Core Git commands

git init: Initializes a new Git repository in the current directory.
git add <file>: Stages a file for the next commit.
git add .: Stages all changes in the current directory and subdirectories.
git commit -m "<message>": Commits staged changes with a meaningful commit message.
git log: Displays a log of all commits made to the repository.
git status: Displays the status of the repository, including any changes that need to be committed.
git branch <branch-name>: Creates a new branch with the given name.
git checkout <branch-name>: Switches to the specified branch.
git merge <branch-name>: Merges changes from the specified branch into the current branch.
git remote add <name> <url>: Adds a remote repository to the local repository.
git fetch: Downloads changes from a remote repository.
git push: Pushes local changes to a remote repository.
git pull: Pulls changes from a remote repository and merges them into the current branch.

Techniques for branching, merging, and conflict resolution

Branching

git branch branch_name : To create a new branch
git checkout branch_name : To switch to an existing branch
git branch: Lists all branches.
git branch -d branch-name: Deletes a branch\

Merging

git merge branch_name: If the main branch has not moved since the feature branch was created, Git will simply move the pointer forward
git merge branch_name: If both branches have diverged, Git creates a new commit that combines changes
git merge --squash branch_name: combines all commits from the feature branch into a single commit

Conflict Resolution

Edit files manually: Resolve conflicts by editing files with conflict markers.
Git Merge Tool: Use Git's built-in visual merge tool to resolve conflicts.
Git Status: Check git status to see which files have conflicts and changes.
Git Log: Use git log to find the commits that caused the conflict.
Git Blame: Use git blame to find the last commit that changed a specific line of code.