

Problem Statement 5:

Git Ignore and Stash Utilize a .gitignore file to exclude specific files or directories from version control. Use the git stash command to temporarily save changes and switch between different branches

Solution :

1. Git Ignore:

A .gitignore file is used to specify intentionally untracked files and directories that Git should ignore. Here's a step-by-step guide:

Create a .gitignore file:

Create the file:

- You can create a .gitignore file in the root directory of your Git repository.
Edit the file:
- Open the file in a text editor and add the names or patterns of files and directories you want Git to ignore. For example:

Ignore log files

***.log**

Ignore build directories

/build/

- Each line in the file specifies a pattern to ignore. Lines starting with # are comments.
Save the file:
- Save the .gitignore file.
Commit and push:
- Add and commit the .gitignore file to your repository.

git add .gitignore

git commit -m "Add .gitignore file"

git push origin master # Replace 'master' with your branch name

2. Git Stash:

The git stash command is used to save changes in your working directory that are not ready to be committed. This is particularly useful when you need to switch branches but don't want to commit your current changes. Here's how you can use it:

Stash changes:

Stash your changes:

Run the following command to stash your changes:

git stash save "Your stash message"

Switch branches:

You can now switch to a different branch:

git checkout <branch_name>

Apply the stash:

When you're ready to continue with the changes you stashed, apply the stash:

git stash apply

If you have multiple stashes, you can specify which stash to apply.

Remove the stash (if needed):

If you want to remove the stash after applying, use:

git stash drop

This way, you can effectively use git stash to save your changes temporarily and switch between branches without committing unfinished work.