Name: Madhuri Ramakrishnan
PRN: 119A3032
TE-IT(E2)

EXPERIMENT: 2A

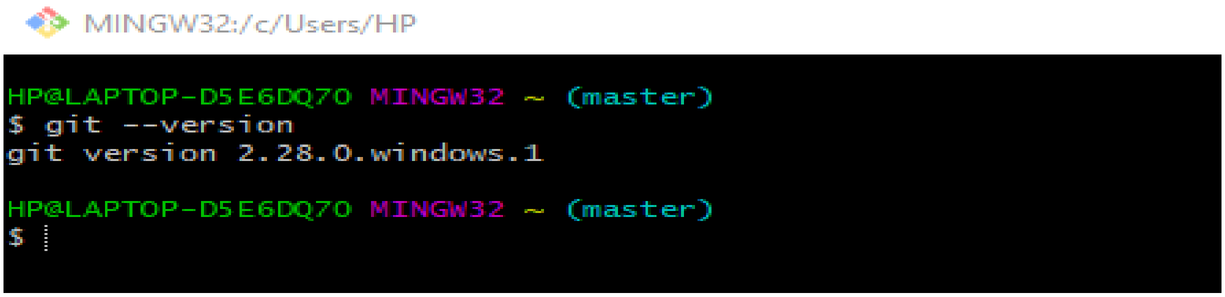
Aim: To understand the version control system / Source code management, install git and create a GitHub account.

Theory:

GitHub is a global company that provides hosting for software development version control using Git. It is a subsidiary of Microsoft, which acquired the company in 2018 for $7.5 billion. It offers all of the distributed version control and source code management(SCM) functionality of Git as well as adding its own features. It provides access control and collaboration features such as bug tracking, feature requests, task management, and wikis for every project. GitHub offers plans for free, professional, and enterprise accounts. Free GitHub accounts are commonly used to host open-source projects. As of January 2019, GitHub offers unlimited private repositories to all plans, including free accounts. As of May 2019, GitHub reports having over 37 million users and more than 100 million repositories(including at least 28 million public repositories), making it the largest host of source code in the world.

- Distributed Version Control :
  A Distributed Version Control is a type of version control where the complete codebase - including its full version history- is mirrored in every developer's computer. Changes to files are tracked between computers.

- Installation of git:
  In windows,
  Download from https://git-scm.com/
  Right-click and select GIT Bash here
  $git --version



```
HP@LAPTOP-D5E6DQ70 MINGW32 ~ (master)
$ git --version
git version 2.28.0.windows.1

HP@LAPTOP-D5E6DQ70 MINGW32 ~ (master)
$
```

In Ubuntu,
$sudo apt install git
Once the installation is done, open the terminal in Ubuntu and confirm the version by $git --version.

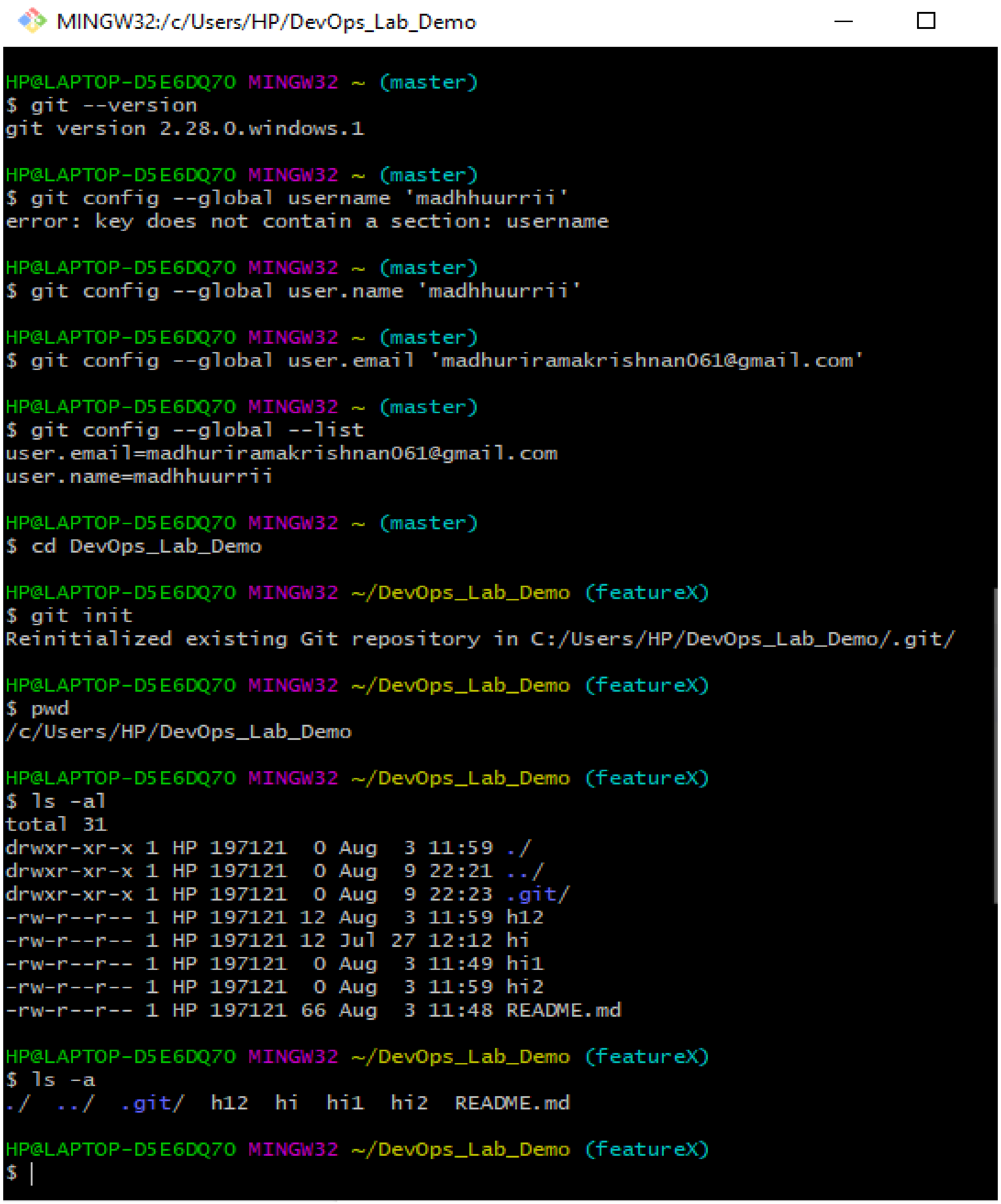
- Create a GITHUB Account :
  Go to https://github.com/
  Select Sign Up option to create a new account
  Enter your Email, Password, and Username
  Click on create account
  GITHUB Account is created!

Conclusion: Thus, we have successfully installed git into our device and created GitHub Account.

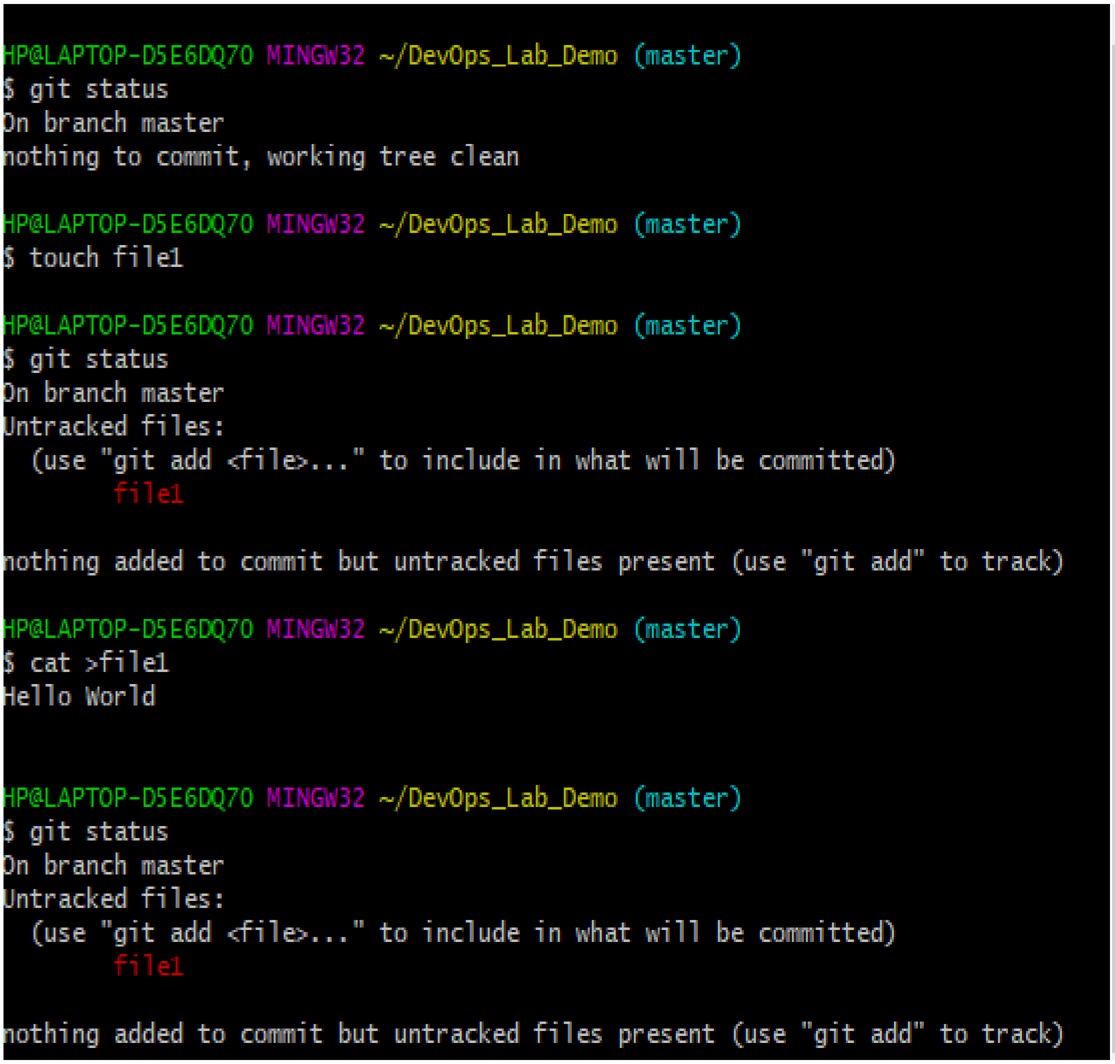
EXPERIMENT: 2B

Aim: To perform various GIT operations on local repository using GIT cheat sheet

Initializing a local repository and configuring username and email :

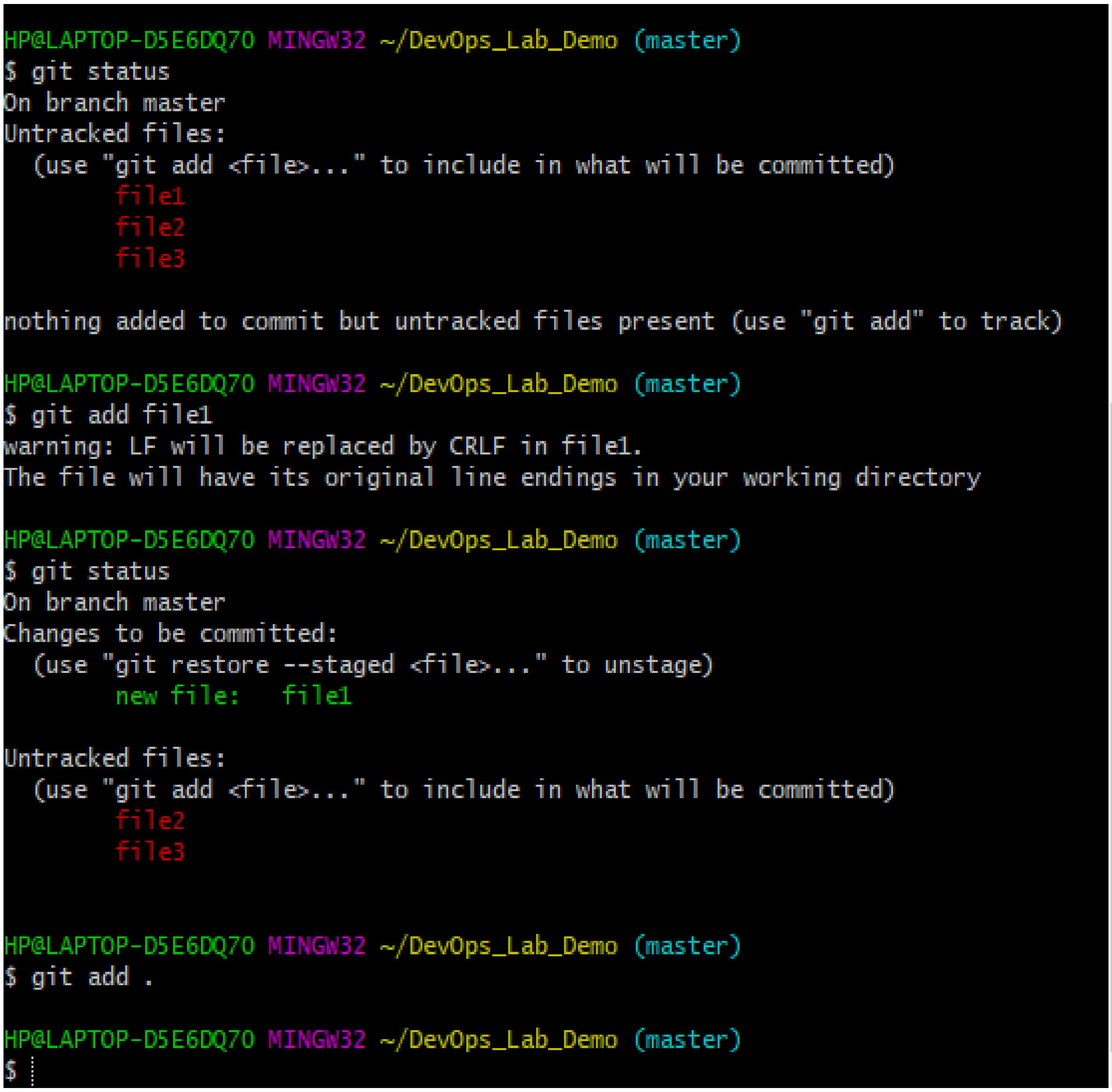


```
HP@LAPTOP-D5E6DQ70 MINGW32 ~ (master)
$ git --version
git version 2.28.0.windows.1

HP@LAPTOP-D5E6DQ70 MINGW32 ~ (master)
$ git config --global username 'madhhuurrii'
error: key does not contain a section: username

HP@LAPTOP-D5E6DQ70 MINGW32 ~ (master)
$ git config --global user.name 'madhhuurrii'

HP@LAPTOP-D5E6DQ70 MINGW32 ~ (master)
$ git config --global user.email 'madhuriramakrishnan061@gmail.com'

HP@LAPTOP-D5E6DQ70 MINGW32 ~ (master)
$ git config --global --list
user.email=madhuriramakrishnan061@gmail.com
user.name=madhhuurrii

HP@LAPTOP-D5E6DQ70 MINGW32 ~ (master)
$ cd DevOps_Lab_Demo

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (featureX)
$ git init
Reinitialized existing Git repository in C:/Users/HP/DevOps_Lab_Demo/.git/

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (featureX)
$ pwd
/c/Users/HP/DevOps_Lab_Demo

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (featureX)
$ ls -al
total 31
drwxr-xr-x 1 HP 197121  0 Aug  3 11:59 ./
drwxr-xr-x 1 HP 197121  0 Aug  9 22:21 ../
drwxr-xr-x 1 HP 197121  0 Aug  9 22:23 .git/
-rw-r--r-- 1 HP 197121 12 Aug  3 11:59 h12
-rw-r--r-- 1 HP 197121 12 Jul 27 12:12 hi
-rw-r--r-- 1 HP 197121  0 Aug  3 11:49 hi1
-rw-r--r-- 1 HP 197121  0 Aug  3 11:59 hi2
-rw-r--r-- 1 HP 197121 66 Aug  3 11:48 README.md

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (featureX)
$ ls -a
./  ../  .git/  h12  hi  hi1  hi2  README.md

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (featureX)
$ |
```



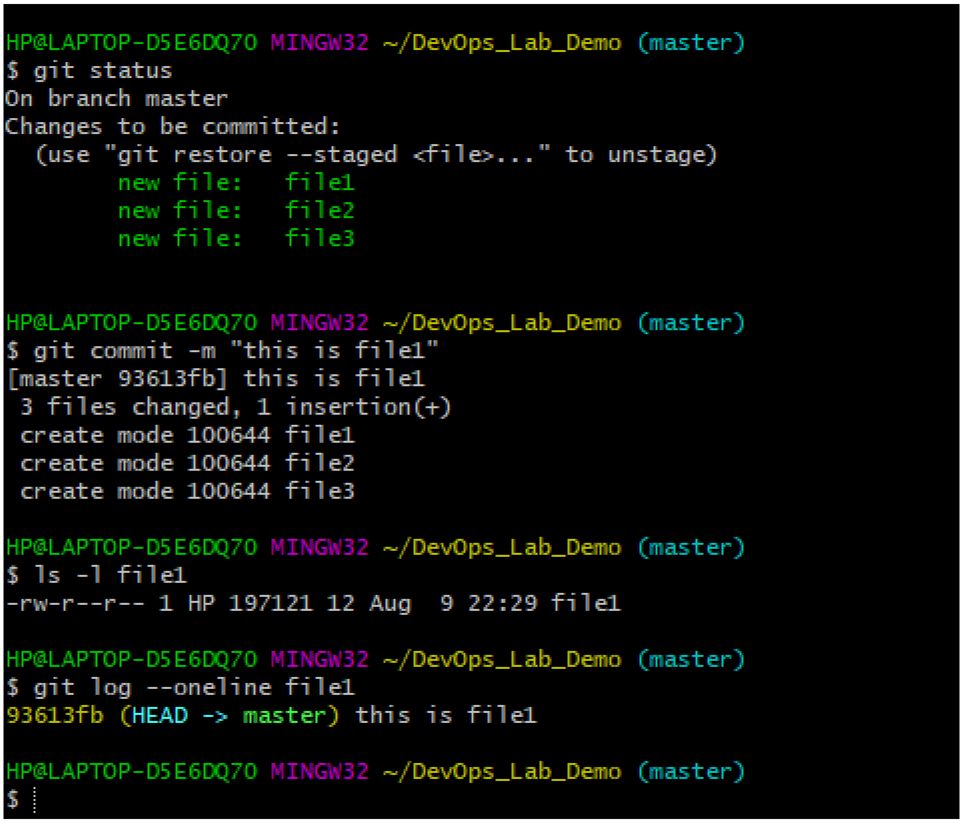
```
HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ git status
On branch master
nothing to commit, working tree clean

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ touch file1

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file1

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

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ cat >file1
Hello World


HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file1

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

Files in the unstaged area:

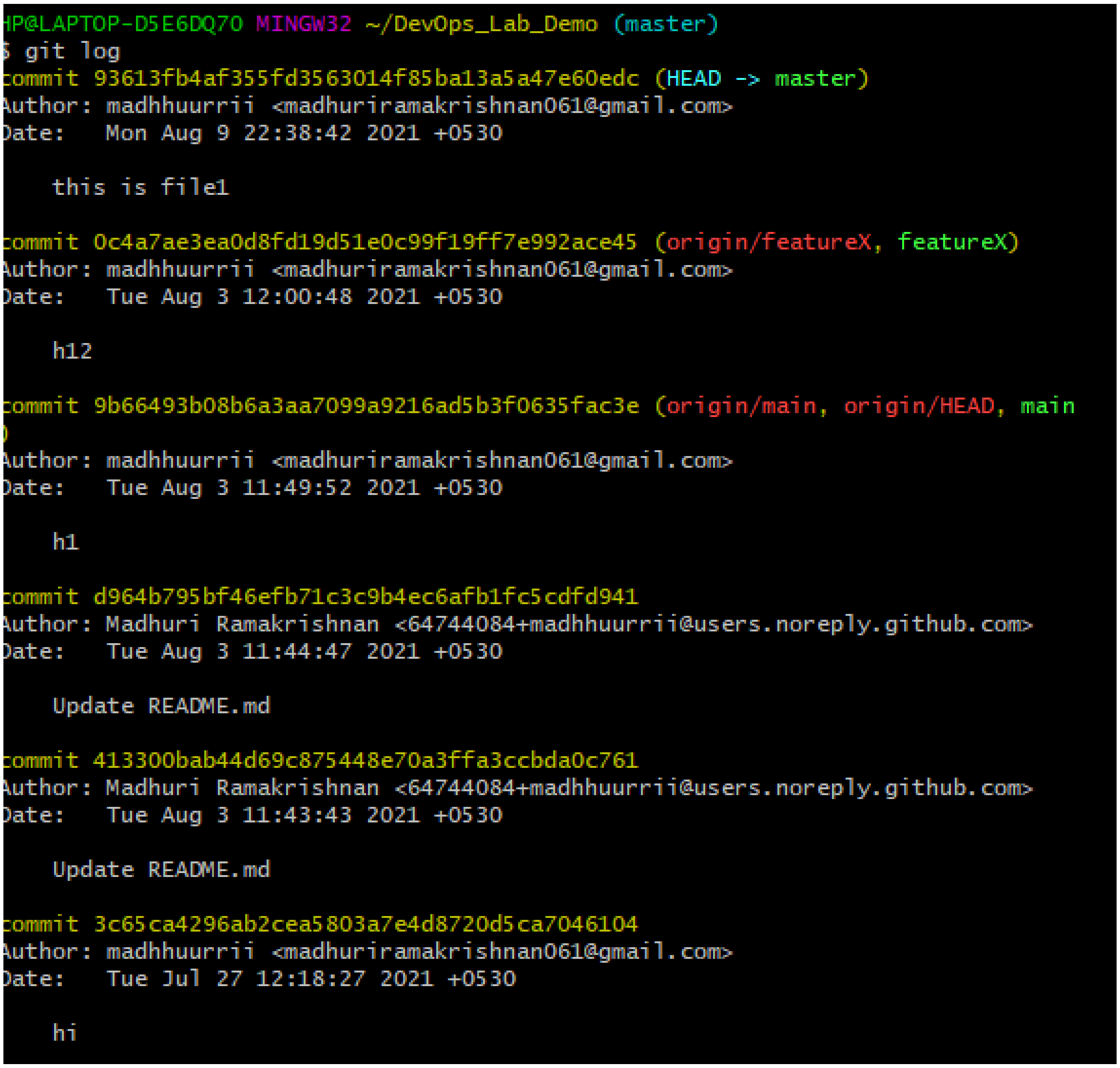


```
HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file1
        file2
        file3

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

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ git add file1
warning: LF will be replaced by CRLF in file1.
The file will have its original line endings in your working directory

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   file1

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file2
        file3


HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ git add .

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$
```

Adding File in staging area:

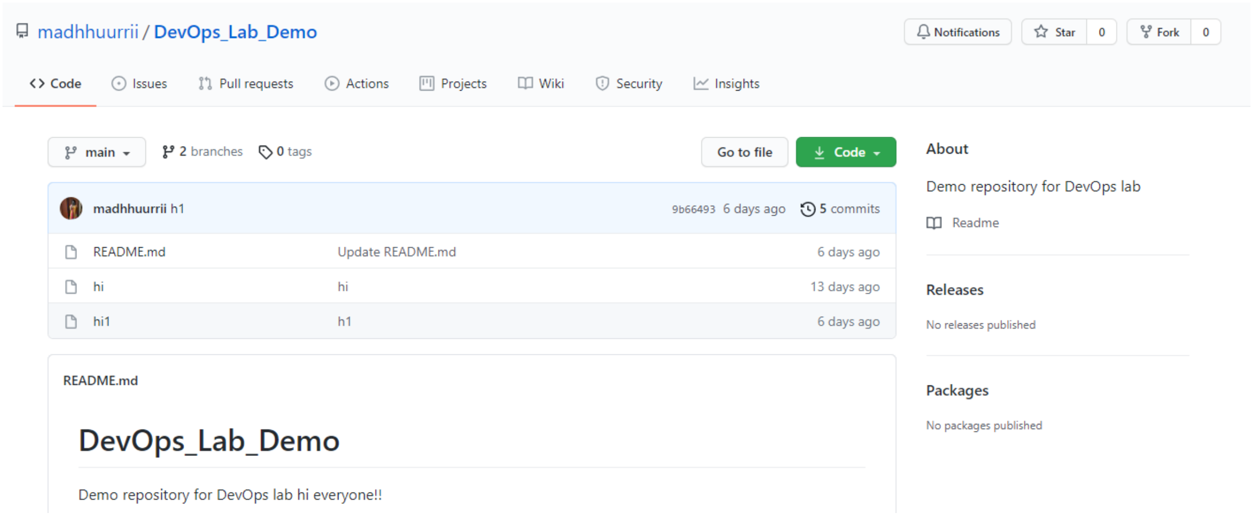


```
HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   file1
        new file:   file2
        new file:   file3


HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ git commit -m "this is file1"
[master 93613fb] this is file1
 3 files changed, 1 insertion(+)
 create mode 100644 file1
 create mode 100644 file2
 create mode 100644 file3

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ ls -l file1
-rw-r--r-- 1 HP 197121 12 Aug  9 22:29 file1

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ git log --oneline file1
93613fb (HEAD -> master) this is file1

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$
```



```
HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ git log
commit 93613fb4af355fd3563014f85ba13a5a47e60edc (HEAD -> master)
Author: madhhuurrii <madhuriramakrishnan061@gmail.com>
Date:   Mon Aug 9 22:38:42 2021 +0530

    this is file1

commit 0c4a7ae3ea0d8fd19d51e0c99f19ff7e992ace45 (origin/featureX, featureX)
Author: madhhuurrii <madhuriramakrishnan061@gmail.com>
Date:   Tue Aug 3 12:00:48 2021 +0530

    h12

commit 9b66493b08b6a3aa7099a9216ad5b3f0635fac3e (origin/main, origin/HEAD, main
)
Author: madhhuurrii <madhuriramakrishnan061@gmail.com>
Date:   Tue Aug 3 11:49:52 2021 +0530

    h1

commit d964b795bf46efb71c3c9b4ec6afb1fc5cdfd941
Author: Madhuri Ramakrishnan <64744084+madhhuurrii@users.noreply.github.com>
Date:   Tue Aug 3 11:44:47 2021 +0530

    Update README.md

commit 413300bab44d69c875448e70a3ffa3ccbda0c761
Author: Madhuri Ramakrishnan <64744084+madhhuurrii@users.noreply.github.com>
Date:   Tue Aug 3 11:43:43 2021 +0530

    Update README.md

commit 3c65ca4296ab2cea5803a7e4d8720d5ca7046104
Author: madhhuurrii <madhuriramakrishnan061@gmail.com>
Date:   Tue Jul 27 12:18:27 2021 +0530

    hi
```

Create a remote repository in GitHub :



madhhuurrii / DevOps_Lab_Demo

madhhuurrii h1 9b66493 6 days ago 5 commits

README.md	Update README.md	6 days ago
hi	hi	13 days ago
hi1	h1	6 days ago

README.md

DevOps_Lab_Demo

Demo repository for DevOps lab hi everyone!!

About

Demo repository for DevOps lab

Readme

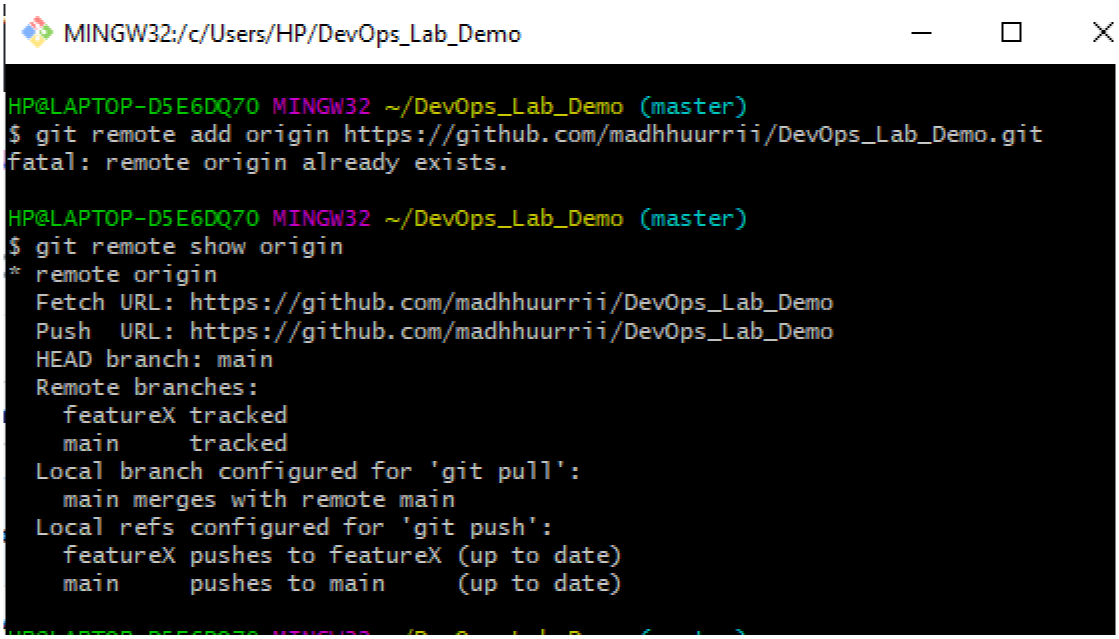
Releases

No releases published

Packages

No packages published

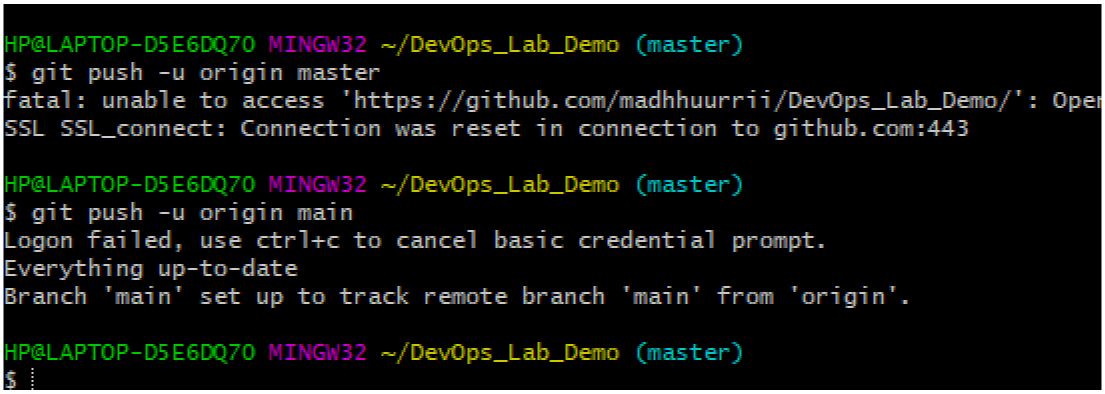
Add remote repository and showing the remote repository:



MINGW32:/c/Users/HP/DevOps_Lab_Demo

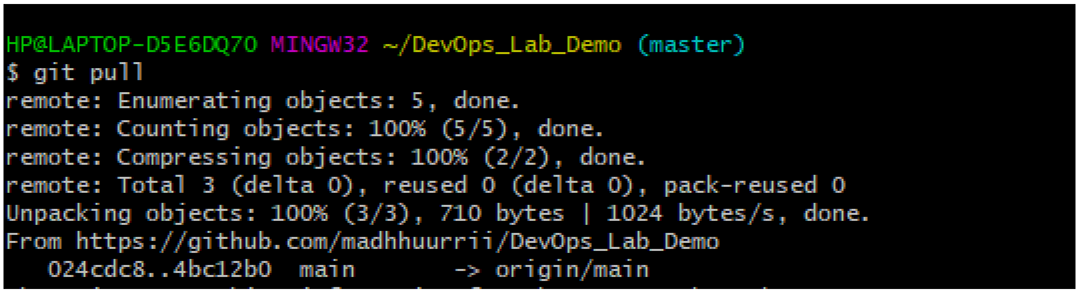
```
HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ git remote add origin https://github.com/madhhuurrii/DevOps_Lab_Demo.git
fatal: remote origin already exists.

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ git remote show origin
* remote origin
  Fetch URL: https://github.com/madhhuurrii/DevOps_Lab_Demo
  Push  URL: https://github.com/madhhuurrii/DevOps_Lab_Demo
  HEAD branch: main
  Remote branches:
    featureX tracked
    main     tracked
  Local branch configured for 'git pull':
    main merges with remote main
  Local refs configured for 'git push':
    featureX pushes to featureX (up to date)
    main     pushes to main     (up to date)
```

Pushing a remote repository :



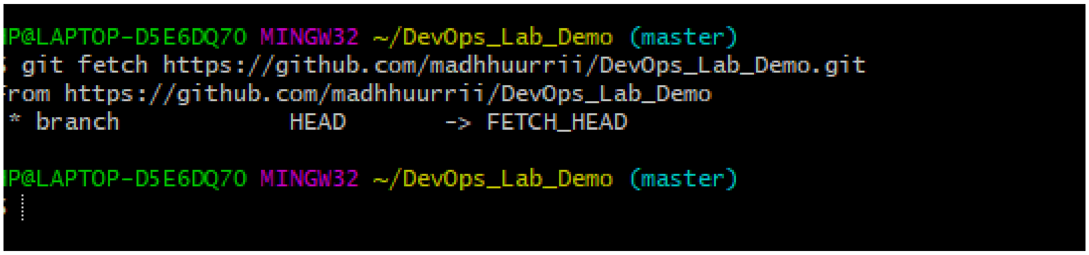
```
HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ git push -u origin master
fatal: unable to access 'https://github.com/madhhuurrii/DevOps_Lab_Demo/': Oper
SSL SSL_connect: Connection was reset in connection to github.com:443

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ git push -u origin main
Logon failed, use ctrl+c to cancel basic credential prompt.
Everything up-to-date
Branch 'main' set up to track remote branch 'main' from 'origin'.

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$
```

Pulling from remote repository:



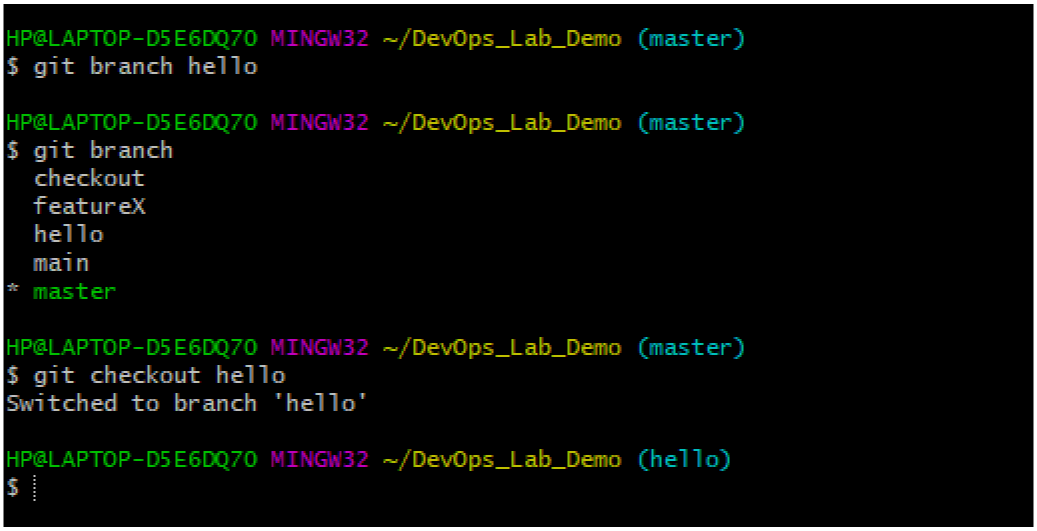
```
HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 710 bytes | 1024 bytes/s, done.
From https://github.com/madhhuurrii/DevOps_Lab_Demo
   024cdc8..4bc12b0  main       -> origin/main
```

Git fetch:



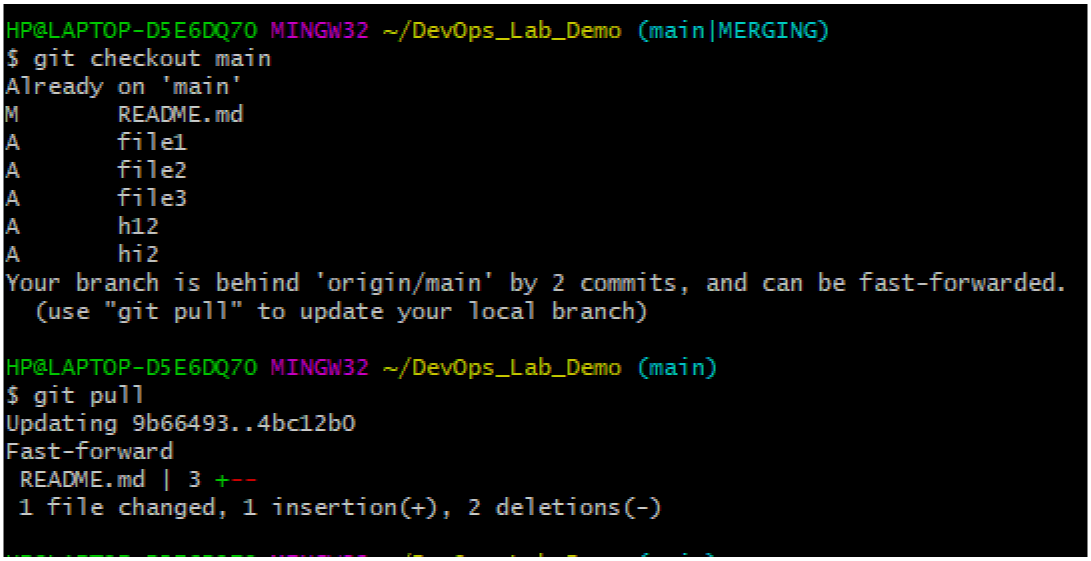
```
P@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
 git fetch https://github.com/madhhuurrii/DevOps_Lab_Demo.git
rom https://github.com/madhhuurrii/DevOps_Lab_Demo
* branch            HEAD       -> FETCH_HEAD

P@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
```

Git branch and checkout:



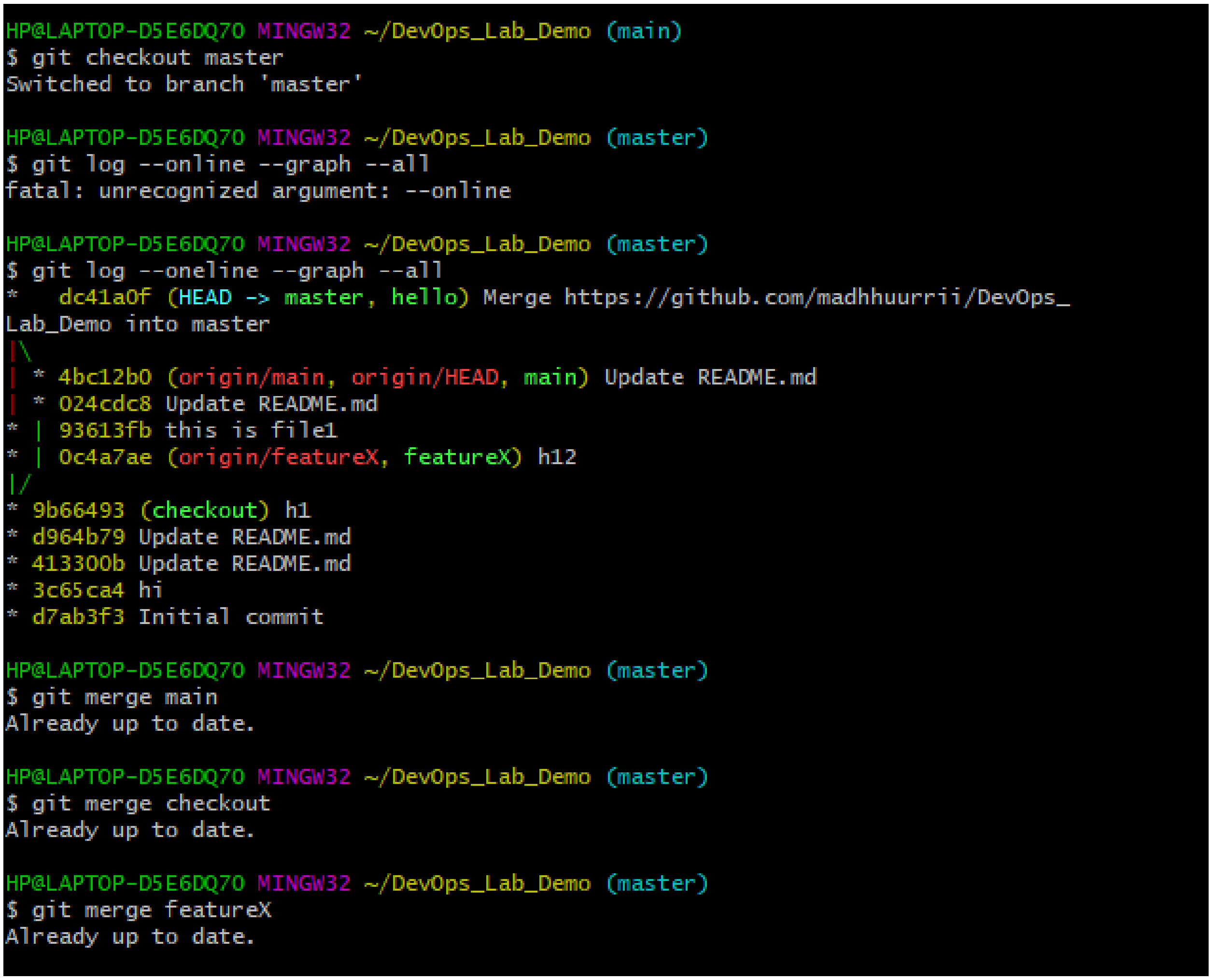
```
HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ git branch hello

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ git branch
  checkout
  featureX
  hello
  main
* master

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ git checkout hello
Switched to branch 'hello'

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (hello)
$
```

Synchronizing (local and remote) repository:



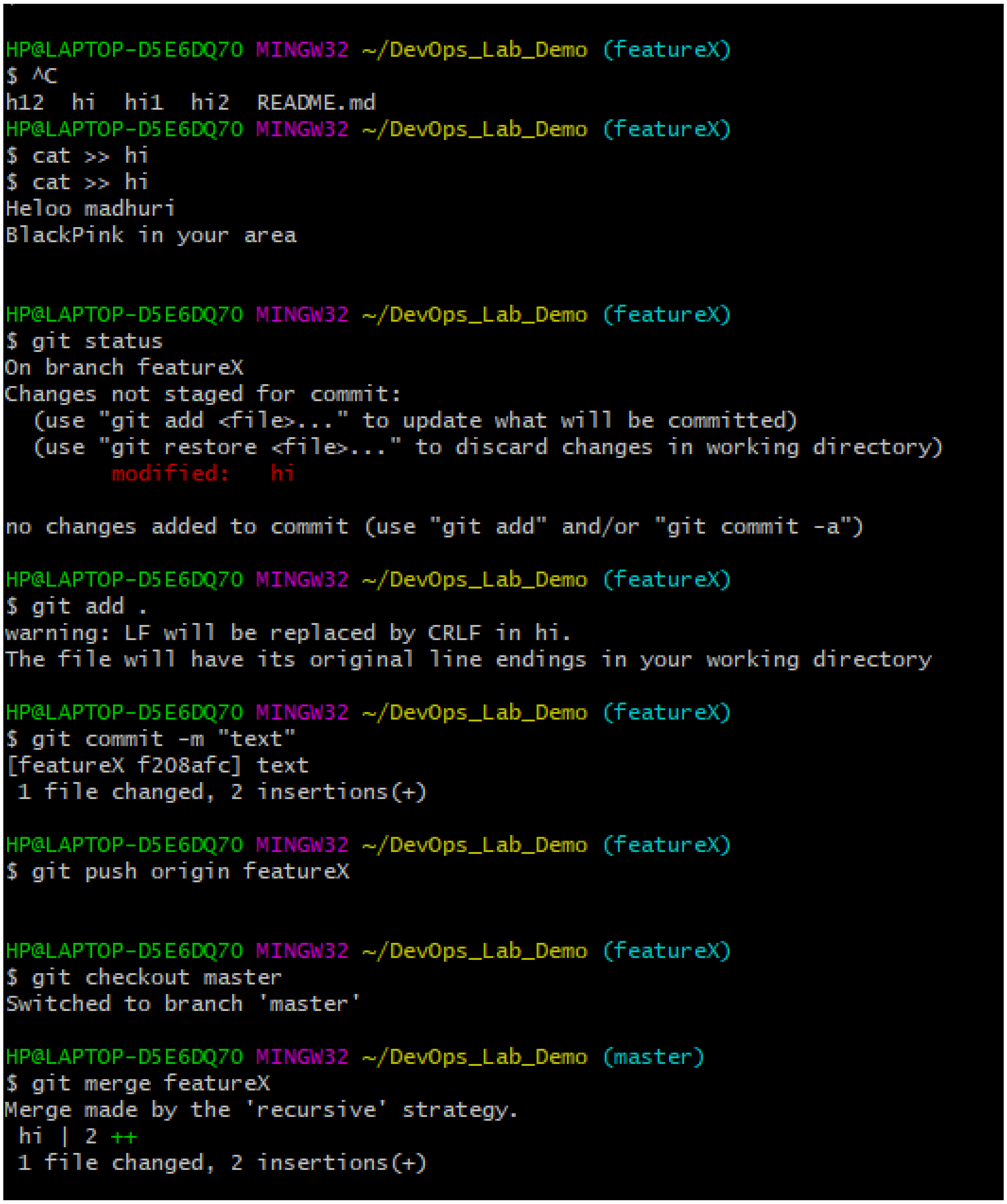
```
HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (main|MERGING)
$ git checkout main
Already on 'main'
M       README.md
A       file1
A       file2
A       file3
A       h12
A       hi2
Your branch is behind 'origin/main' by 2 commits, and can be fast-forwarded.
  (use "git pull" to update your local branch)

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (main)
$ git pull
Updating 9b66493..4bc12b0
Fast-forward
 README.md | 3 +--
 1 file changed, 1 insertion(+), 2 deletions(-)
```

Checking out master branch and switching to older version :



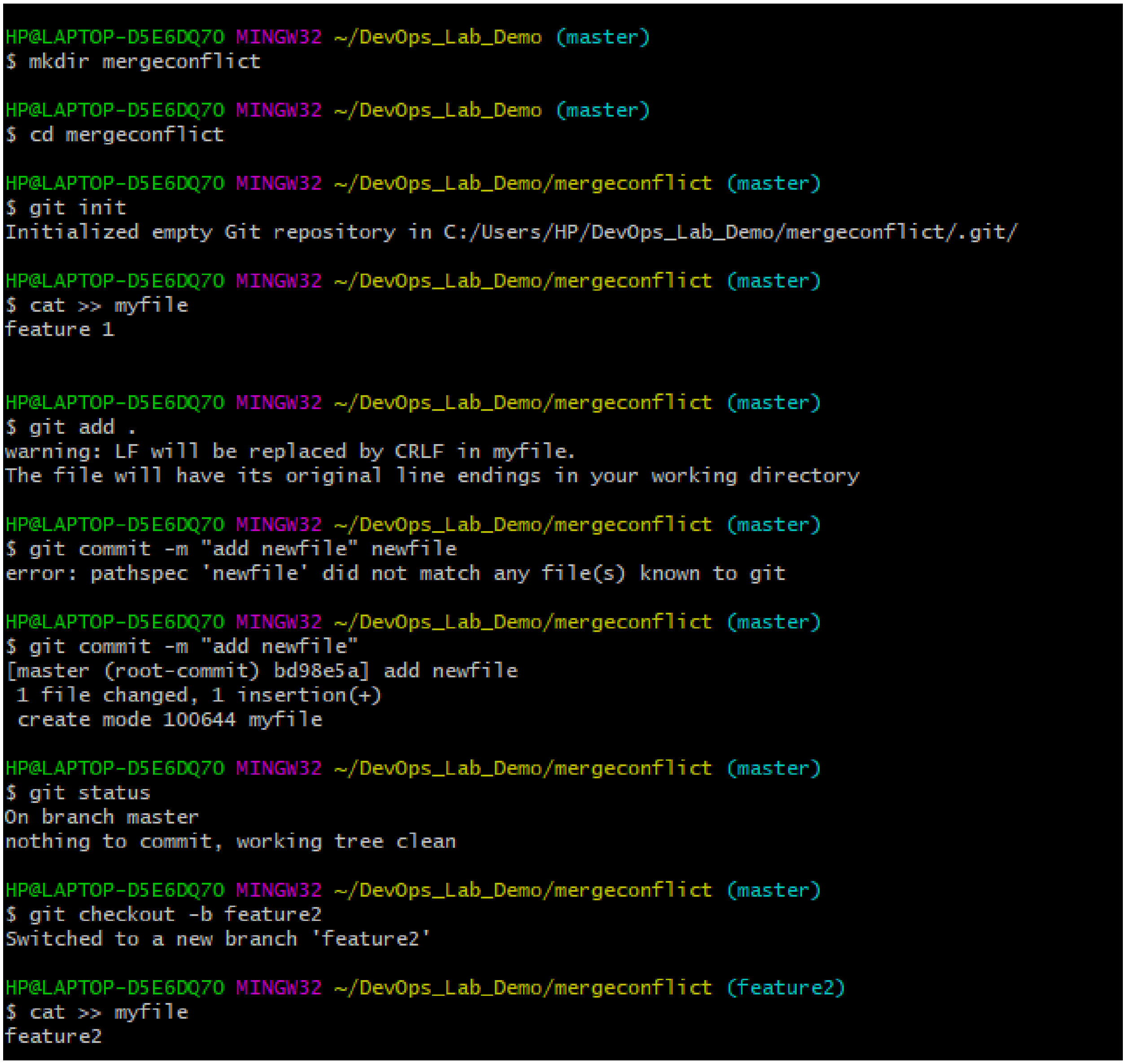
```
HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (main)
$ git checkout master
Switched to branch 'master'

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ git log --online --graph --all
fatal: unrecognized argument: --online

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ git log --oneline --graph --all
*   dc41a0f (HEAD -> master, hello) Merge https://github.com/madhhuurrii/DevOps_
Lab_Demo into master
|\
| * 4bc12b0 (origin/main, origin/HEAD, main) Update README.md
| * 024cdc8 Update README.md
* | 93613fb this is file1
* | 0c4a7ae (origin/featureX, featureX) h12
|/
* 9b66493 (checkout) h1
* d964b79 Update README.md
* 413300b Update README.md
* 3c65ca4 hi
* d7ab3f3 Initial commit

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ git merge main
Already up to date.

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ git merge checkout
Already up to date.

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ git merge featureX
Already up to date.
```

Merge with a merge commit :

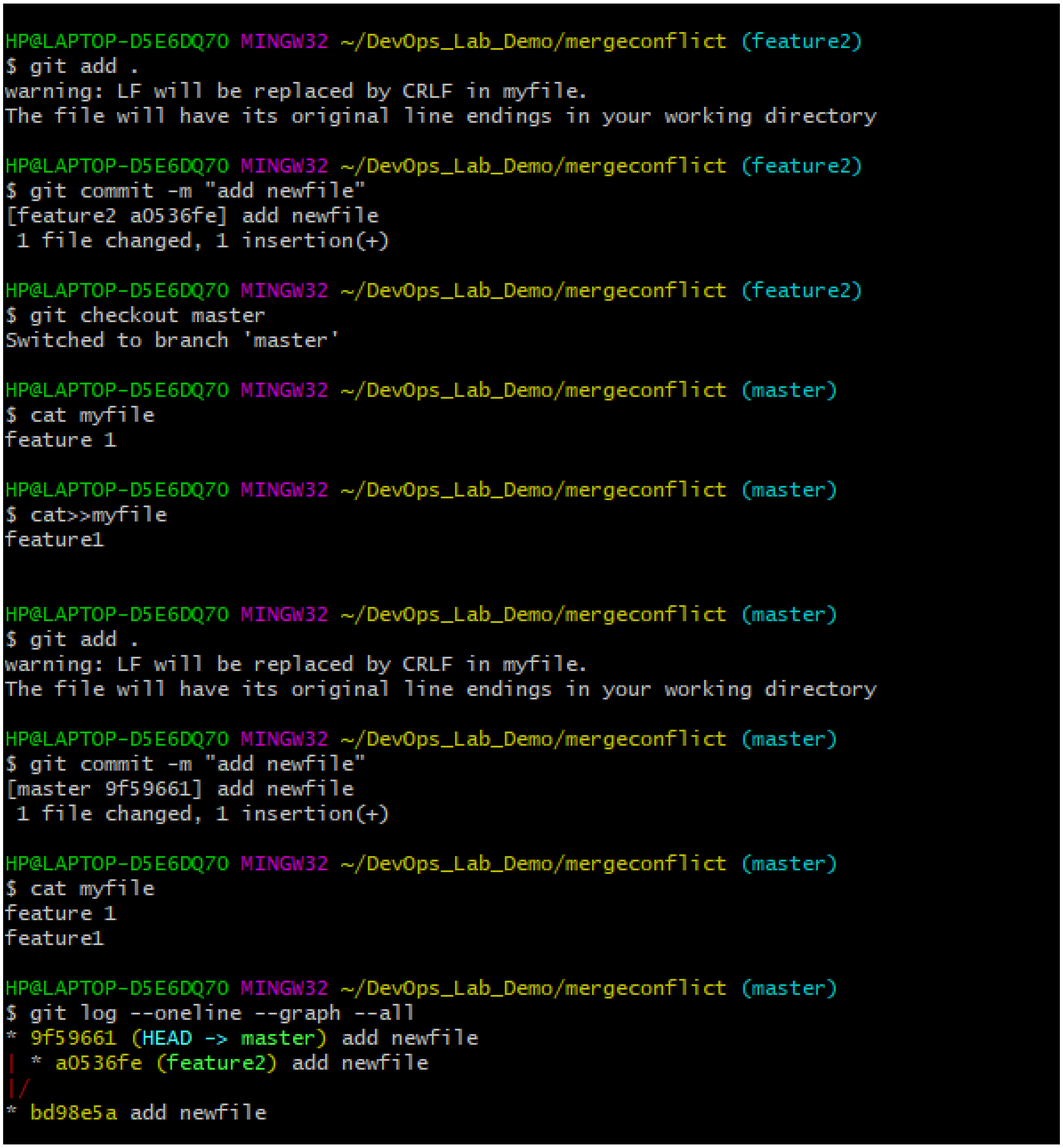


```
HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (featureX)
$ ^C
h12  hi  hi1  hi2  README.md
HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (featureX)
$ cat >> hi
$ cat >> hi
Heloo madhuri
BlackPink in your area


HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (featureX)
$ git status
On branch featureX
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:   hi

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

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (featureX)
$ git add .
warning: LF will be replaced by CRLF in hi.
The file will have its original line endings in your working directory

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (featureX)
$ git commit -m "text"
[featureX f208afc] text
 1 file changed, 2 insertions(+)

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (featureX)
$ git push origin featureX


HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (featureX)
$ git checkout master
Switched to branch 'master'

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ git merge featureX
Merge made by the 'recursive' strategy.
 hi | 2 ++
 1 file changed, 2 insertions(+)
```

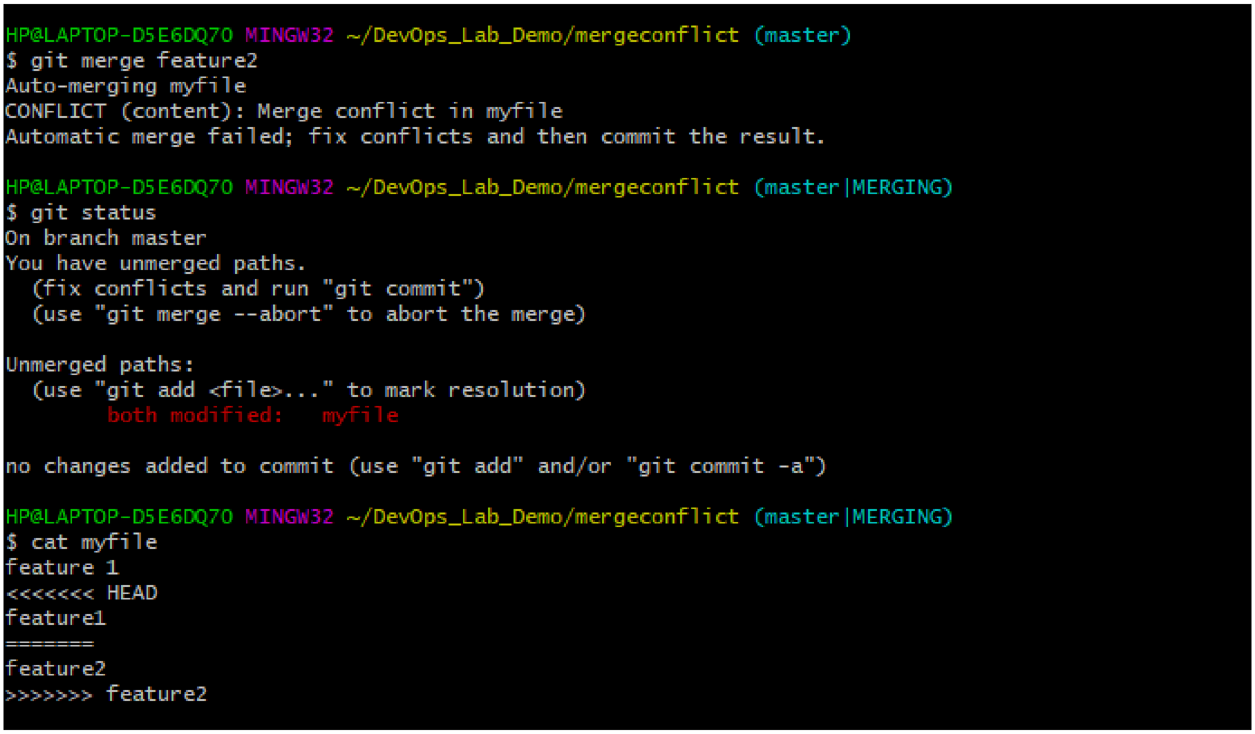
Merge with merge conflict :



```
HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ mkdir mergeconflict

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ cd mergeconflict

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (master)
$ git init
Initialized empty Git repository in C:/Users/HP/DevOps_Lab_Demo/mergeconflict/.git/

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (master)
$ cat >> myfile
feature 1


HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (master)
$ git add .
warning: LF will be replaced by CRLF in myfile.
The file will have its original line endings in your working directory

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (master)
$ git commit -m "add newfile" newfile
error: pathspec 'newfile' did not match any file(s) known to git

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (master)
$ git commit -m "add newfile"
[master (root-commit) bd98e5a] add newfile
 1 file changed, 1 insertion(+)
 create mode 100644 myfile

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (master)
$ git status
On branch master
nothing to commit, working tree clean

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (master)
$ git checkout -b feature2
Switched to a new branch 'feature2'

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (feature2)
$ cat >> myfile
feature2
```

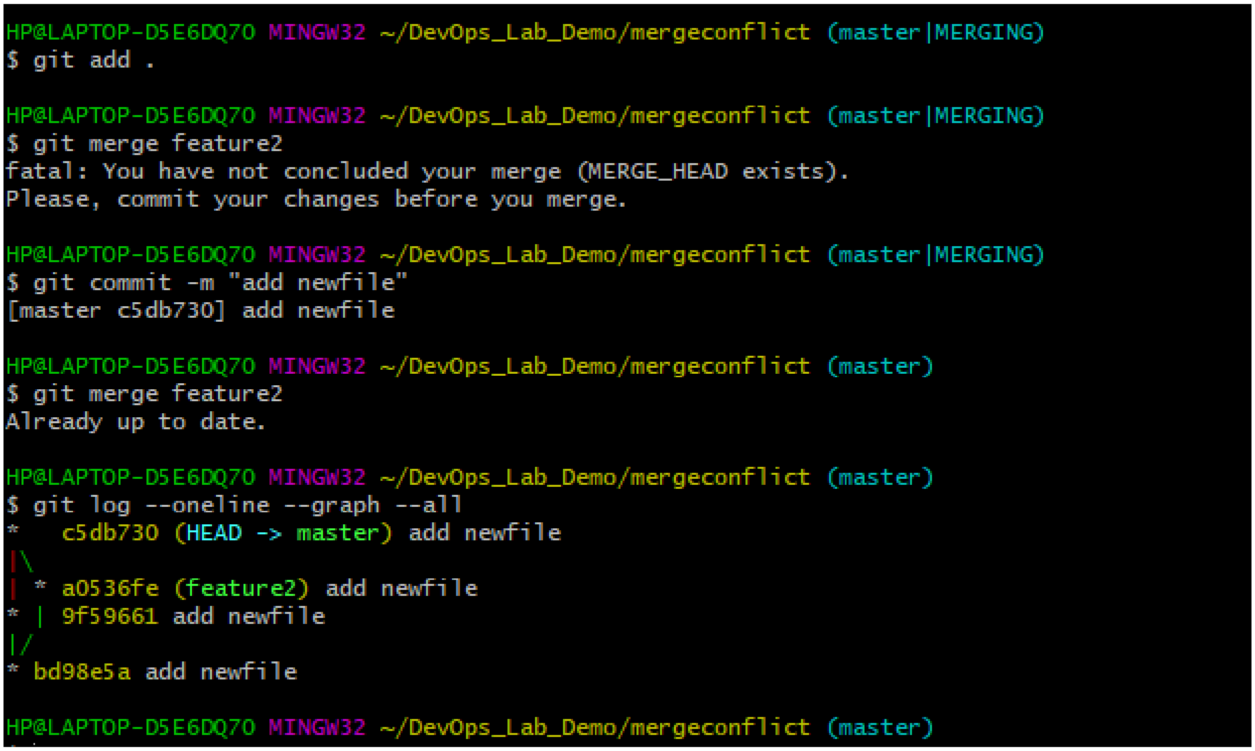


```
HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (feature2)
$ git add .
warning: LF will be replaced by CRLF in myfile.
The file will have its original line endings in your working directory

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (feature2)
$ git commit -m "add newfile"
[feature2 a0536fe] add newfile
 1 file changed, 1 insertion(+)

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (feature2)
$ git checkout master
Switched to branch 'master'

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (master)
$ cat myfile
feature 1

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (master)
$ cat>>myfile
feature1


HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (master)
$ git add .
warning: LF will be replaced by CRLF in myfile.
The file will have its original line endings in your working directory

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (master)
$ git commit -m "add newfile"
[master 9f59661] add newfile
 1 file changed, 1 insertion(+)

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (master)
$ cat myfile
feature 1
feature1

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (master)
$ git log --oneline --graph --all
* 9f59661 (HEAD -> master) add newfile
| * a0536fe (feature2) add newfile
|/
* bd98e5a add newfile
```

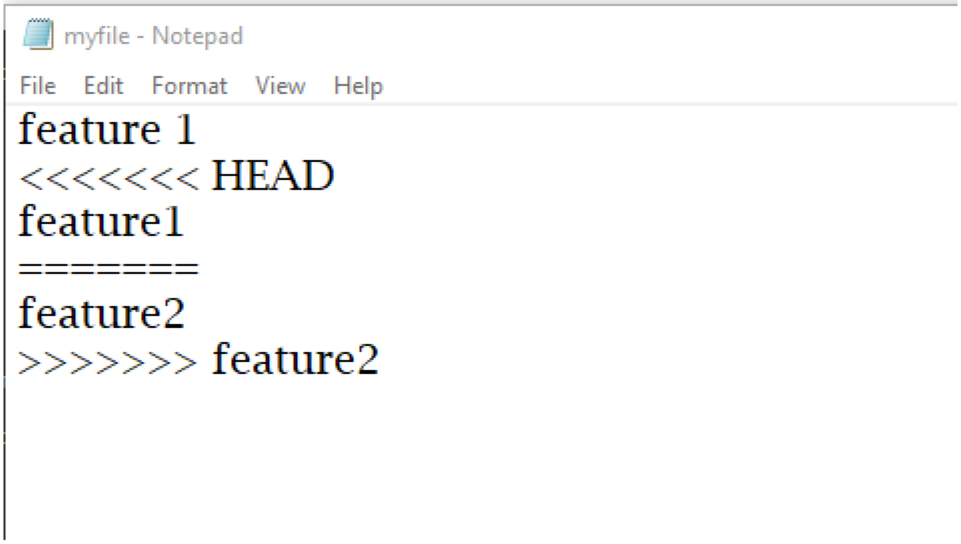


```
HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (master)
$ git merge feature2
Auto-merging myfile
CONFLICT (content): Merge conflict in myfile
Automatic merge failed; fix conflicts and then commit the result.

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (master|MERGING)
$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
        both modified:   myfile

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

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (master|MERGING)
$ cat myfile
feature 1
<<<<<<< HEAD
feature1
=======
feature2
>>>>>>> feature2
```

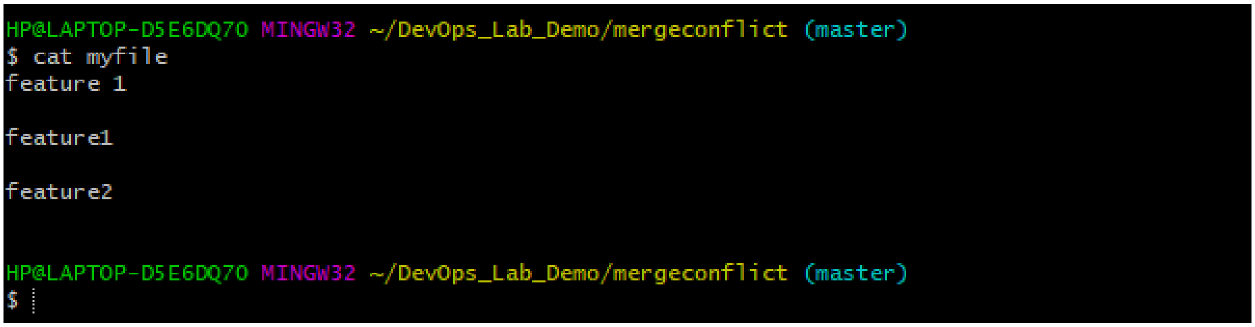
Merge the branches resolving the merged conflict :



```
HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (master|MERGING)
$ git add .

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (master|MERGING)
$ git merge feature2
fatal: You have not concluded your merge (MERGE_HEAD exists).
Please, commit your changes before you merge.

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (master|MERGING)
$ git commit -m "add newfile"
[master c5db730] add newfile

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (master)
$ git merge feature2
Already up to date.

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (master)
$ git log --oneline --graph --all
*   c5db730 (HEAD -> master) add newfile
|\
| * a0536fe (feature2) add newfile
* | 9f59661 add newfile
|/
* bd98e5a add newfile

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (master)
```



```
feature 1
<<<<<<< HEAD
feature1
=======
feature2
>>>>>>> feature2
```

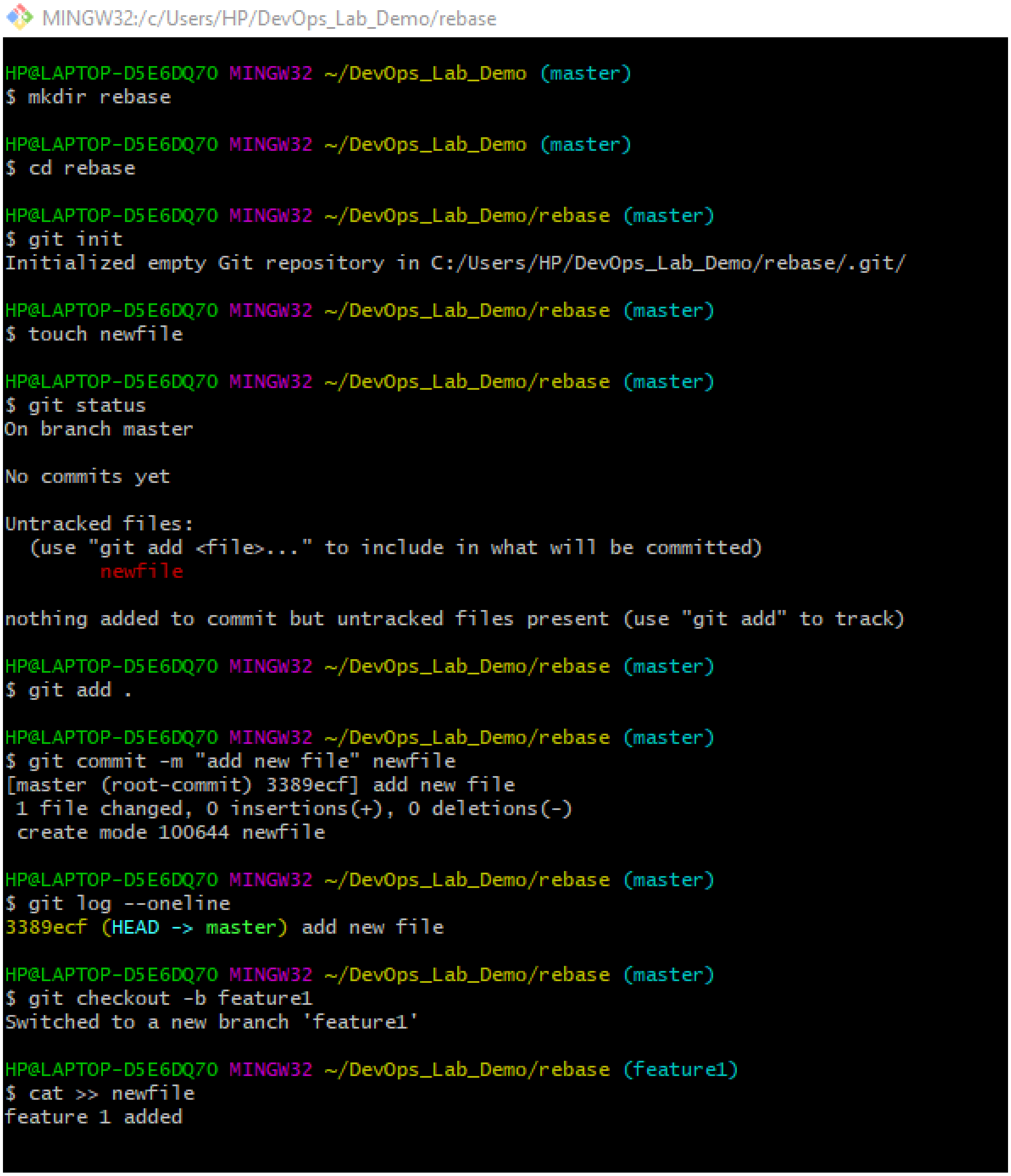


```
HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (master)
$ cat myfile
feature 1

feature1

feature2


HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/mergeconflict (master)
$
```

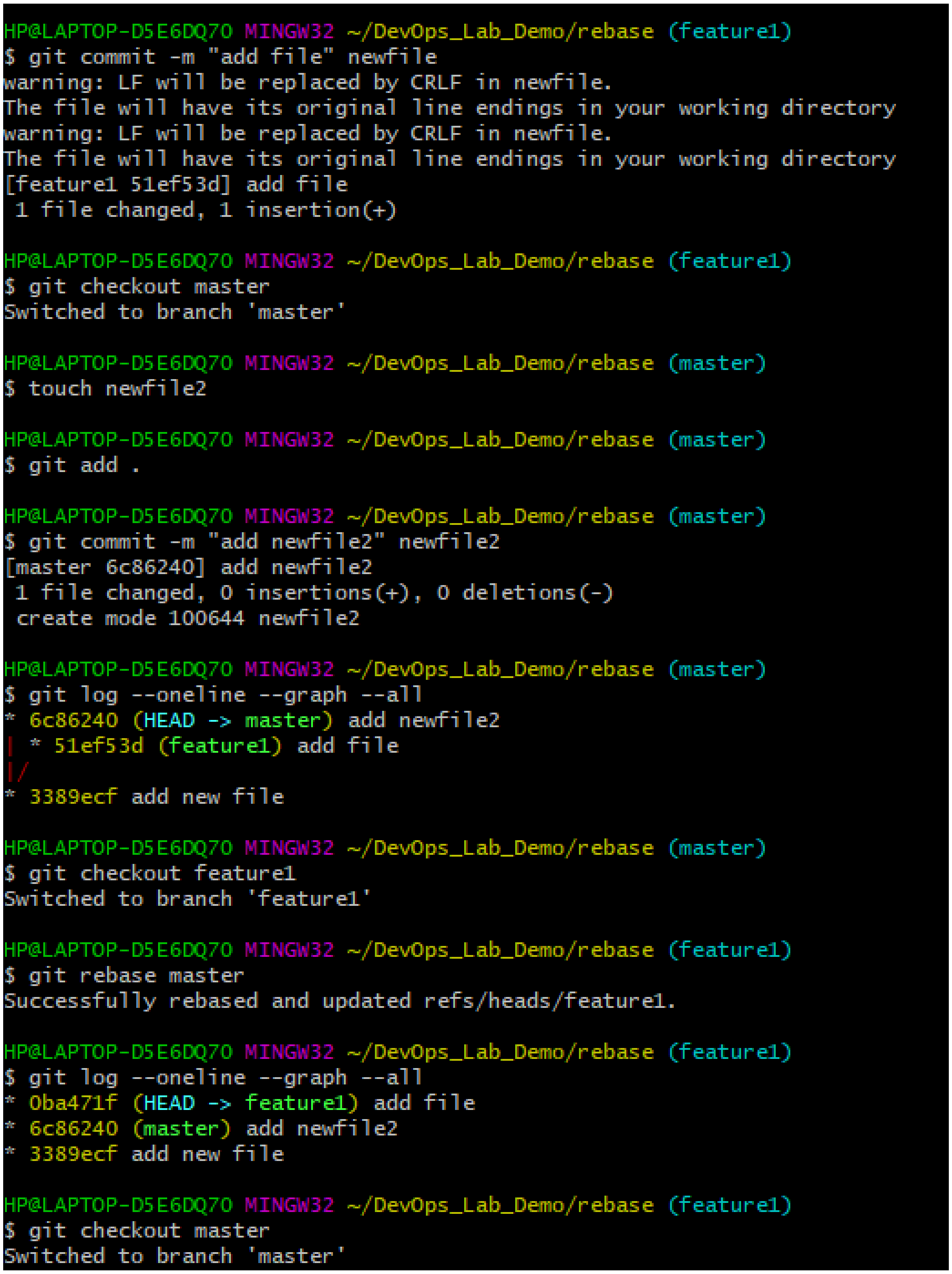
Conclusion: Hence, we have performed various GIT operations on local and remote repository using GIT Cheatsheet successfully.

POST LAB QUESTION

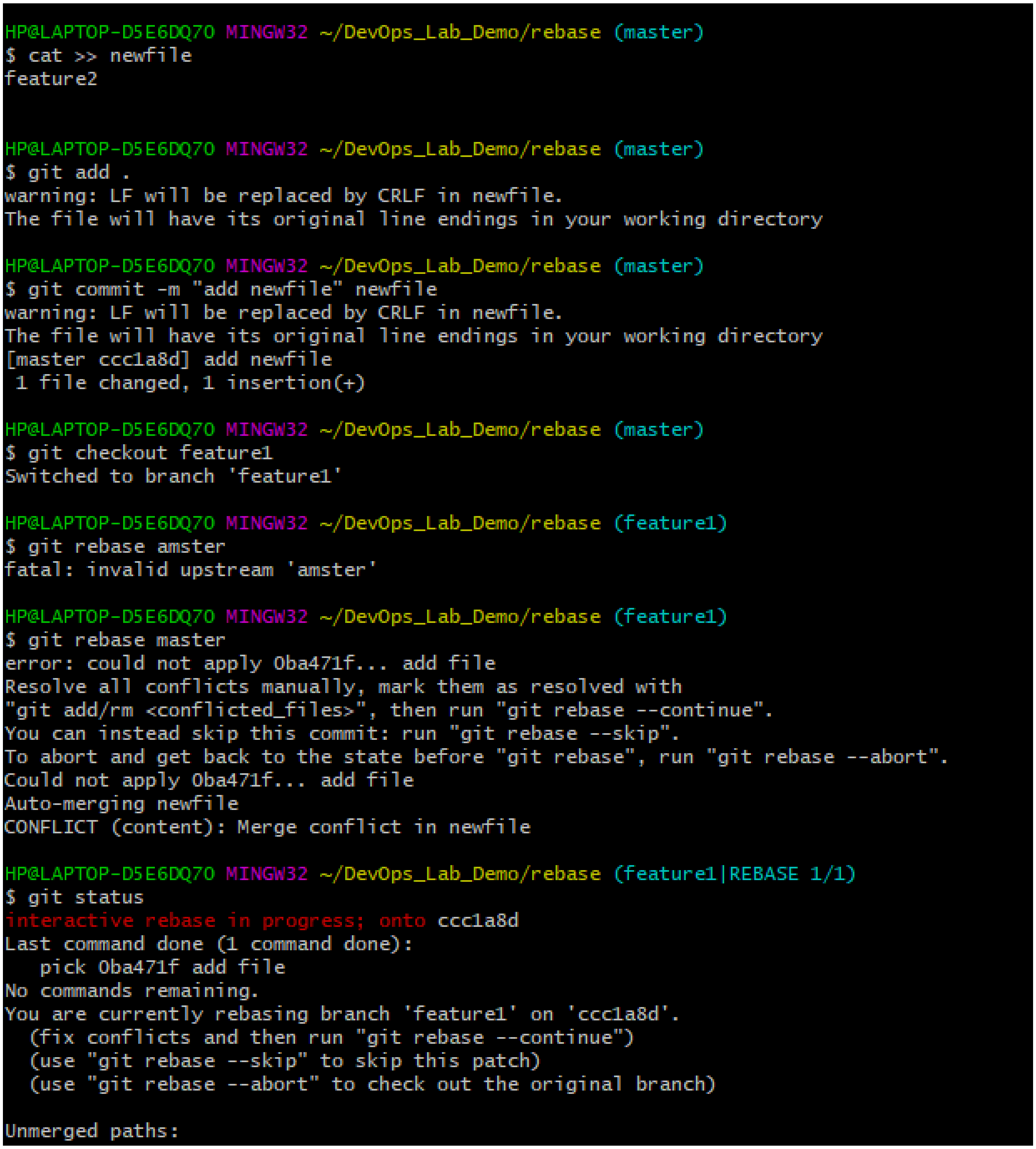
Rebasing:



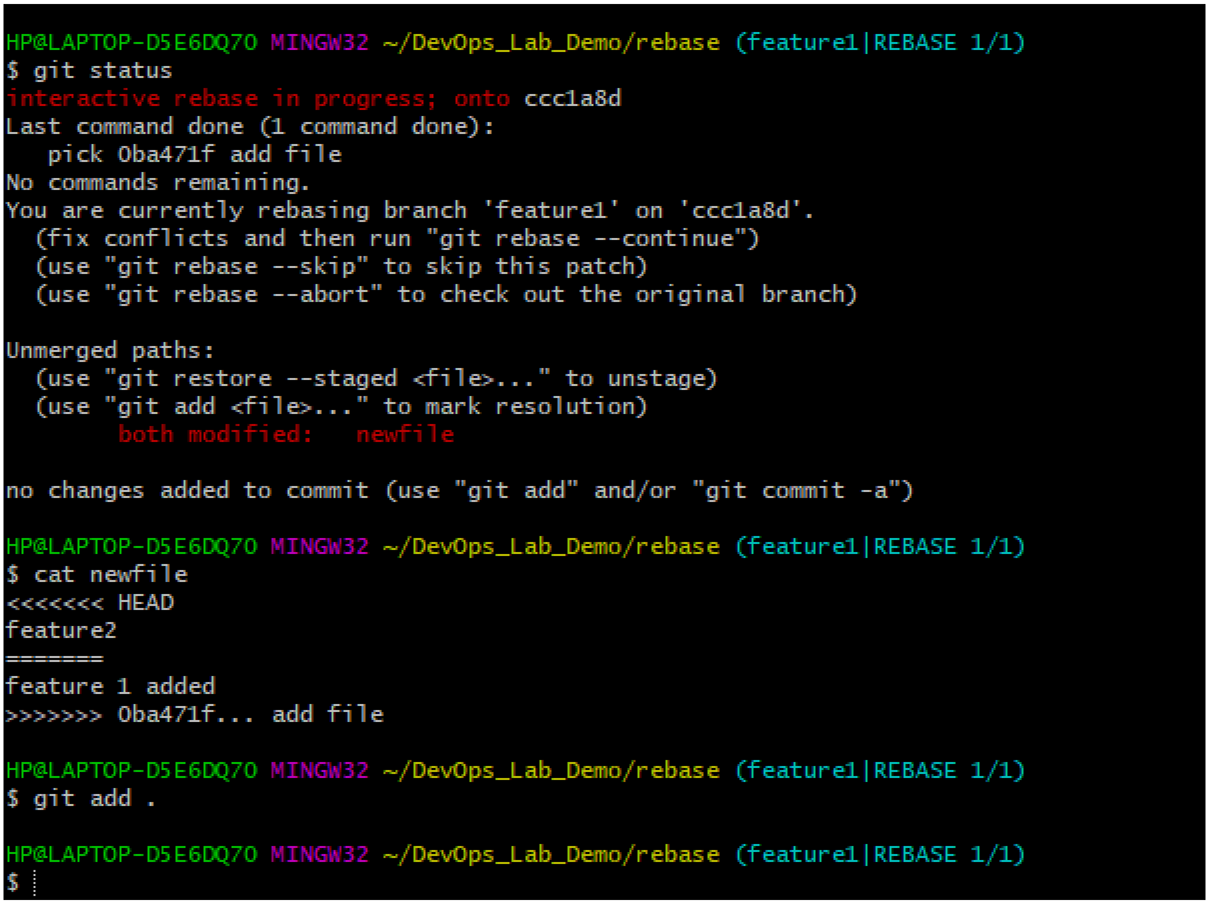
```
HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ mkdir rebase

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo (master)
$ cd rebase

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (master)
$ git init
Initialized empty Git repository in C:/Users/HP/DevOps_Lab_Demo/rebase/.git/

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (master)
$ touch newfile

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (master)
$ git status
On branch master

No commits yet

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

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

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (master)
$ git add .

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (master)
$ git commit -m "add new file" newfile
[master (root-commit) 3389ecf] add new file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 newfile

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (master)
$ git log --oneline
3389ecf (HEAD -> master) add new file

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (master)
$ git checkout -b feature1
Switched to a new branch 'feature1'

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (feature1)
$ cat >> newfile
feature 1 added
```



```
HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (feature1)
$ git commit -m "add file" newfile
warning: LF will be replaced by CRLF in newfile.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in newfile.
The file will have its original line endings in your working directory
[feature1 51ef53d] add file
 1 file changed, 1 insertion(+)

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (feature1)
$ git checkout master
Switched to branch 'master'

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (master)
$ touch newfile2

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (master)
$ git add .

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (master)
$ git commit -m "add newfile2" newfile2
[master 6c86240] add newfile2
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 newfile2

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (master)
$ git log --oneline --graph --all
* 6c86240 (HEAD -> master) add newfile2
| * 51ef53d (feature1) add file
|/
* 3389ecf add new file

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (master)
$ git checkout feature1
Switched to branch 'feature1'

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (feature1)
$ git rebase master
Successfully rebased and updated refs/heads/feature1.

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (feature1)
$ git log --oneline --graph --all
* 0ba471f (HEAD -> feature1) add file
* 6c86240 (master) add newfile2
* 3389ecf add new file

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (feature1)
$ git checkout master
Switched to branch 'master'
```

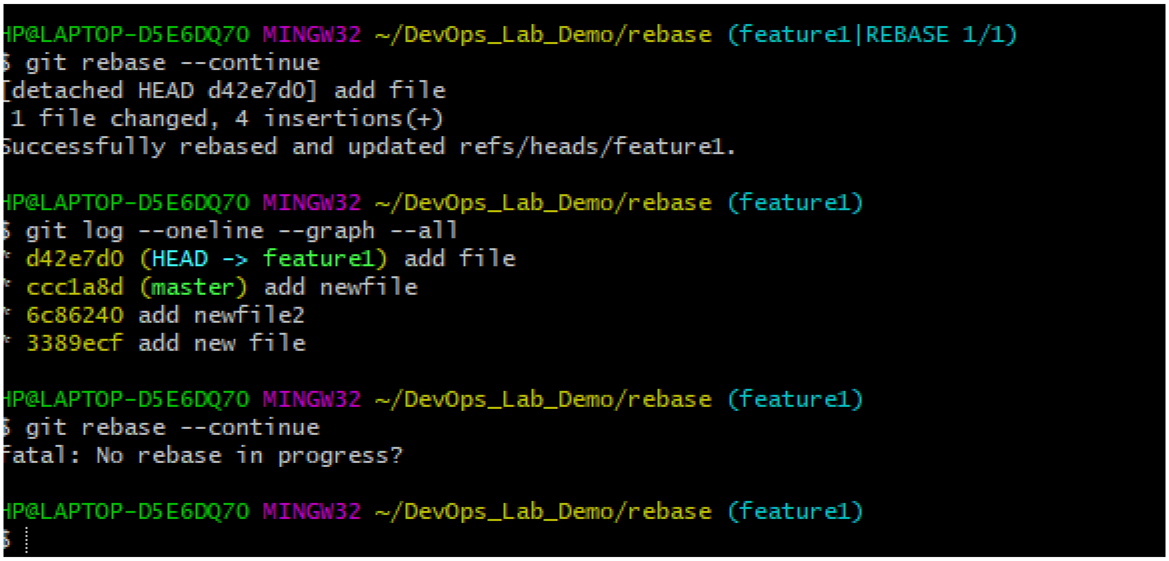


```
HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (master)
$ cat >> newfile
feature2


HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (master)
$ git add .
warning: LF will be replaced by CRLF in newfile.
The file will have its original line endings in your working directory

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (master)
$ git commit -m "add newfile" newfile
warning: LF will be replaced by CRLF in newfile.
The file will have its original line endings in your working directory
[master ccc1a8d] add newfile
 1 file changed, 1 insertion(+)

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (master)
$ git checkout feature1
Switched to branch 'feature1'

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (feature1)
$ git rebase amster
fatal: invalid upstream 'amster'

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (feature1)
$ git rebase master
error: could not apply 0ba471f... add file
Resolve all conflicts manually, mark them as resolved with
"git add/rm <conflicted_files>", then run "git rebase --continue".
You can instead skip this commit: run "git rebase --skip".
To abort and get back to the state before "git rebase", run "git rebase --abort".
Could not apply 0ba471f... add file
Auto-merging newfile
CONFLICT (content): Merge conflict in newfile

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (feature1|REBASE 1/1)
$ git status
interactive rebase in progress; onto ccc1a8d
Last command done (1 command done):
   pick 0ba471f add file
No commands remaining.
You are currently rebasing branch 'feature1' on 'ccc1a8d'.
  (fix conflicts and then run "git rebase --continue")
  (use "git rebase --skip" to skip this patch)
  (use "git rebase --abort" to check out the original branch)

Unmerged paths:
```



```
HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (feature1|REBASE 1/1)
$ git status
interactive rebase in progress; onto ccc1a8d
Last command done (1 command done):
   pick 0ba471f add file
No commands remaining.
You are currently rebasing branch 'feature1' on 'ccc1a8d'.
  (fix conflicts and then run "git rebase --continue")
  (use "git rebase --skip" to skip this patch)
  (use "git rebase --abort" to check out the original branch)

Unmerged paths:
  (use "git restore --staged <file>..." to unstage)
  (use "git add <file>..." to mark resolution)
        both modified:   newfile

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

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (feature1|REBASE 1/1)
$ cat newfile
<<<<<<< HEAD
feature2
=======
feature 1 added
>>>>>>> 0ba471f... add file

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (feature1|REBASE 1/1)
$ git add .

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (feature1|REBASE 1/1)
$
```

Rebasing and resolve a merged conflict :



```
HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (feature1|REBASE 1/1)
$ git rebase --continue
[detached HEAD d42e7d0] add file
 1 file changed, 4 insertions(+)
Successfully rebased and updated refs/heads/feature1.

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (feature1)
$ git log --oneline --graph --all
* d42e7d0 (HEAD -> feature1) add file
* ccc1a8d (master) add newfile
* 6c86240 add newfile2
* 3389ecf add new file

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (feature1)
$ git rebase --continue
fatal: No rebase in progress?

HP@LAPTOP-D5E6DQ70 MINGW32 ~/DevOps_Lab_Demo/rebase (feature1)
$
```