**PURBANCHAL UNIVERSITY**

****

**KHWOPA ENGINEERING COLLEGE**

**LIBALI-08, BHAKTAPUR**



LAB REPORT ON .NET

LAB NO. 01

**SUBMITTED BY: SUBMITTED TO:**

Name: Rashik Prajapati Department of

Roll No. : 770330 Computer Engineering

Submission: 2081/12/09

**LAB-1**

**Git and Github**

**Theory**

**Git**  [Git](https://www.simplilearn.com/tutorials/git-tutorial/git-tutorial-for-beginner) is distributed version control system used for source code management. It is a free and open-source version control system used to handle small to very large projects efficiently. Git is used to track changes in the source code, enabling multiple developers to work together on non-linear development. Linus Torvalds created Git in 2005 for the development of the [Linux](https://www.simplilearn.com/linux-programming-for-beginners-article) kernel.

**Working with Git**

* Initialize Git on a folder, making it a **Repository**
* Git now creates a hidden folder to keep track of changes in that folder
* When a file is changed, added or deleted, it is considered **modified**
* You select the modified files you want to **Stage**
* The **Staged** files are **Committed**, which prompts Git to store a **permanent** snapshot of the files
* Git allows you to see the full history of every commit.
* You can revert back to any previous commit.
* Git does not store a separate copy of every file in every commit, but keeps track of changes made in each commit!

**GitHub**

GitHub is a web-based platform that provides version control and collaboration tools for developers, enabling them to manage, store, and track changes in their code efficiently. Built on Git, an open-source version control system created by Linus Torvalds, GitHub allows developers to work on projects individually or as a team. It offers features such as repositories for storing code, branches for parallel development, pull requests for code reviews, and issue tracking for project management.

**Forking & Cloning**

* **Forking** creates a personal copy of another user’s repository.
* **Cloning** downloads a repository to a local computer for offline development.

**Common Git and Github Commands**

**Git Configuration**

*git config --global user.name “Your Name”*

This command sets the global username for the Git commits.

*git config --global user.email “your\_email@example.com”*

This command sets the global email associated with Git commits.

**Git Initializing**

*git init*

This command initializes a new git repository in the current directory.

**Git Linking**

*git remote add origin* <repo>

This command links the new repository of github with the local codes.

**Git Staging and Commits**

*git add .*

This command adds files to the staging area.

*git commit -m “message”*

This command commits and saves changes of stage area with a message.

**Git Status and Log**

*git status*

This command checks the status of the working directory.

*git log*

This command view commit history.

**Git Branching and Merging**

*git branch*

This command lists all the branches that exist in the repository.

*git branch <branch\_name>*

This command creates a new branch for separate development.

*git checkout <branch>*

This command switches to another branch.

*git switch <branch\_name>*

This command switches to another branch.

*git merge <branch\_name>*

This command merges a specified branch into the current branch.

**Git Push and Pull**

*git push -u origin <branch\_name>*

This command uploads commits to a remote repository

*git pull origin*

This command fetch and merge changes from a remote repository

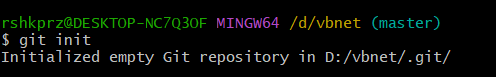
**Git Clone**

*git clone <repo\_url>*

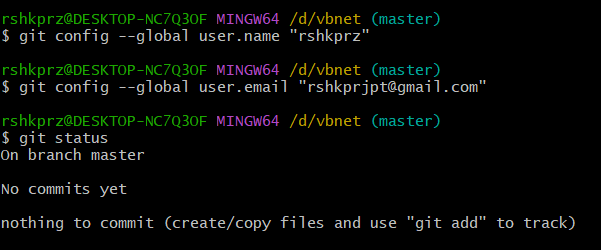
This command copy(clone) an existing repository.

**Lab Work**

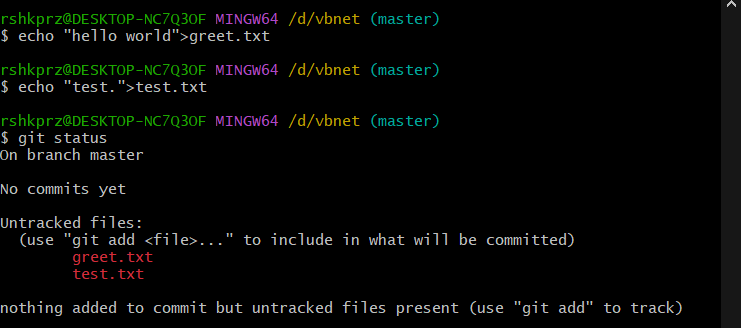
Initially, a folder is created tocreate, change the files using the version control git with different commands and git is initialized.



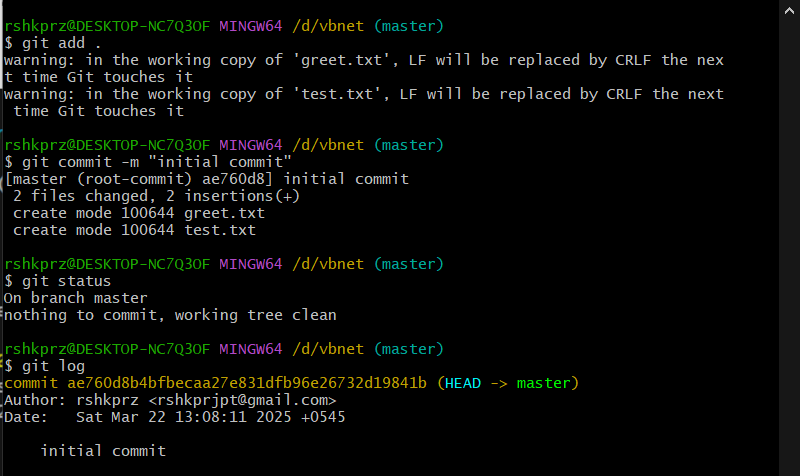
We set the global username and email of GitHub. As no any files are created there is nothing to commit.



Then we create two empty files in master branch and with the help of echo we insert text on those files and checked the status , it is in untracked stage and we sent the files to the staging stage.

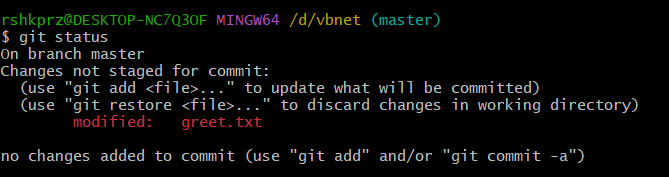


The files are then added for staging and commit the files with the message such that the files are stored in the local repository. Then we checked git status, there was nothing left to commit.



We again made certain changes in file text.txt to see certain changes in the file status.

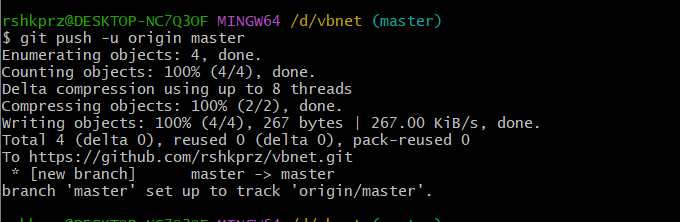
And again commit it so that all of the files are saved in the local repository.



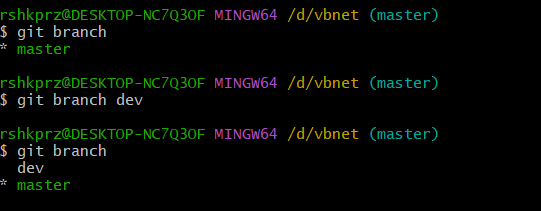
We then add the files in the remote repository by creating the repository in the GitHub and copying the url of the repo and using the above code.



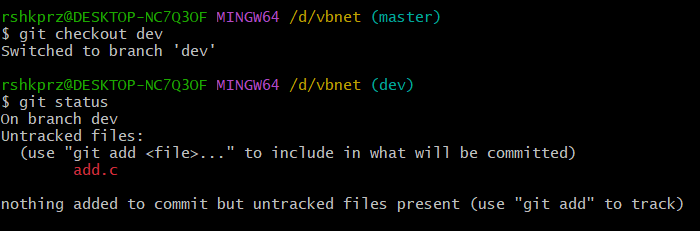
After that we push the files to the created repository.



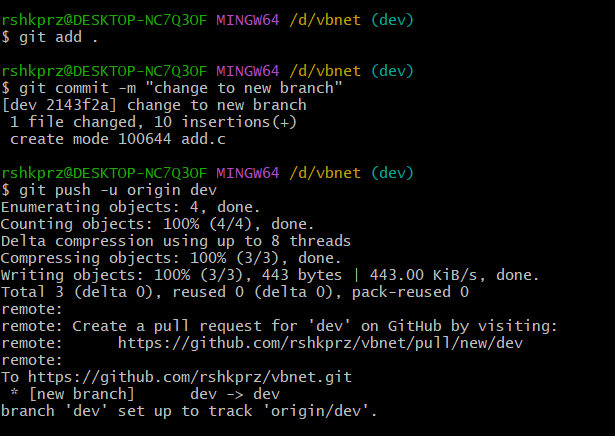
We checked the existing branch in our local repository. Then we create branches for working different versions of programs without affecting the main code.



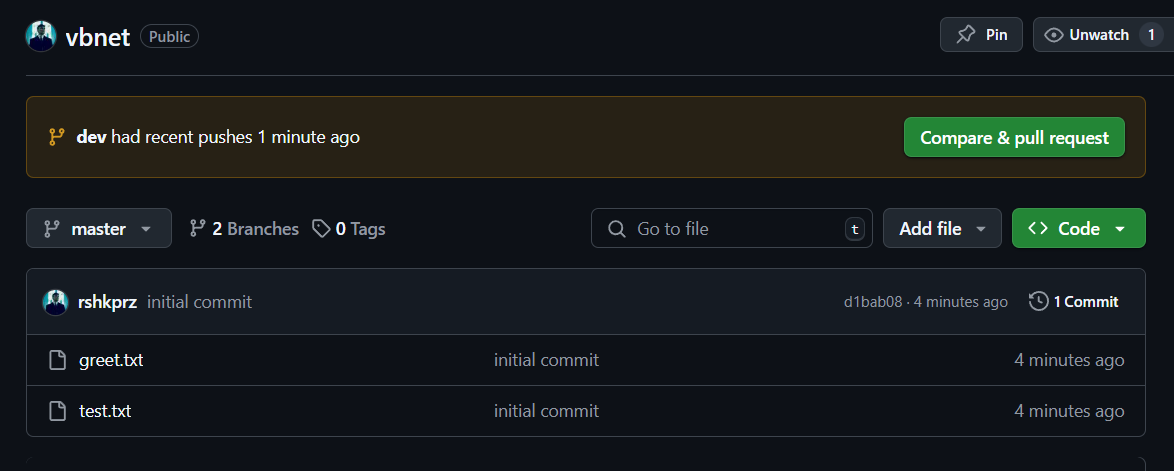
We switched to new branch “dev” branch where we modify and add different files without affecting the main code. We here added new files “add.c”. Initially it is in untracked stage.



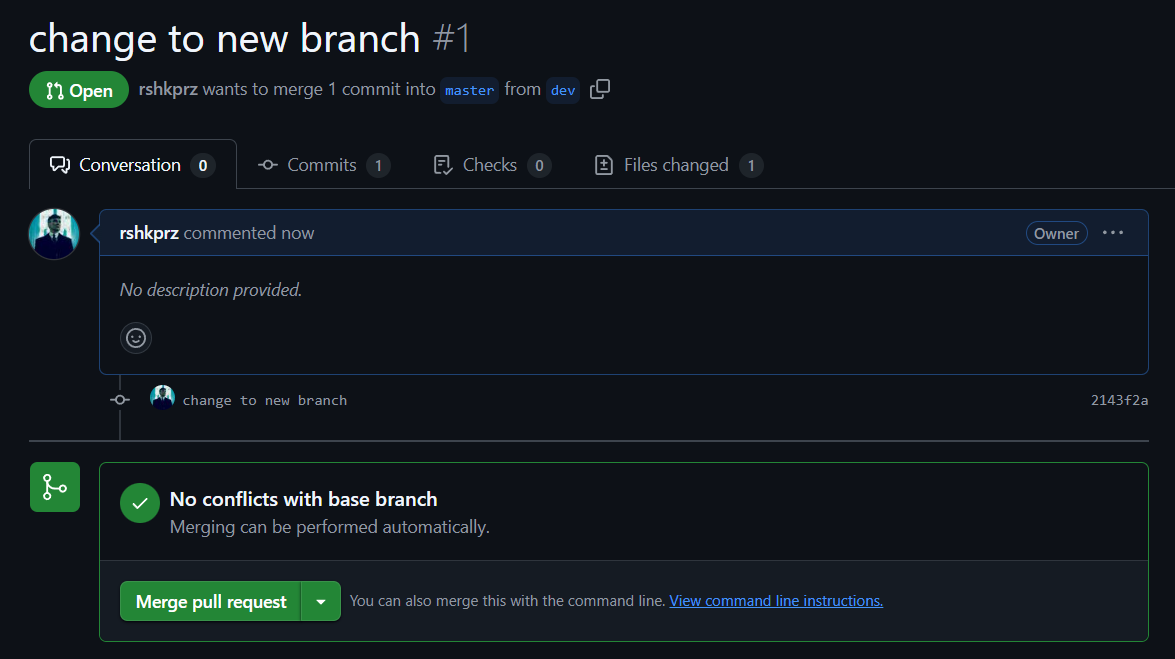
Then we commit the changes and push the branch in the GitHub to make sure the branch is visible to other users of the repository.



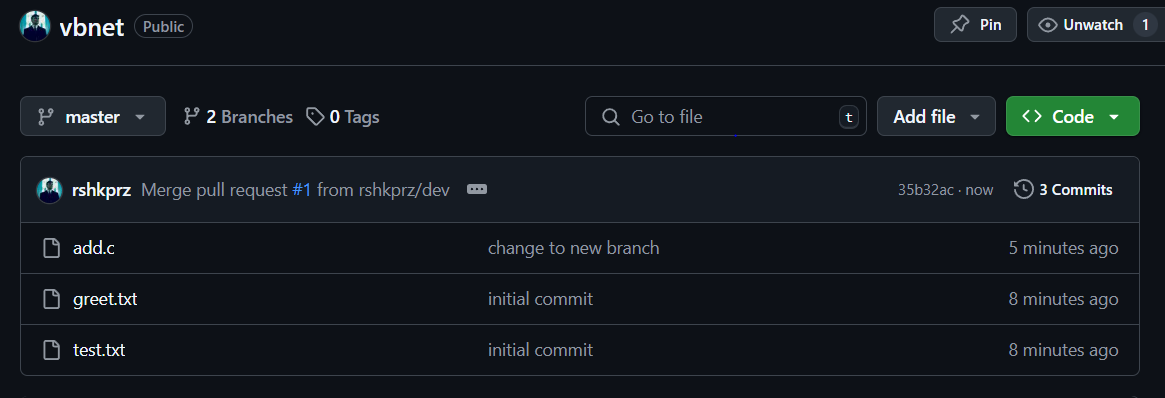
Then we create pull request for merging the latest branch pushed.



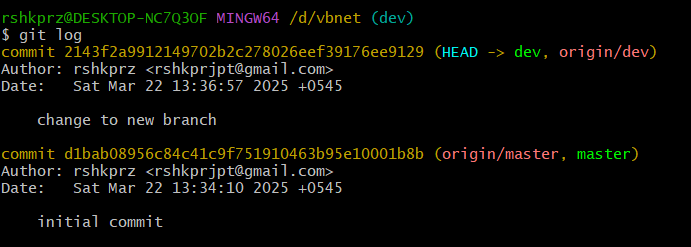
There was no any conflict so we merged the new branch into master branch.



After completion of merge we can see all the files in master branch.



With the help of command “git log” we viewed the history of the commits.



**Conclusion**

In this lab, we learned and performed different commands for initializing, linking, committing, branching and merging in Git and Github.