Git Test 22/01/24
By:Joel Jacob John
ID:STB03T0003

Q1) Explain the Git branching strategy and mention some popular branching models.

Ans1) Branching is an essential feature in Git and version control. It allows developers to work on new features or bug fixes independently without affecting the main codebase.A Git branching strategy is a strategy adopted by the software development team when creating, merging, and deploying code. It represents a set of rules developers can follow to determine how everyone interacts with a shared codebase.A branching strategy helps organize Git repositories and prevents application errors and merge conflicts. Such conflicts occur when multiple developers work on the same project, and everyone adds their changes simultaneously.
Popular branching models include:

- Gitflow Workflow: Uses branches like master, develop, feature, release, and hotfix to manage the development lifecycle.
- GitHub Flow: Similar to Gitflow but simplified, with a main branch (typically master), and feature branches merged via pull requests.

Q2) What is a pull request in GitHub?

Ans2) A pull request (PR) is a GitHub feature that allows developers to propose changes to a repository. It serves as a way to discuss, review, and merge code changes. The process typically involves creating a branch, making changes, pushing the branch to the repository, and then creating a pull request to merge the changes into the target branch.

Q3) How do you revert a commit that has already been pushed and shared with others?

Ans3) To revert a commit that has been pushed and shared, you can use the git revert command. This command creates a new commit that undoes the changes made by the previous commit. After reverting, you need to push the new commit.

Q4) Explain the purpose of .gitignore.

Ans4) The .gitignore file is a text file that instructs Git to ignore certain files or folders in a project. A local .gitignore file is normally kept in the project's root directory. You can also create a global .gitignore file, which will be ignored in all of your Git repositories if any entries in it are found. To create a local .gitignore file, create a text file and name it .gitignore (remember to include the . at the beginning).

Q5) How do you squash multiple Git commits into a single commit?

Ans5) In order to squash the commits you'll need to use the rebase command like this:
git rebase -i HEAD~n
This tells Git to re-apply the last n commits on top of another base tip. The -i flag is short for –interactive, which will bring up your default text editor so you can edit the commands before rebasing.If you replace "pick" with "squash" then that commit will be combined with the previous one.

Q6) What is Git bisect, and how is it used?

Ans6) Git bisect is a command that helps find the commit that introduced a bug by performing a binary search through the commit history. You mark a known "good" commit and a known "bad" commit, and Git navigates between them, letting you test and identify the commit where the bug was introduced. The process involves running git bisect start, git bisect good, and git bisect bad commands until the problematic commit is found.