Daily Learning and Assessment Report

< Git Advance>

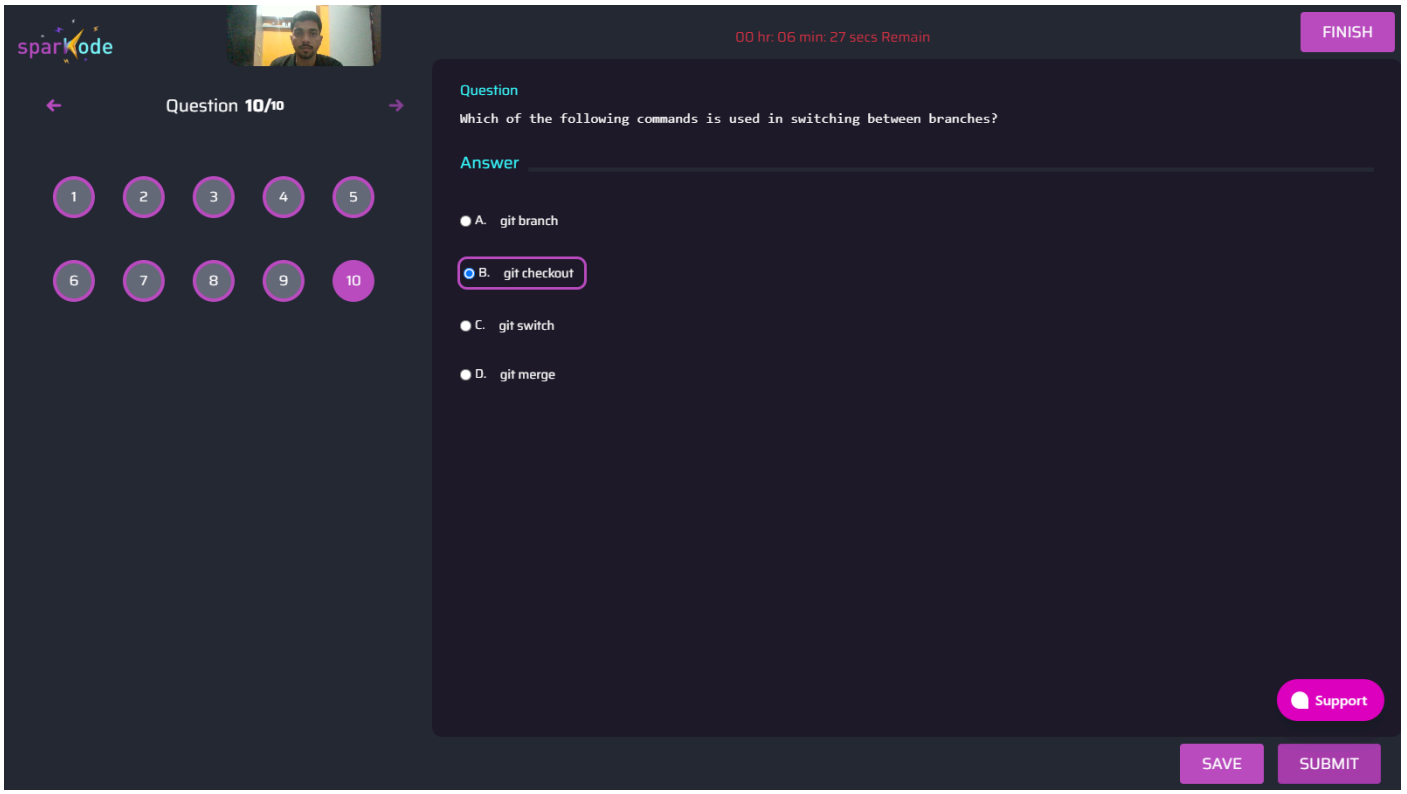
Friday 03/01/2025

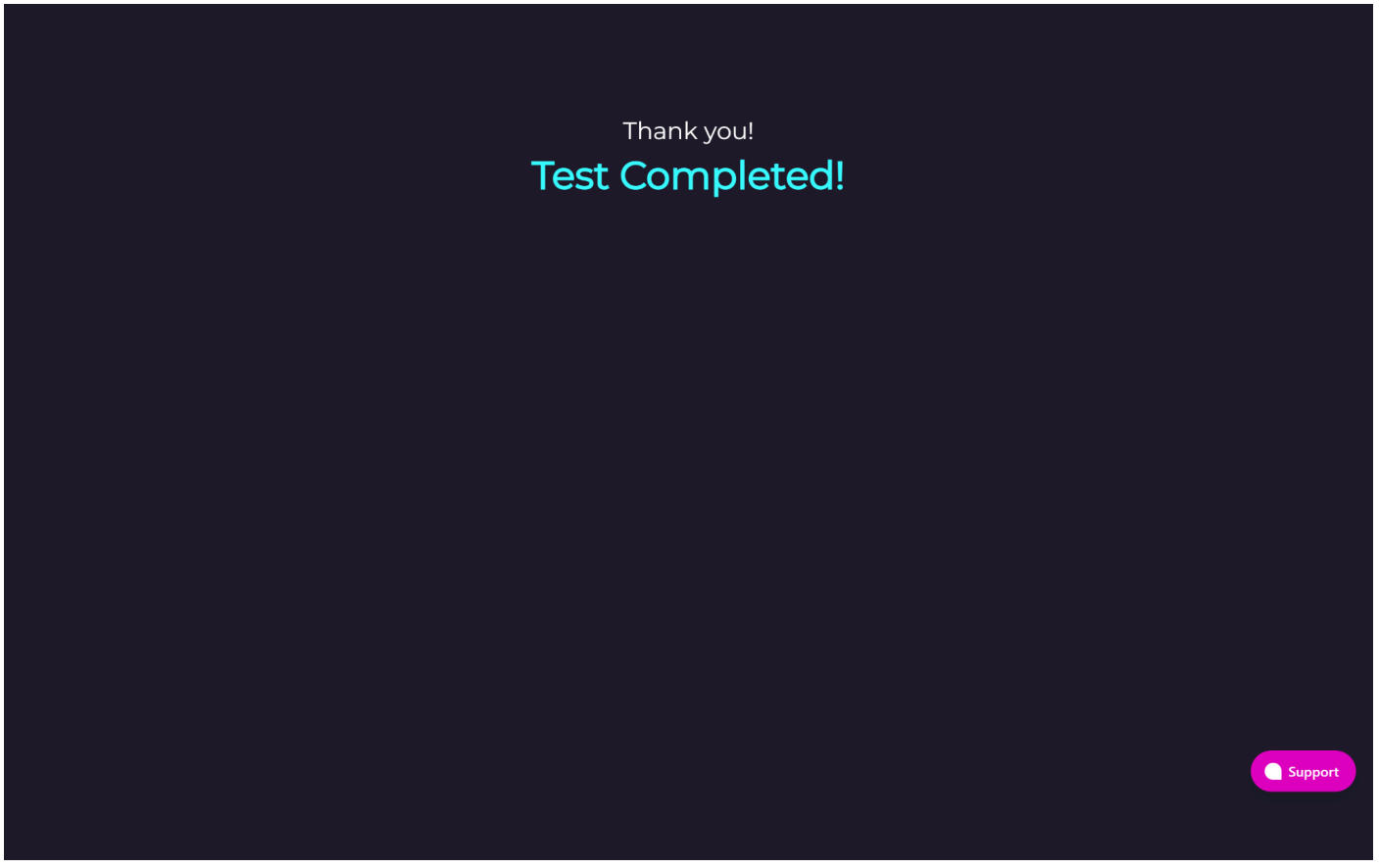
Session Guides: Ajinkya Karanjikar Chetan Satpute Mentor: Akansha kumari

Made by Sahil Bheke

Table of Contents

SparKode Assessment



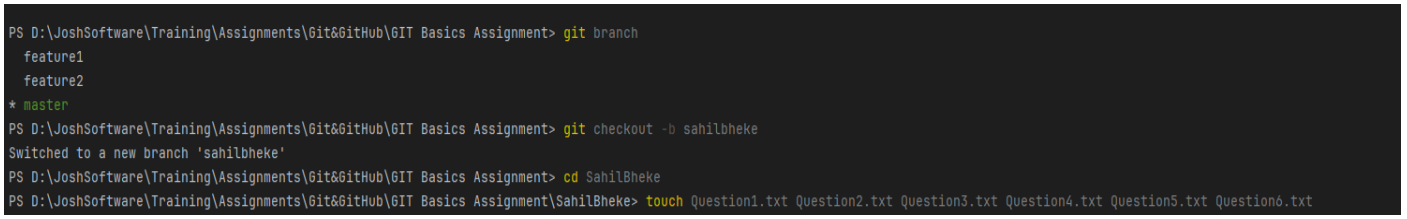


Assignment Questions

01. Creating branch and folder

```
git branch sahilbheke or git checkout -b sahilbheke
```

```
cd sahilbheke
```



02. What happens when a file is deleted in the main branch and updated in a feature branch, and how is this handled during a merge?

Ans) Conflict occurs and the situation is flagged as a "modify/delete" conflict.
Conflict can be resolved by:
1) Giving precedence to the deletion in the main branch.
2) Preserving updated file in the feature branch.

In my case git preserved the updated file in the feature branch.
As git assumed that the modification done in the feature branch is more recent.

```
PS D:\JoshSoftware\Training\Assignments\Git&GitHub\GIT Basics Assignment> git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)
PS D:\JoshSoftware\Training\Assignments\Git&GitHub\GIT Basics Assignment> git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
PS D:\JoshSoftware\Training\Assignments\Git&GitHub\GIT Basics Assignment> git merge feature3
CONFLICT (modify/delete): Feature_1.2.txt deleted in HEAD and modified in feature3.  Version feature3 of Feature_1.2.txt left in tree.
CONFLICT (modify/delete): Feature_2.1.txt deleted in HEAD and modified in feature3.  Version feature3 of Feature_2.1.txt left in tree.
Automatic merge failed; fix conflicts and then commit the result.
PS D:\JoshSoftware\Training\Assignments\Git&GitHub\GIT Basics Assignment> git branch
  feature1
  feature2
  feature3
* master
  sahilbheke
PS D:\JoshSoftware\Training\Assignments\Git&GitHub\GIT Basics Assignment> git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Changes to be committed:
        new file:   SahilBheke/Question1.txt
        new file:   SahilBheke/Question2.txt
        new file:   SahilBheke/Question3.txt
        new file:   SahilBheke/Question4.txt
        new file:   SahilBheke/Question5.txt
        new file:   SahilBheke/Question6.txt

Unmerged paths:
  (use "git add/rm <file>..." as appropriate to mark resolution)
        deleted by us:   Feature_1.2.txt
        deleted by us:   Feature_2.1.txt
```

03. How to set multiple remote repositories for the same project? Explain the use case.

Ans) We can set multiple remote by using below command.

```
git remote add <remote-name> <remote-url>
```

For example:

```
git remote add origin  https://github.com/SahilBheke25/GitAssignment.git
```

```
git remote add gitlab https://gitlab.com/trainee4052801/GitAssignment.git
```

We can also check the remote available by using following command.

```
git remote -v
```

```
PS D:\JoshSoftware\Training\Assignments\Git&GitHub\GIT Basics Assignment> git remote -v
gitlab  https://gitlab.com/trainee4052801/GitAssignment.git (fetch)
gitlab  https://gitlab.com/trainee4052801/GitAssignment.git (push)
origin  https://github.com/SahilBheke25/GitAssignment.git (fetch)
origin  https://github.com/SahilBheke25/GitAssignment.git (push)
```

Use case:

1) Primary remote hosted on GitHub(origin) can be used as open-source contributions.
   Secondary remote hosted on GitLab(GitLab) can be used for internal backups and testing.
2) Backup: If one remote is unavailable, we have a complete mirror of the repository on different platform.
3) Collaboration: Some collaborators might prefer different platforms.

04. What does the git `ls-files` command do, and what information does its output provide regarding the state of files in a Git repository?

Ans) git ls-files give the list of all the files which are tracked by the git in that branch such that files which are staged or committed. Except the deleted files which are staged. It also list the subdirectories and files.

If a file is created and not staged it will not show that file.
If a file is created and staged it will show the file.
If a file is deleted and not staged it will show the file.
If a file is deleted and staged it will not show the file.

Whereas, git status only shows the modified, deleted and new files which are not committed.

```
git ls-status
```

```
PS D:\JoshSoftware\Training\Assignments\Git&GitHub\GIT Basics Assignment> git ls-files
.gitignore
Feature_1.2.txt
Feature_1.txt
Feature_2.1.txt
Feature_2.txt
File1/SahilBheke.txt
File2/inputfile.txt
SahilBheke/Question1.txt
SahilBheke/Question2.txt
SahilBheke/Question3.txt
SahilBheke/Question4.txt
SahilBheke/Question5.txt
SahilBheke/Question6.txt
PS D:\JoshSoftware\Training\Assignments\Git&GitHub\GIT Basics Assignment> git add .
PS D:\JoshSoftware\Training\Assignments\Git&GitHub\GIT Basics Assignment> git ls-files
.gitignore
Feature3.txt
Feature_1.2.txt
Feature_1.txt
Feature_2.1.txt
Feature_2.txt
File1/SahilBheke.txt
File2/inputfile.txt
SahilBheke/Question1.txt
SahilBheke/Question2.txt
SahilBheke/Question3.txt
SahilBheke/Question4.txt
SahilBheke/Question5.txt
SahilBheke/Question6.txt
PS D:\JoshSoftware\Training\Assignments\Git&GitHub\GIT Basics Assignment> git commit -m "add feature3"
[feature3 b928a54] add feature3
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Feature3.txt
PS D:\JoshSoftware\Training\Assignments\Git&GitHub\GIT Basics Assignment> git ls-files
.gitignore
Feature3.txt
Feature_1.2.txt
Feature_1.txt
Feature_2.1.txt
Feature_2.txt
File1/SahilBheke.txt
```

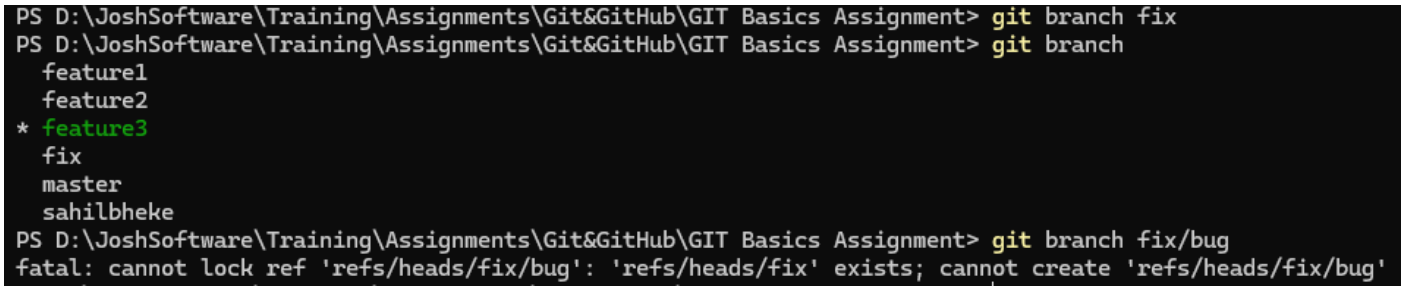
05. Under what conditions will Git consider a file that has been deleted as being renamed, and what criteria does Git use to determine that a new file is a renamed version of the deleted file?

Ans) File Content Similarity: if the content of the file that was deleted is similar to the new file, Git may consider the new file as the renamed version of the deleted file.
File Deletion and Creation: A files deletion and the creation of a new file with the similar content are considered as a rename.

06. Why is it not possible to create two branches with the names `fix` and `fix/bug` in a Git repository, and what causes this conflict?

Ans) We cannot create two branches simultaneously with the name fix and fix/bug due to the name fix and fix/bug because of how Git organizes and stores the branch references in its internal directory structure.

Conflict:
If we create fix branch we cannot create fix/bug branch because git creates a file .git/refs/heads/fix to store reference to the commit that fix points to.
If we later try to create a branch named fix/bug, Git tries to create a directory .git/refs/heads/fix to hold the bug file inside it. However, since fix is already a files, this results in a conflict.
Similarly, conflict occur if we create branch named fix/bug and later attempt to created branch named fix.



07. How to un-stage changes from the staged area. write command for it.

Ans) We can un-stage changes from the staged area with the following command

```
git reset or git reset mixed
```

Both the commands do the exact same thing without changing the content of working directory.

```
PS D:\JoshSoftware\Training\Assignments\Git&GitHub\GIT Basics Assignment> git status
On branch feature3
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:   Feature_2.txt

no changes added to commit (use "git add" and/or "git commit -a")
PS D:\JoshSoftware\Training\Assignments\Git&GitHub\GIT Basics Assignment> git add Feature_2.txt
PS D:\JoshSoftware\Training\Assignments\Git&GitHub\GIT Basics Assignment> git status
On branch feature3
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   Feature_2.txt

PS D:\JoshSoftware\Training\Assignments\Git&GitHub\GIT Basics Assignment> git reset
Unstaged changes after reset:
M       Feature_2.txt
PS D:\JoshSoftware\Training\Assignments\Git&GitHub\GIT Basics Assignment> git status
On branch feature3
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:   Feature_2.txt

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