**1. Explain .gitignore**

.gitignore is a special file used by Git to tell the version control system which files or directories to **ignore**. Git will **not track** or include these files in commits, meaning they won't appear when you run git status, git add, or git commit.

This is useful for excluding:

* Temporary files (e.g., \*.log, \*.tmp)
* Build outputs (e.g., /dist/, /build/)
* Dependencies (e.g., node\_modules/)
* Sensitive data (e.g., .env)
* System files (e.g., .DS\_Store on macOS)

The .gitignore file is usually placed in the root directory of your project.

**2. Explain How to Ignore Unwanted Files Using .gitignore**

Here’s a step-by-step process to ignore files using .gitignore:

**Step 1: Create a .gitignore File**

In your Git repository root:

touch .gitignore

**Step 2: Add Ignore Rules**

Open .gitignore in a text editor and add patterns for files/folders to ignore:

*# Ignore all log files*

**\*.log**

*# Ignore node modules directory*

**node\_modules/**

*# Ignore environment config*

**.env**

*# Ignore system files*

**.DS\_Store**

**Step 3: Save and Commit the .gitignore File**

git add .gitignore

git commit -m "Add .gitignore file"

**Step 4: Untrack Already Tracked Files (If Needed)**

If a file is already being tracked by Git:

git rm -r --cached node\_modules/

git commit -m "Untrack ignored files"

**Step 5: Verify**

Run:

git status

The ignored files should no longer appear.

**Optional: Use Global .gitignore**

To ignore files across all your projects:

touch ~/.gitignore\_global

git config --global core.excludesfile ~/.gitignore\_global

OUTPUT:





