

Question1. How git workflow works?

Answer1. There is one local repository and the developer that wants to make changes clones the repository and make changes using the commit command and further pushes it to the remote repository from where developers can pull and can work on it.

The workflow mainly consists of 4 parts, named:

1. Workspace
2. Index
3. Local Repo
4. Remote Repo

Question2. What are the different stages of git project?

Answer2. The different stages of git project are:

- Modified:

A repository is created using Git through the files that are stored in a particular folder. Anyone can make changes to a repository that is public by cloning them inside their system and modify as their requirement and then uploading it onto GitHub so that the file can be used for Staging.

- Staged:

Git add command is used to add a file from the workspace or the working directory to the Index or the staging stage. After that the file is staged i.e., it is marked to commit but is not yet committed.

- Commit:

When we used the Git commit command, all the files that were staged in the Index are released here and the changes are committed to the local repository. Each commit command represents the changes made to project in the past, with the details about the time at which commit was made and the author of the code. So, finally when you make a commit, and it gets committed, then this simply means that you have successfully applied a certain modification to the code.

Question3. Is it possible to do a git commit before git add? If No, explain why?

Answer3. No, it is not possible as git add command adds the files from the working directory to the staging area and then the changes are saved using commit command in the local repository.

The rough map can be marked as: Working Directory > Staging Area > Repository

Question 4. Why is git diff used?

Answer4. Git diff command is used to track the differences or changes made on a file in a local or a working directory from the Index/staging state.

Question5. Can we leave the commit messages as blank?

Answer5. Yes, that can be done using the following command

```
git commit -a --allow-empty-message -m ""
```

My GitHub repository link is:

<https://github.com/ashburn3r/CodeWayy-Github-Series>