GANDAKI COLLEGE OF ENGINEERING AND SCIENCE

Lamachaur,Pokhara

LAB REPORT OF

Agile Software Development

LAB – 1

SUBMITTED BY:
Rajiv thapa
Roll No: 39
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 is a cornerstone of agile development, especially when multiple developers are working concurrently on different features. It ensures smooth integration, continuous delivery, and effective collaboration. Git, being a distributed version control system, supports agile practices by allowing decentralized development workflows, enabling developers to work independently while ensuring code consistency and traceability. This lab provides hands-on experience using Git for managing versions and collaborating on a JavaScript web application project in an agile environment.

OBSERVATION:

1. Initial Git Configuration After Installation

```
git config --global user.name "Rajiv Thapa"
git config --global user.email "rajiv@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 provided a practical introduction to Git as a version control system within the context of agile development. Through hands-on use of Git commands, we explored how Git facilitates team collaboration, change tracking, and efficient management of code versions essential skills for modern software development practices.