

Task 1: Configure Git

```
exam@Dac10 MINGW64 ~/Desktop/GitDemo
$ git config --global user.name "Sumit Dhamane"
```

```
exam@Dac10 MINGW64 ~/Desktop/GitDemo
$ git config --global user.email "dhamanesumit7@gmail.com"
```

```
exam@Dac10 MINGW64 ~/Desktop/GitDemo
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=Sumit Dhamane
user.email=dhamanesumit7@gmail.com
```

```
exam@Dac10 MINGW64 ~/Desktop/GitDemo
$ git config --global user.name "Sumit Dhamane"

exam@Dac10 MINGW64 ~/Desktop/GitDemo
$ git config --global user.email "dhamanesumit7@gmail.com"

exam@Dac10 MINGW64 ~/Desktop/GitDemo
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=Sumit Dhamane
user.email=dhamanesumit7@gmail.com
```

Task 2: Initialize a Local Repository

exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on

\$ git init

Initialized empty Git repository in C:/Users/exam/Desktop/GitDemo/git-hands-on/.git/

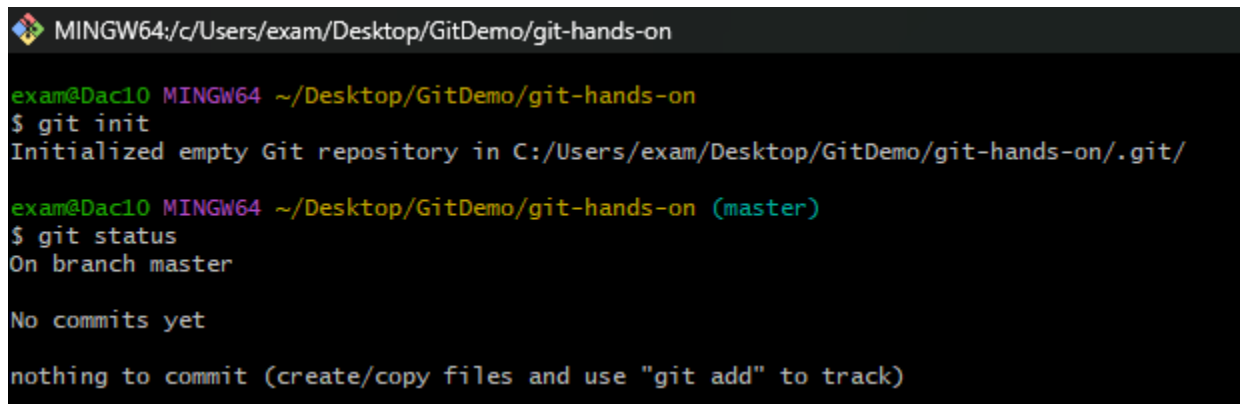
exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (master)

\$ git status

On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

A screenshot of a Windows terminal window with a dark background. The title bar shows the path 'MINGW64:/c/Users/exam/Desktop/GitDemo/git-hands-on'. The terminal text is as follows:

```
exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on
$ git init
Initialized empty Git repository in C:/Users/exam/Desktop/GitDemo/git-hands-on/.git/

exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

Task 3: Create and Track Files

exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (master)

\$ git status

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: documentation.txt

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: documentation.txt

```
exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (master)
$ git add -A
```

```
exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (master)
$ git commit -m "Description Aded"
[master (root-commit) 5f6ab61] Description Added
1 file changed, 1 insertion(+)
create mode 100644 documentation.txt
```

```
exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   documentation.txt

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:   documentation.txt

exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (master)
$ git add -A

exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (master)
$ git commit -m "Description Aded"
[master (root-commit) 5f6ab61] Description Aded
1 file changed, 1 insertion(+)
create mode 100644 documentation.txt
```

Task 4: Modify and Commit Changes

exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (master)

\$ git diff

diff --git a/documentation.txt b/documentation.txt

index 7dae4f0..e290877 100644

--- a/documentation.txt

+++ b/documentation.txt

@@ -1 +1,4 @@

-The Volunteer and Temporary Staff Allocation System is a web-based application designed to streamline the management of short-term workforce requirements for events, examinations, and similar activities. The system allows organizations to post staff requirements, enables volunteers to apply for available shifts, and supports admin-led verification, shift assignment, attendance tracking, and payment processing. By automating these processes, the platform reduces manual effort, improves coordination, and ensures transparent and efficient staff allocation.

\ No newline at end of file

+The Volunteer and Temporary Staff Allocation System is a web-based application designed to streamline the management of short-term workforce requirements for events, examinations, and similar activities. The system allows organizations to post staff requirements, enables volunteers to apply for available shifts, and supports admin-led verification, shift assignment, attendance tracking, and payment processing. By automating these processes, the platform reduces manual effort, improves coordination, and ensures transparent and efficient staff allocation.

+project by Sumit dhamane

\ No newline at end of file

exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (master)

\$ git add -A

exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (master)

\$ git commit -m "Update the description"

[master 8a1b090] Update the description

1 file changed, 4 insertions(+), 1 deletion(-)

```
exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (master)
$ git diff
diff --git a/documentation.txt b/documentation.txt
index 7dae4f0..e290877 100644
--- a/documentation.txt
+++ b/documentation.txt
@@ -1,4 @@
-The Volunteer and Temporary Staff Allocation System is a web-based application designed to streamline the management of short-term workforce requirements for events, examinations, and similar
activities. The system allows organizations to post staff requirements, enables volunteers to apply for available shifts, and supports admin-led verification, shift assignment, attendance tra
cking, and payment processing. By automating these processes, the platform reduces manual effort, improves coordination, and ensures transparent and efficient staff allocation.
\ No newline at end of file
+The Volunteer and Temporary Staff Allocation System is a web-based application designed to streamline the management of short-term workforce requirements for events, examinations, and similar
activities. The system allows organizations to post staff requirements, enables volunteers to apply for available shifts, and supports admin-led verification, shift assignment, attendance tra
cking, and payment processing. By automating these processes, the platform reduces manual effort, improves coordination, and ensures transparent and efficient staff allocation.
+
+project by Sumit dhamane
\ No newline at end of file
exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (master)
$ git add -A
exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (master)
$ git commit -m "Update the description"
[master 8a1b090] Update the description
1 file changed, 4 insertions(+), 1 deletion(-)
```

Task 5: View Commit History

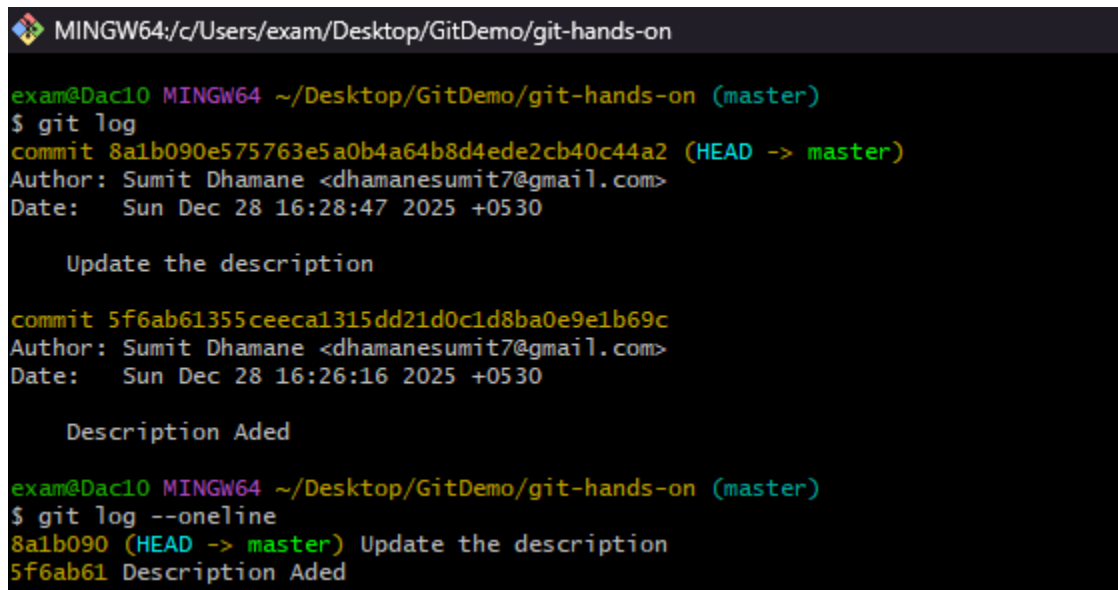
```
exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (master)
$ git log
commit 8a1b090e575763e5a0b4a64b8d4ede2cb40c44a2 (HEAD -> master)
Author: Sumit Dhamane <dhamanesumit7@gmail.com>
Date: Sun Dec 28 16:28:47 2025 +0530
```

Update the description

```
commit 5f6ab61355ceeca1315dd21d0c1d8ba0e9e1b69c
Author: Sumit Dhamane <dhamanesumit7@gmail.com>
Date: Sun Dec 28 16:26:16 2025 +0530
```

Description Aded

```
exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (master)
$ git log --oneline
8a1b090 (HEAD -> master) Update the description
5f6ab61 Description Aded
```

A screenshot of a Windows terminal window with a dark background. The title bar shows the path 'MINGW64:/c/Users/exam/Desktop/GitDemo/git-hands-on'. The terminal content shows the execution of 'git log' and 'git log --oneline' commands. The first 'git log' command shows two commits: 8a1b090e575763e5a0b4a64b8d4ede2cb40c44a2 (HEAD -> master) and 5f6ab61355ceeca1315dd21d0c1d8ba0e9e1b69c. The second 'git log --oneline' command shows the same two commits in a more concise format. The commit messages are 'Update the description' and 'Description Aded'.

```
MINGW64:/c/Users/exam/Desktop/GitDemo/git-hands-on

exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (master)
$ git log
commit 8a1b090e575763e5a0b4a64b8d4ede2cb40c44a2 (HEAD -> master)
Author: Sumit Dhamane <dhamanesumit7@gmail.com>
Date: Sun Dec 28 16:28:47 2025 +0530

    Update the description

commit 5f6ab61355ceeca1315dd21d0c1d8ba0e9e1b69c
Author: Sumit Dhamane <dhamanesumit7@gmail.com>
Date: Sun Dec 28 16:26:16 2025 +0530

    Description Aded

exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (master)
$ git log --oneline
8a1b090 (HEAD -> master) Update the description
5f6ab61 Description Aded
```

Task 6: Create and Work with Branches

```
exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (master)
$ git branch feature-1
```

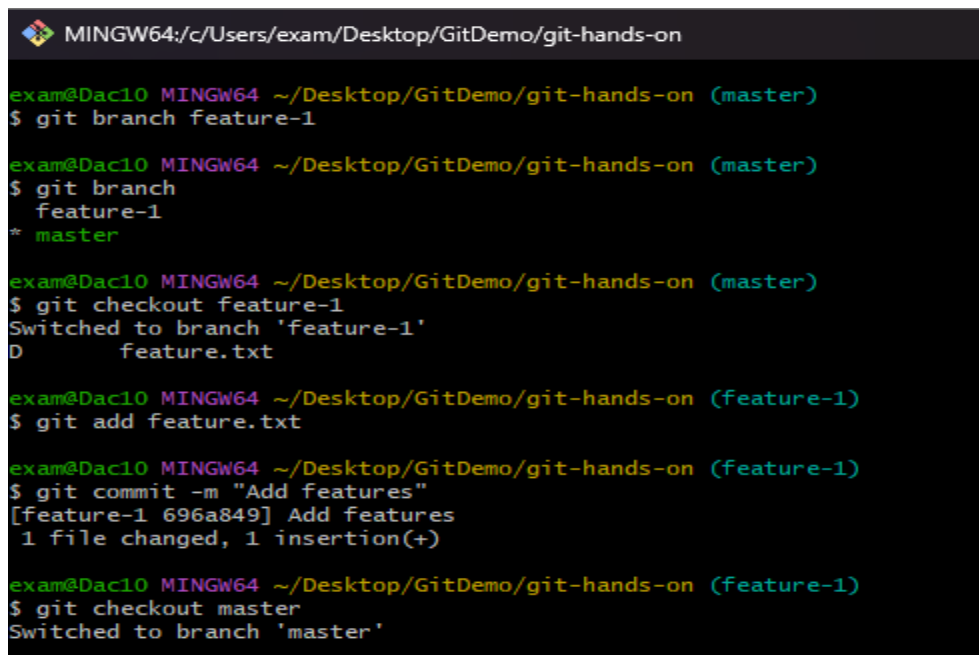
```
exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (master)
$ git branch
  feature-1
* master
```

```
exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (master)
$ git checkout feature-1
Switched to branch 'feature-1'
D    feature.txt
```

```
exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (feature-1)
$ git add feature.txt
```

```
exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (feature-1)
$ git commit -m "Add features"
[feature-1 696a849] Add features
1 file changed, 1 insertion(+)
```

```
exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (feature-1)
$ git checkout master
Switched to branch 'master'
```

A screenshot of a Windows terminal window with a dark background. The title bar shows the MINGW64 icon and the path 'c:/Users/exam/Desktop/GitDemo/git-hands-on'. The terminal displays a series of Git commands and their outputs, matching the text in the previous blocks. The prompt 'exam@Dac10' is shown in green, and the current directory and branch are shown in yellow. The output of 'git checkout' and 'git commit' is shown in white. The terminal shows the creation of a 'feature-1' branch, switching to it, adding a file, committing it, and then switching back to 'master'.

Task 7: Merge Branches

exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (main)

\$ git merge feature-1

Updating d180b13..696a849

Fast-forward

feature.txt | 1 +

1 file changed, 1 insertion(+)

exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (main)

\$ git log

commit 696a849e536d3c49d1ae33228bafa9df01d710f7 (HEAD -> main, feature-1)

Author: Sumit Dhamane <dhamanesumit7@gmail.com>

Date: Sun Dec 28 16:42:15 2025 +0530

Add features

commit d180b132e2e3aa6156857a80492bfac8036eb6f8 (origin/main)

Author: Sumit Dhamane <dhamanesumit7@gmail.com>

Date: Sun Dec 28 16:38:18 2025 +0530

Feature file added

commit 8a1b090e575763e5a0b4a64b8d4ede2cb40c44a2

Author: Sumit Dhamane <dhamanesumit7@gmail.com>

Date: Sun Dec 28 16:28:47 2025 +0530

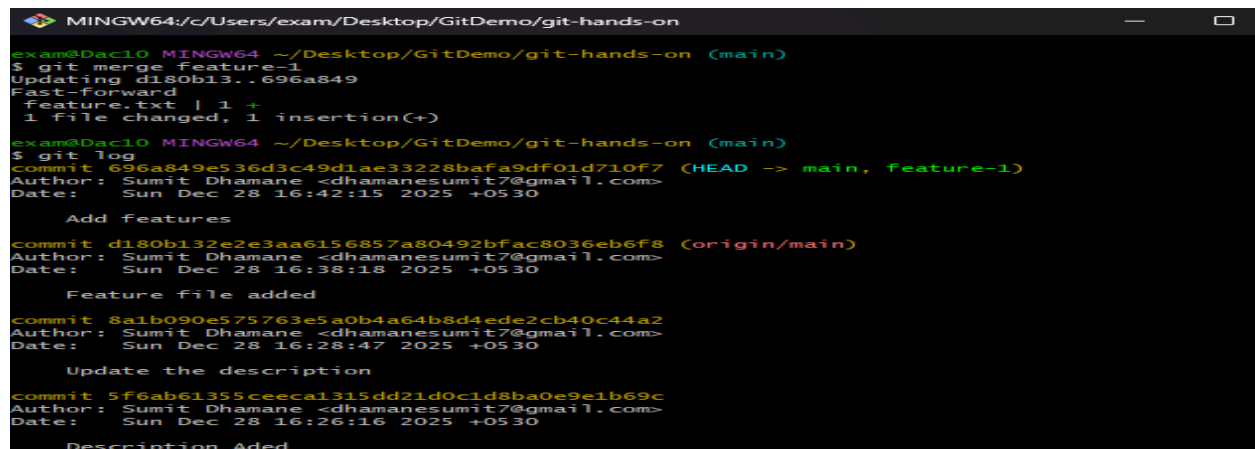
Update the description

commit 5f6ab61355ceeca1315dd21d0c1d8ba0e9e1b69c

Author: Sumit Dhamane <dhamanesumit7@gmail.com>

Date: Sun Dec 28 16:26:16 2025 +0530

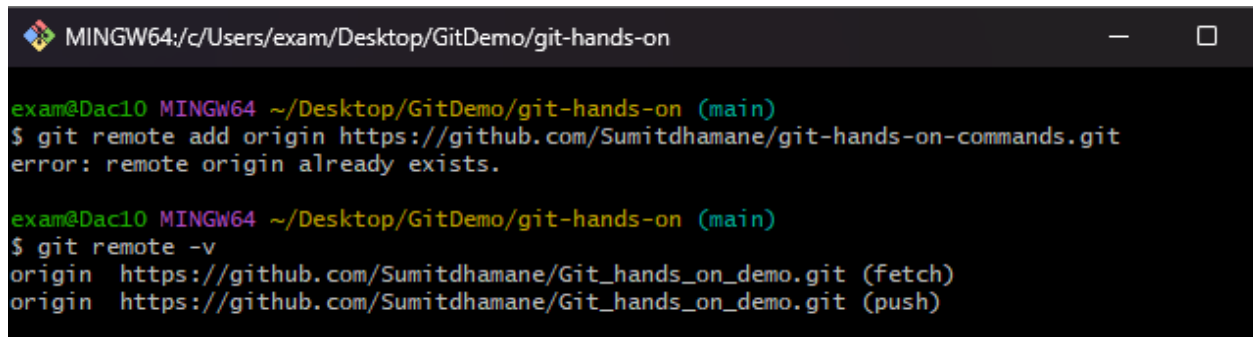
Description Added

A screenshot of a terminal window titled 'MINGW64:/c/Users/exam/Desktop/GitDemo/git-hands-on'. The terminal shows the execution of several Git commands. First, 'git merge feature-1' is run, resulting in a fast-forward merge of commit d180b13 to 696a849, adding feature.txt. Then, 'git log' is run, displaying a log of three commits: the merge commit, a commit adding features, and a commit updating the description. The terminal output is color-coded, with commit hashes in green, branch names in red, and file names in blue. The window has standard Windows taskbar icons at the top right.

Task 8: Connect to a Remote Repository

```
exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (main)
$ git remote add origin https://github.com/Sumitdhamane/git-hands-on-commands.git
error: remote origin already exists.
```

```
exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (main)
$ git remote -v
origin https://github.com/Sumitdhamane/Git_hands_on_demo.git (fetch)
origin https://github.com/Sumitdhamane/Git_hands_on_demo.git (push)
```

A screenshot of a terminal window titled "MINGW64:/c/Users/exam/Desktop/GitDemo/git-hands-on". The terminal shows the same commands and output as the previous blocks: attempting to add a remote origin that already exists, and then listing the remotes with their fetch and push URLs.

```
exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (main)
$ git remote add origin https://github.com/Sumitdhamane/git-hands-on-commands.git
error: remote origin already exists.

exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (main)
$ git remote -v
origin https://github.com/Sumitdhamane/Git_hands_on_demo.git (fetch)
origin https://github.com/Sumitdhamane/Git_hands_on_demo.git (push)
```

Task 9: Push Code to GitHub

```
exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (main)
$ git remote -v
origin https://github.com/Sumitdhamane/git_demo.git (fetch)
origin https://github.com/Sumitdhamane/git_demo.git (push)
```

```
exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (main)
$ git push -u origin main
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 12 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (12/12), 1.33 KiB | 1.33 MiB/s, done.
Total 12 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Sumitdhamane/git_demo.git
 * [new branch]    main -> main
branch 'main' set up to track 'origin/main'.
```



```
MINGW64:/c/Users/exam/Desktop/GitDemo/git-hands-on

exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (main)
$ git remote -v
origin  https://github.com/Sumitdhamane/git_demo.git (fetch)
origin  https://github.com/Sumitdhamane/git_demo.git (push)

exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (main)
$ git push -u origin main
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 12 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (12/12), 1.33 KiB | 1.33 MiB/s, done.
Total 12 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Sumitdhamane/git_demo.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Task 10: Clone a Repository

```
exam@Dac10 MINGW64 ~/Desktop/GitDemo/clone-folder
$ git clone https://github.com/Sumitdhamane/git_demo.git
Cloning into 'git_demo'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 12 (delta 1), reused 12 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (12/12), done.
Resolving deltas: 100% (1/1), done.
```

```
exam@Dac10 MINGW64 ~/Desktop/GitDemo/clone-folder
$ git clone https://github.com/Sumitdhamane/git_demo.git
Cloning into 'git_demo'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 12 (delta 1), reused 12 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (12/12), done.
Resolving deltas: 100% (1/1), done.
```

Task 11: Undo Changes

exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (main)

\$ git status

On branch main

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

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: documentation.txt

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

exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (main)

\$ git restore documentation.txt

exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (main)

\$ git status

On branch main

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

nothing to commit, working tree clean

exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (main)

\$ git status

On branch main

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

nothing to commit, working tree clean

```
MINGW64:/c/Users/exam/Desktop/GitDemo/git-hands-on

exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

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:   documentation.txt

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

exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (main)
$ git restore documentation.txt

exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Task 12: Remove Files

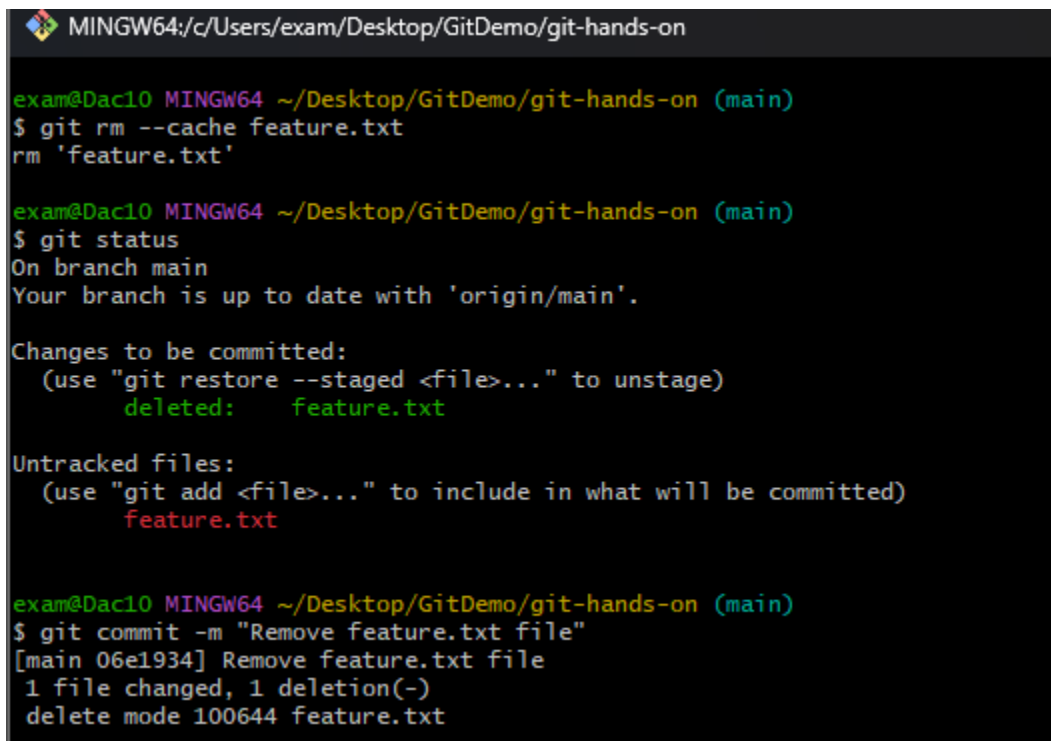
```
exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (main)
$ git rm --cache feature.txt
rm 'feature.txt'
```

```
exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.
```

```
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:   feature.txt
```

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

```
exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (main)
$ git commit -m "Remove feature.txt file"
[main 06e1934] Remove feature.txt file
1 file changed, 1 deletion(-)
delete mode 100644 feature.txt
```



```
MINGW64:/c/Users/exam/Desktop/GitDemo/git-hands-on

exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (main)
$ git rm --cache feature.txt
rm 'feature.txt'

exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    feature.txt

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

exam@Dac10 MINGW64 ~/Desktop/GitDemo/git-hands-on (main)
$ git commit -m "Remove feature.txt file"
[main 06e1934] Remove feature.txt file
1 file changed, 1 deletion(-)
delete mode 100644 feature.txt
```