VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Machhe, Belagavi, Karnataka-590018

Lab Experiment Record

Project Management with Git [BCSL58C]

Submitted in partial fulfillment towards AEC of 3rd semester of

Bachelor of Engineering
in
Computer Science and Engineering
(Artificial Intelligence & Machine Learning)

Submittedby

SANIKA.P
4GW24CI039

DEPARTMENT OF CSE (ArtificialIntelligence & Machine Learning)
GSSS INSTITUTE OF ENGINEERING & TECHNOLOGY FOR WOMEN
(Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi & Govt. of Karnataka)
K.R.S ROAD, METAGALLI, MYSURU-570016, KARNATAKA
(Accreditedby NAAC)

TABLE OF CONTENT

12.AnalysingandChangingGitHistory

Writethecommandtoundothechanges introduced by the commit with the ID "abc123".

EXPERIMENT 1
SETTINGUPANDBASICCOMMANDS

Git is a version control system used to track changes in files and manage project history

Command:

- Ihave created new folder with my USN.
- Inside the folder i have all the program files.
- Thegit config command is used set user the information.
- Thegit init command was used to initialize a new Git repository.
- Thestaging area stores changes before committing the files.
- Thegit commit command was used to save the changes permanently.
- Aproper commit message was given to describe the changes.

```
user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git init
Reinitialized existing Git repository in C:/Users/user/OneDrive/Desktop/4GW24CI039_SanikaP/.git/

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git config --global user.name "Sanika"

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git config --global user.email "sanikapapanna@gamil.com"

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        DSA/

nothing added to commit but untracked files present (use "git add" to track)

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git add DSA/

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   DSA/program4.txt
        new file:   DSA/program5a.txt
        new file:   DSA/program5b.txt
        new file:   DSA/program7.txt
        new file:   DSA/program8.txt


user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git remote add origin https://github.com/sanikapapanna-cmd/4GW24CI039_Sanika.P.git

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git commit -m "DSA progrms added"
[master (root-commit) 7298415] DSA progrms added
 5 files changed, 576 insertions(+)
```

```
user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git push origin master
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 4.01 KiB | 315.00 KiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/sanikapapanna-cmd/4GW24CI039_Sanika.P.git
 * [new branch]      master -> master
```

- The git push command is used to send our code to GitHub.
- It uploads the changes from our computer to the online repository.
- Push is done after saving changes using commit.

EXPERIMENT 2
CREATING AND MANAGING BRANCHES

Createanewbranchnamed“feature-branch”. Switch to the “master” branch merge the “feature-branch” into “master”.

- Gitcheckout master
- This command is used to move to the *master* branch.
- Gitcheckout -b feature-branch
- This command creates a new branch named *feature-branch* and switches to it.
- Gitstatus
- This command shows the current branch and file status.
- Gitadd
- This command adds the modified file to the staging area.

Gitmerge feature-branch

This command merges the changes from *feature-branch* into *master*.

```
user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git checkout master
Already on 'master'

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (feature-branch)
$ git status
On branch feature-branch
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   DSA/program5a.txt

no changes added to commit (use "git add" and/or "git commit -a")

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (feature-branch)
$ git add DSA/program5a.txt

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (feature-branch)
$ git commit -m "DSA program 5a has been modified"
[feature-branch 9aa5bca] DSA program 5a has been modified
 1 file changed, 1 insertion(+)

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (feature-branch)
$ git checkout master
Switched to branch 'master'

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git merge feature-branch
Updating 7298415..9aa5bca
Fast-forward
 DSA/program5a.txt | 1 +
 1 file changed, 1 insertion(+)
```

Experiment 3

CREATING AND MANAGING BRANCHES

Write The commends to. Stash your changes. Switch branches. And then apply the stashed changes.

```
user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git add DSA/program8.txt

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git stash save "8th program modified"
Saved working directory and index state On master: 8th program modified

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git checkout feature-branch
Switched to branch 'feature-branch'

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (feature-branch)
$ git stash apply
On branch feature-branch
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   DSA/program8.txt

no changes added to commit (use "git add" and/or "git commit -a")

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (feature-branch)
$ git add DSA/program8.txt

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (feature-branch)
$ git stash apply
On branch feature-branch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   DSA/program8.txt
```

Command:

- git add DSA/program8.txt
  This command adds the modified file to the staging area
- git stash save “8th program modified”
  This command temporarily saves the current change
- git checkout feature-branch
  This command switches from the master branch to the feature branch.
- git stash apply
  This command brings back the stashed changes into the current branch.

EXPERIMENT 4

COLLABORATION AND REMOTE REPOSITORIES.

Clonearemote Git repository to your local machine.

Command:

- gitclone
  The git clone command is used to copy a project from GitHub to the computer.
- Ittakes the URL the clone to the respective repository.

```
user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (feature-branch)
$ git clone https://github.com/sanikapapanna-cmd/4GW24CI039_Sanika.P.git
Cloning into '4GW24CI039_Sanika.P'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 8 (delta 0), reused 8 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (8/8), done.
```

EXPERIMENT 5
COLLABORATIONANDREMOTEREPOSITORIES

Fetchthelatestchangesfrom a remote repository and rebase your local branch onto the updated remote branch.

Commands:

- git checkout master
  This command switches to the master branch.
- git fetch origin
  This command downloads the latest updates from the remote repository.
- git commit -m “added”
  This command saves the local changes in the repository with a commit message..
- git rebase origin/master
  This command updates the local master branch with the latest remote changes.
- git push origin master
  This command uploads the updated local master branch to the remote repository.

```
user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (feature-branch)
$ git checkout master
M       DSA/program8.txt
Switched to branch 'master'

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git fetch origin

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git commit -m "added"
[master 2d5b185] added
 1 file changed, 1 insertion(+)

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git rebase origin /master
fatal: no such branch/commit 'C:/Program Files/Git/master'

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git rebase origin/master
Current branch master is up to date.

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git rebase --continue
fatal: no rebase in progress

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git push origin master
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (8/8), 638 bytes | 212.00 KiB/s, done.
Total 8 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 3 local objects.
To https://github.com/sanikapapanna-cmd/4GW24CI039_Sanika.P.git
   7298415..2d5b185  master -> master
```

EXPERIMENT 6

COLLABORATION AND REMOTE REPOSITORIES

Write the command to merge "feature-branch" into "master" while providing a custom commit message for the merge

Command:

- git checkout master.

  This command switches to the master branch

  Merging is always done from the branch we want to update

- Git merge feature-branch -m “Merged feature-branch into master”

  This command merges the changes from feature-branch into master.

  The -m option is used to give a custom message for the merge commit.

```
user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (feature-branch)
$ git checkout master
Switched to branch 'master'

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git merge feature-branch -m "merged master and feature-branch"
Already up to date.

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$
```

EXPERIMENT 7

GIT TAGS AND RELEASES

Write the command to create a lightweight Git tag named "v1.0" for a commit in your local repository.

Command:

- gittag v1.0
  git tags are used to mark important versions of a project.
  The command git tag v1.0 creates a lightweight tag named v1.0.
  This tag points to the current commit.
- gitlog -oneline
  Shows the history of commits
  Displays one commit per line

```
user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git tag v1.0

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git log --oneline
2d5b185 (HEAD -> master, tag: v1.0, origin/master, origin/HEAD) added
9aa5bca (feature-branch) DSA program 5a has been modified
7298415 DSA progrms added

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git tag v1.1 9aa5bca

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git push origin --tags
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/sanikapapanna-cmd/4GW24CI039_Sanika.P.git
 * [new tag]         v1.0 -> v1.0
 * [new tag]         v1.1 -> v1.1

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$
```

EXPERIMENT 8

ADVANCED GIT OPERATIONS

Write the command to cherry-pick for a range of commits from "source-branch" to the current.

Command:

- gitlog–one line command.
  This command shows commit history in one line per commit.
  It displays a short commit ID and commit message.
- Gitcherry-pick id command
  It is used to copy a specific commit from one branch to another.
  It takes the id of the commit and copies it.

```
user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ echo "changes for another branch">README.md

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git commit -m "changes"
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        4GW24CI039_Sanika.P/
        README.md

nothing added to commit but untracked files present (use "git add" to track)

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git add .
warning: adding embedded git repository: 4GW24CI039_Sanika.P
hint: You've added another git repository inside your current repository.
hint: Clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:
hint:   git submodule add <url> 4GW24CI039_Sanika.P
hint:
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
hint:   git rm --cached 4GW24CI039_Sanika.P
hint:
hint: See "git help submodule" for more information.
hint: Disable this message with "git config set advice.addEmbeddedRepo false"
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git add README.md

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git log --oneline
2d5b185 (HEAD -> master, tag: v1.0, origin/master, origin/HEAD) added
9aa5bca (tag: v1.1, feature-branch) DSA program 5a has been modified
7298415 DSA progrms added

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git cherry-pick 649e108
fatal: bad revision '649e108'

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git cherry-pick 9aa5bca
```

```
user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git cherry-pick 9aa5bca
On branch master
You are currently cherry-picking commit 9aa5bca.
  (all conflicts fixed: run "git cherry-pick --continue")
  (use "git cherry-pick --skip" to skip this patch)
  (use "git cherry-pick --abort" to cancel the cherry-pick operation)
```

```
[master 51b390b] DSA progrms added
 Author: Sanika <sanikapapanna@gamil.com>
 Date: Tue Jan 6 10:08:18 2026 +0530
 2 files changed, 6 insertions(+)
```

EXPERIMENT 9

ANALYSING AND CHANGING GIT HISTORY

Given a commit ID, how would you use Git to view the details of that specific commit,including the author, date, and commit message?

Command:

- gitlog–oneline command.
  Thiscommand shows commit history in one line per commit.
  Itdisplays a short commit ID and commit message.
- gitshowidcommand
  Thiscommand shows details of a commit.
  Itdisplays changes made in the latest commit.
  Itdisplays the commit message, author and the date.

```
user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git log --oneline
51b390b (HEAD -> master) DSA progrms added
2d5b185 (tag: v1.0, origin/master, origin/HEAD) added
9aa5bca (tag: v1.1, feature-branch) DSA program 5a has been modified
7298415 DSA progrms added

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git show 2d5b185
commit 2d5b1856cc8dc0a403f06ee081b047eee940fb90 (tag: v1.0, origin/master, origin/HEAD)
Author: Sinchana <sinchanamj006@gmail.com>
Date:   Tue Jan 6 10:45:16 2026 +0530

    added

diff --git a/DSA/program8.txt b/DSA/program8.txt
index 3ab1b21..0359570 100644
--- a/DSA/program8.txt
+++ b/DSA/program8.txt
@@ -192,3 +192,4 @@ printf("Invalid choice!\n");
 }
 return 0;
 }
+////////////////////////////////////////////////////////////////////////////////

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git log -n 1 2d5b185
commit 2d5b1856cc8dc0a403f06ee081b047eee940fb90 (tag: v1.0, origin/master, origin/HEAD)
Author: Sinchana <sinchanamj006@gmail.com>
Date:   Tue Jan 6 10:45:16 2026 +0530

    added

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$
```

EXPERIMENT 10

ANALYSING AND CHANGING GIT HISTORY

Write the command to list all commits made by the author "JohnDoe" between"2023-01- 01"and "202312-31."

Command:

- git log –author=” “ --since=’ ‘ --until=’ ‘ command.

  Is used to view the history of commit
  It shows all commits made in the repository.
  It includes author so that it know which person’s commit has to be shown
  It also include the range from which date to which date the history of commits has to be shown

```
user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git log --author="sanika" --since='2026-01-06' --until='2026-01-06'
```

EXPERIMENT 11

ANALYSING AND CHANGING GIT HISTORY

Writethecommand to display the last five commits in the repository's history

Command:

- gitlog-ncommand

The git log command is used to view the history of commits.

The -n option is used with git log to limit the number of commits that will be shown. Itshows only the latest commits instead of the full history.

Here it will show the last 6 commits that has been done as n is 6

```
user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git log --author="sanika" --since='2026-01-06' --until='2026-01-06'

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$ git log -n 6
commit 51b390bb708f7206181ed6fed4bcd8acc0298e9e (HEAD -> master)
Author: Sanika <sanikapapanna@gamil.com>
Date:   Tue Jan 6 10:08:18 2026 +0530

    DSA progrms added

commit 2d5b1856cc8dc0a403f06ee081b047eee940fb90 (tag: v1.0, origin/master, origin/HEAD)
Author: Sinchana <sinchanamj006@gmail.com>
Date:   Tue Jan 6 10:45:16 2026 +0530

    added

commit 9aa5bca249513fc09eda941d5d841e49578b2e34 (tag: v1.1, feature-branch)
Author: Sinchana <sinchanamj006@gmail.com>
Date:   Tue Jan 6 10:30:59 2026 +0530

    DSA program 5a has been modified

commit 729841566f7e373b7dc1b50fa6acd5c791bf083c
Author: Sanika <sanikapapanna@gamil.com>
Date:   Tue Jan 6 10:08:18 2026 +0530

    DSA progrms added

user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master)
$
```

EXPERIMENT 12

ANALYSING AND CHANGING GIT HISTORY

Write the command to undo the changes introduced by the commit with the ID "abc123"

- gitrevert command
  The git revert command is used to undo a commit .
  It takes the commit which has to be reverted.

- gitrevert “commit”.
  It does not delete the commit history.
  Instead, it creates a new commit that reverses the changes .

```
user@DESKTOP-7H2SLKJ MINGW64 ~/OneDrive/Desktop/4GW24CI039_SanikaP (master|REVERTING)
$ git revert 5a21f98
```