**Clone vs Fork in GitHub**

The functionality clone and Fork is same they copies the repository’s. But the different is

**Clone:**  Cloning is used to create a local copy of a repository on your computer. That means it downloads a copy of a repository to your local machine.

**Fork:**  Forking is used to create a personal copy of someone else's repository under your own GitHub account. That means it creates a copy of a repository under your GitHub account.

| **Feature** | **Fork** | **Clone** |
| --- | --- | --- |
| **What It Does** | Creates a copy of a repository **on GitHub** under your account. | Creates a copy of a repository **on your local machine**. |
| **Where It Happens** | On **GitHub's website**. (Forking is done on **GitHub's web interface.)** | On your local **computer** using Git commands.  (Cloning is done using the**Git command-line tool** or a Git client.) |
| **Ownership** | The forked repository belongs to **your GitHub account** and you have full control over it. | The cloned repository is just a **local copy**; ownership remains unchanged. |
| **Purpose** | Used to **contribute to open-source projects** or experiment with someone else's code. | Used to **work on the repository locally** (edit, test, or run the code). |
| **Link to Original** | Maintains a **link to the original repository** (you can sync updates). | No link to the original repository; it’s just a standalone copy. |
| **GitHub-Specific** | Yes, forking is a **GitHub feature**. | No, cloning is a **Git feature** (works with any Git repository). |
| **Example** | You fork userA/project to your-username/project. | You clone your-username/project to your computer. |
| **Use Case** | - Contributing to open-source projects. - Experimenting with someone else's code. | - Working on the code locally. - Making changes and testing. |

Let’s work with FORK in GitHub practically.

For this we require two GitHub accounts, I have two accounts “sheriharish” and “Harish-1000”.

Let’s choose a repository (static-website) from sheriharish GitHub account. And fork this repository from the Harish-1000 GitHub account.

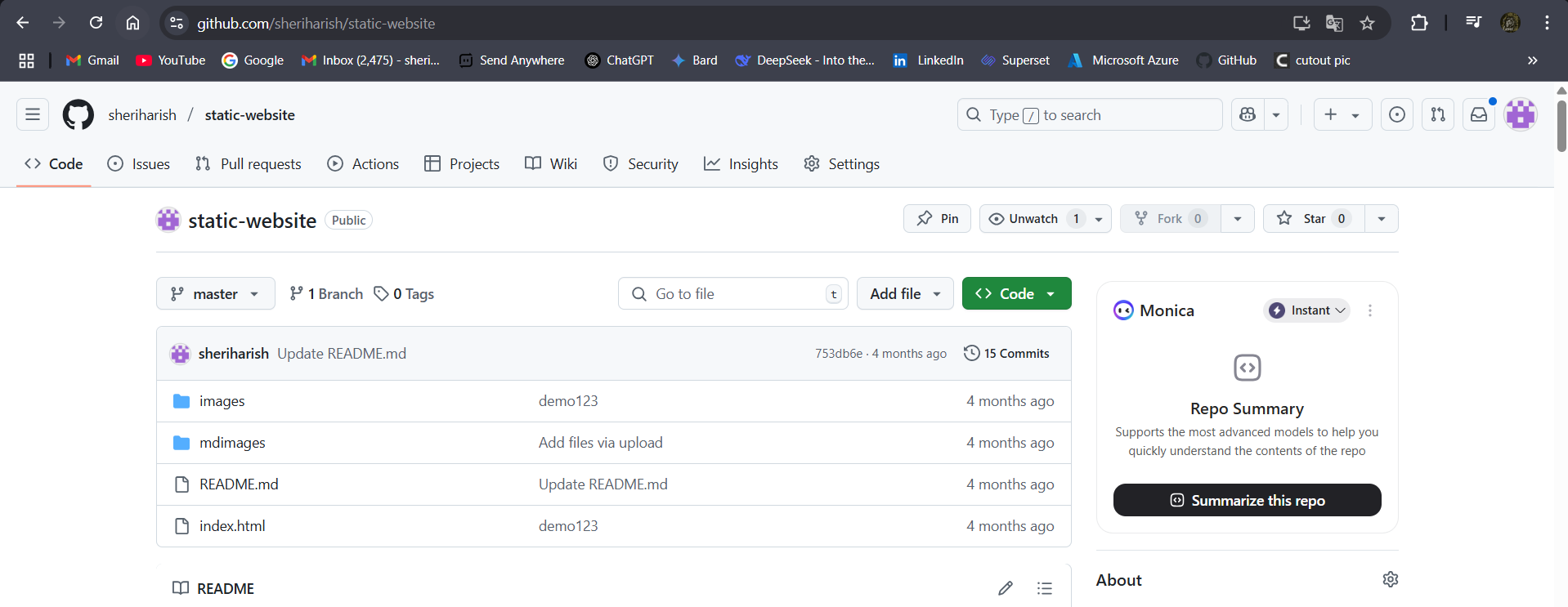


Fig: static-website repository of sheriharish GitHub account.

The static-website repository has “image”, “mdimages” folders and “README.md”, “index.html” are the files.

Now Fork this repository from the Harish-1000 GitHub account.

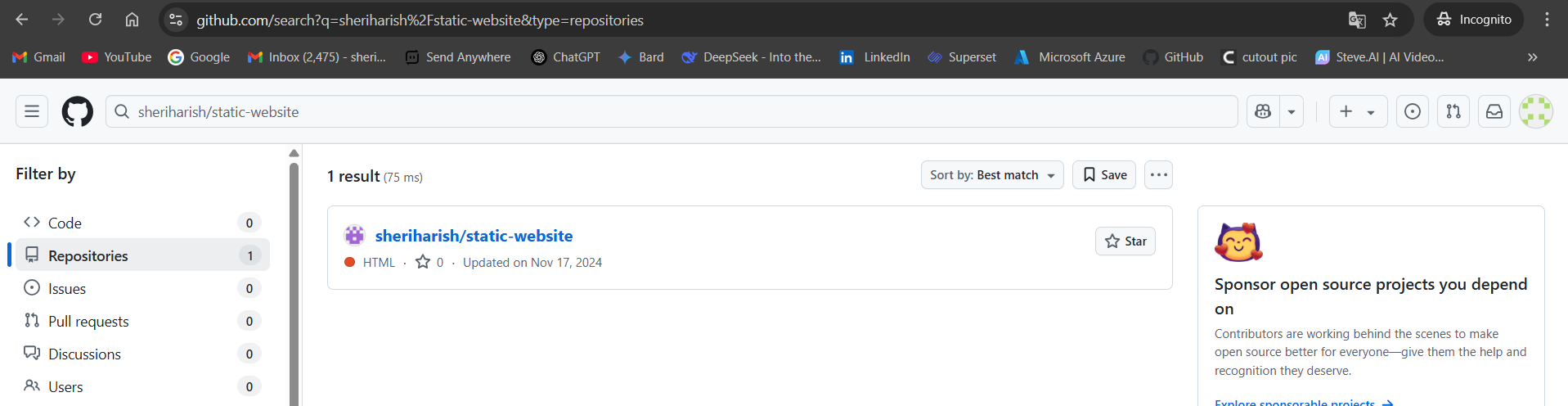
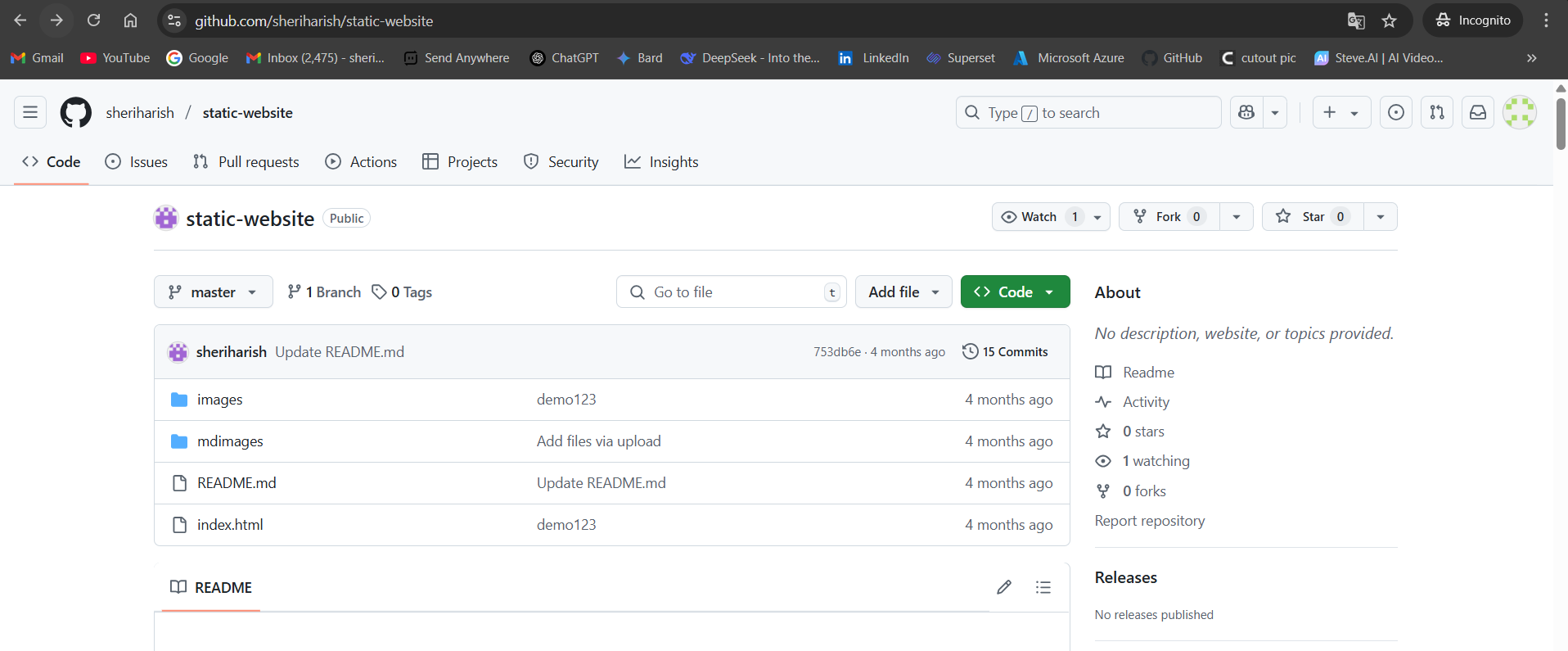


Fig: Search the repository (sheriharish/static-website) from the Harish-1000 account.



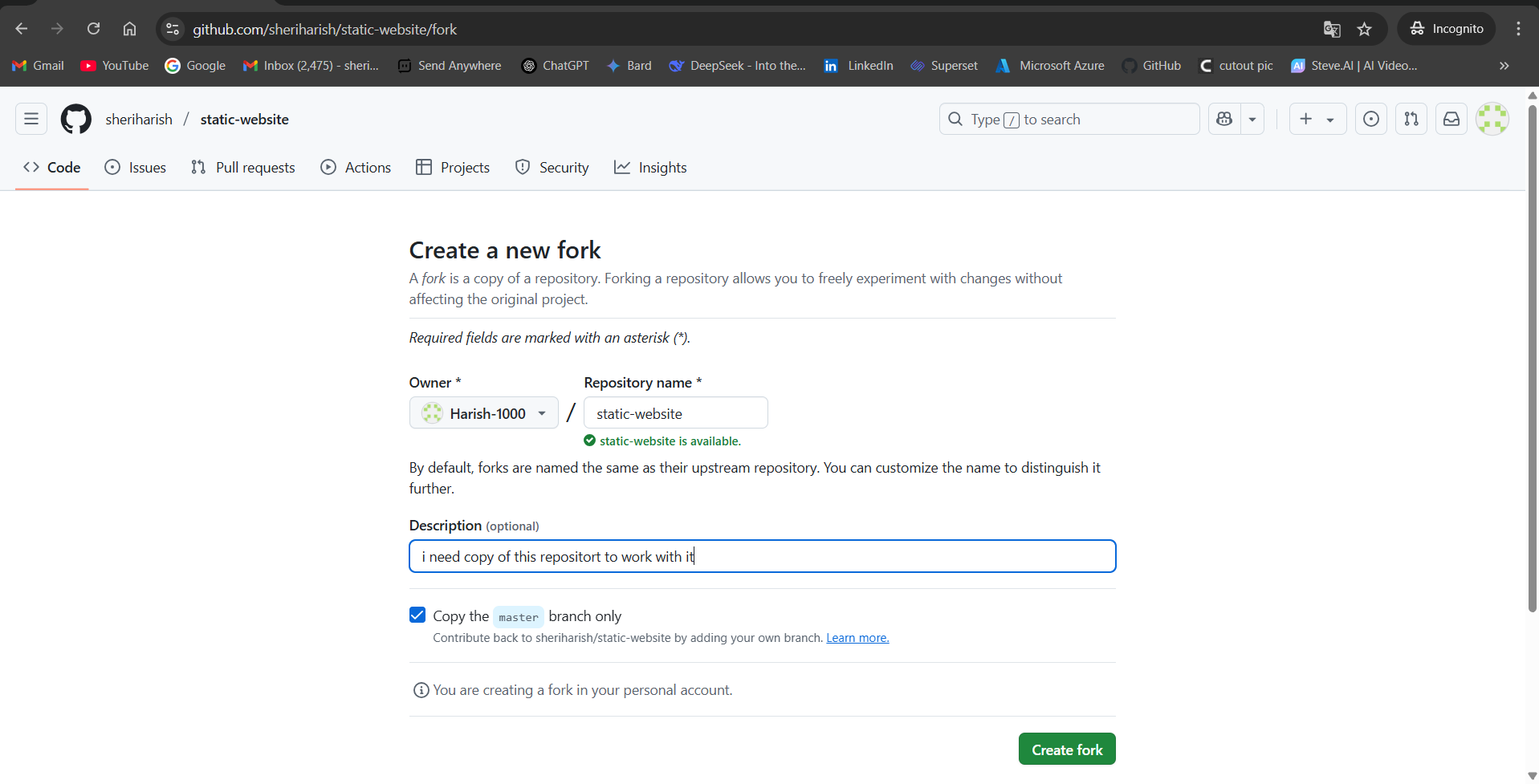
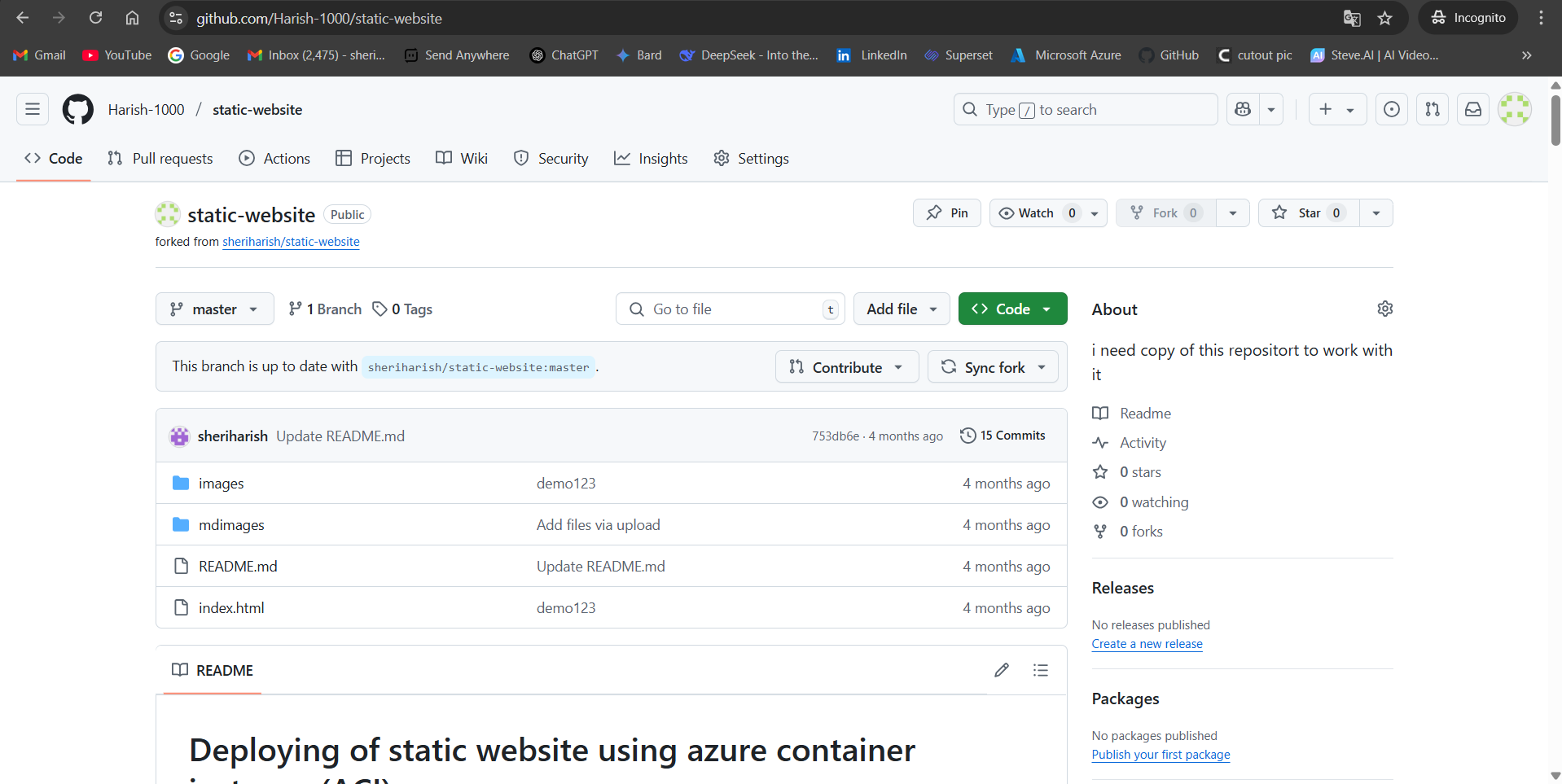
Fig: Click on Fork.

Fig: Create a Fork of repository (static-website).

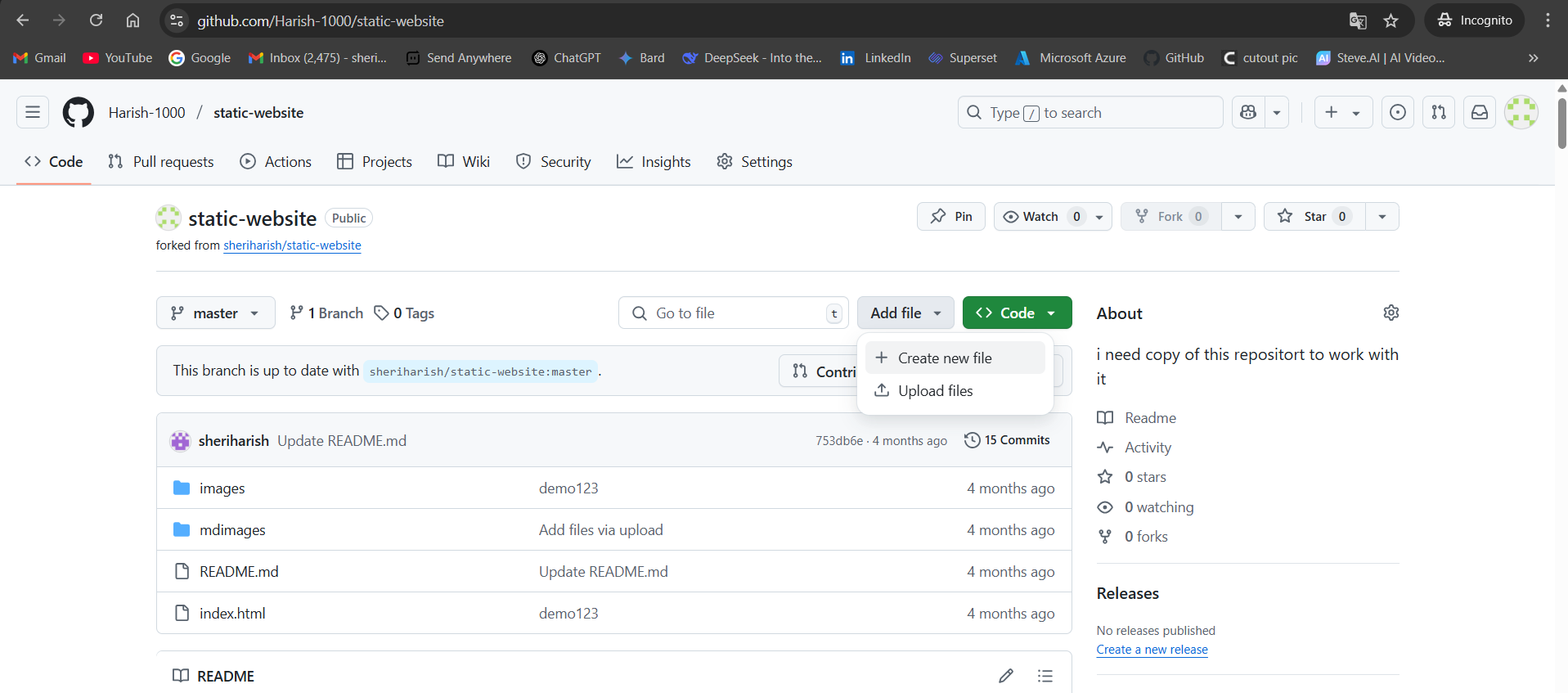


**Fig:** Forking is done successfully.

In above figure the static-website repository is successfully forked from the sheriharish GitHub account to the Harish-1000 GitHub account. As a result the ownership is changed from sheriharis to the Harish-1000.

Now let’s create a text file in the fork repository and merge it to the original repository (sheriharish/static-website) as a contributor, then we have to create a pull request.

Step1: Create file in the fork repository (Harish-1000/static-website).



Step2: Write any content in it and commit it.

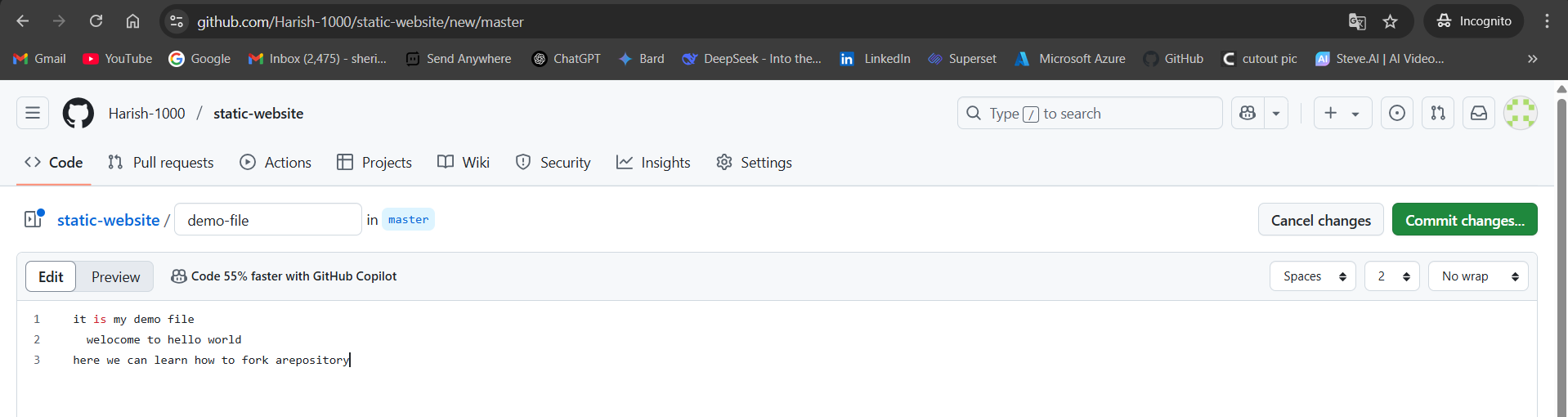


Fig: Demo file.

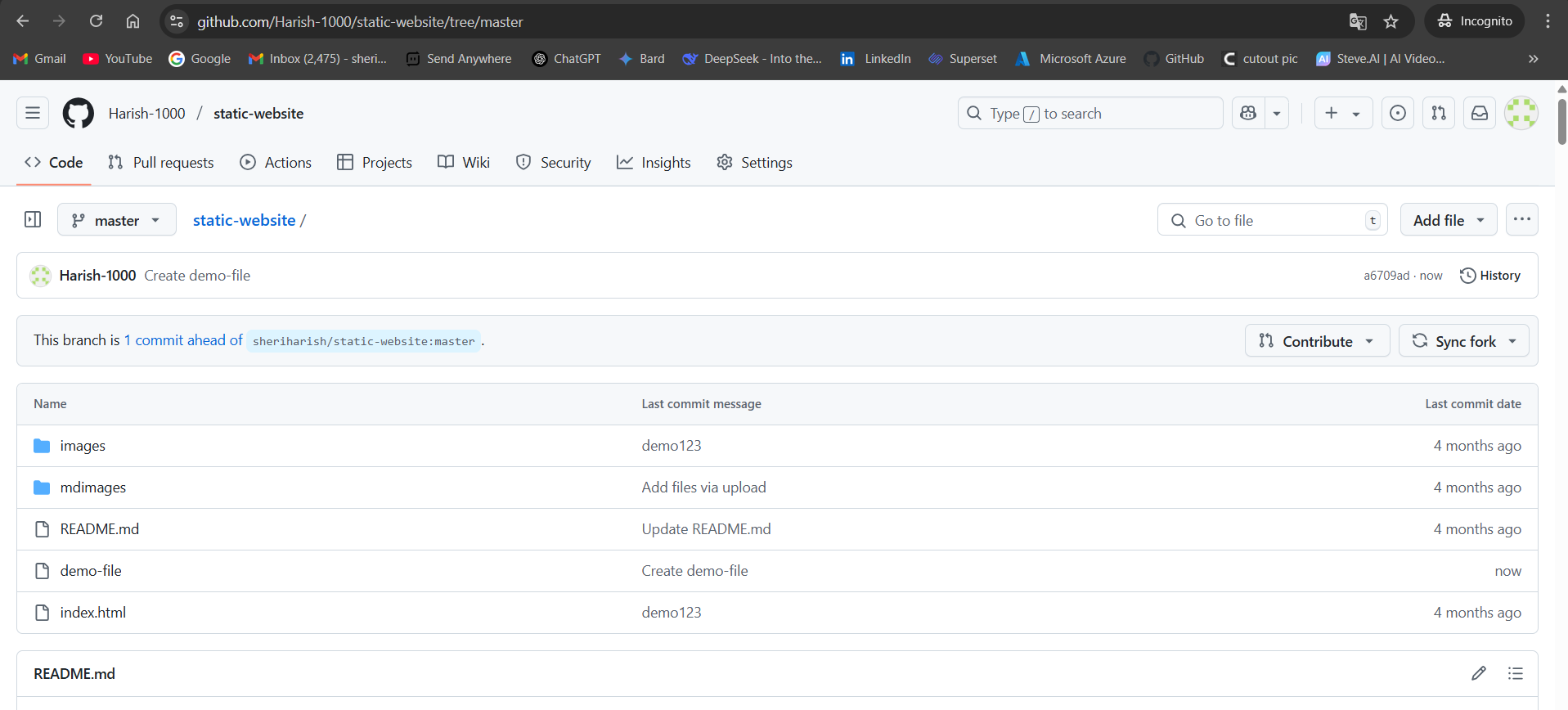
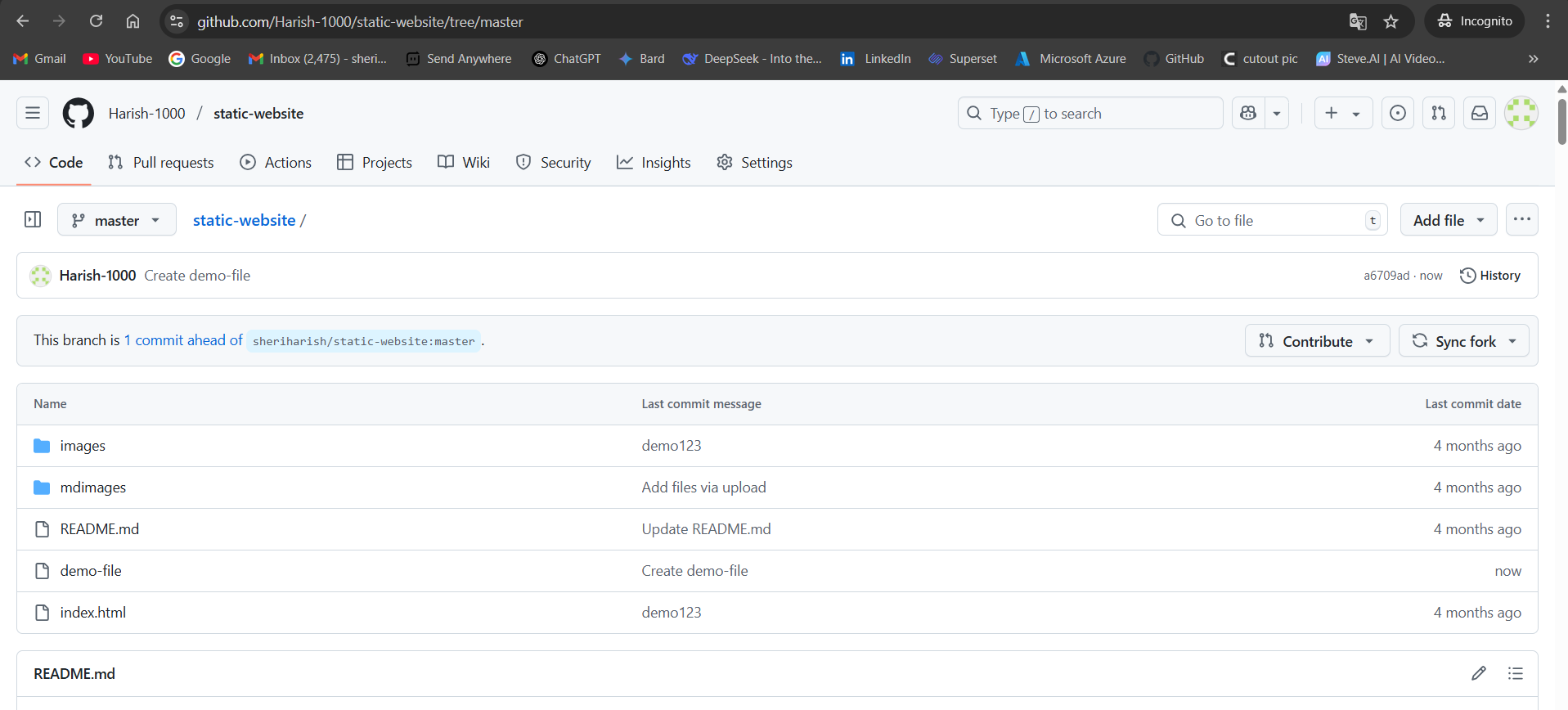


Fig: file (demo-file) is created successfully in Harish/static-website repository.

In order to merge this new change in the original repository (sheriharish/static-website) we have to create a pull request to the main owner of this static-website repository (sheriharish).

Step3: Create a pull request to merge the changes to the original repository.



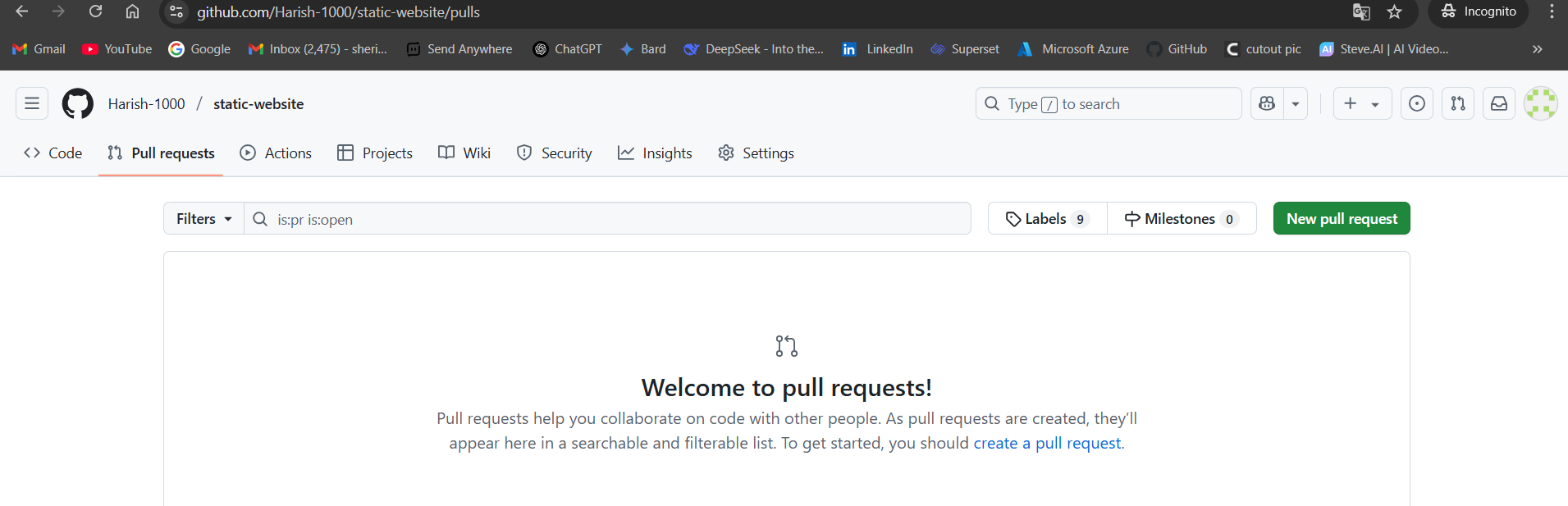
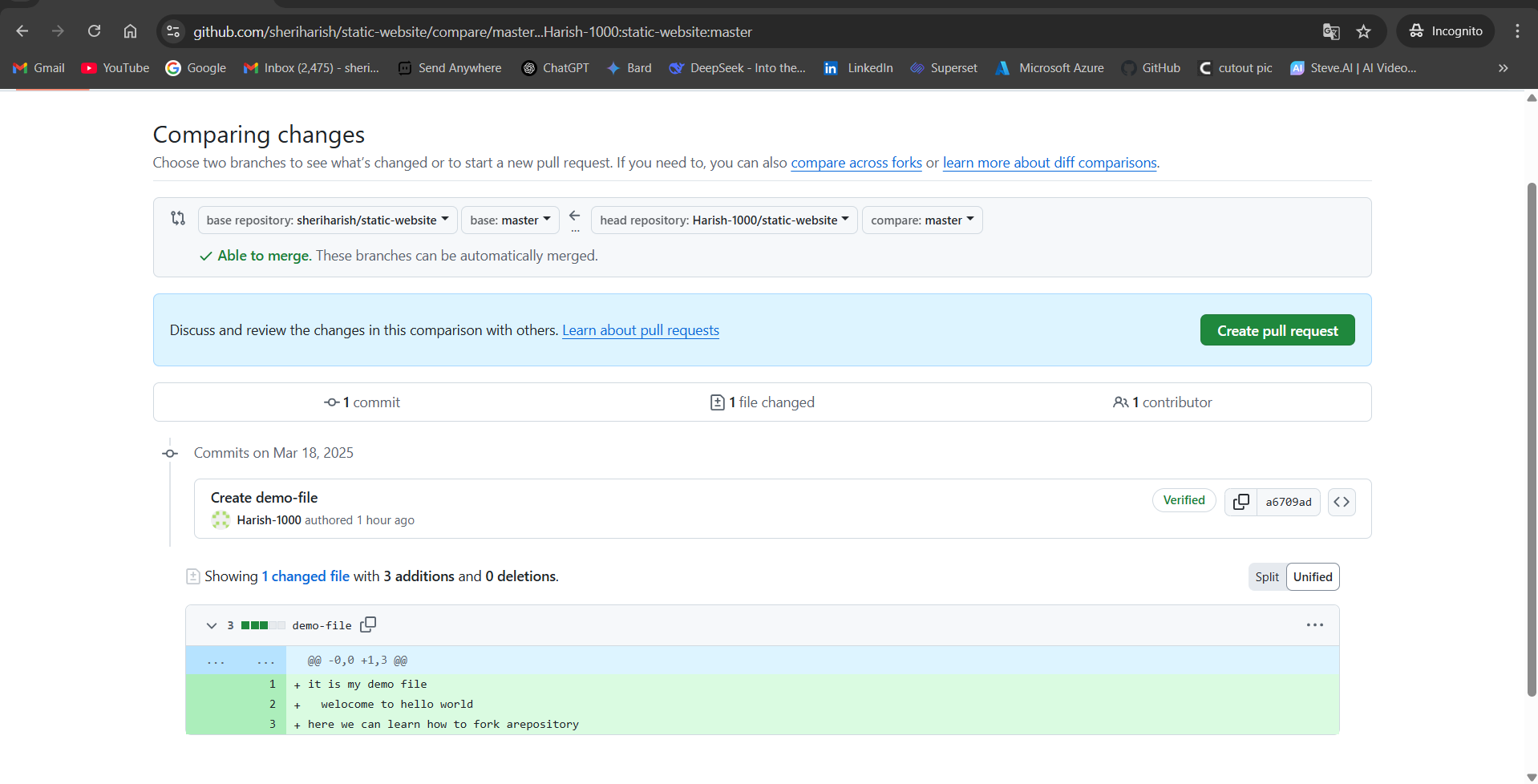


Fig: Create a new pull request.



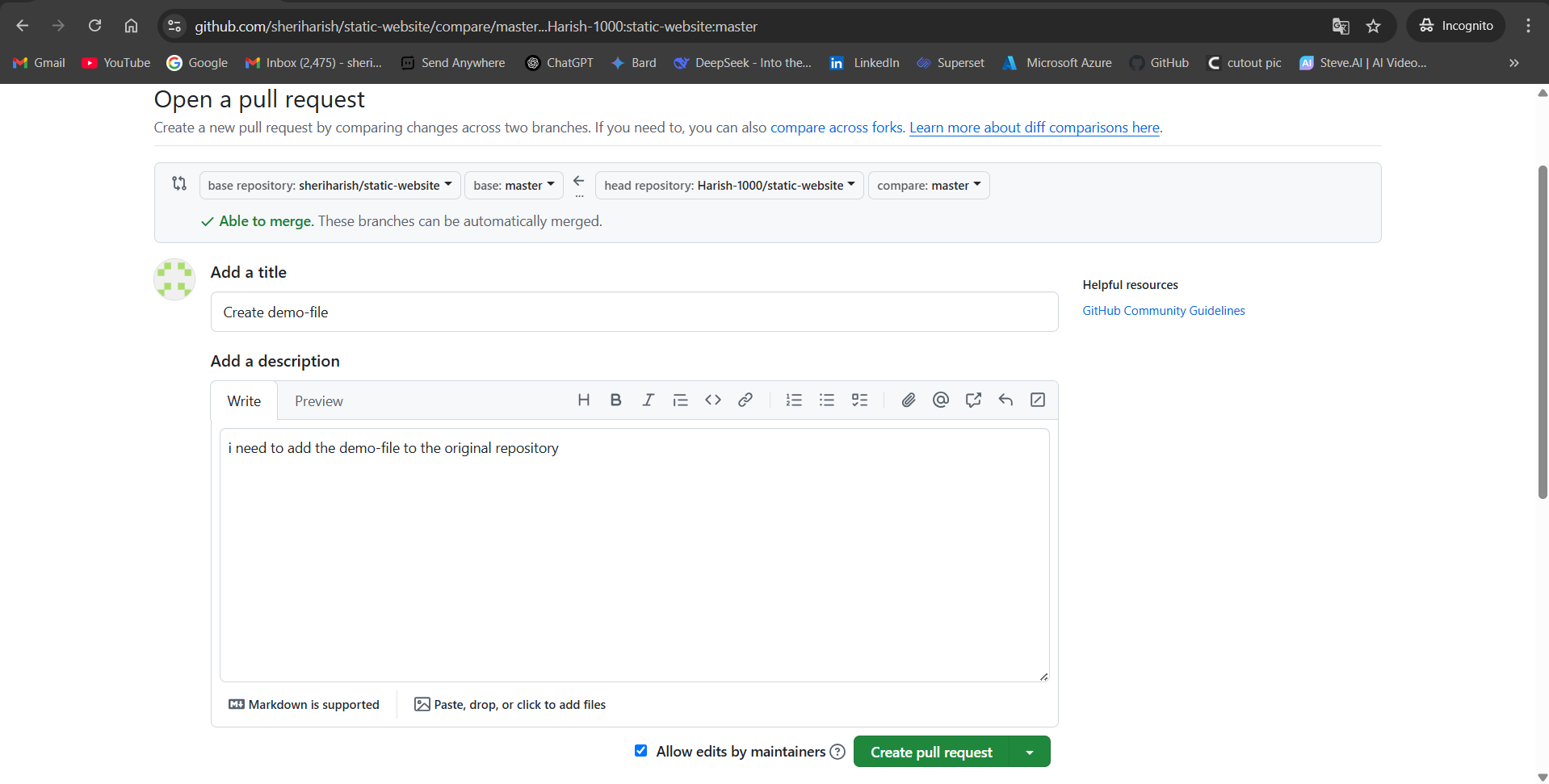
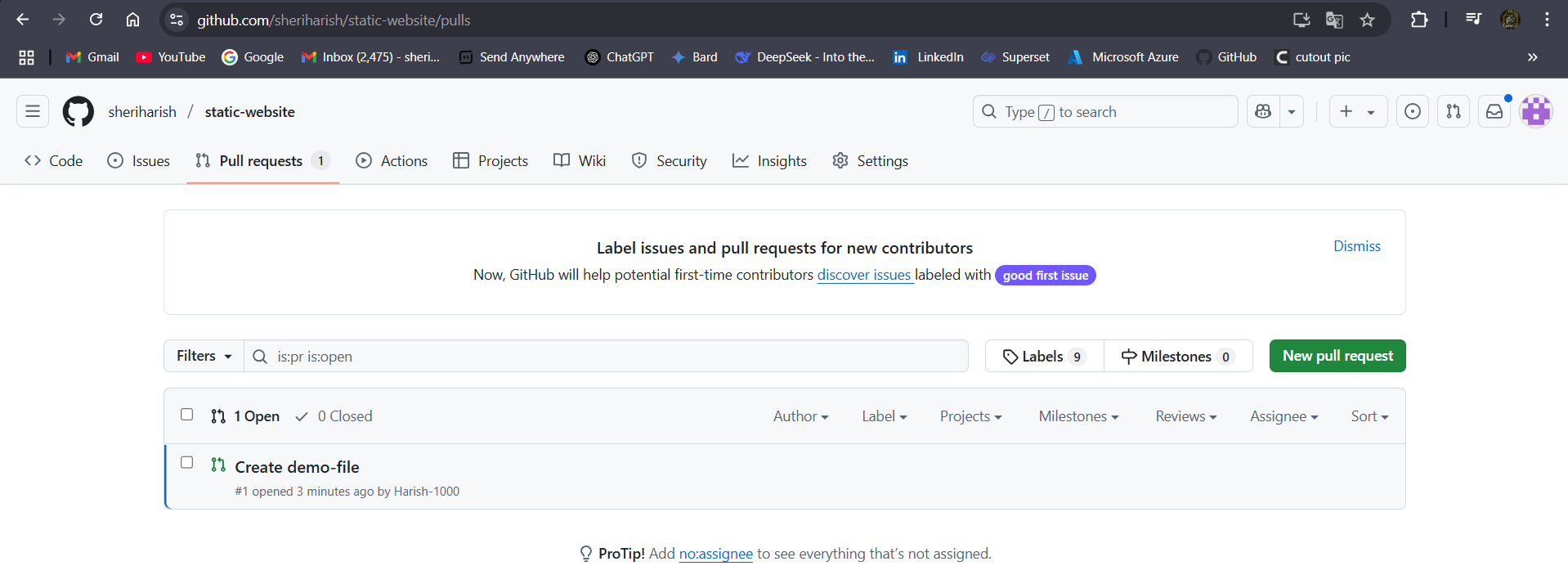


Fig: Add any description and create pull request.

After creating a pull request from the fork repository account (Harish-1000/static-website), we will get a pull request notification to the original main account sheriharish/static-website as show in below figure.



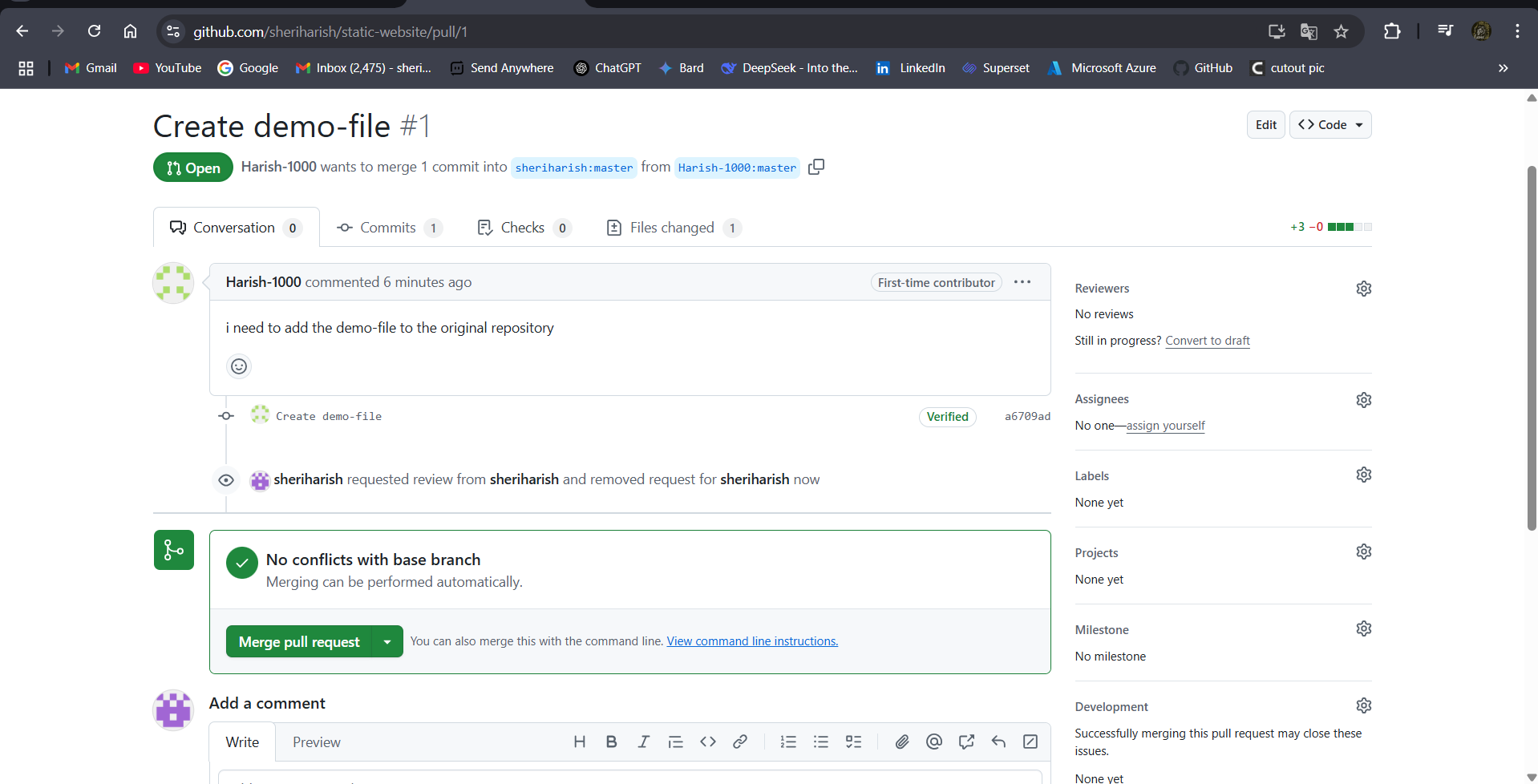


Fig: merge the pull request.

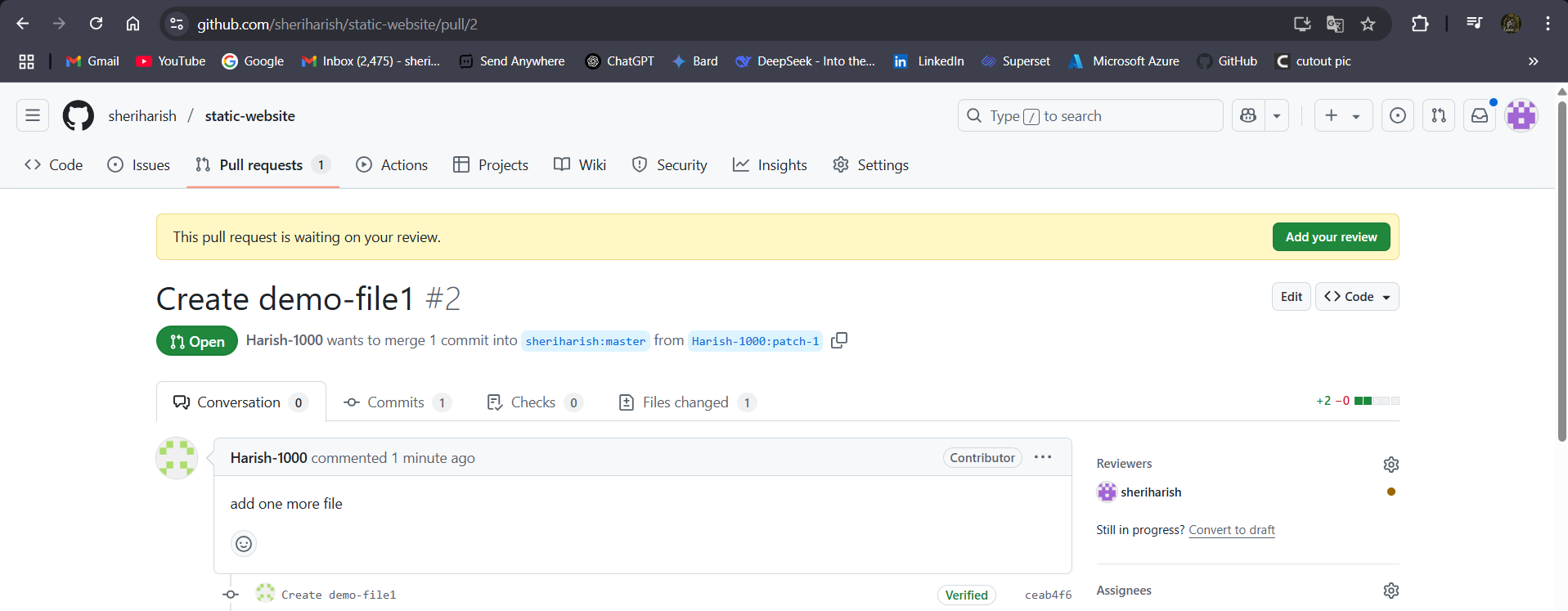


Fig: Add a review for the pull request.

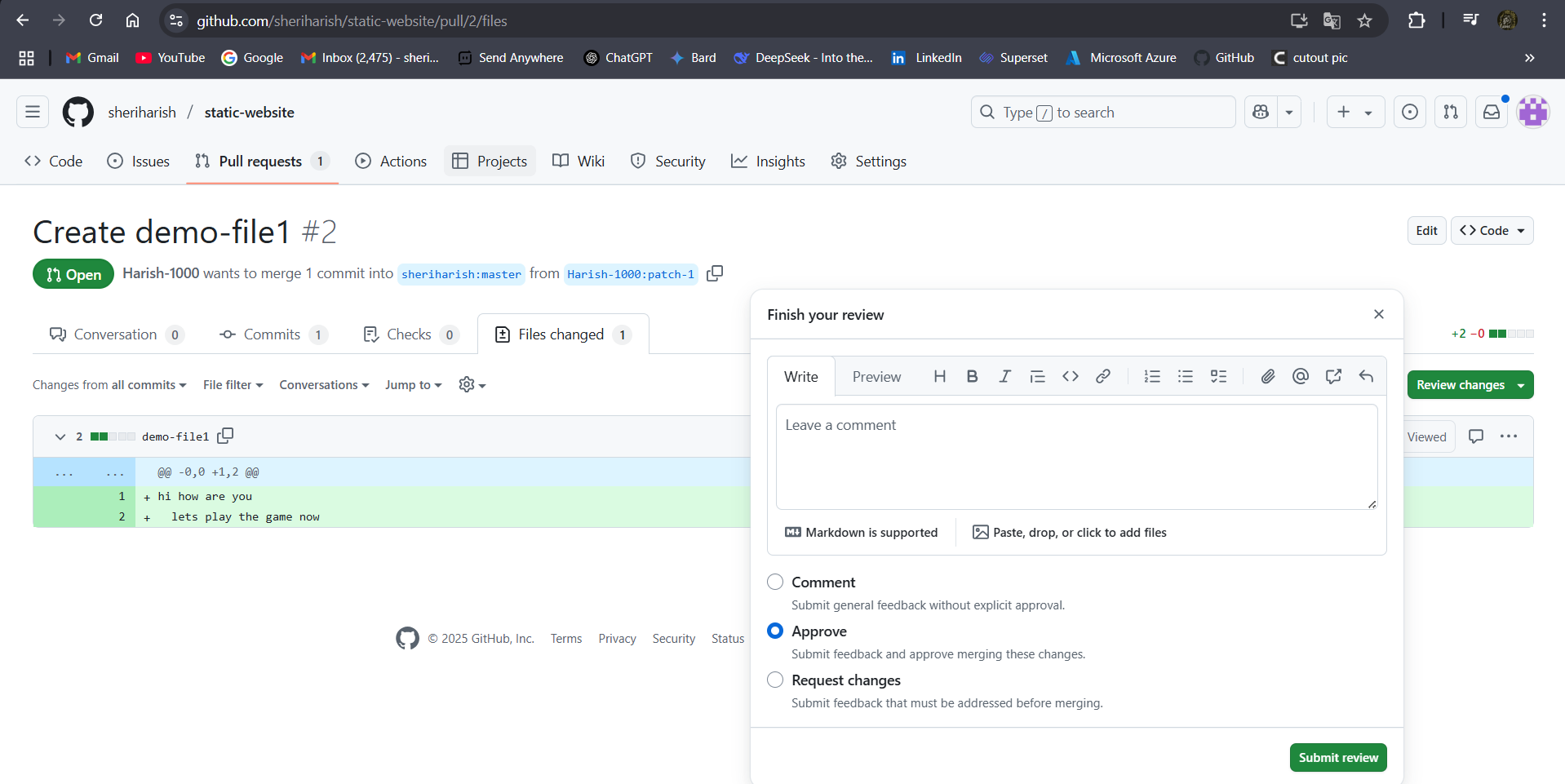


Fig: Approve the pull request.

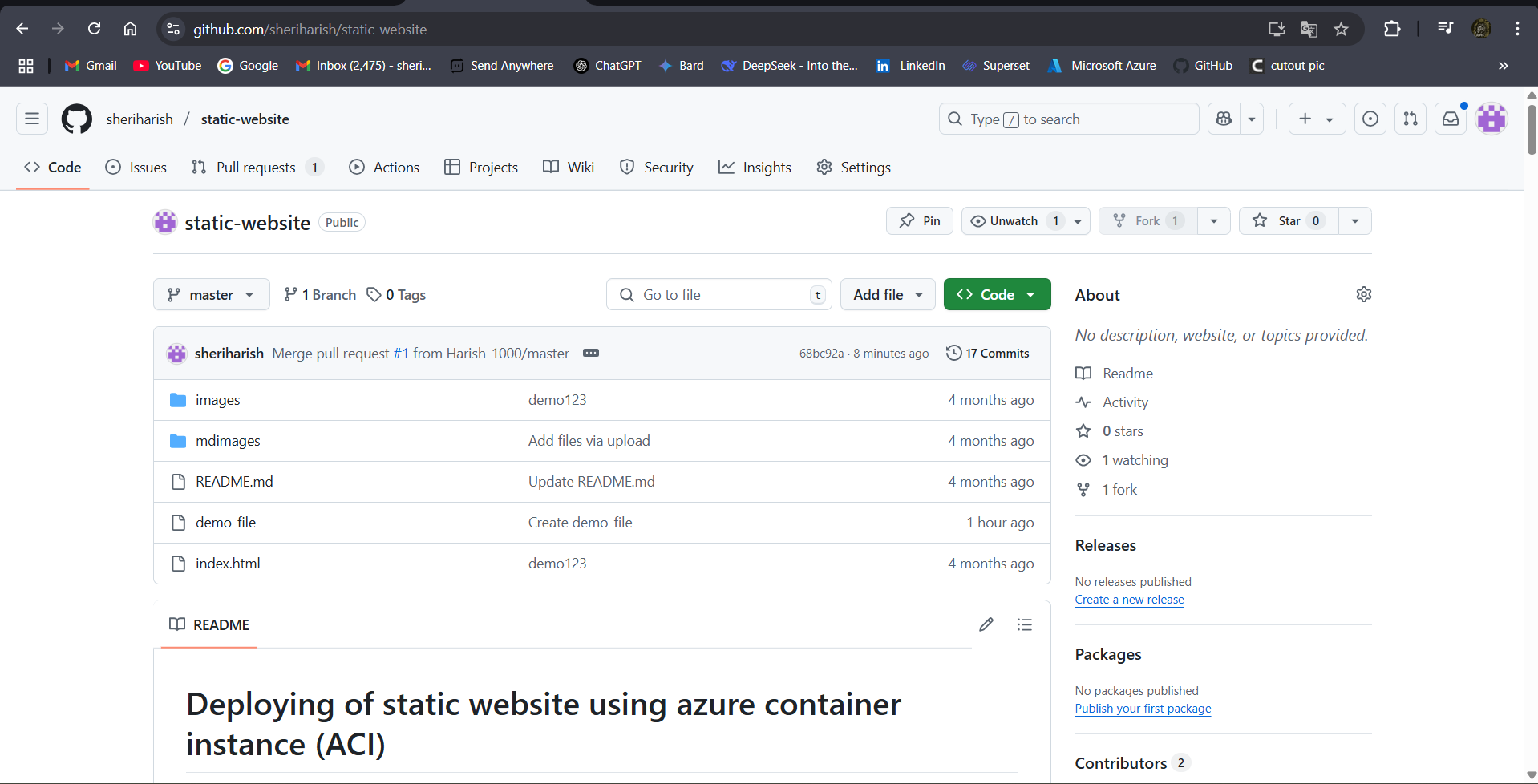


Fig: The changes are merged to the original repository successfully.