# GANDAKI COLLEGE OF ENGINEERING AND SCIENCE

**Lamachaur,Pokhara**


LAB REPORT OF

## Agile Software Development

**LAB – 1**

**SUBMITTED BY:**

Ashik Chapagain

Roll No: 11

6th Semester

BE Software

**SUBMITTED TO:**

Er. Rajendra Bdr. Thapa

# LAB 1: Implementation of Git VCS in Agile Software Development

## OBJECTIVE:

To understand the fundamentals of Git Version Control System.

## TOOLS / METHODOLOGIES:

- Git Bash
- IDE / Text Editor
- GitHub

## THEORY:

Version control systems are essential in agile development where multiple developers work simultaneously on features, requiring frequent integration and deployment. Git's distributed nature aligns perfectly with agile principles by enabling autonomous work while maintaining code integrity.

Git is a distributed version control system that plays a crucial role in agile software development methodologies. This lab focuses on understanding and implementing Git as a fundamental tool for collaborative software development, enabling teams to track changes, manage code versions, and facilitate seamless collaboration in agile environments through a practical JavaScript web application project.

## OBSERVATION:

1. Initial Git Configuration After Installation

   ```
   git config --global user.name "Ashik Chapagain"
   git config --global user.email "ashik@github.com"
   ```

   Git needs to identify, who made the changes, and it uses these global configurations for identifying the developer who made the changes.

2. Initialize Git Repository

   ```
   git init
   ```

   This creates a hidden .git folder that tracks changes, commits, and branches in your project.

3. Add Files to Git Repository

   ```
   git add .
   ```

   git add . stages all changes (new, modified, deleted files) in the current directory and its subdirectories for the next commit.

4. Commit the changes

   ```
   git commit -m “initial commit”
   ```

   Creates a snapshot of the staged changes with a message.

5. Add remote repository

   ```
   git remote add origin <repository-url>
   ```

   Links your local repository to a remote one (usually on GitHub).

6. Push to remote

   ```
   git push -u origin master (or main, depending on the branch)
   ```

   Uploads your committed changes to the remote repository and sets the upstream branch.

## CONCLUSION:

This lab successfully demonstrated the implementation of Git as a version control system. The hands-on experience with Git commands, practical understanding of how version control supports software development.