

**Gandaki College of Engineering and Science**

**Enterprise Application Development**

*Lab Report: Git and GitHub Demonstration*

Lab: 01

|  |  |
| --- | --- |
| **Submitted By:**  Bishal Koirala Roll No: 15 | **Submitted To:**  Er. Prativa Nyaupane Lecturer, Gandaki College of Engineering and Science |
| **GitHub Profile Name:** koiralabishal **GitHub Repository Name:** EAD-Lab **Repository Link:** https://github.com/koiralabishal/EAD-Lab | |

# Objectives

**1. Version Control Understanding**

* Master the fundamental concepts of Git version control system
* Understand distributed version control principles
* Learn repository management and initialization

**2. Branch Management**

* Create and manage multiple branches effectively
* Implement proper branching strategies
* Handle branch synchronization with remote repository

**3. Collaboration Skills**

* Master pull request workflow
* Practice code review processes
* Learn team collaboration techniques

# Theory

**1. Git Version Control System**

* Distributed version control system for tracking changes
* Enables multiple developers to work simultaneously
* Maintains complete history of source code changes

**2. Git Repository Structure**

* Working Directory: Where files are edited
* Staging Area: Preparation for commits
* Local Repository: Stores commit history
* Remote Repository: Centralized storage on GitHub

**3. Branching and Merging**

* Branches allow parallel development
* Feature branches isolate new development
* Merging combines changes from different branches
* Pull requests facilitate code review

# Implementation Process

Step-by-step guide through the Git workflow implementation with detailed commands and explanations.

## Step 1: Project Setup

$ mkdir EAD-Lab-01

$ cd EAD-Lab-01

## Step 2: Git Repository Setup

$ git init

Initialized empty Git repository in D:/EAD Lab/EAD Lab 1/.git/

## Step 3: Remote Repository Integration

$ git remote add EAD https://github.com/koiralabishal/EAD-Lab.git

$ git remote -v

EAD https://github.com/koiralabishal/EAD-Lab.git (fetch)  
EAD https://github.com/koiralabishal/EAD-Lab.git (push)

## Step 4: Branch Management

$ git branch -a

bugfid  
\* main  
 remotes/EAD/bugfid  
 remotes/EAD/develop  
 remotes/EAD/feature  
 remotes/EAD/main

## Step 5: Feature Development and Pull Request

### 1. Feature Branch Work

$ git checkout feature

Switched to branch 'feature'  
Your branch is up to date with 'EAD/feature'

### 2. Make Changes and Commit

$ git add index.html

$ git commit -m "feat: Add modified EAD Lab 1"

[feature 646eaee] Add modified EAD Lab 1  
1 file changed, 150 insertions(+), 0 deletions(-)

### 3. Push Feature Branch

$ git push EAD feature

To https://github.com/koiralabishal/EAD-Lab.git  
 0829ef0..646eaee feature -> feature

### 4. Create Pull Request (Feature to Develop)

$ gh pr create --base develop --head feature --title "Add EAD Lab 1 Modified" --body "Added lab report with implementation details"

Creating pull request for feature into develop in koiralabishal/EAD-Lab  
https://github.com/koiralabishal/EAD-Lab/pull/1

## Step 6: Merge Process

### 1. Merge Feature into Develop

$ git checkout develop

$ git pull EAD develop

$ git merge feature --no-ff

Switched to branch 'develop'  
Already up to date.  
Merge made by the 'recursive' strategy.  
index.html | 150 ++++++++++++++++++  
1 file changed, 150 insertions(+)

### 2. Push Develop Branch

$ git push EAD develop

To https://github.com/koiralabishal/EAD-Lab.git  
 646eaee..d580c75 develop -> develop

### 3. Create Pull Request (Develop to Main)

$ gh pr create --base main --head develop --title "Merge Develop into Main" --body "Merging completed feature work into main branch"

Creating pull request for develop into main in koiralabishal/EAD-Lab  
https://github.com/koiralabishal/EAD-Lab/pull/2

### 4. Merge into Main

$ git checkout main

$ git pull EAD main

$ git merge develop --no-ff

$ git push EAD main

Switched to branch 'main'  
Already up to date.  
Merge made by the 'recursive' strategy.  
To https://github.com/koiralabishal/EAD-Lab.git  
 d580c75..e892f41 main -> main

### 5. Clean Up Feature Branch

$ git branch -d feature

$ git push EAD --delete feature

Deleted branch feature (was 646eaee).  
To https://github.com/koiralabishal/EAD-Lab.git  
 - [deleted] feature

## Step 7: Final Verification

$ git status

On branch main  
Your branch is up to date with 'EAD/main'  
nothing to commit, working tree clean

## Implementation Overview

|  |  |  |  |
| --- | --- | --- | --- |
| **Total Steps** | **Branches Created** | **Git Commands** | **Pull Requests** |
| 7 | 4 | 18 | 2 |

# Conclusion

**Achievements**

* Successfully implemented Git version control system
* Mastered branch creation and management
* Learned pull request workflow and collaboration
* Gained practical experience with GitHub

**Learning Outcomes**

* Understanding of modern version control practices
* Proficiency in Git commands and operations
* Experience with collaborative development workflows
* Knowledge of professional Git branching strategies