

Version Control Using Git

1. Using Git Hooks for Automated Checks

Objective:

- Set up a Git hook to run scripts (like linters or tests) before commits are finalized.

Requirements:

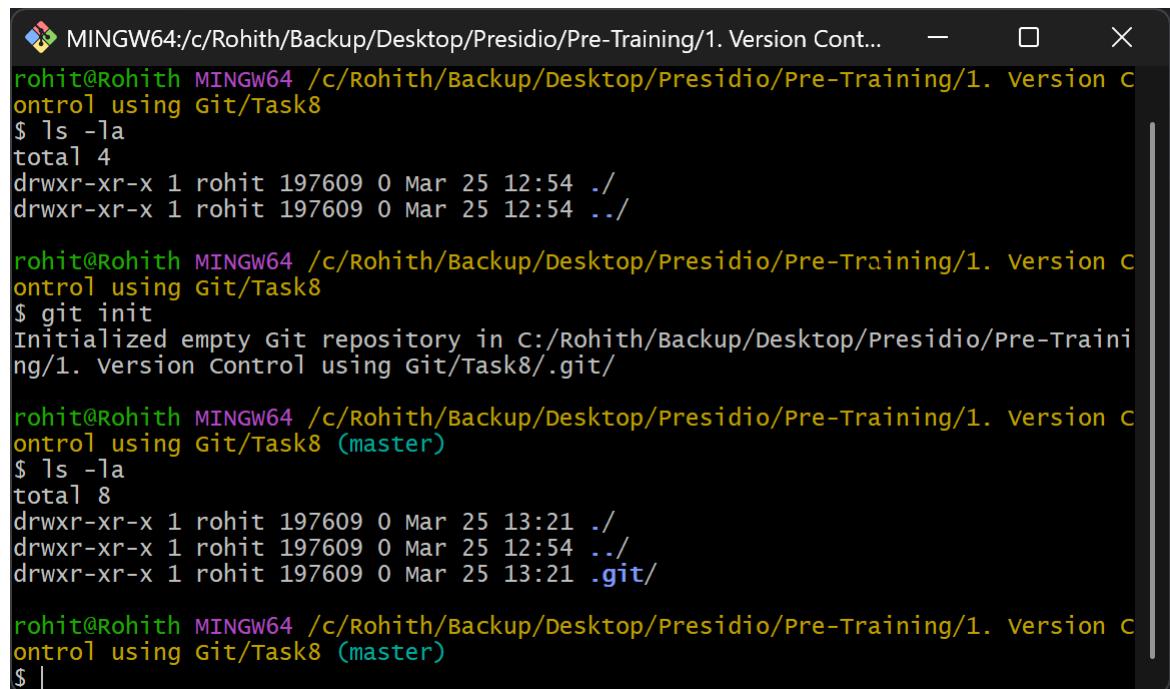
- Create a pre-commit hook in the .git/hooks directory.
- Write a simple script (e.g., a shell script or Node script) that runs a linter or a unit test.
- Ensure that if the tests or linting fail, the commit is aborted.
- Document how Git hooks can improve code quality in collaborative projects.

Git Hooks: Automating Checks Before Commit (Simple & Easy Guide)

Git hooks allow us to automate checks before commits, ensuring code quality and consistency.

Step 1: Navigate to the .git/hooks Directory

Every Git repository has a hidden .git folder. The hooks are stored in .git/hooks.



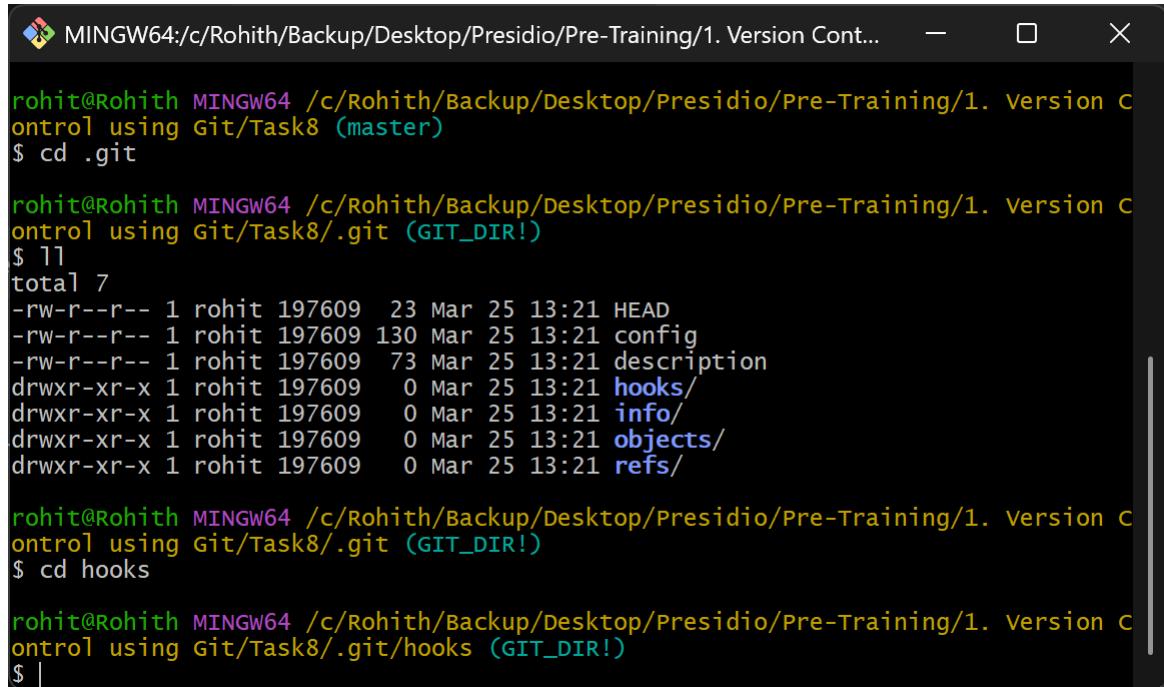
```
MINGW64:/c/Rohith/Backup/Desktop/Presidio/Pre-Training/1. Version Cont... — ×
rohit@Rohith MINGW64 /c/Rohith/Backup/Desktop/Presidio/Pre-Training/1. Version C
ontrol using Git/Task8
$ ls -la
total 4
drwxr-xr-x 1 rohit 197609 0 Mar 25 12:54 ../
drwxr-xr-x 1 rohit 197609 0 Mar 25 12:54 ./

rohit@Rohith MINGW64 /c/Rohith/Backup/Desktop/Presidio/Pre-Training/1. Version C
ontrol using Git/Task8
$ git init
Initialized empty Git repository in C:/Rohith/Backup/Desktop/Presidio/Pre-Training/1. Version Control using Git/Task8/.git/

rohit@Rohith MINGW64 /c/Rohith/Backup/Desktop/Presidio/Pre-Training/1. Version C
ontrol using Git/Task8 (master)
$ ls -la
total 8
drwxr-xr-x 1 rohit 197609 0 Mar 25 13:21 ../
drwxr-xr-x 1 rohit 197609 0 Mar 25 12:54 .//
drwxr-xr-x 1 rohit 197609 0 Mar 25 13:21 .git/

rohit@Rohith MINGW64 /c/Rohith/Backup/Desktop/Presidio/Pre-Training/1. Version C
ontrol using Git/Task8 (master)
$ |
```

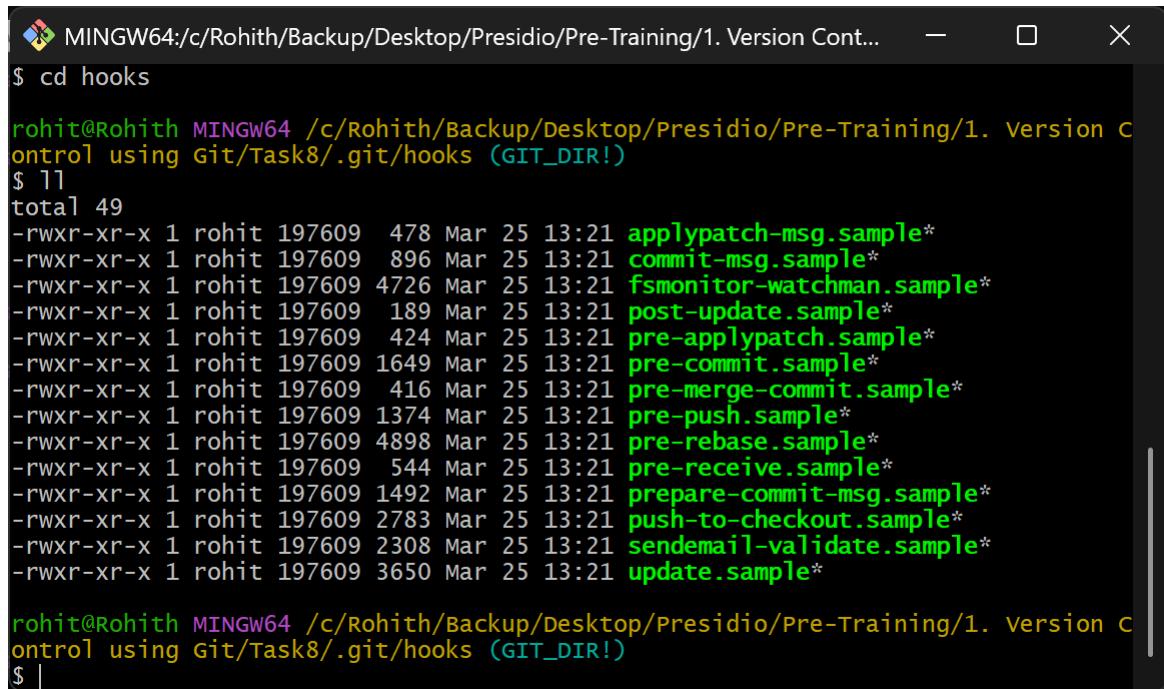
If .git/hooks is missing, **initialize Git** first:



```
MINGW64:/c/Rohith/Backup/Desktop/Presidio/Pre-Training/1. Version Control using Git/Task8 (master)
$ cd .git
rohit@Rohith MINGW64 /c/Rohith/Backup/Desktop/Presidio/Pre-Training/1. Version Control using Git/Task8 (.git (GIT_DIR!))
$ ll
total 7
-rw-r--r-- 1 rohit 197609 23 Mar 25 13:21 HEAD
-rw-r--r-- 1 rohit 197609 130 Mar 25 13:21 config
-rw-r--r-- 1 rohit 197609 73 Mar 25 13:21 description
drwxr-xr-x 1 rohit 197609 0 Mar 25 13:21 hooks/
drwxr-xr-x 1 rohit 197609 0 Mar 25 13:21 info/
drwxr-xr-x 1 rohit 197609 0 Mar 25 13:21 objects/
drwxr-xr-x 1 rohit 197609 0 Mar 25 13:21 refs/
rohit@Rohith MINGW64 /c/Rohith/Backup/Desktop/Presidio/Pre-Training/1. Version Control using Git/Task8 (.git (GIT_DIR!))
$ cd hooks
rohit@Rohith MINGW64 /c/Rohith/Backup/Desktop/Presidio/Pre-Training/1. Version Control using Git/Task8/.git/hooks (GIT_DIR!)
$ |
```

The ll command is a shortcut for listing files in a directory with **detailed information**.

To enable hook, rename pre-commit.sample to pre-commit



```
MINGW64:/c/Rohith/Backup/Desktop/Presidio/Pre-Training/1. Version Control using Git/Task8 (.git (GIT_DIR!))
$ cd hooks
rohit@Rohith MINGW64 /c/Rohith/Backup/Desktop/Presidio/Pre-Training/1. Version Control using Git/Task8/.git/hooks (GIT_DIR!)
$ ll
total 49
-rwxr-xr-x 1 rohit 197609 478 Mar 25 13:21 applypatch-msg.sample*
-rwxr-xr-x 1 rohit 197609 896 Mar 25 13:21 commit-msg.sample*
-rwxr-xr-x 1 rohit 197609 4726 Mar 25 13:21 fsmonitor-watchman.sample*
-rwxr-xr-x 1 rohit 197609 189 Mar 25 13:21 post-update.sample*
-rwxr-xr-x 1 rohit 197609 424 Mar 25 13:21 pre-applypatch.sample*
-rwxr-xr-x 1 rohit 197609 1649 Mar 25 13:21 pre-commit.sample*
-rwxr-xr-x 1 rohit 197609 416 Mar 25 13:21 pre-merge-commit.sample*
-rwxr-xr-x 1 rohit 197609 1374 Mar 25 13:21 pre-push.sample*
-rwxr-xr-x 1 rohit 197609 4898 Mar 25 13:21 pre-rebase.sample*
-rwxr-xr-x 1 rohit 197609 544 Mar 25 13:21 pre-receive.sample*
-rwxr-xr-x 1 rohit 197609 1492 Mar 25 13:21 prepare-commit-msg.sample*
-rwxr-xr-x 1 rohit 197609 2783 Mar 25 13:21 push-to-checkout.sample*
-rwxr-xr-x 1 rohit 197609 2308 Mar 25 13:21 sendemail-validate.sample*
-rwxr-xr-x 1 rohit 197609 3650 Mar 25 13:21 update.sample*
rohit@Rohith MINGW64 /c/Rohith/Backup/Desktop/Presidio/Pre-Training/1. Version Control using Git/Task8/.git/hooks (GIT_DIR!)
$ |
```

Step 2: Create a Pre-Commit Hook

Create a new pre-commit file inside .git/hooks:

```
MINGW64:/c/Rohith/Backup/Desktop/Presidio/Pre-Training/1. Version Cont... — □ ×
$ mv pre-commit.sample pre-commit

rohit@Rohith MINGW64 /c/Rohith/Backup/Desktop/Presidio/Pre-Training/1. Version C
ontrol using Git/Task8/.git/hooks (GIT_DIR!)
$ ll
total 49
-rwxr-xr-x 1 rohit 197609 478 Mar 25 13:21 applypatch-msg.sample*
-rwxr-xr-x 1 rohit 197609 896 Mar 25 13:21 commit-msg.sample*
-rwxr-xr-x 1 rohit 197609 4726 Mar 25 13:21 fsmonitor-watchman.sample*
-rwxr-xr-x 1 rohit 197609 189 Mar 25 13:21 post-update.sample*
-rwxr-xr-x 1 rohit 197609 424 Mar 25 13:21 pre-applypatch.sample*
-rwxr-xr-x 1 rohit 197609 1649 Mar 25 13:21 pre-commit*
-rwxr-xr-x 1 rohit 197609 416 Mar 25 13:21 pre-merge-commit.sample*
-rwxr-xr-x 1 rohit 197609 1374 Mar 25 13:21 pre-push.sample*
-rwxr-xr-x 1 rohit 197609 4898 Mar 25 13:21 pre-rebase.sample*
-rwxr-xr-x 1 rohit 197609 544 Mar 25 13:21 pre-receive.sample*
-rwxr-xr-x 1 rohit 197609 1492 Mar 25 13:21 prepare-commit-msg.sample*
-rwxr-xr-x 1 rohit 197609 2783 Mar 25 13:21 push-to-checkout.sample*
-rwxr-xr-x 1 rohit 197609 2308 Mar 25 13:21 sendemail-validate.sample*
-rwxr-xr-x 1 rohit 197609 3650 Mar 25 13:21 update.sample*

rohit@Rohith MINGW64 /c/Rohith/Backup/Desktop/Presidio/Pre-Training/1. Version C
ontrol using Git/Task8/.git/hooks (GIT_DIR!)
$ vi pre-commit
```

```
MINGW64:/c/Rohith/Backup/Desktop/Presidio/Pre-Training/1. Version Cont... - □ X
#!/bin/sh
#
# An example hook script to verify what is about to be committed.
# Called by "git commit" with no arguments. The hook should
# exit with non-zero status after issuing an appropriate message if
# it wants to stop the commit.
#
# To enable this hook, rename this file to "pre-commit".
#
if git rev-parse --verify HEAD >/dev/null 2>&1
then
    against=HEAD
else
    # Initial commit: diff against an empty tree object
    against=$(git hash-object -t tree /dev/null)
fi

# If you want to allow non-ASCII filenames set this variable to true.
allownonascii=$(git config --type=bool hooks.allownonascii)

# Redirect output to stderr.
exec 1>&2
pre-commit [unix] (13:21 25/03/2025) 1,1 Top
"pre-commit" [unix] 49L, 1649B
```

```
GNU nano 8.3          .git/hooks/pre-commit          Modified
#!/bin/sh
#
# An example hook script to verify what is about to be committed.
# Called by "git commit" with no arguments. The hook should
# exit with non-zero status after issuing an appropriate message if
# it wants to stop the commit.
#
# To enable this hook, rename this file to "pre-commit".
#!/bin/bash

echo "Running pre-commit checks for files..."

if git rev-parse --verify HEAD >/dev/null 2>&1
then
    against=HEAD
else
    # Initial commit: diff against an empty tree object
    against=$(git hash-object -t tree /dev/null)
fi
```

^G Help ^O Write Out ^F Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

:wq! To save and exit

:qa and enter to exit

Saving a File in Nano

1. Write the changes to the file

Press **Ctrl + O** (O for "Write Out")

- Nano will show: File Name to Write: <filename>
- If you want to keep the same name, press **Enter**.

2. Exit Nano

Press **Ctrl + X** (X for "Exit")

- If you made changes and didn't save, nano will ask:
Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES)?
- Press **Y** to confirm, then **Enter**.

MINGW64:/c/Rohith/Backup/Desktop/Presidio/Pre-Training/1. Version Control using Git/Task8 (master)

```
rohit@Rohith MINGW64 /c/Rohith/Backup/Desktop/Presidio/Pre-Training/1. Version Control using Git/Task8 (master)
$ touch test.py

rohit@Rohith MINGW64 /c/Rohith/Backup/Desktop/Presidio/Pre-Training/1. Version Control using Git/Task8 (master)
$
```

File Edit Selection View Go Run ... Task8

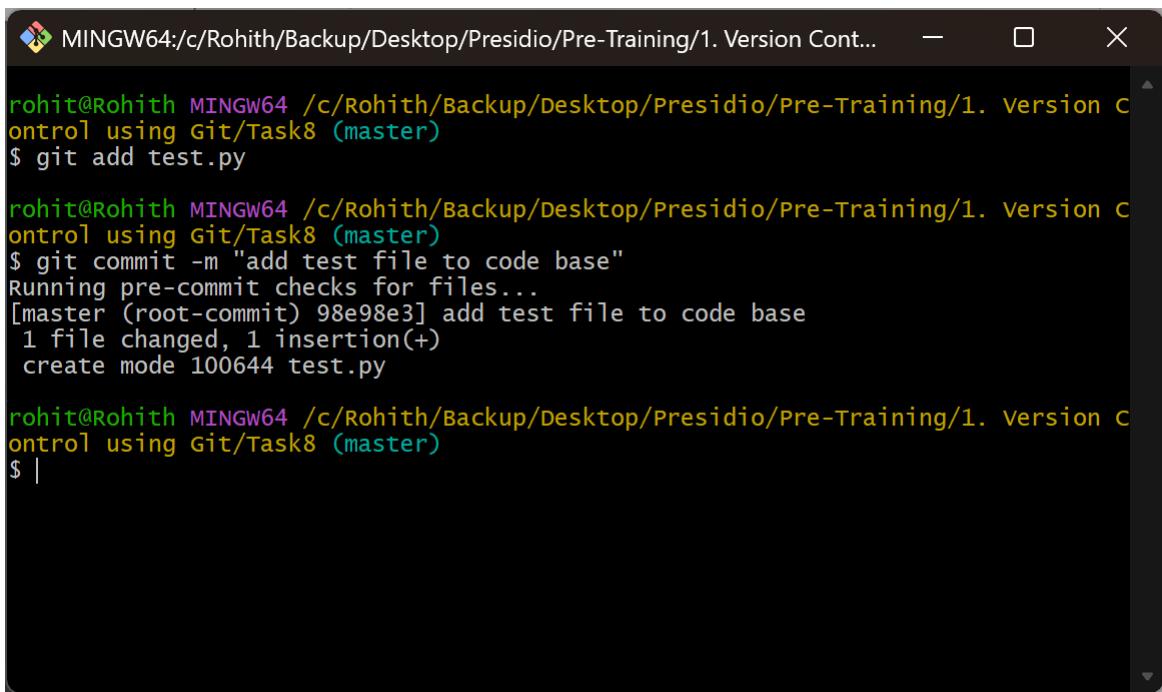
EXPLORER OPEN EDITORS ... test.py U X test.py U 1 print("Hello everyone!")

TASK8 git test.py

OUTLINE TIMELINE

Ln 1, Col 25 Spaces: 4 UTF-8 CRLF Python 3.10.7 64-bit Prettier

```
print("Hello everyone!")
```



A screenshot of a terminal window titled "MINGW64:/c/Rohith/Backup/Desktop/Presidio/Pre-Training/1. Version Cont...". The window shows a command-line session where a user named "rohit" is performing a git commit. The session starts with "git add test.py", followed by "git commit -m 'add test file to code base'". The commit message is "Running pre-commit checks for files...". The commit summary is "[master (root-commit) 98e98e3] add test file to code base". It shows "1 file changed, 1 insertion(+)" and "create mode 100644 test.py". The session ends with a final "git commit" command.

```
rohit@Rohith MINGW64 /c/Rohith/Backup/Desktop/Presidio/Pre-Training/1. Version Control using Git/Task8 (master)
$ git add test.py

rohit@Rohith MINGW64 /c/Rohith/Backup/Desktop/Presidio/Pre-Training/1. Version Control using Git/Task8 (master)
$ git commit -m "add test file to code base"
Running pre-commit checks for files...
[master (root-commit) 98e98e3] add test file to code base
 1 file changed, 1 insertion(+)
 create mode 100644 test.py

rohit@Rohith MINGW64 /c/Rohith/Backup/Desktop/Presidio/Pre-Training/1. Version Control using Git/Task8 (master)
$ |
```