

GANDAKI COLLEGE OF ENGINEERING AND SCIENCE

Lamachaur, Pokhara



LAB REPORT OF Agile Software Development LAB – 1

SUBMITTED BY:

Baradan Parajuli

Roll No: 16

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 crucial in agile development, enabling numerous developers to collaborate on features at the same time, necessitating regular integration and deployment. The distributed nature of Git aligns seamlessly with agile principles by allowing independent work while preserving code integrity.

Git is a decentralized version control system that is essential in agile software development practices. This lab centers on grasping and applying Git as an essential instrument for cooperative software development, allowing teams to monitor modifications, handle code revisions, and promote smooth collaboration in agile settings via a hands-on JavaScript web application endeavor.

OBSERVATION:

1. Initial Git Configuration After Installation

```
git config --global user.name "Baradan Parajuli"  
git config --global user.email "bardan@github.com"
```

This is to identify 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 .
```

Prepares all changes—including new, updated, and removed files—in the current directory and its subdirectories to be included in the next commit.

4. Commit the changes

```
git commit -m "message"
```

Captures the staged changes as a saved version along with a descriptive 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 effectively showcased the use of Git as a version control system. It provided hands-on experience with Git commands and offered practical insight into how version control facilitates software development.