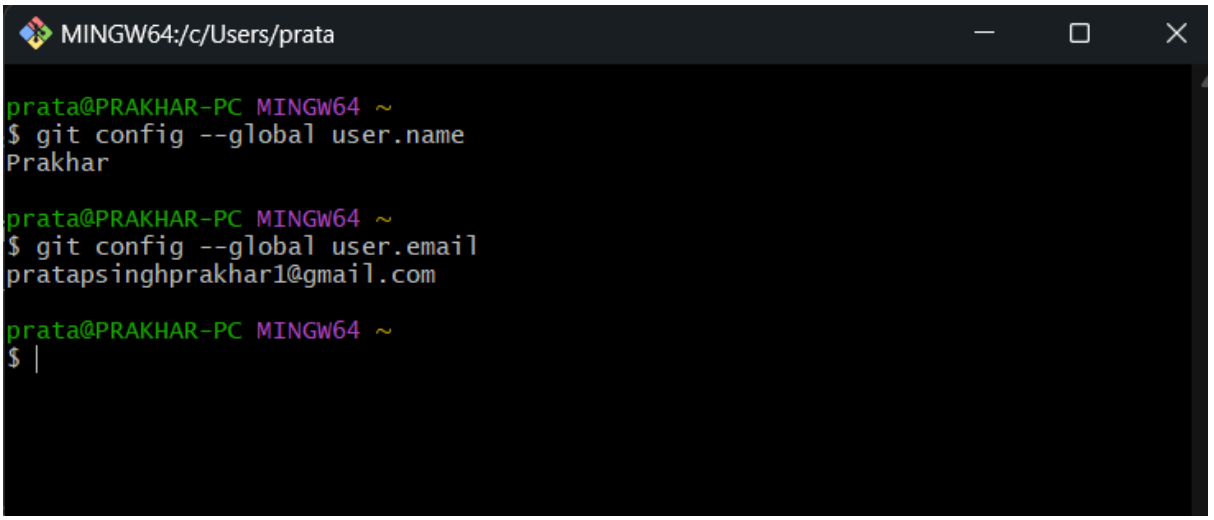
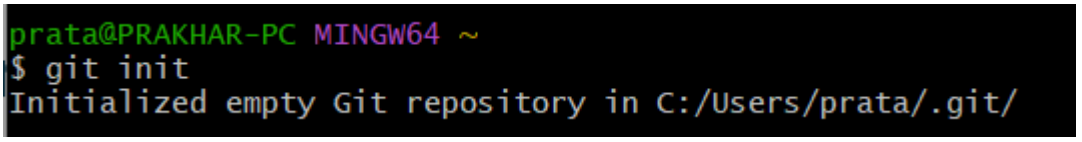
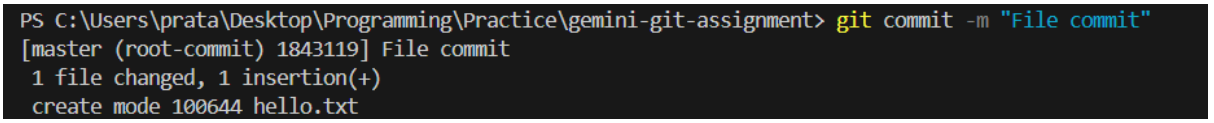
GIT Assignment Questions

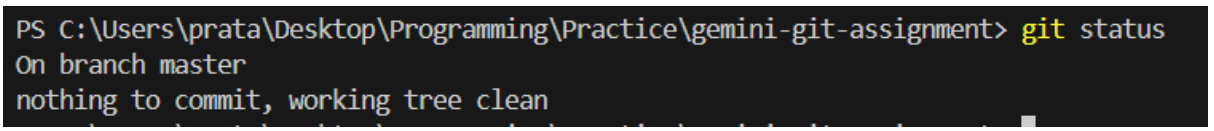
1) 

2) git --version is the required command.

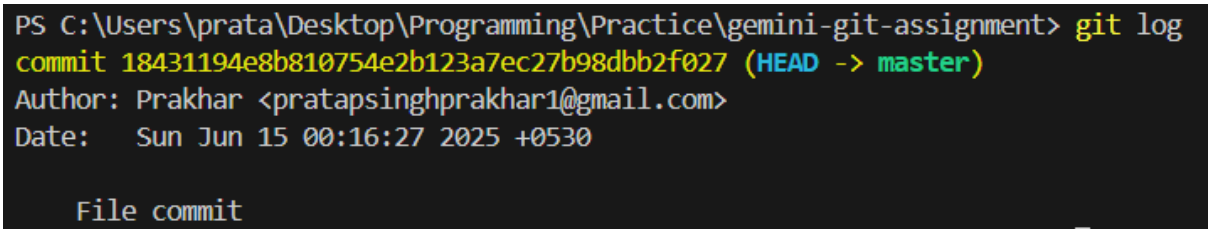
3) 

4) 

5) 

6) 

The git status command results in “On branch master” which means our repository is on the master branch, and since everything is already committed we get nothing to commit message.

7) 

The git log command provides the commit history. It provides details like the

author, and date.

8) Branching in a git is helpful especially when a large number of people work parallely on different features in a repository. Branches help keeping different features on different features to avoid clashes and already existing features affecting due to other ones.

9)
```
PS C:\Users\prata\Desktop\Programming\Practice\gemini-git-assignment> git switch -c feature-branch
Switched to a new branch 'feature-branch'
PS C:\Users\prata\Desktop\Programming\Practice\gemini-git-assignment>
```

This command creates and switches in a single attempt. The other traditional way is to first do git branch branch-name and then git checkout branch-name.

10)
```
PS C:\Users\prata\Desktop\Programming\Practice\gemini-git-assignment> git checkout feature-branch
Switched to branch 'feature-branch'
PS C:\Users\prata\Desktop\Programming\Practice\gemini-git-assignment> git add .
PS C:\Users\prata\Desktop\Programming\Practice\gemini-git-assignment> git commit -m "New file"
[feature-branch d8a8aa3] New file
 1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\prata\Desktop\Programming\Practice\gemini-git-assignment> git checkout main
Switched to branch 'main'
PS C:\Users\prata\Desktop\Programming\Practice\gemini-git-assignment>
```

11)
```
PS C:\Users\prata\Desktop\Programming\Practice\gemini-git-assignment> git merge feature-branch
Updating 1843119..d8a8aa3
Fast-forward
 hello.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

The command used is git merge feature-branch. If there are no conflicts then both the branches get merged without any issue.

12) A merge conflict is a situation in which different branches have edited the same file, hence while merging it becomes unclear to chose which version to keep.

```
 create mode 100644 Practice/story.txt
PS C:\Users\prata\Desktop\Programming\Practice> git checkout -b feature/hero-branch
Switched to a new branch 'feature/hero-branch'
PS C:\Users\prata\Desktop\Programming\Practice> echo "The hero lights a torch to see in dark cave" > story.txt
PS C:\Users\prata\Desktop\Programming\Practice> git add story.txt
PS C:\Users\prata\Desktop\Programming\Practice> git commit -m "Feature : Hero lights a torch"
[feature/hero-branch d5432e6] Feature : Hero lights a torch
 1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\prata\Desktop\Programming\Practice> git checkout main
Switched to branch 'main'
PS C:\Users\prata\Desktop\Programming\Practice> git checkout -b feature/villain-action
Switched to a new branch 'feature/villain-action'
PS C:\Users\prata\Desktop\Programming\Practice> echo "The villain springs a tap in the dark cave!" > story.txt
PS C:\Users\prata\Desktop\Programming\Practice> git add story.txt
PS C:\Users\prata\Desktop\Programming\Practice> git commit -m "Feature : Villain springs a trap"
[feature/villain-action 672c441] Feature : Villain springs a trap
 1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\prata\Desktop\Programming\Practice> git checkout main
Switched to branch 'main'
PS C:\Users\prata\Desktop\Programming\Practice> git merge feature/hero-branch
Updating 583665f..d5432e6
Fast-forward
 Practice/story.txt | Bin 64 -> 92 bytes
 1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\prata\Desktop\Programming\Practice> git merge feature/villain-action
warning: Cannot merge binary files: Practice/story.txt (HEAD vs. feature/villain-action)
Auto-merging Practice/story.txt
CONFLICT (content): Merge conflict in Practice/story.txt
Automatic merge failed; fix conflicts and then commit the result.
PS C:\Users\prata\Desktop\Programming\Practice>
```

To resolve merge conflicts, one needs to manually verify and fix the mismatch.

13) A remote repository is a version of your project that is hosted on the internet. It allows collaboration and acts a central backup of your code. Unlike the local repository, which exists on one's computer, it's cloud based.

14)

```
PS C:\Users\prata\Desktop\Programming\Practice> git clone https://github.com/prakhar5100/Momentum.git
Cloning into 'Momentum'...
remote: Enumerating objects: 120, done.
remote: Counting objects: 100% (120/120), done.
remote: Compressing objects: 100% (98/98), done.
remote: Total 120 (delta 33), reused 97 (delta 18), pack-reused 0 (from 0)
Receiving objects: 100% (120/120), 147.79 KiB | 738.00 KiB/s, done.
Resolving deltas: 100% (33/33), done.
PS C:\Users\prata\Desktop\Programming\Practice>
```

The URL is https://github.com/prakhar5100/Momentum.git

15)

```
PS C:\Users\prata\Desktop\Programming\Practice\Momentum> git add .
PS C:\Users\prata\Desktop\Programming\Practice\Momentum> git commit -m "Change Readme"
[main dfa3c23] Change Readme
 1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\prata\Desktop\Programming\Practice\Momentum>
```

16) git push origin main is used for pushing the code to the main branch.

```
PS C:\Users\prata\Desktop\Programming\Practice\Momentum> git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 305 bytes | 305.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/prakhar5100/Momentum.git
   3b7381d..dfa3c23  main -> main
PS C:\Users\prata\Desktop\Programming\Practice\Momentum>
```

17) git fetch command is used to fetch the latest changes from the remote repository. However, it doesn't apply them to the local repository. It simply fetches the change. Git pull however, also merges the changes from the remote repository to the local repository.

18) We can use git commit --amend to edit the previous commit messages.