Name: Mrs. Rupali N Hosmani.

Course: Executive Post Graduate Certification in Cloud Computing

Contact No. 7720003531

**Assignment 1: GIT Assignment**

**Task To Be performed:**

**A] GIT Assignment 1:**

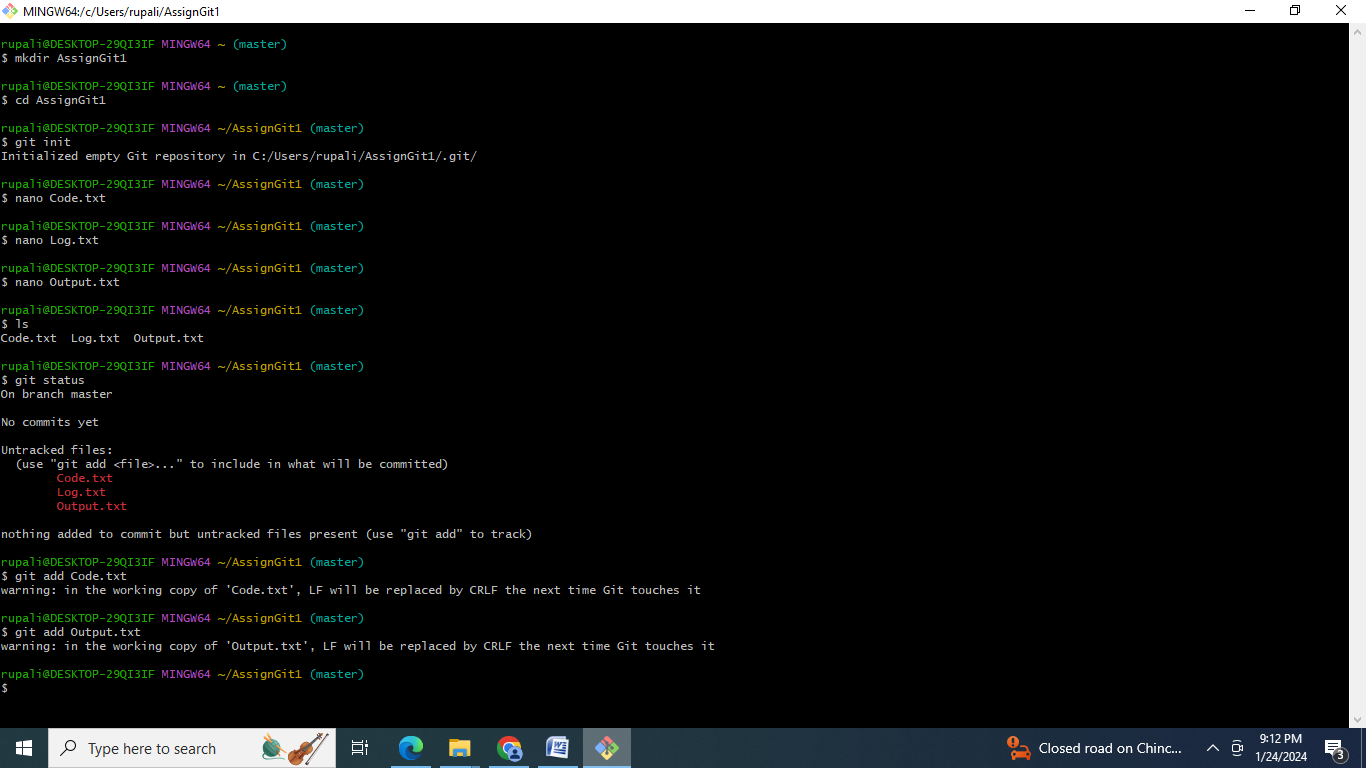
● Create a new folder

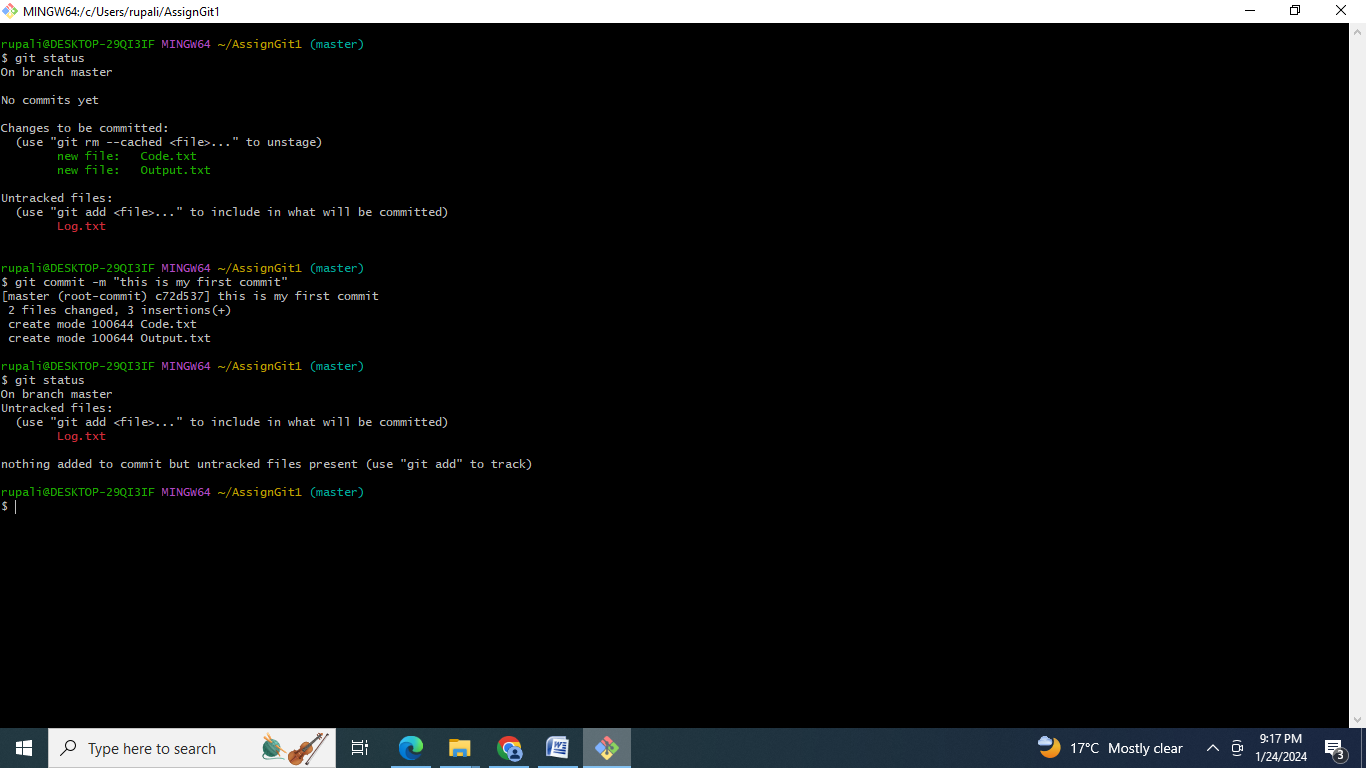
● Put the following files in the folder ○ Code.txt ○ Log.txt ○ Output.txt

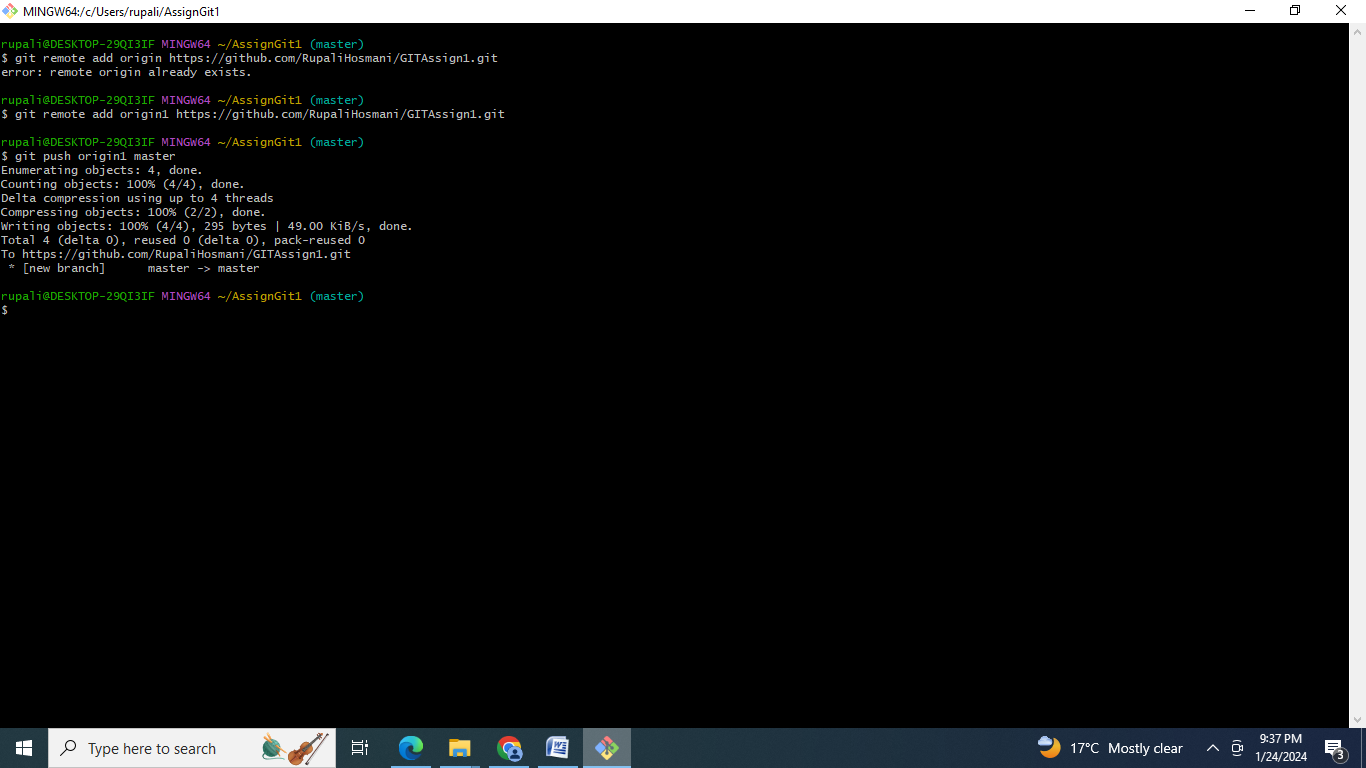
● Stage the Code.txt and Output.txt files

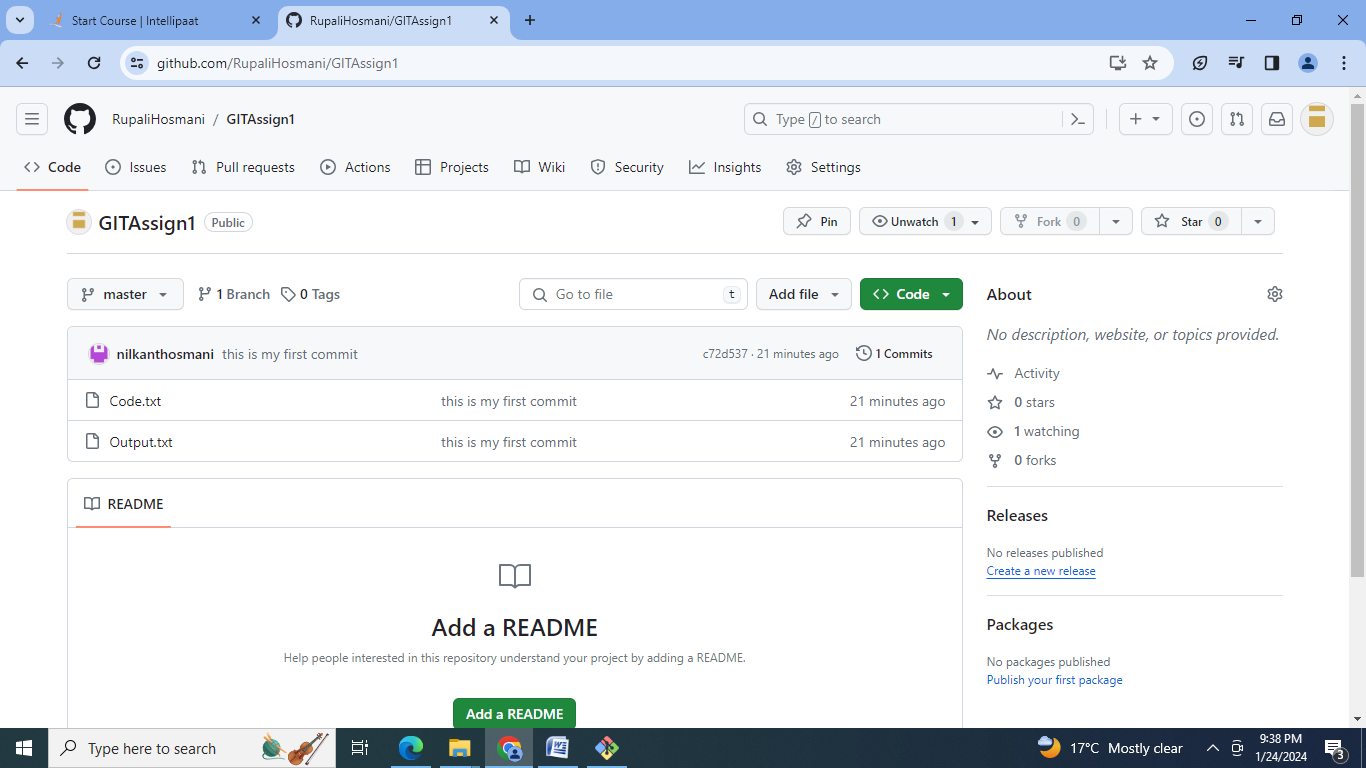
● Commit them

● And finally push them to github.









Commands:

mkdir AssignGit1

cd AssignGit1

git init

nano Code.txt

nano Log.txt

nano Output.txt

ls

git status

git add Code.txt

git add Output.txt

clear

git status

git commit -m "this is my first commit"

git status

git remote add origin <https://github.com/RupaliHosmani/GITAssign1.git>

git remote add origin1 https://github.com/RupaliHosmani/GITAssign1.git

git push origin1 master

**B] GIT Assignment2:**

● Create a git working directory with feature1.txt and feature2.txt in the master branch

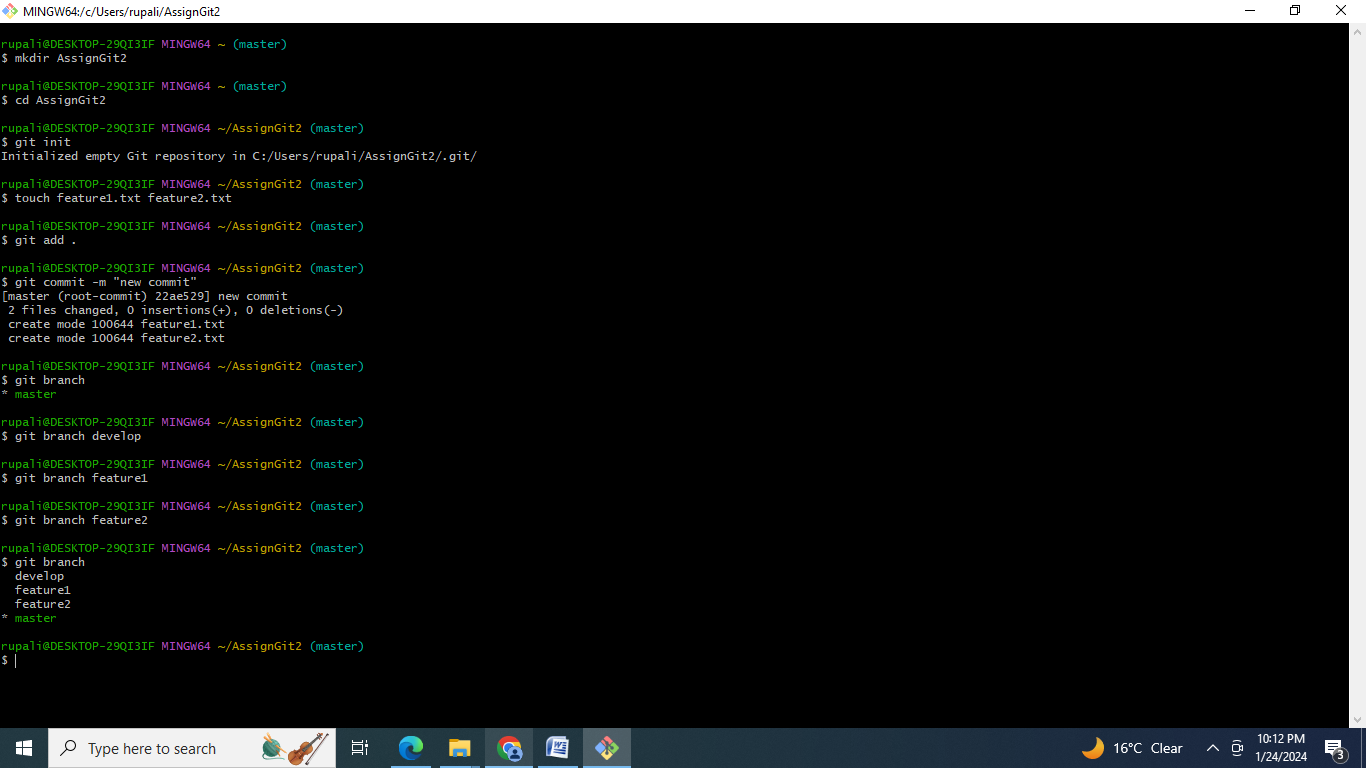
● Create 3 branches develop, feature1 and feature2

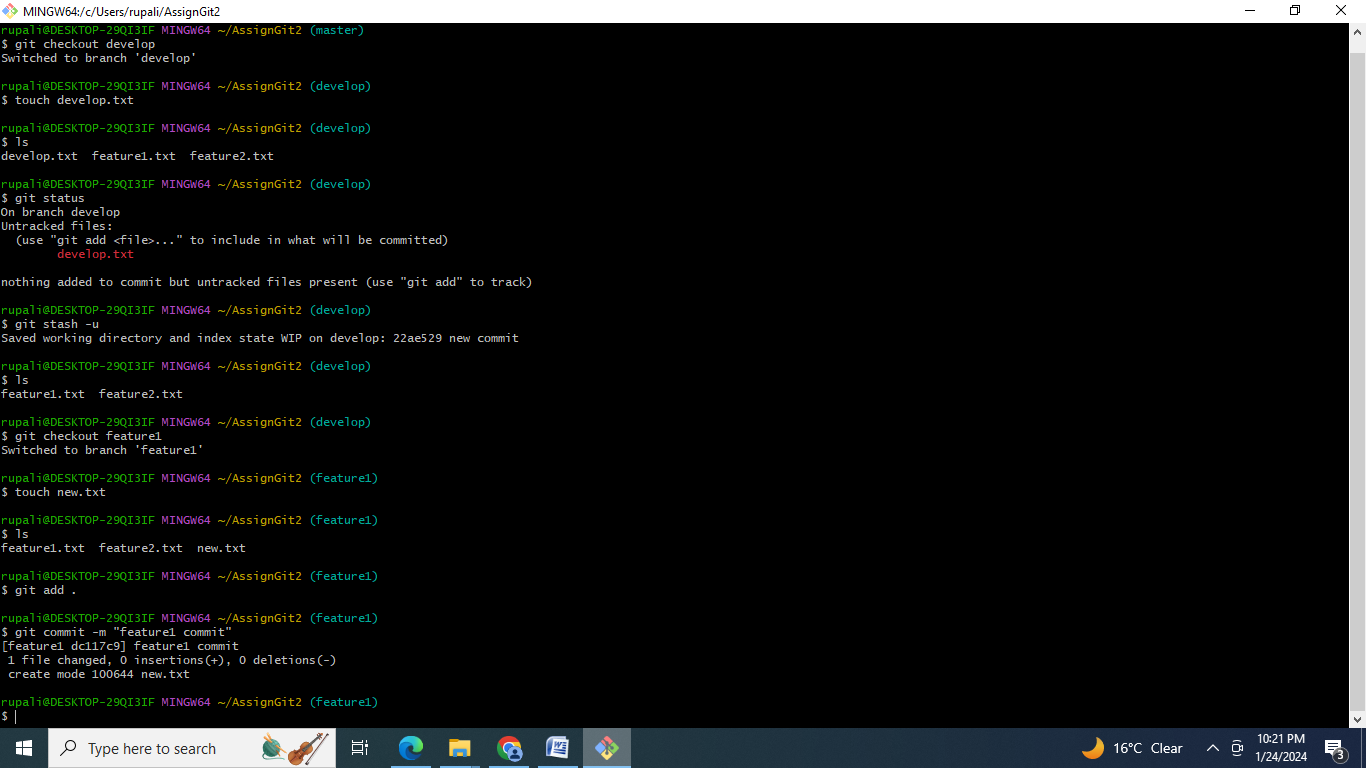
● In develop branch create develop.txt, do not stage or commit it

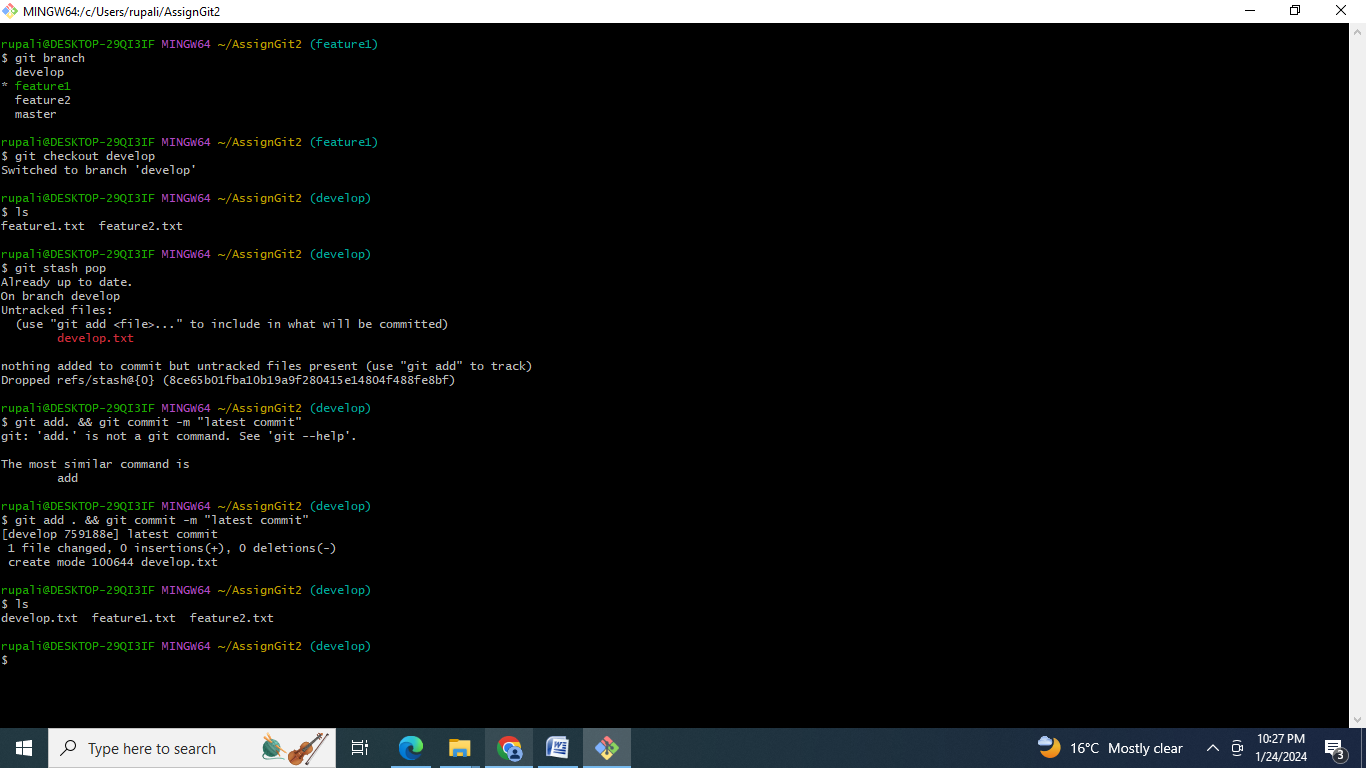
● Stash this file, and checkout to feature1 branch

● Create new.txt file in feature1 branch, stage and commit this file

● Checkout to develop, unstash this file and commit.







Commands:

mkdir AssignGit2

cd AssignGit2

git init

touch feature1.txt feature2.txt

git add .

git commit -m "new commit"

git branch

git branch develop

git branch feature1

git branch feature2

git branch

clear

git checkout develop

touch develop.txt

ls

git status

git stash -u

ls

git checkout feature1

touch new.txt

ls

git add .

git commit -m "feature1 commit"

clear

git branch

git checkout develop

ls

git stash pop

git add . && git commit -m "latest commit"

ls.

**C] GIT Assignment 3**

● Create a git working directory, with the following branches ○ Develop ○ F1 ○ f2

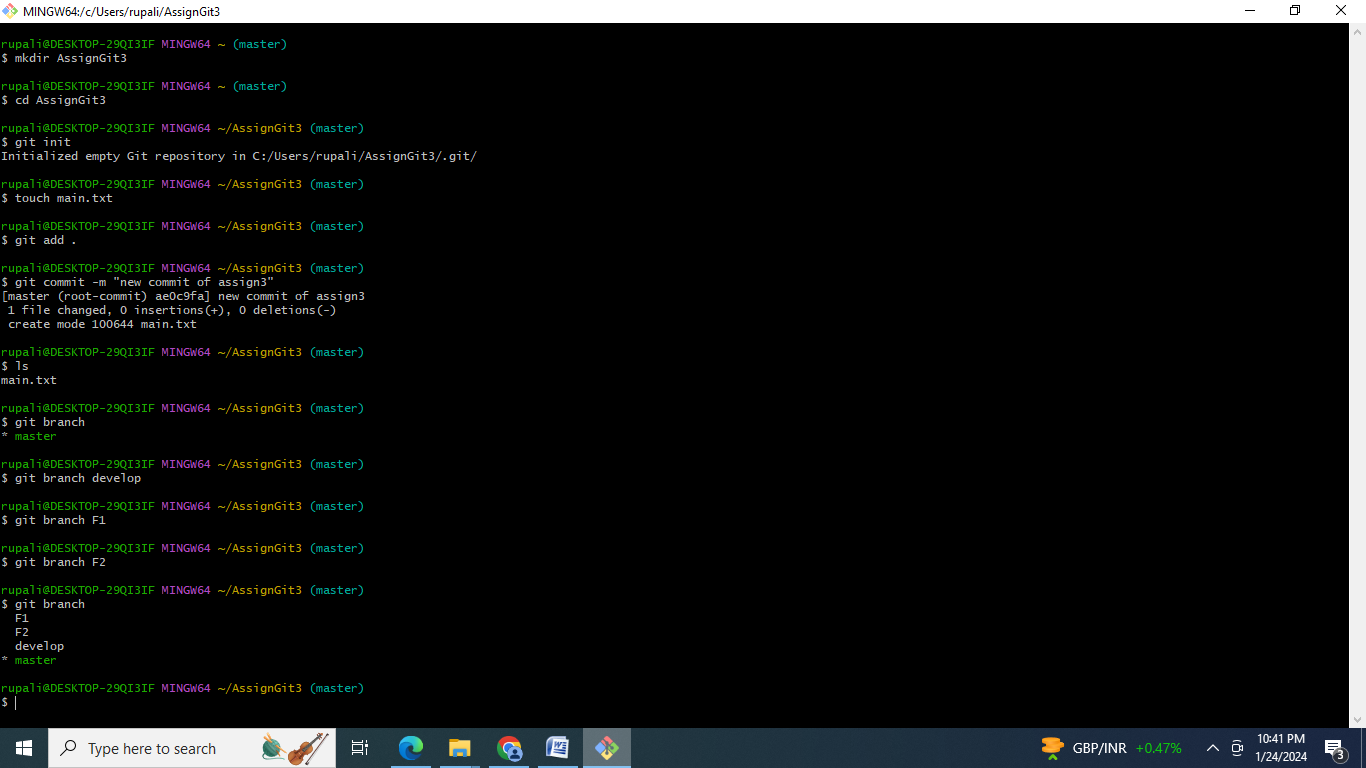
● In the master branch, commit main.txt file

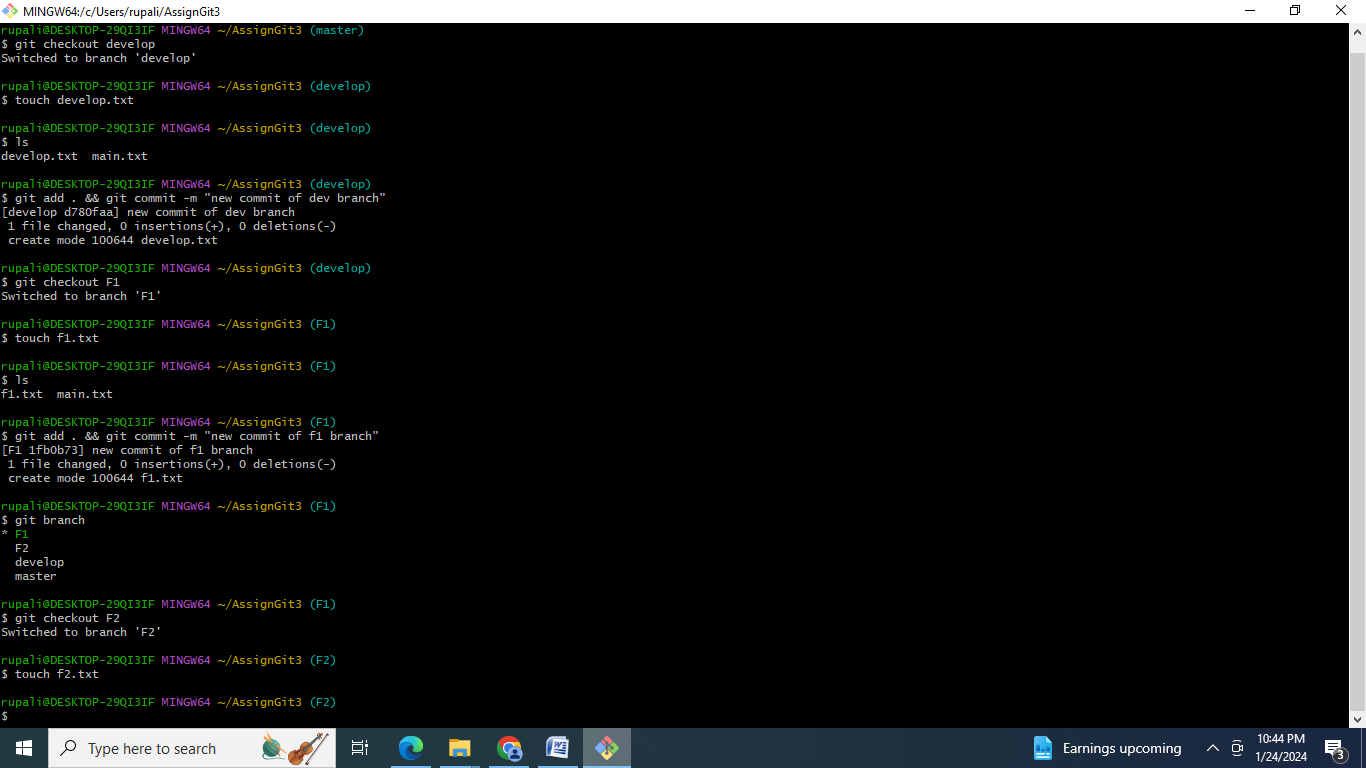
● Put develop.txt in develop branch, f1.txt and f2.txt in f1 and f2 respectively

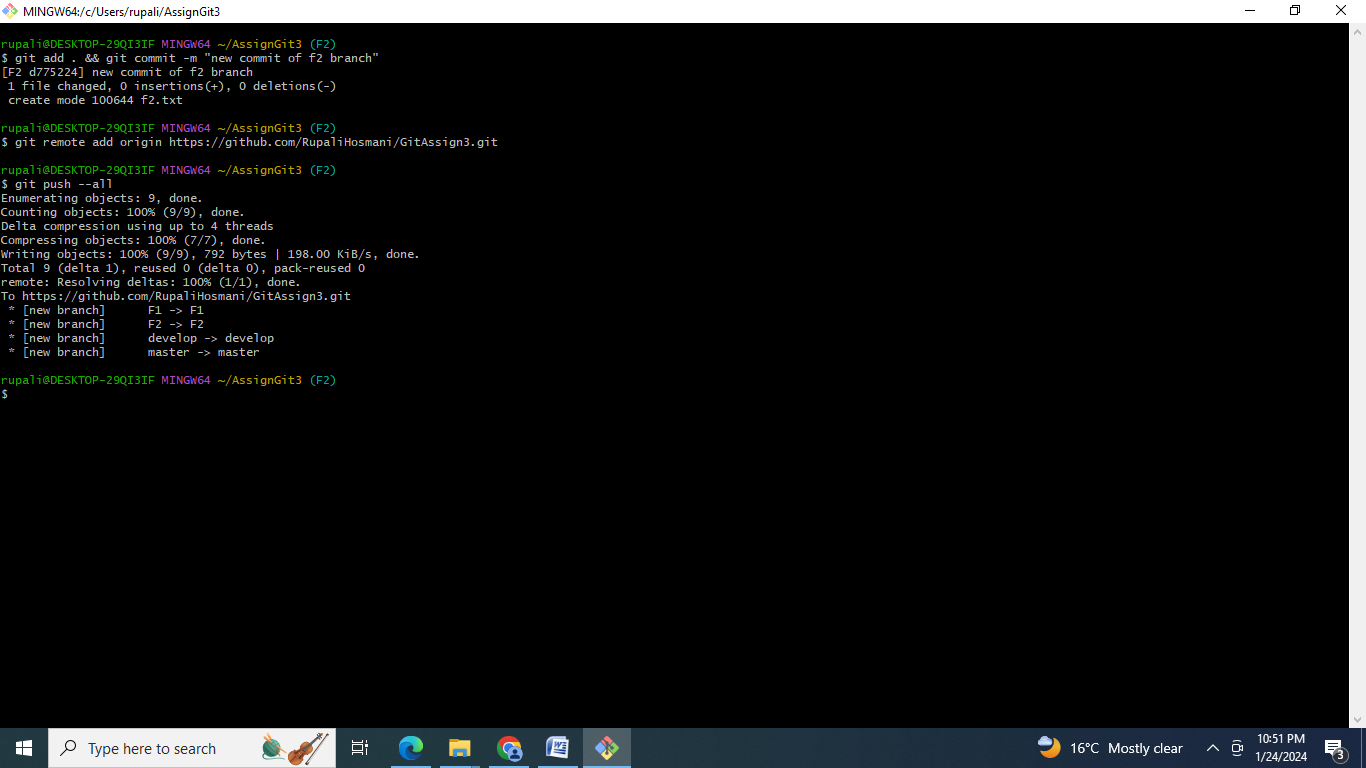
● Push all these branches to github

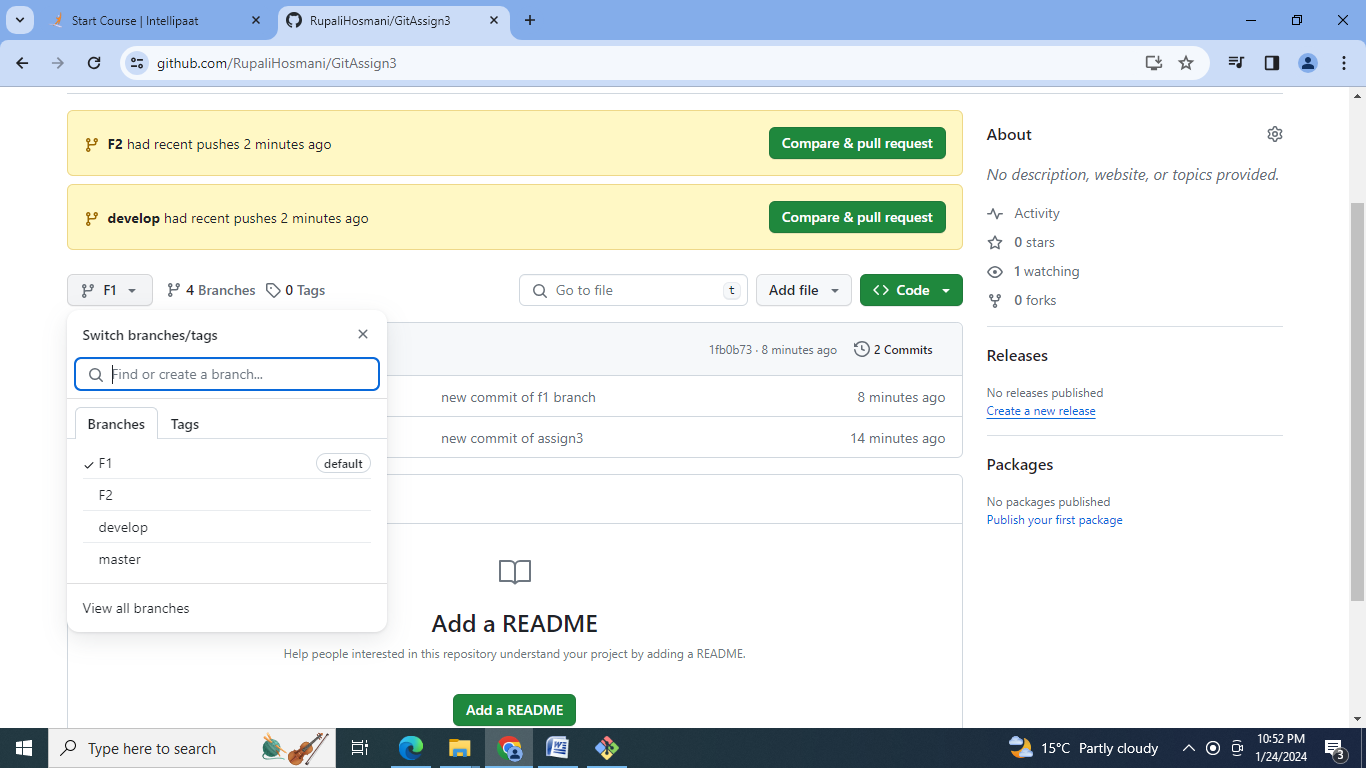
● On local delete f2 branch

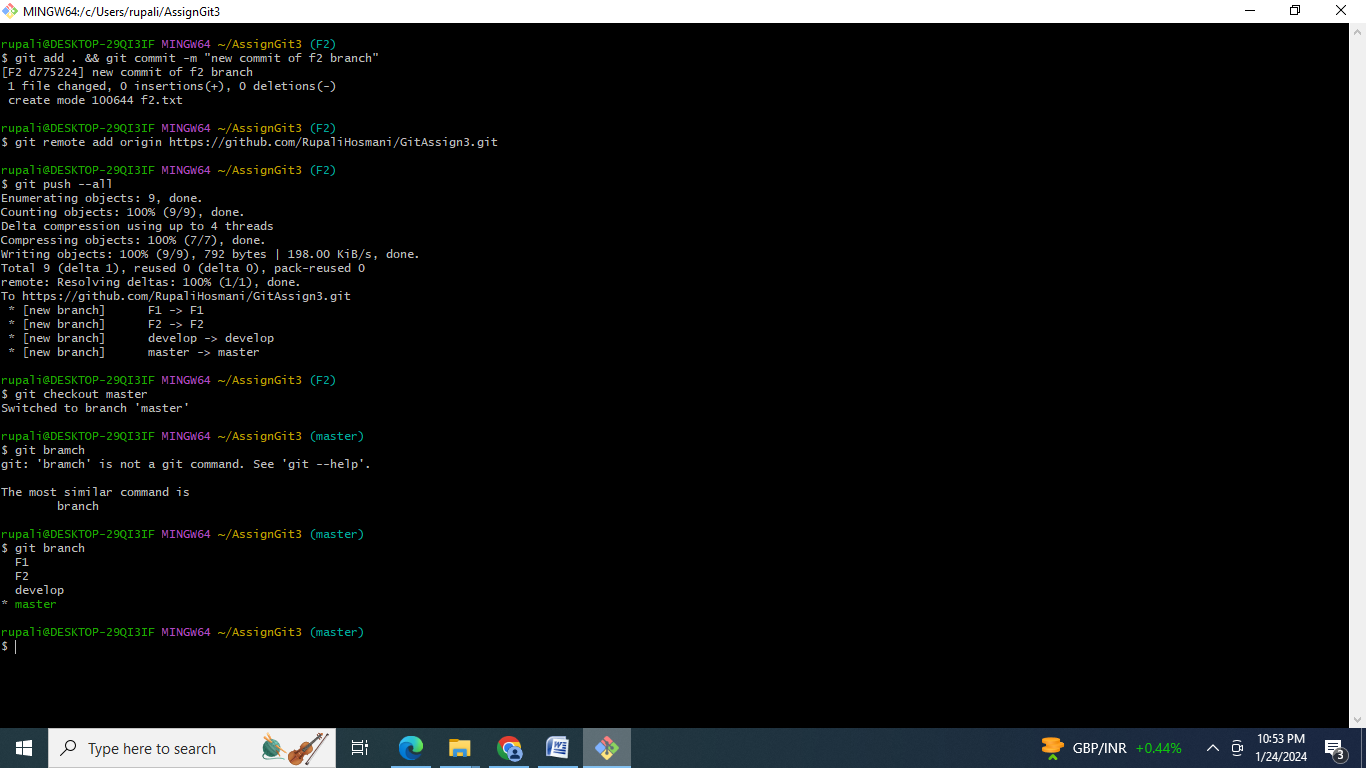
● Delete the same branch on github as well

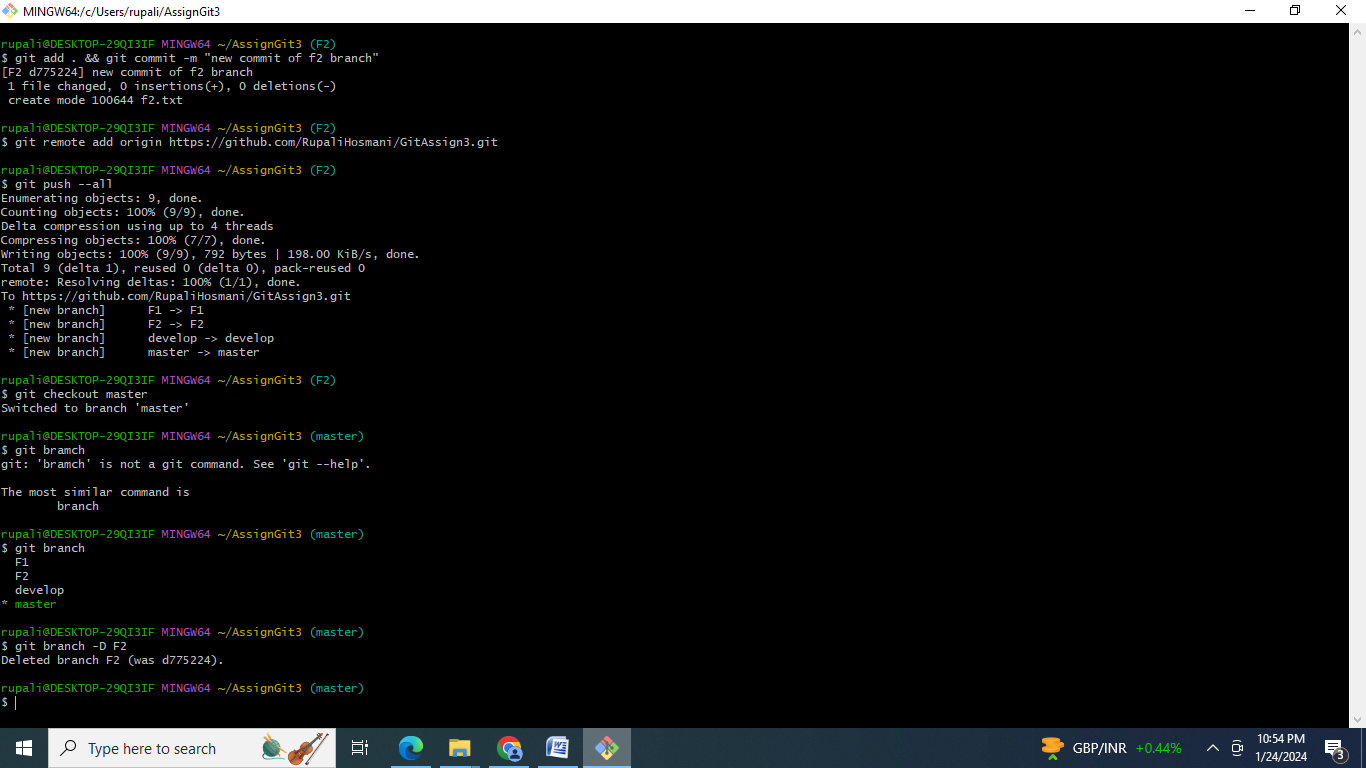


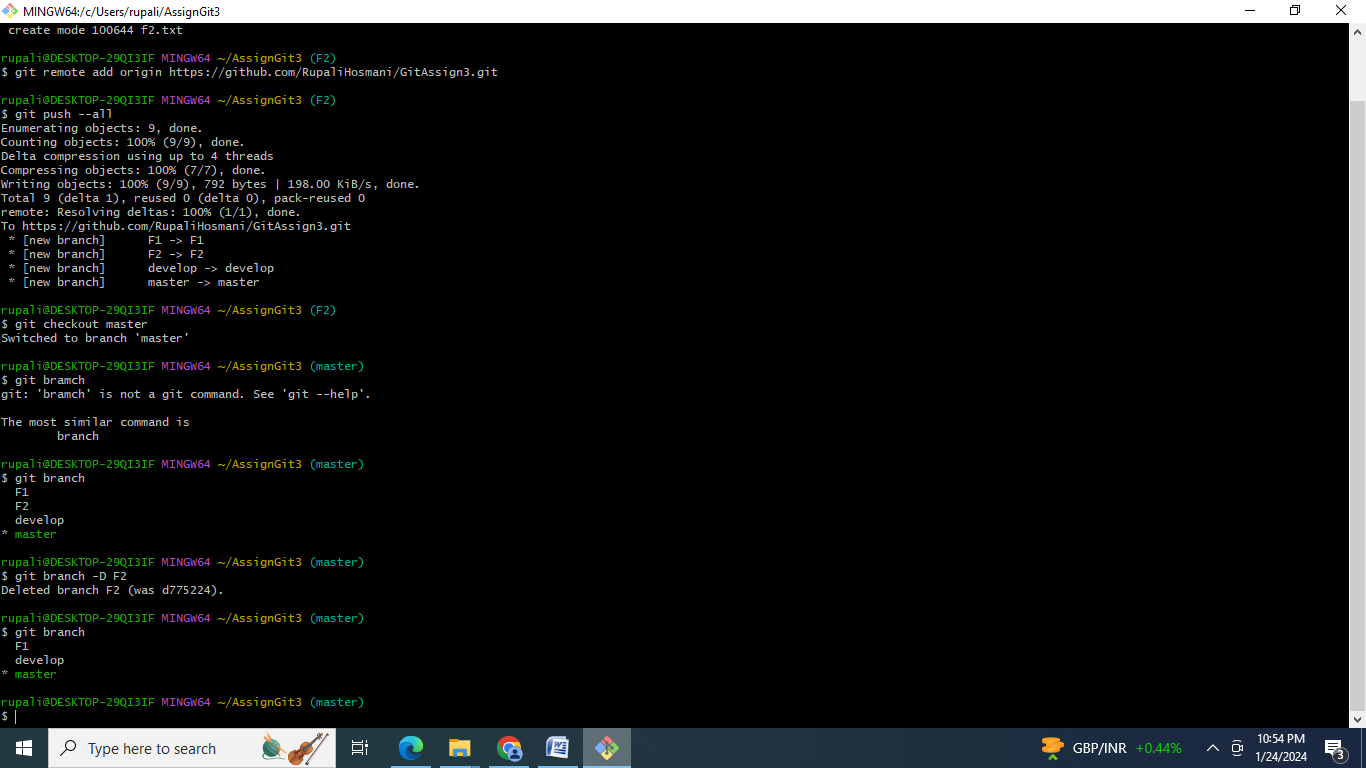


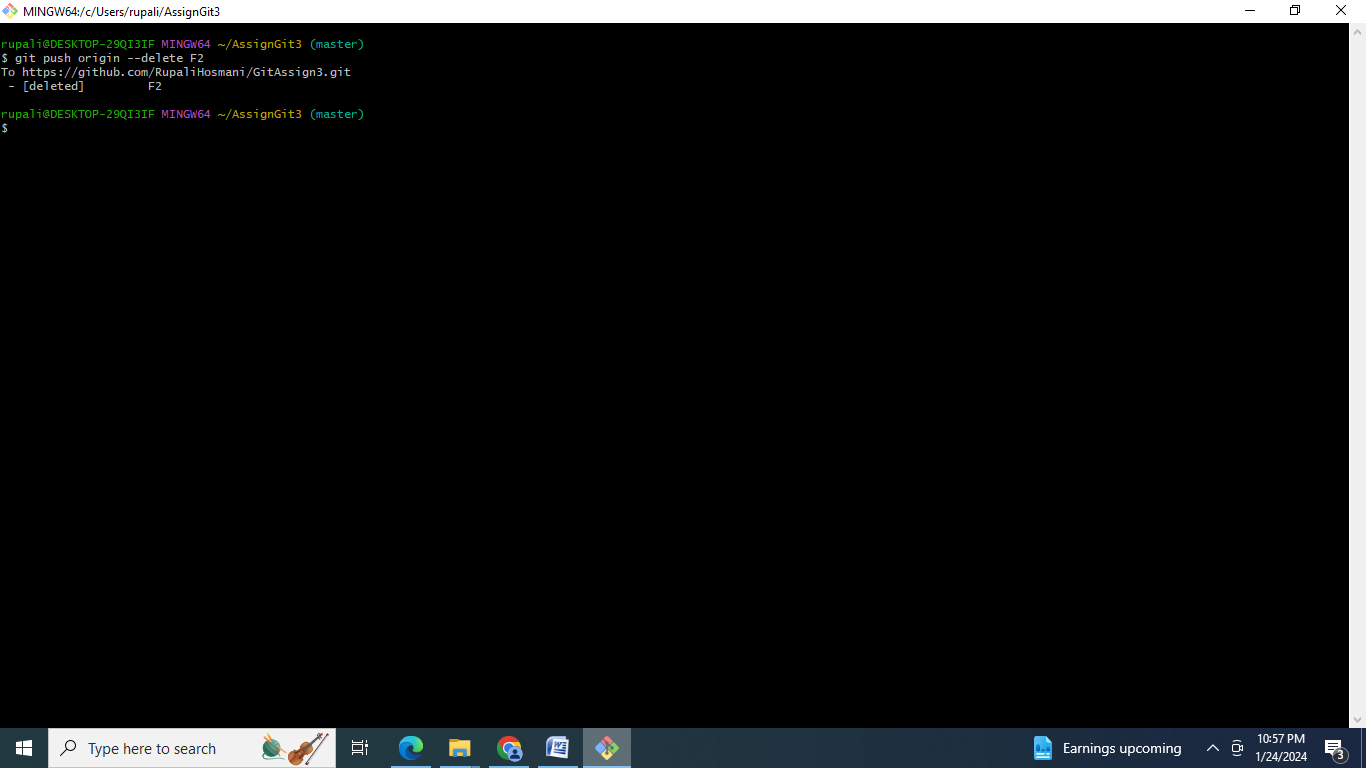


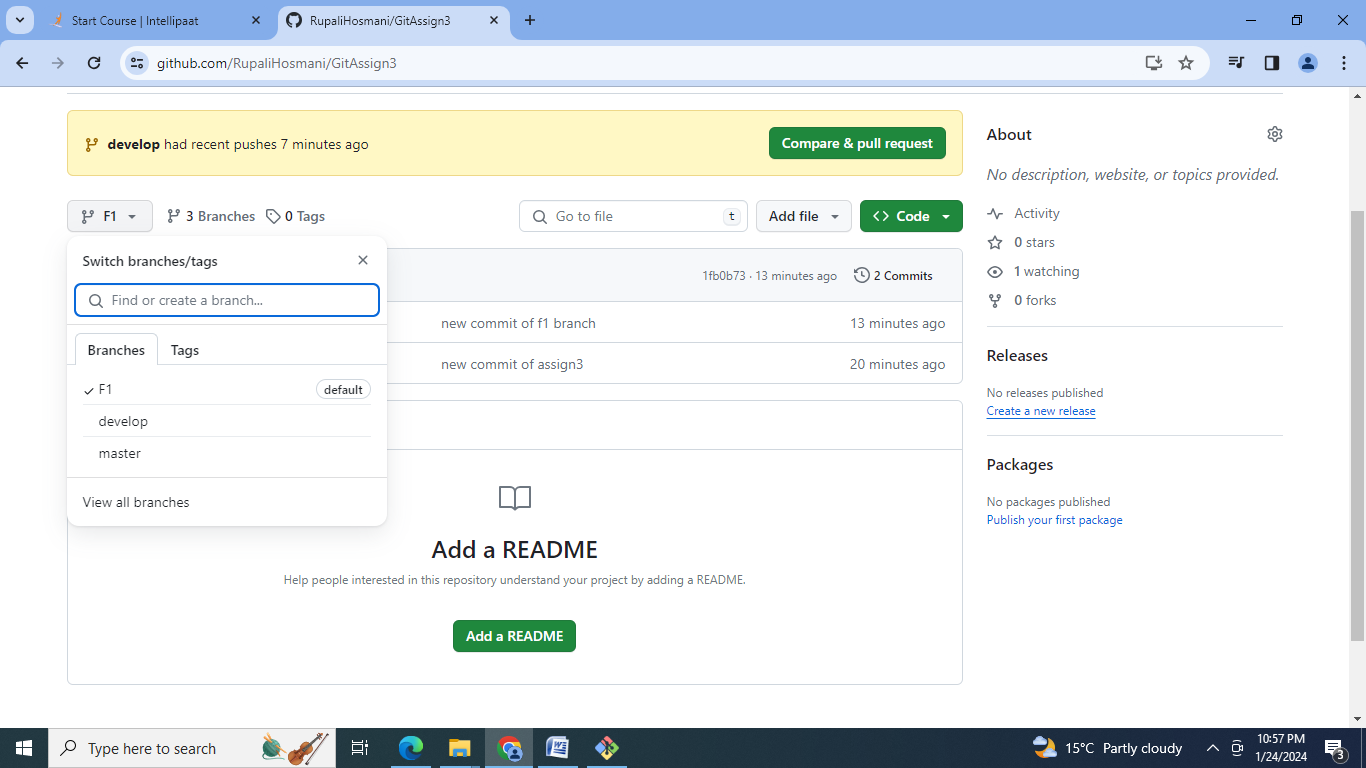












Commands:

mkdir AssignGit3

cd AssignGit3

git init

touch main.txt

git add .

git commit -m "new commit of assign3"

ls

git branch

git branch develop

git branch F1

git branch F2

git branch

clear

git checkout develop

touch develop.txt

ls

git add . && git commit -m "new commit of dev branch"

git checkout F1

touch f1.txt

ls

git add . && git commit -m "new commit of f1 branch"

git branch

git checkout F2

touch f2.txt

git add . && git commit -m "new commit of f2 branch"

git remote add origin https://github.com/RupaliHosmani/GitAssign3.git

git push --all

git checkout master

git branch

git branch -D F2

git branch

git push origin --delete F2.

D] GIt Assignment 4:

● Put master.txt on master branch, stage and commit

● Create 3 branches: public1, public2 and private

● Put public1.txt on public 1 branch, stage and commit

● Merge public 1 on master branch

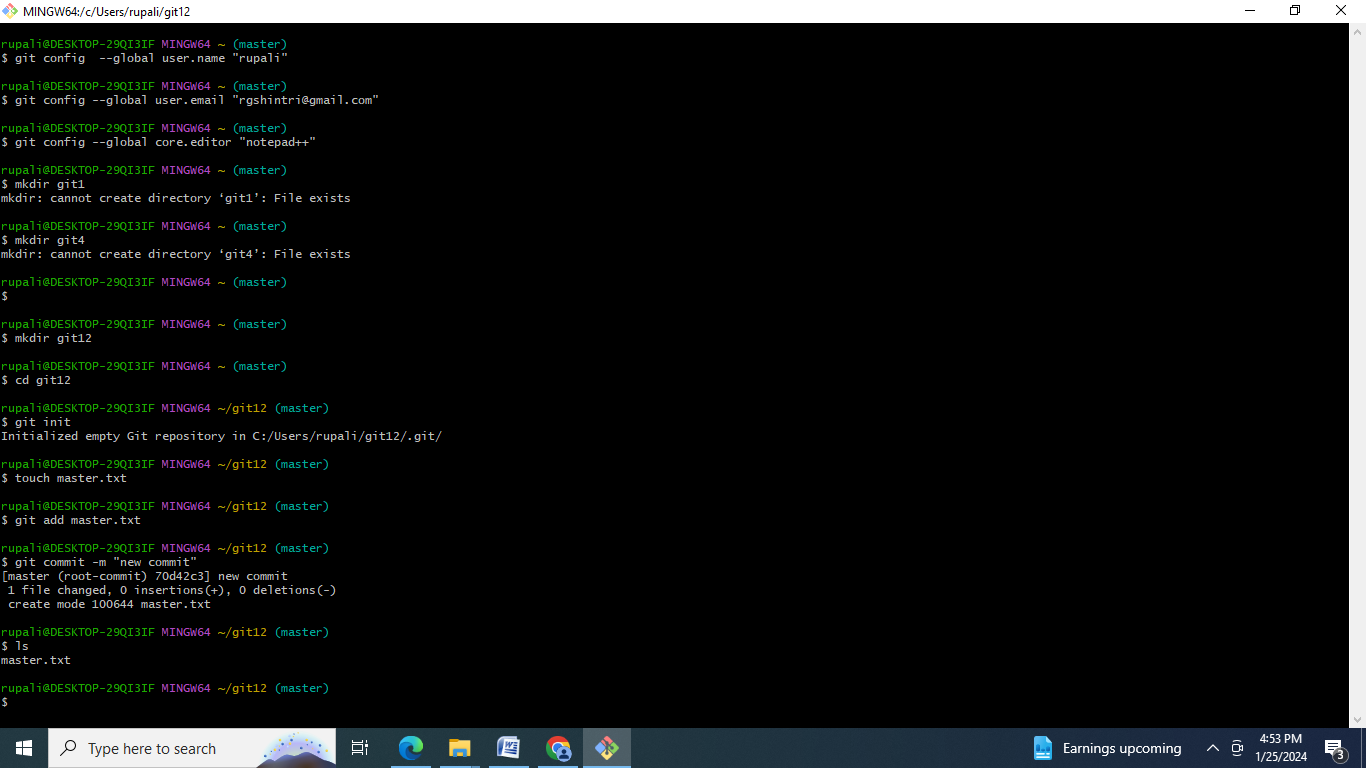
● Merge public 2 on master branch

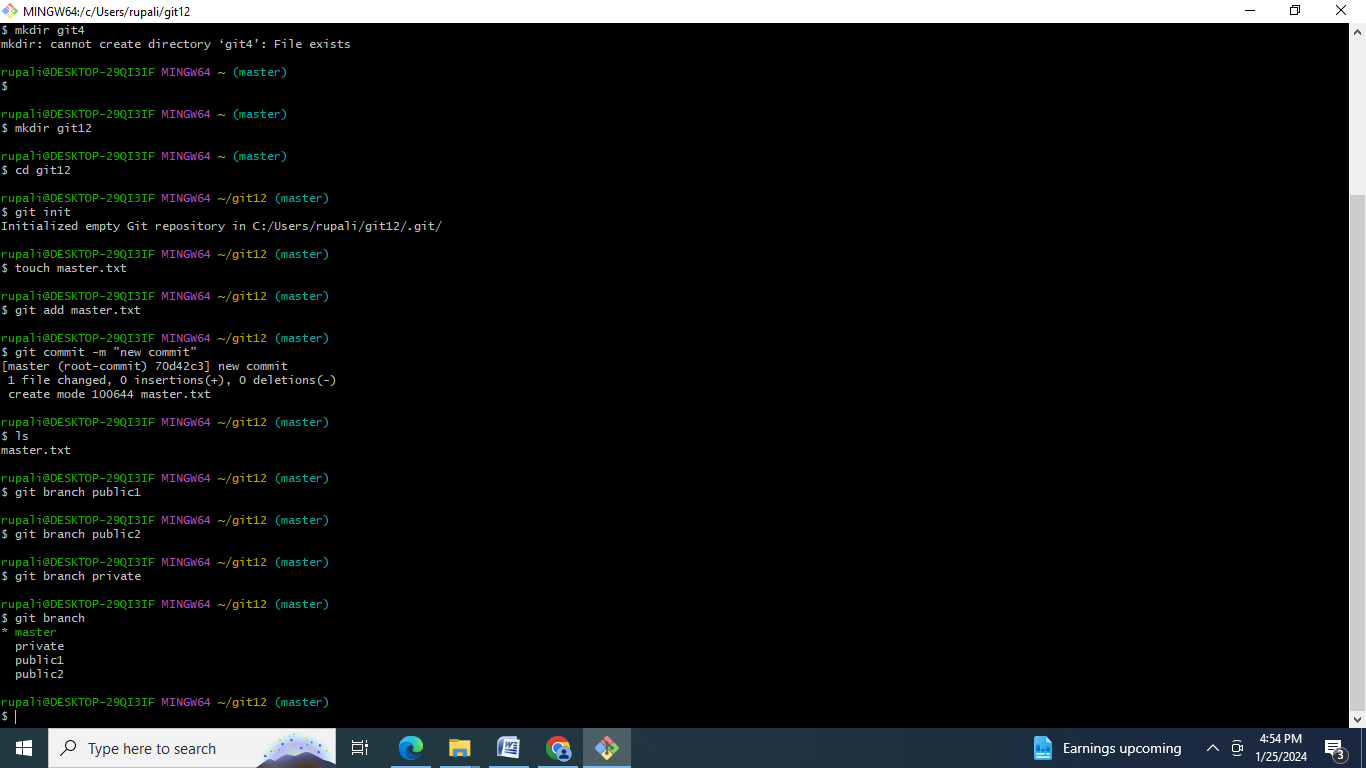
● Edit master.txt on private branch, stage and commit

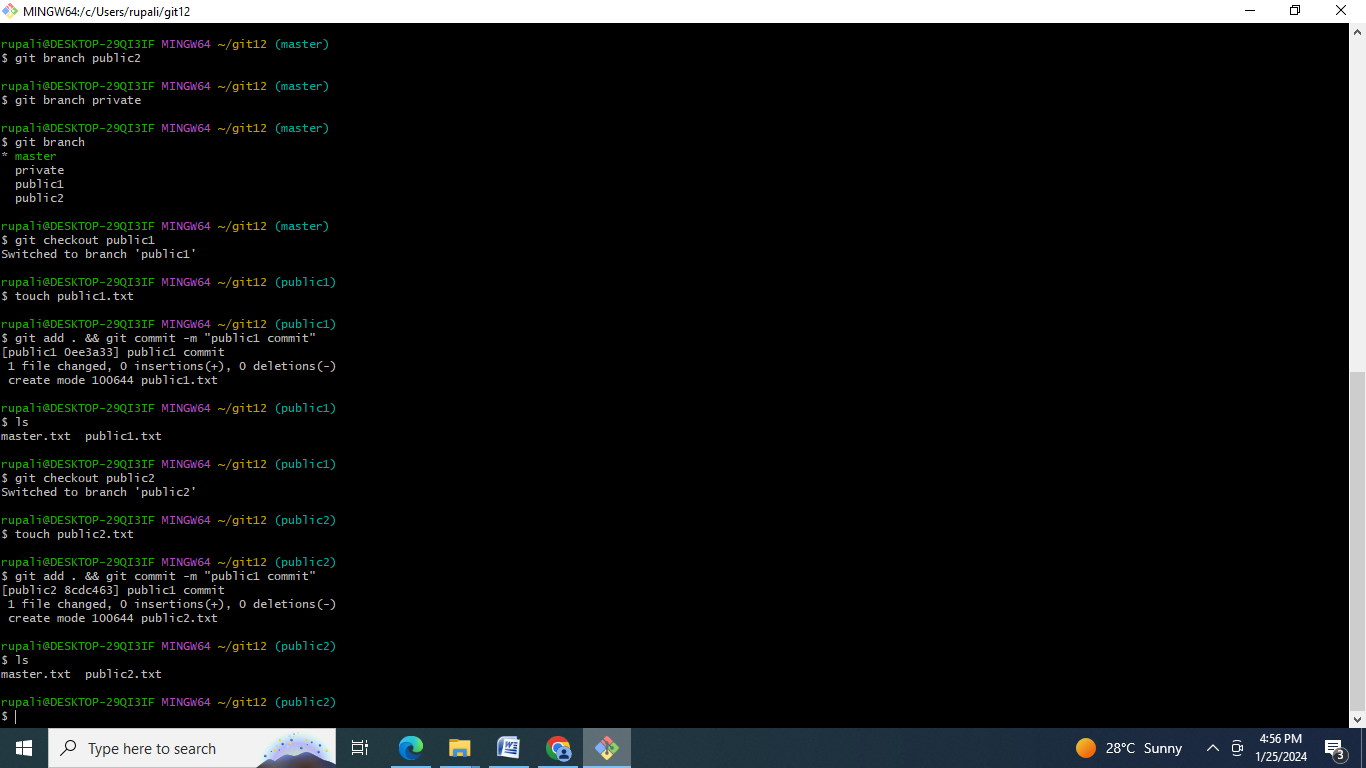
● Now update branch public 1 and public 2 with new master code in private

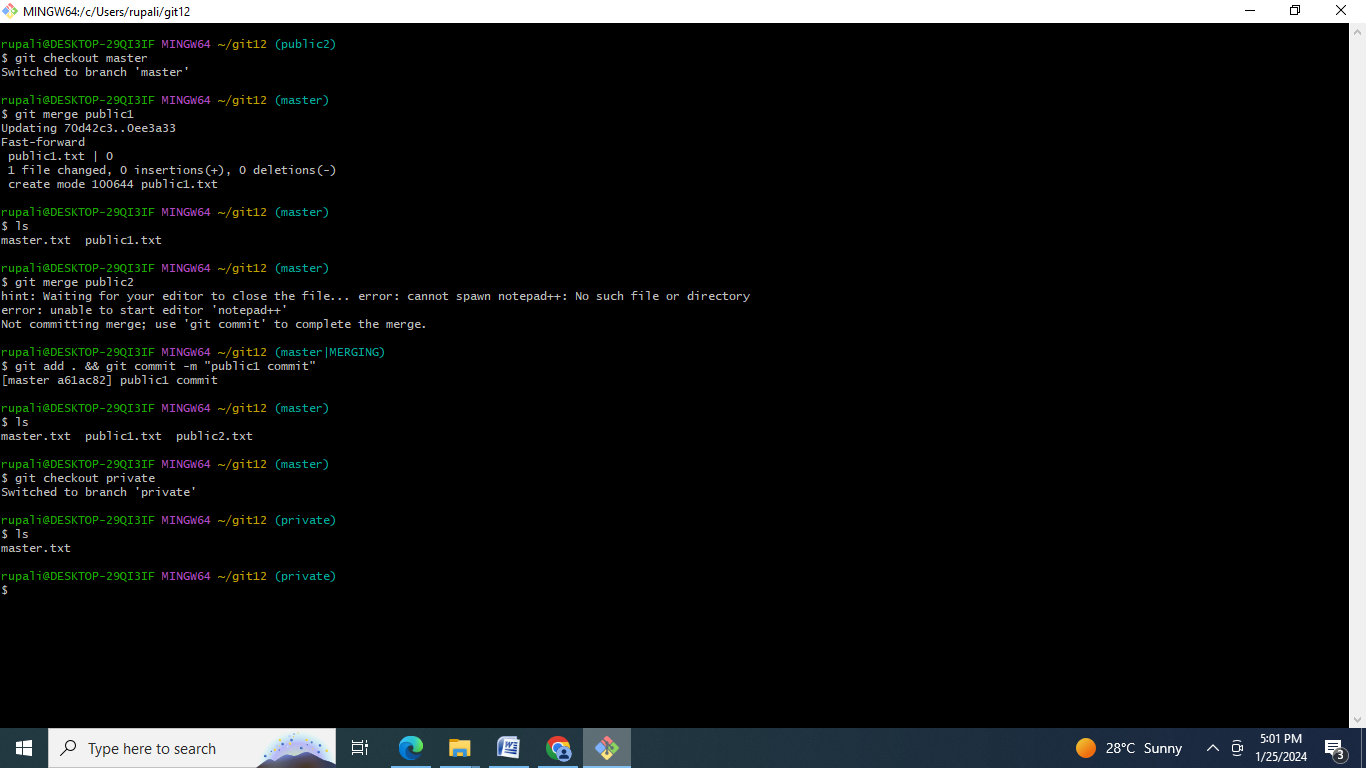
● Also update new master code on master

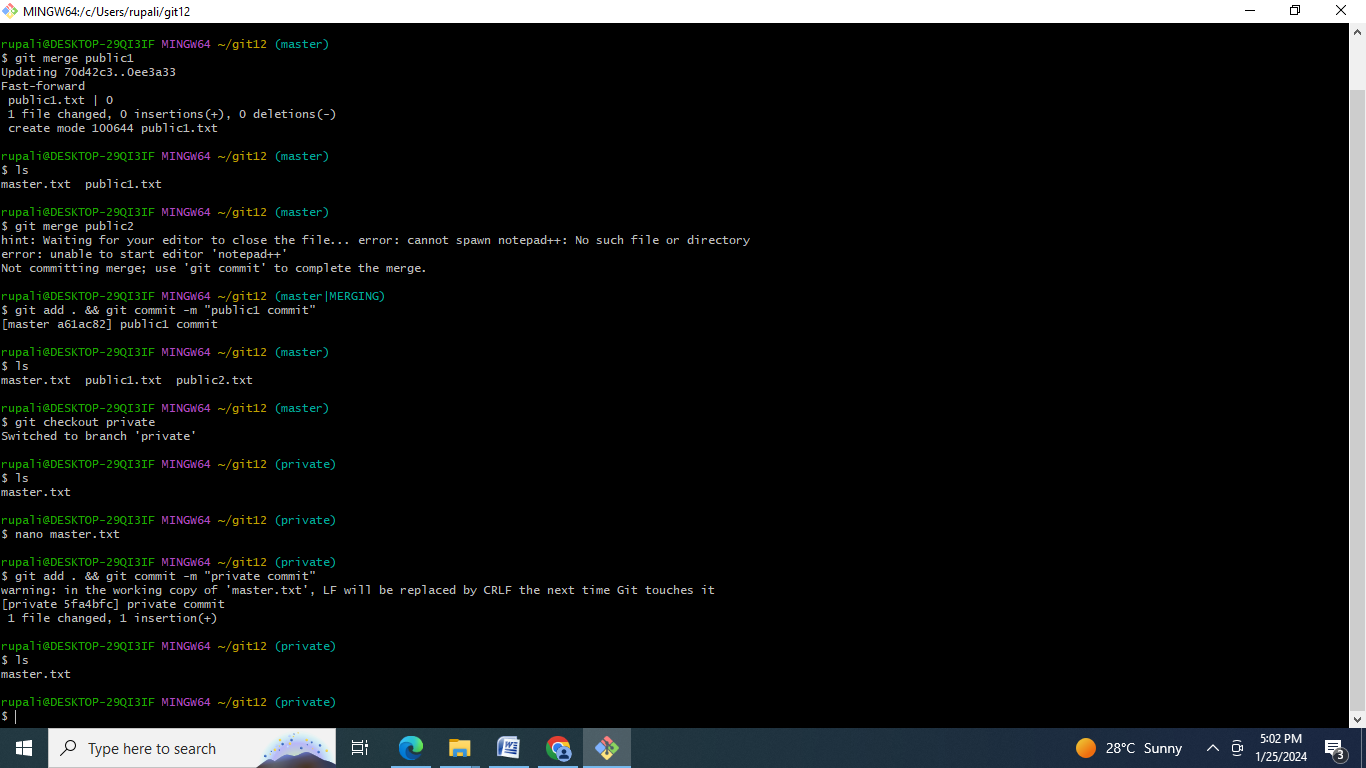
● Finally update all the code on the private branch.

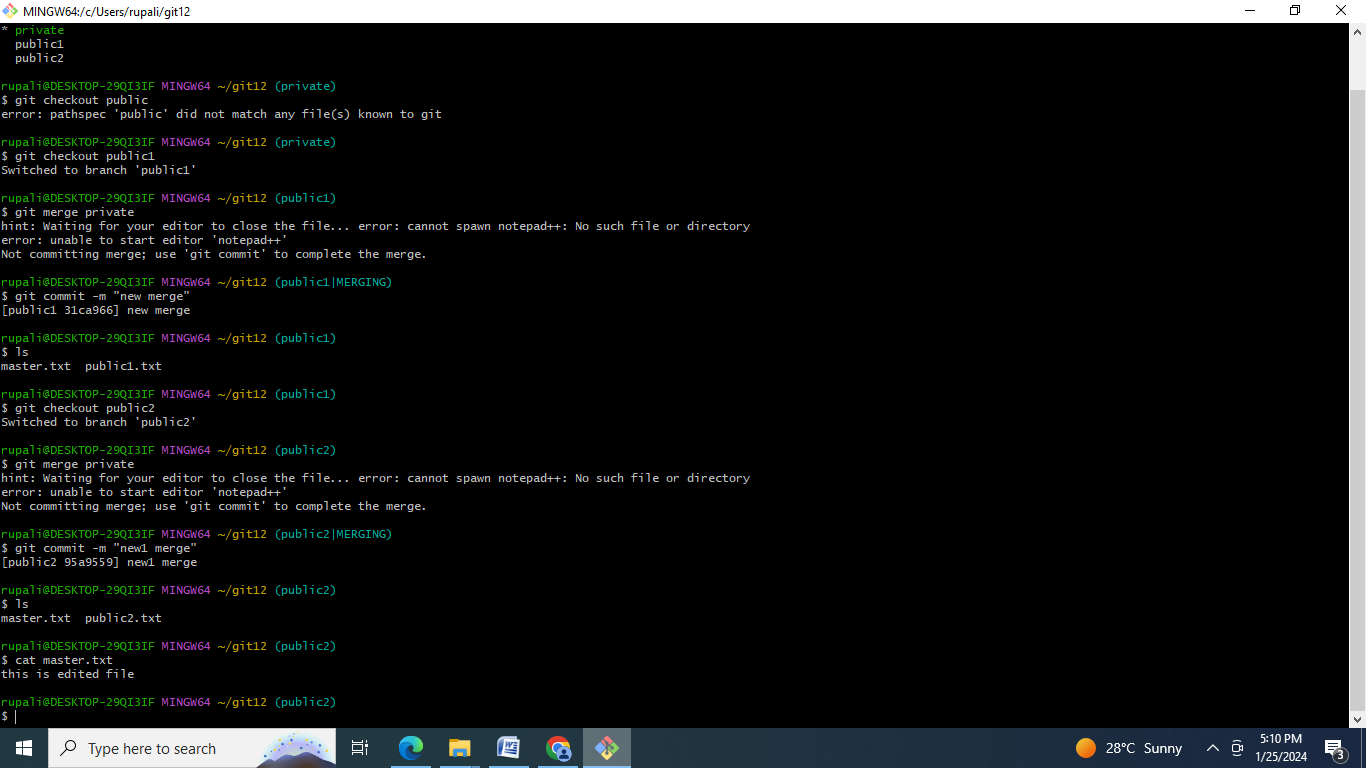


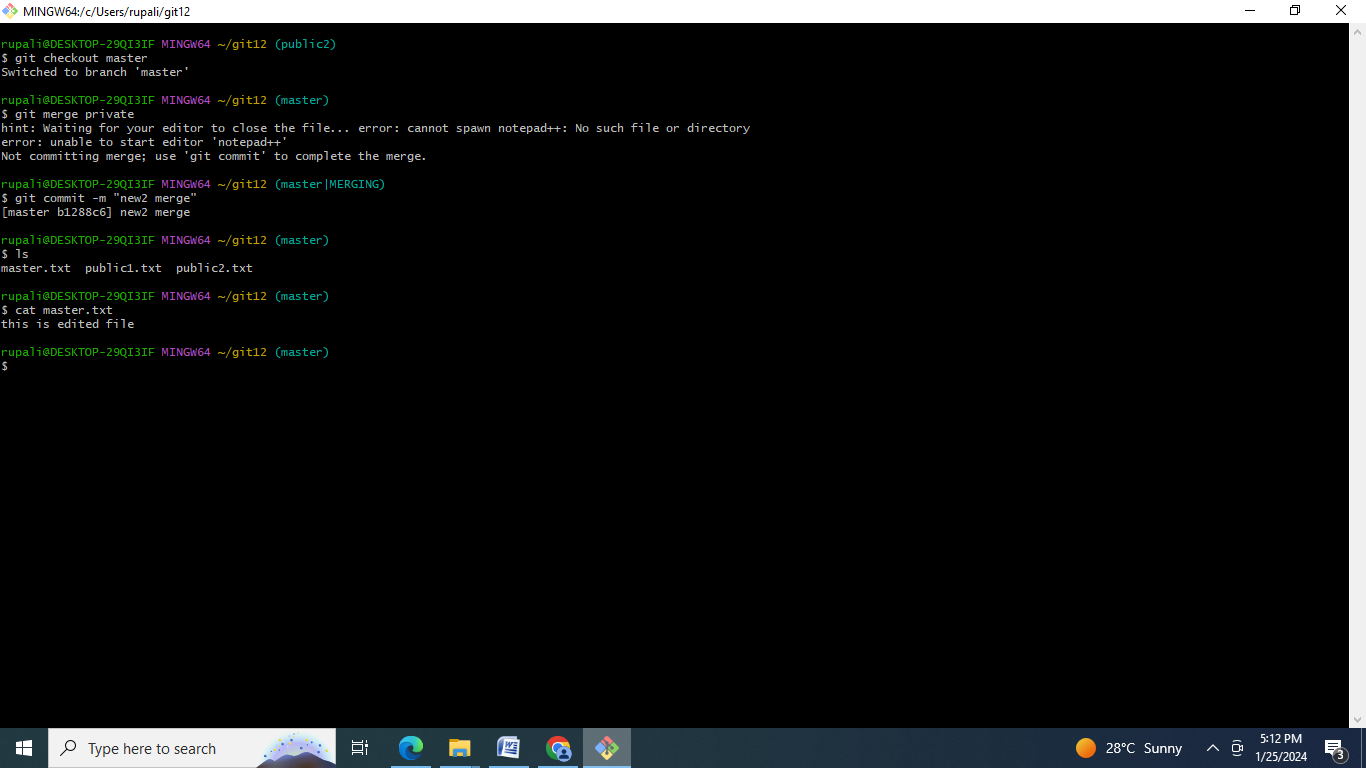












Commands:

mkdir git12

cd git12

git init

touch master.txt

git add master.txt

git commit -m "new commit"

ls

git branch public1

git branch public2

git branch private

git branch

git checkout public1

touch public1.txt

git add . && git commit -m "public1 commit"

ls

git checkout public2

touch public2.txt

git add . && git commit -m "public1 commit"

ls

clear

git checkout master

git merge public1

ls

git merge public2

git add . && git commit -m "public1 commit"

ls

git checkout private

ls

nano master.txt

git add . && git commit -m "private commit"

ls

clear

git branch

git checkout public

git checkout public1

git merge private

git commit -m "new merge"

ls

git checkout public2

git merge private

git commit -m "new1 merge"

ls

cat master.txt

clear

git checkout master

git merge private

git commit -m "new2 merge"

ls

cat master.txt

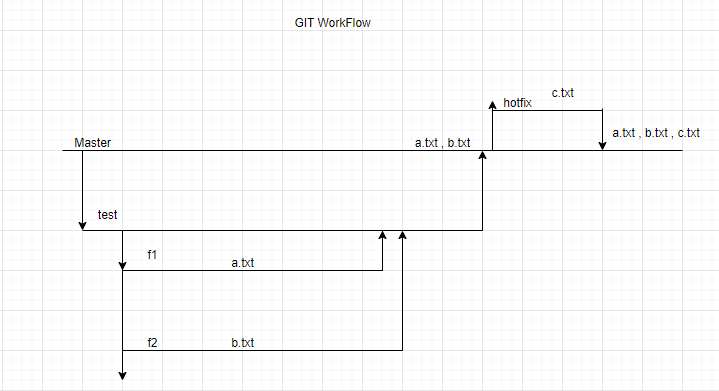
**E] GIT Assignment 5**

● Create a gitflow workflow architecture on git

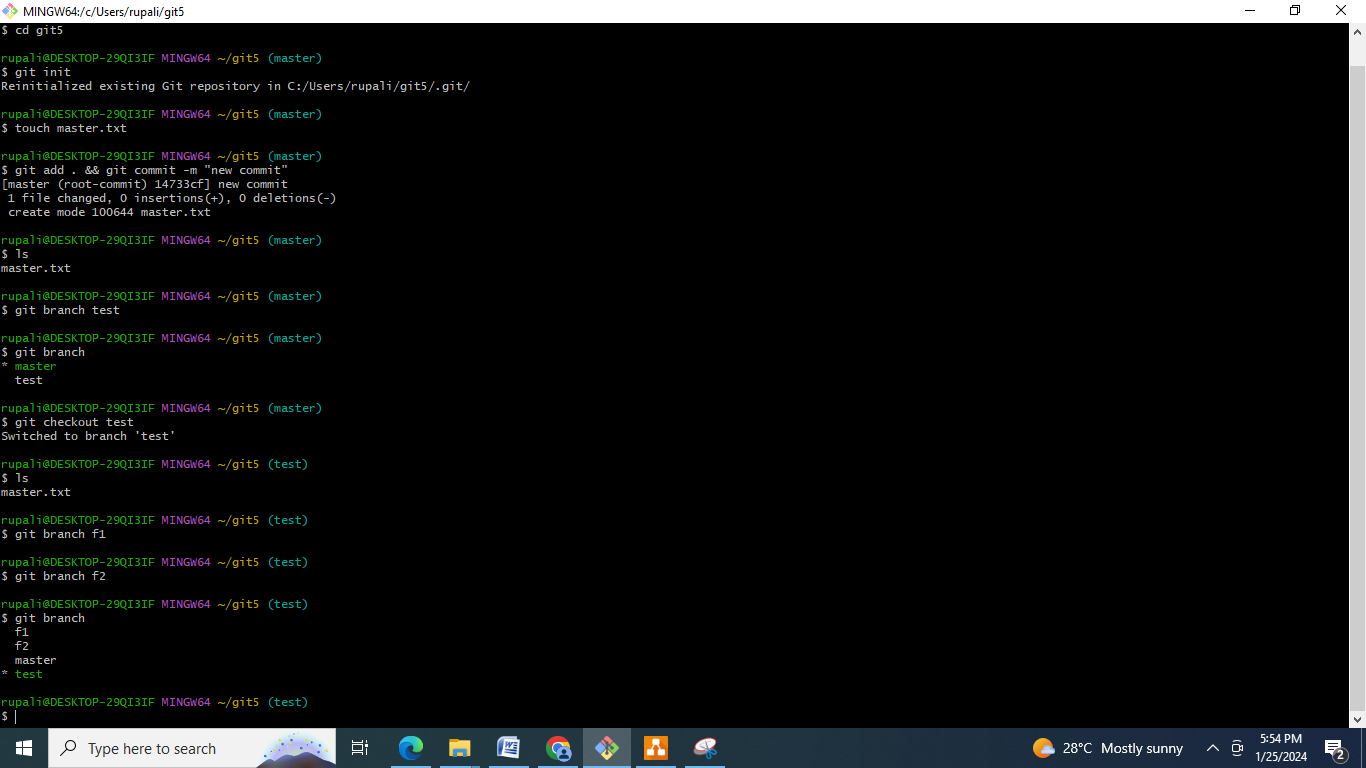
● Create all the required branches

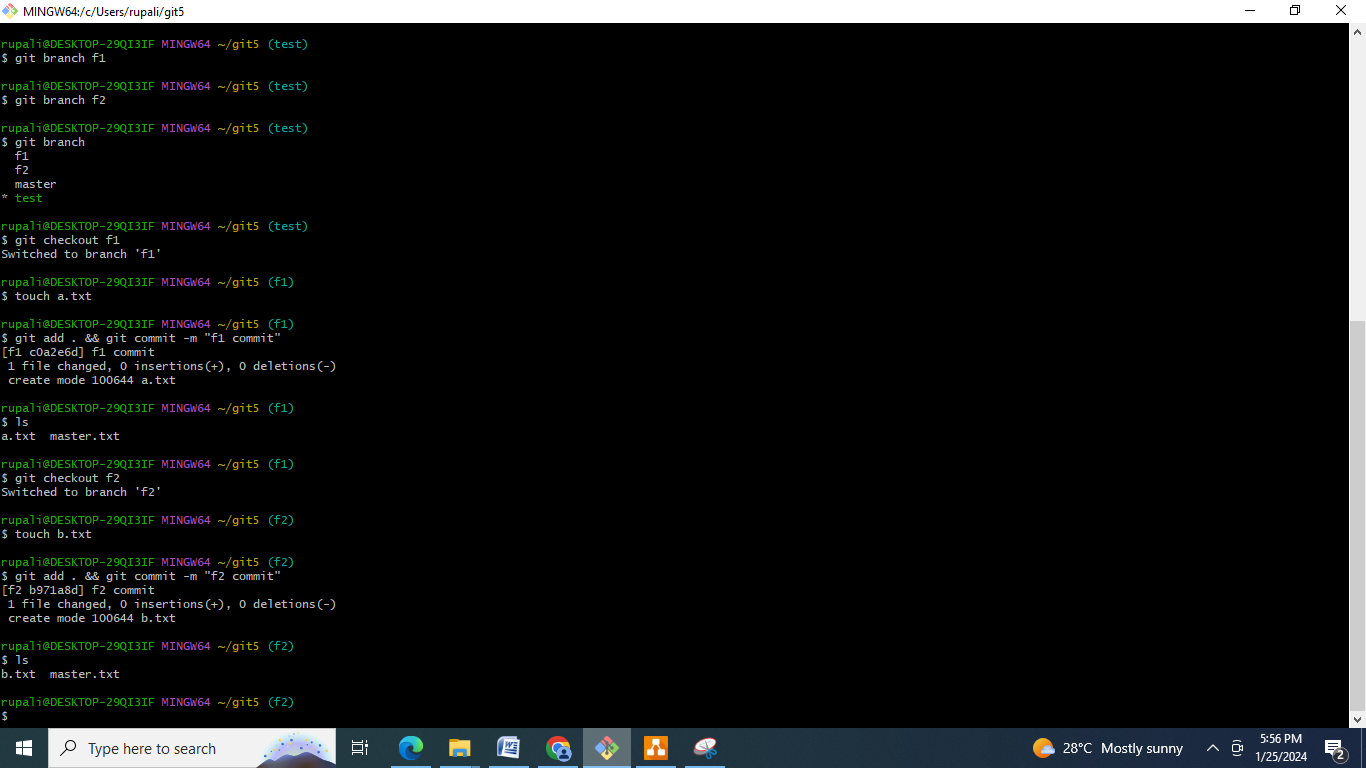
● Starting from the feature branch, push the branch to the master, following the architecture

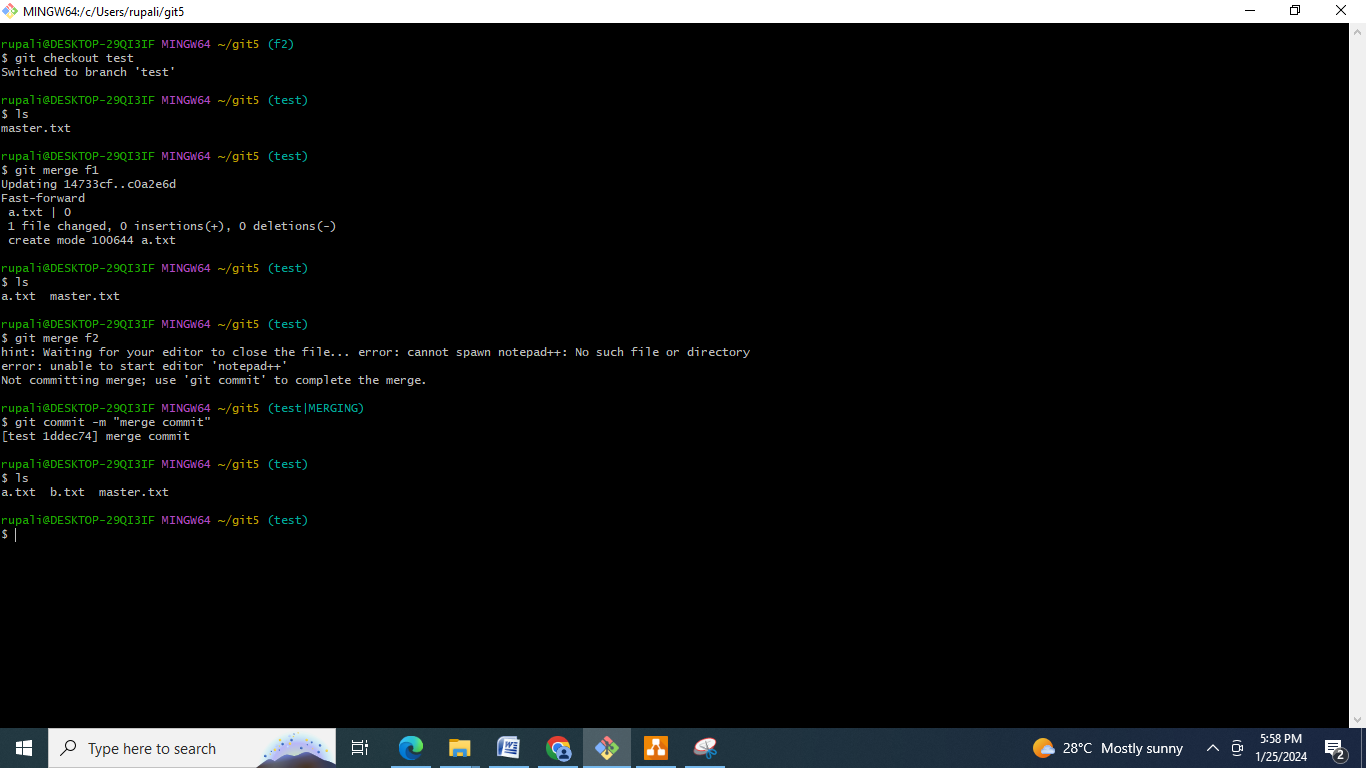
● Push a urgent.txt on master using hotfix

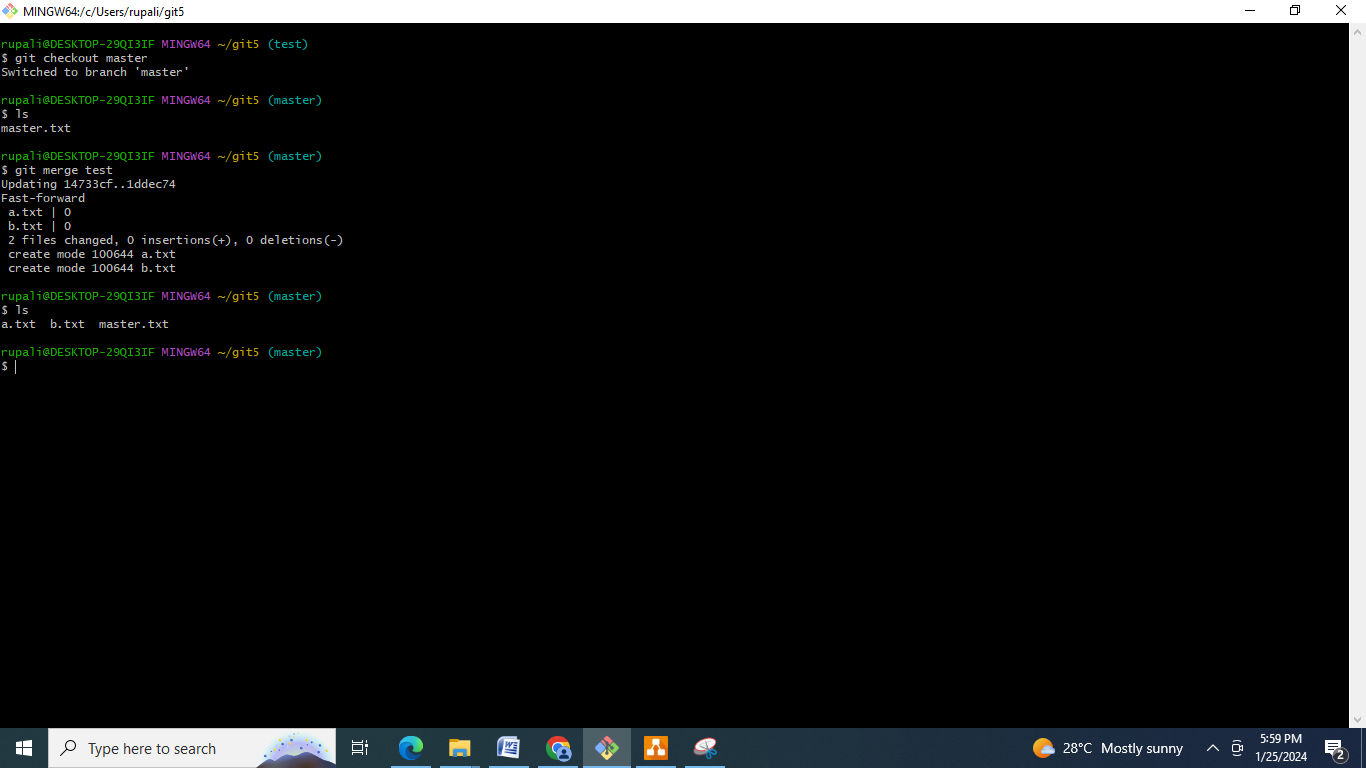


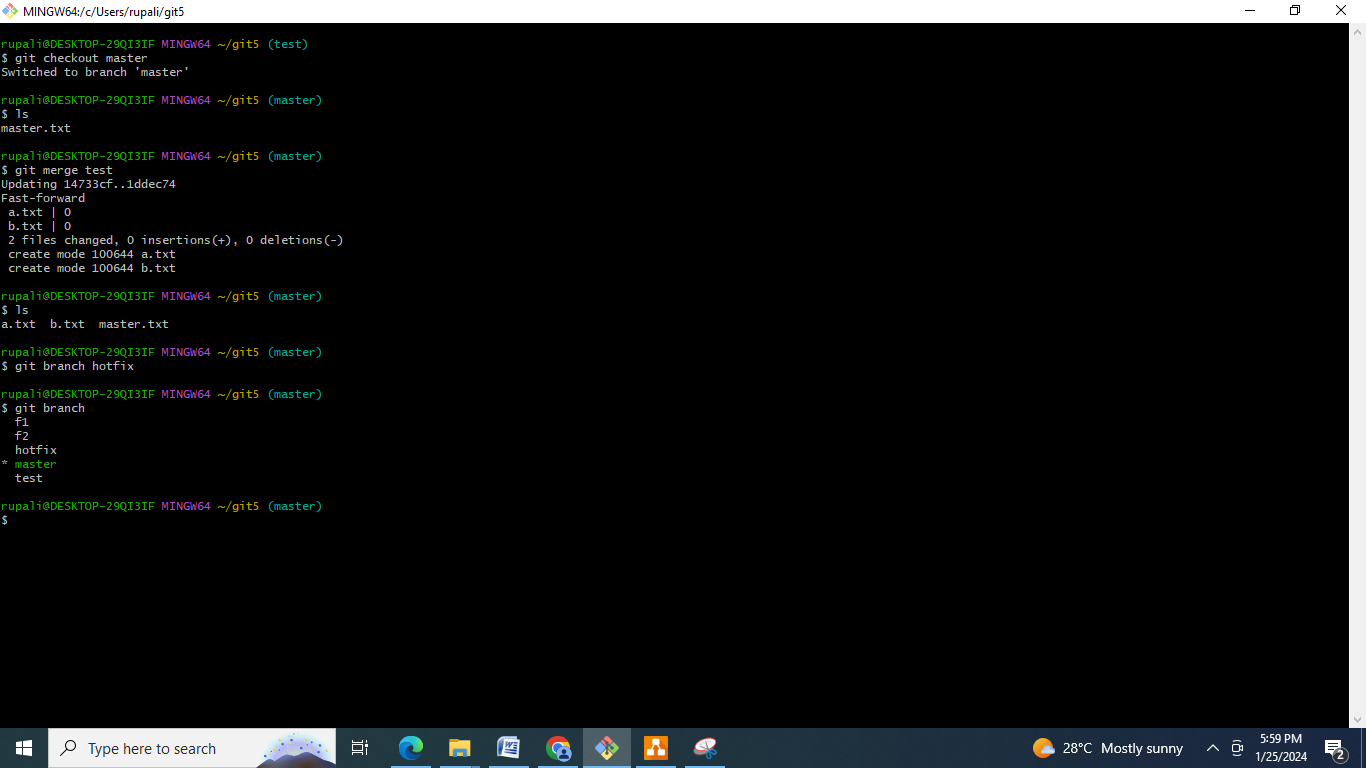
Steps:

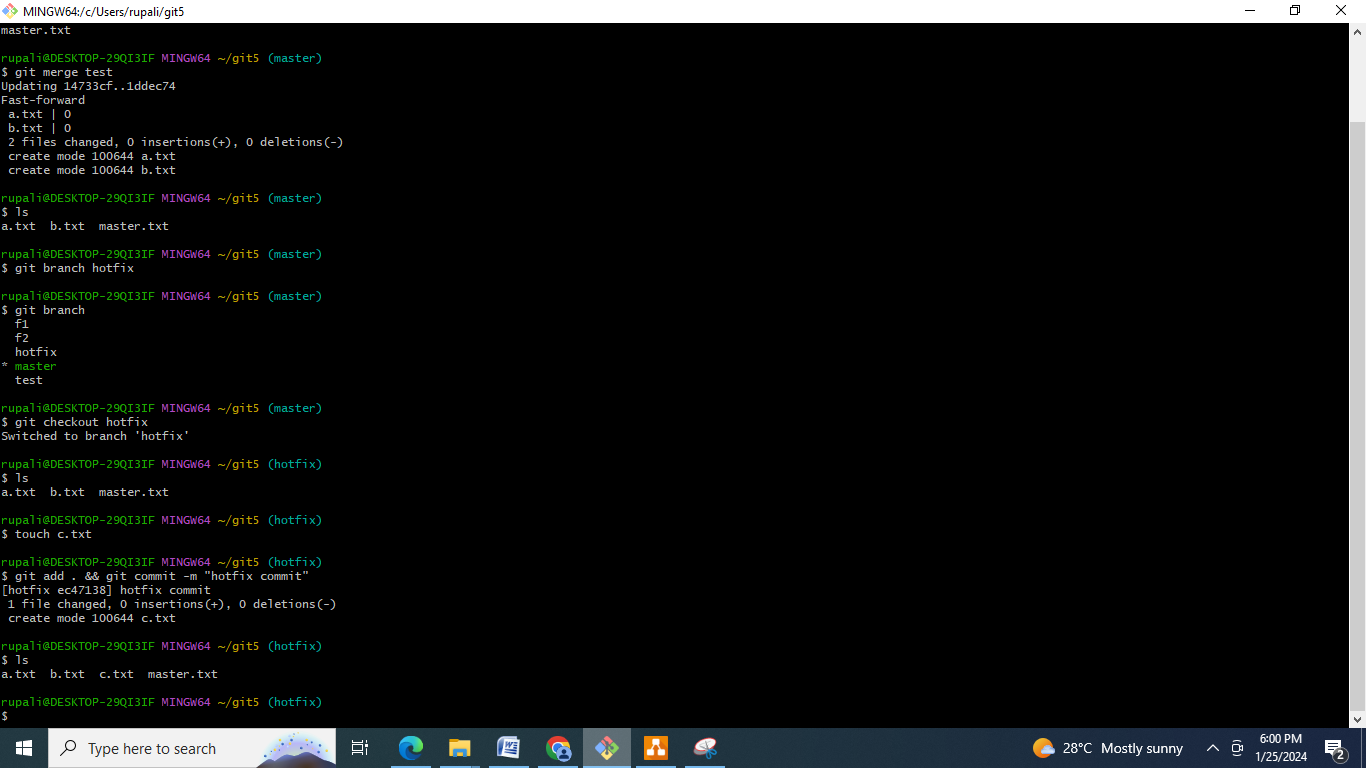


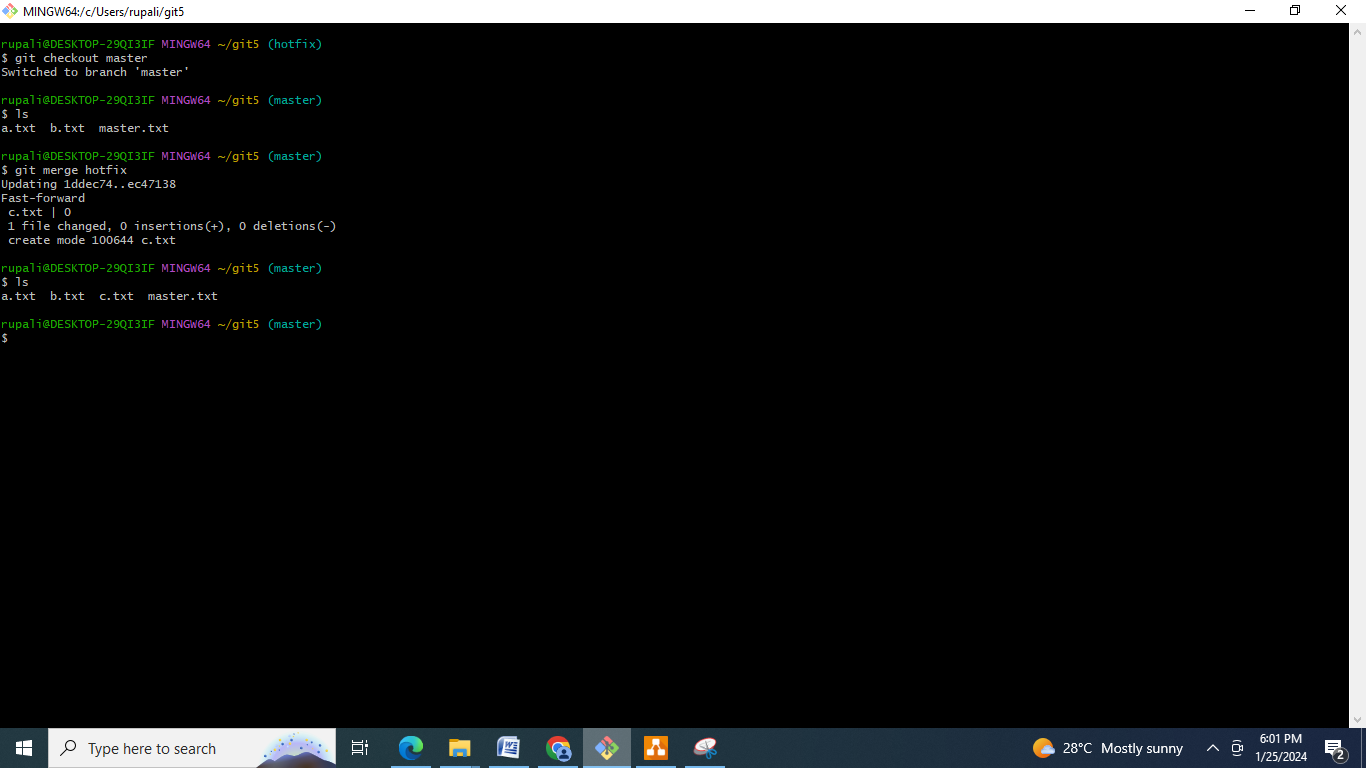












Commands:

cd git5

git init

touch master.txt

git add . && git commit -m "new commit"

ls

git branch test

git branch

git checkout test

ls

git branch f1

git branch f2

git branch

git checkout f1

touch a.txt

git add . && git commit -m "f1 commit"

ls

git checkout f2

touch b.txt

git add . && git commit -m "f2 commit"

ls

clear

git checkout test

ls

git merge f1

ls

git merge f2

git commit -m "merge commit"

ls

clear

git checkout master

ls

git merge test

ls

git branch hotfix

git branch

git checkout hotfix

ls

touch c.txt

git add . && git commit -m "hotfix commit"

ls

clear

git checkout master

ls

git merge hotfix

ls

**Conclusion :** Git is a version control system for updating the changes in working repositories locally and can be connected to remote repository in GitHub which maintains the files centrally which can be accessed publically.

