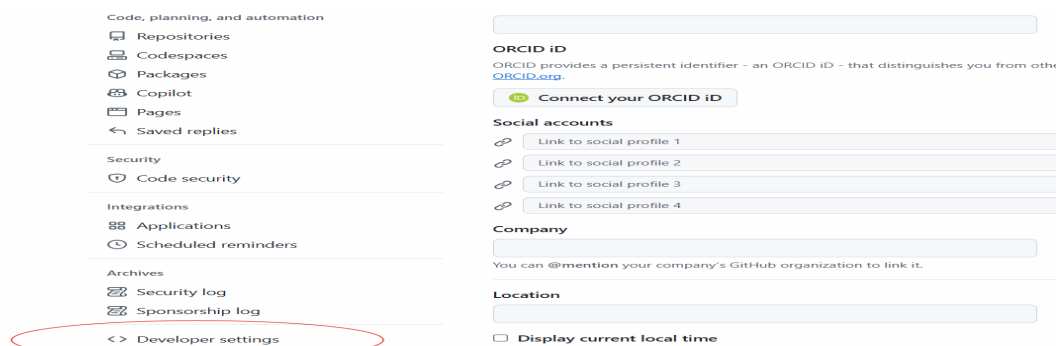
```

11. Go to your profile and **create token** to set url

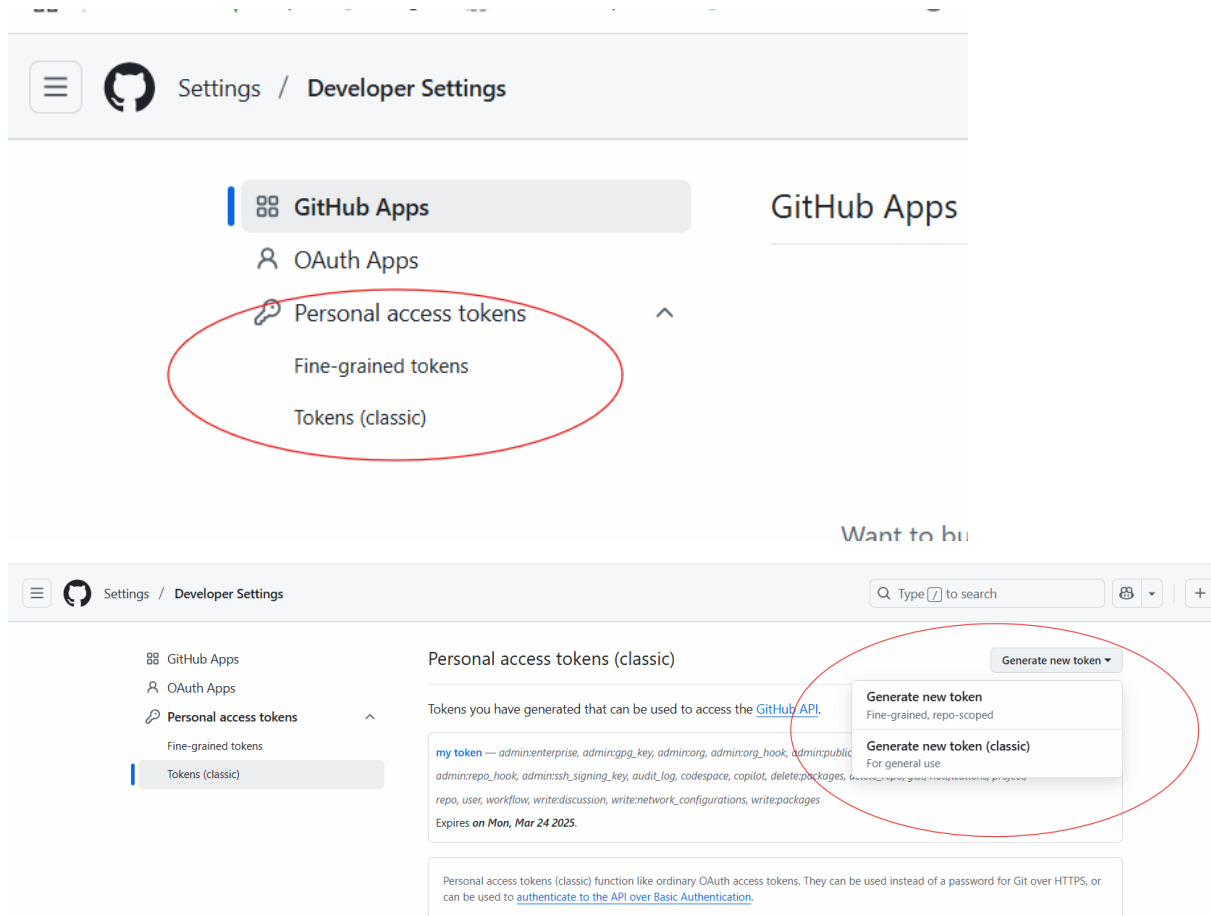
Profile → Settings →



→ Developer Settings

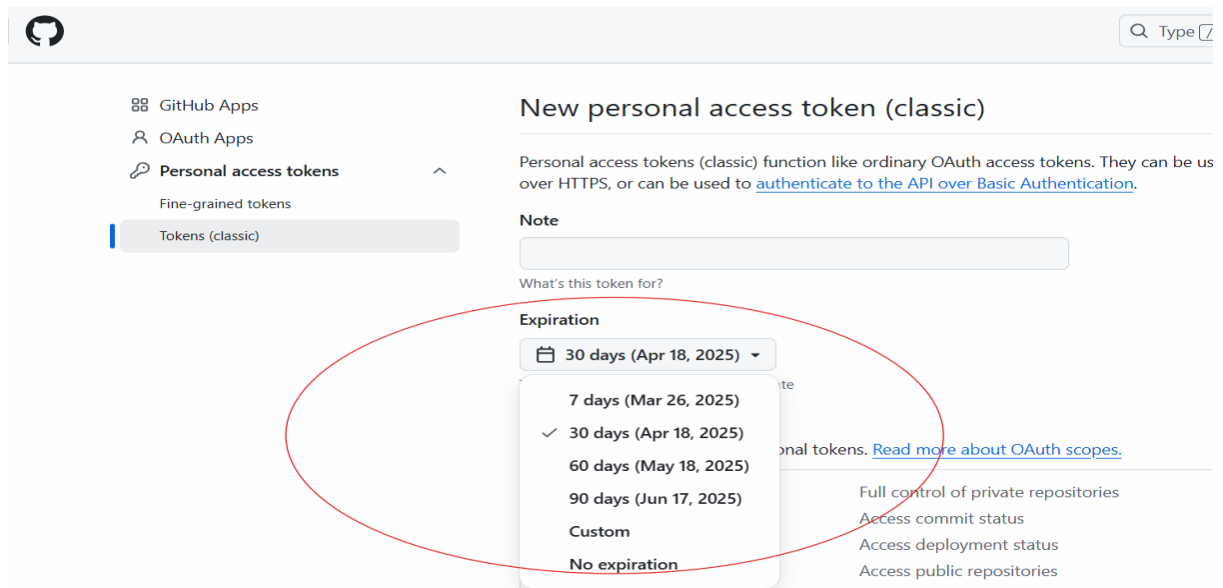


→ Personal access tokens → Tokens (Classic)



→ It will ask you for authentication → Perform Authentication

→ You can set expiration date of your token



→ you can also select "custom".

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens

Tokens (classic)

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

my token

What's this token for?

Expiration **Select date ***

Custom 20-03-2025

The token will expire on the selected date

Select scopes

→ Select all the scopes for your token.

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo:deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input checked="" type="checkbox"/> workflow	Update GitHub Action workflows
<input checked="" type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input checked="" type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input checked="" type="checkbox"/> delete:packages	Delete packages from GitHub Package Registry
<input checked="" type="checkbox"/> admin:org	Full control of orgs and teams, read and write org projects
<input checked="" type="checkbox"/> write:org	Read and write org and team membership, read and write org projects
<input checked="" type="checkbox"/> read:org	Read org and team membership, read org projects
<input checked="" type="checkbox"/> manage_runners:org	Manage org runners and runner groups
<input checked="" type="checkbox"/> admin:public_key	Full control of user public keys
<input checked="" type="checkbox"/> write:public_key	Write user public keys
<input checked="" type="checkbox"/> read:public_key	Read user public keys
<input checked="" type="checkbox"/> admin:repo_hook	Full control of repository hooks
<input checked="" type="checkbox"/> write:repo_hook	Write repository hooks

→ Click on "Generate Token"

<input checked="" type="checkbox"/> admin:pgp_key	Full control of public user GPG keys
<input checked="" type="checkbox"/> write:pgp_key	Write public user GPG keys
<input checked="" type="checkbox"/> read:pgp_key	Read public user GPG keys
<input checked="" type="checkbox"/> admin:ssh_signing_key	Full control of public user SSH signing keys
<input checked="" type="checkbox"/> write:ssh_signing_key	Write public user SSH signing keys
<input checked="" type="checkbox"/> read:ssh_signing_key	Read public user SSH signing keys

Generate token Cancel

→ Your token has been generated

Settings / Developer Settings

Some of the scopes you've selected are included in other scopes. Only the minimum set of necessary scopes has been saved.

Personal access tokens (classic)

Generate new token

Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your personal access token now. You won't be able to see it again!

ghp_kg8sfqeY7e4kJW3PQkv3AApc16zfx61rvcFU

my token: admin:enterprise, admin:pgp_key, admin:org, admin:org_hook, admin:public_key, admin:repo_hook, admin:ssh_signing_key, audit_log, codespace, copilot, delete:packages, delete_repo, gist, notifications, project, repo, user, workflow, write:discussion, writenetwork_configurations, write:packages

Expires on Mon, Mar 24 2025.

Delete

12. Set url using generated token.

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitNew/jasmine (main)
$ git remote add origin https://github.com/SofiDevOps/jasmine.git
```

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitNew/jasmine (main)
$ git remote set-url origin https://ghp_kg8sfqeY7e4kjw3MQkv3AApcI6zfx61rvCFU@github.com/SofiDevOps/jasmine.git
```

13. Perform "Push" operation on main.

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitNew/jasmine (main)
$ git remote set-url origin https://ghp_kg8sfqeY7e4kjw3MQkv3AApcI6zfx61rvCFU@github.com/SofiDevOps/jasmine.git

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitNew/jasmine (main)
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 255 bytes | 255.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/SofiDevOps/jasmine.git
7363a93..baca4aa  main -> main
```

14. Refresh your Git hub and check the content of your file.

The screenshot shows the GitHub repository 'jasmine' (Public) forked from 'AquilaShaikh/jasmine'. The 'main' branch is selected, showing '1 Branch' and '0 Tags'. A commit by 'AquilaShaikh' is highlighted, titled 'adding update', with commit hash 'baca4aa' and '3 Commits' ago. The file 'test.txt' is shown with its content:

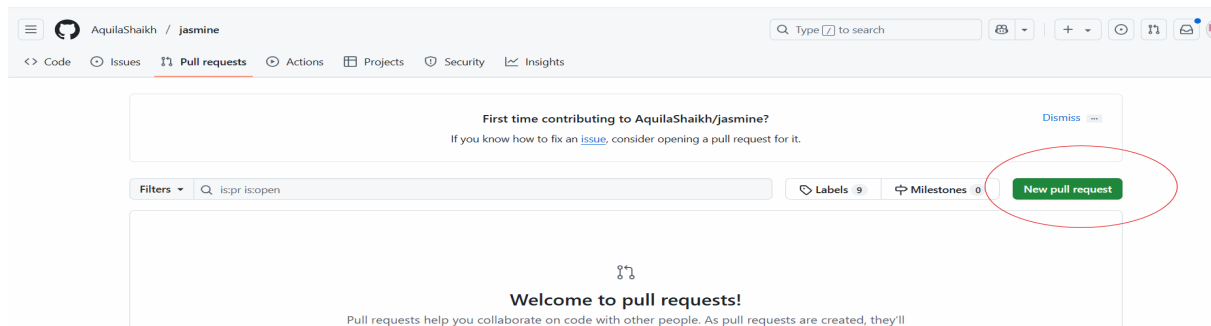
```
1 hello
2 hi
3 hi
4 hi
5 hello
6 hello
```

15. Create Pull request from second account.

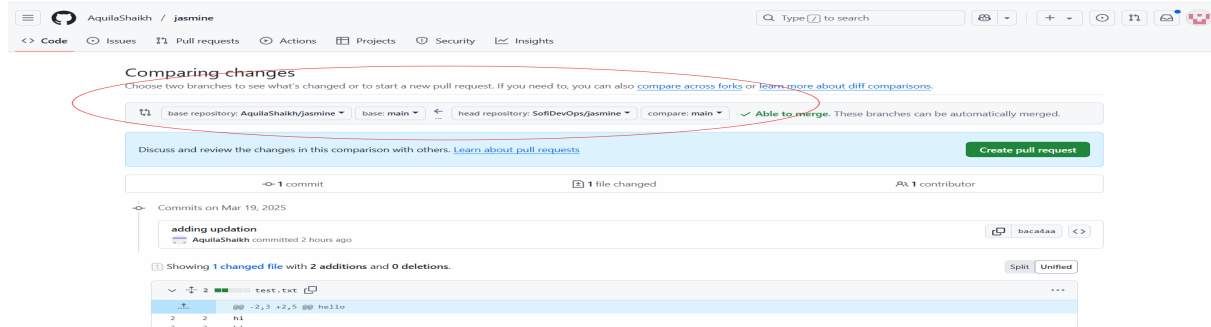
The screenshot shows the GitHub repository 'jasmine' (Public) with a pull request created by 'AquilaShaikh'. The pull request is titled 'added' and shows '2 Commits' ago. The file 'test.txt' is shown with its content:

```
1 hello
2 hi
3 hi
4 hi
5 hello
6 hello
```

16. Click on new Pull request.

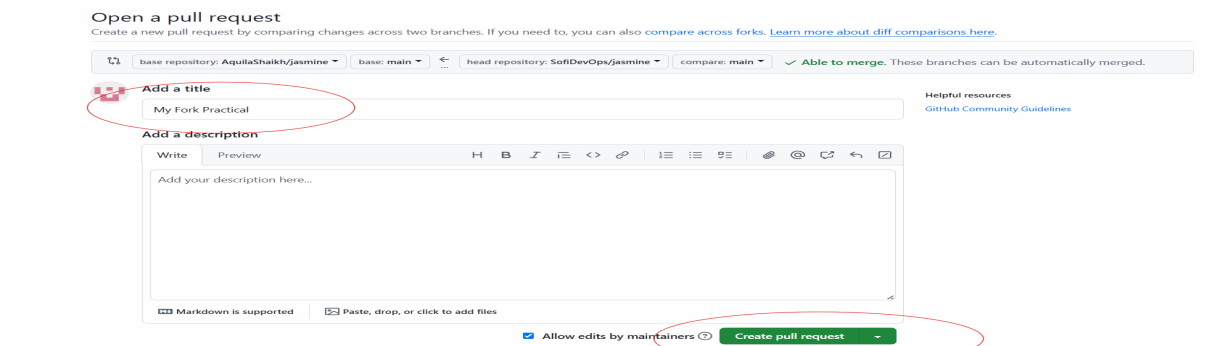


17. Select proper branches and repository

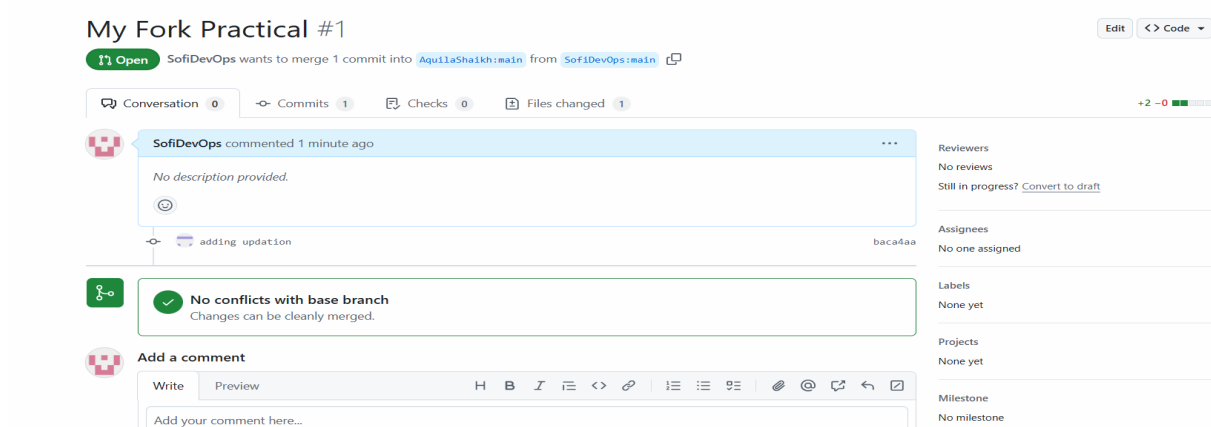


18. Create Pull request

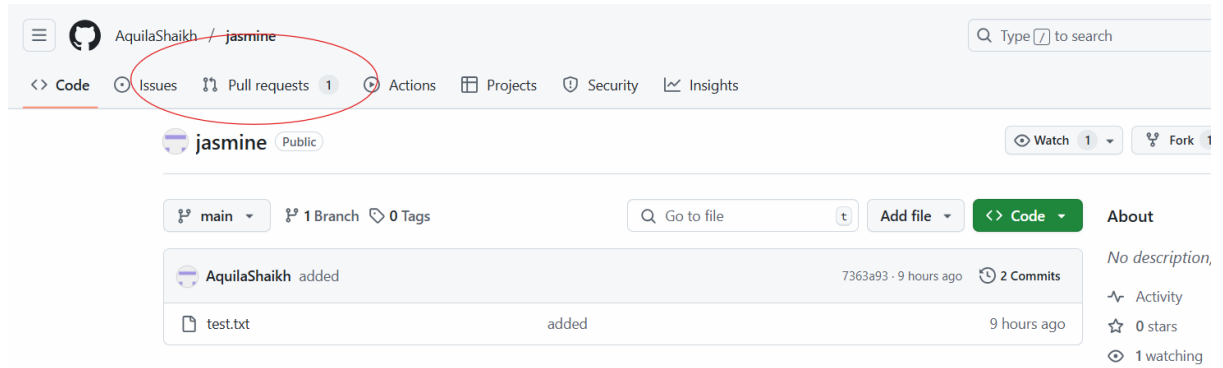
19. Give title and click on "Create pull request"



20. Your Pull request has been generated.



21. Check your first account and repository for Pull request



22. Click on Merge request.