Git & GitHub - Interview Notes

Why Git & GitHub?

- **Git** is a distributed version control system used to track changes in source code, collaborate with team members, and maintain project history.
- **GitHub** is a cloud-based hosting platform for Git repositories with collaboration features like pull requests, issues, and code reviews.
- Together, they streamline teamwork and software development.

Advantages of Git & GitHub

- Tracks changes and maintains history of code.
- Supports branching & merging for parallel development.
- Enables collaboration among developers.
- Provides centralized backup & storage via GitHub.
- Encourages open-source contributions.
- Integrates with CI/CD tools.
- Handles large projects efficiently.

Important Git Commands

Command	Description
git init	Initialize a new Git repository
git clone <url>	Clone a remote repository
git status	Show repository status
git add <file>	Stage file changes
git add .	Stage all changes
git commit -m "msg"	Commit staged changes

Command	Description
git log	View commit history
git branch	List branches
git branch <name>	Create a new branch
git checkout <branch>	Switch to a branch
git checkout -b <name>	Create & switch to new branch
git merge <branch>	Merge branch into current branch
git remote add origin <url>	Link local repo to remote
git push origin main	Push changes to remote
git pull origin main	Fetch & merge from remote
git reset --hard <hash>	Reset to specific commit
git revert <hash>	Undo commit with new commit
git stash	Save uncommitted changes temporarily

Key GitHub Features

- Pull Requests: Propose and review changes.
- Issues: Track bugs & tasks.
- GitHub Actions: Automate workflows (CI/CD).
- Projects: Task management (Kanban).
- Code Review: Peer collaboration.
- Forking: Contribute to public projects.

Typical Workflow

1. Initialize repo with git init or clone with git clone.
2. Edit files and make changes.
3. Stage changes with git add.
4. Commit changes with git commit -m "message".
5. Push to remote with git push.
6. Create feature branches with git checkout -b feature.
7. Merge branches into main with git merge.
8. Collaborate using Pull Requests & Code Reviews on GitHub.