PURBANCHAL UNIVERSITY

KHWOPA ENGINEERING COLLEGE
LIBALI-08, BHAKTAPUR

LAB REPORT ON .NET LAB NO. 01

SUBMITTED BY:
Name: Saurav Basukala
Roll No. : 770339

SUBMITTED TO:
Department of
Computer Engineering

Submission: 2081/12/09

LAB 1 -GIT AND GITHUB

THEORY:-

Git is a distributed version control system designed for efficient source code management. As a free, open-source tool, it supports projects of varying scale, tracking changes to enable collaborative, non-linear development. Created by Linus Torvalds in 2005 for the Linux kernel, Git has become integral to software development.

Git Functionality

- Initialize a repository with git init, creating a .git subdirectory to monitor changes.
- Modifications occur when files are edited, added, or removed.
- Stage selected changes, then commit them to save a permanent snapshot.
- Access commit history and revert to prior states as needed.
- Git tracks changes incrementally, avoiding redundant file storage.

GitHub: A Collaborative Platform

GitHub is a web-based platform leveraging Git for version control and collaboration. It provides repositories for code storage, branches for parallel development, pull requests for reviews, and issue tracking for project management, supporting both individual and team workflows.

Forking and Cloning

- Forking duplicates another user's repository for personal use.
- Cloning retrieves a repository to a local machine for offline work.

Essential Git and GitHub Commands

Configuration

- git config --global user.name "Your Name"

Sets the username for commits.

- git config --global user.email "your_email@example.com"

Assigns an email to commits.

Initialization

- git init

Creates a new repository.

Remote Linking

- git remote add origin <repo>

Connects a local repository to a GitHub remote.

Staging and Committing

- git add .
  Stages all changes.
- git commit -m "message"
  Commits staged changes with a description.

Status and History

- git status
  Displays the working directory’s state.
- git log
  Shows commit history.

Branch Management

- git branch
  Lists existing branches.
- git branch <branch_name>
  Creates a new branch.
- git checkout <branch> or git switch <branch_name>
  Switches branches.
- git merge <branch_name>
  Merges a branch into the current one.

Remote Operations

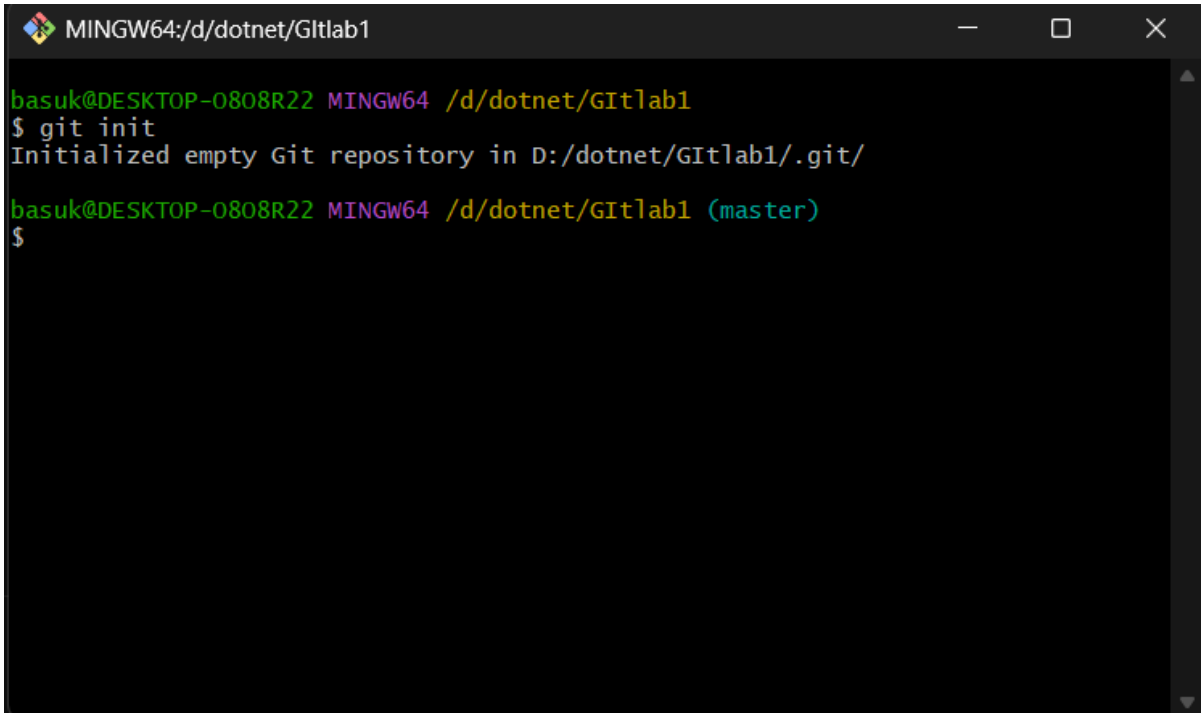
- git push -u origin <branch_name>
  Uploads commits to a remote repository.
- git pull origin
  Retrieves and integrates remote changes.

Cloning

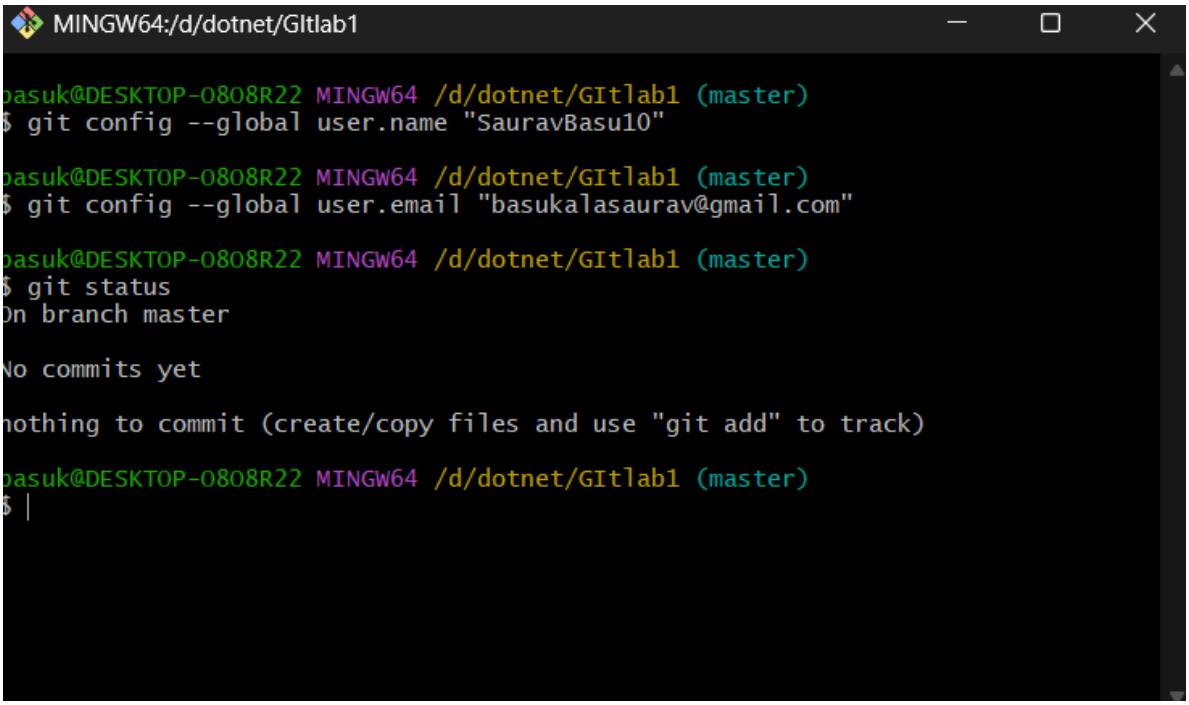
- git clone <repo_url>
  Downloads an existing repository.

Lab Work

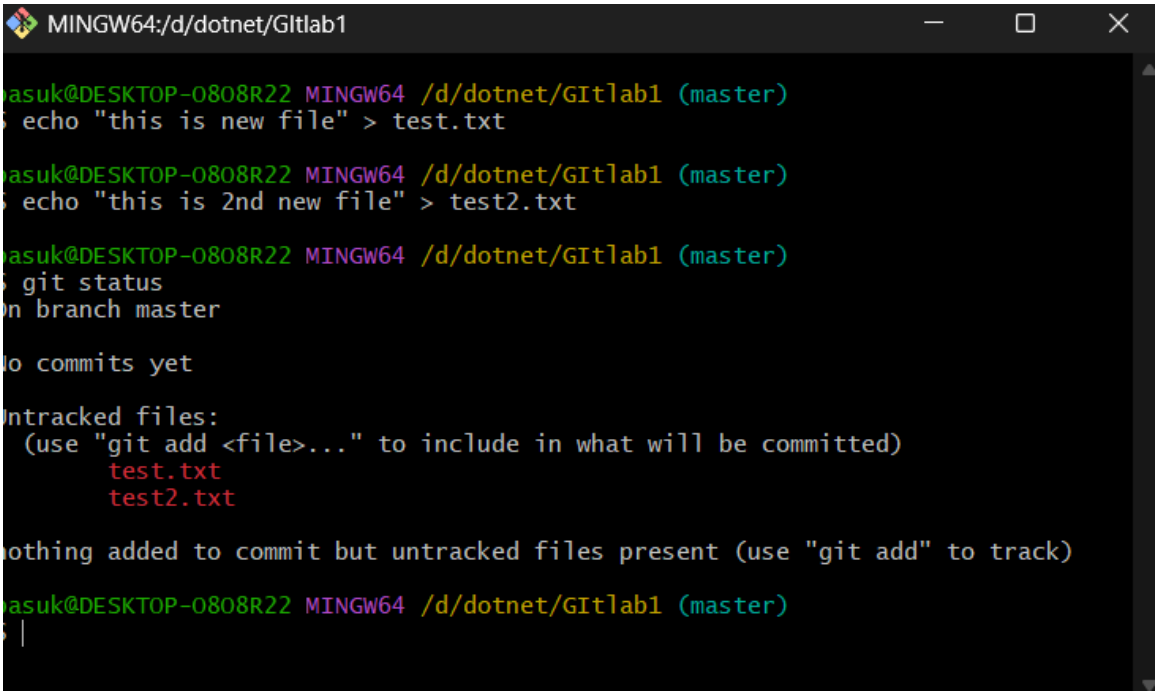
Initially, a folder is created to create, change the files using the version control git with different commands and git is initialized.



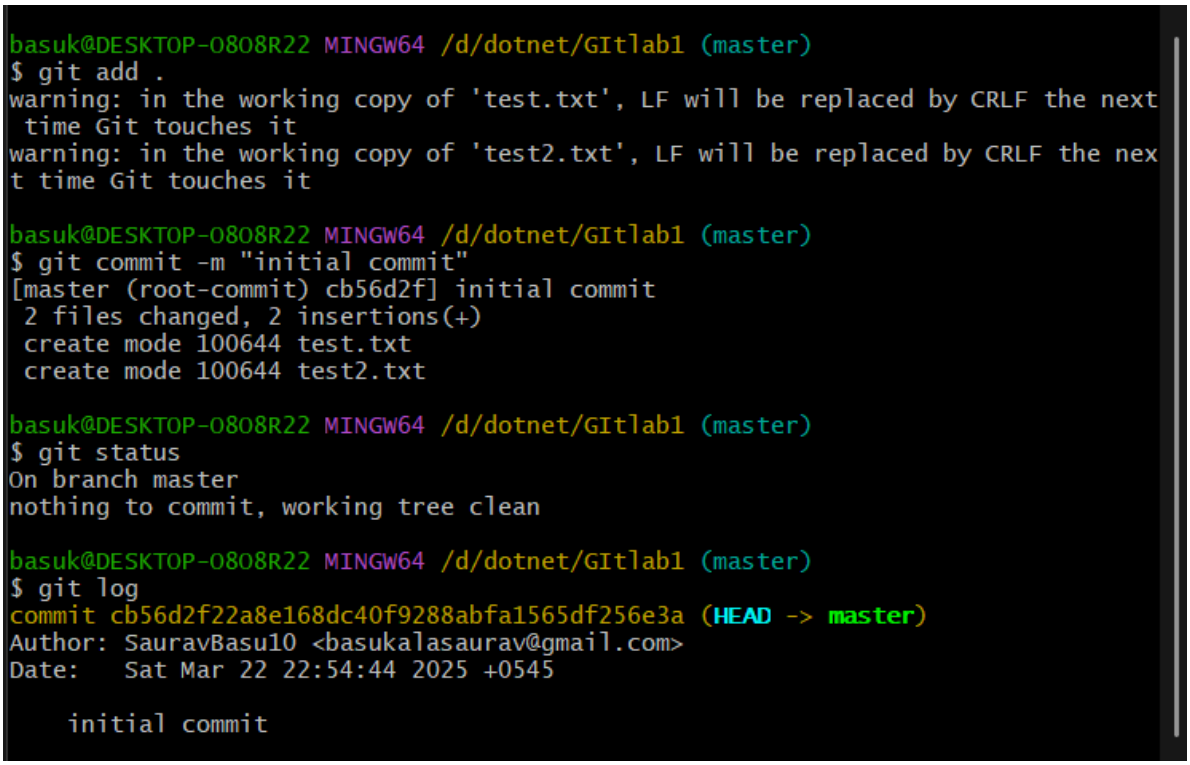
We set the global username and email of the GitHub. As no any files are created there is nothing to commit.



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 check 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.

```
basuk@DESKTOP-O8O8R22 MINGW64 /d/dotnet/GItlab1 (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   test.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

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.

```
basuk@DESKTOP-O8O8R22 MINGW64 /d/dotnet/GItlab1 (master)
$ git remote add origin https://github.com/Sauravbasu10/Dotnet.git
```

After that we push the files to the created repository.

```
basuk@DESKTOP-O8O8R22 MINGW64 /d/dotnet/GItlab1 (master)
$ git push -u origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 289 bytes | 96.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Sauravbasu10/Dotnet.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

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

```
basuk@DESKTOP-O8O8R22 MINGW64 /d/dotnet/GItlab1 (master)
$ git branch
* master

basuk@DESKTOP-O8O8R22 MINGW64 /d/dotnet/GItlab1 (master)
$ git branch Feature1

basuk@DESKTOP-O8O8R22 MINGW64 /d/dotnet/GItlab1 (master)
$ git branch
  Feature1
* master
```

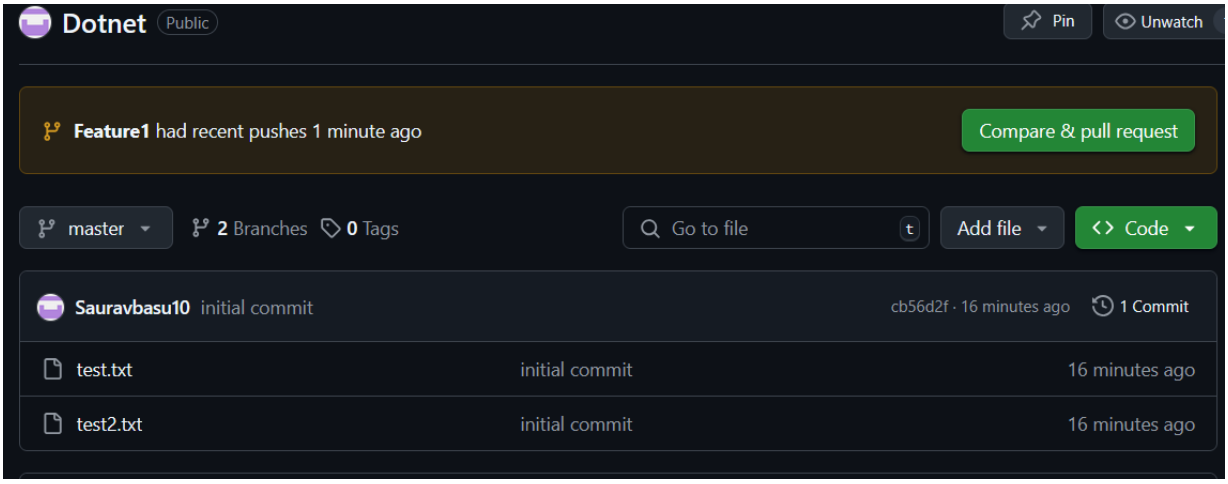
We switched to new branch “Feature1” branch where we modify and add different files without affecting the main code. We here added new files “calculation.py”. Initially it is in untracked stage.

```
basuk@DESKTOP-O8O8R22 MINGW64 /d/dotnet/GItlab1 (master)
$ git checkout Feature1
Switched to branch 'Feature1'
M       test.txt

basuk@DESKTOP-O8O8R22 MINGW64 /d/dotnet/GItlab1 (Feature1)
$ git status
On branch Feature1
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   test.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

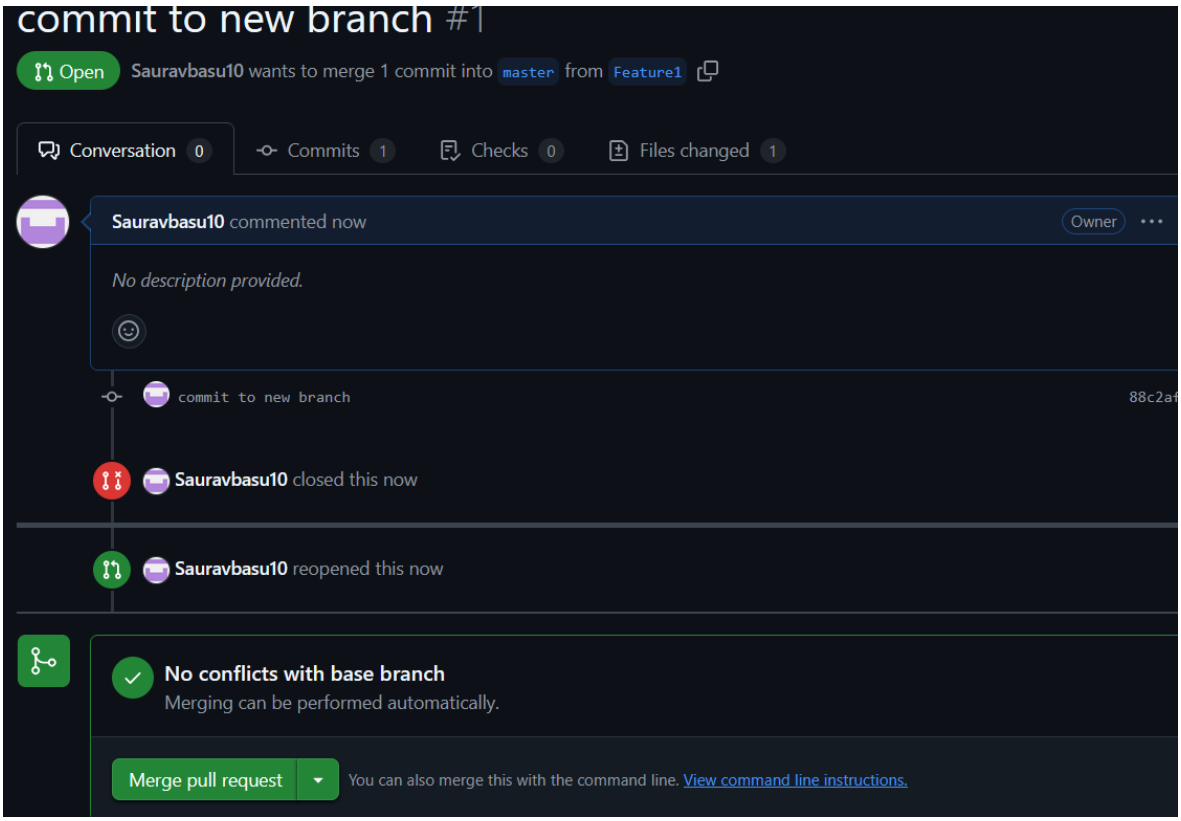
Then we commit the changes and push the branch in the GitHub to make sure the branch is visible to other users of the repository.

```
basuk@DESKTOP-O8O8R22 MINGW64 /d/dotnet/GItlab1 (Feature1)
$ git add .
warning: in the working copy of 'test.txt', LF will be replaced by CRLF the
 time Git touches it

basuk@DESKTOP-O8O8R22 MINGW64 /d/dotnet/GItlab1 (Feature1)
$ git commit -m "commit to new branch"
[Feature1 88c2afc] commit to new branch
 1 file changed, 1 insertion(+)

basuk@DESKTOP-O8O8R22 MINGW64 /d/dotnet/GItlab1 (Feature1)
$ git push -u origin Feature1
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 296 bytes | 296.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'Feature1' on GitHub by visiting:
remote:      https://github.com/Sauravbasu10/Dotnet/pull/new/Feature1
remote:
To https://github.com/Sauravbasu10/Dotnet.git
 * [new branch]      Feature1 -> Feature1
branch 'Feature1' set up to track 'origin/Feature1'.
```

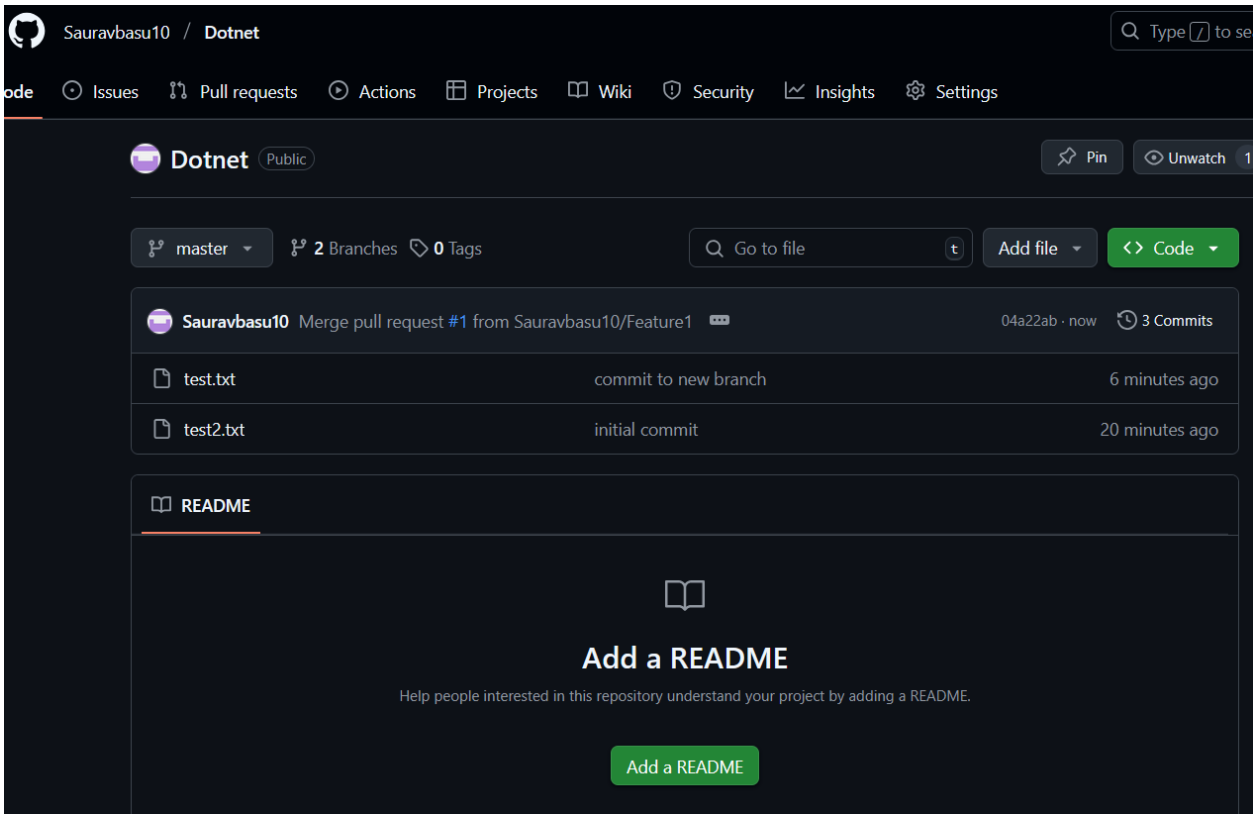
Then we create pull request for merging the latest branch pushed.



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, commiting, branching and merging in Git and Github.