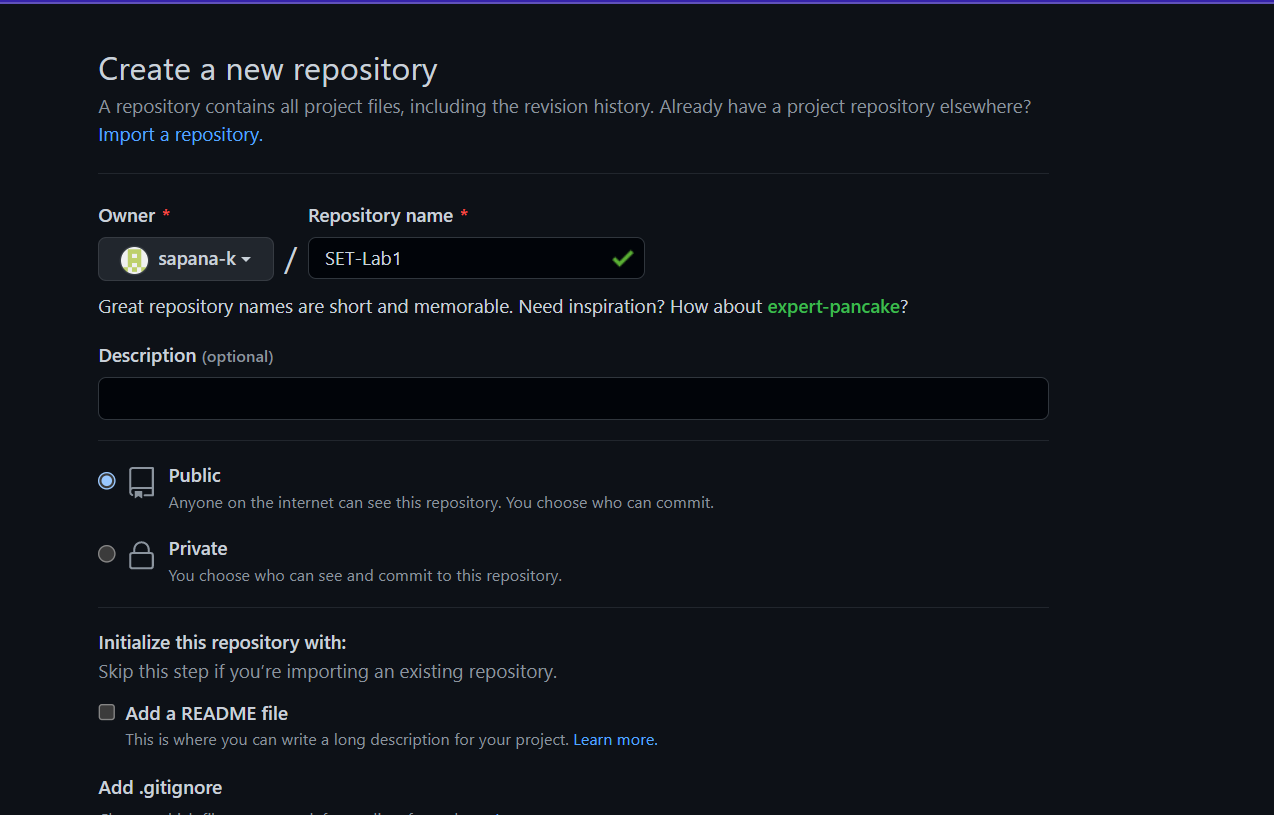
**SET ASSIGNMENT – 4**

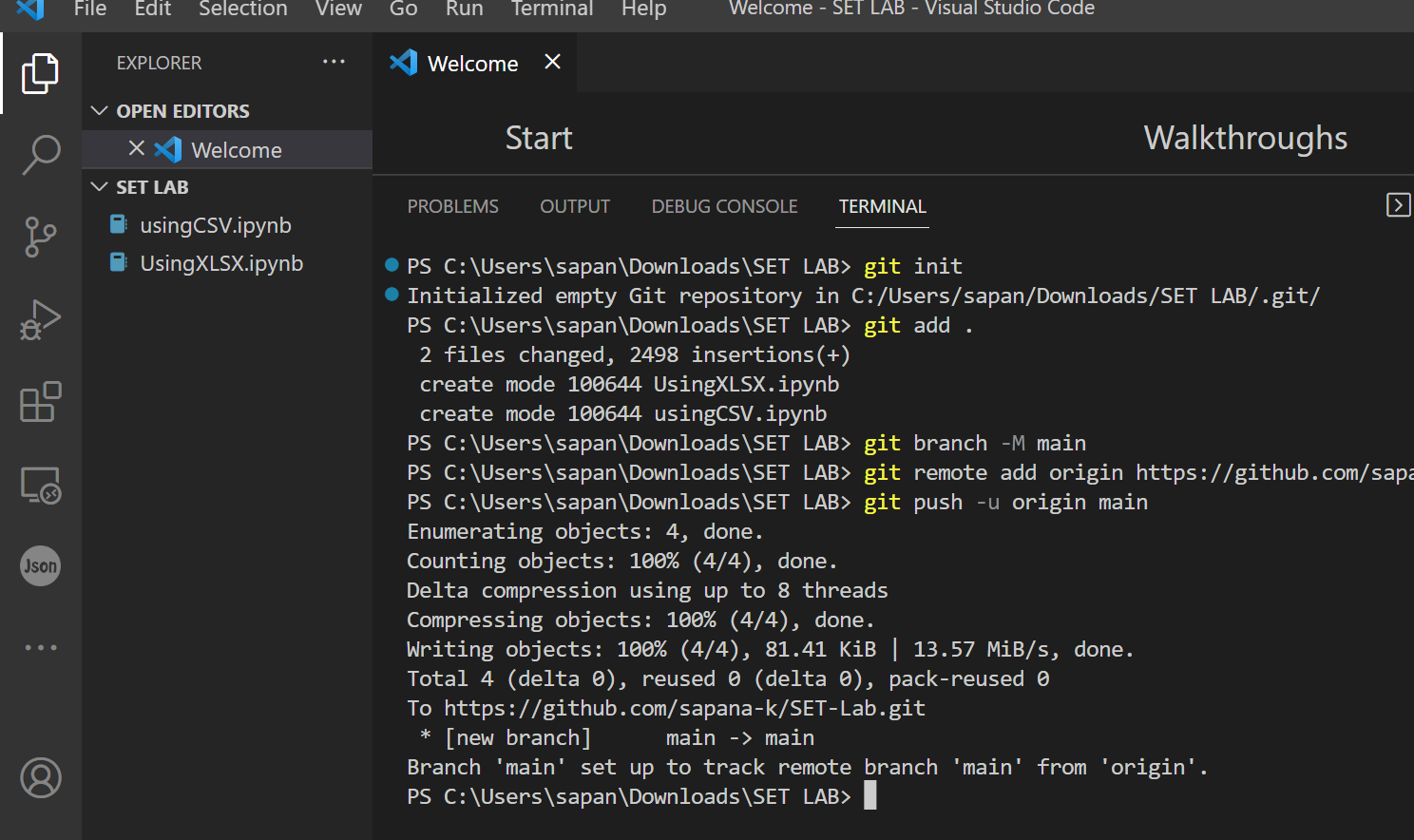
**Q 1. Create a repository on GitHub named SET Lab and add files into it (you can**

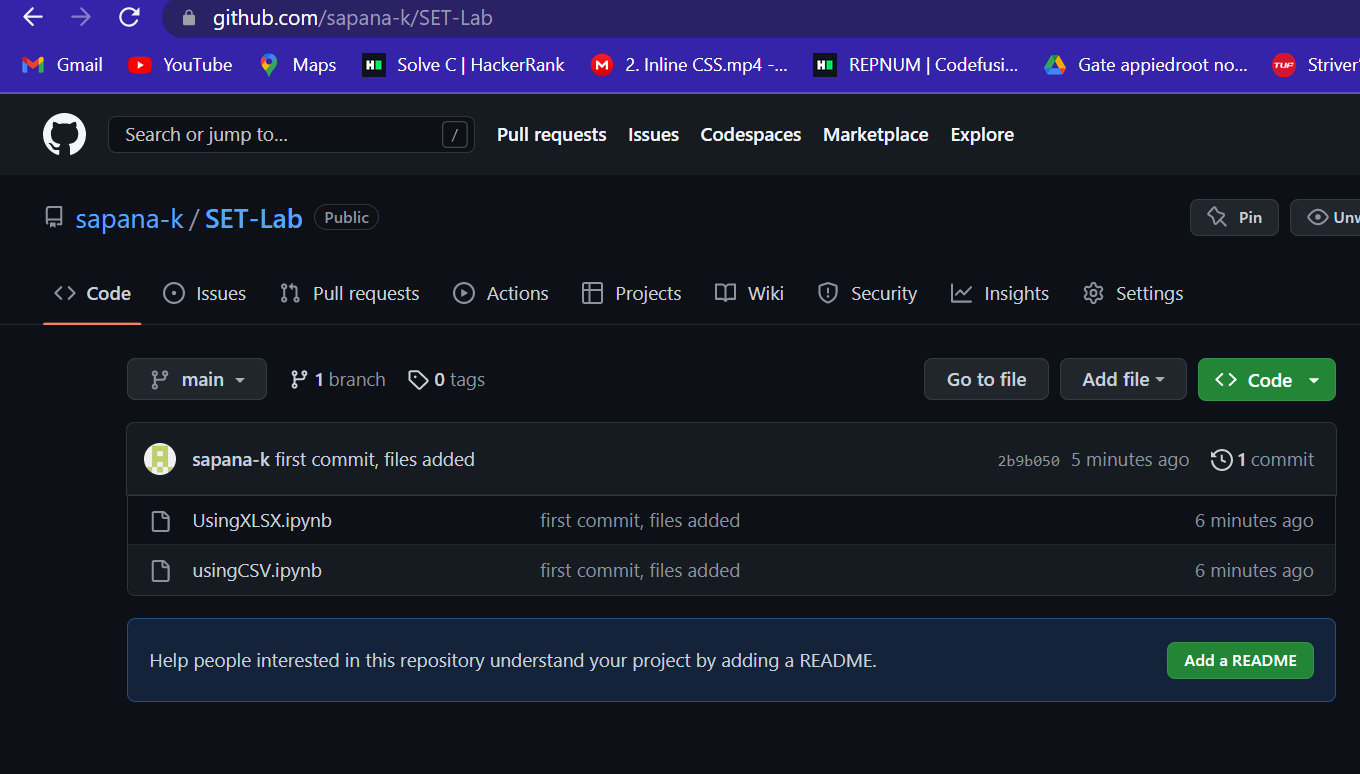
**add implementation files of previous assignment) perform below operations on it.**

****

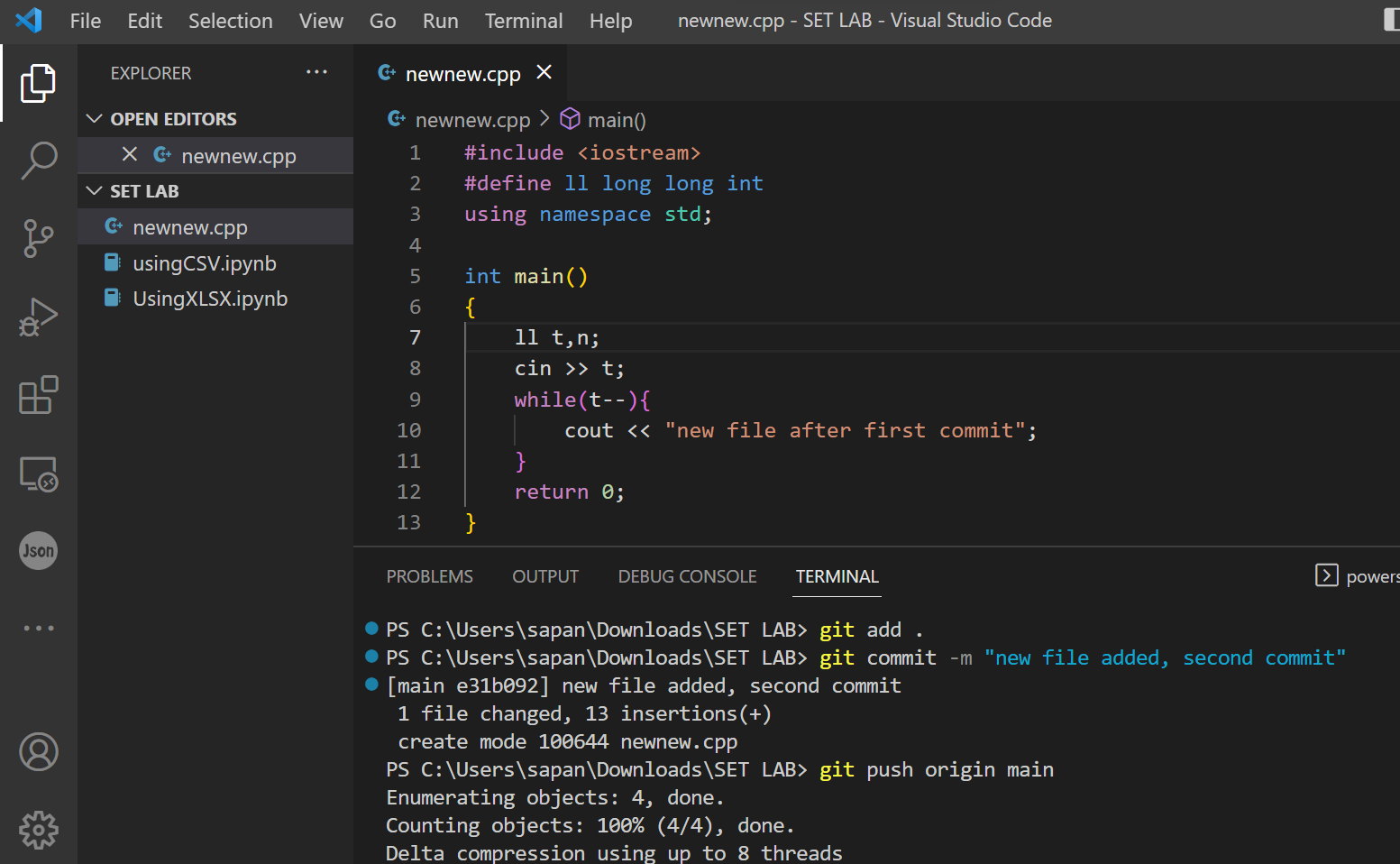
**(Add screenshot as an answer to every question)**

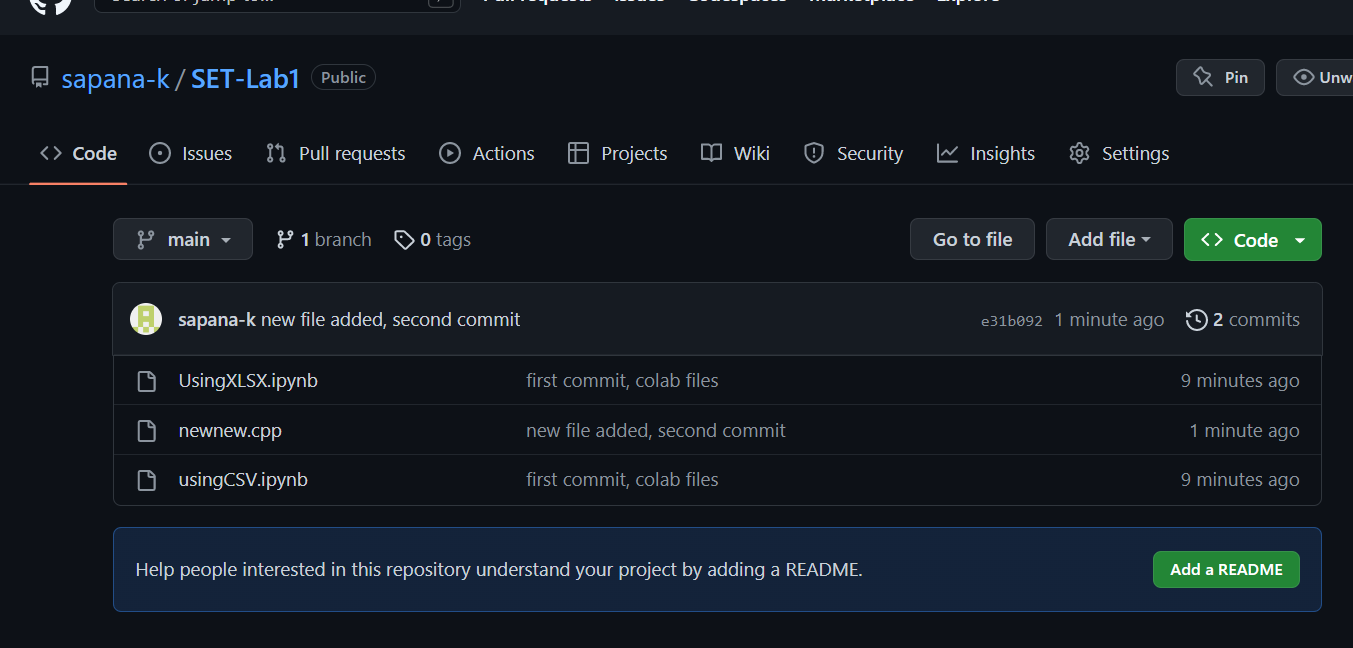
1. **Perform commit on added files**

****

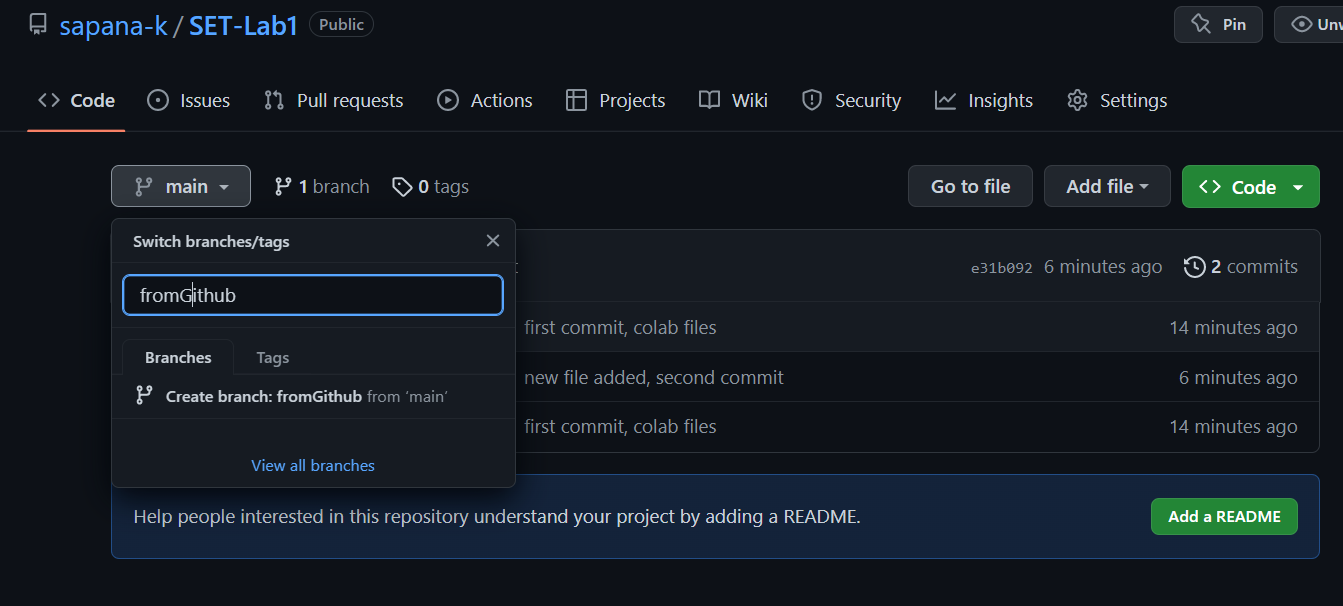
****

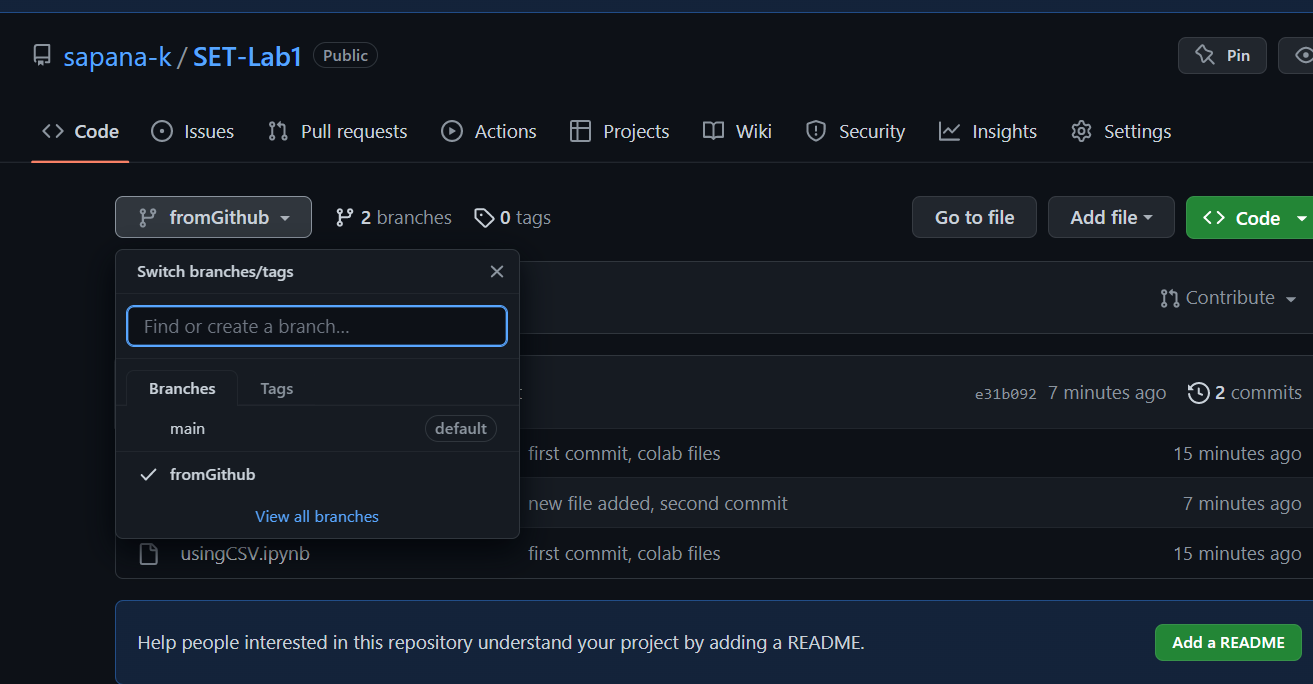
1. **Perform update to the existing files (show history)**

****

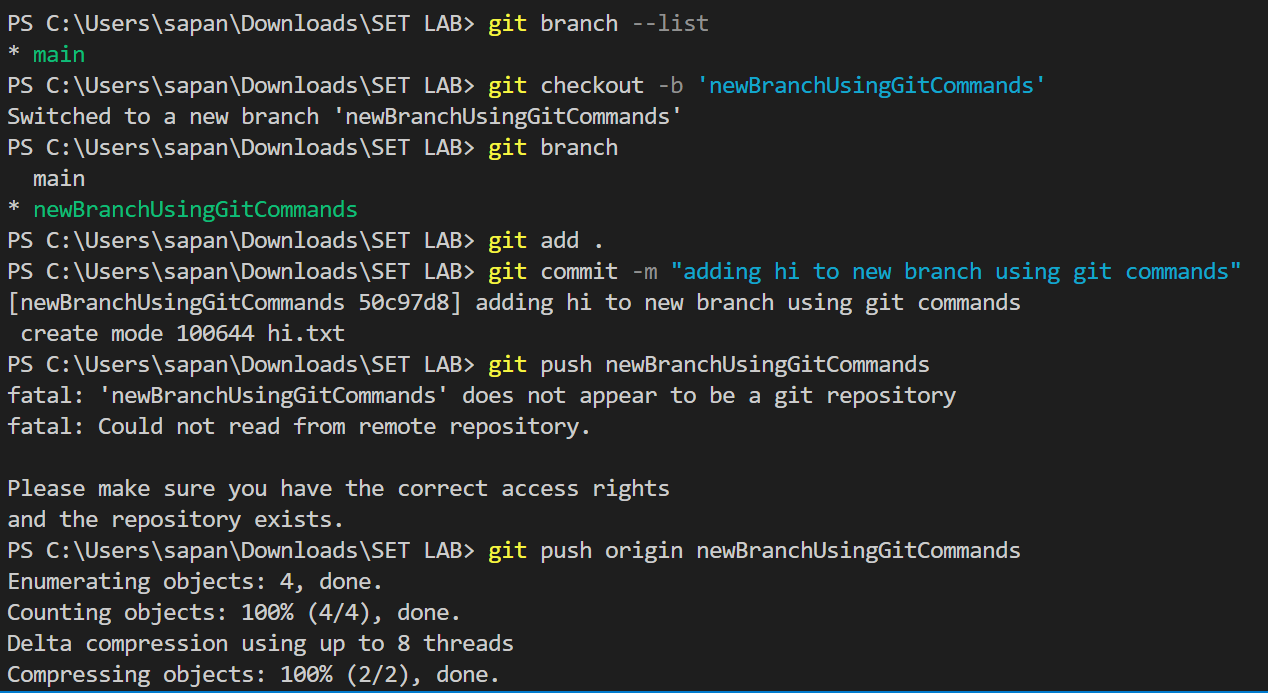
****

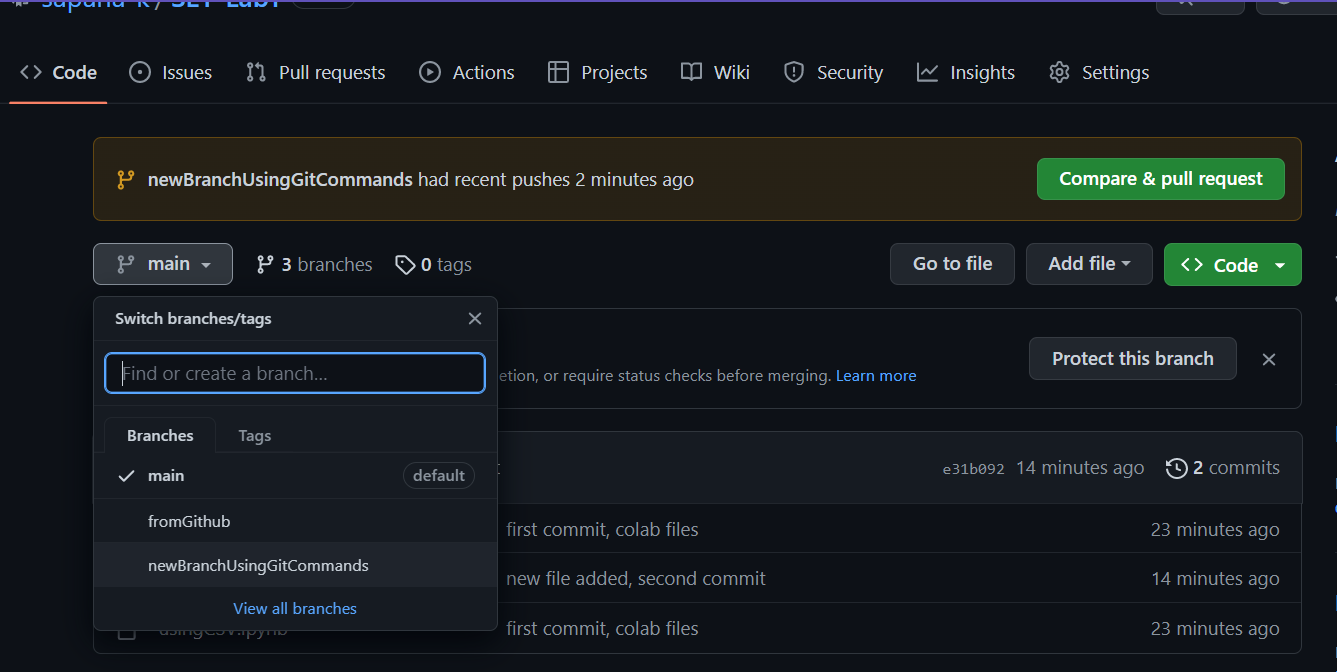
1. **Create another branch**

****

****

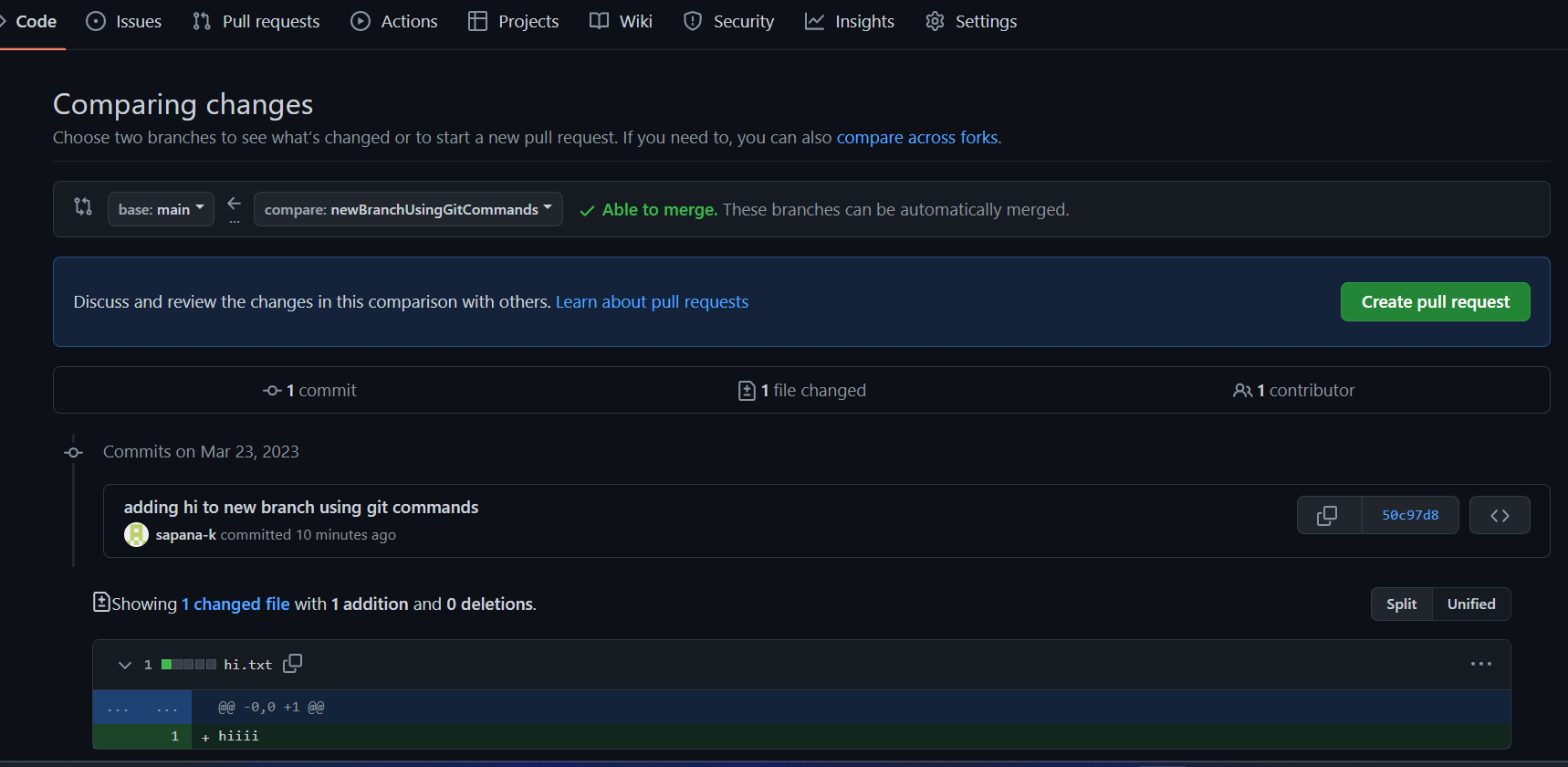
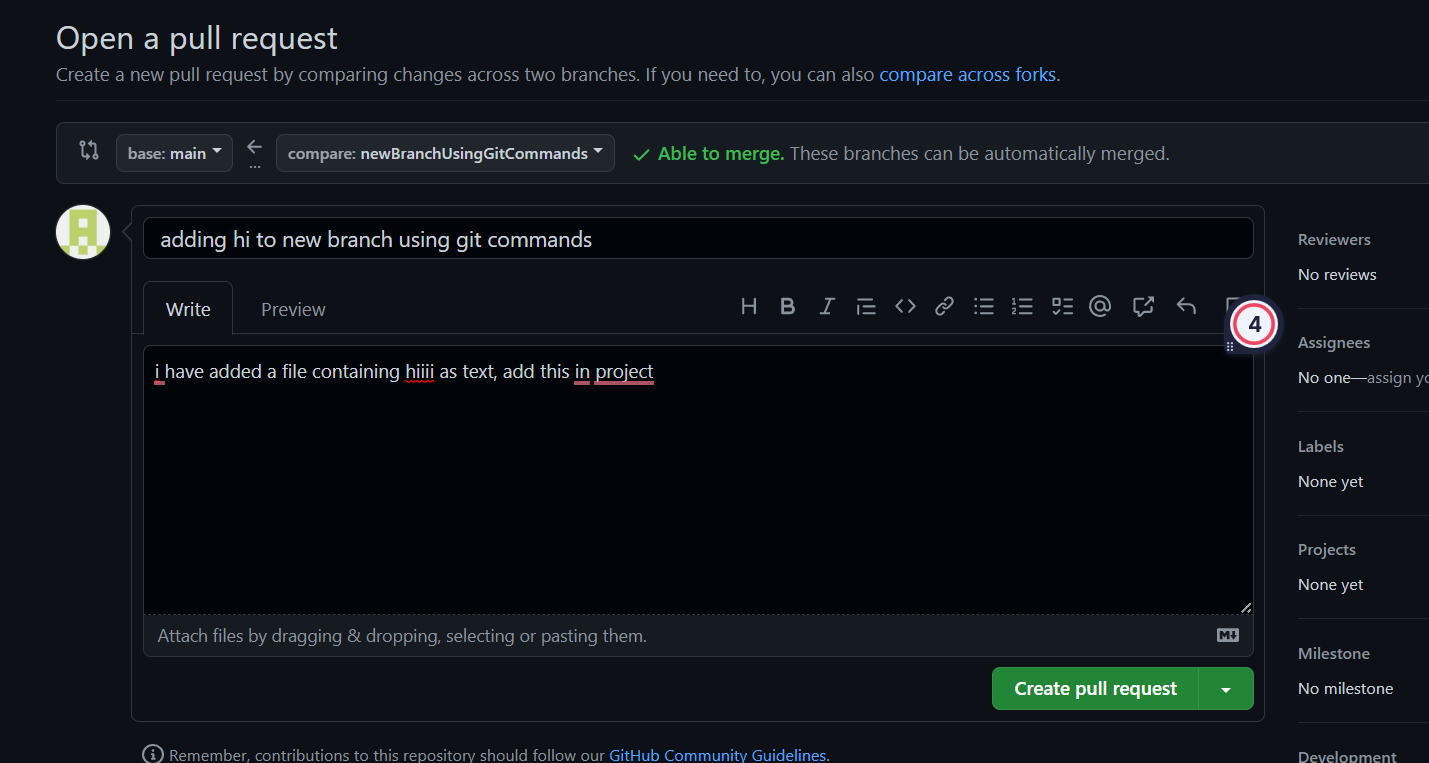
**Using git commands**

****

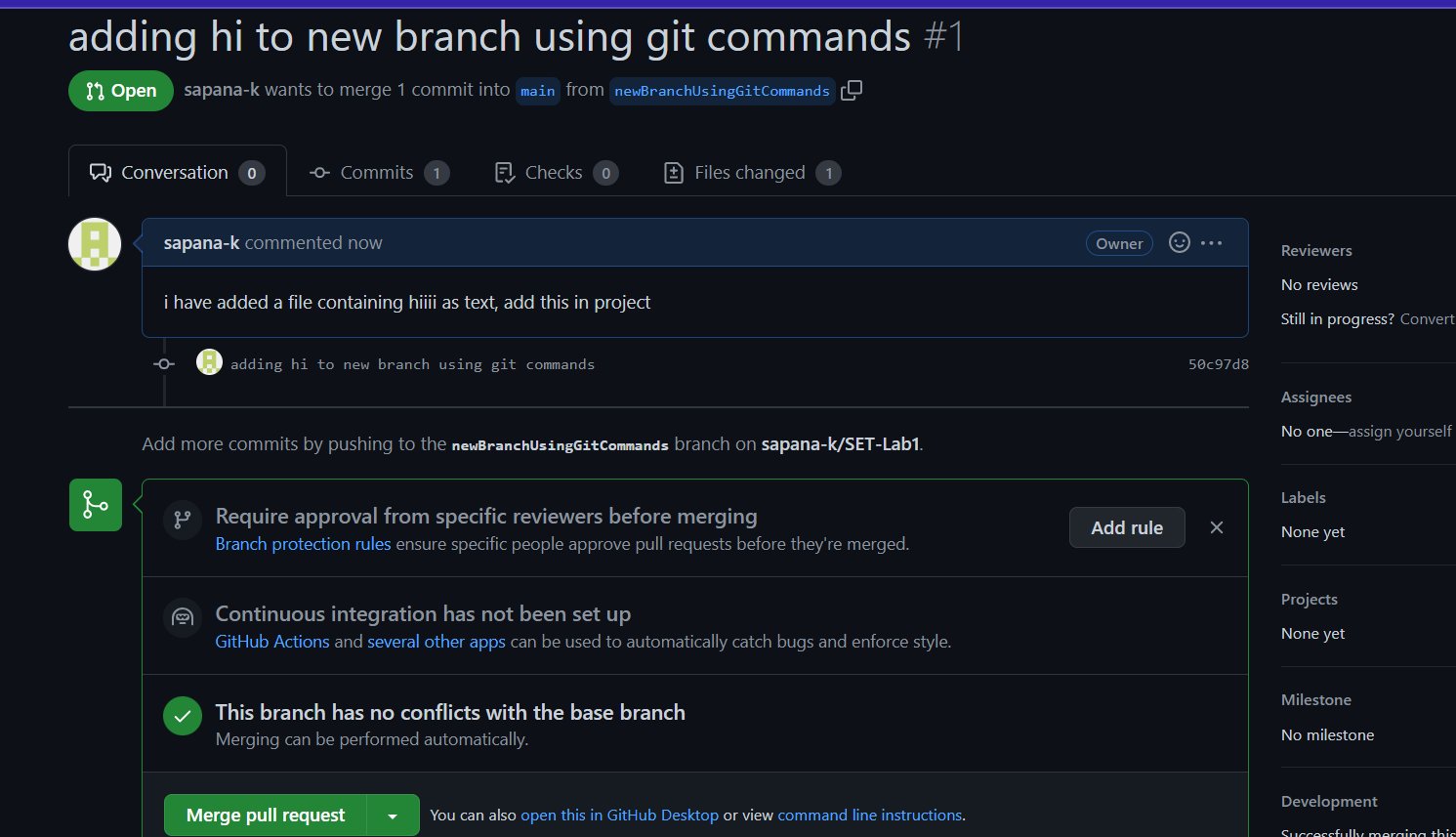
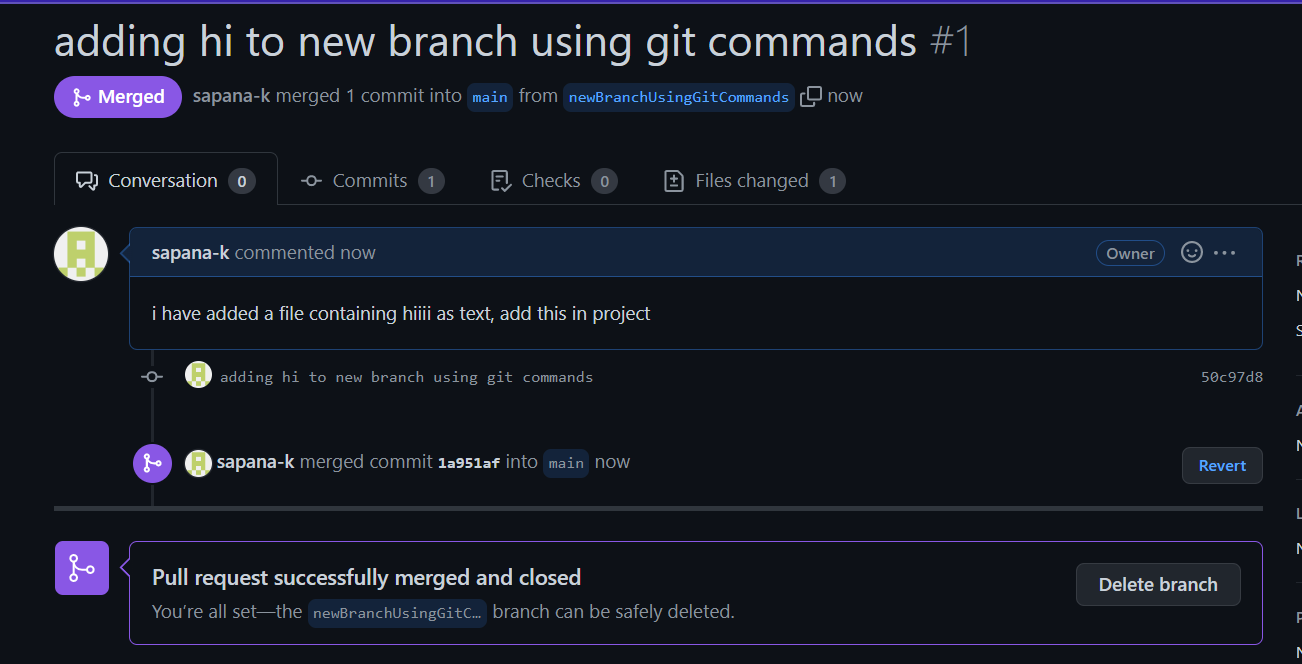
****

1. **Create pull request**

A pull request in GitHub is a mechanism used for suggesting and discussing changes to a project's codebase with other collaborators. It is essentially a request to pull in changes from a branch of a repository to another branch of the same repository, usually the mainbranch.

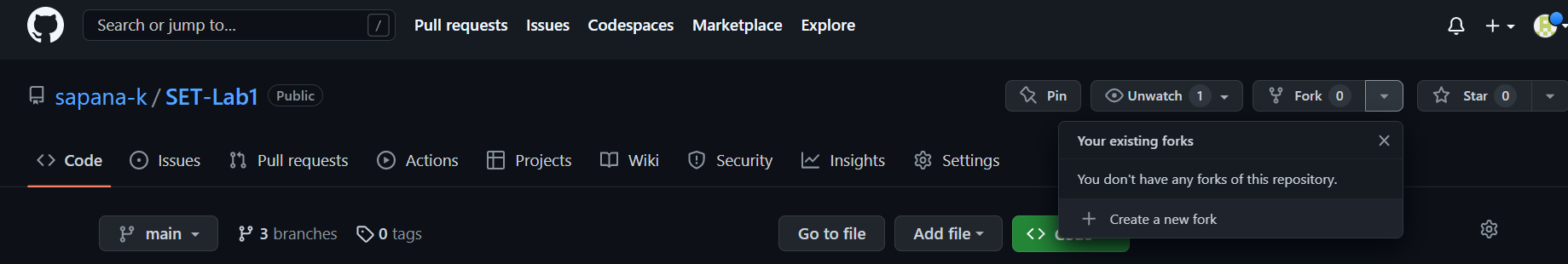
**** ****

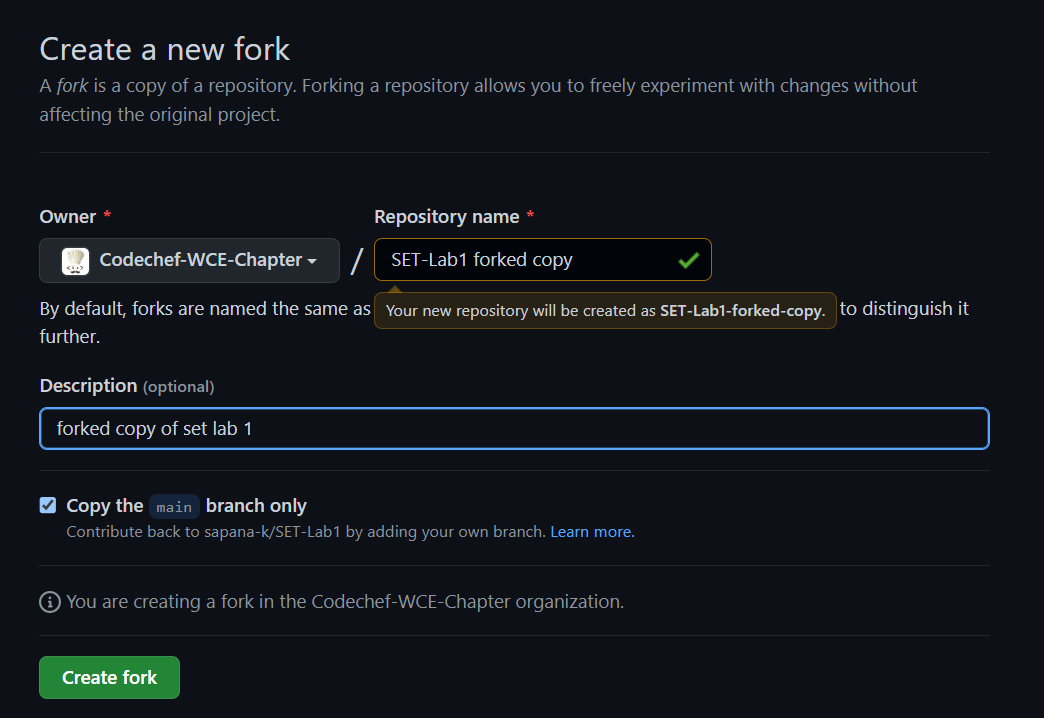
1. **Perform merging of both branches**

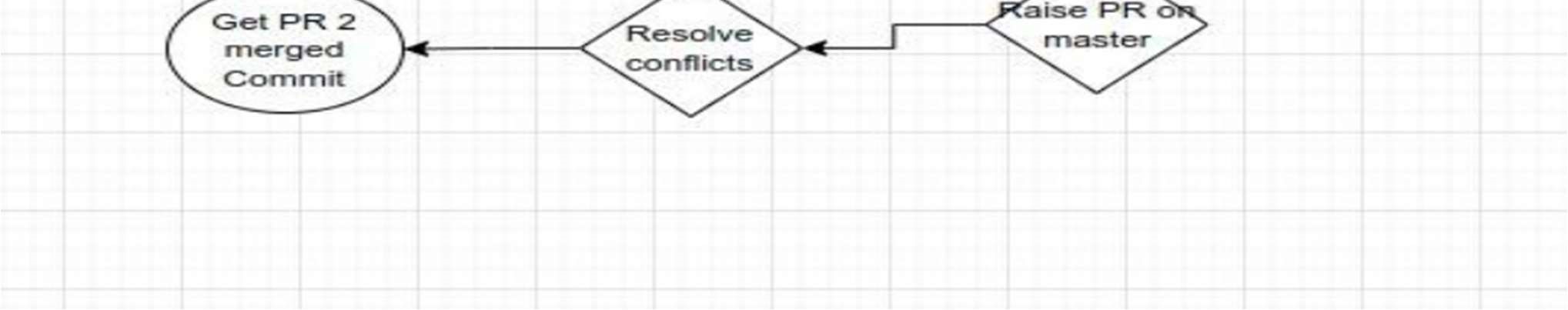
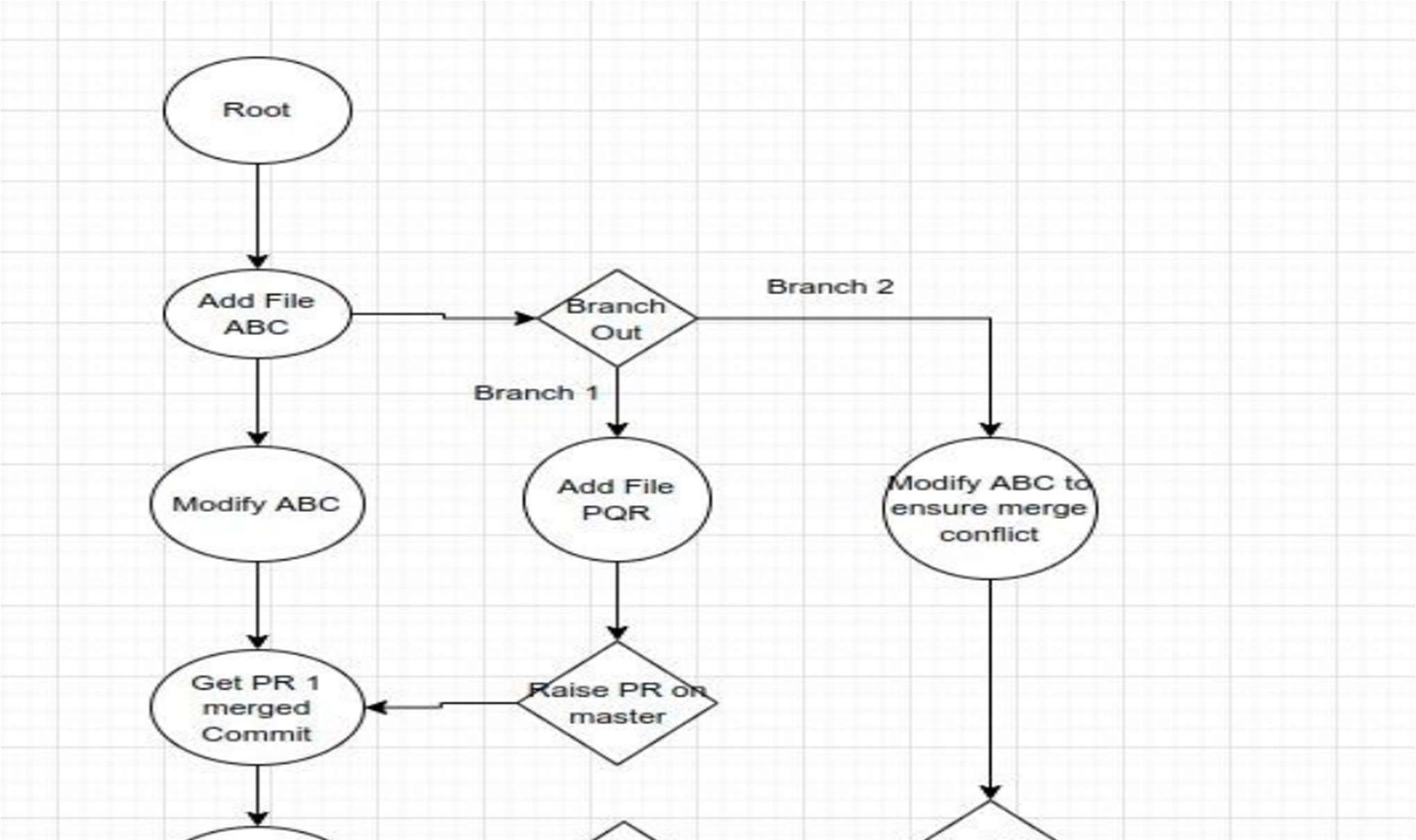
**6. Perform Fork operation**

Forking a repository in GitHub means creating a copy of an existing repository into your own account. Forking allows you to freely experiment with changes without affecting the original repository.

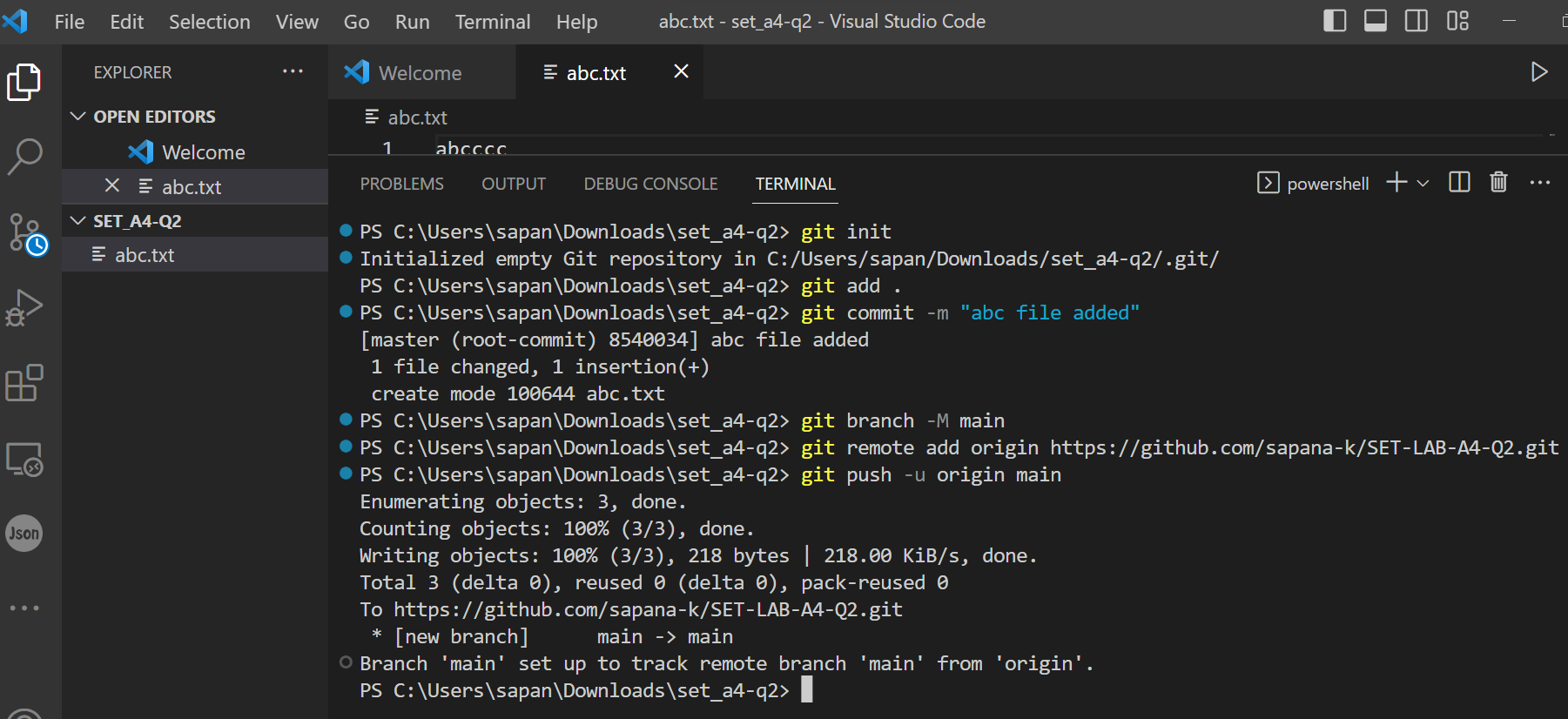




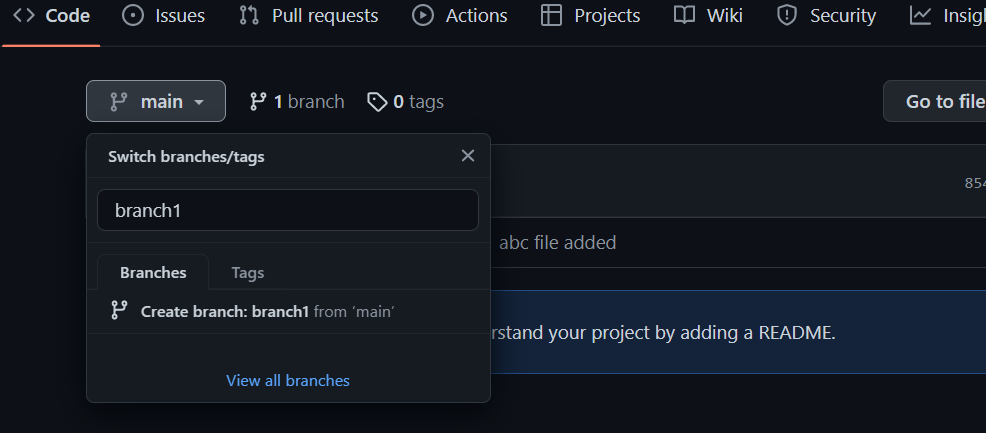
**Q 2. For the diagram given below create a GitHub repository and perform operations given in the diagram. (Perform commit operations as given)(Add screenshots as an answer to this question)**

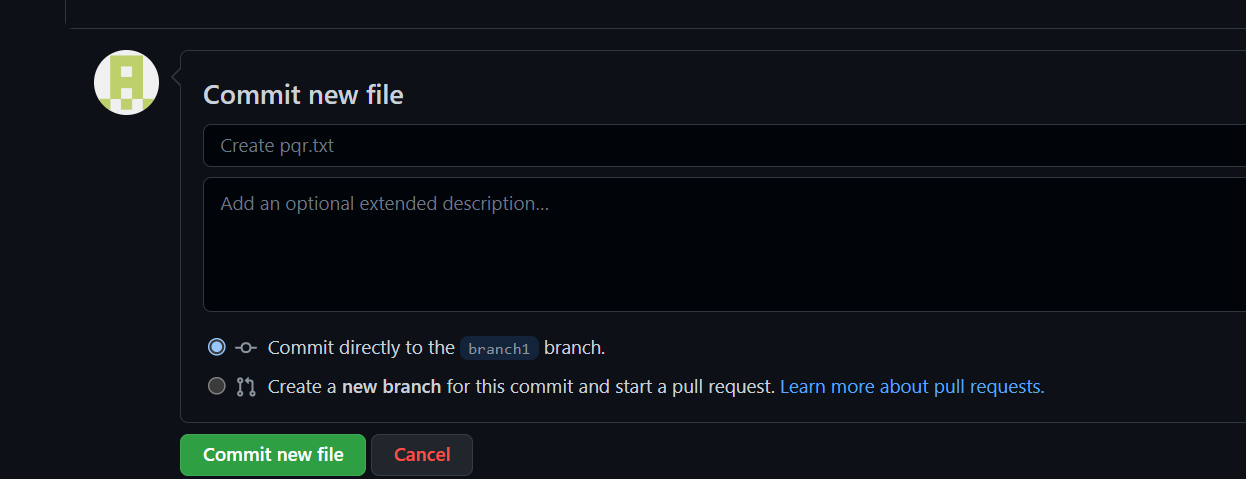
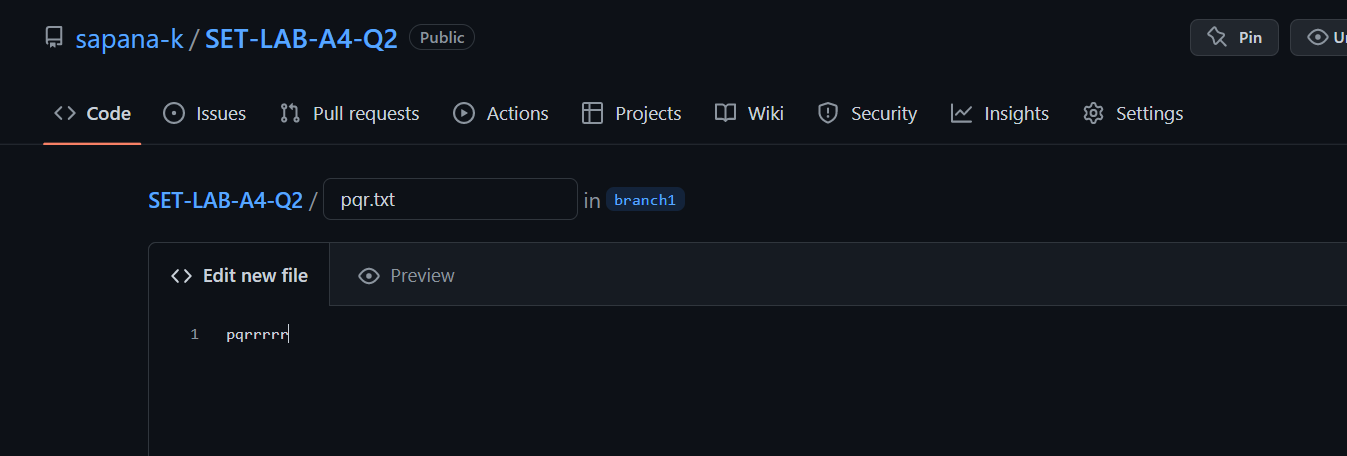


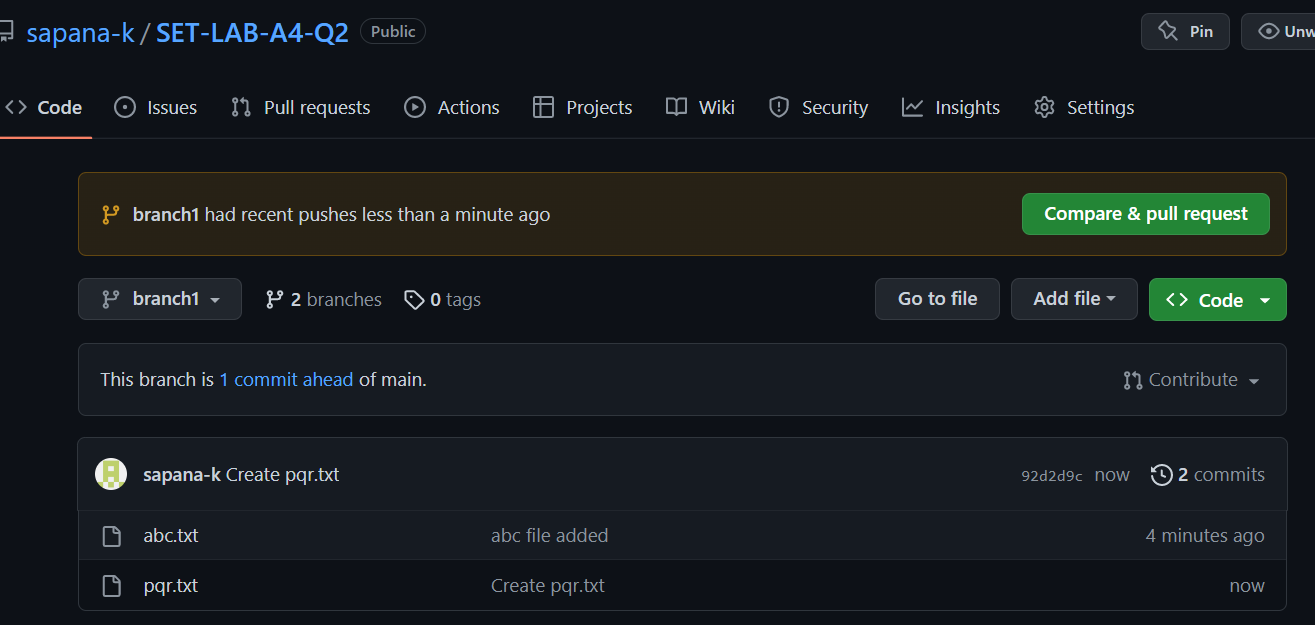
Add abc.txt in root repository



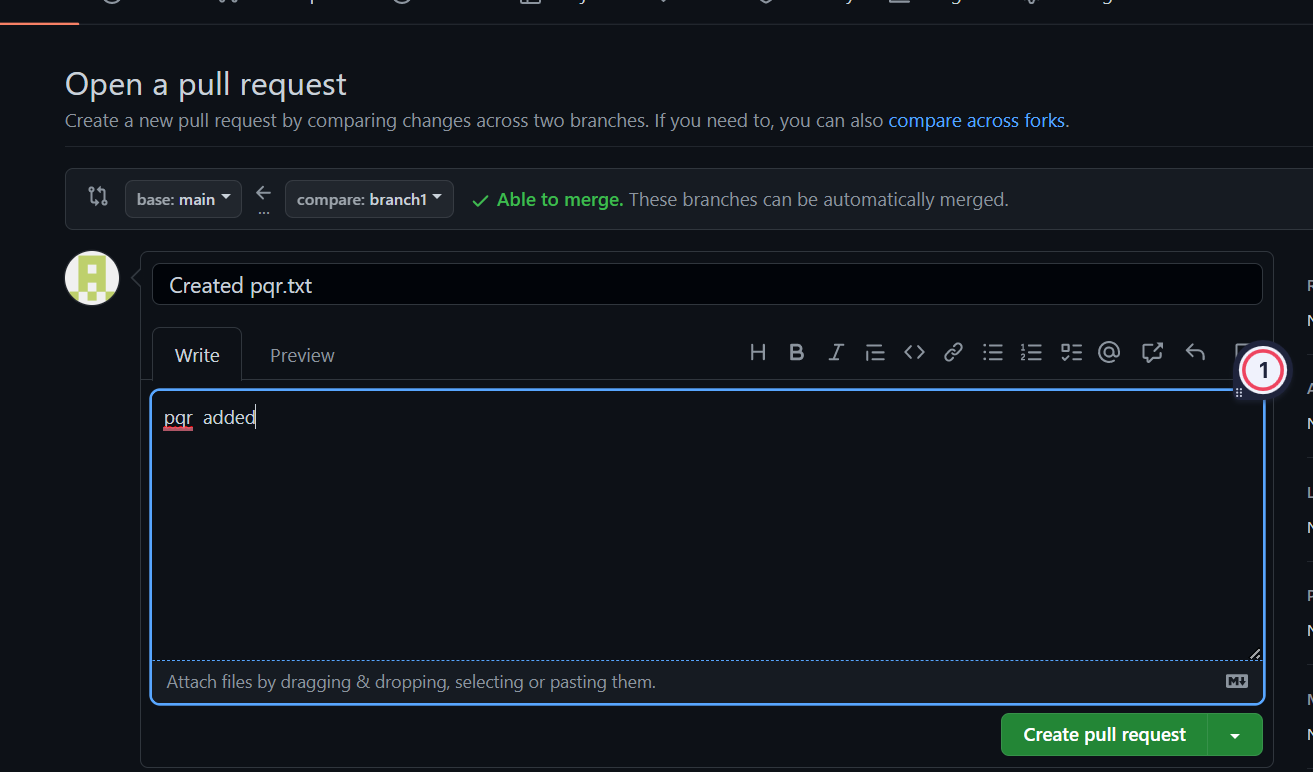
Create branch1 and add pqr.txt



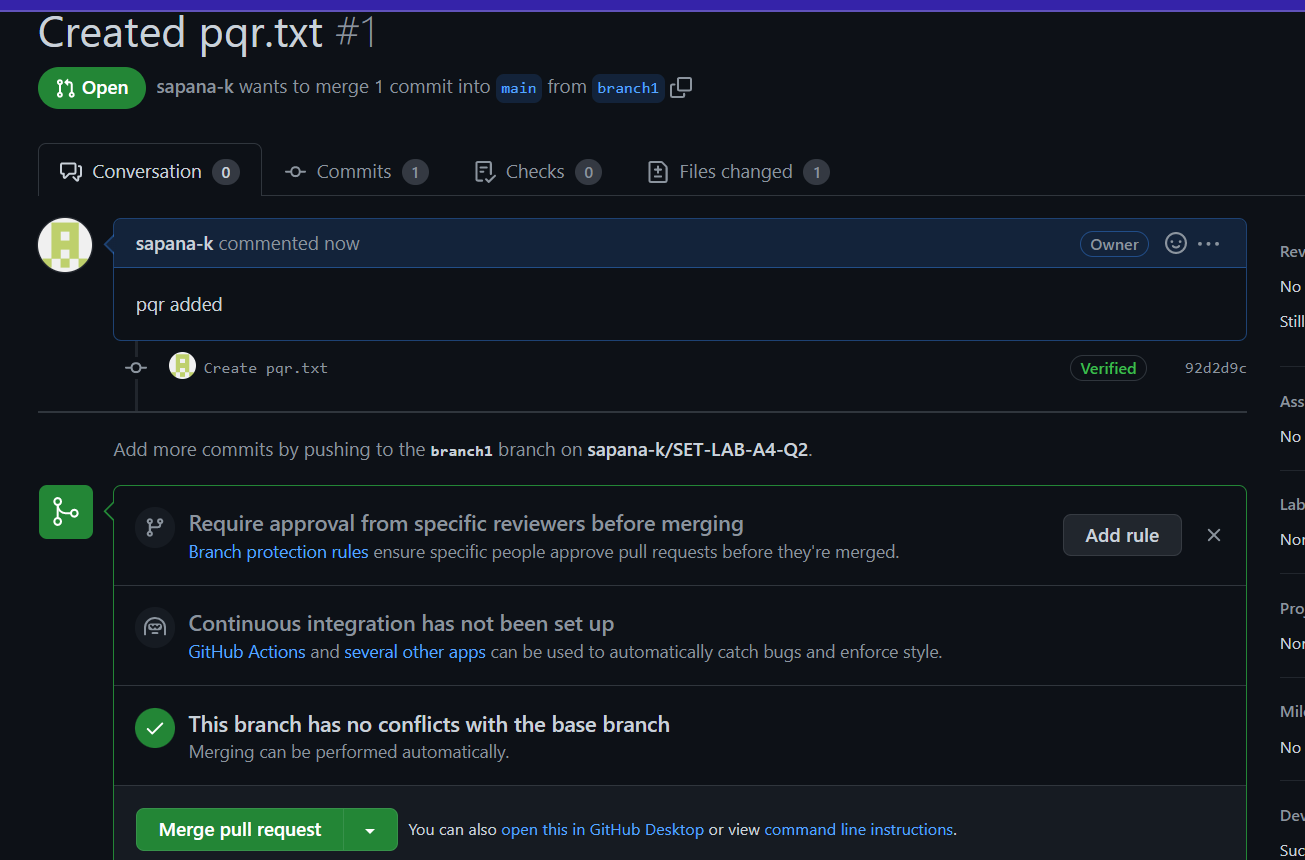


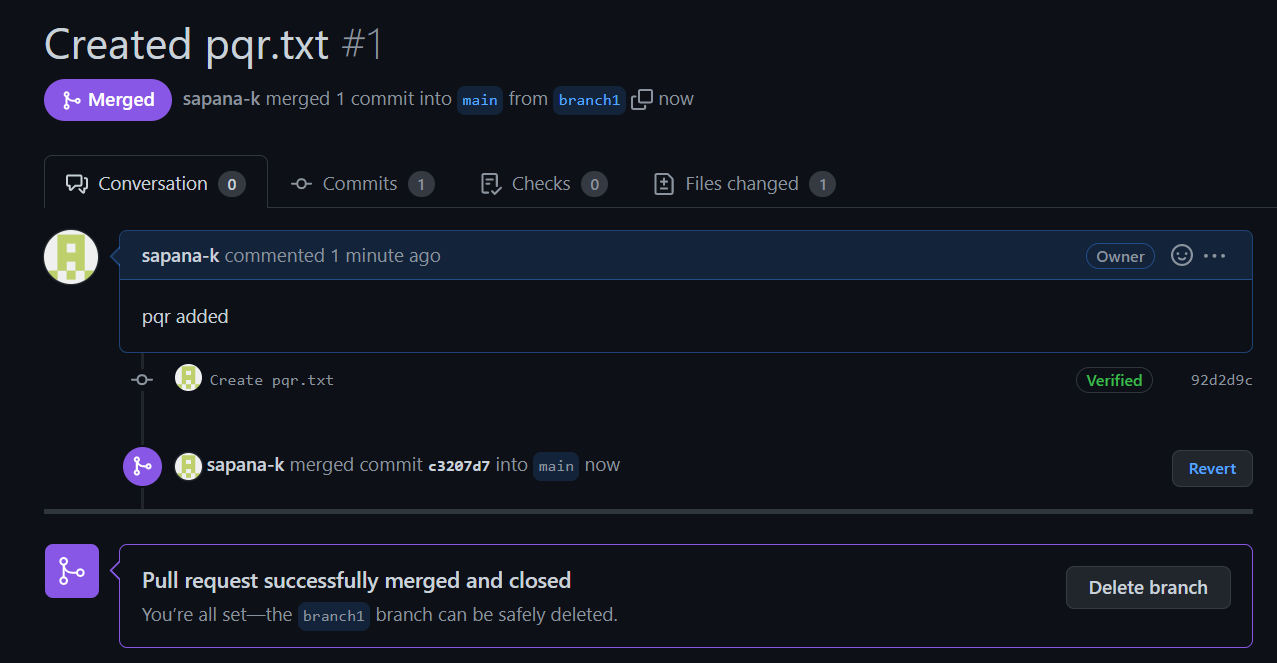


Raise Pull request on main from branch1



Merge pull request and commit

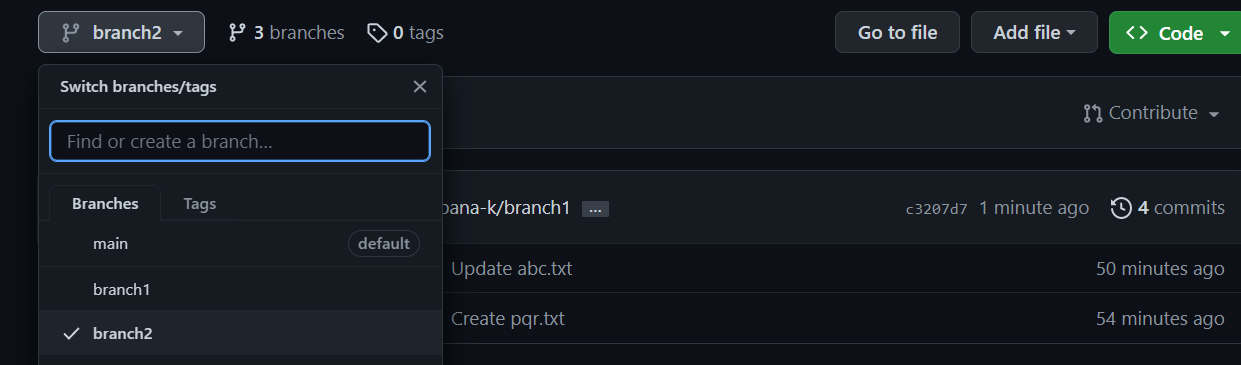




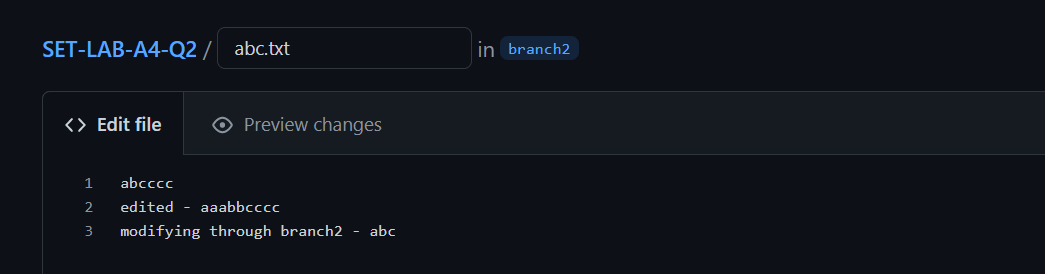
Modify abc.txt from main



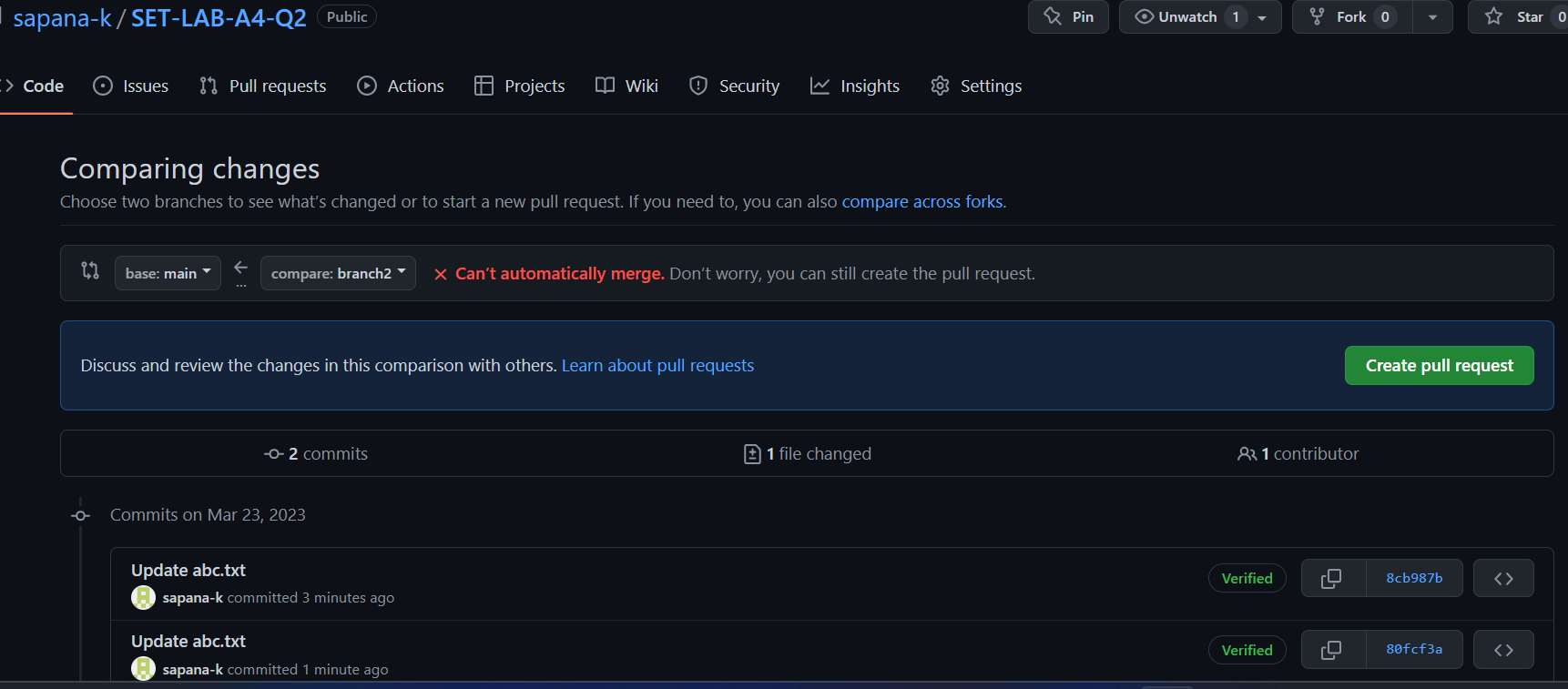
Create branch2

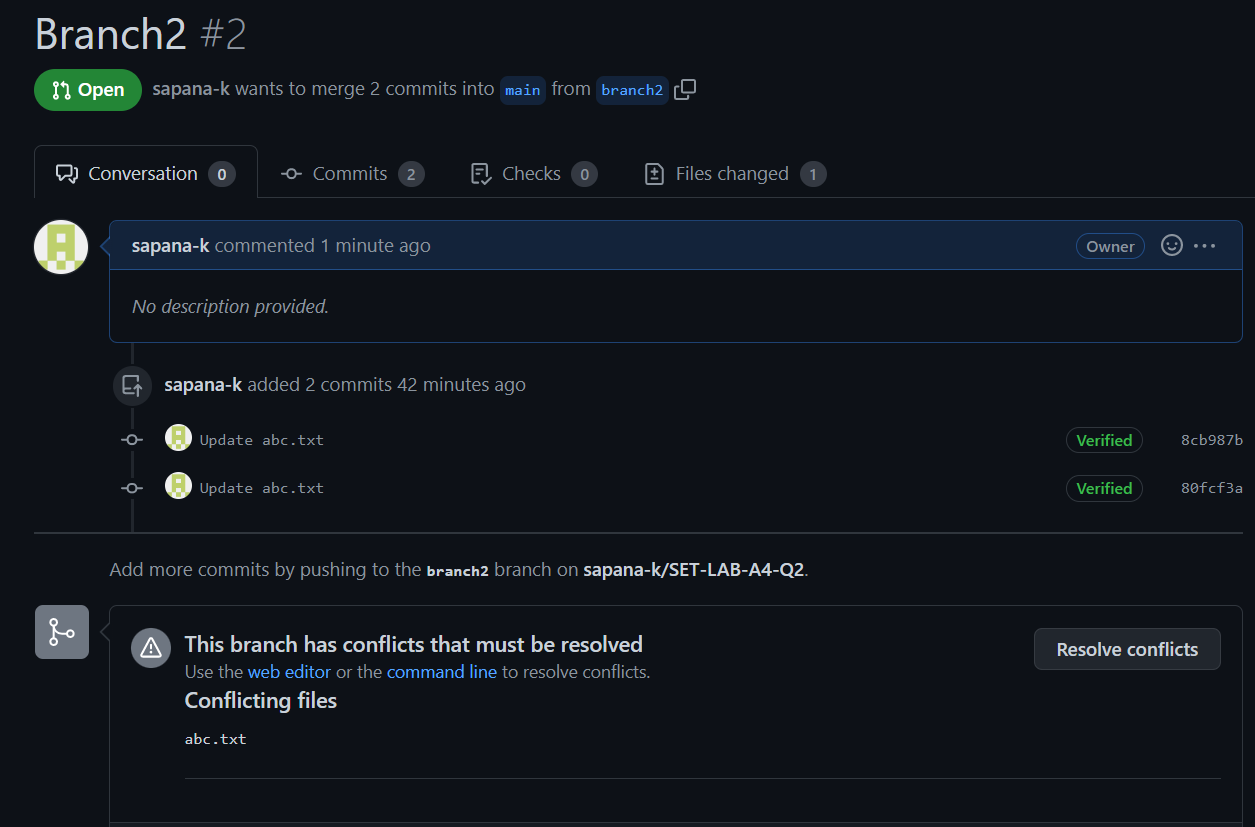


Modify abc from branch2

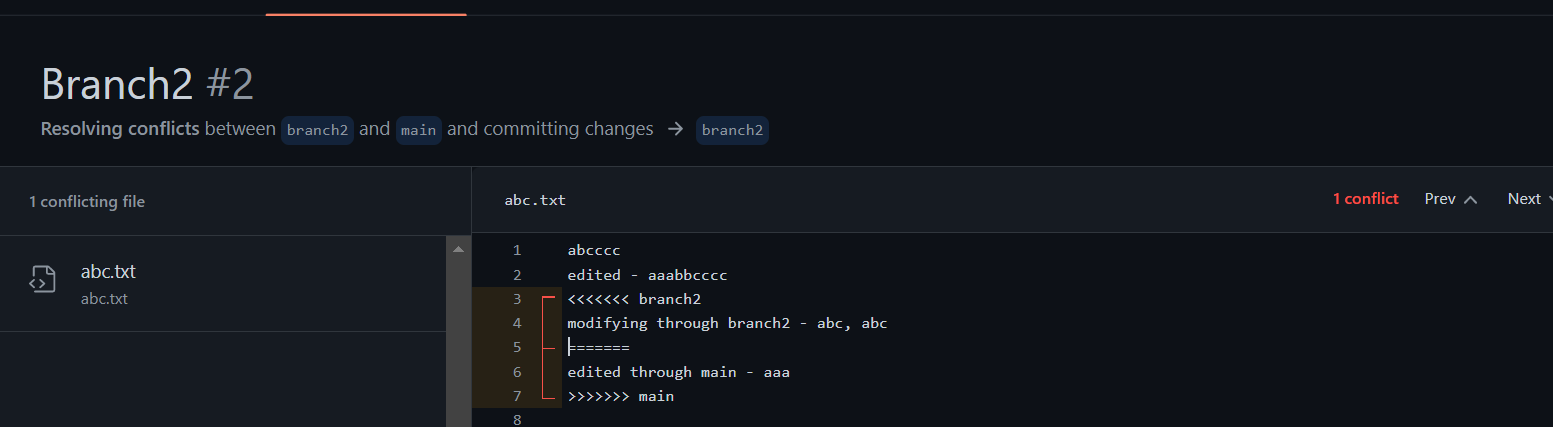


Create pull request and merge

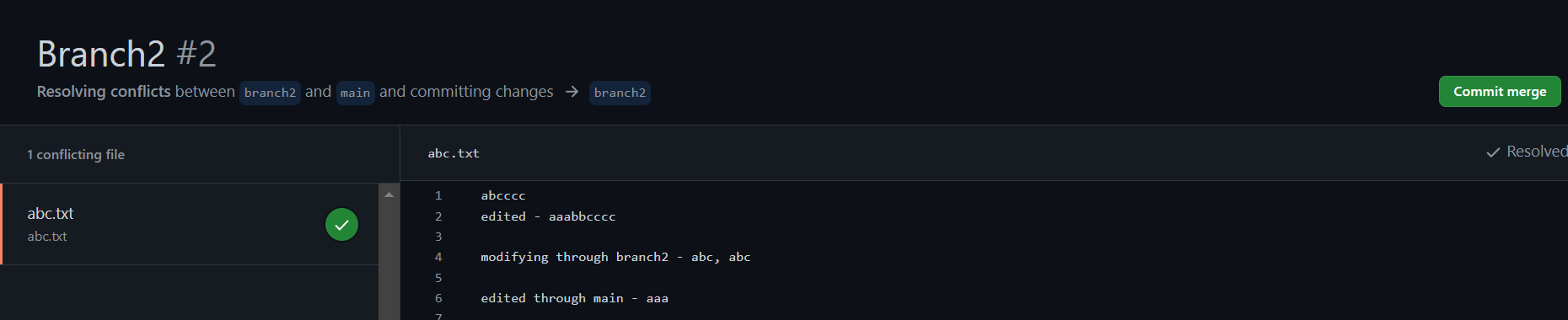


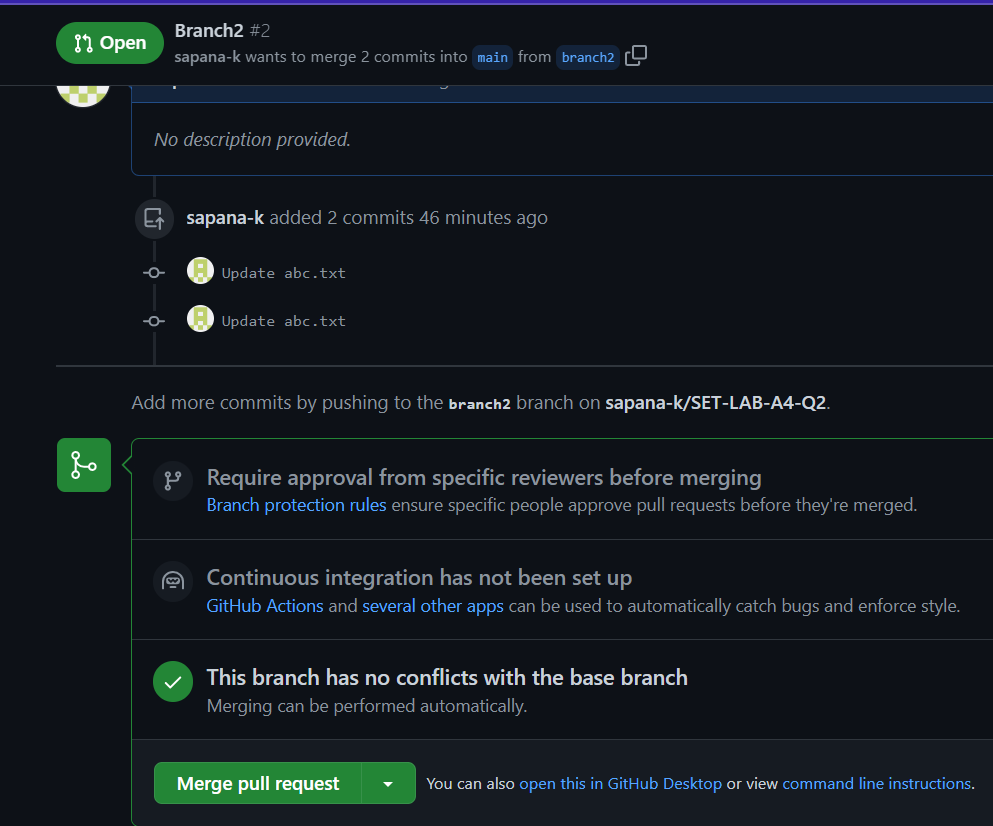


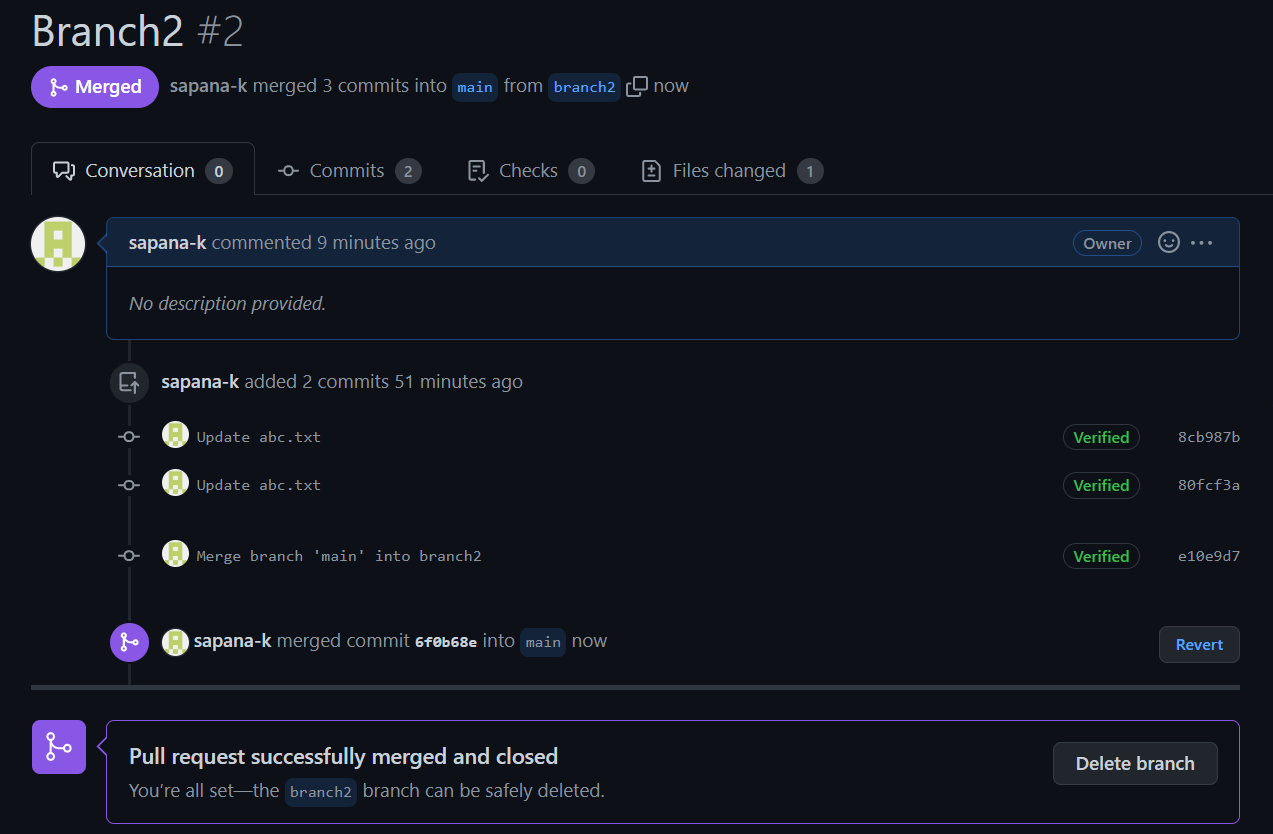
Resolve conflicts



Merge and commit





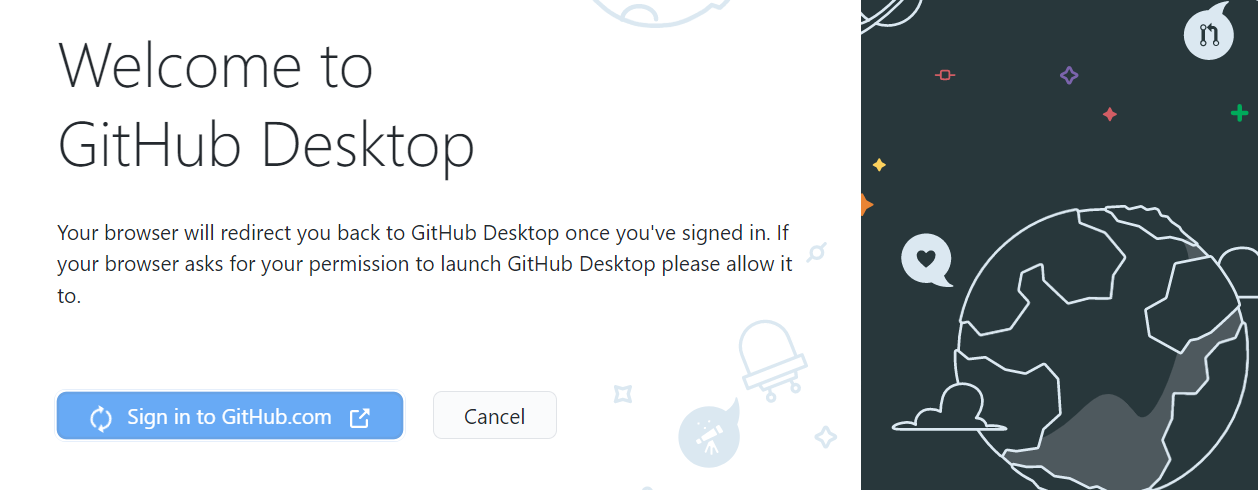


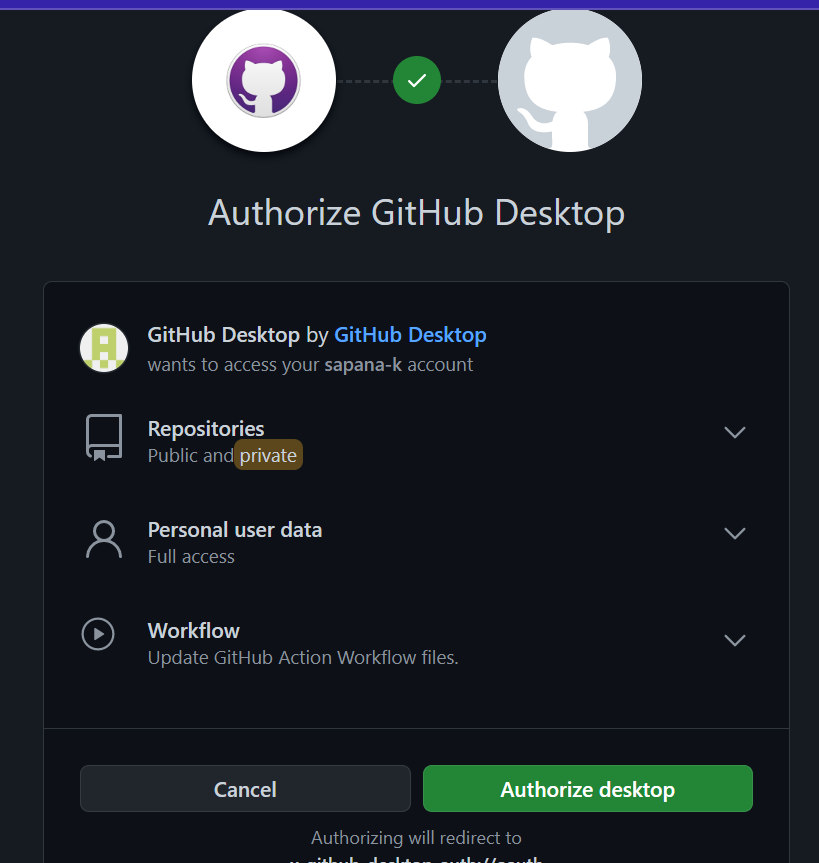
**Q 3. What is GitHub desktop? How to install GitHub on local machine? Install GitHub on your local machine and access repository created in question no 1 (add screenshots).**

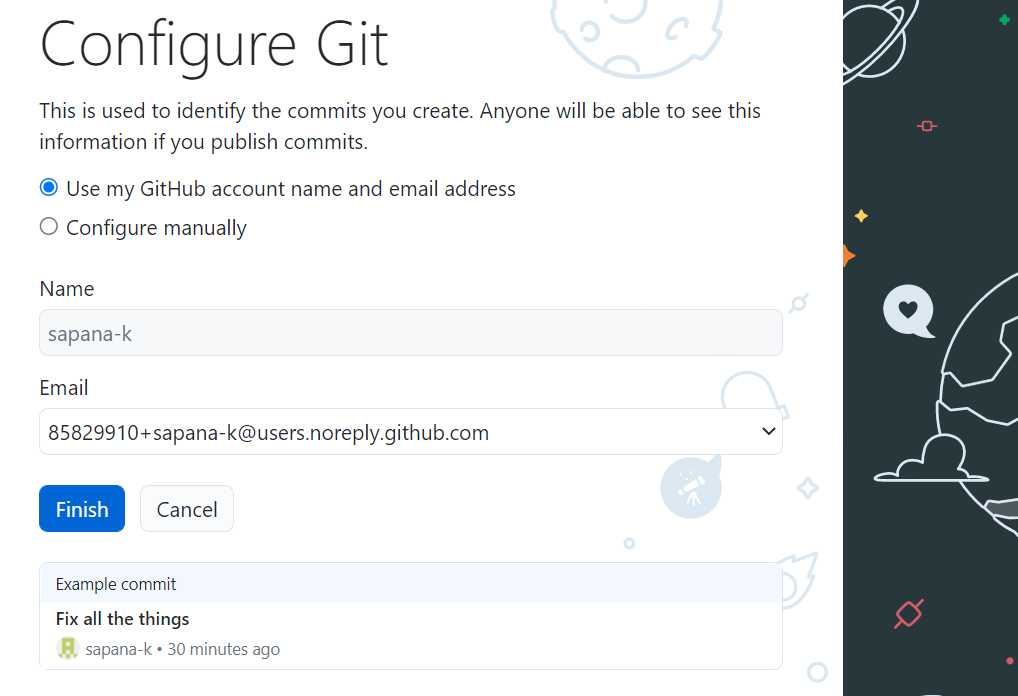
GitHub Desktop is an application that enables you to interact with GitHub using a GUI instead of the command line or a web browser. GitHub Desktop encourages you and your team to collaborate using best practices with Git and GitHub.

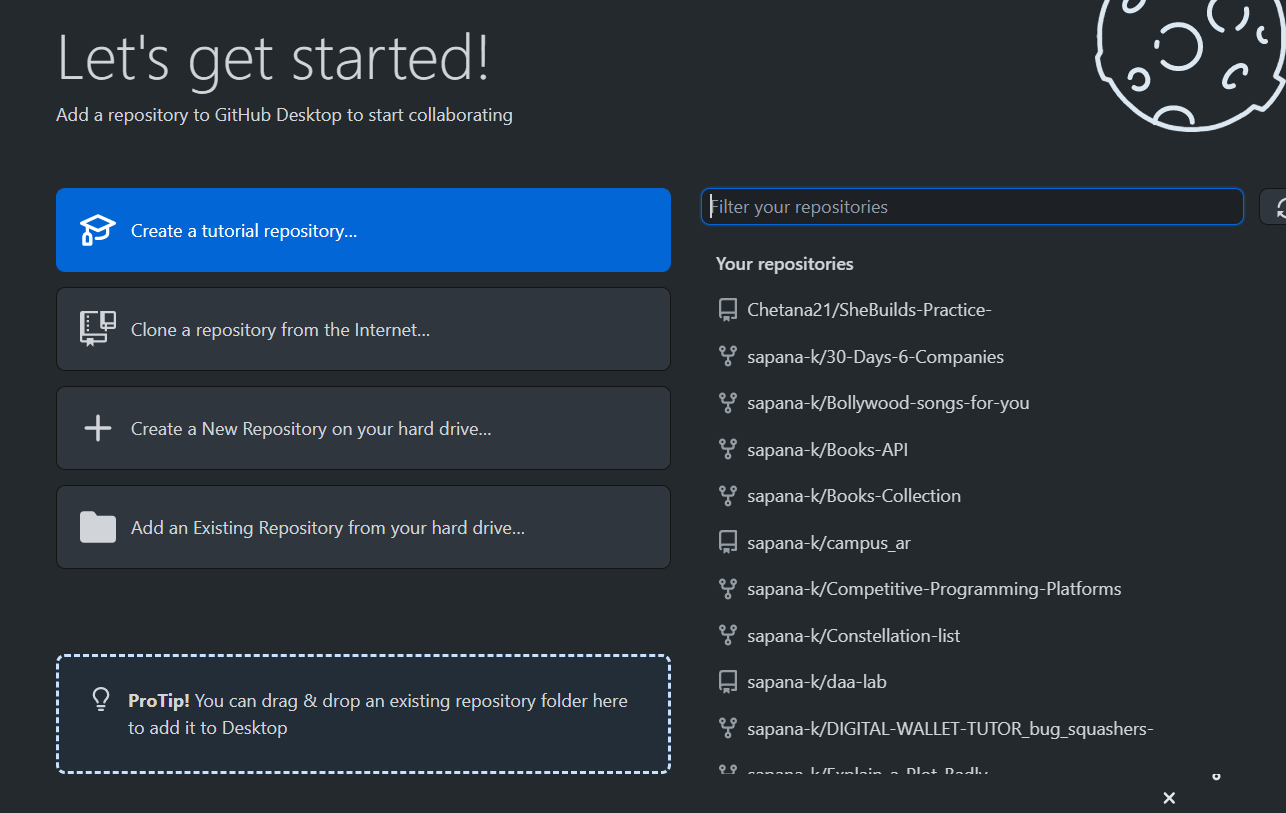
* Visit the download page for GitHub Desktop. Click Download for Windows.
* In your computer's Downloads folder, double-click the GitHub Desktop setup file. GitHub Desktop will launch after installation is complete.

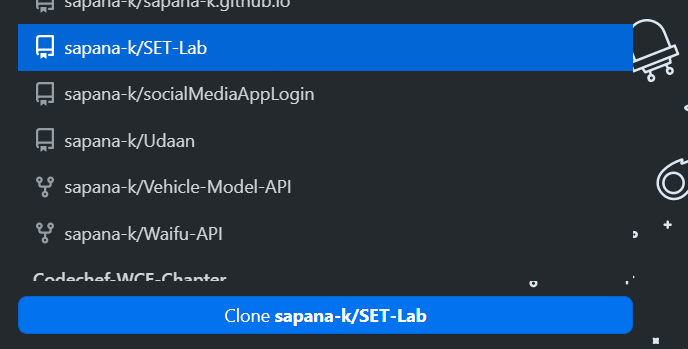


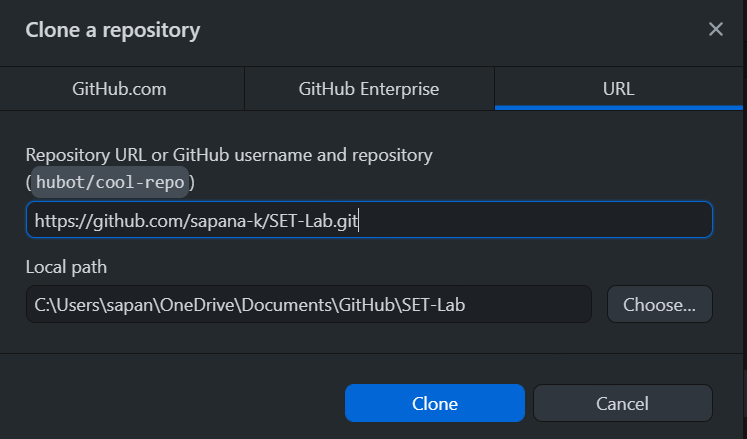


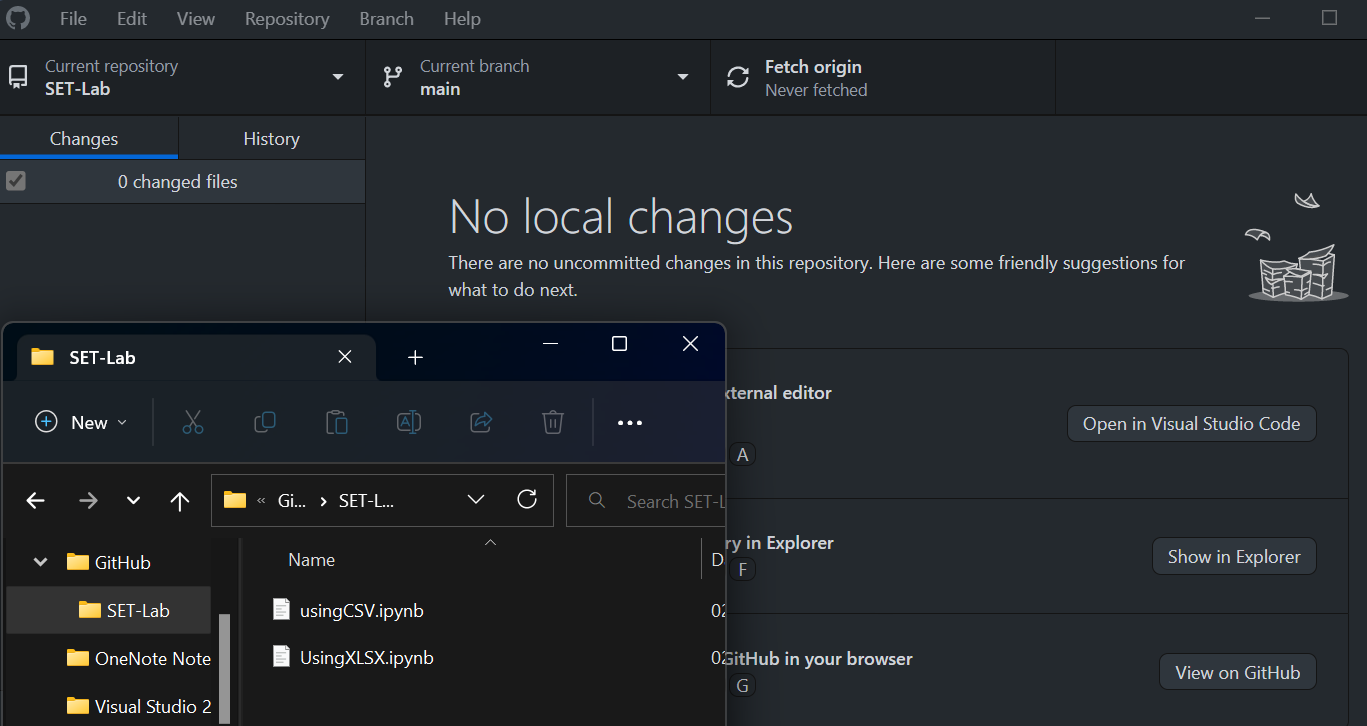












**Q 4. Differentiate in between GitHub, Git and GitLab.**

Git, GitHub, and GitLab are all related to version control systems, but they have different functionalities and purposes.

* Git is a distributed version control system designed to manage source code history. It was created by Linus Torvalds, the same person who created Linux. Git allows multiple developers to work on the same project simultaneously, keeping track of changes and ensuring that everyone has access to the most up-to-date code. Git is often used in software development, but it can be used for any project that involves files that change over time.
* GitHub is a web-based platform that hosts Git repositories. It provides a graphical interface and a suite of tools to help manage Git repositories. GitHub makes it easy to collaborate on projects with other developers, track issues and bugs, and manage pull requests. It also offers additional features such as project management tools, code review tools, and integration with third-party services.
* GitLab is similar to GitHub in that it provides a web-based platform for managing Git repositories, but it offers additional features such as continuous integration and deployment, issue tracking, and a built-in container registry. GitLab is designed to be a complete DevOps platform, providing tools for every stage of the software development lifecycle, from planning and coding to testing and deployment. GitLab can be self-hosted or used as a cloud-based service.

In summary, Git is the underlying version control system, while GitHub and GitLab are web-based platforms that provide a graphical interface and additional tools to manage Git repositories. GitHub is primarily used for collaboration and code sharing, while GitLab is designed as a complete DevOps platform with additional tools for continuous integration and deployment.

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

Version control is a software tool that manages changes to a set of files over time. It is commonly used in software development to keep track of changes to source code, but it can be used for any type of file that changes over time, such as documents, images, or configurations.

Version control systems track changes to files by creating a history of changes, or a timeline, that can be viewed, reverted, or merged. This allows multiple people to work on the same files simultaneously without creating conflicts or losing changes.

One popular version control system is Git, which is used by millions of developers around the world. Git allows developers to create a local copy of a repository, make changes to the files, and then merge those changes back into the central repository. Git also allows developers to revert changes or create branches to work on different features or fixes without affecting the main codebase.

For example, let's say a team of developers is working on a website. They use Git to manage changes to the codebase. Each developer creates a local copy of the repository on their computer. They make changes to the code, such as adding new features or fixing bugs, and then commit those changes to their local repository with a message describing the changes.

When they are ready to share their changes with the rest of the team, they push their changes to the central repository. Git tracks the changes and allows the team to review and merge the changes into the main codebase.

If there is a conflict between changes made by different developers, Git will alert them and provide tools to resolve the conflict, such as merging the changes or reverting one set of changes.

Version control systems like Git provide a way for developers to collaborate on a project without creating conflicts or losing changes. They also provide a history of changes that can be used to track bugs, revert changes, or audit code changes.

and ensuring the integrity of the code.