**1.What is Git and why is it used?**

Git is a distributed version control system that allows us to track changes in code, collaborate with others, and manage different versions of the projects.It tracks changes in project files. It allows the creation of branches. Developers can work on different features or fixes in parallel without affecting the main codebase.If needed changes made in different branches can be merged back into the main branch.

**2.Explain the difference between Git pull and Git fetch.**

**git fetch-** fetch down all the branches from that Git remote

**git pull-** fetch and merge any commits from the tracking remote branch

**3.How do you revert a commit in Git?**

To revert a commit in Git, we can use the **git revert** command

**4.Describe the Git staging area**

Git staging area is an intermediate area between your working directory and the Git repository. It is important in the Git workflow because it allows us to control and organize changes before committing them to the repository.

**5..What is a merge conflict, and how can it be resolved?**

A merge conflict in Git occurs when two branches have changes that cannot be automatically merged.

**6.How does Git branching contribute to collaboration?**

Collaboration helps to work on different features or fixes at the same time by preventing interference with the main codebase.this helps to make changes that can be independently developed, tested, and merged back into the main branch, promoting workflow.

**7.What is the purpose of Git rebase?**

Rebase helps to make changes from one branch into another by moving or combining a sequence of commits.

**8.Explain the difference between Git clone and Git fork.**

**git clone** is used to create a copy of a Git repository on your local machine.

**git fork**  creates a copy of someone else's repository under our own GitHub account, enabling us to freely experiment with changes.

**9.How do you delete a branch in Git?**

We can delete a branch by using **git branch -d <branch name>**

**10. What is a Git hook, and how can it be used?**

Git hooks are scripts that run automatically every time a particular event occurs in a Git repository. They let us customize Git's internal behavior and trigger customizable actions at key points in the development life cycle