**5. .gitignore and Git Branches**

**1. .gitignore**

* **Purpose:** Tells Git which files or folders to **ignore** so they are not tracked or committed.
* Useful for excluding:
  + Temporary files (\*.log, \*.tmp)
  + Build artifacts (/build/, /dist/)
  + OS/system files (.DS\_Store, Thumbs.db)
  + Secrets/configs (.env, \*.key)

**Examples of a .gitignore file:**

# Ignore log files

\*.log

# Ignore build directory

/build/

# Ignore environment variables

.env

**Commands:**

* Check if a file is ignored:
* git check-ignore -v filename

**2. Git Branches**

* **Purpose:** A branch is a **pointer to a commit**. Branches let you work on features or fixes **without affecting the main code**.
* The default branch is usually main or master.

**Common Branch Commands:**

* Create a new branch:
* git branch branch\_name
* Switch to a branch:
* git checkout branch\_name
* Create and switch in one step:
* git checkout -b branch\_name
* List all branches:
* git branch
* Delete a branch:
* git branch -d branch\_name
* Merge a branch into current branch:
* git merge branch\_name

**✅ In short**

* **.gitignore** → Excludes files/folders from Git tracking.
* **Branches** → Let you work on features/fixes independently and merge when ready.