**Branching:**

1. Create a new branch **“GitNewBranch”.**
2. List all the local and remote branches available in the current trunk. Observe the “\*” mark which denote the current pointing branch.
3. Switch to the newly created branch. Add some files to it with some contents.
4. Commit the changes to the branch.
5. Check the status with **“git status”** command.

**Merging:**

1. Switch to the master
2. List out all the differences between trunk and branch. These provide the differences in command line interface.
3. List out all the visual differences between master and branch using **P4Merge tool**.
4. Merge the source branch to the trunk.
5. Observe the logging after merging using **“git log –oneline –graph –decorate”**
6. Delete the branch after merging with the trunk and observe the git status.

**Repository link:**  
<https://github.com/rishi-demo/GitDemo>

### Branching

1. **Create a new branch named** GitNewBranch**:**

bash

Copy code

git branch GitNewBranch

1. **List all local and remote branches:**

bash

Copy code

git branch -a

1. **Switch to the new branch:**

bash

Copy code

git checkout GitNewBranch

1. **Add a new file and commit changes:**

* Created a file named branchfile.txt with content:  
  this is a new branch file
* Added and committed the file:

bash

Copy code

echo "this is a new branch file" > branchfile.txt

git add branchfile.txt

git commit -m "Add branchfile.txt in GitNewBranch"

1. **Verify branch status:**

bash

Copy code

git status

Output showed “nothing to commit, working tree clean”.

### Merging

1. **Switch back to the** master **branch:**

bash

Copy code

git checkout master

1. **Check the differences between** master **and** GitNewBranch**:**

bash

Copy code

git diff master..GitNewBranch

Output showed the new file and its content difference.

1. **Merge** GitNewBranch **into** master**:**

bash

Copy code

git merge GitNewBranch

Resulted in a fast-forward merge.

1. **View commit log graph:**

bash

Copy code

git log --oneline --graph --decorate

Showed the merge commit at the top.

1. **Delete the merged branch:**

bash

Copy code

git branch -d GitNewBranch

1. **Check status:**

bash

Copy code

git status

Output showed the branch is ahead of remote by 1 commit.

1. **Push changes to remote repository:**

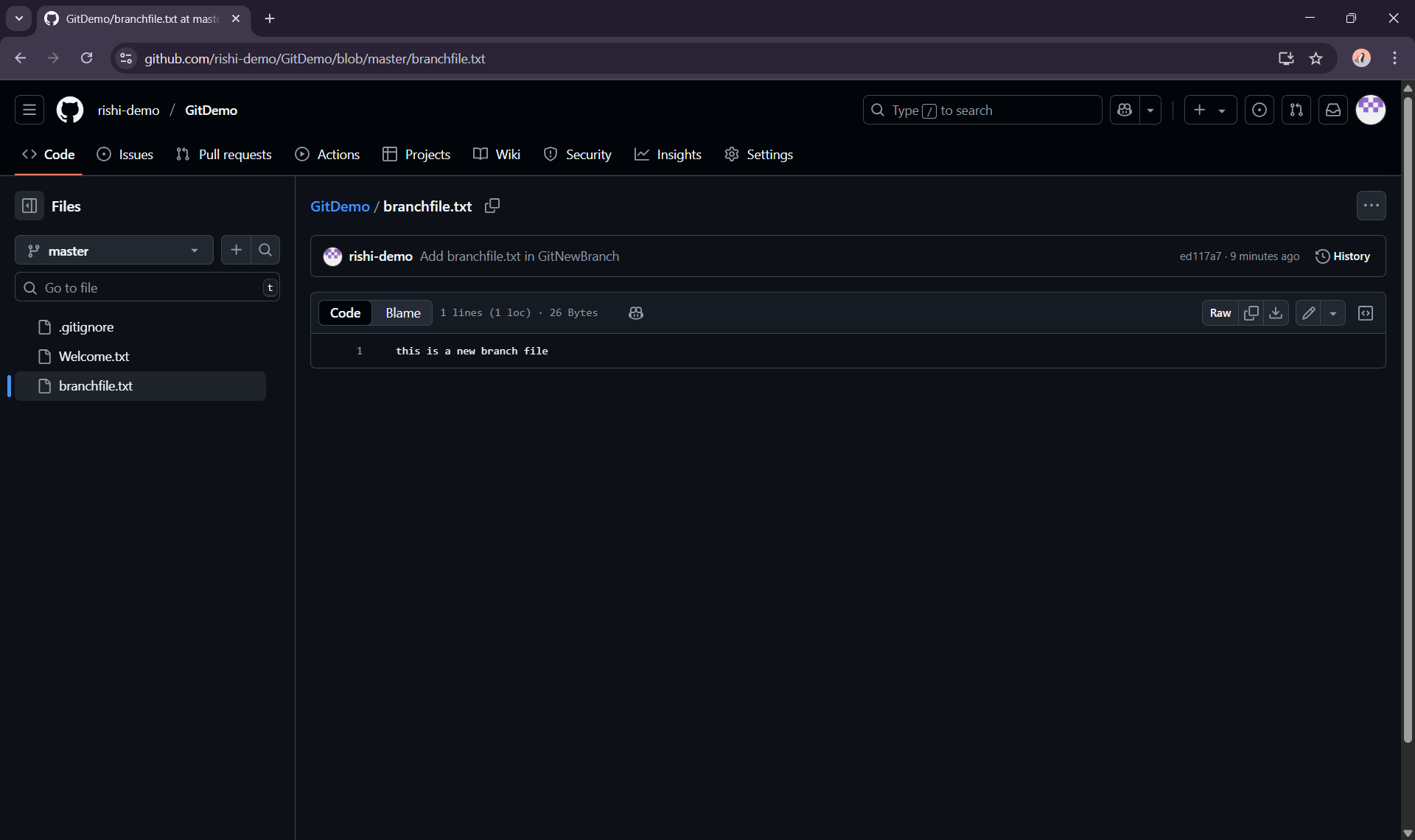
bash

Copy code

git push origin master

This updated the remote master branch on GitHub.

**OUTPUT:**

****