SSL CLOUD

GIT Training

Day 1: Introduction to Git

- **What is Git?**
  - Version control system to track changes in source code
  - Helps in collaboration and maintaining a history of work
- **Why Use Git?**
  - Collaboration on codebases
  - Branching and merging for parallel development
  - Undo mistakes with version history
- **Setting up Git**
  - Installing Git on different operating systems
  - Configuring user name and email
  - Initializing a repository
- **Hands-On Activity:**
  - Install Git and create your first local repository

Day 2: Basic Git Commands

- **Git Workflow Overview**
  - Working directory, staging area, and repository
- **Core Commands:**
  - git init: Initialize a repository
  - git status: Check repository status
  - git add: Stage changes
  - git commit: Save changes to repository
  - git log: View commit history

- **Hands-On Activity:**
    - Create files, make changes, stage them, and commit to the repository

Day 3: Branching and Merging

- **What are Branches?**
    - Parallel lines of development
    - Master/main branch and feature branches
- **Branch Management Commands:**
    - git branch: List, create, or delete branches
    - git checkout: Switch branches
    - git merge: Merge branches
- **Conflict Resolution:**
    - Understanding merge conflicts
    - Resolving conflicts manually
- **Hands-On Activity:**
    - Create and merge branches
    - Resolve a simulated merge conflict

Day 4: Working with Remote Repositories

- **What is a Remote Repository?**
    - Shared repository hosted on platforms like GitHub, GitLab, or Bitbucket
- **Commands for Remote Repositories:**
    - git remote: Connect to a remote repository
    - git clone: Copy a repository to local machine
    - git push: Upload local changes to remote
    - git pull: Download changes from remote
    - git fetch: Fetch changes without merging
- **Hands-On Activity:**
    - Create a repository on GitHub
    - Push and pull changes

Day 5: Advanced Git Features

- **Stashing Changes:**
    - Temporarily save uncommitted changes

    - git stash and git stash pop
- **Reverting Changes:**
    - Undo changes with git revert and git reset
- **Tagging:**
    - Mark important commits with git tag
- **Rebasing:**
    - Streamline commit history with git rebase
- **Hands-On Activity:**
    - Experiment with stashing, tagging, and rebasing

Day 6: Collaboration Best Practices

- **Pull Requests and Code Reviews:**
    - Use pull requests for collaboration
    - Conduct effective code reviews
- **Working with Teams:**
    - Handling large codebases
    - Setting branch protection rules
- **Git Workflows:**
    - Centralized workflow
    - Feature branch workflow
    - Gitflow workflow
- **Hands-On Activity:**
    - Simulate a collaborative project with pull requests

Day 7: Git Tools and Automation

- **Graphical User Interfaces for Git:**
    - Tools like GitKraken, SourceTree, and GitHub Desktop
- **Git Hooks:**
    - Automate tasks with pre-commit and post-commit hooks
- **CI/CD with Git:**
    - Integrate Git with Jenkins, GitHub Actions, or GitLab CI
- **Hands-On Activity:**
    - Set up a pre-commit hook and configure a basic CI pipeline

Day 8: Wrap-Up and Q&A

- **Recap of All Topics**
    - Summary of key concepts and commands
- **Tips for Effective Git Usage**
    - Commit messages, branching strategies, and collaboration tips
- **Open Q&A Session**
    - Address queries and clarify doubts
- **Final Hands-On Activity:**
    - Consolidate learning by managing a mini project

Thank You

- Ready to master Git? Start collaborating and coding with confidence!