COMMAND OF GIT TOOLS (PART-2)

ADVANCED GIT COMMANDS AND TECHNIQUES

GIT COMMANDS

- Git is a distributed version control system that helps developers manage changes to source code during software development.

SOME OF GIT COMMANDS

- git init: Initializes a new Git repository.
- git clone: Clones a repository into a new directory.
- git status: Shows the status of changes in the working directory.
- git add: Adds files to the staging area.
- git commit: Records changes to the repository with a message.
- git push: Pushes changes to a remote repository.
- git pull: Fetches and merges changes from a remote repository.
- git branch: Lists, creates, or deletes branches.
- git checkout: Switches branches or restores working tree files.

CONTINUE…

- git merge: Merges changes from one branch into the current branch.
- git log: Shows the commit history.
- git diff: Shows differences between commits, commit and working tree, etc.
- git remote: Manages set of tracked repositories.
- git fetch: Downloads objects and refs from another repository.
- git rebase: Reapplies commits on top of another base tip.

GIT TECHNIQUES

- **Rebasing vs. Merging**:
- **Rebase**: Reapplies commits on top of another base tip, creating a linear history.
- **Merge**: Combines the histories of two branches, maintaining the original commits.
- **Git Hooks**:
- Scripts that Git executes before or after events like commit, push, and receive.
- Example: pre-commit, pre-push, post-merge.
- **Amending Commits**:
- Use git commit --amend to modify the most recent commit message or add changes to it.

CONTINUE….

- **Squashing Commits**:
- Combine multiple commits into one using interactive rebase (git rebase -i).
- **Branch Management**:
- Regularly clean up branches using git branch -d for local branches and git push origin --delete for remote branches.
- **Handling Merge Conflicts**:
- Use git merge tool to resolve conflicts with a visual tool.