Git Stash & .gitignore

What is Git Stash?

`git stash` is used to **temporarily save (shelve)** your uncommitted changes so you can work on something else and come back later.

It saves changes from:

- Working Directory
- Staging Area (index)

And makes your branch **clean**.

Why do we use Git Stash?

- You want to switch branches but have unfinished work
- Urgent bug fix needed on another branch
- Don't want to commit half-done code
- Avoid losing local changes

Stash = temporary storage for code

Basic Git Stash Commands

1. `git stash`

```
git stash
```

✔ Saves modified + staged files ✔ Cleans working directory

---

2. `git stash push -m "message"`

```
git stash push -m "login page changes"
```

✔ Stash with a custom message (recommended)

---

3. `git stash list`

```
git stash list
```

Shows all stashed changes

Example:

```
stash@{0}: On main: login page changes
stash@{1}: On dev: css update
```

4.`git stash apply`

```
git stash apply
```

✔ Applies **latest stash** ✔ Stash remains in list

5.`git stash apply stash@{1}`

```
git stash apply stash@{1}
```

✔ Apply specific stash

6.`git stash pop`

```
git stash pop
```

✔ Apply latest stash ✔ **Removes** it from stash list

Most commonly used

---

7. `git stash drop`

```
git stash drop stash@{0}
```

✖ Deletes a specific stash

---

8. `git stash clear`

```
git stash clear
```

Deletes **all stashes** (dangerous)

9.Stash including untracked files

```
git stash -u
```

✔ Saves untracked files also

10.Stash including ignored files

```
git stash -a
```

Saves **all files** (tracked + untracked + ignored)

.gitignore

What is .gitignore?

`.gitignore` is a file used to **tell Git which files/folders should NOT be tracked or pushed** to the repository.

Common examples:

- Passwords
- Build files
- Logs
- Node modules
- Environment files

Why we use .gitignore?

- Security (hide secrets)
- Avoid unnecessary files
- Reduce repo size
- Keep repo clean

.gitignore File Location

📍 Create `.gitignore` in **root directory** of project

```
touch .gitignore
```

Common .gitignore Examples

Ignore files

```
cred.txt
secrate.txt
```

Ignore folders

```
node_modules/
dist/
build/
```

Ignore all .log files

```
*.log
```

Ignore OS files

```
.DS_Store
Thumbs.db
```

Ignore compiled files

```
*.class
*.exe
*.out
```

Important Rules of .gitignore

`.gitignore` does NOT work on already tracked files

To remove tracked file:

```
git rm --cached file_name
git commit -m "Removed tracked file"
```

Git Stash vs Commit

Git Stash	Git Commit
Temporary	Permanent
Local only	Saved in repo
No history	Full history
Used for quick switch	Used for version control
