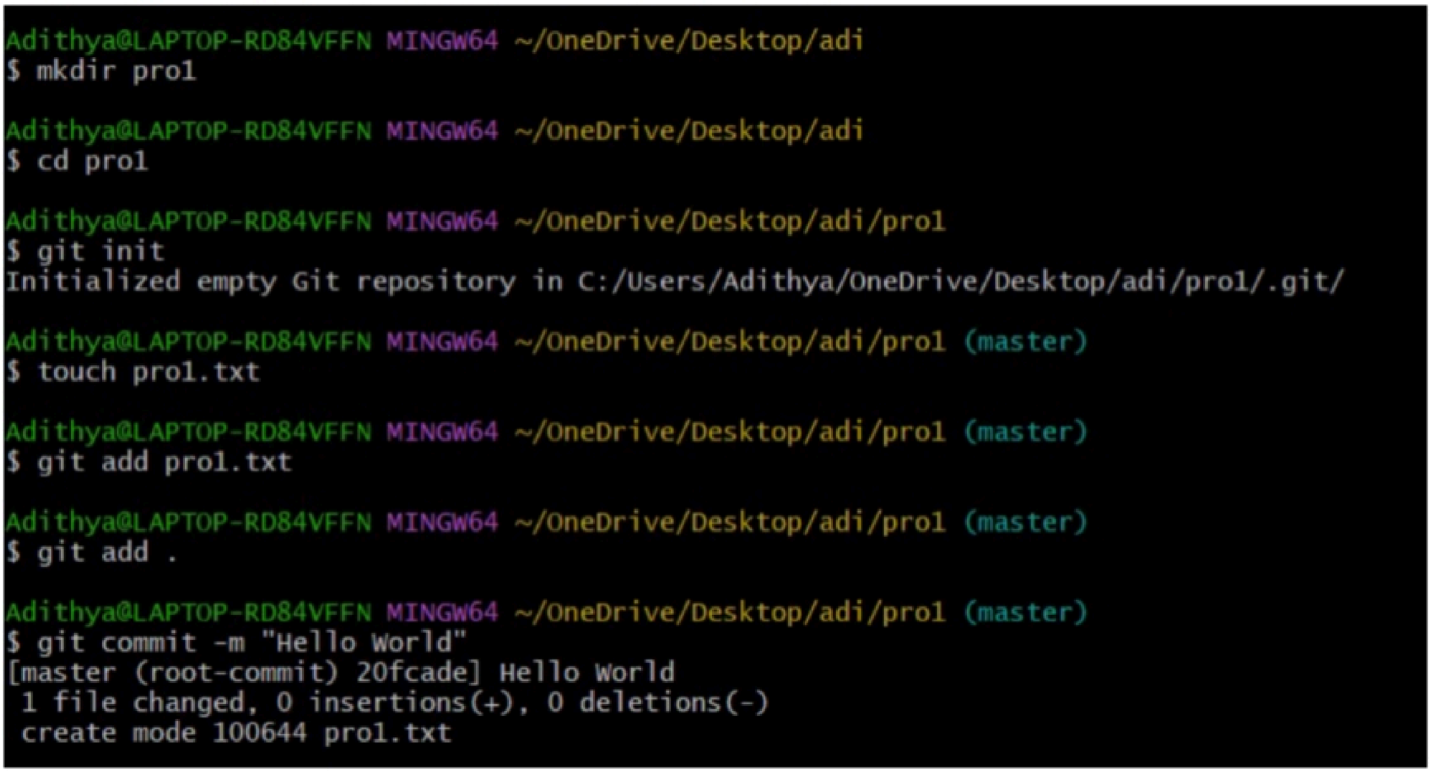
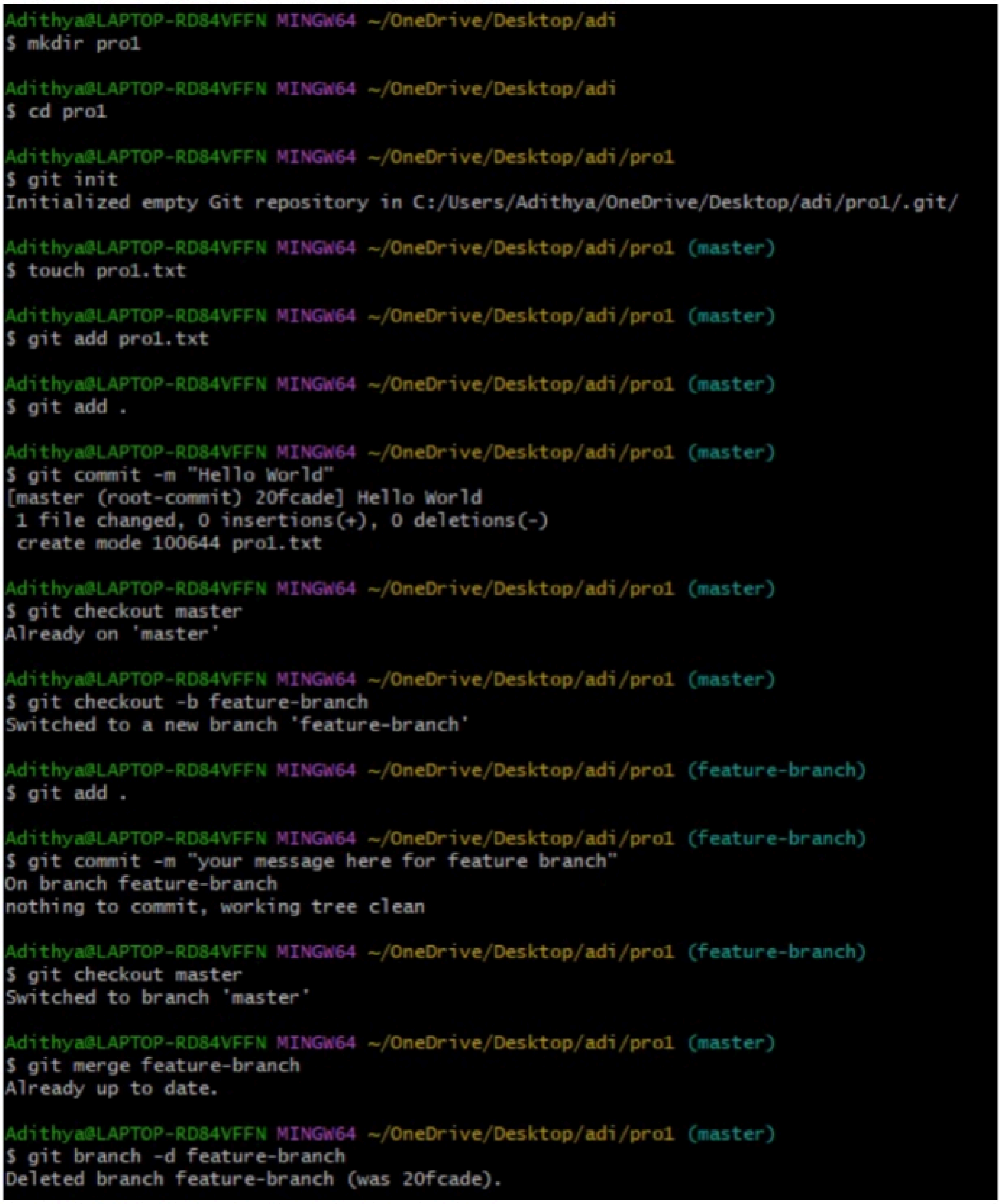
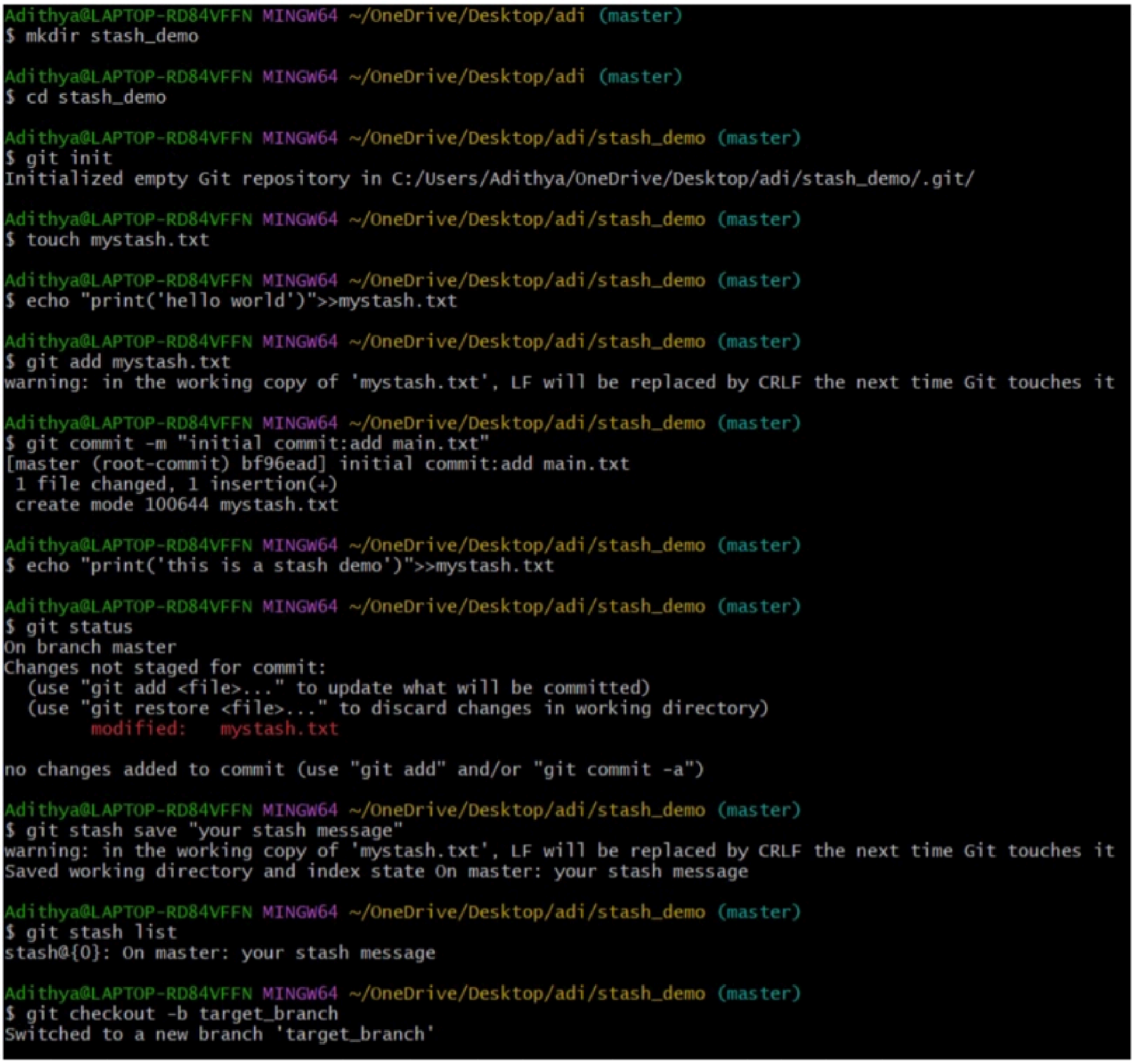


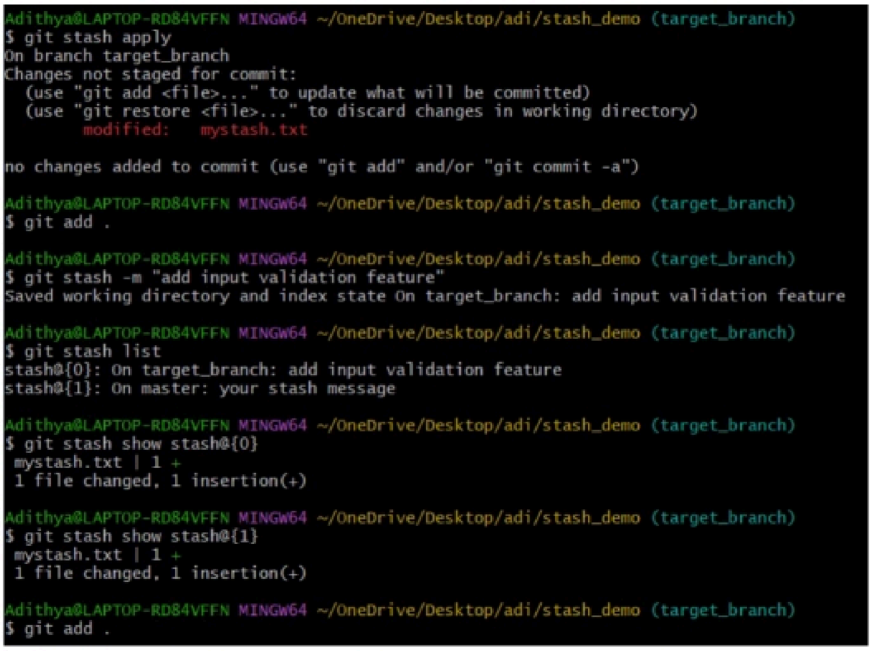
```
Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi
$ mkdir pro1

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi
$ cd pro1

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/pro1
$ git init
Initialized empty Git repository in C:/Users/Adithya/OneDrive/Desktop/adi/pro1/.git/

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/pro1 (master)
$ touch pro1.txt

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/pro1 (master)
$ git add pro1.txt

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/pro1 (master)
$ git add .

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/pro1 (master)
$ git commit -m "Hello World"
[master (root-commit) 20fcade] Hello World
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 pro1.txt
```



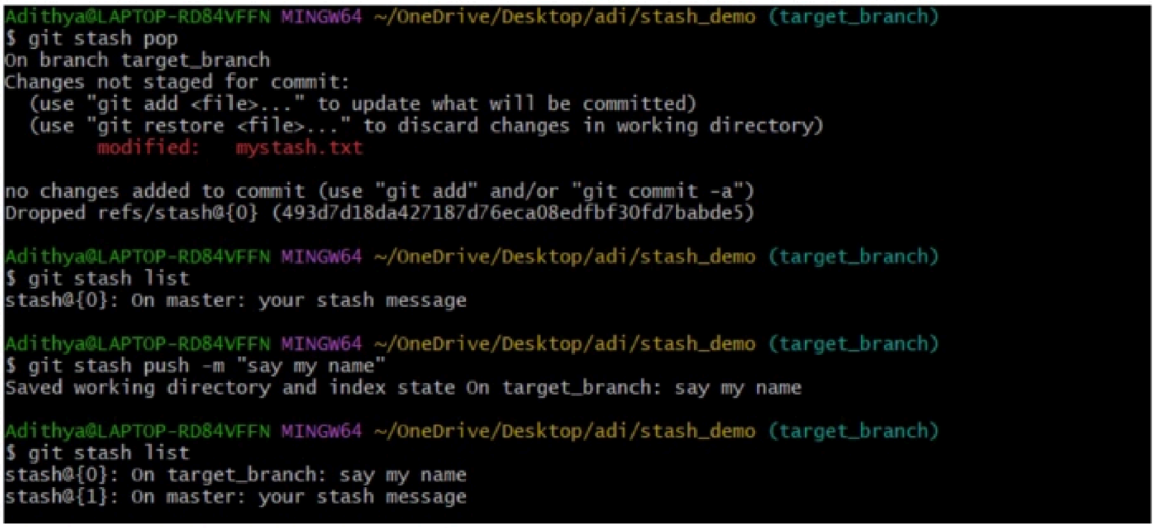
```
Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi
$ mkdir pro1

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi
$ cd pro1

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/pro1
$ git init
Initialized empty Git repository in C:/Users/Adithya/OneDrive/Desktop/adi/pro1/.git/

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/pro1 (master)
$ touch pro1.txt

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/pro1 (master)
$ git add pro1.txt

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/pro1 (master)
$ git add .

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/pro1 (master)
$ git commit -m "Hello World"
[master (root-commit) 20fcade] Hello World
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 pro1.txt

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/pro1 (master)
$ git checkout master
Already on 'master'

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/pro1 (master)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/pro1 (feature-branch)
$ git add .

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/pro1 (feature-branch)
$ git commit -m "your message here for feature branch"
On branch feature-branch
nothing to commit, working tree clean

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/pro1 (feature-branch)
$ git checkout master
Switched to branch 'master'

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/pro1 (master)
$ git merge feature-branch
Already up to date.

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/pro1 (master)
$ git branch -d feature-branch
Deleted branch feature-branch (was 20fcade).
```



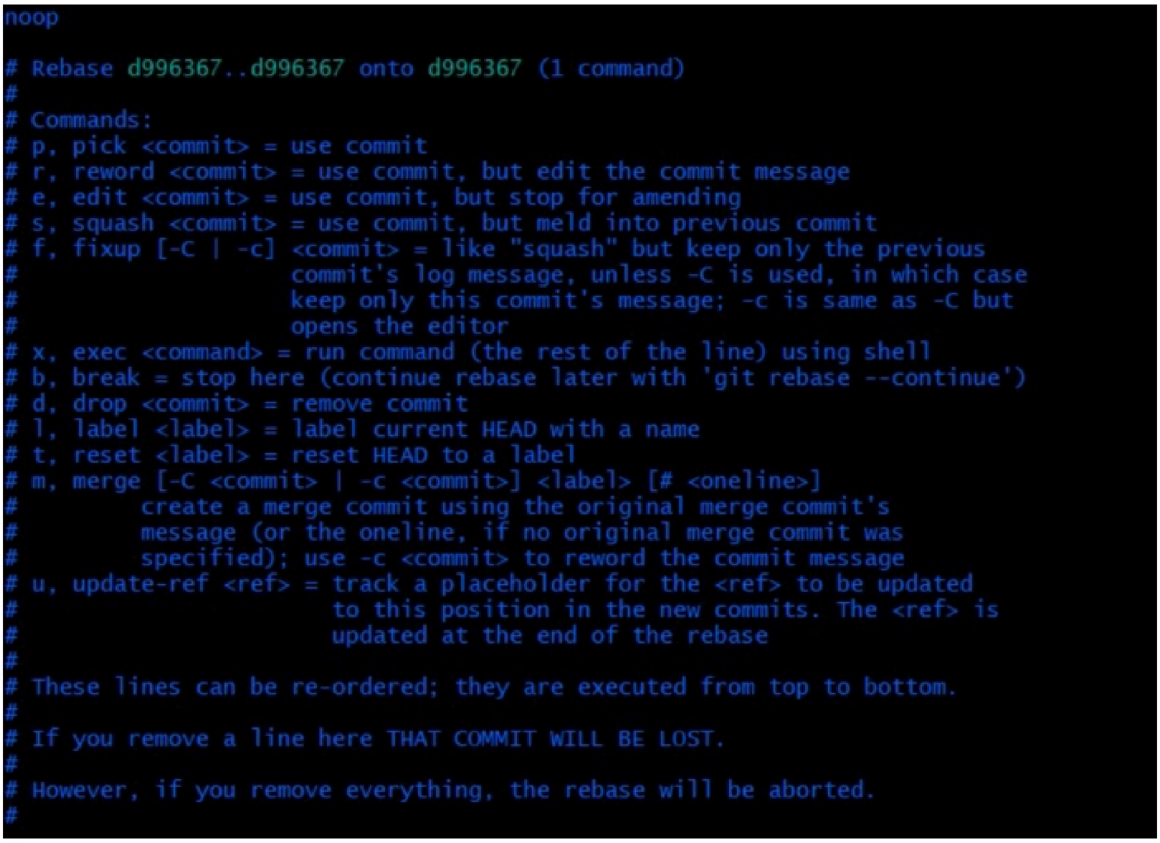
```
Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi (master)
$ mkdir stash_demo

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi (master)
$ cd stash_demo

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/stash_demo (master)
$ git init
Initialized empty Git repository in C:/Users/Adithya/OneDrive/Desktop/adi/stash_demo/.git/

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/stash_demo (master)
$ touch mystash.txt

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/stash_demo (master)
$ echo "print('hello world')">>mystash.txt

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/stash_demo (master)
$ git add mystash.txt
warning: in the working copy of 'mystash.txt', LF will be replaced by CRLF the next time Git touches it

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/stash_demo (master)
$ git commit -m "initial commit:add main.txt"
[master (root-commit) bf96ead] initial commit:add main.txt
 1 file changed, 1 insertion(+)
 create mode 100644 mystash.txt

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/stash_demo (master)
$ echo "print('this is a stash demo')">>mystash.txt

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/stash_demo (master)
$ git status
On branch master
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:   mystash.txt

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

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/stash_demo (master)
$ git stash save "your stash message"
warning: in the working copy of 'mystash.txt', LF will be replaced by CRLF the next time Git touches it
Saved working directory and index state On master: your stash message

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/stash_demo (master)
$ git stash list
stash@{0}: On master: your stash message

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/stash_demo (master)
$ git checkout -b target_branch
Switched to a new branch 'target_branch'
```



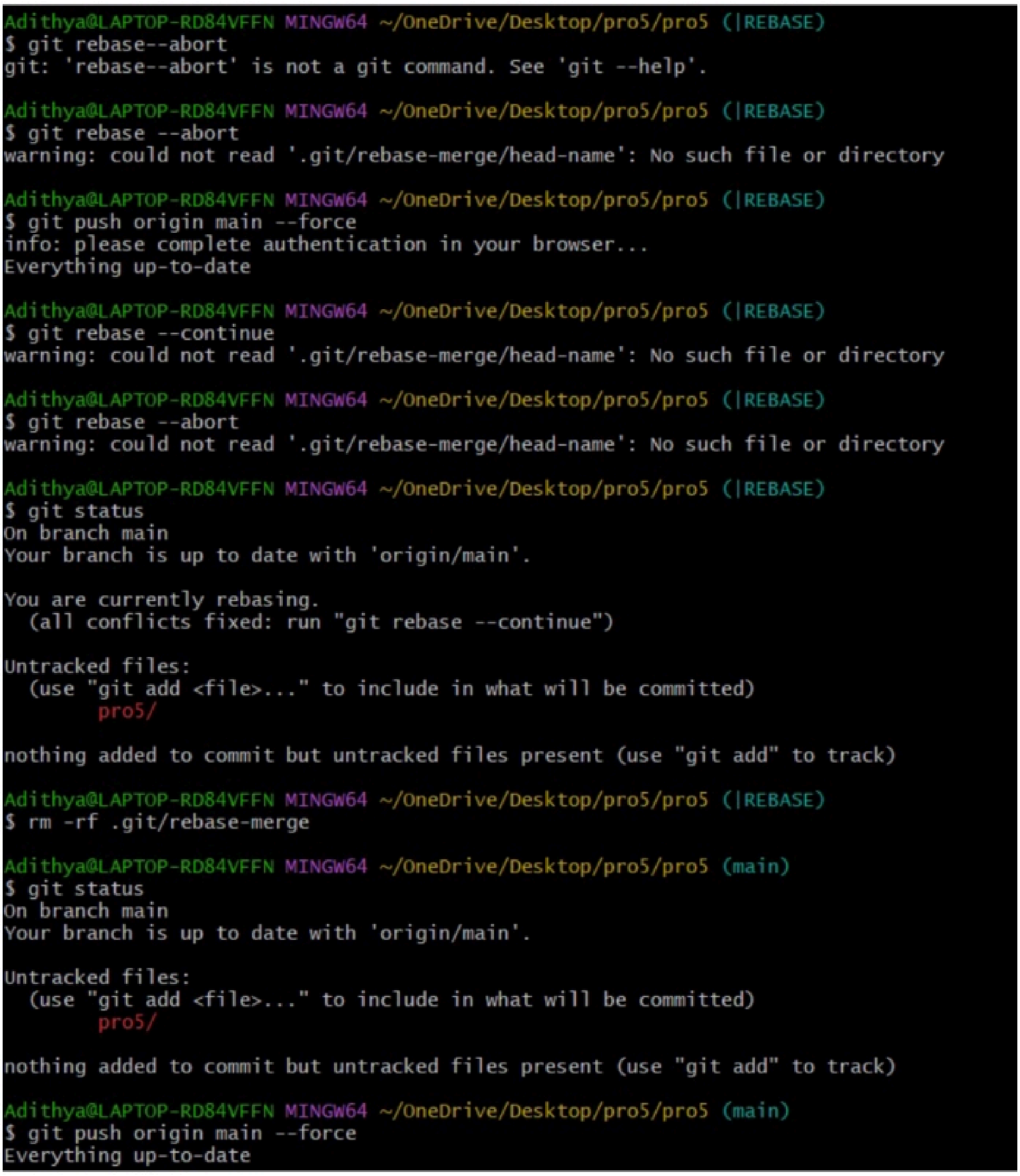
```
Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/stash_demo (target_branch)
$ git stash apply
On branch target_branch
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   mystash.txt

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

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/stash_demo (target_branch)
$ git add .

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/stash_demo (target_branch)
$ git stash -m "add input validation feature"
Saved working directory and index state On target_branch: add input validation feature

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/stash_demo (target_branch)
$ git stash list
stash@{0}: On target_branch: add input validation feature
stash@{1}: On master: your stash message

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/stash_demo (target_branch)
$ git stash show stash@{0}
 mystash.txt | 1 +
 1 file changed, 1 insertion(+)

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/stash_demo (target_branch)
$ git stash show stash@{1}
 mystash.txt | 1 +
 1 file changed, 1 insertion(+)

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/stash_demo (target_branch)
$ git add .
```



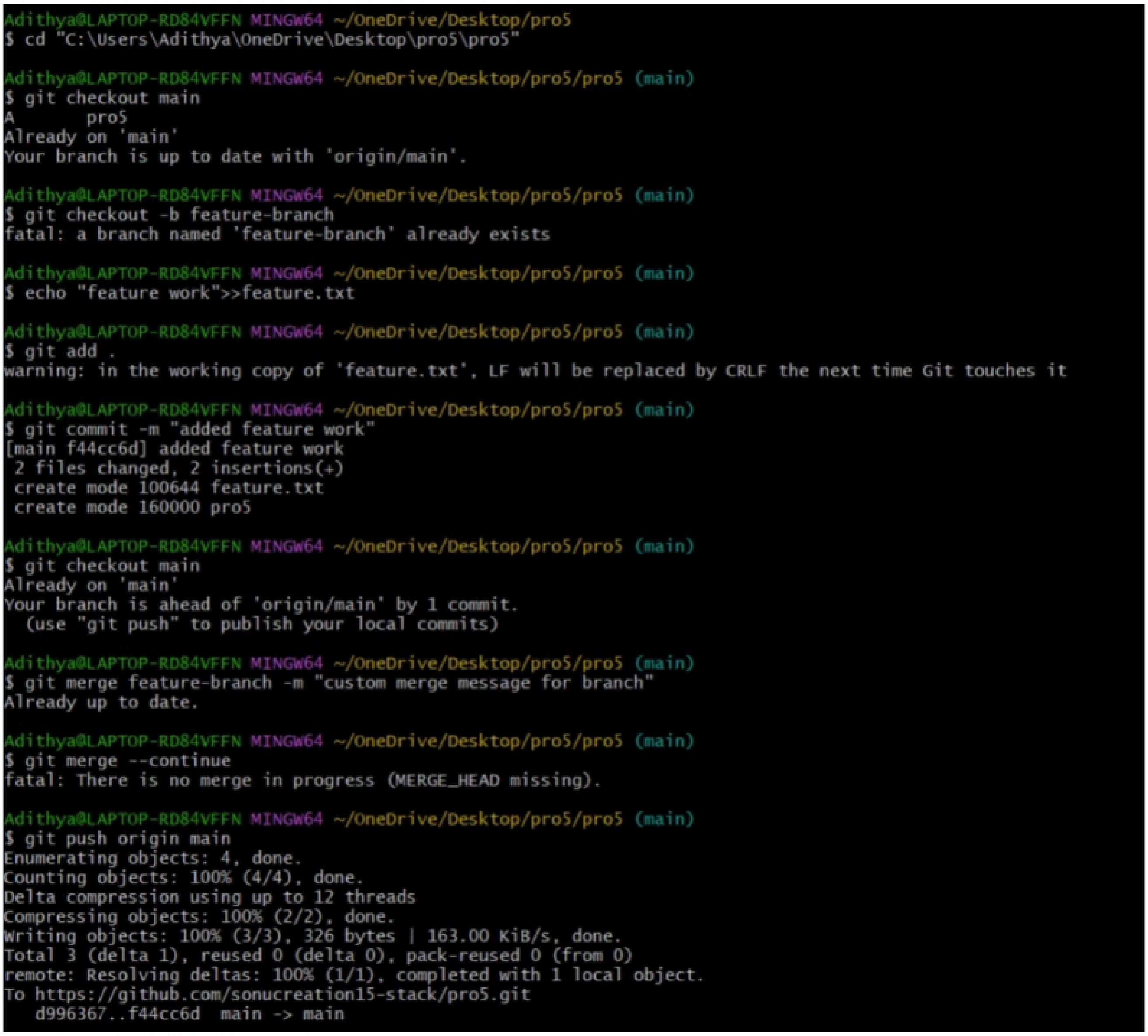
```
Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/stash_demo (target_branch)
$ git stash pop
On branch target_branch
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   mystash.txt

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (493d7d18da427187d76eca08edfbf30fd7babde5)

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/stash_demo (target_branch)
$ git stash list
stash@{0}: On master: your stash message

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/stash_demo (target_branch)
$ git stash push -m "say my name"
Saved working directory and index state On target_branch: say my name

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/adi/stash_demo (target_branch)
$ git stash list
stash@{0}: On target_branch: say my name
stash@{1}: On master: your stash message
```



```
noop

# Rebase d996367..d996367 onto d996367 (1 command)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup [-C | -c] <commit> = like "squash" but keep only the previous
#                    commit's log message, unless -C is used, in which case
#                    keep only this commit's message; -c is same as -C but
#                    opens the editor
# x, exec <command> = run command (the rest of the line) using shell
# b, break = stop here (continue rebase later with 'git rebase --continue')
# d, drop <commit> = remove commit
# l, label <label> = label current HEAD with a name
# t, reset <label> = reset HEAD to a label
# m, merge [-C <commit> | -c <commit>] <label> [# <oneline>]
#         create a merge commit using the original merge commit's
#         message (or the oneline, if no original merge commit was
#         specified); use -c <commit> to reword the commit message
# u, update-ref <ref> = track a placeholder for the <ref> to be updated
#                       to this position in the new commits. The <ref> is
#                       updated at the end of the rebase
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
#
```



```
Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/pro5/pro5 (|REBASE)
$ git rebase--abort
git: 'rebase--abort' is not a git command. See 'git --help'.

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/pro5/pro5 (|REBASE)
$ git rebase --abort
warning: could not read '.git/rebase-merge/head-name': No such file or directory

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/pro5/pro5 (|REBASE)
$ git push origin main --force
info: please complete authentication in your browser...
Everything up-to-date

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/pro5/pro5 (|REBASE)
$ git rebase --continue
warning: could not read '.git/rebase-merge/head-name': No such file or directory

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/pro5/pro5 (|REBASE)
$ git rebase --abort
warning: could not read '.git/rebase-merge/head-name': No such file or directory

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/pro5/pro5 (|REBASE)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

You are currently rebasing.
  (all conflicts fixed: run "git rebase --continue")

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        pro5/

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

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/pro5/pro5 (|REBASE)
$ rm -rf .git/rebase-merge

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/pro5/pro5 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        pro5/

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

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/pro5/pro5 (main)
$ git push origin main --force
Everything up-to-date
```



```
Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/pro5
$ cd "C:\Users\Adithya\OneDrive\Desktop\pro5\pro5"

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/pro5/pro5 (main)
$ git checkout main
A       pro5
Already on 'main'
Your branch is up to date with 'origin/main'.

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/pro5/pro5 (main)
$ git checkout -b feature-branch
fatal: a branch named 'feature-branch' already exists

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/pro5/pro5 (main)
$ echo "feature work">>feature.txt

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/pro5/pro5 (main)
$ git add .
warning: in the working copy of 'feature.txt', LF will be replaced by CRLF the next time Git touches it

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/pro5/pro5 (main)
$ git commit -m "added feature work"
[main f44cc6d] added feature work
 2 files changed, 2 insertions(+)
 create mode 100644 feature.txt
 create mode 160000 pro5

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/pro5/pro5 (main)
$ git checkout main
Already on 'main'
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/pro5/pro5 (main)
$ git merge feature-branch -m "custom merge message for branch"
Already up to date.

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/pro5/pro5 (main)
$ git merge --continue
fatal: There is no merge in progress (MERGE_HEAD missing).

Adithya@LAPTOP-RD84VFFN MINGW64 ~/OneDrive/Desktop/pro5/pro5 (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 326 bytes | 163.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/sonucreation15-stack/pro5.git
   d996367..f44cc6d  main -> main
```
]