Software Engineering Tools Lab
Assignment No-4

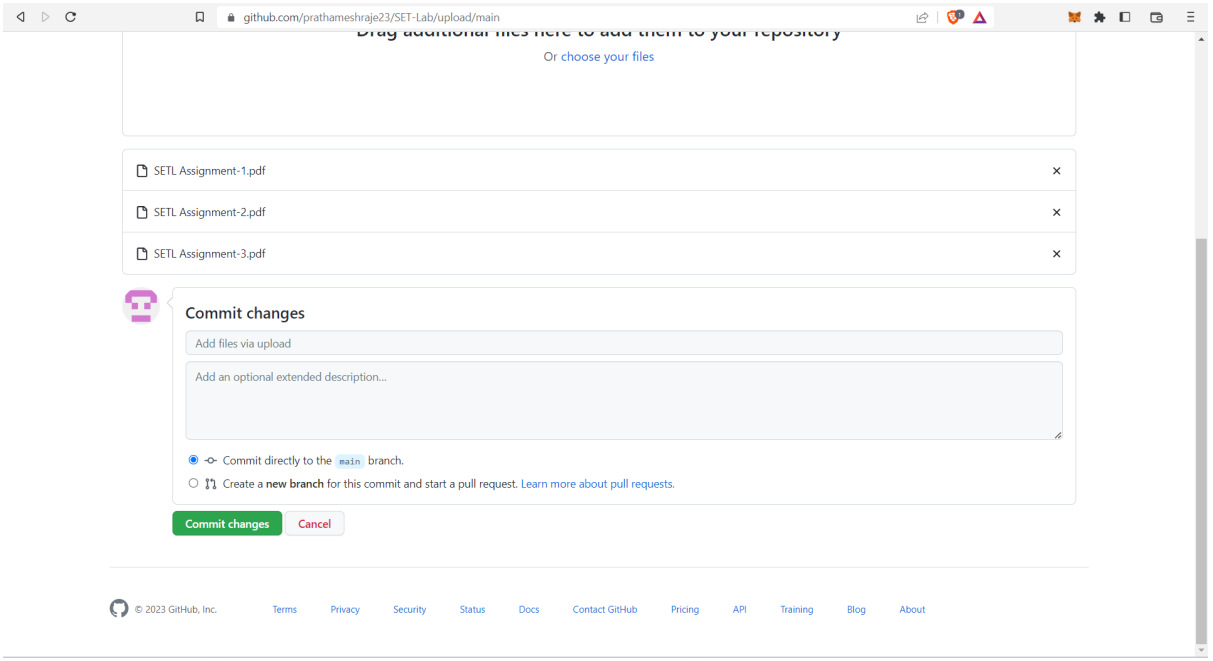
PRN: 2020BTECS00033
Name: Prathamesh Raje
Batch: T6

Title - GitHub

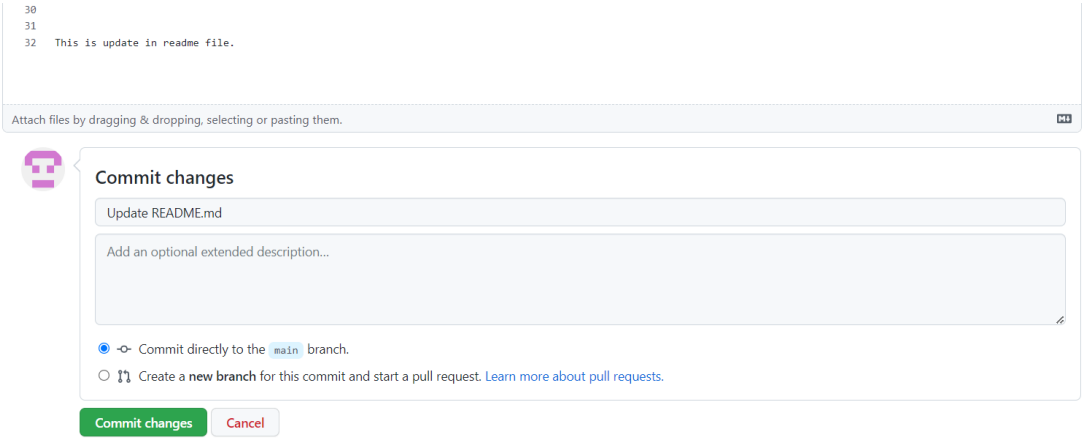
Q 1. Create a repository on GitHub named SET Lab and add files into it (you can add implementation files of previous assignment) perform below operations on it.
(Add screenshot as an answer to every question)

Ans.

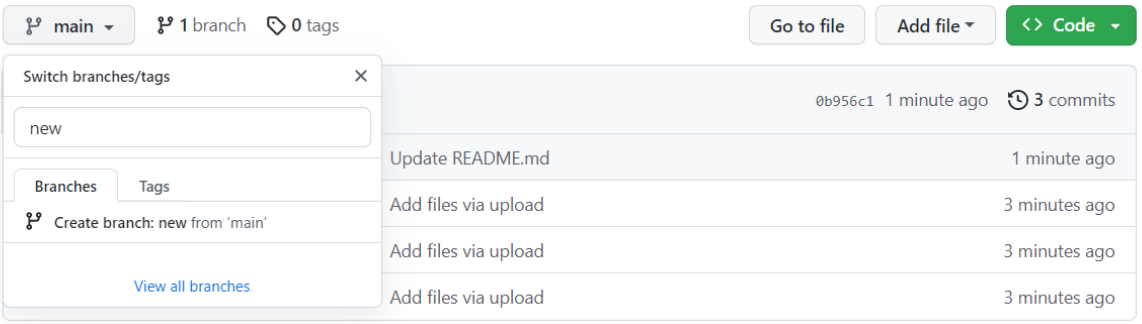
1. Perform commit on added files



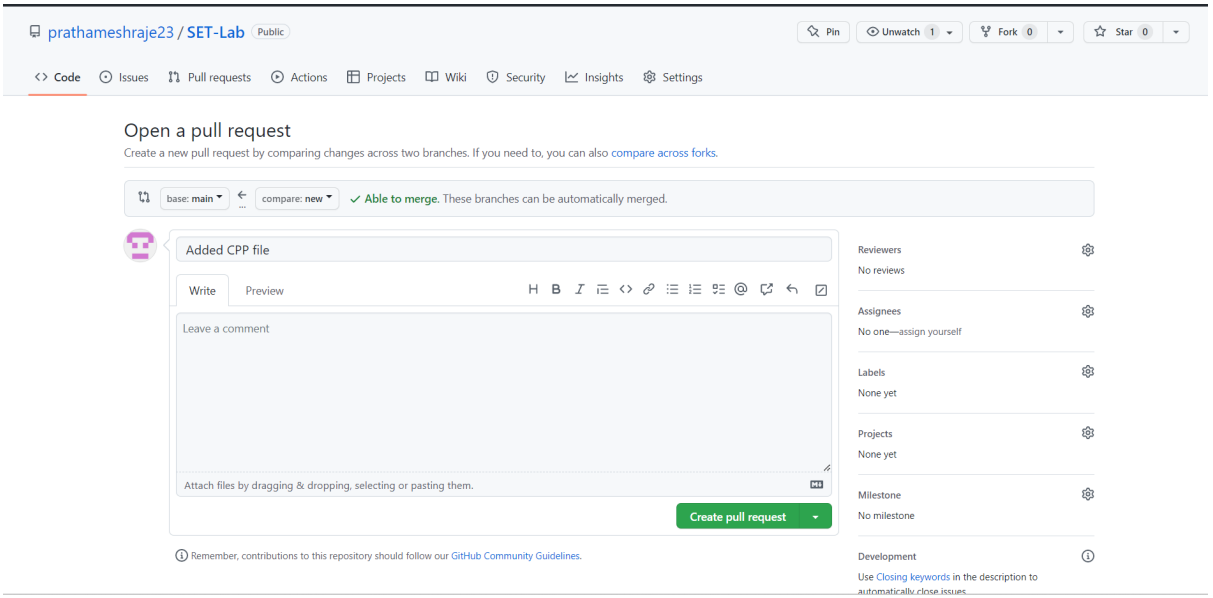
2. Perform update to the existing files (show history)



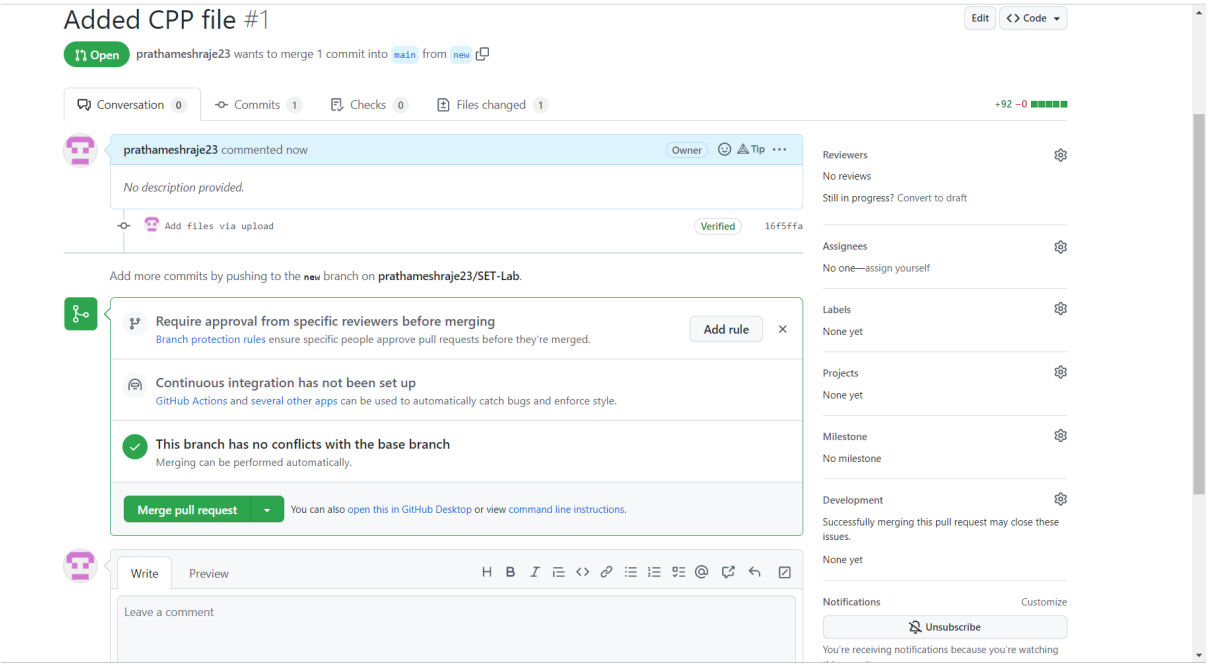
3. Create another branch



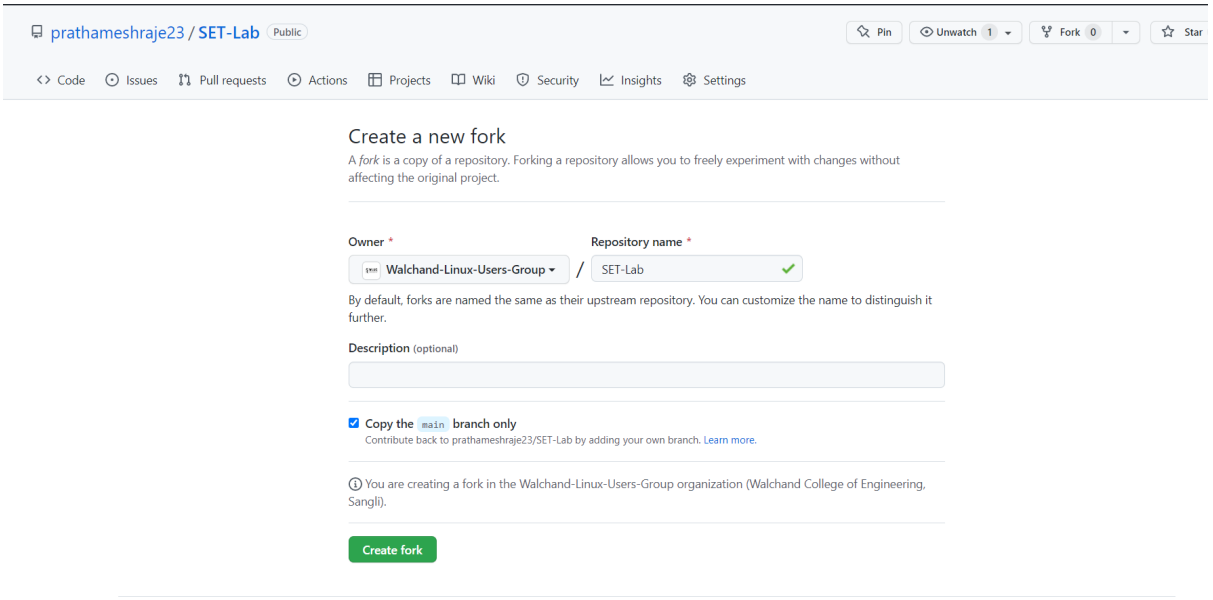
4. Create pull request



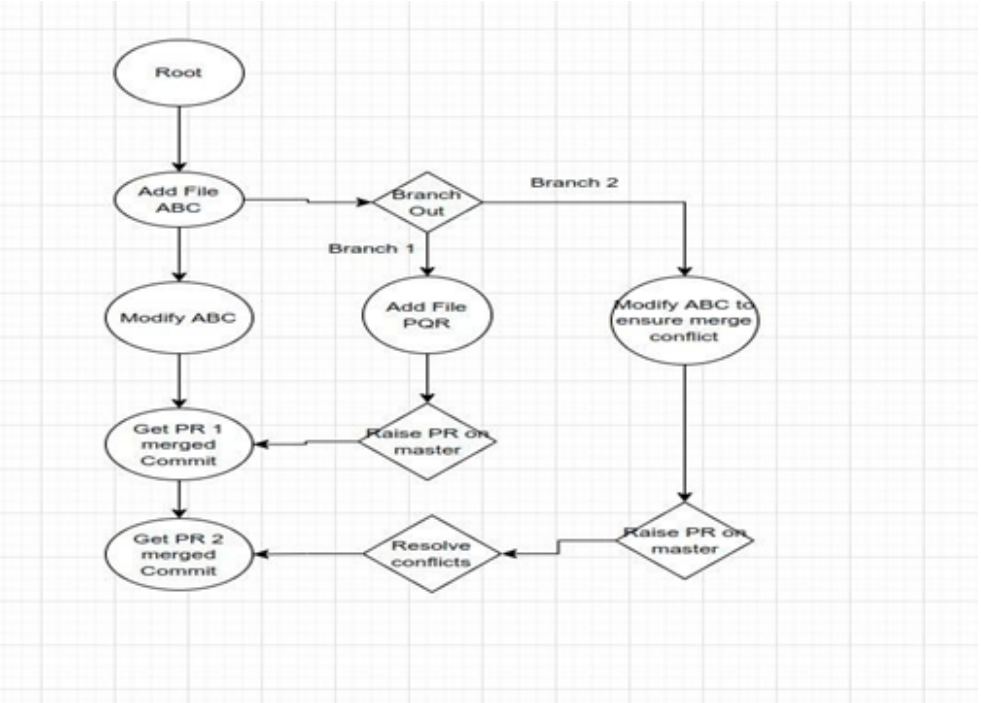
5. Perform merging of both branches



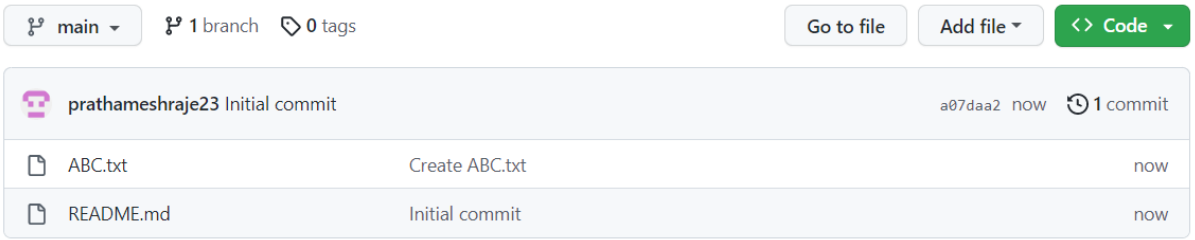
6. Perform Fork operation



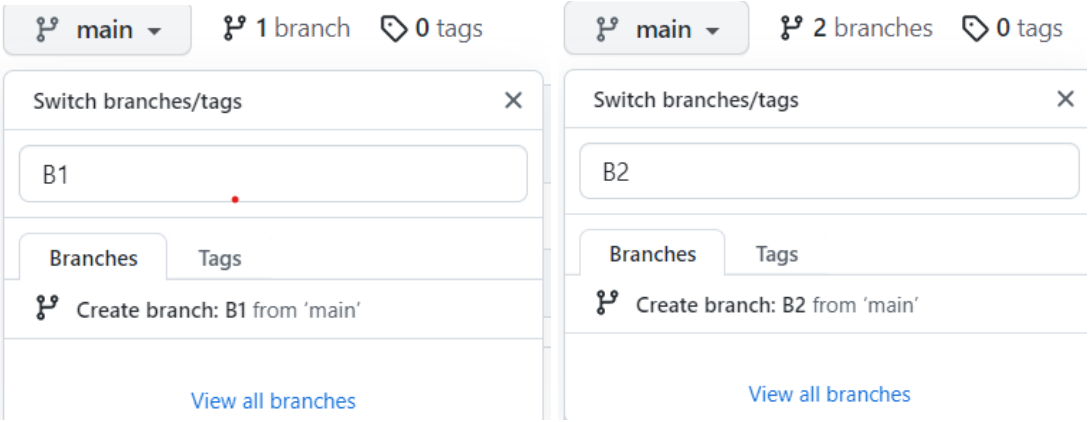
Q 2. For the diagram given below create a GitHub repository and perform operations given in the diagram. (Perform commit operations as given)(Add screenshots as an answer to this question)



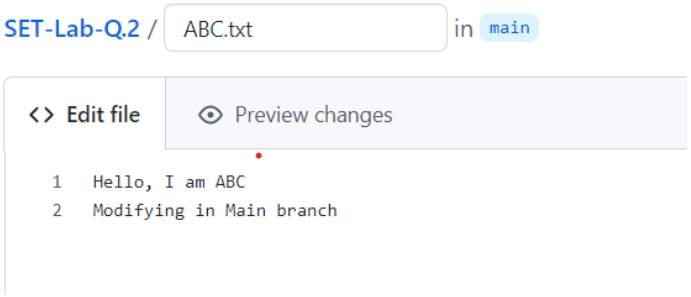
1. ABC.txt in Main Branch:



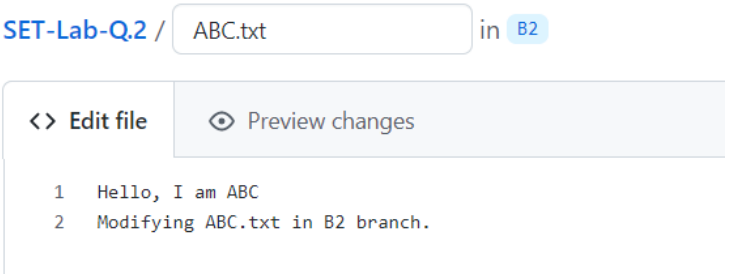
2. Creating Branch:



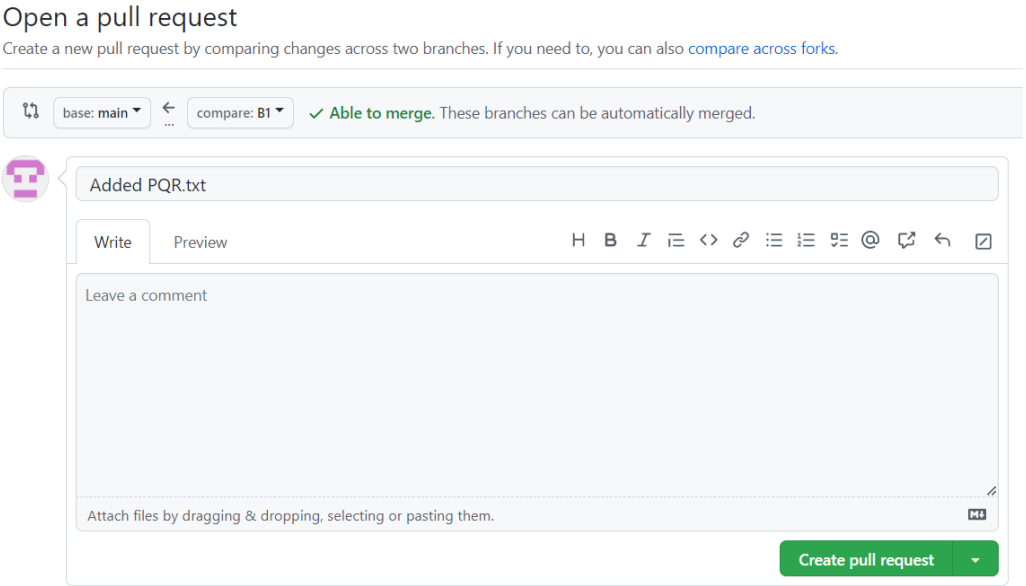
3. Adding PQR.txt in branch B1
4. Modifying ABC.txt in Main branch

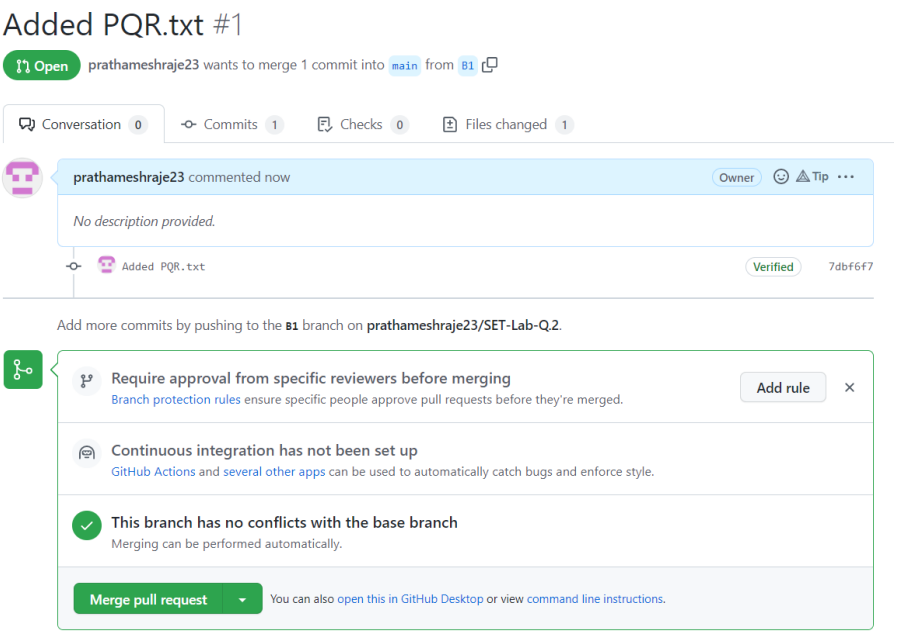


5. Modifying ABC.txt in B2 branch to ensure merge conflict

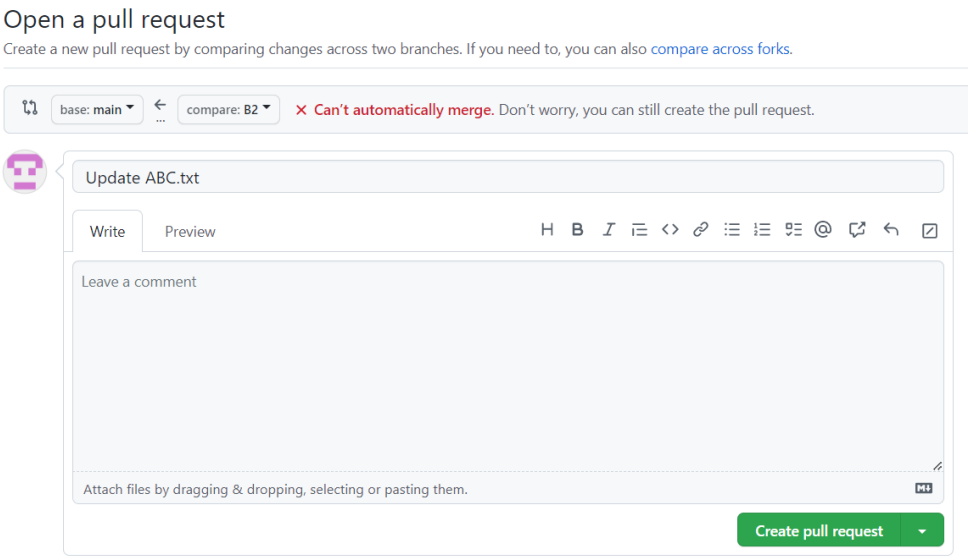


6. Raising and merging PR1 from B1

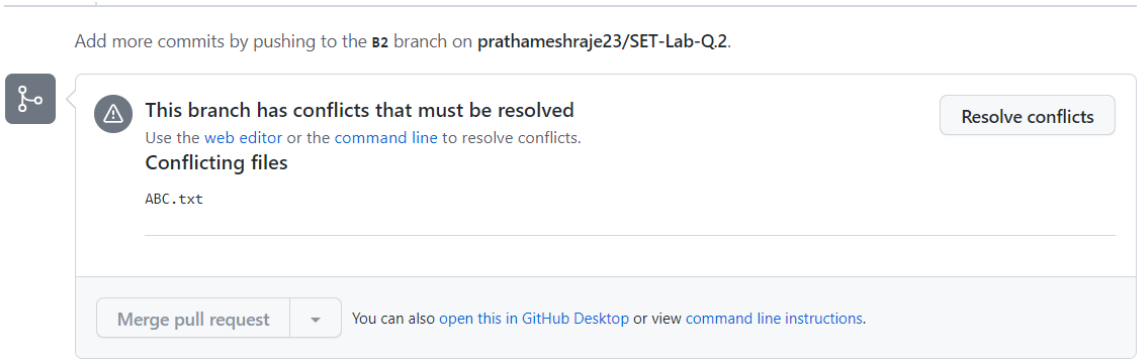




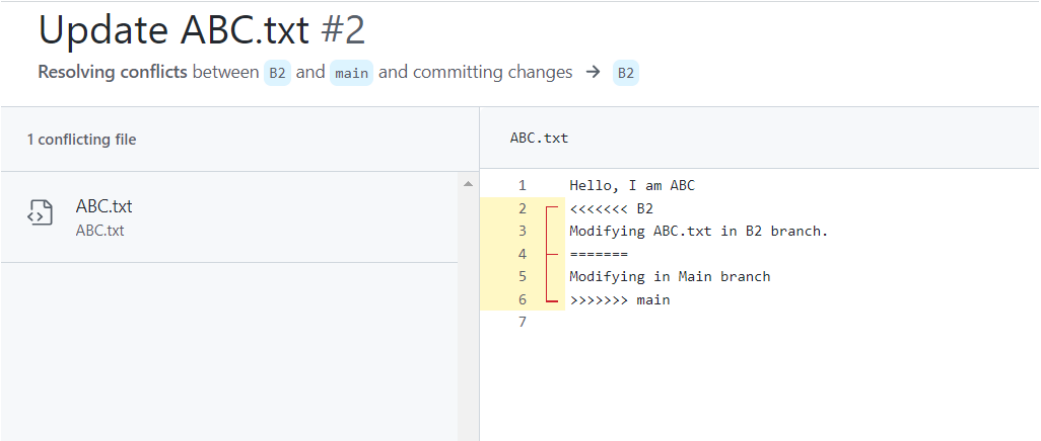
7. Raising PR2 from B2 and resolving merge conflict



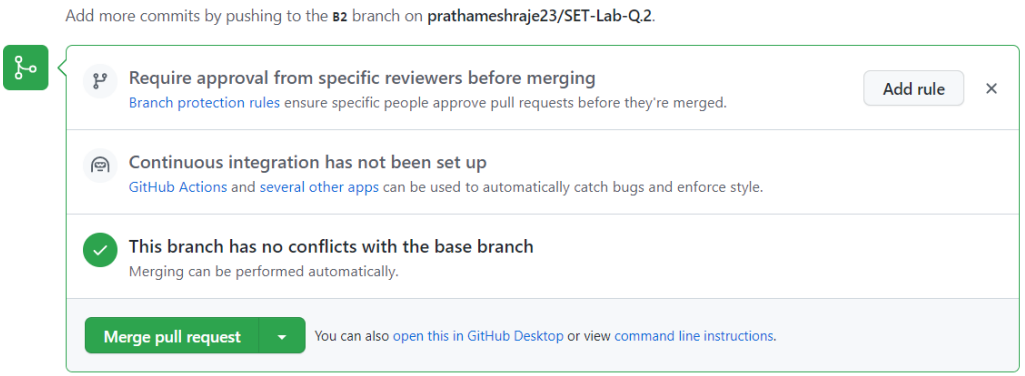
8. Conflict



9. Resolve Conflict



10. Final merge after resolving



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

GitHub Desktop is an open-source application that lets you interact with GitHub via a graphic user interface (GUI) instead of relying on a command line or web browser. GitHub Desktop incentivizes you and your team to work together while employing best practices with Git and GitHub.

GitHub Desktop enables developers to activate commands such as repository creation, pull requests, and commits with just a simple click. This extra convenience adds an extra element of flexibility to working with Git and collaborating with other developers.

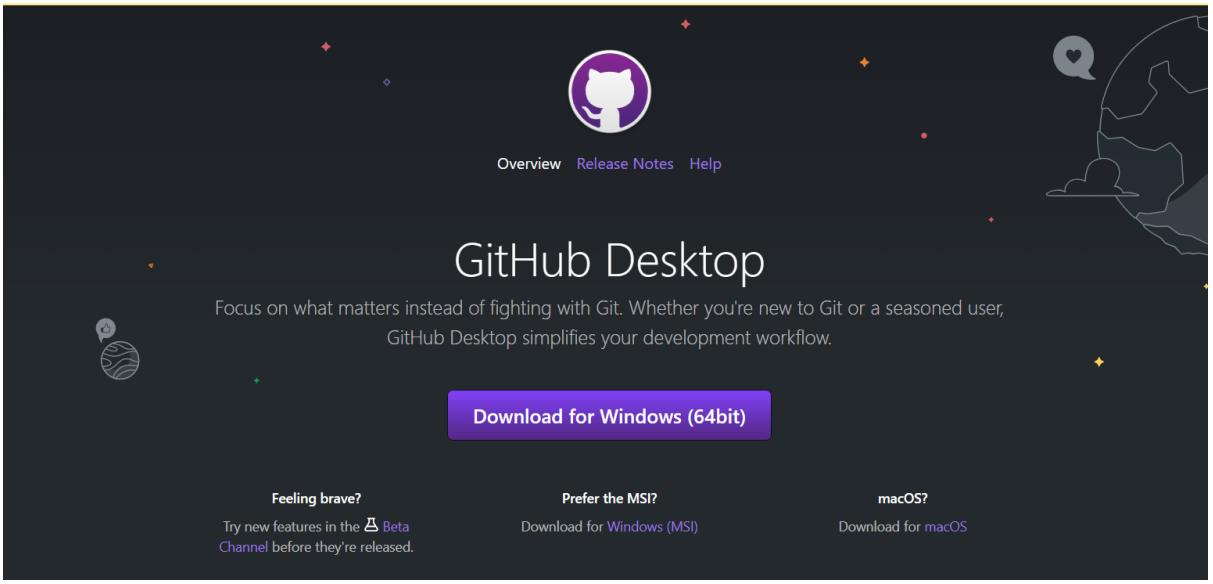
So, to sum it up, Git is a version control system that helps you manage your code and keep track of it, and GitHub is a cloud-based hosting platform that enables developers to manage their Git repositories. GitHub Desktop is an application that lets users interact better with GitHub through a GUI.

The installation of GitHub Desktop is as simple as any other Windows application installation. All you need to do is:

Open a browser.

Visit desktop.github.com.

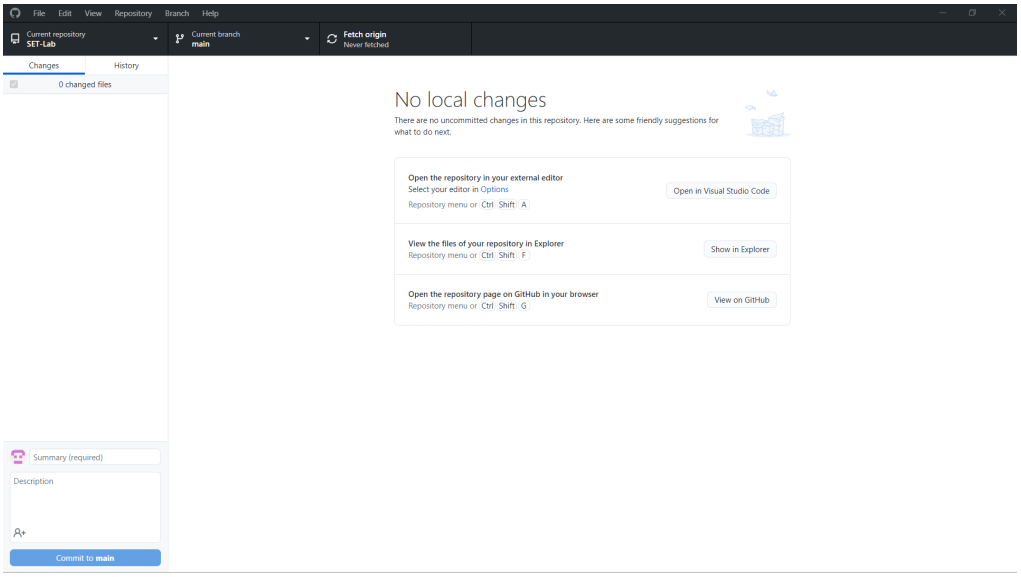
Click Download for Windows (64bit).



When prompted, click Run.

Allow the installation to download and install.

Once the installation completes, GitHub Desktop will launch.



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

Git	GitHub
1. It is a software	1. It is a service
2. It is installed locally on the system	2. It is hosted on Web
3. It is a command line tool	3. It provides a graphical interface
4. It is a tool to manage different versions of edits, made to files in a git repository	4. It is a space to upload a copy of the Git repository
5. It provides functionalities like Version Control System Source Code Management	5. It provides functionalities of Git like VCS, Source Code Management as well as adding few of its own features

Q 5. What is version control? Explain with examples.
Ans.

1. Version control (also known as revision control or source control) is a category of processes and tools designed to keep track of multiple different versions of software, content, documents, websites and other information in development.
2. Any system that provides change tracking and control over programming source code and documentation can be considered version control software.

3. The practice has been a part of creative processes almost as long as writing has existed.
4. The purpose of version control is ensuring that content changes under development go as planned.
5. While version control is often carried out by a separate application, it can also be embedded into programs such as integrated development environments (IDEs), word processors, spreadsheets and, especially, collaborative web documents and pages.
6. Version control allows servers in multiple locations to run different versions on different sites, even while those versions are being updated simultaneously.
7. The most powerful and complex version control systems are used in software development.
8. Version control often operates by locking files and using a check-out / check-in system for changed versions. Versions may be identified by labels or tags; approved versions or those that are especially significant may be designated baselines.
9. Checked-out versions may be worked on by different groups or individuals as branching code from the main trunk.
10. When versions are checked out and checked in, the first to check in is sure to succeed. If other versions are checked out, some systems may provide for version merging to allow further changes to be added into the central repository.
11. Another method used in version control is branching, in which programs in development are copied for development in parallel versions, retaining the original and working on the branch or making different changes to each.
12. Each copy is considered a branch; the original program from which the branch is taken is referred to as the trunk, the baseline, the mainline or the master.
13. Version control is generally based on a client-server model. Another method is distributed version control, in which all copies are in a codebase repository and changes are synchronised through patches or changes shared from peer to peer.