Lab 1: Implementation of Git

Objective

To understand and implement the basic operations of Git as a distributed version control system and explore how it supports Agile development practices such as collaboration, continuous integration, and iterative development.

Tools and Technologies Used

- Git (installed locally)
- GitHub
- Command-line interface (Terminal, Git Bash, or Command Prompt)
- Text editor or IDE (e.g., VS Code)

Theory

Git is a distributed version control system that helps manage source code history. It is widely used in Agile development for managing code changes across multiple developers, maintaining branches for different features or sprints, and enabling continuous integration workflows. Key concepts include:

- Repository
- Commit
- Branch
- Merge
- Push/Pull
- Clone

Git enables Agile practices like:

- Frequent commits (iteration)
- Collaboration through branches
- Transparency of progress
- Safe experimentation with branching

Procedure / Steps

1. **Initialize a Git Repository**

   git init

2. **Configure Git**

   git config --global user.name " AnshGrg "

   git config --global user.email "ansh2073@gmail.com"

3. **Create a new file and track it**

   git add README.md

   git commit -m "Initial commit"

4. **Make changes, commit**

   git add .

   git commit -m "Initial commit"

5. **Push to remote repository**

   git remote add origin https://github.com/AnshGrg/Agile.git

   git push -u origin main

Observations

- Git allows multiple developers to work independently and merge work easily.
- Branching helps in isolating features and experimenting without affecting the main codebase.
- Every change is tracked and reversible, which improves accountability and collaboration.

Conclusion

The implementation of Git demonstrated its powerful capabilities in version control, essential for Agile development. It supports parallel development, frequent iterations, and continuous delivery, making it an indispensable tool for Agile teams.