B.Tech. [2nd Year] Student

MODULE 05

VERSION CONTROL WITH GIT AND PROJECT IMPLEMENTATION

➤ ASSIGNMENT 01

Comprehensive Overview of Git: Key Concepts, Commands, and Workflows

In this document, I summarize the essential concepts, commands, and workflows of Git, providing a clear understanding of how to effectively use version control in software development. The topics covered include:

1. Repository Initialization

- Definition: This is the process I used to create a new Git repository to start tracking my project files.
- Key command:

  git init

  I use this command to initialize a new repository in my current directory.

2. Staging Changes

- Definition: The act of selecting specific changes to include in the next commit.
- Key command:

  git add <file>

  This command stages the specified file for commit.

3. Committing Changes

- Definition: Capturing the current state of the project with a descriptive message, marking a point in the project history.
- Key Command: git commit -m "Your commit message"

  This command records the staged changes along with a message explaining what was changed.

4. Branching

- Definition: Creating separate lines of development within a project to work on features or fixes independently.
- Key Commands:
  - To create a new branch:
    git branch <branch-name>
  - To switch to that branch:
    git checkout <branch-name>

5. Merging

- Definition: Integrating changes from one branch into another, typically from a feature branch back into the main branch.
- Key Command:
  git merge <branch-name>
  This command merges the specified branch into the current branch.

6. Collaboration with Remote Repositories

- Definition: Working with shared repositories hosted on platforms like GitHub or GitLab to facilitate teamwork.
- Key Commands:
  - To add a remote repository:
    git remote add origin <repository-url>
  - To push local changes to the remote repository:
    git push origin <branch-name>

♦ Conclusion

This document encapsulates the fundamental aspects of using Git for version control, providing a clear reference for initializing repositories, staging and committing changes, managing branches, merging code, and collaborating through remote repositories. Understanding these concepts and commands is crucial for effective softwa development practices.