**Software Engineering Tools Lab Assignment No-4**

**Sumit Narake**

**2020BTECS00023**

Q 1. Create a repository on GitHub named SET Lab and add files into it (you can add implementation files of previous assignment) perform below operations on it. (Add screenshot as an answer to every question)

1. Perform commit on added files

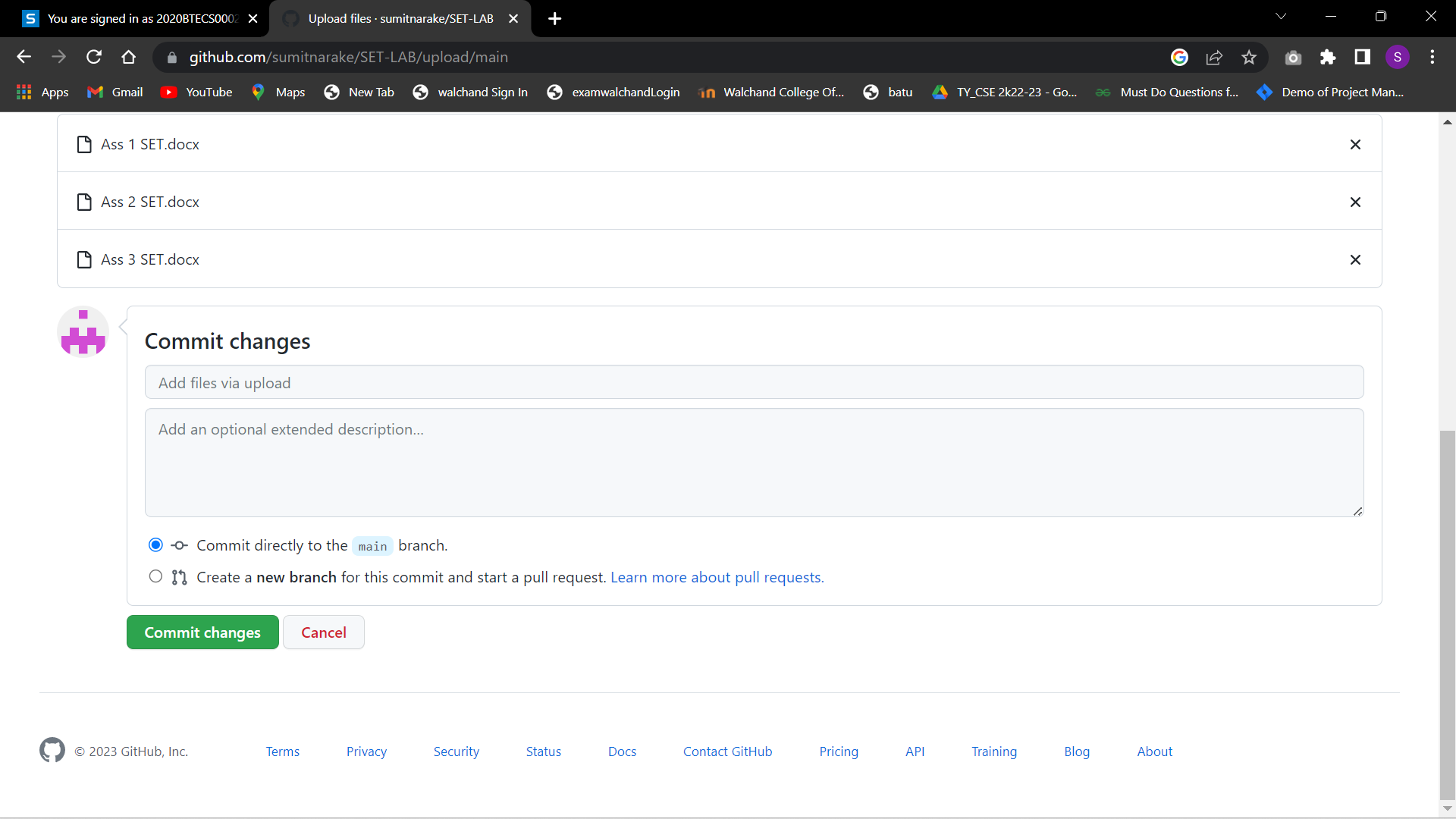
● Go to the main page of your repository on GitHub

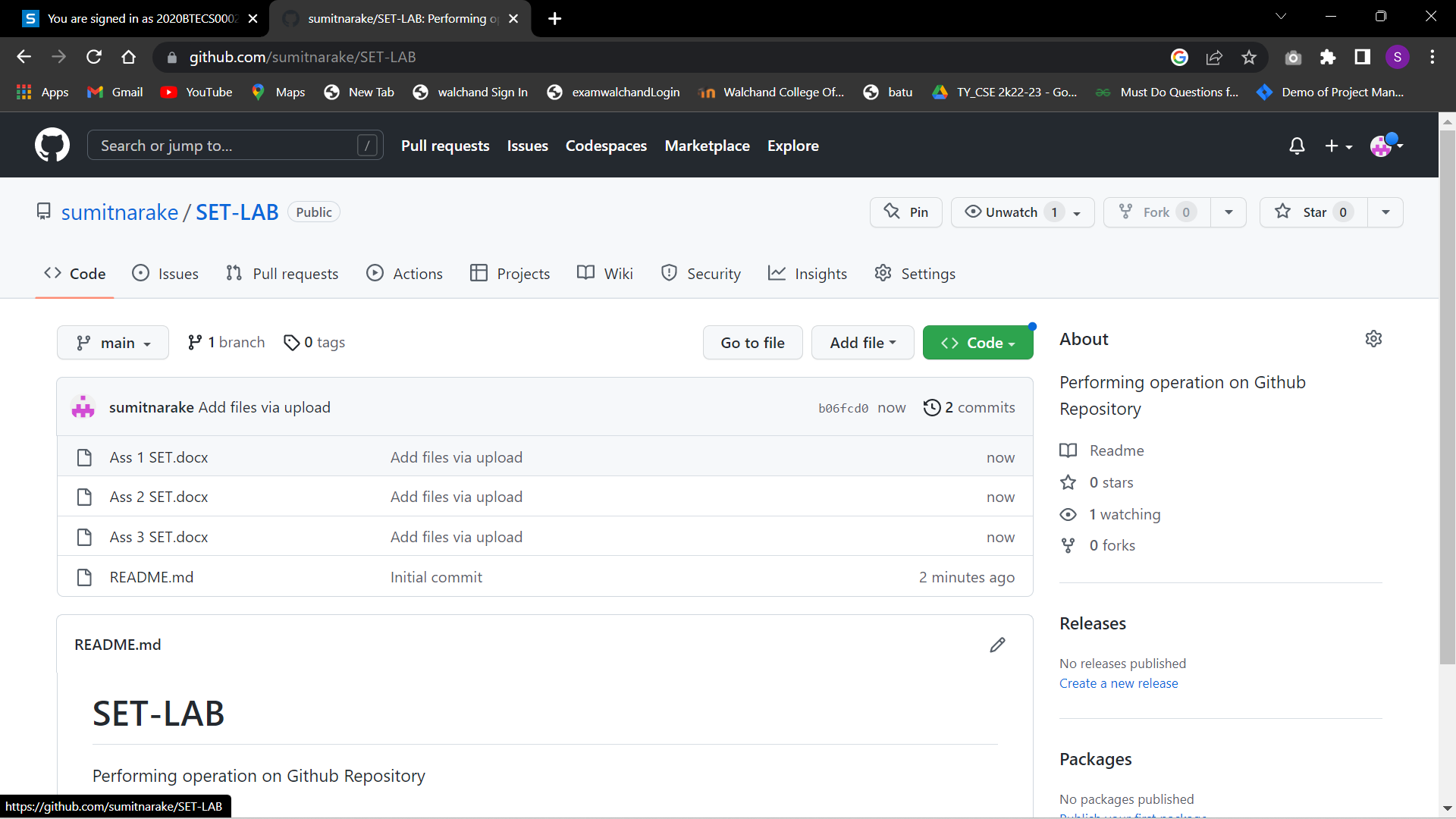
● Click on the file you want to commit changes to.

● Click on the "Edit" button to make changes to the file.

● Enter a commit message to describe the changes you are making.

● Click on the "Commit changes" button.





1. Perform update to the existing files (show history)

● Go to the main page of your repository on GitHub.

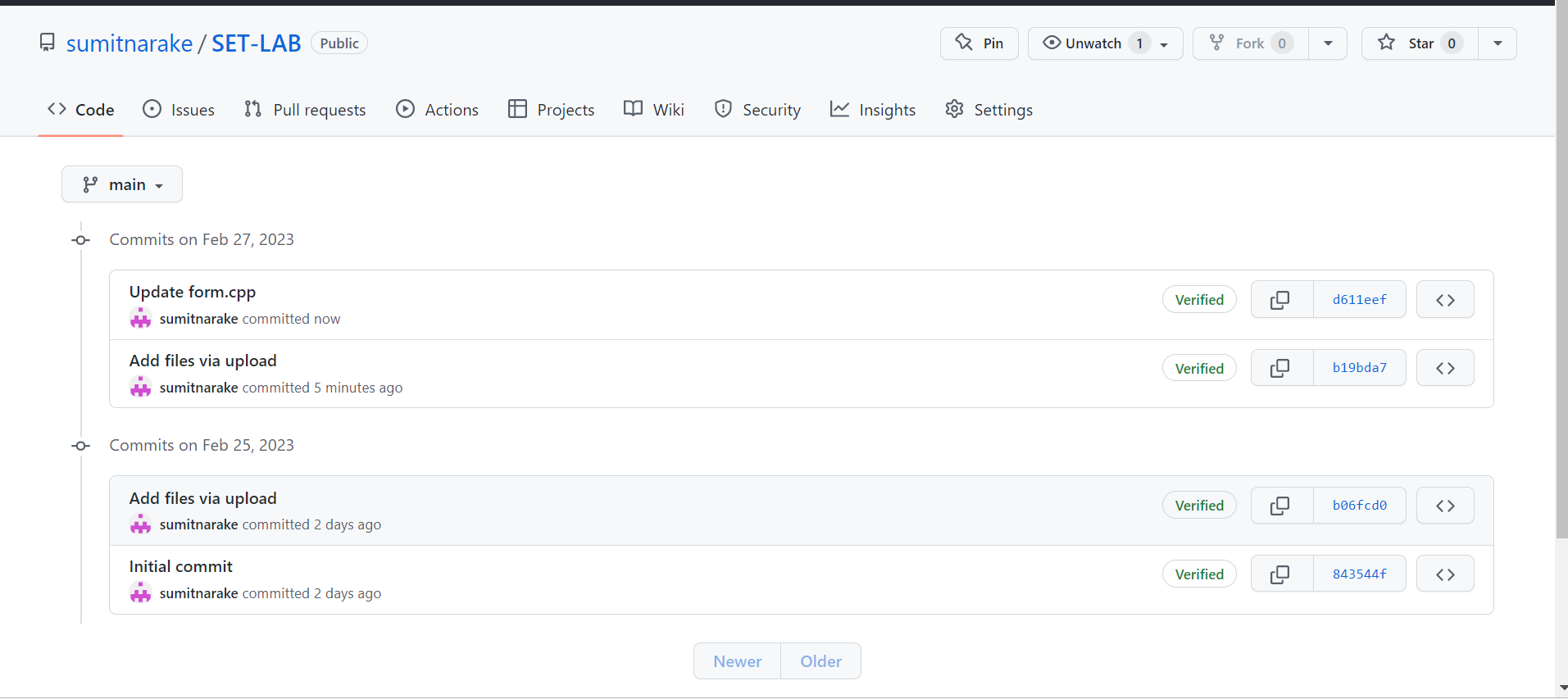
● Click on the file you want to update.

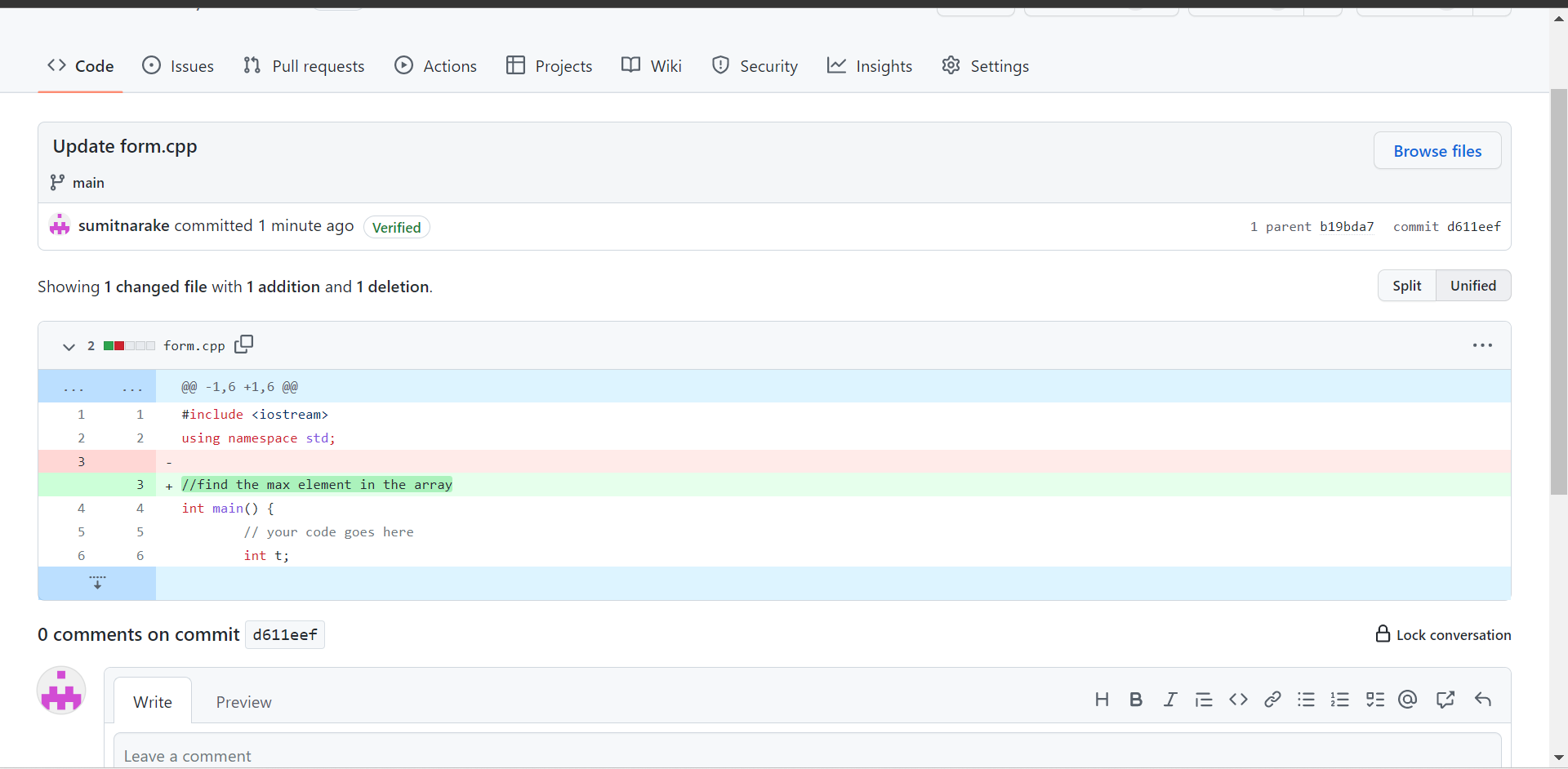
● Click on the "Edit" button to make changes to the file.

● Enter a commit message to describe the changes you are making.

● Click on the "Commit changes" button.

● To view the history of changes, click on the "History" tab





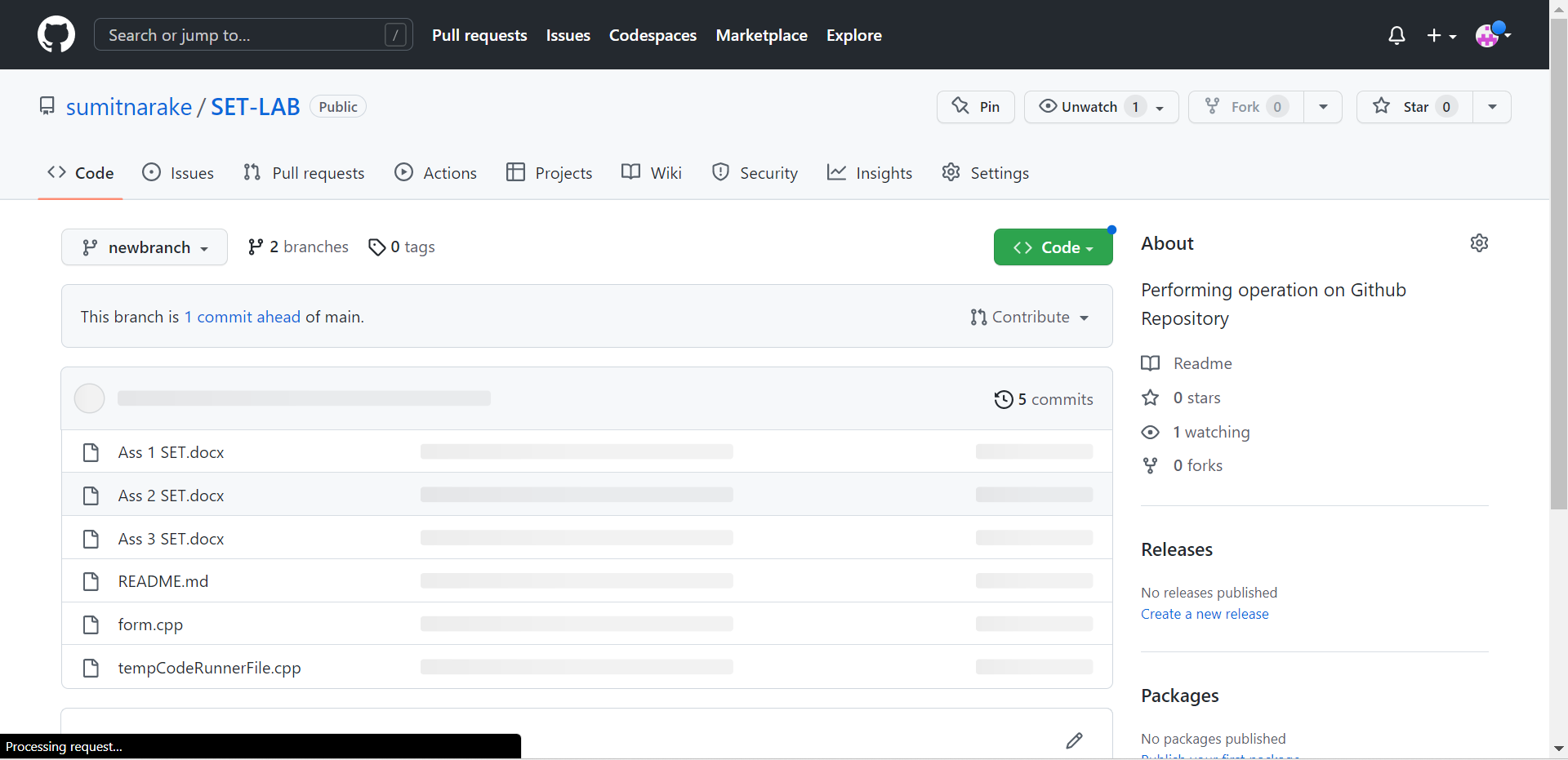
1. Create another branch

To create another branch, follow these steps:

● Go to the main page of your repository on GitHub.

● Click on the "Branch: master" dropdown menu.

● Enter a name for your new branch and click on the "Create branch button.



1. Create pull request

To create a pull request, follow these steps:

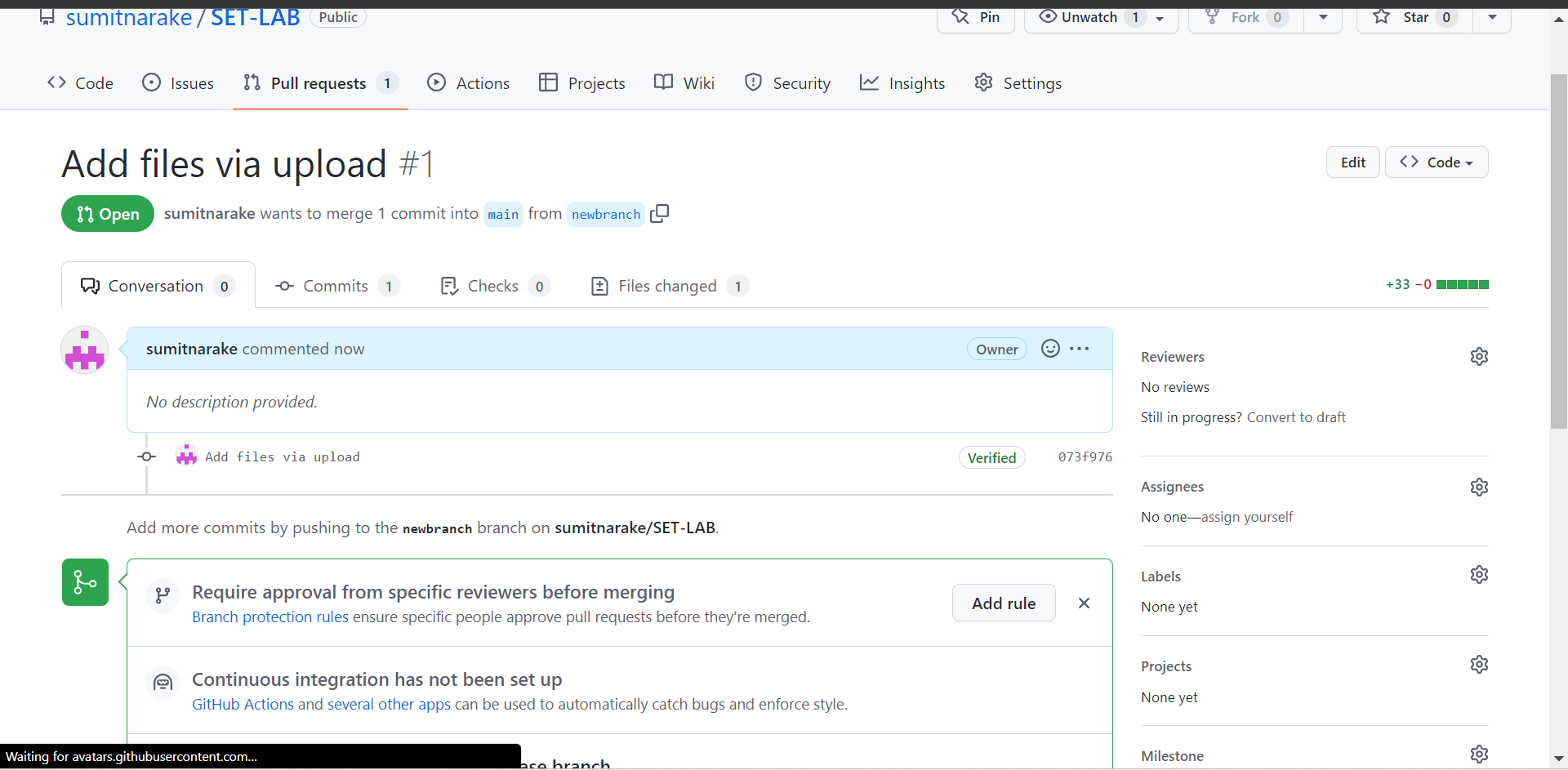
● Go to the main page of your repository on GitHub.

● Switch to the branch you want to merge into the master branch.

● Click on the "Compare & pull request" button.

● Enter a title and a description for your pull request.

● Click on the "Create pull request" button



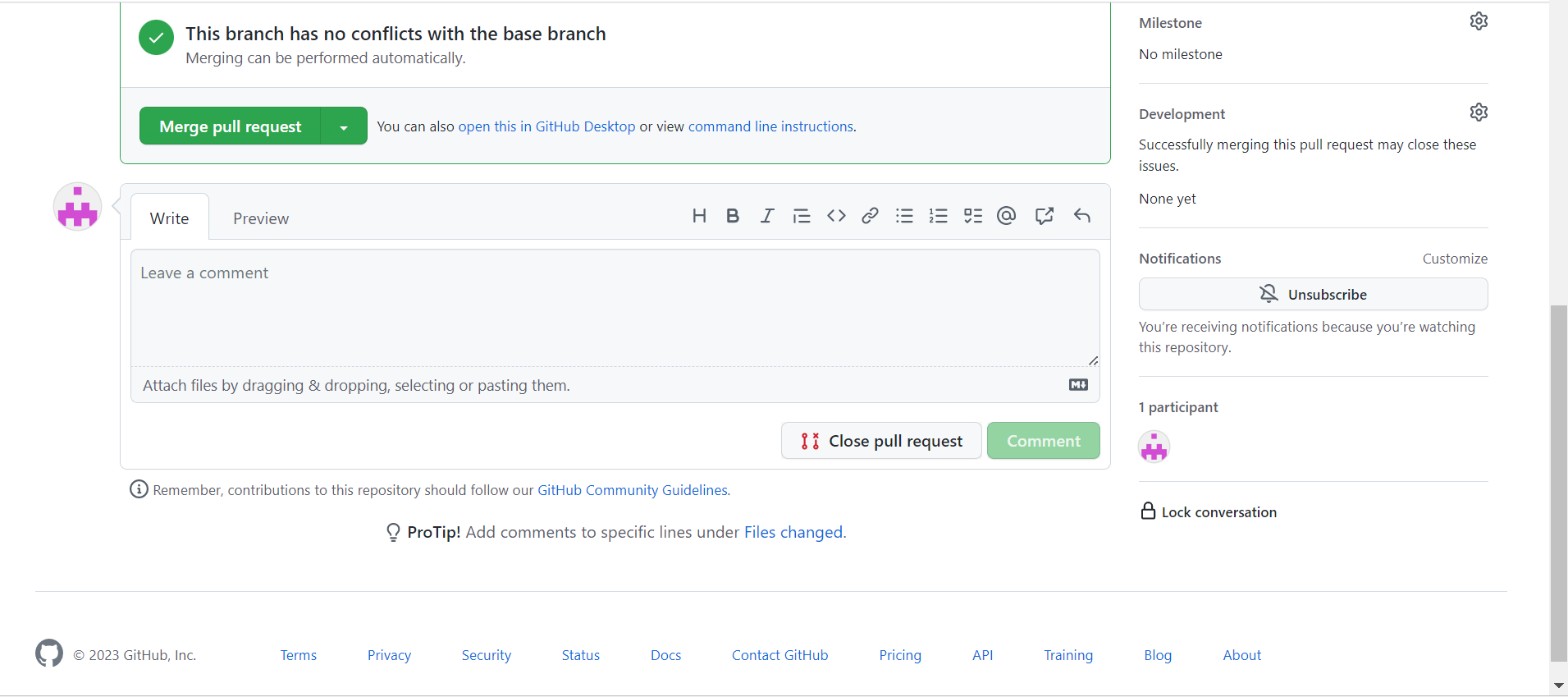
1. Perform merging of both branches

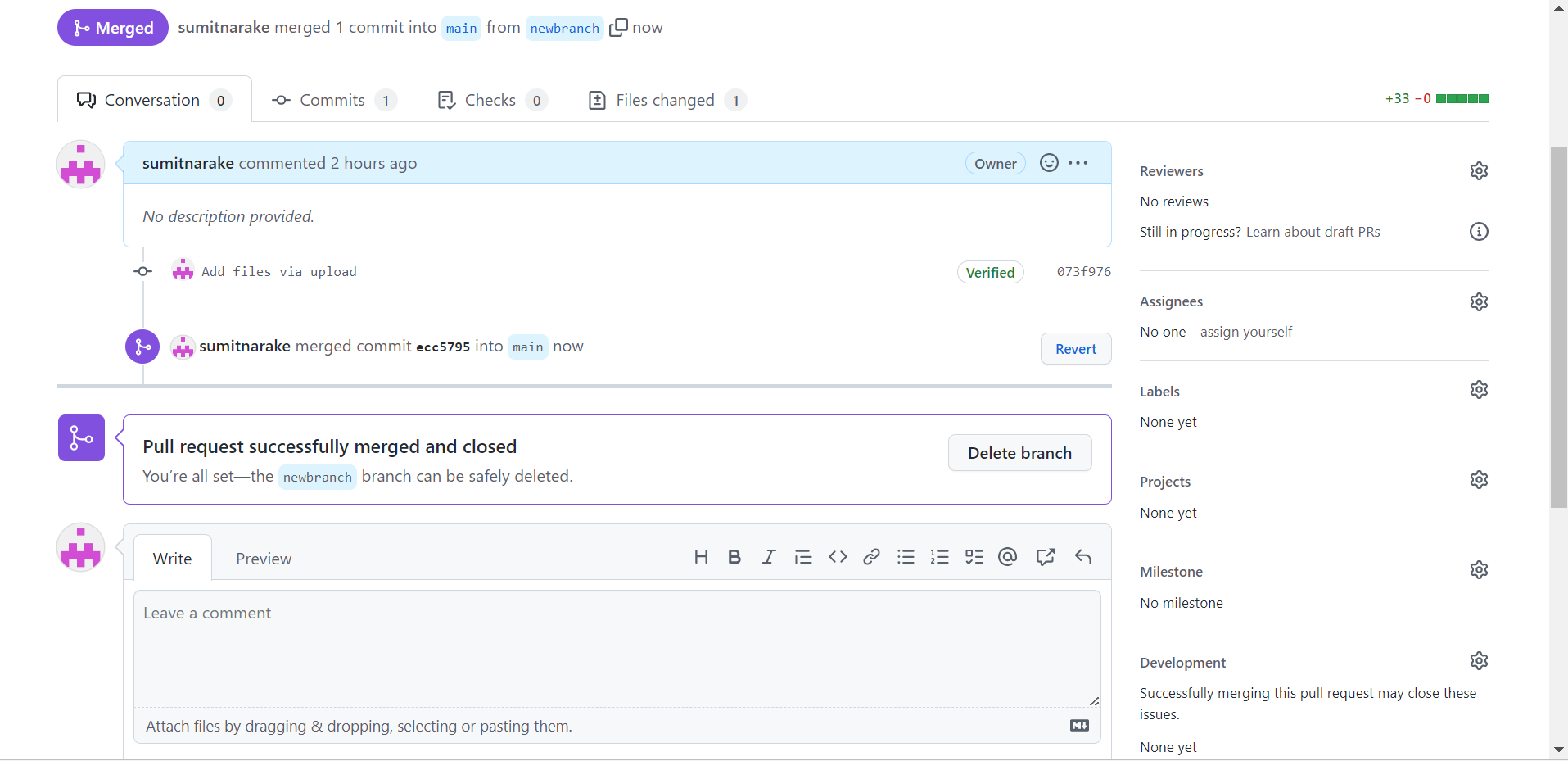
To perform the merging of both branches, follow these steps:

● Go to the pull request you created in the previous step.

● Click on the "Merge pull request" button.

● Click on the "Confirm merge" button.





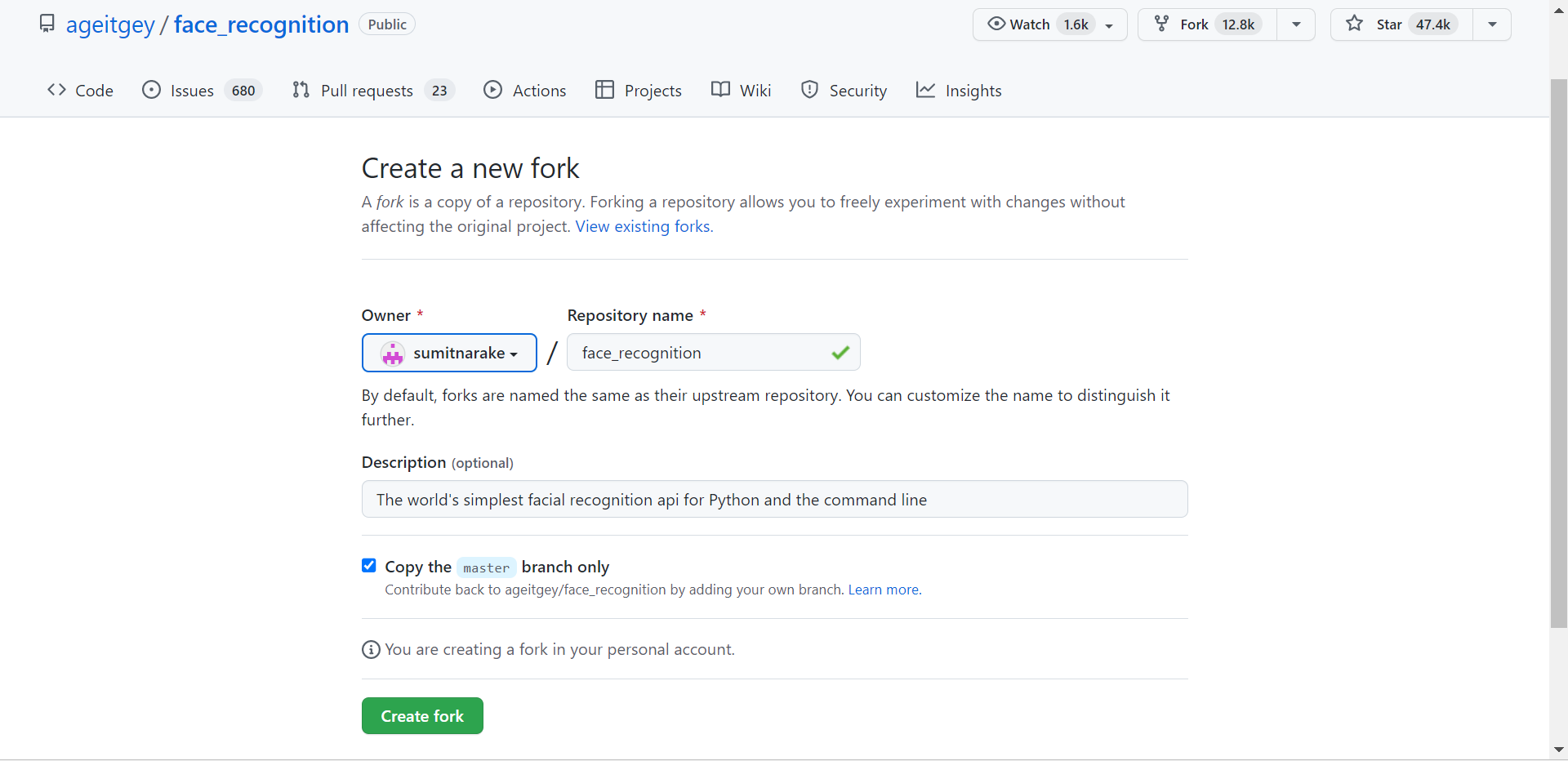
1. Perform Fork operation

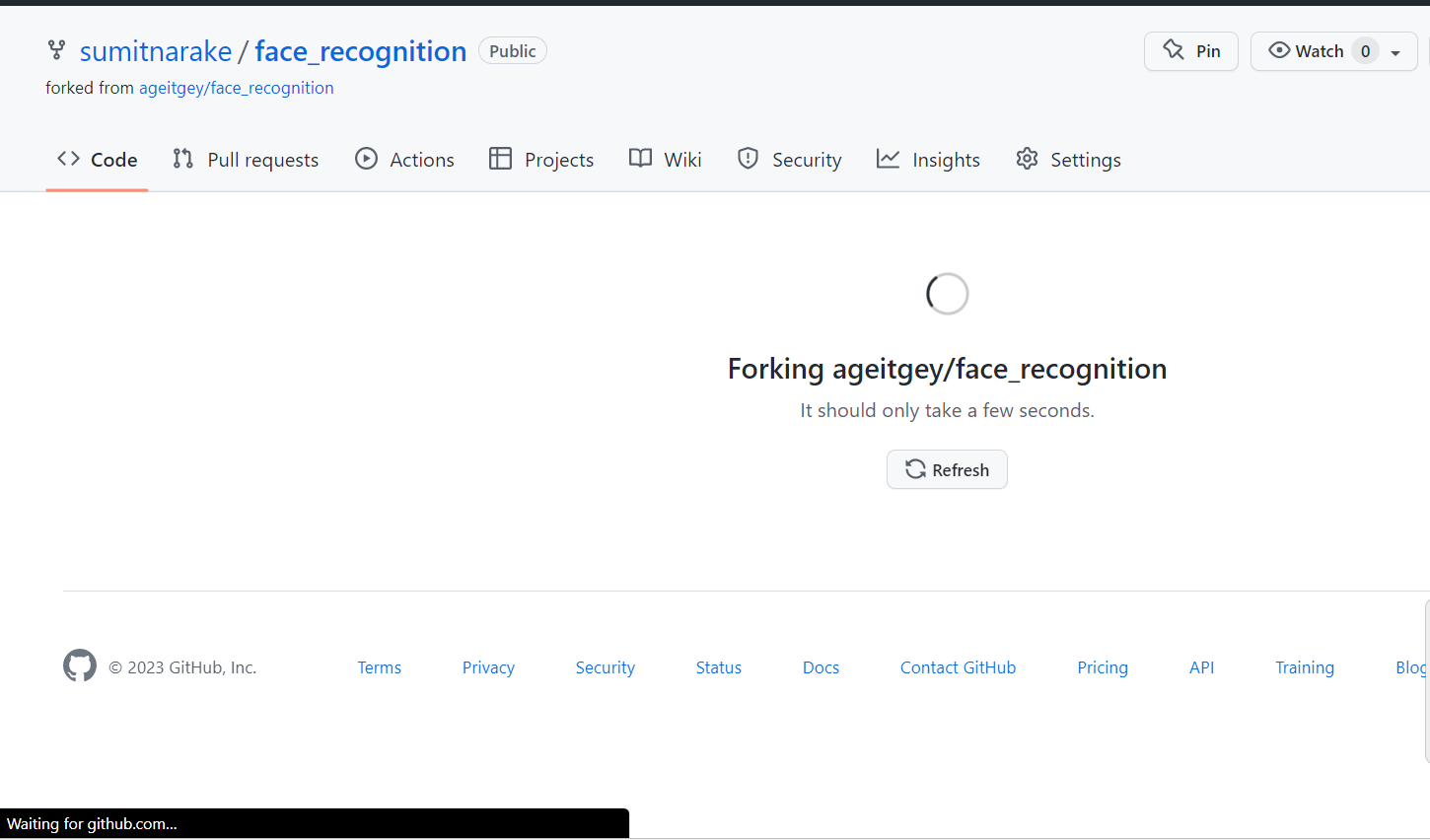
To perform a fork operation, follow these steps:

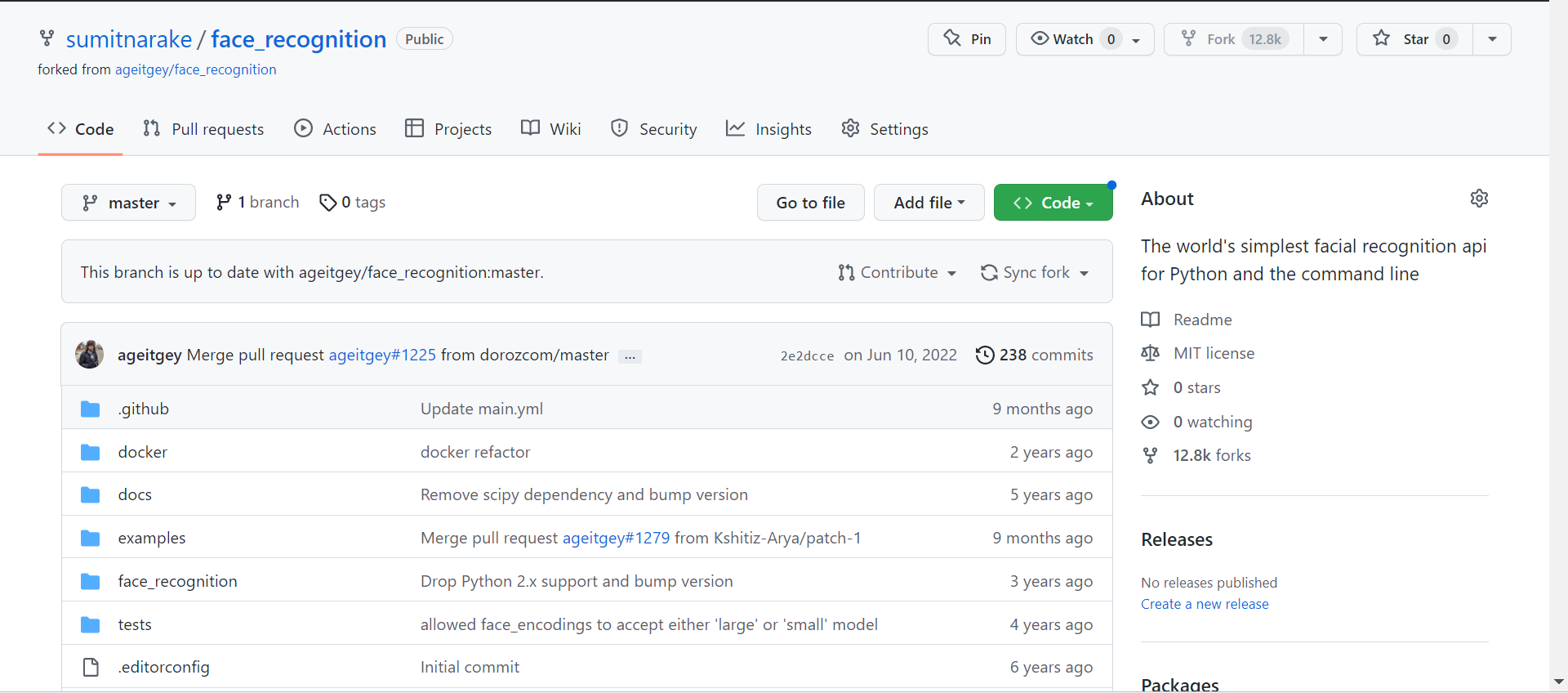
● Go to the main page of the repository you want to fork on GitHub.

● Click on the "Fork" button in the top right corner.

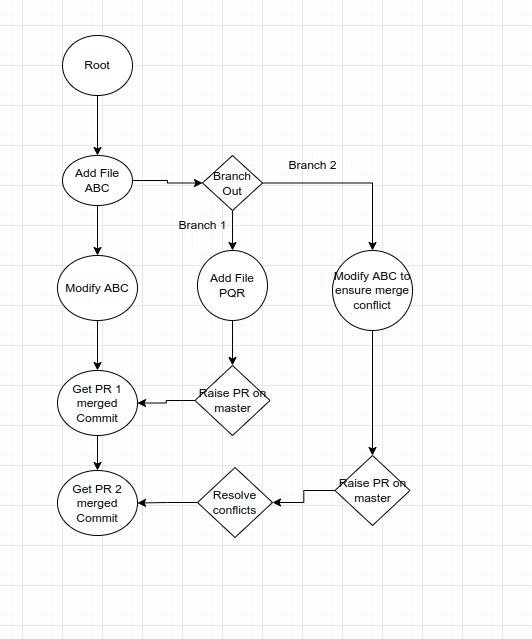
● Select the account you want to fork the repository to



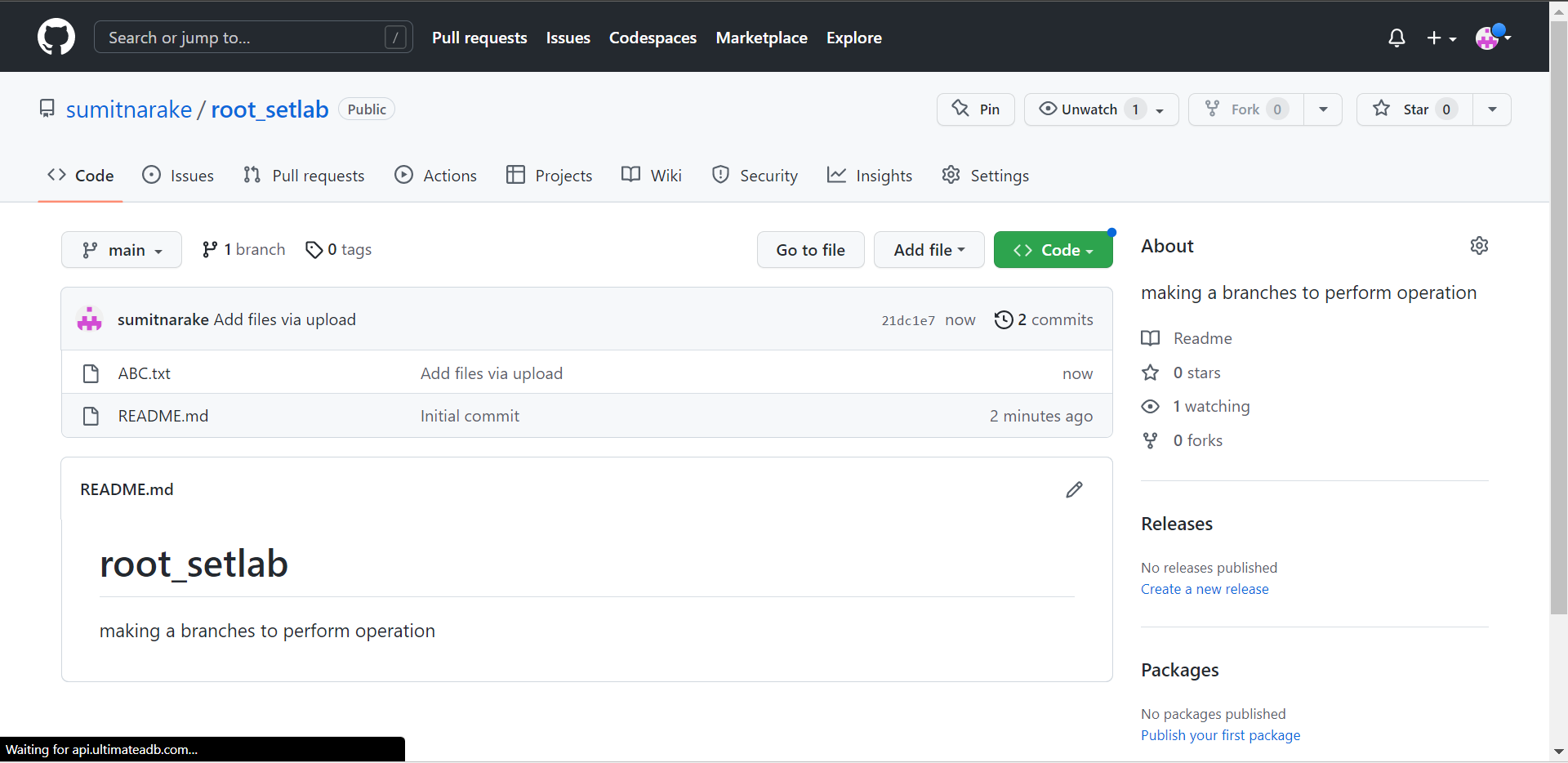




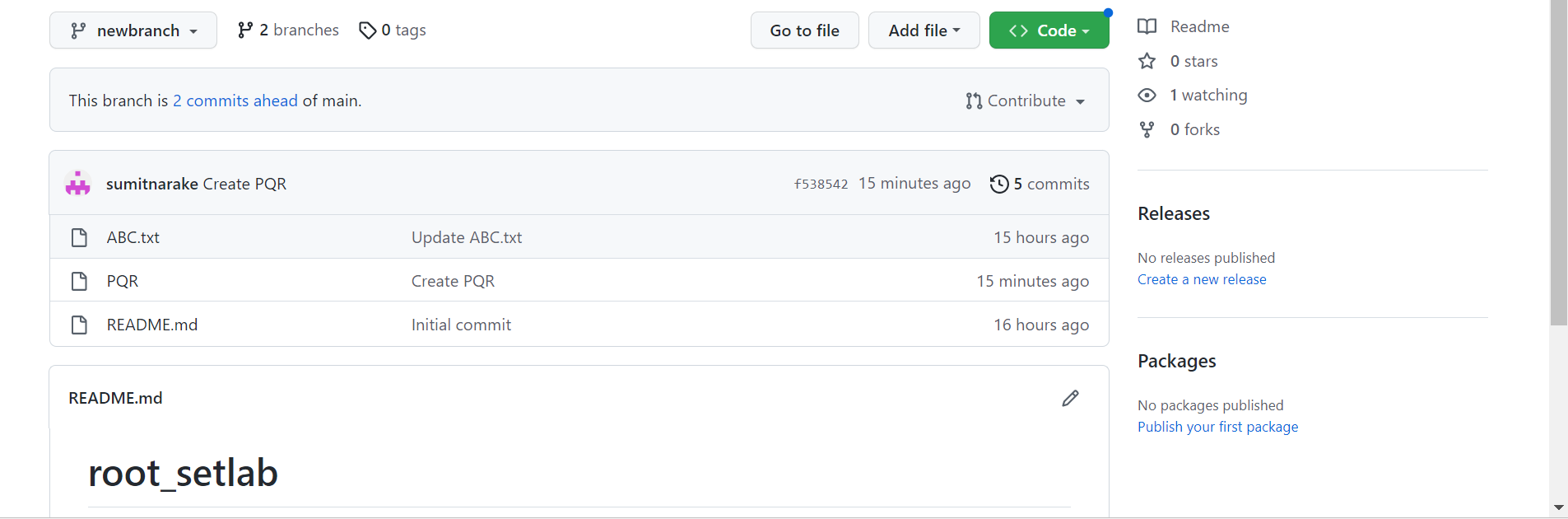
Q 2. For the diagram given below create a GitHub repository and perform operations given in the diagram. (Perform commit operations as given)(Add screenshots as an answer to this question)



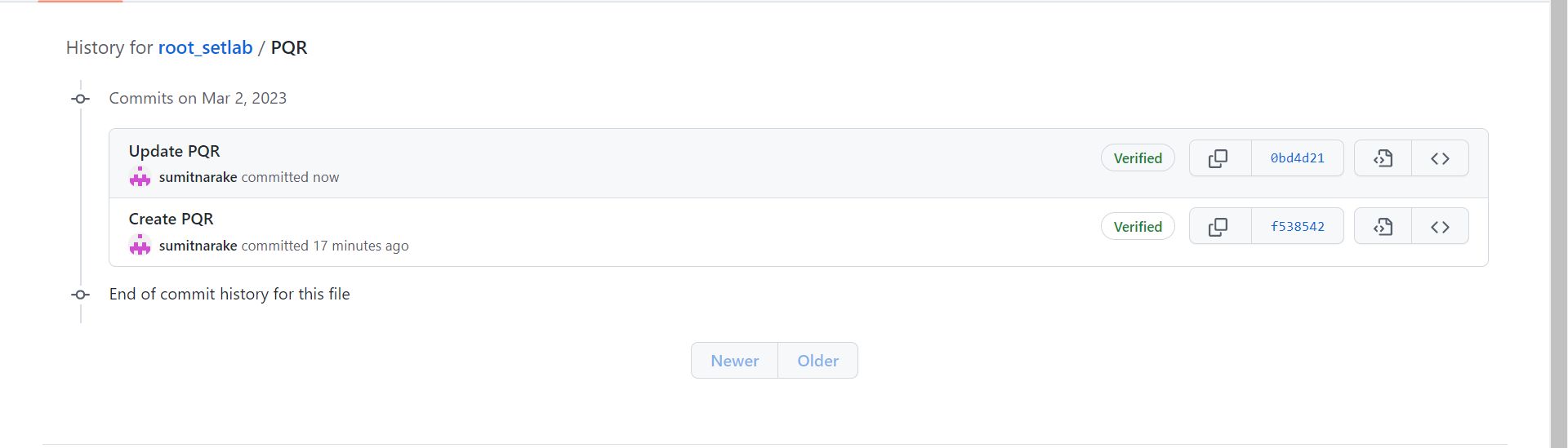
Step 1: Created New Repo and added a file named ABC.txt



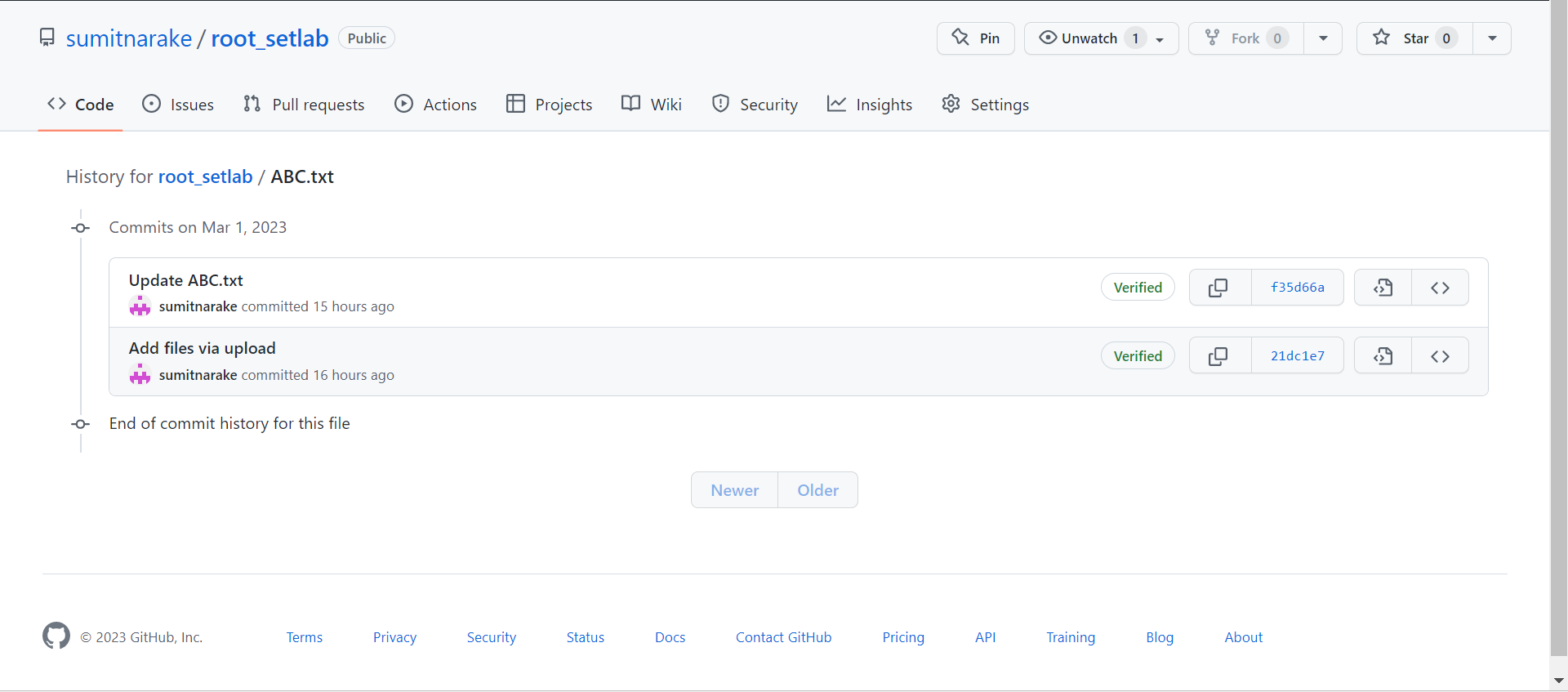
Step 2: Created New Branch.



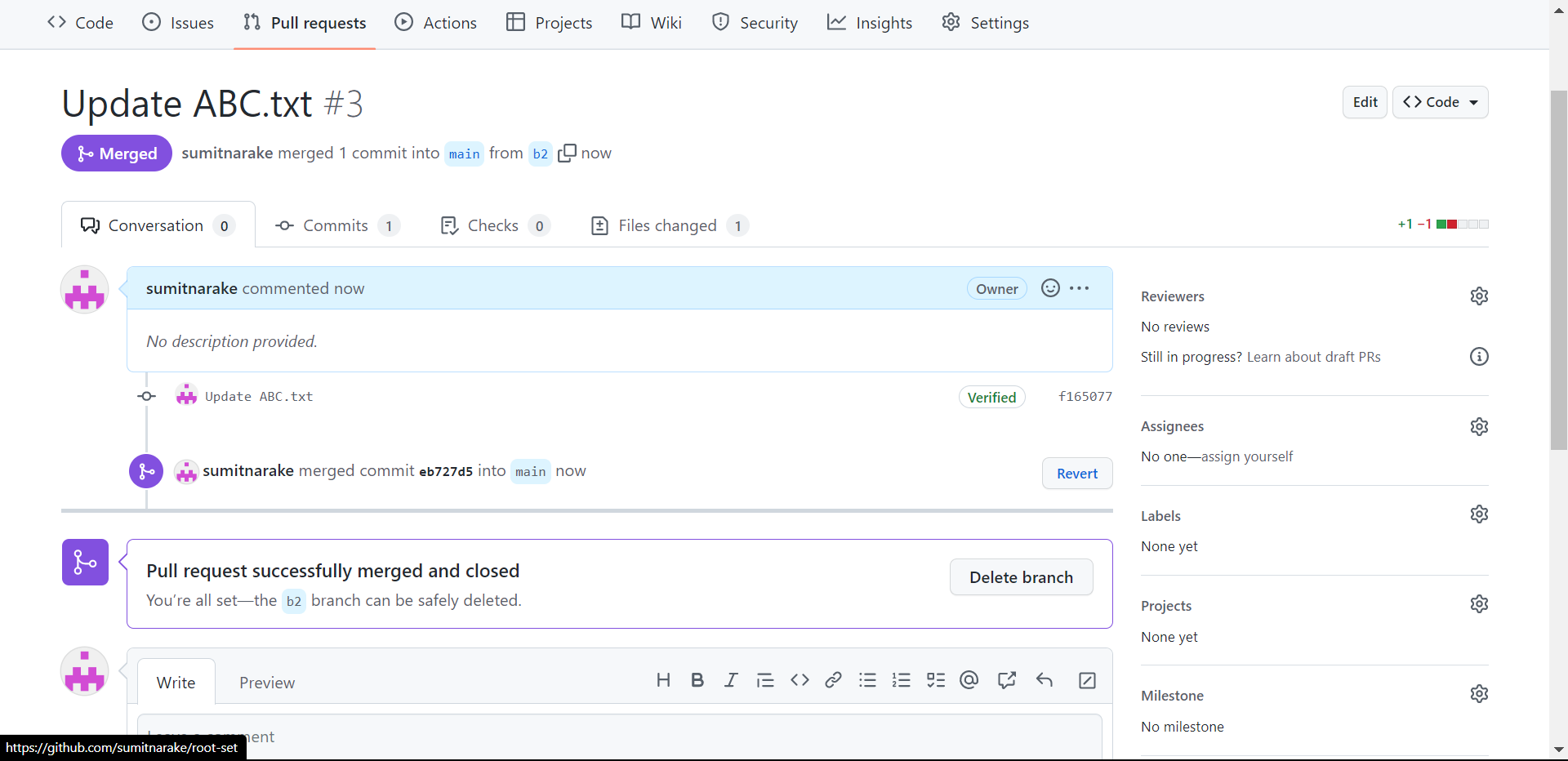
Step3 : Updated file in new branch.



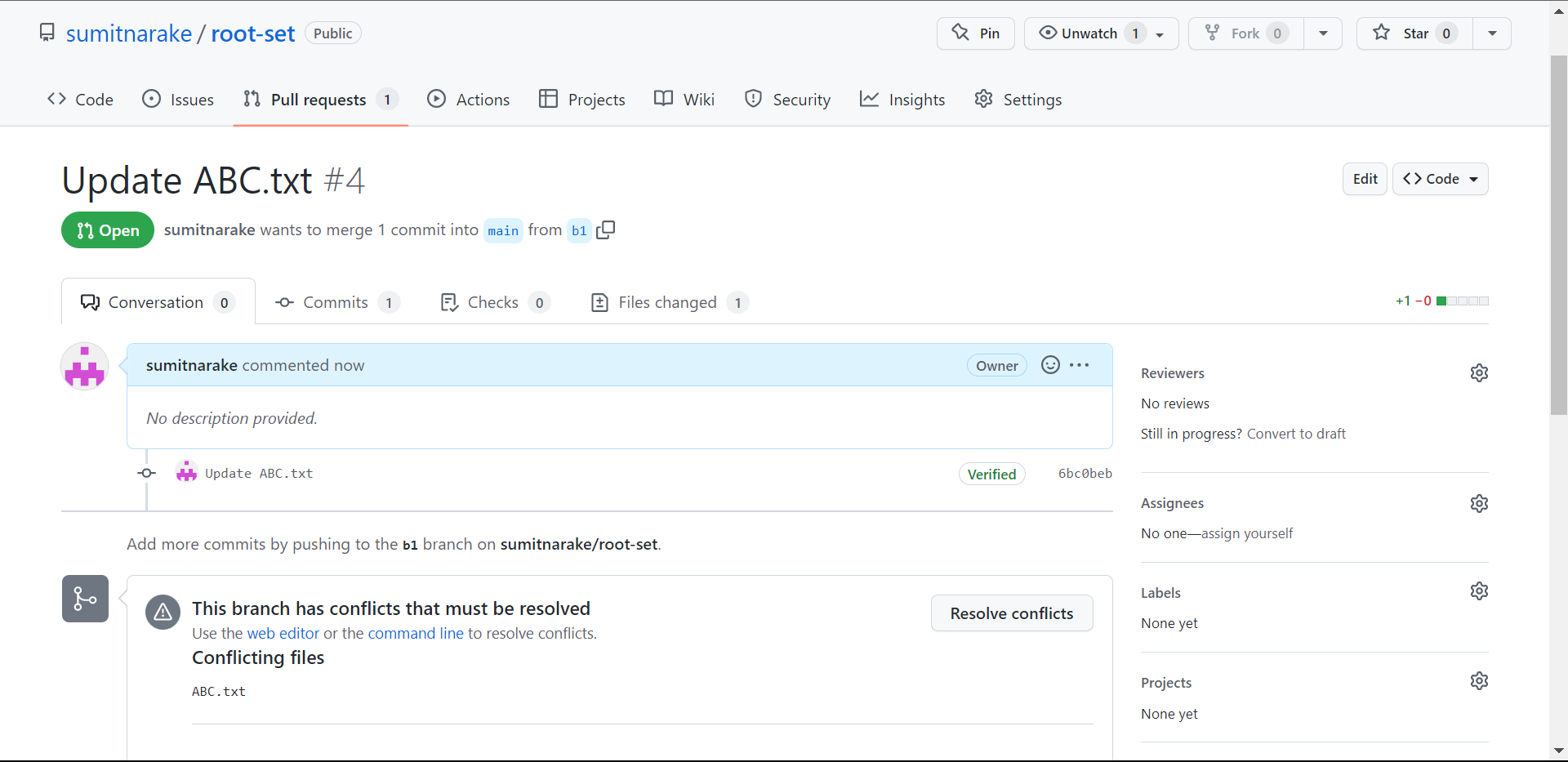
Step 4: Updated main branch file ABC.

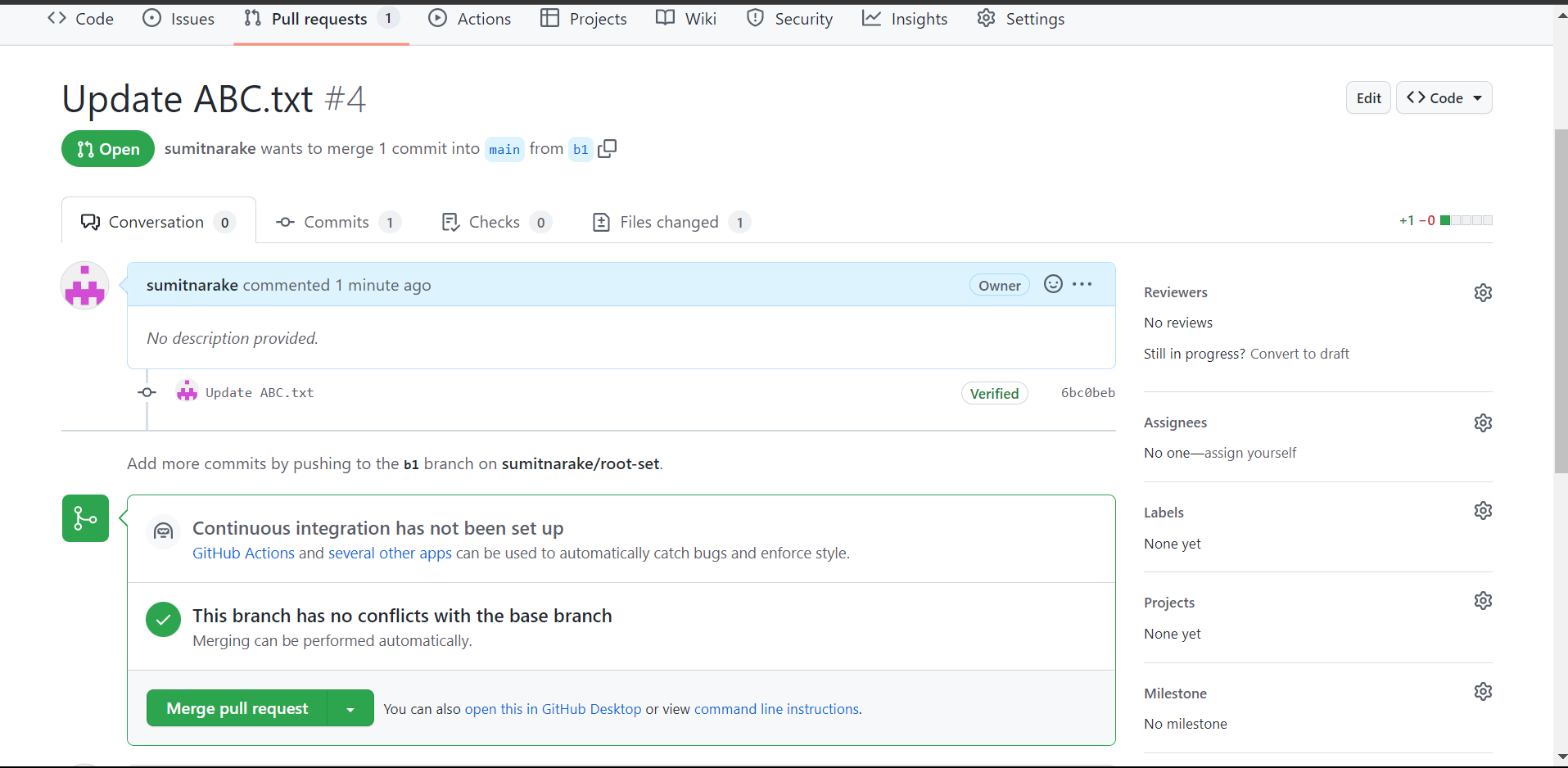


tep 5: Merged new branch to main branch



Resolved Conflicts:





Q 3. What is GitHub desktop? How to install GitHub on local machine? Install GitHub on your local machine and access repository created in question no 1 (add screenshots).

GitHub desktop-

GitHub Desktop is an application that enables you to interact with GitHub using a GUI instead of the command line or a web browser. GitHub Desktop encourages you and your team to collaborate using best practices with Git and GitHub.

How to install GitHub on local machine-

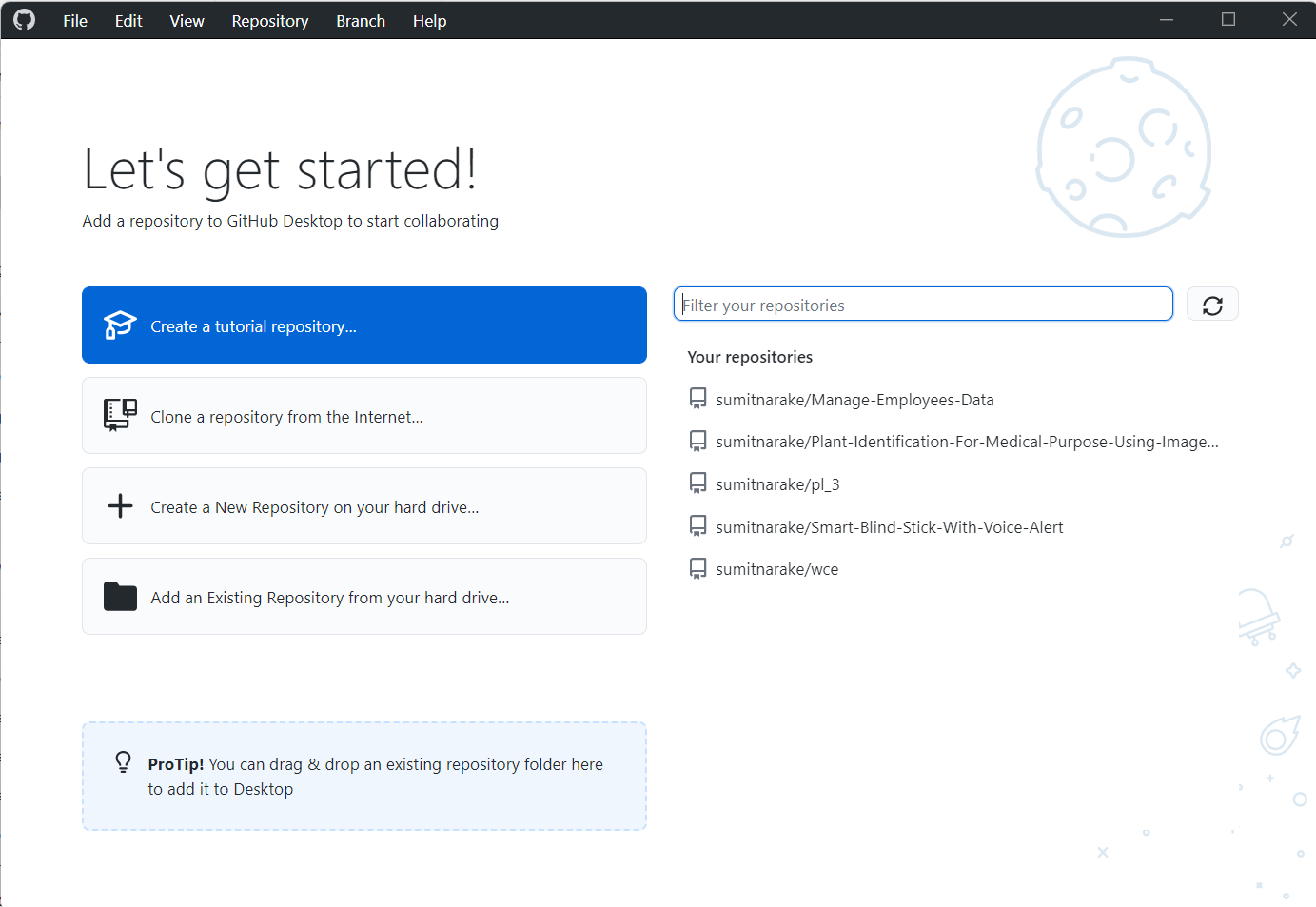
1. Visit the [download page for GitHub Desktop](https://desktop.github.com/).



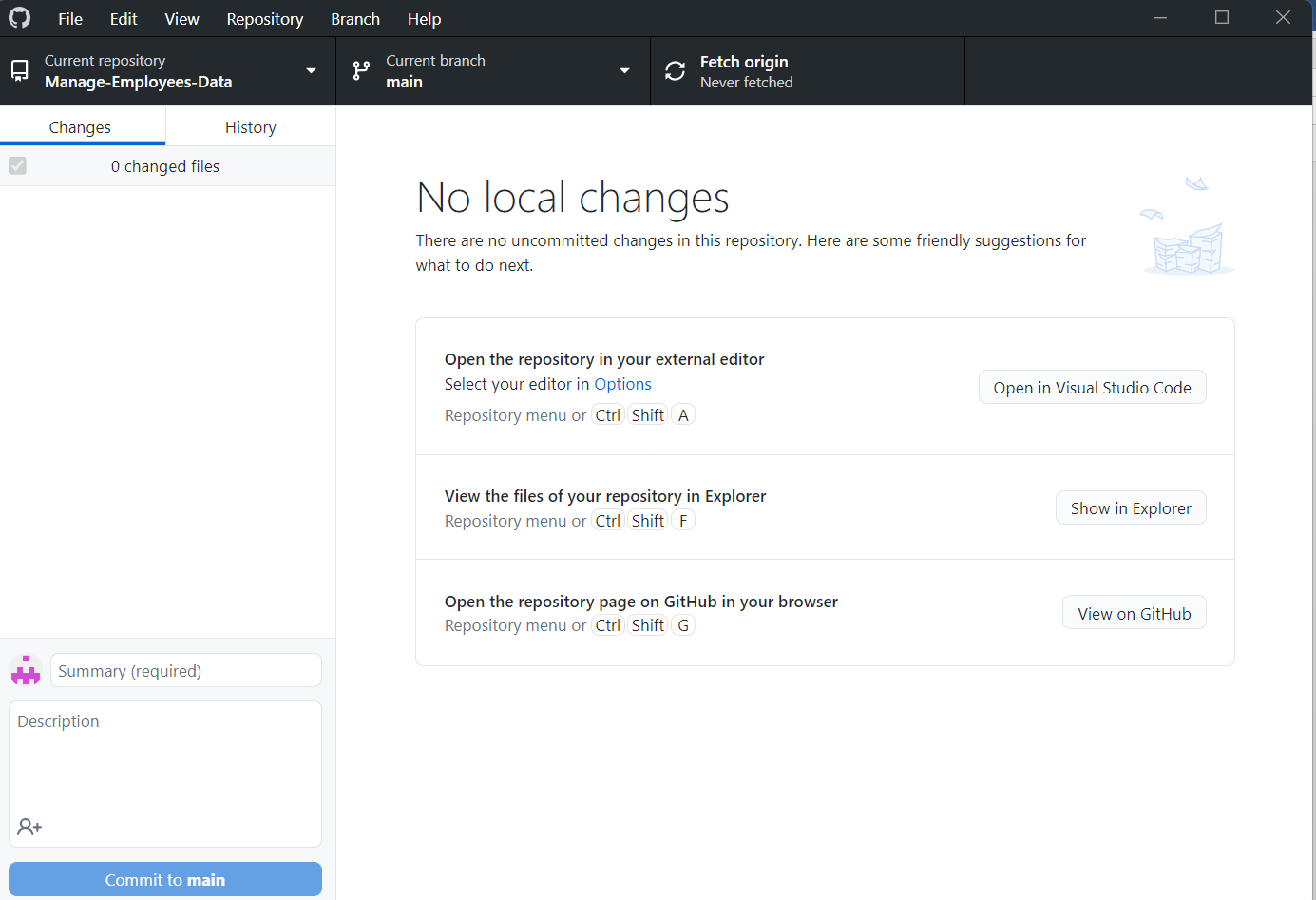
1. Click **Download for Windows**.
2. In your computer's Downloads folder, double-click the **GitHub Desktop** setup file.



1. GitHub Desktop will launch after installation is complete.



GitHub on your local machine and access repository created-



Q 4. Differentiate in between GitHub, Git and GitLab.

Git, GitHub, and GitLab are related but distinct tools in the world of version control and collaboration. Here's a brief overview of each:

1. Git: Git is a distributed version control system that was created by Linus Torvalds in 2005. It allows developers to track changes in their code and collaborate with other developers on a project. Git is the underlying technology that powers most version control systems, including GitHub and GitLab.

2. GitHub: GitHub is a web-based platform that provides hosting for Git repositories, as well as a range of tools for issue tracking, project management, and collaboration. GitHub is one of the largest hosting platforms for open-source projects, and it provides a wide range of tools and integrations to help developers work together.

3. GitLab: GitLab is an open-source Git repository manager that provides a webbased Git repository manager, issue tracking, and project management tools. GitLab also provides a range of additional tools and features, including continuous integration and deployment, security scanning, and more.

Q 5. What is version control? Explain with example.

Version control systems are a category of software tools that helps in recording changes made to files by keeping a track of modifications done in the code.

* Enhances the project development speed by providing efficient collaboration,
* Leverages the productivity, expedites product delivery, and skills of the employees through better communication and assistance,
* Reduce possibilities of errors and conflicts meanwhile project development through traceability to every small change,
* Employees or contributors of the project can contribute from anywhere irrespective of the different geographical locations through this **VCS,**
* For each different contributor to the project, a different working copy is maintained and not merged to the main file unless the working copy is validated. The most popular example is **Git, Helix core, Microsoft TFS,**
* Helps in recovery in case of any disaster or contingent situation,
* Informs us about Who, What, When, Why changes have been made.

# Git Version Control System

A version control system is a software that tracks changes to a file or set of files over time so that you can recall specific versions later. It also allows you to work together with other programmers.

The version control system is a collection of software tools that help a team to manage changes in a source code. It uses a special kind of database to keep track of every modification to the code.

Developers can compare earlier versions of the code with an older version to fix the mistakes.

Git is a distributed version control system that enables software development teams to have multiple local copies of the project’s codebase independent of each other. These copies, or branches, can be created, merged, and deleted quickly, empowering teams to experiment, with little compute cost, before merging into the main branch (sometimes referred to as the [master branch](https://about.gitlab.com/blog/2021/03/10/new-git-default-branch-name/)). Git is known for its speed, [workflow compatibility](https://about.gitlab.com/blog/2020/04/07/15-git-tips-improve-workflow/), and open source foundation.

Most Git actions only add data to the database, and Git makes it easy to undo changes during the three main states.

Git has three file states: modified, staged, and committed.

1. A modified file has been changed but isn’t committed to the database yet.
2. A staged file is set to go into the next commit.
3. When a file is committed, the data has been stored in the database.

With Git, software teams can experiment without fearing that they’ll create lasting damage to the source code, because teams can always revert to a previous version if there are any problems.