PRN:2020BTECS00090

Name:Vishal Chauhan

Software Engineering Tools Lab

Assignment No-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

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

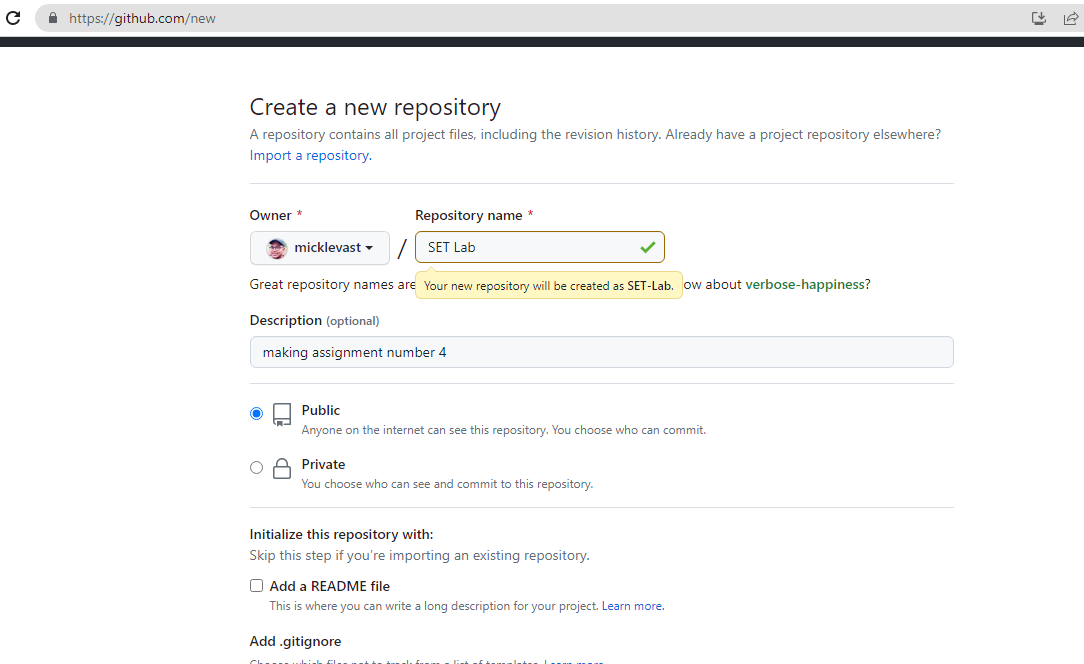
3. Create another branch

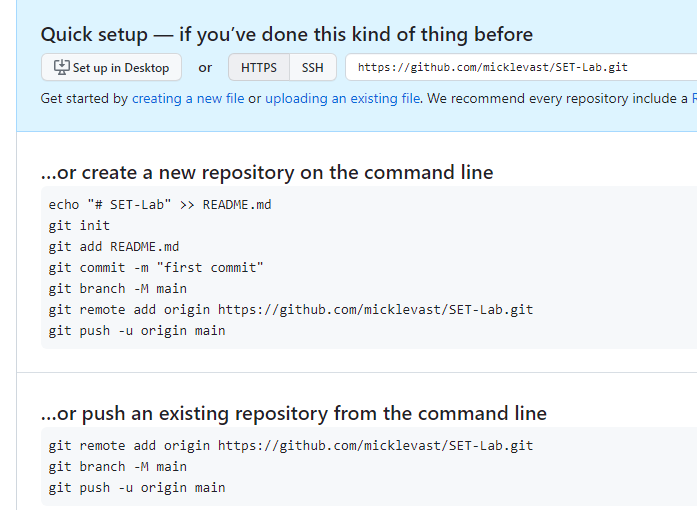
4. Create pull request

5. Perform merging of both branches

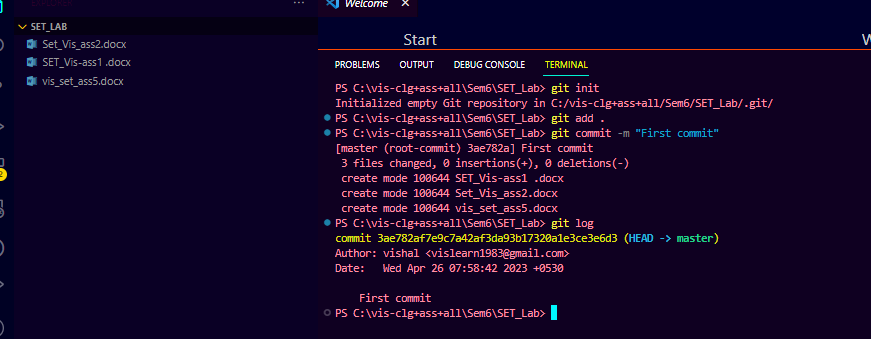
6. Perform Fork operation

Step1: Making new repo

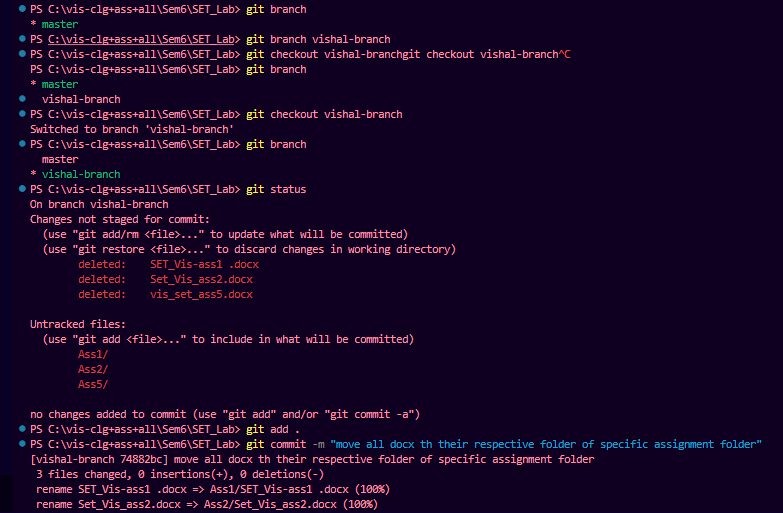




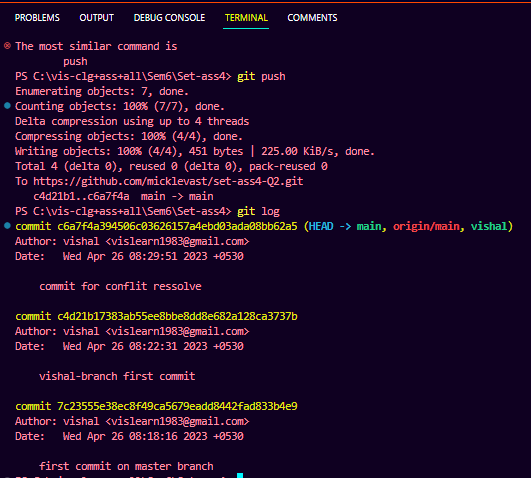
Step2 : Initializing new git repo and adding file and making first commit and showing history

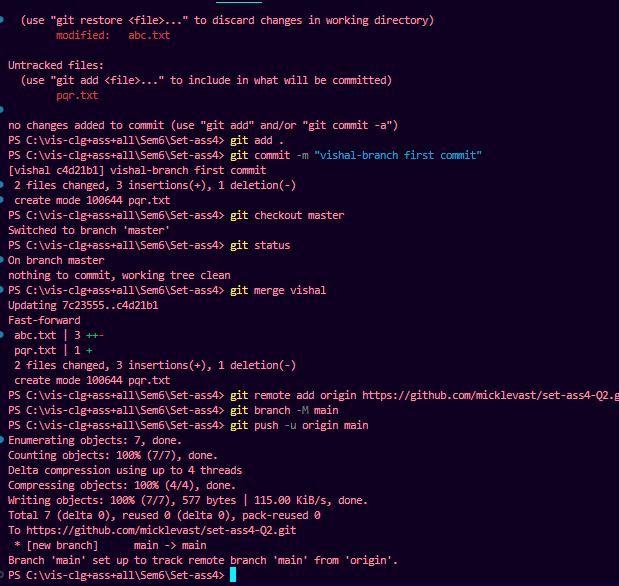


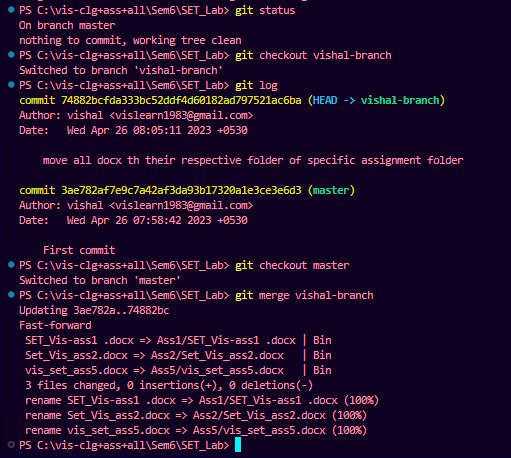
Step3:Creating another branch and making commit over after changes



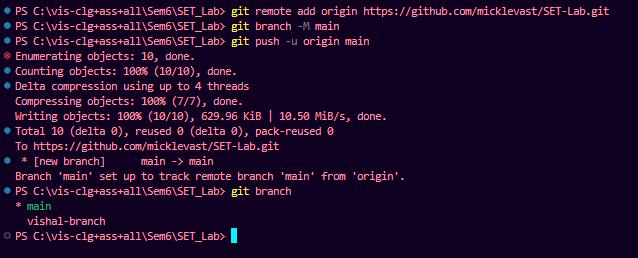
Step4:Merging two branch and pushing it into github remote repo





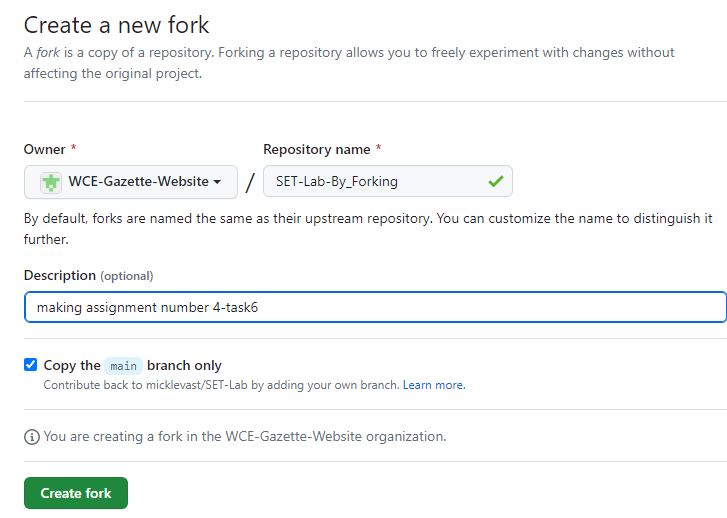


Step5:Pushing local repo over github i.e. on remote repo

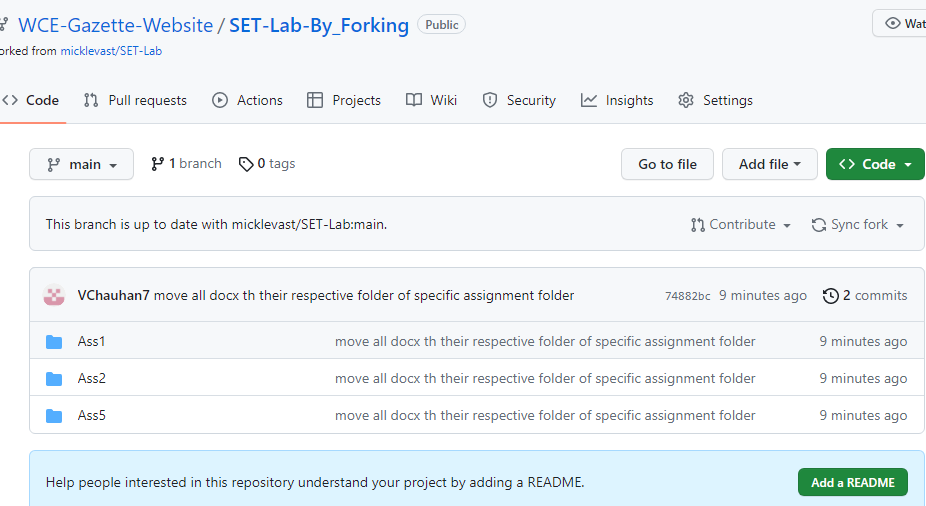


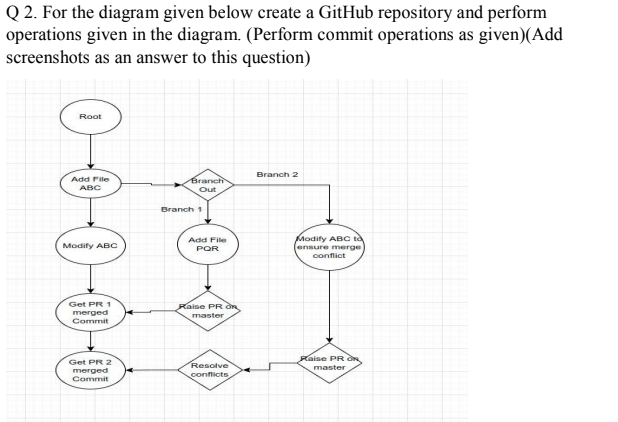
Step6:Performing Fork

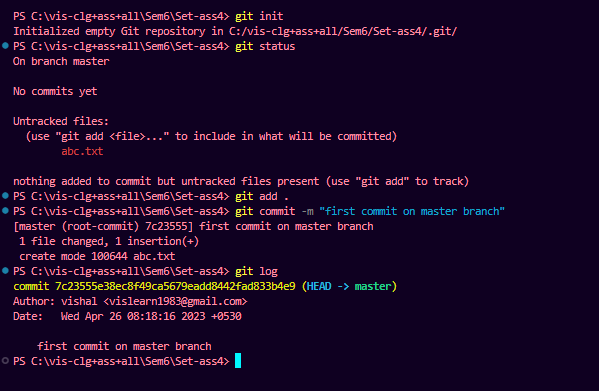
git fork: This command creates a copy of a remote repository on the user's GitHub account.



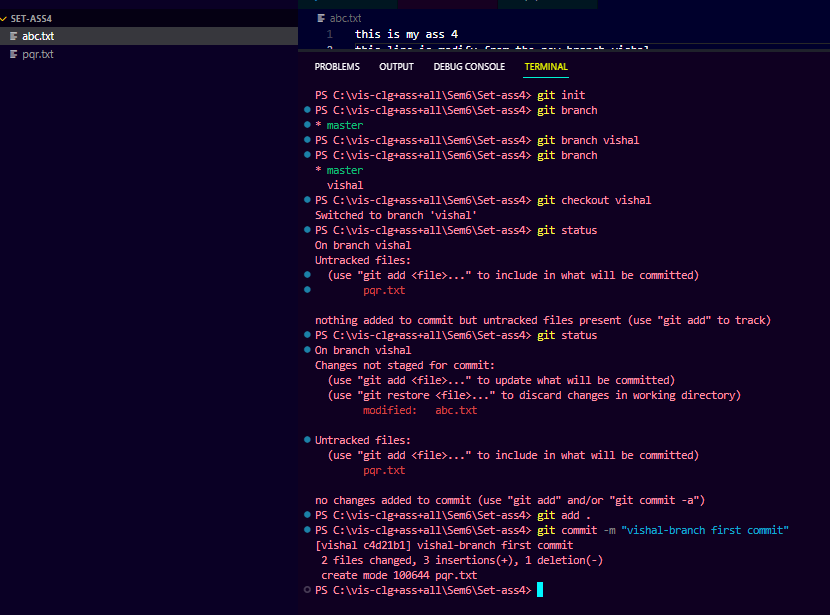
Fork result:







Step:making new branch and adding new file pqr.txt and adding and committing in it



Q 3. What is GitHub desktop? How to install GitHub on local machine? Install

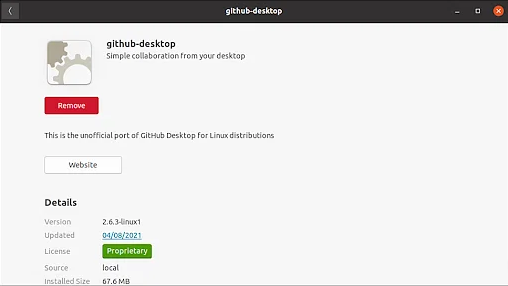
GitHub on your local machine and access repository created in question no 1 (add

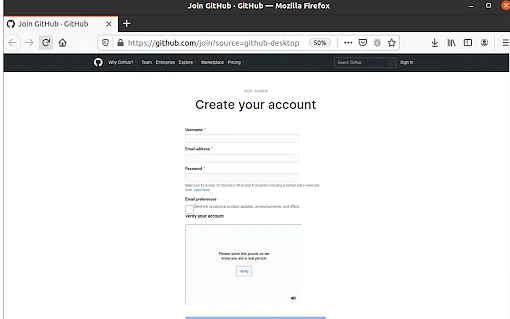
screenshots).

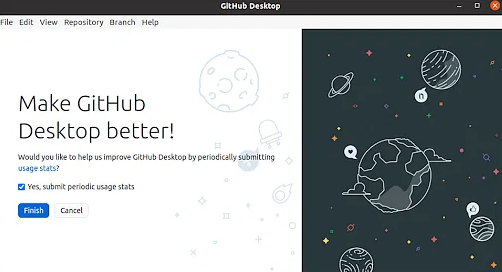
GitHub Desktop is a graphical user interface (GUI) for GitHub, a popular web-based Git repository hosting service. It allows users to manage their Git repositories, perform version control tasks, and collaborate with others using a simple and intuitive interface.

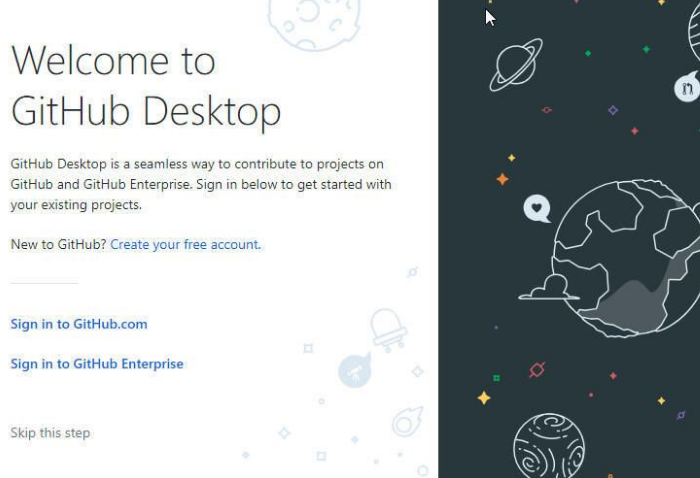
To install GitHub Desktop on your local machine, you can follow these steps:

1. Go to the GitHub Desktop website at<https://desktop.github.com/> and click on the "Download for" button for your operating system.
2. Once the installer is downloaded, run it to start the installation process. Follow the prompts in the installer to complete the installation.
3. After the installation is complete, launch GitHub Desktop and sign in to your GitHub account if prompted.
4. Click on the "Clone repository" button on the GitHub Desktop home screen.
5. Select the "URL" tab and enter the URL for the repository you want to clone. For example, if the repository URL is https://github.com/username/repository-name.git, enter this URL into the "URL" field.

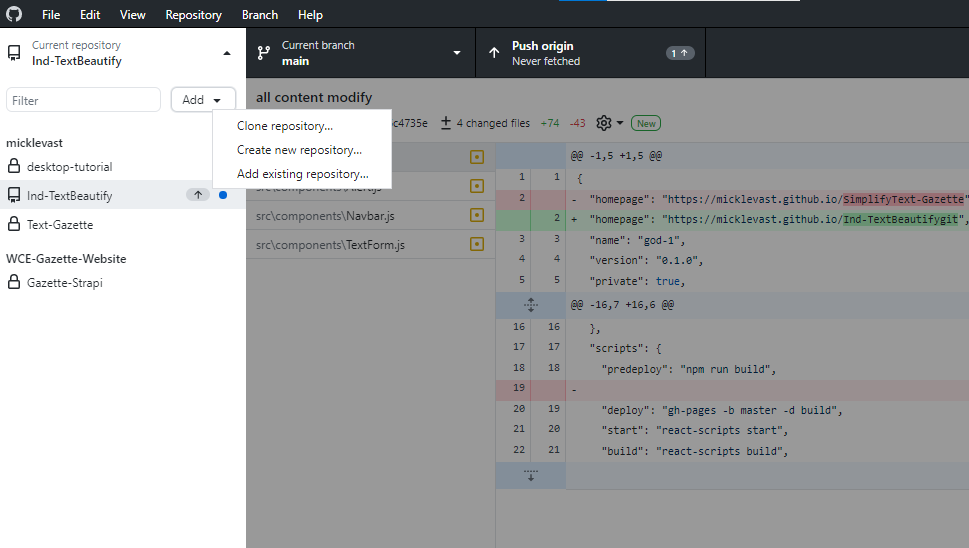


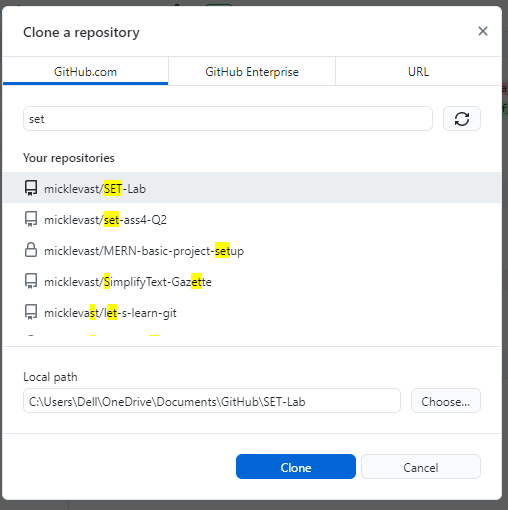






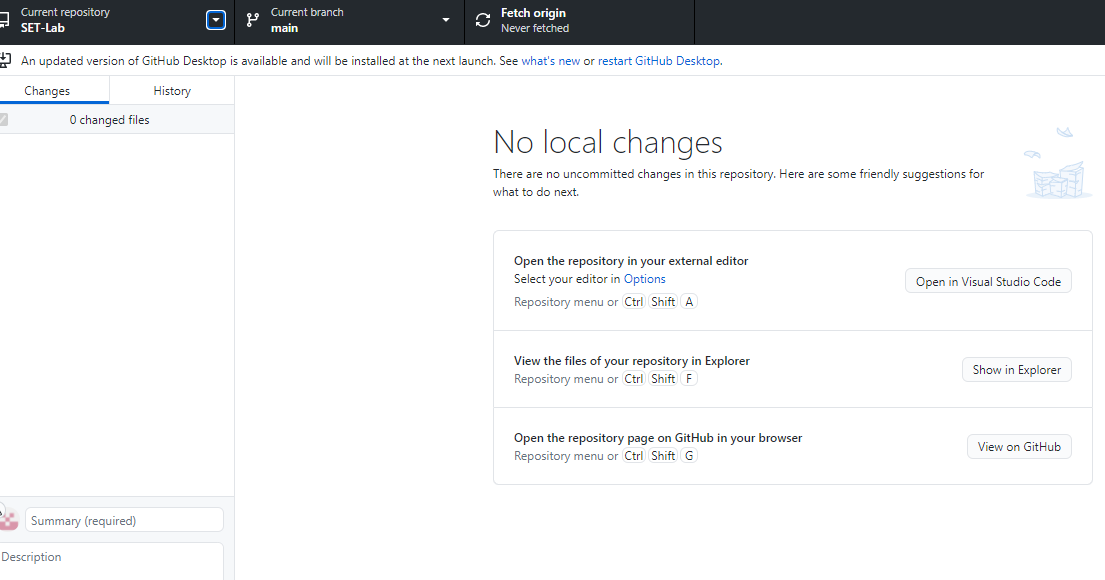
1. Choose a local path where you want to clone the repository, and click on the "Clone" button.





1. Once the repository is cloned, you can open it in your preferred code editor or IDE and start making changes.

⇒Accessing Previous repository using github dextop:



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

Git is a distributed version control system used to manage source code and track changes over time. It allows developers to collaborate on a project and keep track of different versions of their code. Git is primarily used through a command-line interface, but there are also various graphical user interfaces available.

GitHub is a web-based platform that provides hosting for Git repositories. It allows developers to collaborate on projects, share code, and track issues and bugs. GitHub provides additional features such as project management tools, pull requests, code reviews, and access control.

GitLab is also a web-based platform that provides hosting for Git repositories. Like GitHub, it offers project management tools, issue tracking, and access control. In addition, GitLab offers continuous integration and deployment features, which enable developers to automatically build, test, and deploy their code changes. GitLab can also be self-hosted, which provides additional control and customization options.

Q5. what is version control?

Version control is the management of changes to documents, code, or other types of files over time. It enables developers to track modifications made to a file or a set of files, revert to earlier versions, and collaborate with others on the same files.

Version control systems, such as Git, provide a way to organize and keep track of changes to code or other files. Instead of just having a single file with the latest version, version control systems maintain a history of changes made to the file, including who made the changes and when. This history can be used to understand the evolution of the code or file over time and to identify and fix issues or bugs.

Version control also enables collaboration between multiple developers on the same codebase. Developers can work on separate branches of the code, make changes, and merge their changes back into the main codebase. This helps to avoid conflicts and allows for parallel development of different features.

Overall, version control is a critical tool for software development and other types of collaborative projects, as it enables effective management and tracking of changes over time.