1. What is Git?

Git is a **version control system** that tracks changes in your code over time. It lets you save different versions, collaborate with others, and roll back if something breaks. Think of it as a time machine for your code.

---

2. Difference between merge and rebase

- **Merge**: Combines two branches, keeping their history as it is (can create a “merge commit”).
- **Rebase**: Moves your changes to the tip of another branch, making the history cleaner but rewriting commits.

---

3. What is a pull request?

A pull request is how you ask others to review and approve your changes before merging them into the main branch. It’s common in GitHub and GitLab for collaboration.

---

4. How do you resolve merge conflicts?

When two people edit the same part of a file, Git can’t decide which version to keep. You open the file, manually choose the correct changes, save it, and then mark it as resolved with:
bash
CopyEdit

```
git add <file>
git commit
```

---

5. What are Git tags?

Tags are labels for specific commits—often used for marking releases like v1.0 or v2.1. They make it easier to refer back to that exact state of the project.

---

6. What is Git workflow?

A Git workflow is the process your team follows for making changes. Examples:

- **Feature Branch Workflow**: Create a branch for each feature.
- **Gitflow**: Separate branches for development, release, and production.

---

7. Explain git stash

git stash temporarily saves your uncommitted changes so you can work on something else without committing them. Later, you can bring them back with git stash pop.

---

8. What is the use of .gitignore?

The .gitignore file tells Git which files/folders to skip tracking—like log files, temporary builds, or environment configs. This keeps your repo clean and avoids pushing sensitive files.