



Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



DEPARTMENT OF INFORMATION TECHNOLOGY

COURSE CODE: DJS22ITHN1L1

DATE: 31-01-2025

COURSE NAME: DevOps Laboratory

CLASS: TY BTech

NAME: Anish Sharma

ROLL: I011

DIV: IT1-1

EXPERIMENT NO. 2

CO/LO: Apply DevOps principles to meet software development requirements.

AIM / OBJECTIVE: To perform various GIT operations on local and Remote repositories

THEORY:

What is Git?

Git: Git is a distributed version control system that enables developers to track changes, manage branches, and collaborate on projects efficiently. Its distributed nature allows each user to have a full copy of the repository, facilitating offline work and robust version tracking. Git has become the de facto standard for version control in modern software development.

What is GitHub?

GitHub: GitHub is a web-based platform built upon Git. It provides a user-friendly interface for repository hosting, along with additional features like issue tracking, code reviews, and project management tools. GitHub enhances collaboration by allowing developers to share repositories, contribute to open-source projects, and integrate with various services to streamline the development workflow.



Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



Git Commands(<https://git-scm.com/docs-all> commands) **Git**

Commands:



Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)



```
PS C:\Users\djsce.student\Desktop\devops exp\expl> git clone https://github.com/Nair-Abhinav/restful_apis.git
Cloning into 'restful_apis'...
remote: Enumerating objects: 2159, done.
remote: Compressing objects: 100% (1785/1785), done.
remote: Total 2159 (delta 306), reused 2159 (delta 306), pack-reused 0 (from 0)
Updating files: 100% (1909/1909), done.
fatal: not a git repository (or any of the parent directories): .git
PS C:\Users\djsce.student\Desktop\devops exp\expl> git checkout main
fatal: not a git repository (or any of the parent directories): .git
PS C:\Users\djsce.student\Desktop\devops exp\expl> cd .\restful_apis\
PS C:\Users\djsce.student\Desktop\devops exp\expl\restful_apis> git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
PS C:\Users\djsce.student\Desktop\devops exp\expl\restful_apis> git checkout main
Already on 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\djsce.student\Desktop\devops exp\expl\restful_apis> git fetch
PS C:\Users\djsce.student\Desktop\devops exp\expl\restful_apis> git merge
Already up to date.
PS C:\Users\djsce.student\Desktop\devops exp\expl\restful_apis> git pull
Already up to date.
PS C:\Users\djsce.student\Desktop\devops exp\expl\restful_apis>
```

Commits

main	dikshavehal	All time
Commits on Jan 26, 2025		
remove testSphere	dikshavehal committed 4 days ago	535c616
Practs arrangement	dikshavehal committed 5 days ago	2f9c44d
Commits on Jan 25, 2025		
Merge branch 'main' of https://github.com/Celestial-Rouge/TestSphere-DepartmentProject	dikshavehal committed 5 days ago	8b57e83
hodpage	dikshavehal committed 5 days ago	8a955ff
Hodpage terms and conditions	dikshavehal committed last week	1f83998
TimeTable Practicals subjects	dikshavehal committed last week	c83493c
Commits on Jan 24, 2025		
Save practical data	dikshavehal committed last week	7c6a00c
Commits on Jan 23, 2025		
StudentSearch tabs update	dikshavehal committed last week	b6feb71
Commits on Jan 19, 2025		
Add select year in StudentSearch	dikshavehal committed 2 weeks ago	e8e33c4
Commits on Jan 18, 2025		

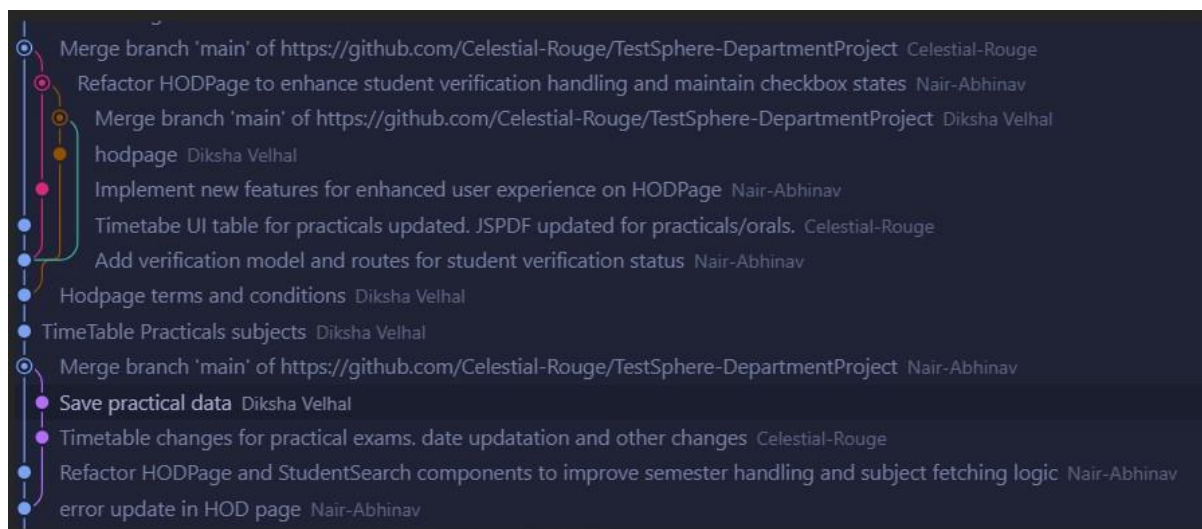


Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)





Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



MINGW64:/c/Users/djsce.student/Desktop/HN/FixIt-Code-Debugging-AI

DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (dbhr)

\$ git log

commit 3178932081c9415a11ae5cd401878a89994c6645 (HEAD -> dbhr)

Author: DJSCE Student <DJSCE.Student@SVKMGRP.COM>

Date: Fri Jan 31 09:07:44 2025 +0530

Tryitout

commit e5e2ac94f16ab6976cba07cc1baa09ba6336aa3d (origin/main, origin/HEAD, main)

Author: Dikshant <dikshantbadawadagi@gmail.com>

Date: Fri Sep 6 01:25:41 2024 +0530

second commit

commit 7d656b3e8f0774f23928eded6840f12289049bd8

Author: Dikshant <dikshantbadawadagi@gmail.com>

Date: Fri Sep 6 01:20:58 2024 +0530

first commit

DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (dbhr)

\$ git push

fatal: The current branch dbhr has no upstream branch.

To push the current branch and set the remote as upstream, use

git push --set-upstream origin dbhr

DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (dbhr)

\$ git push --set-upstream origin dbhr

fatal: helper error (-1): authentication prompt was canceled

error: unable to read askpass response from 'C:/Program Files/Git/mingw64/libexec/git-core/git-gui--

Username for 'https://github.com':

DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (dbhr)

\$ git fetch

DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (dbhr)

\$ git checkout -b main

fatal: A branch named 'main' already exists.

DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (dbhr)

\$ git checkout main

Switched to branch 'main'

Your branch is up to date with 'origin/main'.

DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (main)

\$ git feth

git: 'feth' is not a git command. See 'git --help'.

The most similar command is

fetch

DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (main)

\$ git fetch

DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (main)

\$ git merger

git: 'merger' is not a git command. See 'git --help'.

The most similar command is

merge

DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (main)

\$ git merge

Already up to date.

DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (main)



Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



MINGW64:/c:/Users/djsce.student/Desktop/HN/FixIt-Code-Debugging-AI

```

-C                copy a branch, even if target exists
-l, --list        list branch names
--show-current    show current branch name
--create-reflog    create the branch's reflog
--edit-description edit the description for the branch
-f, --force       force creation, move/rename, deletion
--merged <commit> print only branches that are merged
--no-merged <commit> print only branches that are not merged
--column[=<style>] list branches in columns
--sort <key>       field name to sort on
--points-at <object> print only branches of the object
-i, --ignore-case  sorting and filtering are case insensitive
--format <format>  format to use for the output

```

```
DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (main)
$
```

```
DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (main)
$ git branch dbhr
```

```
DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.
```

```
nothing to commit, working tree clean
```

```
DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (main)
$ git checkout dbhr
Switched to branch 'dbhr'
```

```
DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (dbhr)
$ git status
On branch dbhr
nothing to commit, working tree clean
```

```
DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (dbhr)
$ git add .
```

```
DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (dbhr)
$ git status
On branch dbhr
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   GIThelp/hello.txt
```

```
DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (dbhr)
$ git commit -m "Tryitout"
[dbhr 3178932] Tryitout
Committer: DJSCE Student <DJSCE.Student@SVKMGRP.COM>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:
```

```
git config --global --edit
```

```
After doing this, you may fix the identity used for this commit with:
```

```
git commit --amend --reset-author
```

```
1 file changed, 2 insertions(+)
create mode 100644 GIThelp/hello.txt
```



Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



MINGW64:/c/Users/djsce.student/Desktop/HN/FixIt-Code-Debugging-AI

Date: Fri Sep 6 01:25:41 2024 +0530

second commit

commit 7d656b3e8f0774f23928eded6840f12289049bd8

Author: Dikshant <dikshantbadawadagi@gmail.com>

Date: Fri Sep 6 01:20:58 2024 +0530

first commit

DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (main)

\$ git config list

error: key does not contain a section: list

DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (main)

\$ git status

On branch main

Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (main)

\$ git branch -b dbhr

error: unknown switch 'b'

```
usage: git branch [<options>] [-r | -a] [--merged] [--no-merged]
      or: git branch [<options>] [-l] [-f] <branch-name> [<start-point>]
      or: git branch [<options>] [-r] (-d | -D) <branch-name>...
      or: git branch [<options>] (-m | -M) [<old-branch>] <new-branch>
      or: git branch [<options>] (-c | -C) [<old-branch>] <new-branch>
      or: git branch [<options>] [-r | -a] [--points-at]
      or: git branch [<options>] [-r | -a] [--format]
```

Generic options

```
-v, --verbose      show hash and subject, give twice for upstream branch
-q, --quiet        suppress informational messages
-t, --track        set up tracking mode (see git-pull(1))
-u, --set-upstream-to <upstream>
                  change the upstream info
--unset-upstream   unset the upstream info
--color[=<when>]   use colored output
-r, --remotes      act on remote-tracking branches
--contains <commit> print only branches that contain the commit
--no-contains <commit>
                  print only branches that don't contain the commit
--abbrev[=<n>]     use <n> digits to display object names
```

Specific git-branch actions:

```
-a, --all          list both remote-tracking and local branches
-d, --delete       delete fully merged branch
-D                delete branch (even if not merged)
-m, --move         move/rename a branch and its reflog
-M                move/rename a branch, even if target exists
-c, --copy         copy a branch and its reflog
-C                copy a branch, even if target exists
-l, --list         list branch names
--show-current     show current branch name
--create-reflog    create the branch's reflog
--edit-description edit the description for the branch
-f, --force        force creation, move/rename, deletion
--merged <commit> print only branches that are merged
--no-merged <commit>
                  print only branches that are not merged
--column[=<style>] list branches in columns
--sort <key>       field name to sort on
--points-at <object>
                  print only branches of the object
-i, --ignore-case  sorting and filtering are case insensitive
--format <format> format to use for the output
```



Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



MINGW64:/c/Users/djsce.student/Desktop/HN/FixIt-Code-Debugging-AI

```
$ git --help
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
      [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
      [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
      [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
      <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone          Clone a repository into a new directory
  init           Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add            Add file contents to the index
  mv             Move or rename a file, a directory, or a symlink
  restore        Restore working tree files
  rm             Remove files from the working tree and from the index
  sparse-checkout Initialize and modify the sparse-checkout

examine the history and state (see also: git help revisions)
  bisect         Use binary search to find the commit that introduced a bug
  diff           Show changes between commits, commit and working tree, etc
  grep           Print lines matching a pattern
  log            Show commit logs
  show           Show various types of objects
  status         Show the working tree status

grow, mark and tweak your common history
  branch         List, create, or delete branches
  commit         Record changes to the repository
  merge          Join two or more development histories together
  rebase         Reapply commits on top of another base tip
  reset          Reset current HEAD to the specified state
  switch         Switch branches
  tag            Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
  fetch          Download objects and refs from another repository
  pull           Fetch from and integrate with another repository or a local branch
  push           Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.
```

DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (main)

\$ git log

commit e5e2ac94f16ab6976cba07cc1baa09ba6336aa3d (HEAD -> main, origin/main, origin/HEAD)

Author: Dikshant <dikshantbadawadagi@gmail.com>

Date: Fri Sep 6 01:25:41 2024 +0530

second commit

commit 7d656b3e8f0774f23928eded6840f12289049bd8

Author: Dikshant <dikshantbadawadagi@gmail.com>

Date: Fri Sep 6 01:20:58 2024 +0530

first commit

DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (main)

\$ git config list

error: key does not contain a section: list

DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (main)

\$ git status



Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



MINGW64:/c:/Users/djsce.student/Desktop/HN/FixIt-Code-Debugging-AI

DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (main)

\$ git --help

unknown option: --help

```
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
      [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
      [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
      [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
      <command> [<args>]
```

DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (main)

\$ git --help

```
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
      [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
      [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
      [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
      <command> [<args>]
```

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)

```
clone      Clone a repository into a new directory
init       Create an empty Git repository or reinitialize an existing one
```

work on the current change (see also: git help everyday)

```
add        Add file contents to the index
mv         Move or rename a file, a directory, or a symlink
restore    Restore working tree files
rm         Remove files from the working tree and from the index
sparse-checkout Initialize and modify the sparse-checkout
```

examine the history and state (see also: git help revisions)

```
bisect     Use binary search to find the commit that introduced a bug
diff       Show changes between commits, commit and working tree, etc
grep       Print lines matching a pattern
log        Show commit logs
show       Show various types of objects
status     Show the working tree status
```

grow, mark and tweak your common history

```
branch     List, create, or delete branches
commit     Record changes to the repository
merge      Join two or more development histories together
rebase     Reapply commits on top of another base tip
reset      Reset current HEAD to the specified state
switch     Switch branches
tag        Create, list, delete or verify a tag object signed with GPG
```

collaborate (see also: git help workflows)

```
fetch      Download objects and refs from another repository
pull       Fetch from and integrate with another repository or a local branch
push       Update remote refs along with associated objects
```

'git help -a' and 'git help -g' list available subcommands and some

concept guides. See 'git help <command>' or 'git help <concept>'

to read about a specific subcommand or concept.

See 'git help git' for an overview of the system.

DJSCE.Student@MUM0922CPU0316 MINGW64 ~/Desktop/HN/FixIt-Code-Debugging-AI (main)

\$ git log

commit e5e2ac94f16ab6976cba07cc1baa09ba6336aa3d (HEAD -> main, origin/main, origin/HEAD)

Author: Dikshant <dikshantbadawadagi@gmail.com>

Date: Fri Sep 6 01:25:41 2024 +0530

second commit

commit 7d656b3e8f0774f23928eded6840f12289049bd8



Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



Conclusion:

In this experiment, we understood Version Control System / Source Code Management, install git and create a GitHub account.

References:

1. How to Use Git and GitHub – Version Control Basics for Beginners (freecodecamp.org)
2. Version Control Systems - GeeksforGeeks
3. VCS Program Details - Verra