Software Engineering Tools Lab

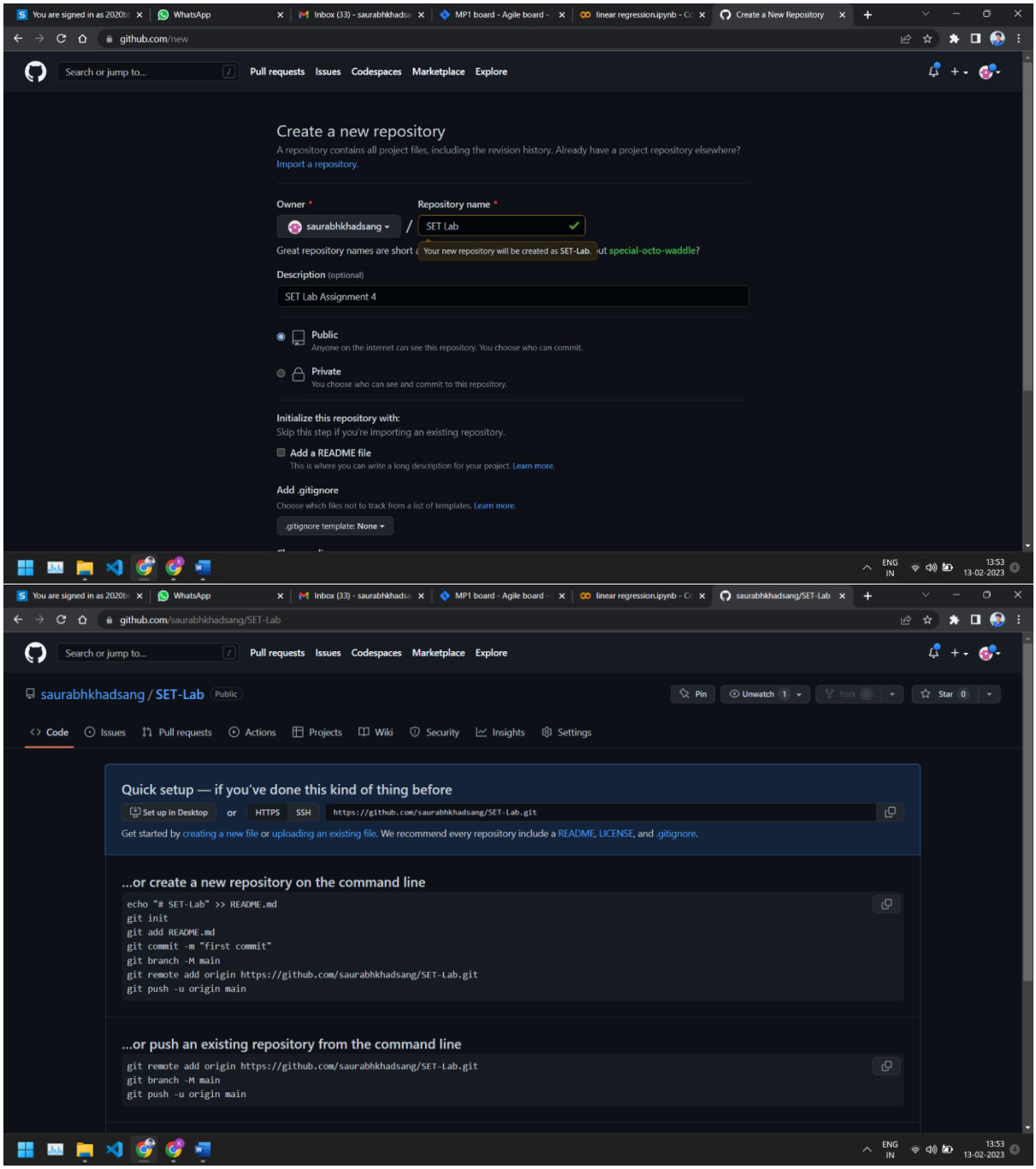
Assignment No-4

Module 3- GitHub

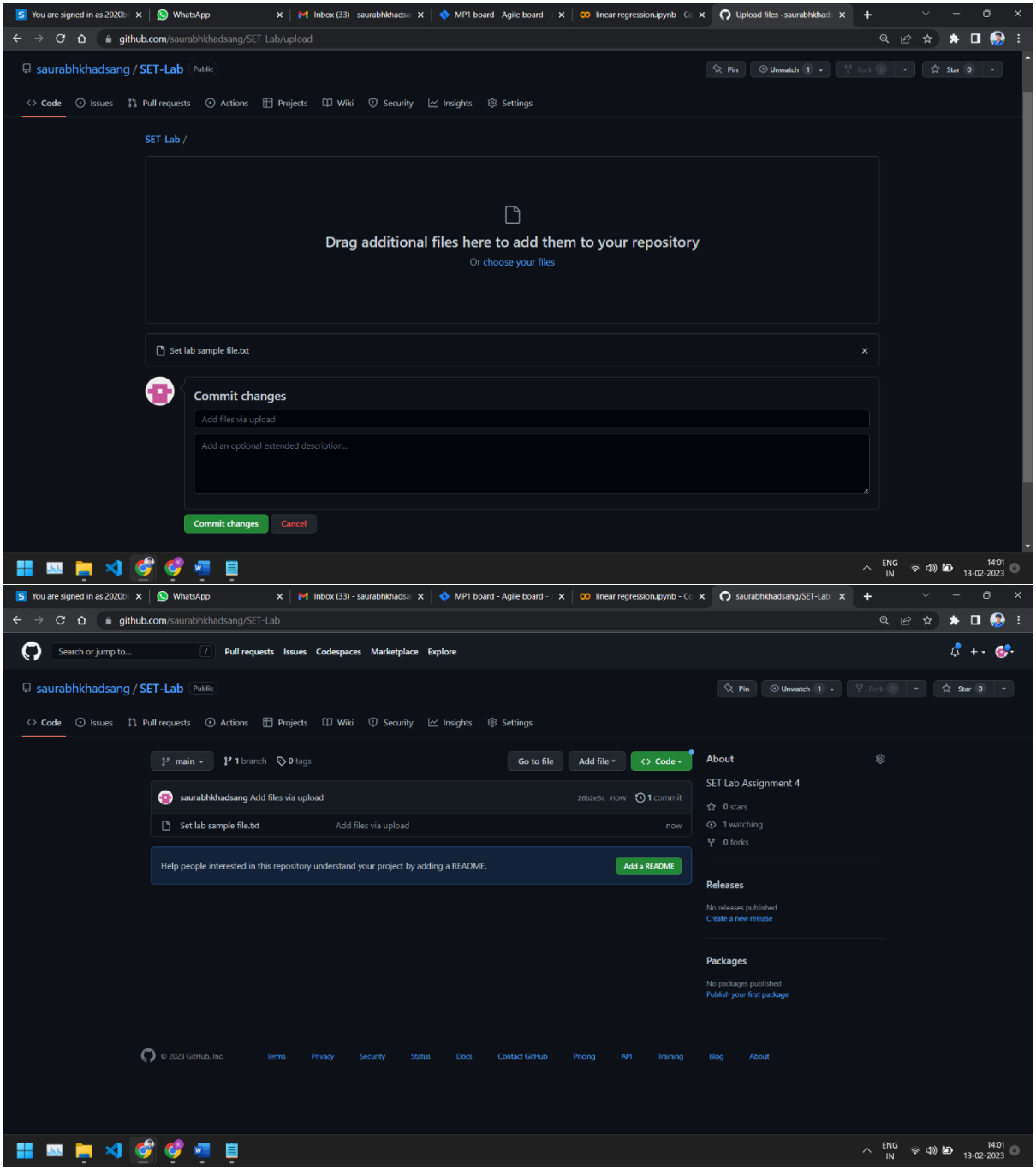
PRN : 2020BTECS00079

Name : Saurabh Khadsang

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)

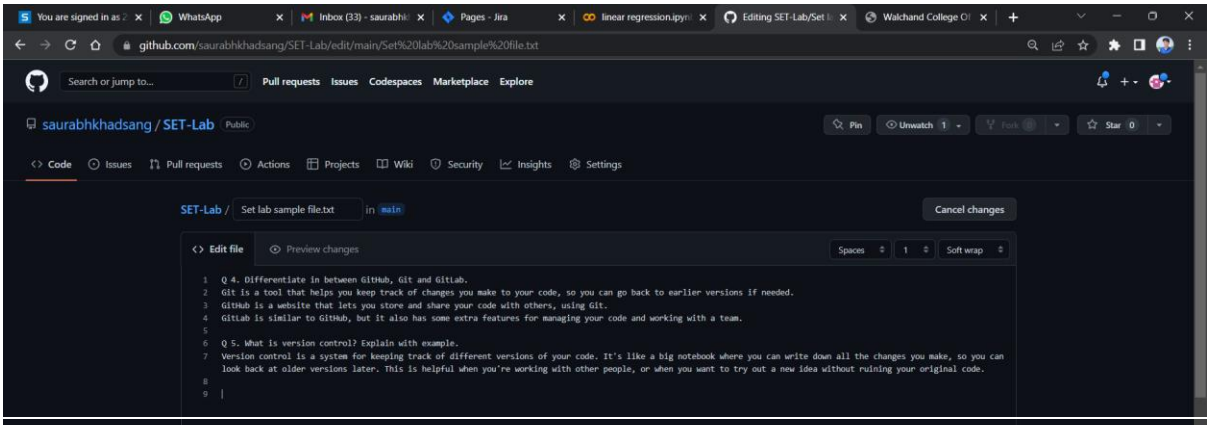


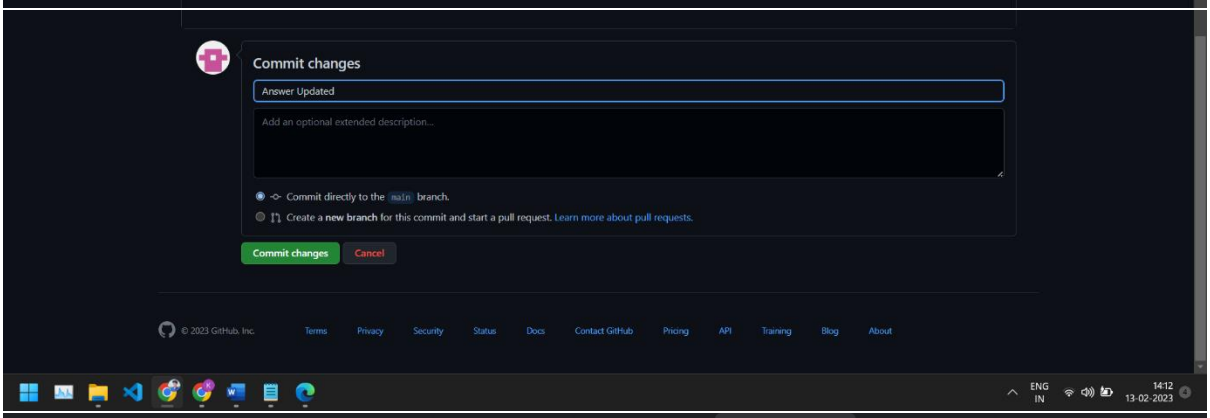
1. **Perform commit on added files**
   Go to your repository on GitHub.
   Select the files you want to commit.
   Click the "Commit changes" button.
   Write a commit message that describes the changes you made.
   Click the "Commit changes" button again.

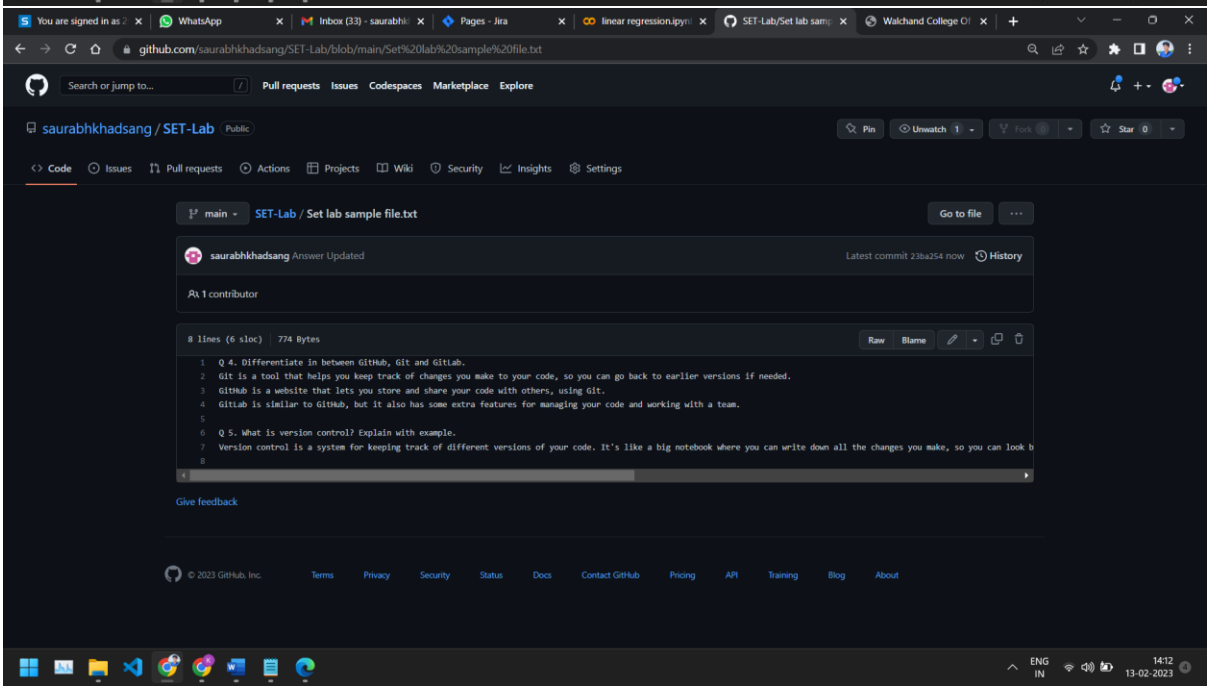


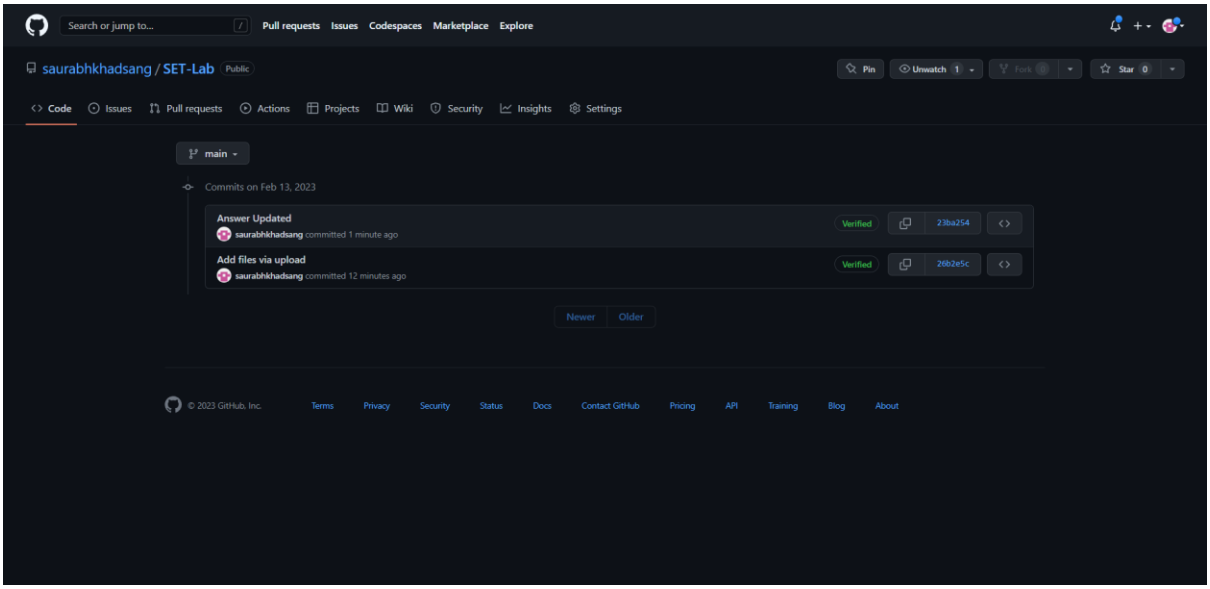
2. **Perform update to the existing files (show history)**

Go to your repository on GitHub.
Select the file you want to update.
Click the "Edit" button.
Make the desired changes.
Write a commit message that describes the changes you made.
Click the "Commit changes" button.
You can see the history of changes made to the file by clicking on "Commits" in the main repository page.


github.com/saurabhkhadsang/SET-Lab/edit/main/Set%20lab%20sample%20file.txt
saurabhkhadsang / SET-Lab
Public
SET-Lab / Set lab sample file.txt in main
Cancel changes
Edit file
Preview changes
1 Q 4. Differentiate in between GitHub, Git and GitLab.
2 Git is a tool that helps you keep track of changes you make to your code, so you can go back to earlier versions if needed.
3 GitHub is a website that lets you store and share your code with others, using Git.
4 GitLab is similar to GitHub, but it also has some extra features for managing your code and working with a team.
6 Q 5. What is version control? Explain with example.
7 Version control is a system for keeping track of different versions of your code. It's like a big notebook where you can write down all the changes you make, so you can look back at older versions later. This is helpful when you're working with other people, or when you want to try out a new idea without ruining your original code.

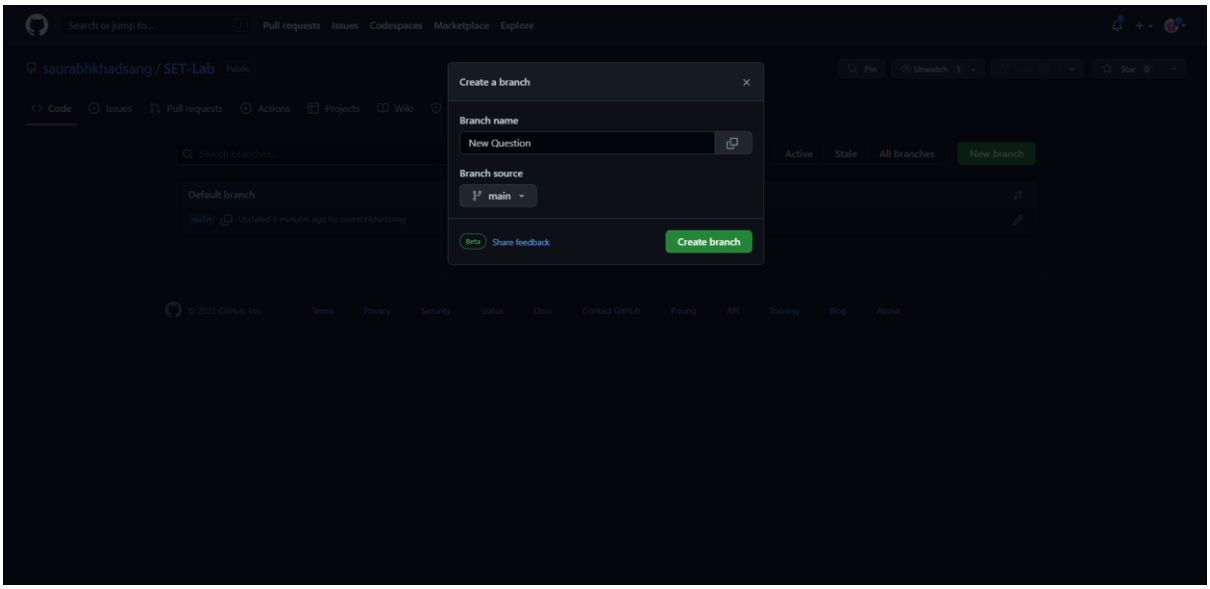

Commit changes
Answer Updated
Add an optional extended description...
Commit directly to the main branch.
Create a new branch for this commit and start a pull request. Learn more about pull requests.
Commit changes
Cancel
© 2023 GitHub, Inc.
Terms
Privacy
Security
Status
Docs
Contact GitHub
Pricing
API
Training
Blog
About

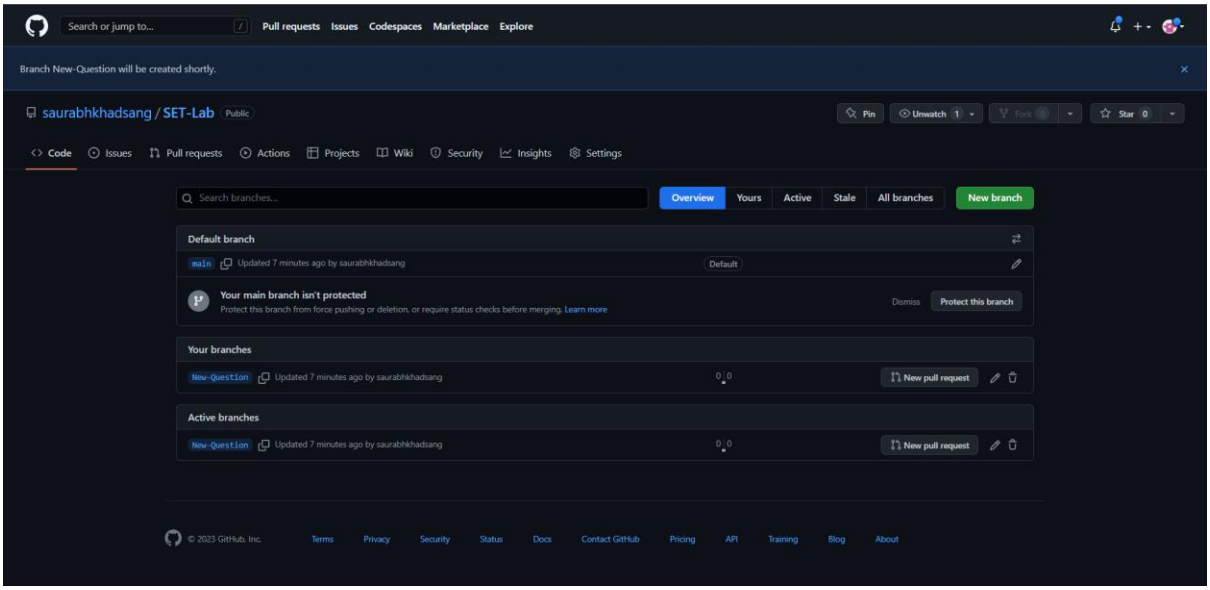

github.com/saurabhkhadsang/SET-Lab/blob/main/Set%20lab%20sample%20file.txt
saurabhkhadsang / SET-Lab
Public
main
SET-Lab / Set lab sample file.txt
Go to file
saurabhkhadsang Answer Updated
Latest commit 23ba254 now
History
1 contributor
8 lines (6 sloc)
774 Bytes
Raw
Blame
1 Q 4. Differentiate in between GitHub, Git and GitLab.
2 Git is a tool that helps you keep track of changes you make to your code, so you can go back to earlier versions if needed.
3 GitHub is a website that lets you store and share your code with others, using Git.
4 GitLab is similar to GitHub, but it also has some extra features for managing your code and working with a team.
6 Q 5. What is version control? Explain with example.
7 Version control is a system for keeping track of different versions of your code. It's like a big notebook where you can write down all the changes you make, so you can look b
Give feedback
© 2023 GitHub, Inc.
Terms
Privacy
Security
Status
Docs
Contact GitHub
Pricing
API
Training
Blog
About



3. Create another branch

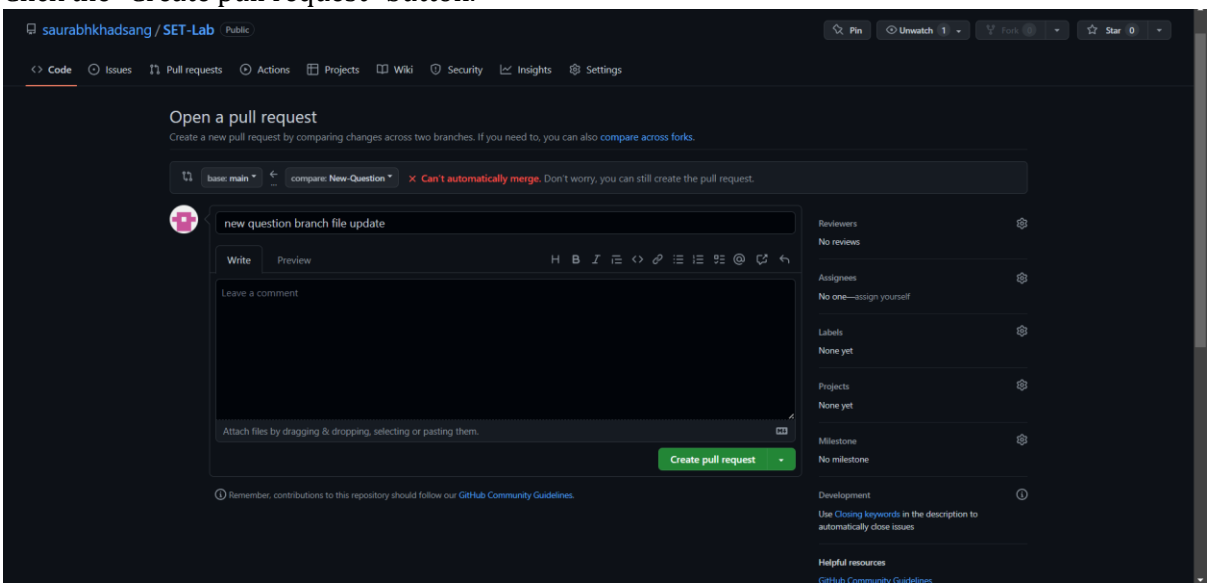
Go to your repository on GitHub.
Click on the "Branch" button.
Write the name for your new branch.
Click the "Create branch" button.
You can switch between branches by using the drop-down menu in the repository page.

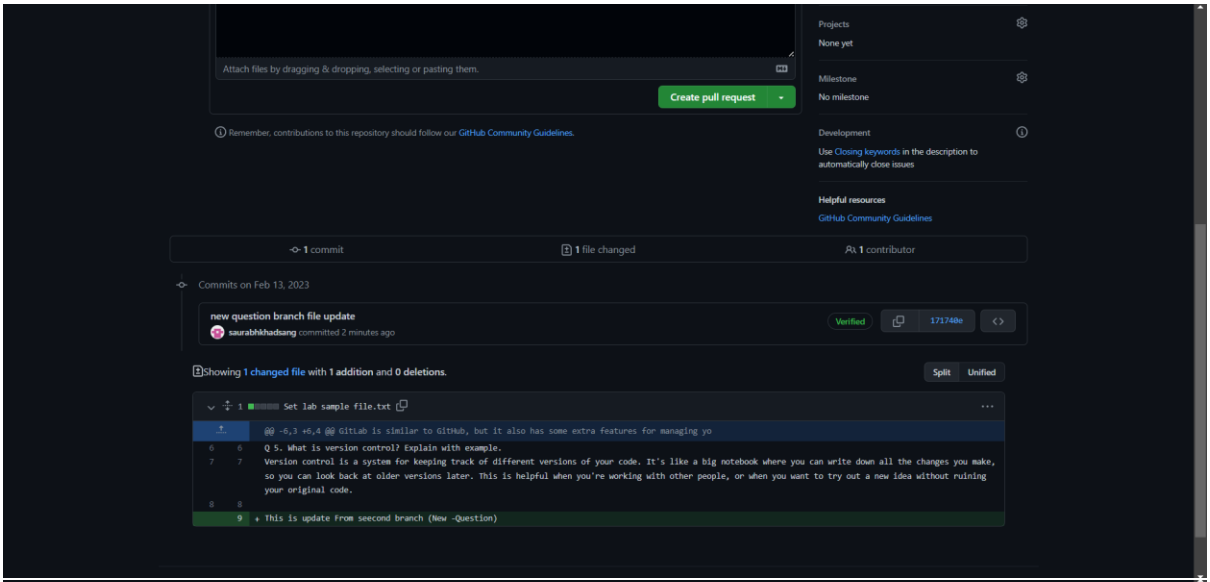


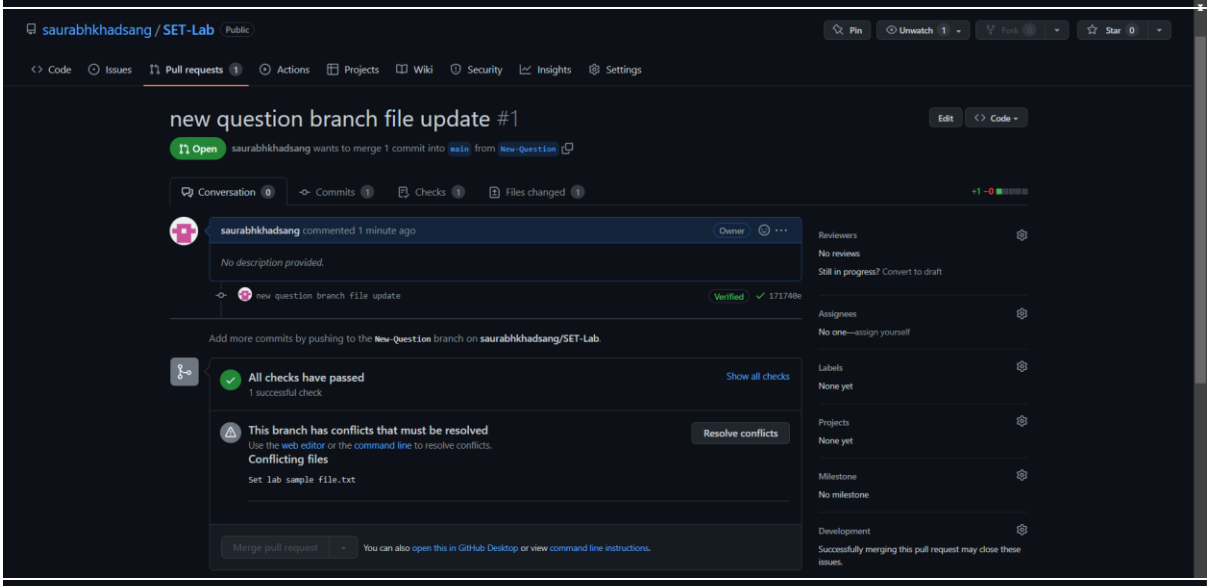


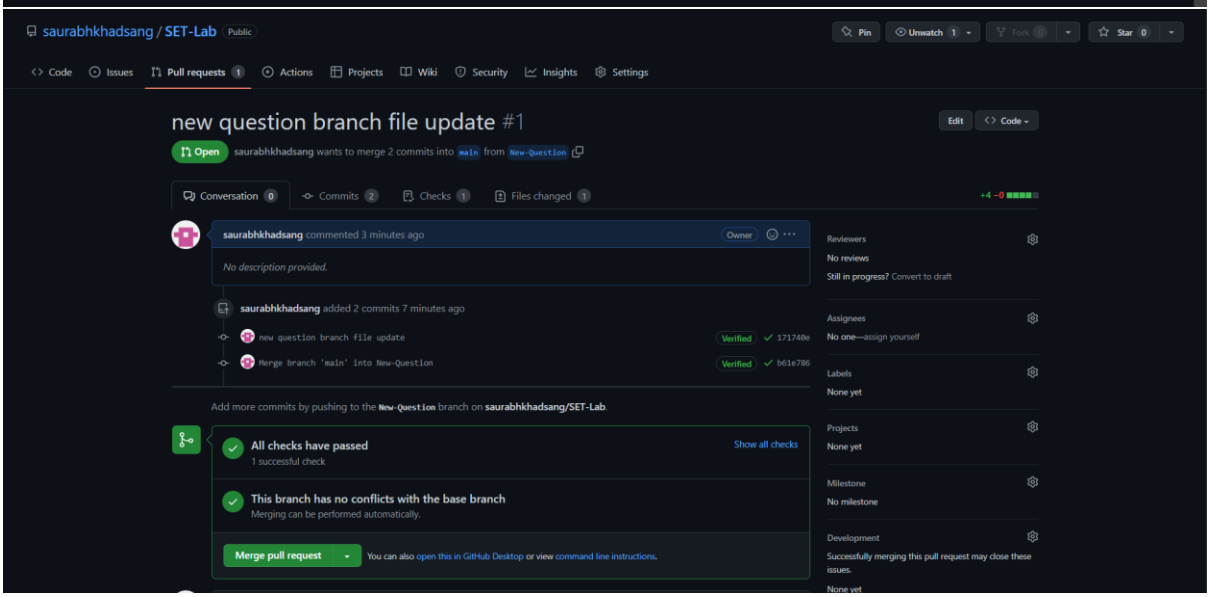
4. **Create pull request**

   Go to your repository on GitHub.
   Switch to the branch you want to merge into the main branch.
   Click the "New pull request" button.
   Select the base branch and the compare branch.
   Write a title and a description for your pull request.
   Click the "Create pull request" button.



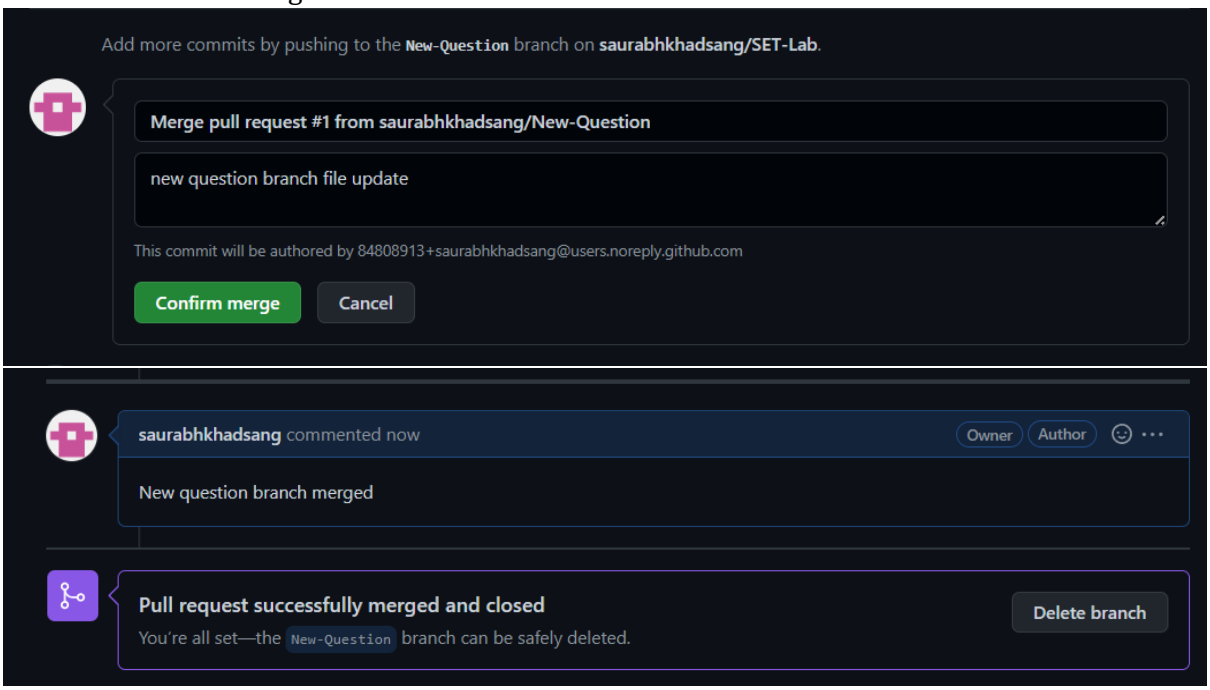






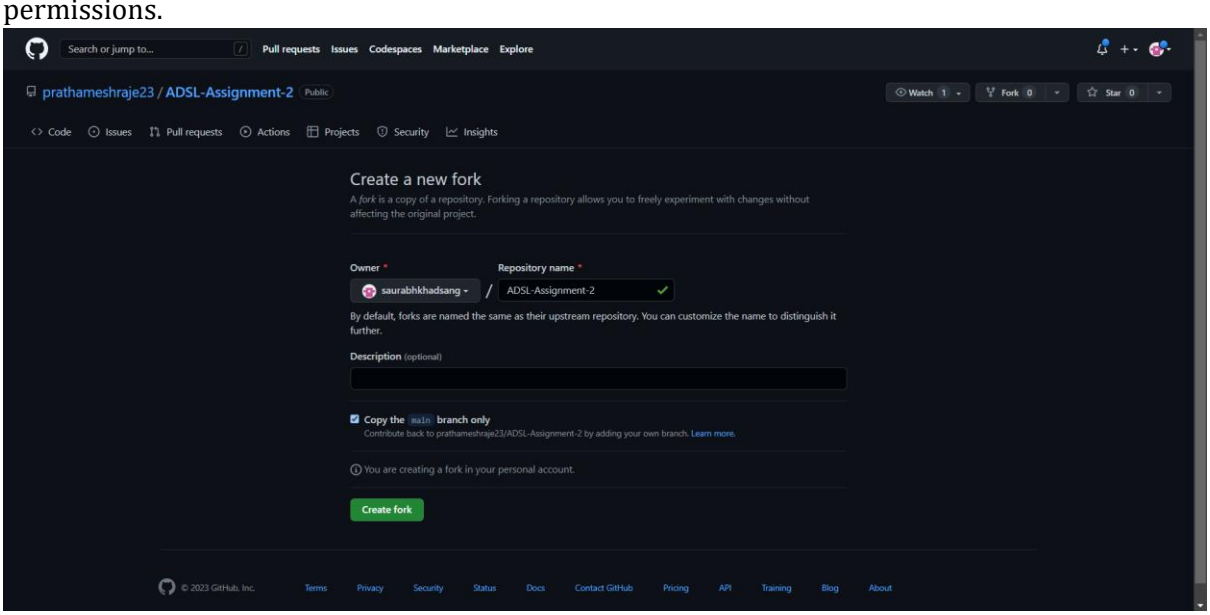
5. **Perform merging of both branches**
   Go to your repository on GitHub.

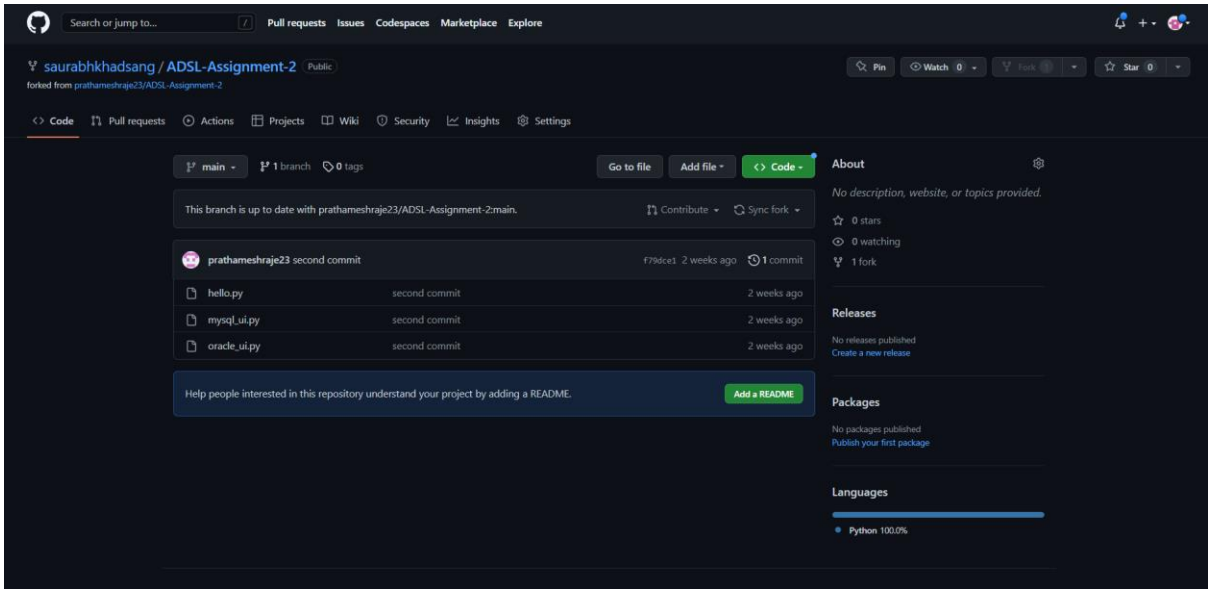
Open the pull request you want to merge.
Click the "Merge pull request" button.
Write a commit message that describes the merge.
Click the "Confirm merge" button



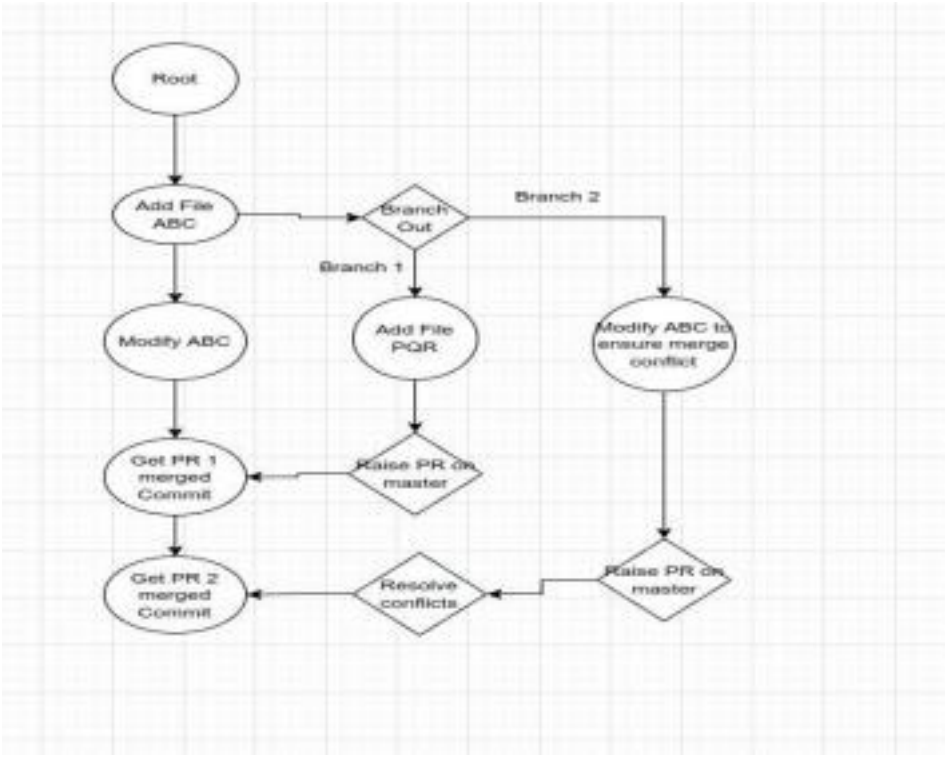
6. **Perform Fork operation**

Go to the repository you want to fork on GitHub.
Click the "Fork" button.
Choose your GitHub account to fork the repository to.
The repository will be copied to your GitHub account.
You can make changes and submit pull requests to the original repository if you have permissions.

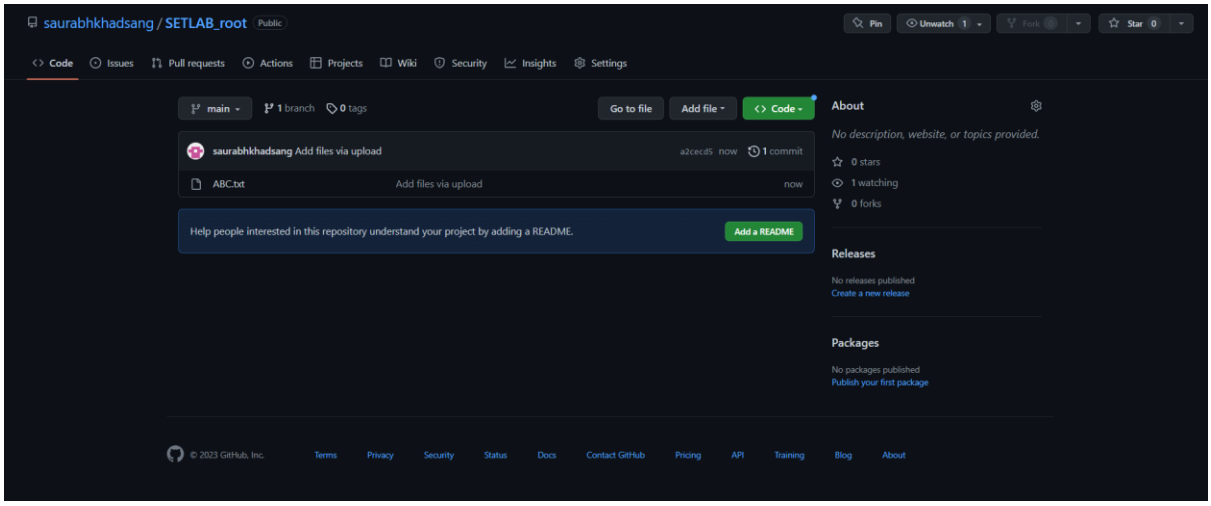




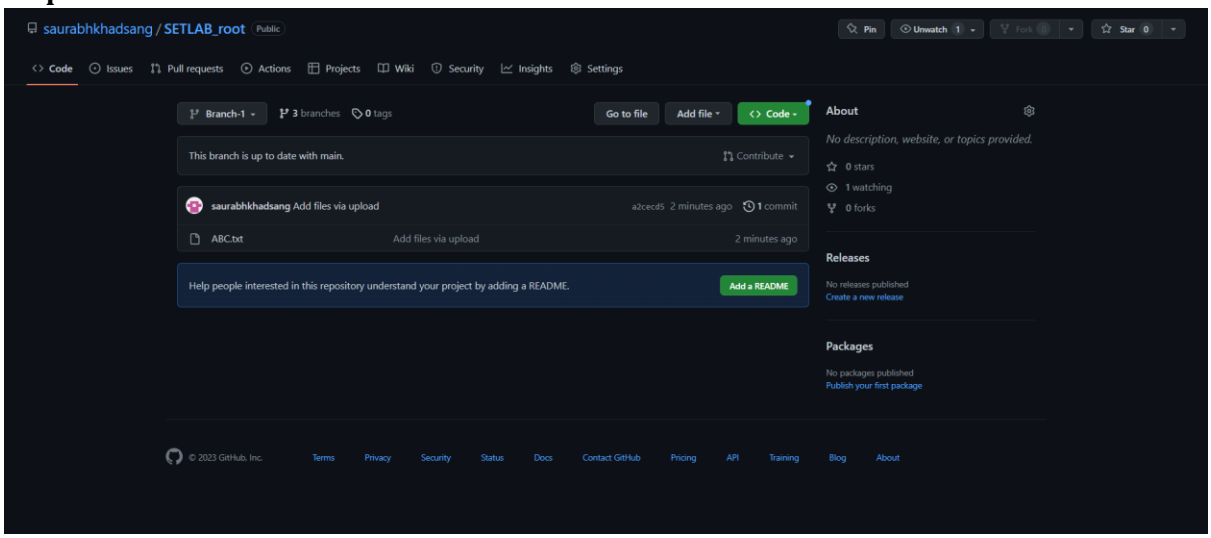
Q 2. For the diagram given below create a GitHub repository and perform operations given in the diagram. (Perform commit operations as given)(Add screenshots as an answer to this question)



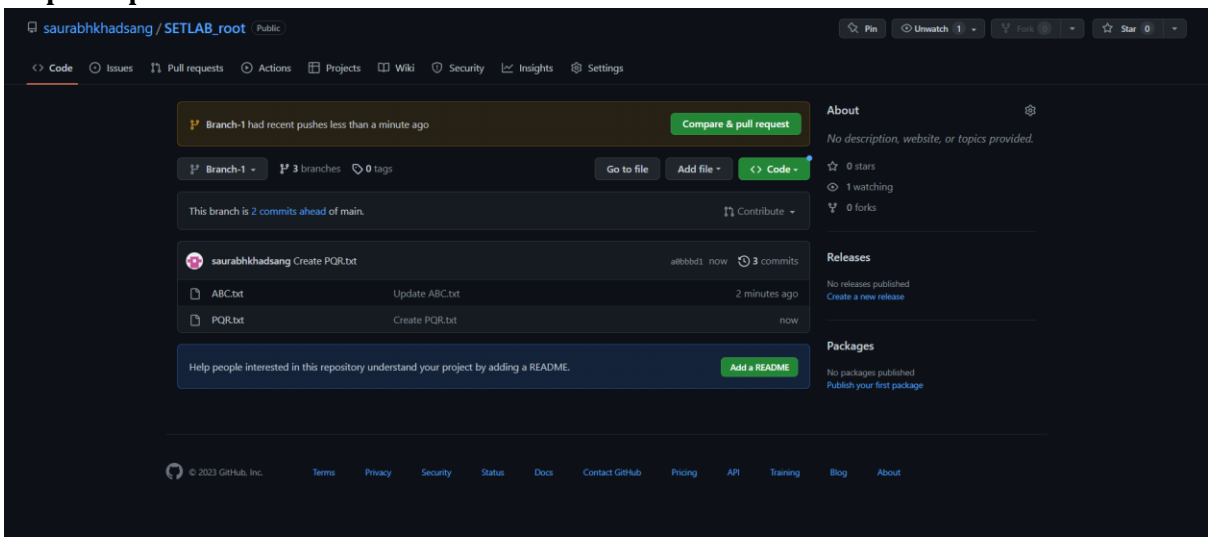
Step 1:
Created New Repo and added a file named ABC.txt



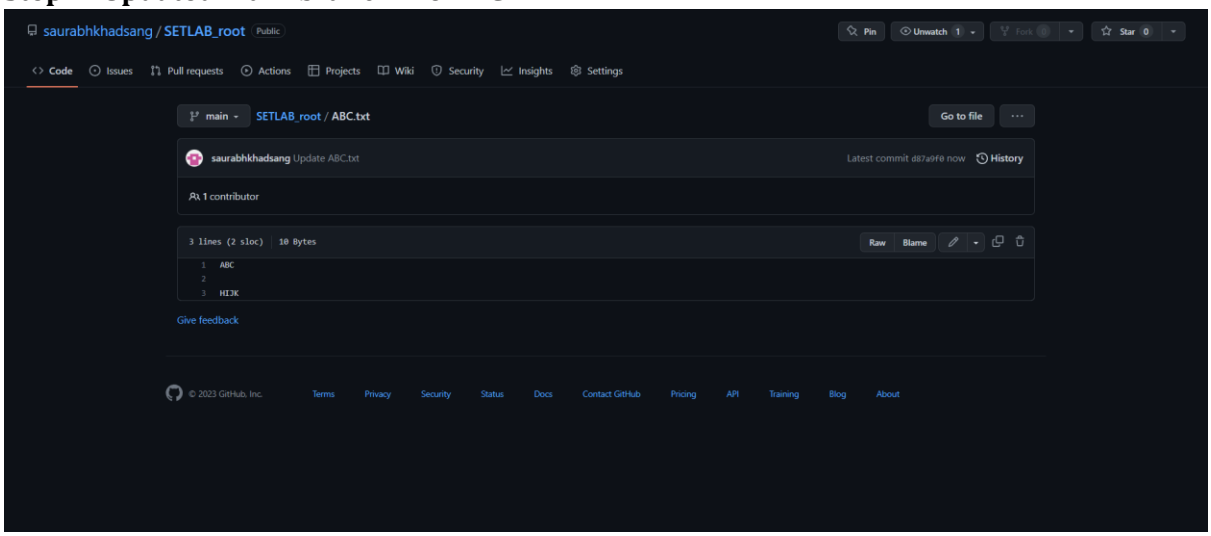
Step 2: Created New Branch 1



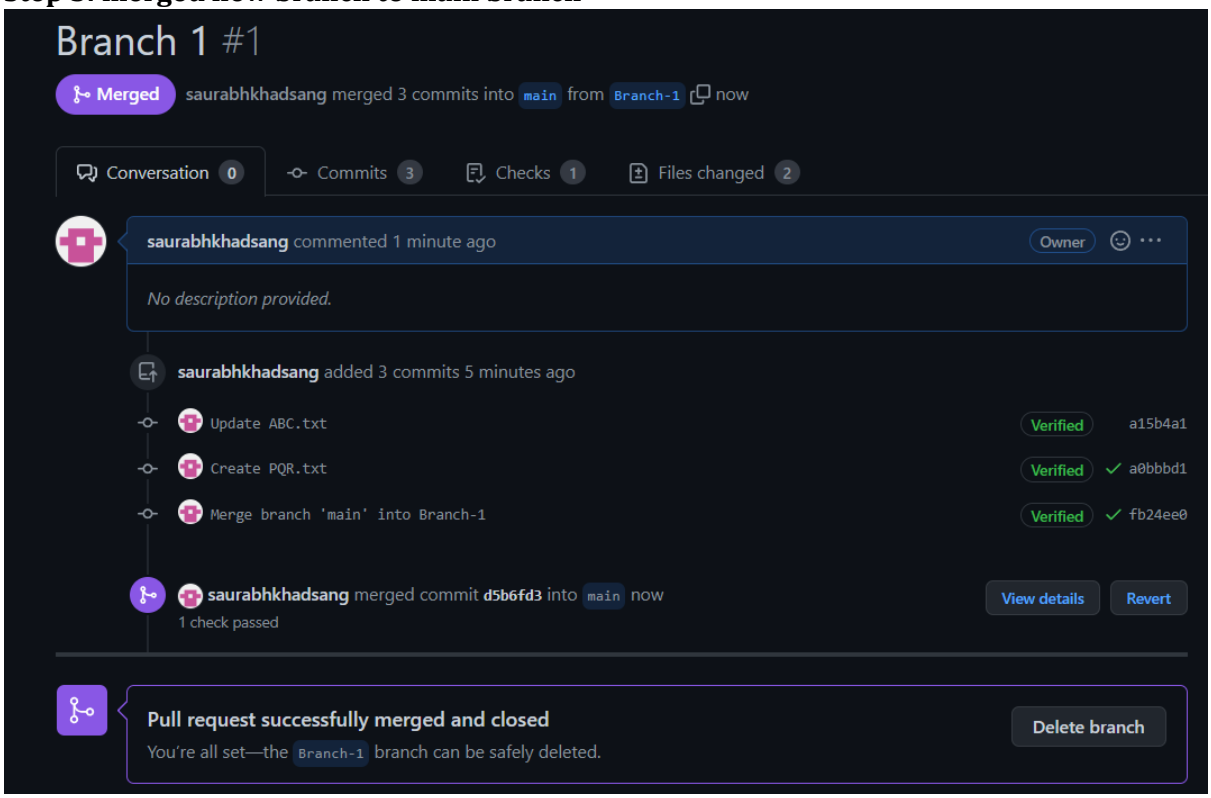
Step3 : Updated file in new branch



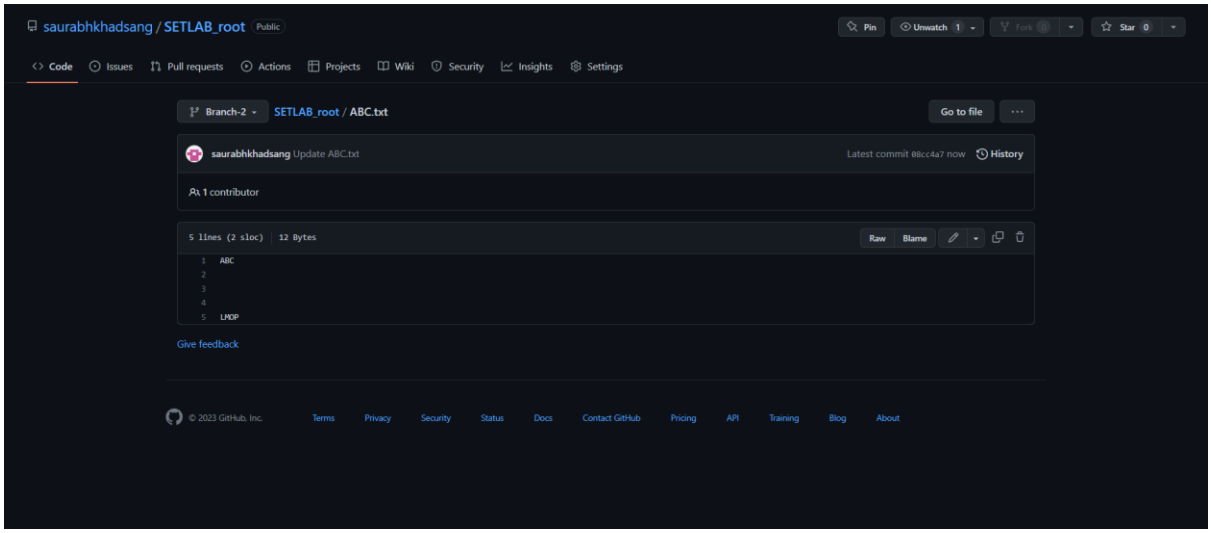
Step 4: Updated main branch file ABC



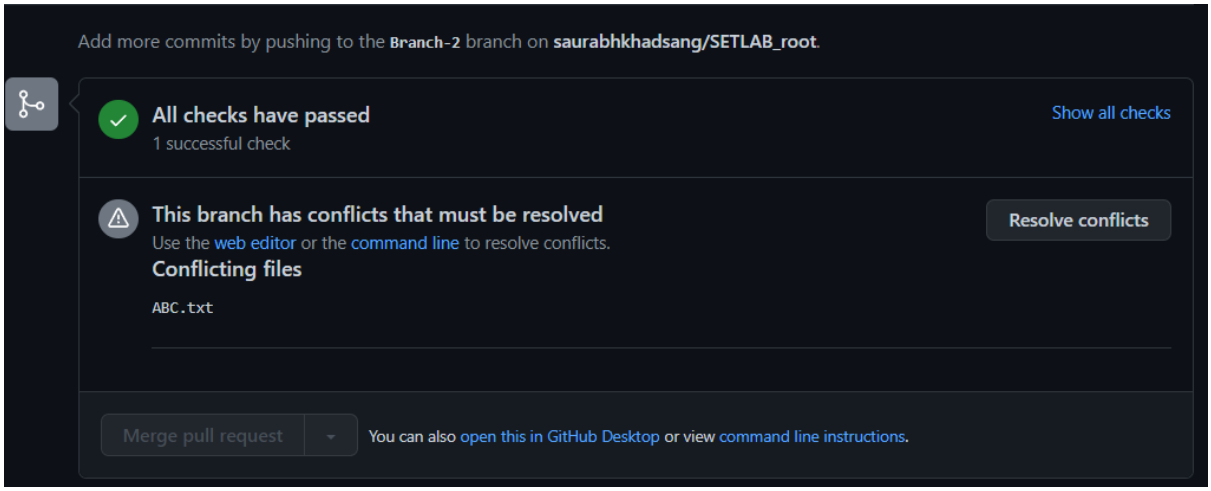
Step 5: merged new branch to main branch



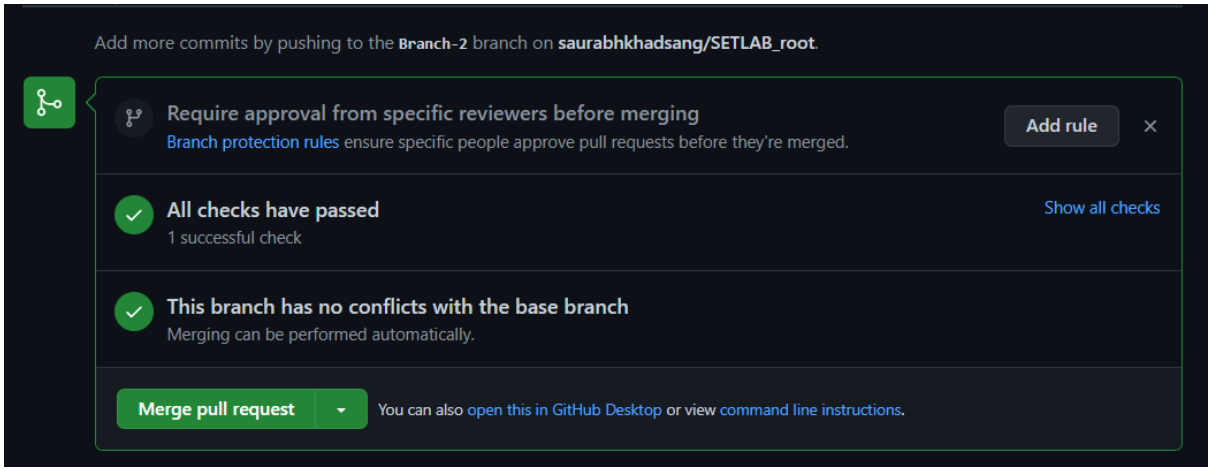
Step 6: Edited file in new branch2 to ensure merge conflict:



conflicts found:



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).

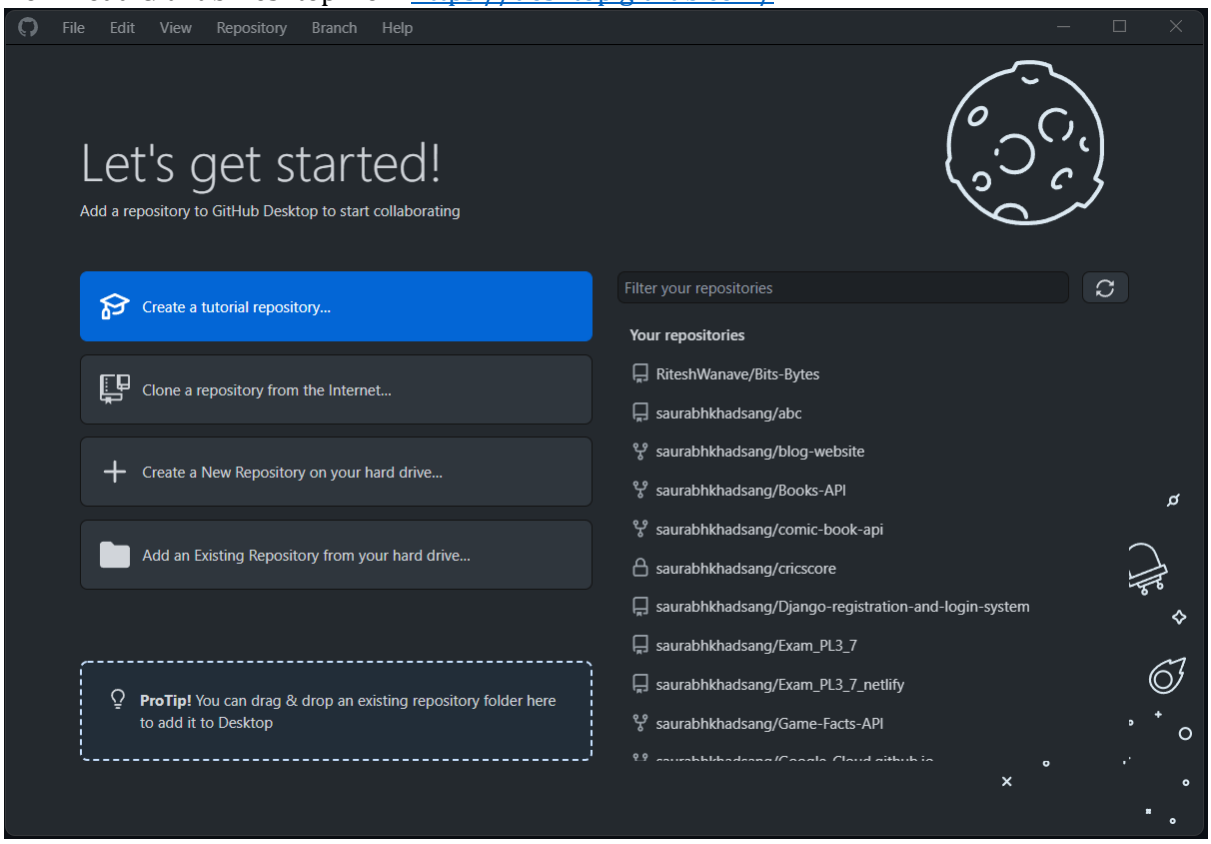
GitHub Desktop is a graphical user interface (GUI) application for working with GitHub repositories. It is available for Windows and macOS and can be downloaded for free from the GitHub website.

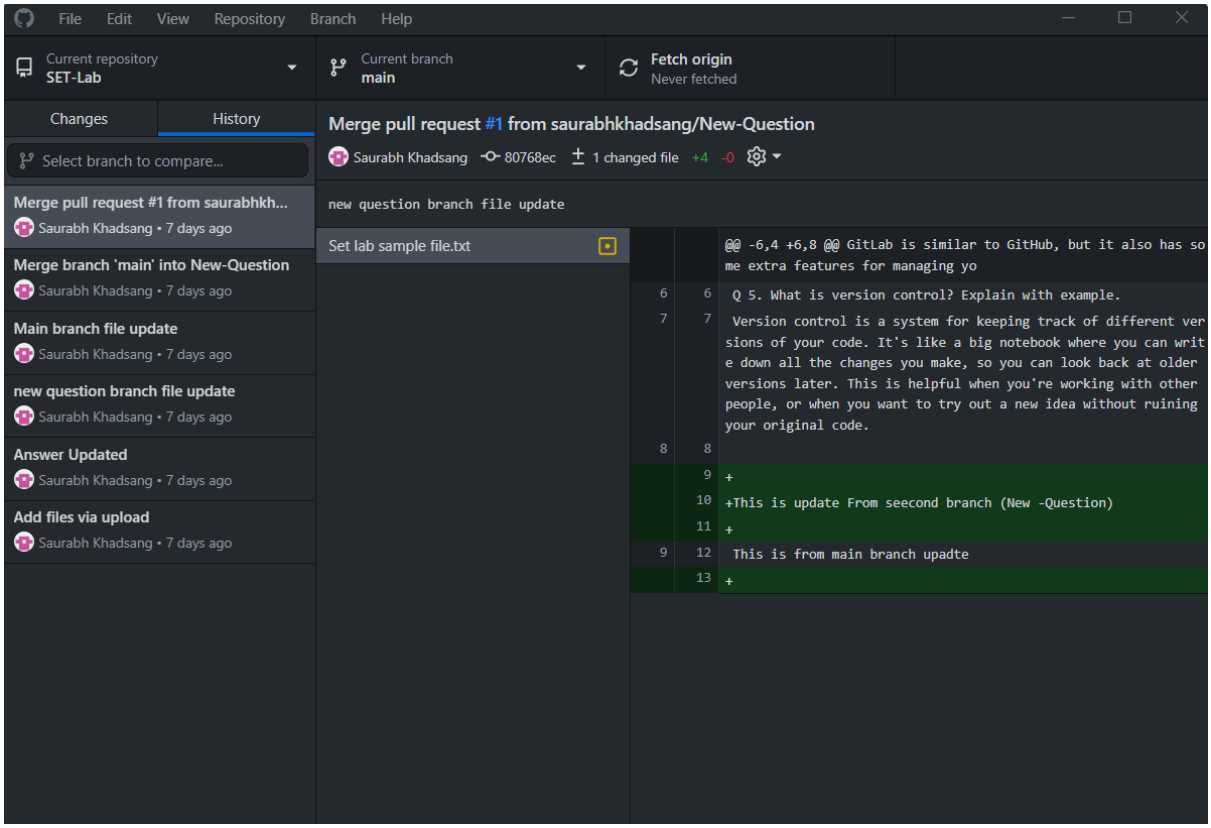
GitHub Desktop provides a user-friendly interface for working with repositories hosted on GitHub, allowing users to perform common version control operations such as committing changes, branching and merging code, and viewing commit history. It also provides an integrated diff viewer, allowing users to easily compare different versions of code.

One of the advantages of GitHub Desktop is that it simplifies the process of working with Git repositories by providing a user-friendly interface. This can be especially helpful for developers who are new to Git and version control, as it allows them to focus on the code rather than the command-line interface.

GitHub Desktop also provides a seamless integration with the GitHub platform, allowing users to quickly create and manage repositories, and easily collaborate with other developers. The application also provides a notification system that keeps users informed of changes and updates to their repositories.

Download GitHub Desktop from https://desktop.github.com/





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

Git: Git is an open-source version control system that allows developers to keep track of the changes made to their code over time. Git is primarily a command-line tool and can be used to manage code repositories locally.

GitHub: GitHub is a web-based platform that uses Git for version control. It provides a cloud-based repository for storing and managing code, which makes it easy for developers to collaborate on a project. In addition to its Git-based version control features, GitHub also provides a variety of other features such as bug tracking, wikis, and project management tools.

GitLab: GitLab is similar to GitHub in that it provides a cloud-based repository for storing and managing code using Git-based version control. However, GitLab offers additional features such as continuous integration and continuous deployment (CI/CD) pipelines, code review, and issue tracking. GitLab can be self-hosted, giving organizations full control over their data and infrastructure.

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

Version control is a system that helps software developers to manage changes to their code over time. It allows them to keep track of every change made to the code, who made it, and when it was made. This helps developers to collaborate on code, work on different features at the same time, and easily revert changes if necessary.

Here's an example of how version control works:

Suppose a team of developers is working on a software project that involves creating a web application. They use version control to manage their code. They create a repository using a version control tool such as Git, which is stored on a server. Each developer has a copy of the repository on their local machine.

The developers can make changes to the code, and then commit those changes to the repository. Each commit includes a message that explains what changes were made. The version control tool keeps track of every commit, so developers can easily see the history of the code changes.

Suppose a developer makes a mistake in the code and introduces a bug. With version control, it's easy to revert to a previous version of the code before the mistake was made. Developers can also work on different features at the same time without interfering with each other's work. The version control tool allows them to merge changes made by different developers, so they can all work on the same codebase.

In summary, version control is a system that helps developers manage changes to their code over time. It allows them to keep track of changes, collaborate on code, and easily revert to previous versions if necessary.