DEPARTMENT OF INFORMATION TECHNOLOGY

Semester	S.E. Semester IV – Information Technology Engineering
Subject	Computer Networks and Network Design Lab
Subject Professor In-charge	Unnati Gohil
Assisting Teachers	-
Laboratory	MS Teams

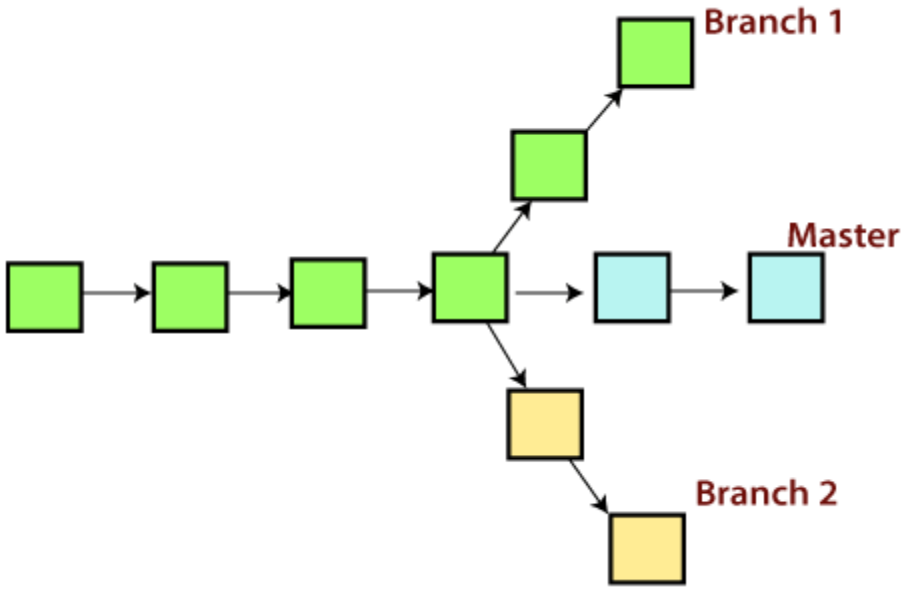
Student Name	Sanika Kate	
Roll Number	22101A2005	
Grade and Subject Teacher's Signature		

Experiment Number	2	
Experiment Title	Using github and git tools made a branch	
Resources / Apparatus Required	Hardware: Basic Desktop with Windows or Linux.	Software: Java/ Python/Wireshark/Cisco Packet Tracer
Objectives (Skill Set / Knowledge Tested / Imparted)		

Theory:

Git Branch

A branch is a version of the repository that diverges from the main working project. It is a feature available in most modern version control systems. A Git project can have more than one branch. These branches are a pointer to a snapshot of your changes. When you want to add a new feature or fix a bug, you spawn a new branch to summarize your changes. So, it is complex to merge the unstable code with the main code base and also facilitates you to clean up your future history before merging with the main branch.



Git Master Branch

The master branch is a default branch in Git. It is instantiated when first commit made on the project. When you make the first commit, you're given a master branch to the starting commit point. When you start making a commit, then master branch pointer automatically moves forward. A repository can have only one master branch.
Master branch is the branch in which all the changes eventually get merged back. It can be called as an official working version of your project.

How it works

A branch represents an independent line of development. Branches serve as an abstraction for the edit/stage/commit process. You can think of them as a way to request a brand new working directory, staging area, and project history. New commits are recorded in the history for the current branch, which results in a fork in the history of the project.
The git branch command lets you create, list, rename, and delete branches. It doesn't let you switch between branches or put a forked history back together again. For this reason, git branch is tightly integrated with the git checkout and git merge commands.

COMMANDS

1. **git branch NewBranch**

You can create a new branch with the help of the git branch command

2. **git checkout NewBranch**

The git checkout command lets you navigate between the branches created by git branch

3. **touch filename.txt** to create a new file

4. **git status**

The git status command displays the state of the working directory and the staging area

5.git add filename.txt

The git add command adds new or changed files in your working directory to the Git staging area

6.git commit -m "message"

Commits can be thought of as snapshots or milestones along the timeline of a Git project. Commits are created with the git commit command to capture the state of a project at that point in time.

7.git push --u origin NewBranch

8.git checkout master

9.git merge NewBranch

Git allows you to merge the other branch with the currently active branch. You can merge two branches with the help of git merge command.

10.git push -u origin master push the file into the master branch

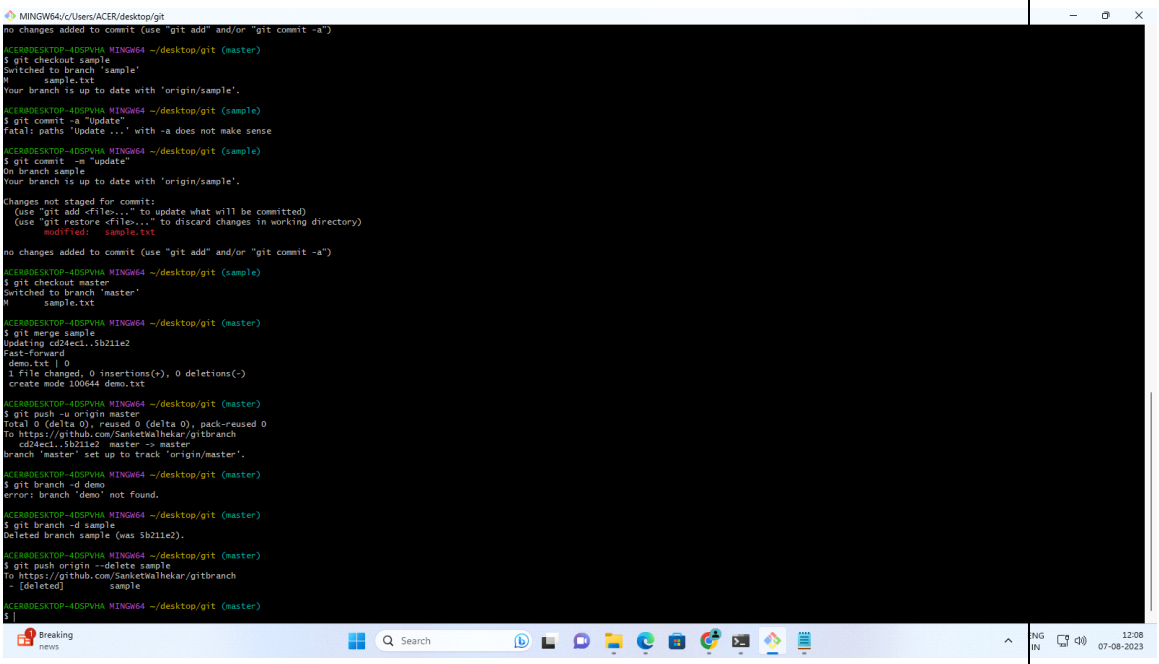
11.git branch -d NewBranch

You can delete the specified branch. It is a safe operation. In this command, Git prevents you from deleting the branch if it has unmerged changes

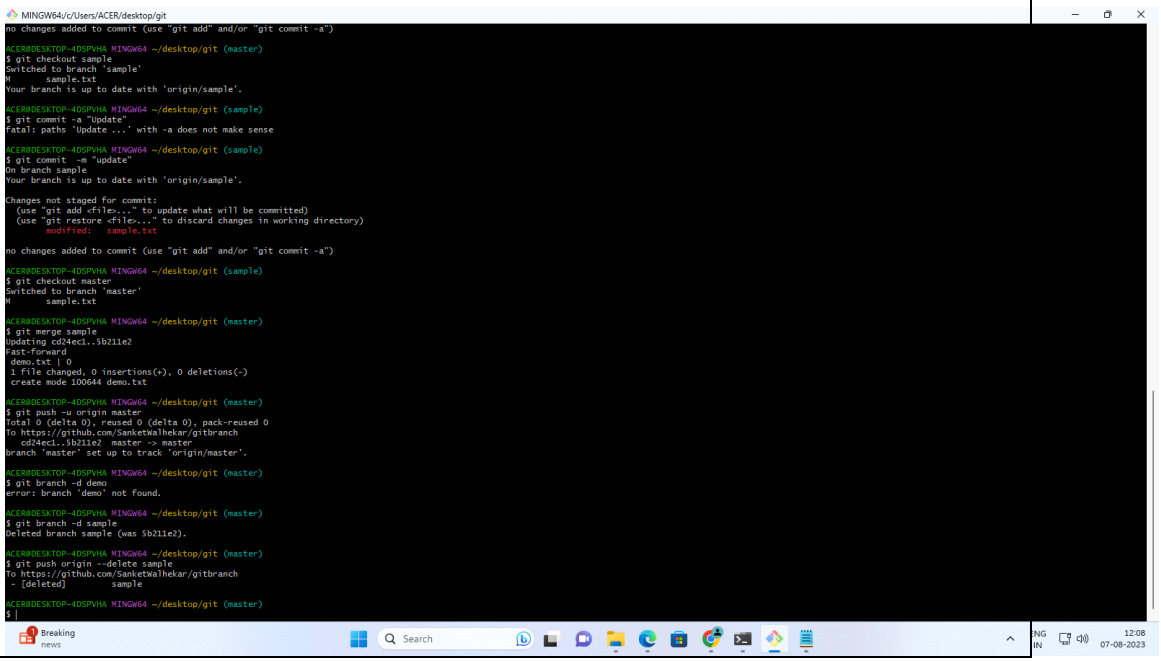
12.git push origin --delete NewBranch

You can delete a remote branch from Git desktop application

Output



```
MINGW64:/c/Users/ACER/desktop/git
no changes added to commit (use "git add" and/or "git commit -a")

ACER@DESKTOP-4DSPVHA MINGW64 ~/desktop/git (master)
$ git checkout sample
Switched to branch 'sample'
M       sample.txt
Your branch is up to date with 'origin/sample'.

ACER@DESKTOP-4DSPVHA MINGW64 ~/desktop/git (sample)
$ git commit -a "Update"
fatal: paths 'Update ...' with -a does not make sense

ACER@DESKTOP-4DSPVHA MINGW64 ~/desktop/git (sample)
$ git commit  -m "update"
On branch sample
Your branch is up to date with 'origin/sample'.

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:   sample.txt

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

ACER@DESKTOP-4DSPVHA MINGW64 ~/desktop/git (sample)
$ git checkout master
Switched to branch 'master'
M       sample.txt

ACER@DESKTOP-4DSPVHA MINGW64 ~/desktop/git (master)
$ git merge sample
Updating cd24ec1..5b211e2
Fast-forward
 demo.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 demo.txt

ACER@DESKTOP-4DSPVHA MINGW64 ~/desktop/git (master)
$ git push -u origin master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/SanketWalhekar/gitbranch
   cd24ec1..5b211e2  master -> master
branch 'master' set up to track 'origin/master'.

ACER@DESKTOP-4DSPVHA MINGW64 ~/desktop/git (master)
$ git branch -d demo
error: branch 'demo' not found.

ACER@DESKTOP-4DSPVHA MINGW64 ~/desktop/git (master)
$ git branch -d sample
Deleted branch sample (was 5b211e2).

ACER@DESKTOP-4DSPVHA MINGW64 ~/desktop/git (master)
$ git push origin --delete sample
To https://github.com/SanketWalhekar/gitbranch
 - [deleted]         sample

ACER@DESKTOP-4DSPVHA MINGW64 ~/desktop/git (master)
$ |
```



```
MINGW64:/c/Users/ACER/desktop/git
no changes added to commit (use "git add" and/or "git commit -a")

ACER@DESKTOP-4DSPVHA MINGW64 ~/desktop/git (master)
$ git checkout sample
Switched to branch 'sample'
M       sample.txt
Your branch is up to date with 'origin/sample'.

ACER@DESKTOP-4DSPVHA MINGW64 ~/desktop/git (sample)
$ git commit -a "Update"
fatal: paths 'Update ...' with -a does not make sense

ACER@DESKTOP-4DSPVHA MINGW64 ~/desktop/git (sample)
$ git commit  -m "update"
On branch sample
Your branch is up to date with 'origin/sample'.

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:   sample.txt

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

ACER@DESKTOP-4DSPVHA MINGW64 ~/desktop/git (sample)
$ git checkout master
Switched to branch 'master'
M       sample.txt

ACER@DESKTOP-4DSPVHA MINGW64 ~/desktop/git (master)
$ git merge sample
Updating cd24ec1..5b211e2
Fast-forward
 demo.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 demo.txt

ACER@DESKTOP-4DSPVHA MINGW64 ~/desktop/git (master)
$ git push -u origin master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/SanketWalhekar/gitbranch
   cd24ec1..5b211e2  master -> master
branch 'master' set up to track 'origin/master'.

ACER@DESKTOP-4DSPVHA MINGW64 ~/desktop/git (master)
$ git branch -d demo
error: branch 'demo' not found.

ACER@DESKTOP-4DSPVHA MINGW64 ~/desktop/git (master)
$ git branch -d sample
Deleted branch sample (was 5b211e2).

ACER@DESKTOP-4DSPVHA MINGW64 ~/desktop/git (master)
$ git push origin --delete sample
To https://github.com/SanketWalhekar/gitbranch
 - [deleted]         sample

ACER@DESKTOP-4DSPVHA MINGW64 ~/desktop/git (master)
$ |
```

Conclusion	In this document we discussed Git's branching behavior and the git branch command. The git branch commands primary functions are to create, list, rename and delete branches. To operate further on the resulting branches the command is commonly used with other commands like git checkout. Learn more about git checkout branch operations; such as switching branches and merging branches, on the git checkout page.