Problem Statement

Hooli Co, Has put Harry and Kane to implement a new set of features in their feature software. Both started working separately, making their own copies of the same source code. Now, it has become difficult for them to track the changes they've made in the original code,and they are finding it difficult to merge their code together.

Select the directory where I want to initialize Git

Initialize Git in the directory

- Syntax: git init

Create a src folder

Syntax: mkdir src

Create two files inside src folder

- Syntax: echo "app.py" >> app.py

```
# created app.py file
def addition(a,b):
    return a+b

addition(4,5)

def division(a,b):
    return a/b

division(4,5)
```

- Syntax: echo "main.py" >> main.py

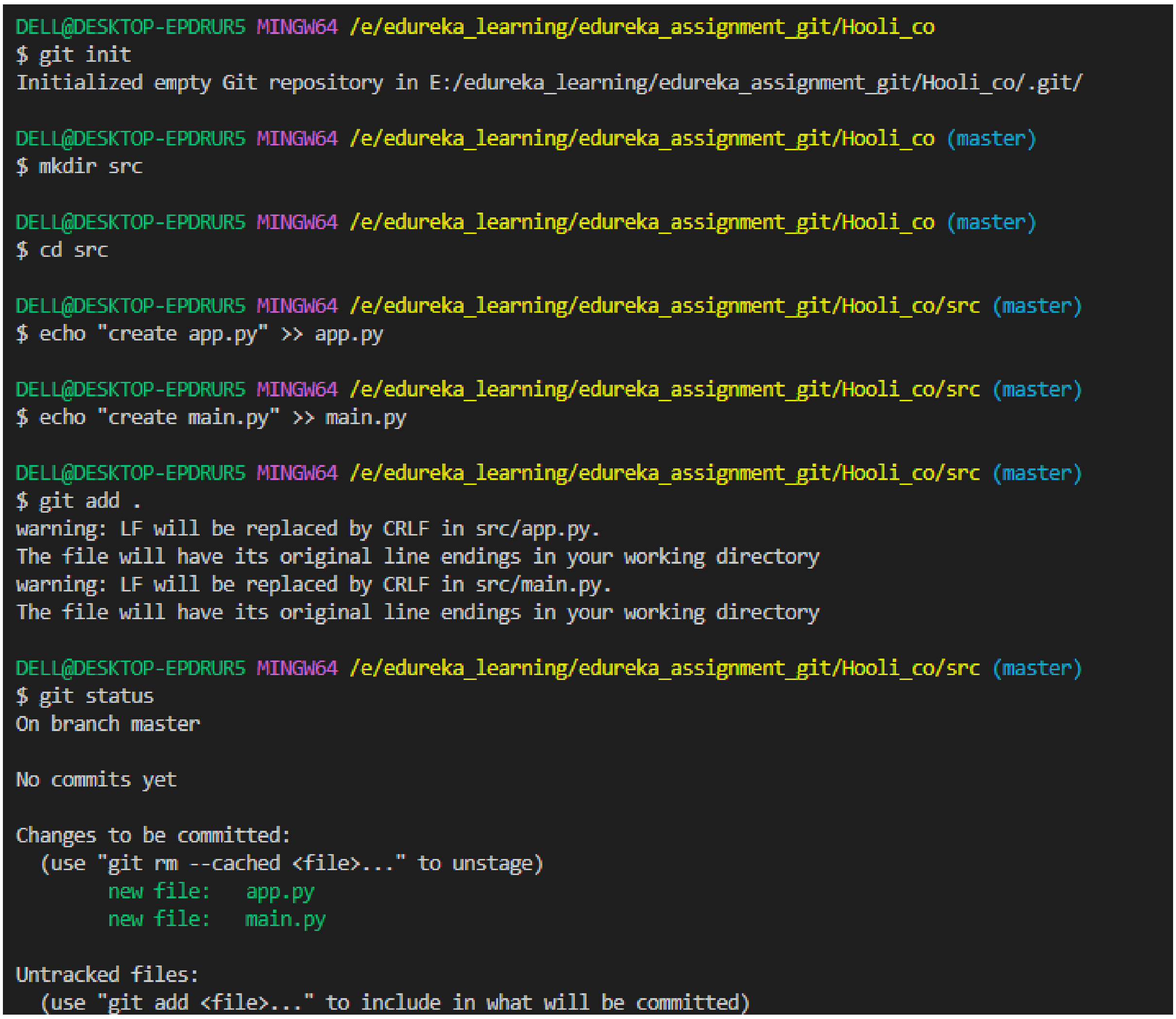
```
# created main.py

class Hooli:
    def __init__(self):
        print("welcome to the python")

python = Hooli()
```

This two files to add the staging area

- Syntax: git add .

To check the working tree status

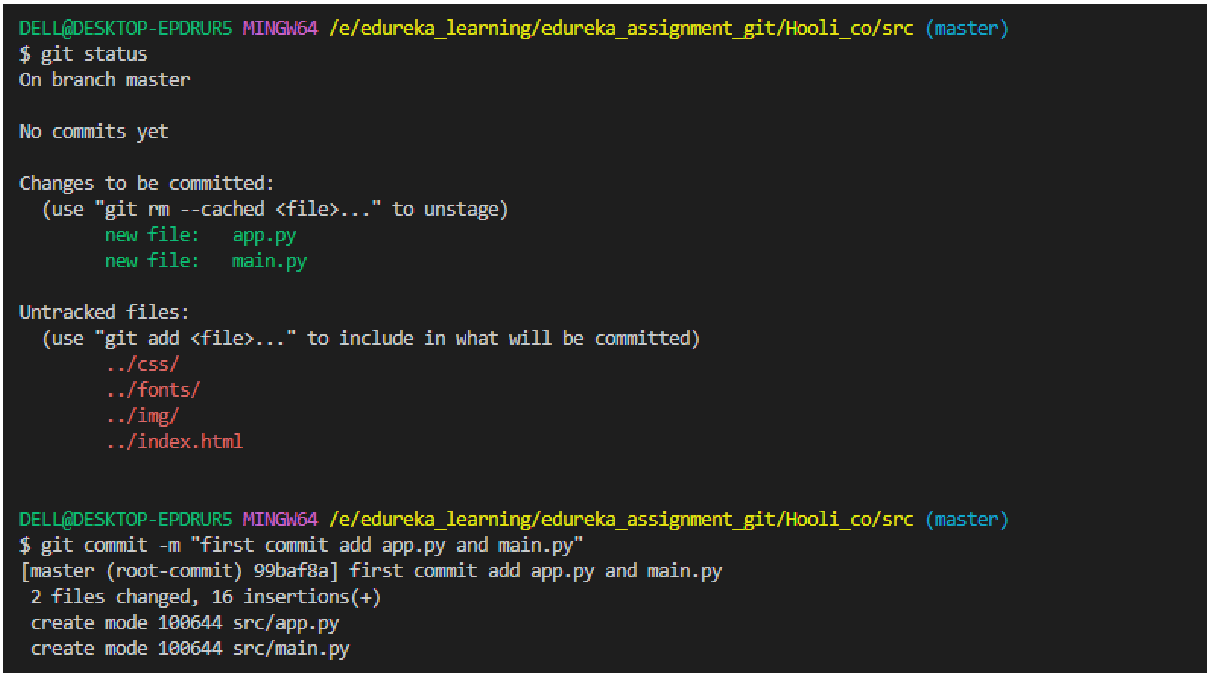
- Syntax: git status



```
DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co
$ git init
Initialized empty Git repository in E:/edureka_learning/edureka_assignment_git/Hooli_co/.git/

DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co (master)
$ mkdir src

DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co (master)
$ cd src

DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co/src (master)
$ echo "create app.py" >> app.py

DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co/src (master)
$ echo "create main.py" >> main.py

DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co/src (master)
$ git add .
warning: LF will be replaced by CRLF in src/app.py.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/main.py.
The file will have its original line endings in your working directory

DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co/src (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   app.py
        new file:   main.py

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

App.py and main.py To commit the staged files to our local repository:

- Syntax: git commit –m “first commit add app.py and main.py”



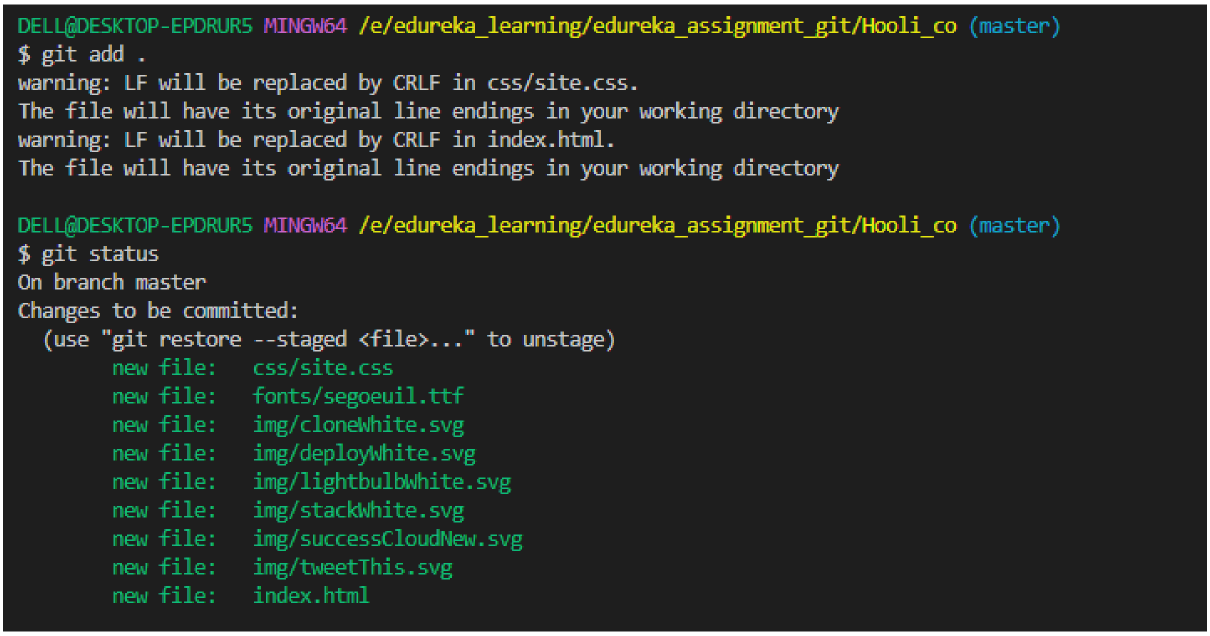
```
DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co/src (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   app.py
        new file:   main.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ../css/
        ../fonts/
        ../img/
        ../index.html


DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co/src (master)
$ git commit -m "first commit add app.py and main.py"
[master (root-commit) 99baf8a] first commit add app.py and main.py
 2 files changed, 16 insertions(+)
 create mode 100644 src/app.py
 create mode 100644 src/main.py
```

All structure files to add the staging area

- Syntax: git add .

To check the working tree status

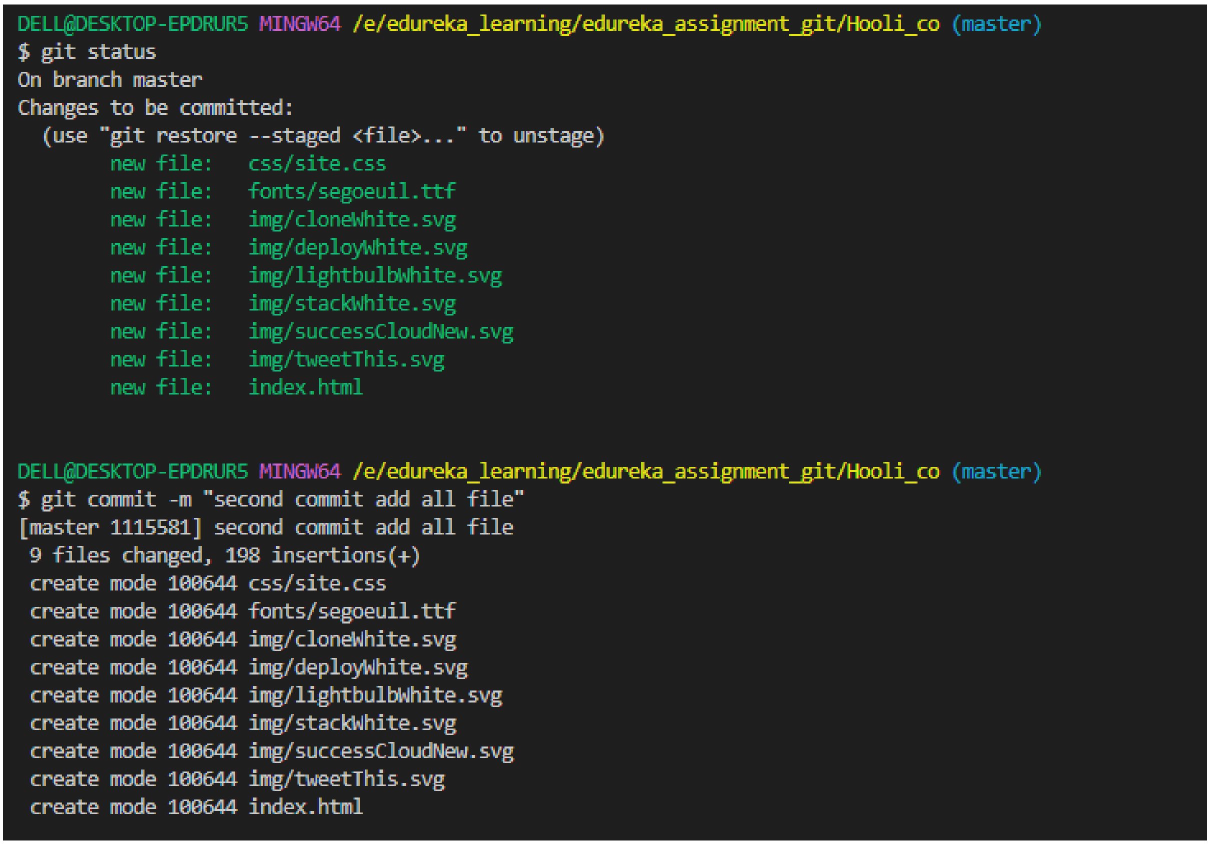
- Syntax: git status



```
DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co (master)
$ git add .
warning: LF will be replaced by CRLF in css/site.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working directory

DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   css/site.css
        new file:   fonts/segoeuil.ttf
        new file:   img/cloneWhite.svg
        new file:   img/deployWhite.svg
        new file:   img/lightbulbWhite.svg
        new file:   img/stackWhite.svg
        new file:   img/successCloudNew.svg
        new file:   img/tweetThis.svg
        new file:   index.html
```

To commit the staged files to our local repository:

- Syntax: git commit –m “first commit add app.py and main.py”



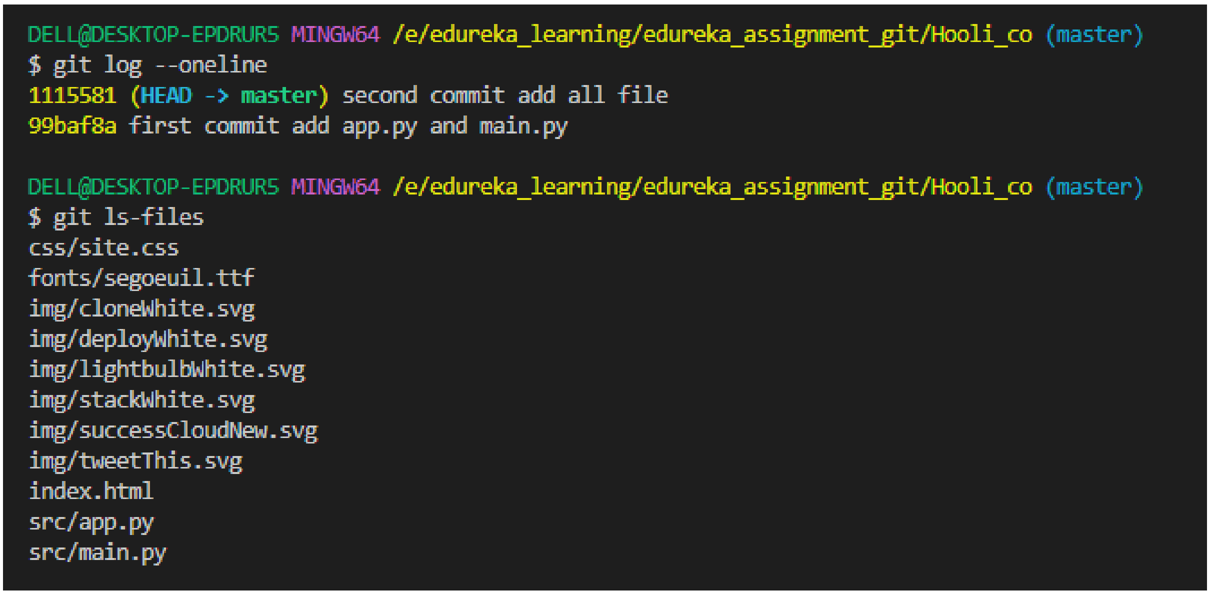
```
DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   css/site.css
        new file:   fonts/segoeuil.ttf
        new file:   img/cloneWhite.svg
        new file:   img/deployWhite.svg
        new file:   img/lightbulbWhite.svg
        new file:   img/stackWhite.svg
        new file:   img/successCloudNew.svg
        new file:   img/tweetThis.svg
        new file:   index.html


DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co (master)
$ git commit -m "second commit add all file"
[master 1115581] second commit add all file
 9 files changed, 198 insertions(+)
 create mode 100644 css/site.css
 create mode 100644 fonts/segoeuil.ttf
 create mode 100644 img/cloneWhite.svg
 create mode 100644 img/deployWhite.svg
 create mode 100644 img/lightbulbWhite.svg
 create mode 100644 img/stackWhite.svg
 create mode 100644 img/successCloudNew.svg
 create mode 100644 img/tweetThis.svg
 create mode 100644 index.html
```

Check log

Syntax:git log –online

Check all files in local repository

Syntax: git ls-files

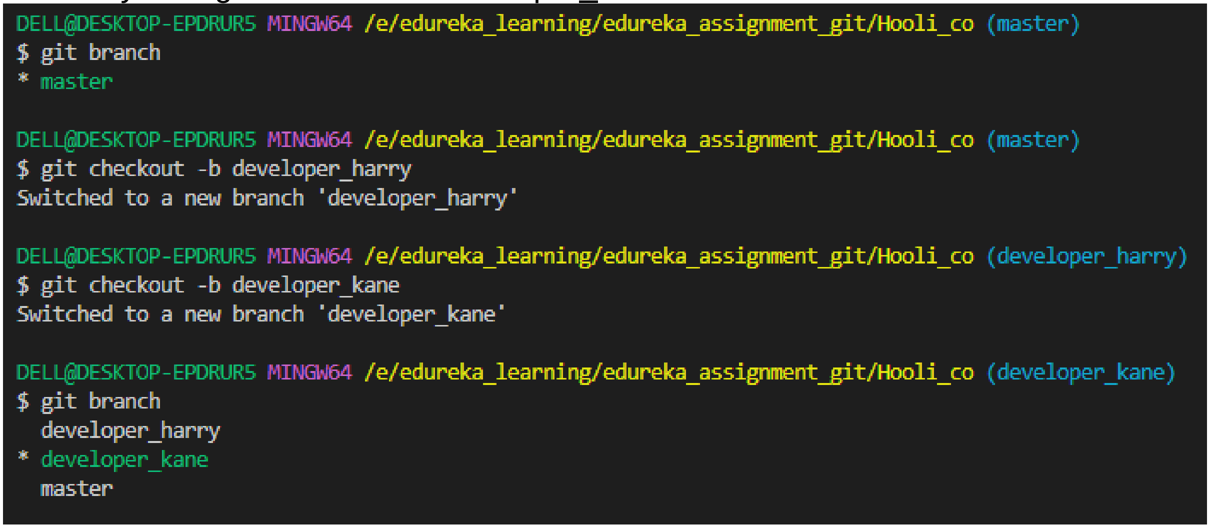


```
DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co (master)
$ git log --oneline
1115581 (HEAD -> master) second commit add all file
99baf8a first commit add app.py and main.py

DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co (master)
$ git ls-files
css/site.css
fonts/segoeuil.ttf
img/cloneWhite.svg
img/deployWhite.svg
img/lightbulbWhite.svg
img/stackWhite.svg
img/successCloudNew.svg
img/tweetThis.svg
index.html
src/app.py
src/main.py
```

1.) Create two separate branches from master

Syntax: git checkout –b developer_harry

Syntax: git checkout –b developer_kane



```
DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co (master)
$ git branch
* master

DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co (master)
$ git checkout -b developer_harry
Switched to a new branch 'developer_harry'

DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co (developer_harry)
$ git checkout -b developer_kane
Switched to a new branch 'developer_kane'

DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co (developer_kane)
$ git branch
  developer_harry
* developer_kane
  master
```

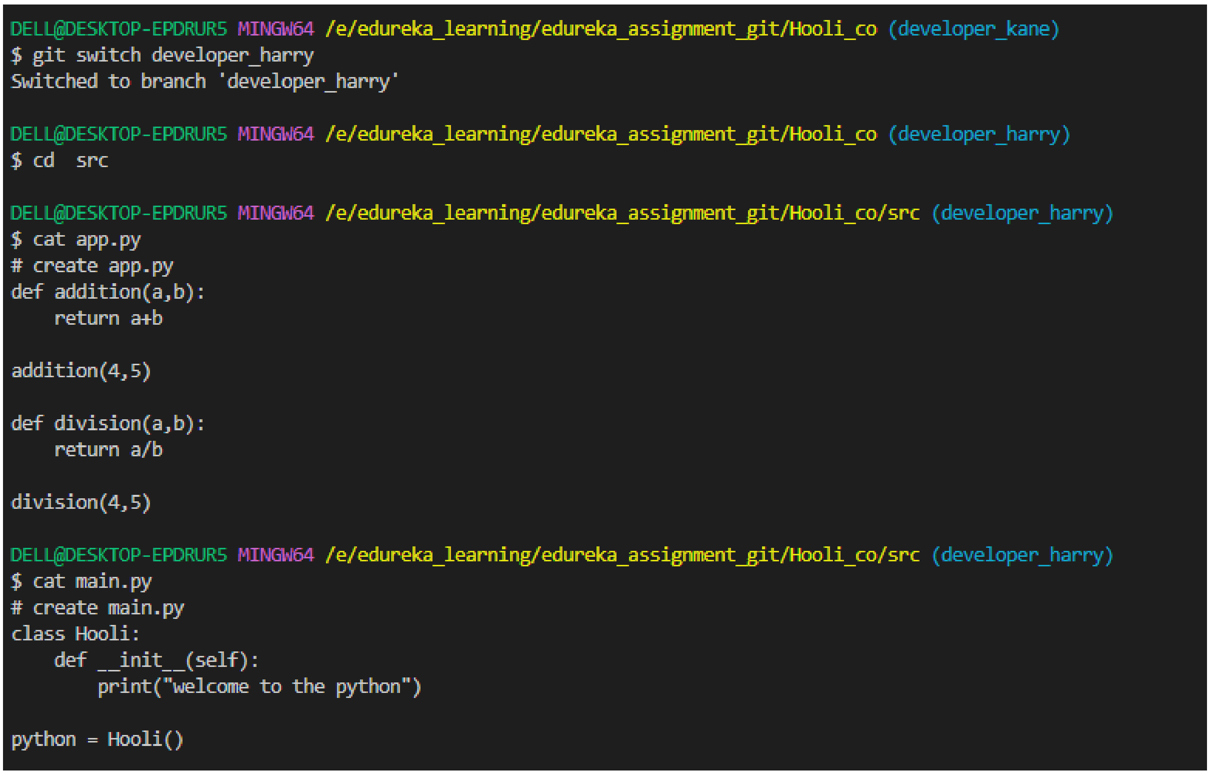
Switch to developer_harry branch

- Syntax: git switch developer_harry

Check app.py and main.py code in developer_harry branch

- Syntax: cat app.py

- Syntax: cat main.py



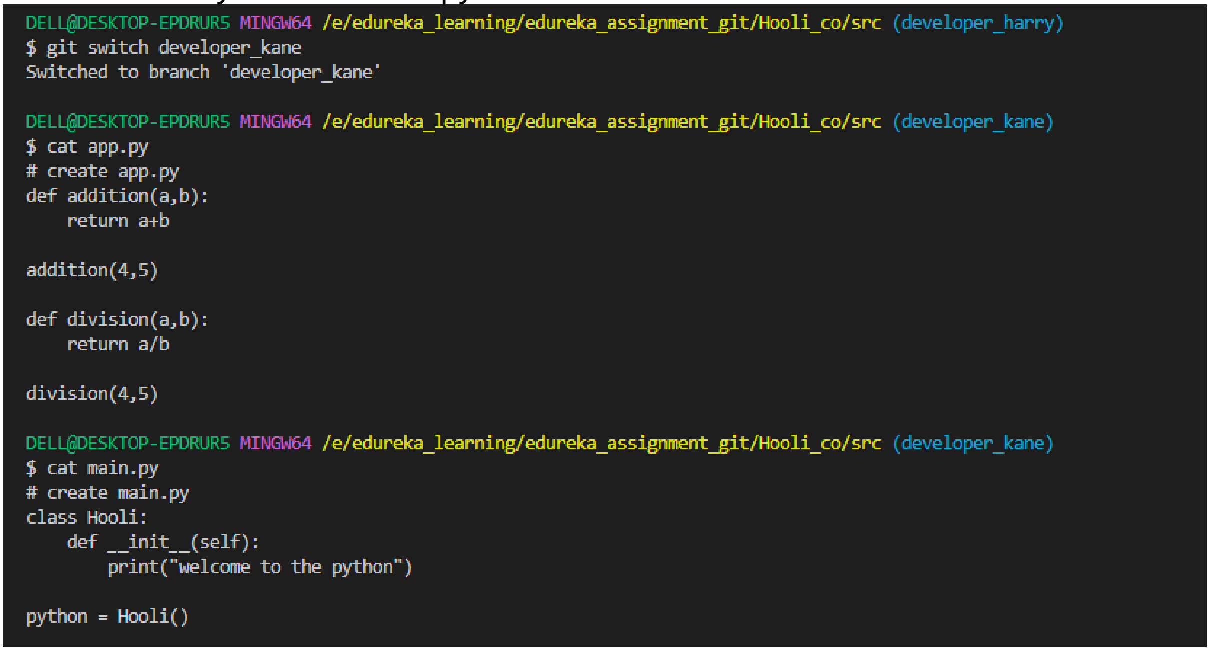
```
DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co (developer_kane)
$ git switch developer_harry
Switched to branch 'developer_harry'

DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co (developer_harry)
$ cd  src

DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co/src (developer_harry)
$ cat app.py
# create app.py
def addition(a,b):
    return a+b

addition(4,5)

def division(a,b):
    return a/b

division(4,5)

DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co/src (developer_harry)
$ cat main.py
# create main.py
class Hooli:
    def __init__(self):
        print("welcome to the python")

python = Hooli()
```

Switch to developer_kane branch

- Syntax: git switch developer_kane

Check app.py and main.py code in developer_kane branch

- Syntax: cat app.py
- Syntax: cat main.py



```
DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co/src (developer_harry)
$ git switch developer_kane
Switched to branch 'developer_kane'

DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co/src (developer_kane)
$ cat app.py
# create app.py
def addition(a,b):
    return a+b

addition(4,5)

def division(a,b):
    return a/b

division(4,5)

DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co/src (developer_kane)
$ cat main.py
# create main.py
class Hooli:
    def __init__(self):
        print("welcome to the python")

python = Hooli()
```

2.) Make changes in the same function of the source code in both the branches

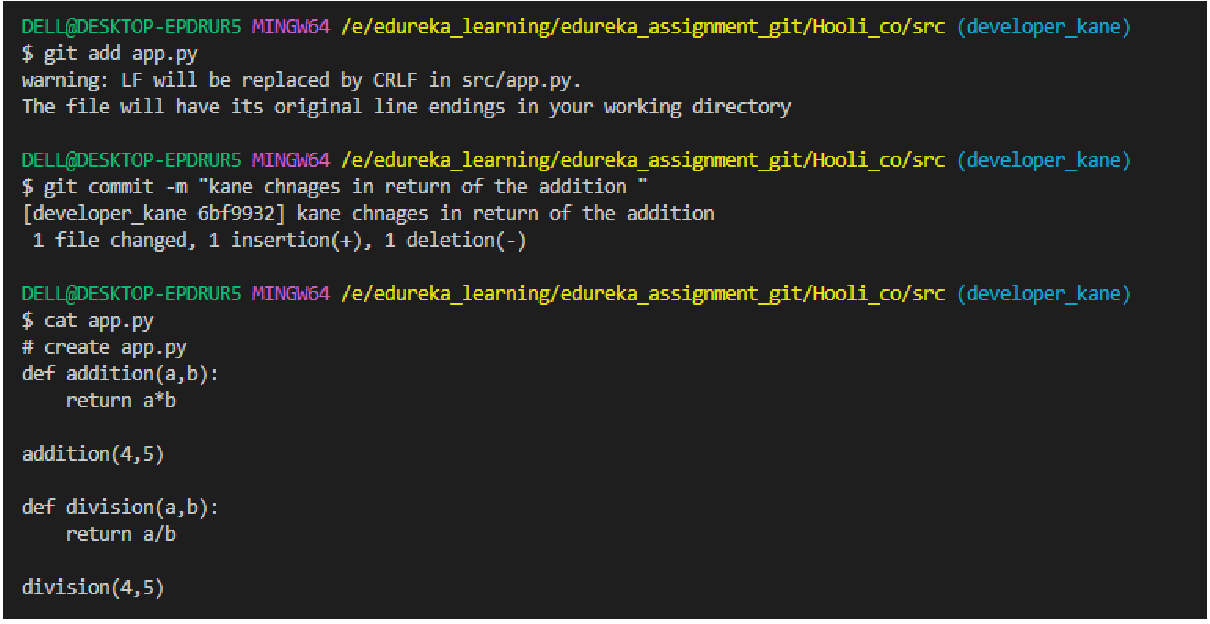
In developer_kane branch:

Developer kane want to change in app.py file

Kane make changes in only “ return” of the addition() function “return a+b ” to “return a*b

```
# create app.py
def addition(a,b):
    return a*b

addition(4,5)

def division(a,b):
     return a/b

 division(4,5)
```

- Syntax: git add .
- Syntax: git commit –m “make changes in return of the addition function”
- Syntax: cat app.py



```
DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co/src (developer_kane)
$ git add app.py
warning: LF will be replaced by CRLF in src/app.py.
The file will have its original line endings in your working directory

DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co/src (developer_kane)
$ git commit -m "kane chnages in return of the addition "
[developer_kane 6bf9932] kane chnages in return of the addition
 1 file changed, 1 insertion(+), 1 deletion(-)

DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co/src (developer_kane)
$ cat app.py
# create app.py
def addition(a,b):
    return a*b

addition(4,5)

def division(a,b):
    return a/b

division(4,5)
```

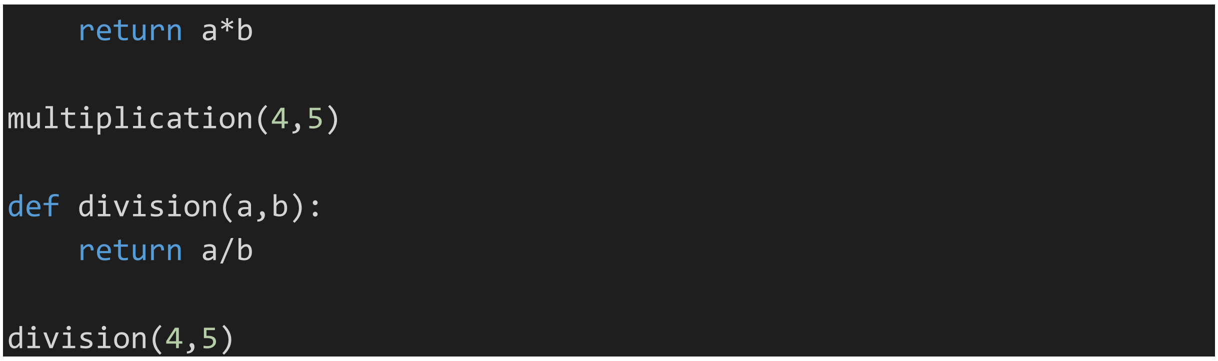
Changes in Branch developer_harry:

Developer harry also want to Change in app.py file

Harry change name of the addition() function to multiplication() and also make changes in return

“return a+b ” to “return a*b

```
# create app.py
def multiplication(a,b):
```

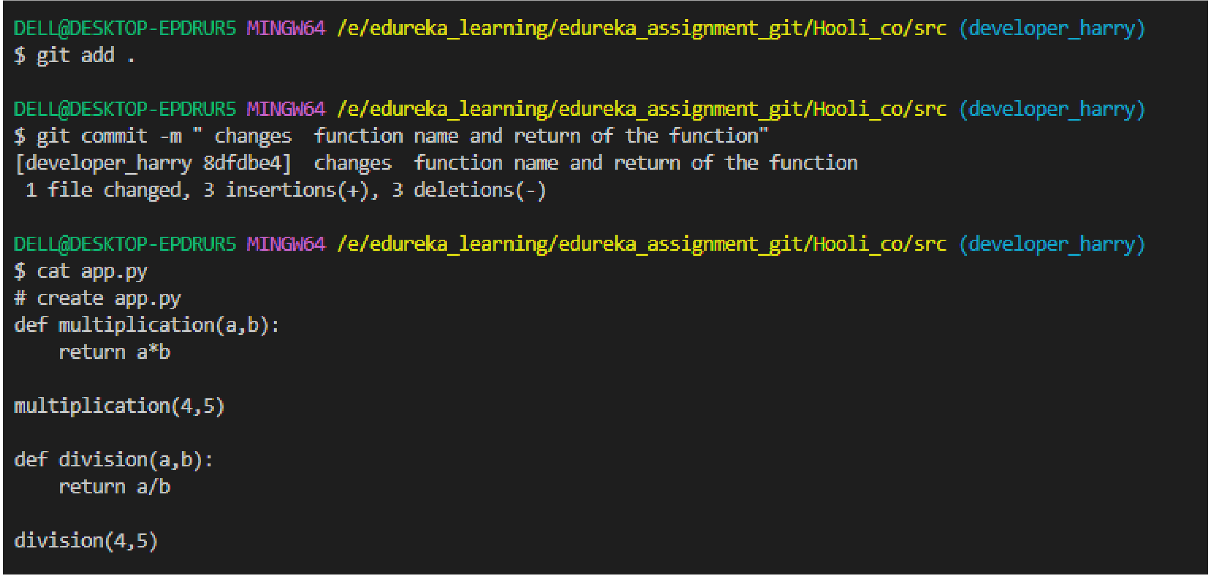


```
    return a*b

multiplication(4,5)

def division(a,b):
    return a/b

division(4,5)
```

- Syntax: git add .

- Syntax: git commit –m “ changes function name and return of the function”

- Syntax: cat app.py

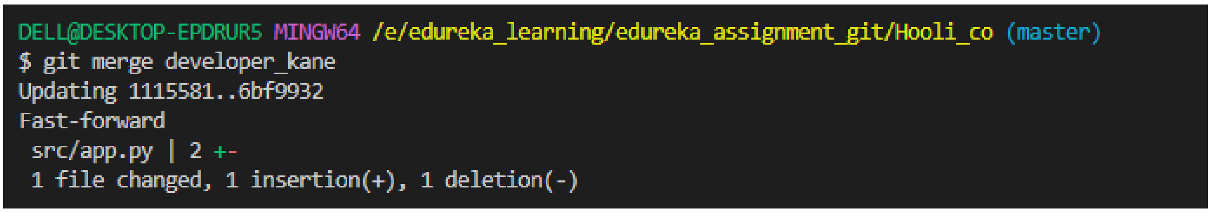


```
DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co/src (developer_harry)
$ git add .

DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co/src (developer_harry)
$ git commit -m " changes  function name and return of the function"
[developer_harry 8dfdbe4]  changes  function name and return of the function
 1 file changed, 3 insertions(+), 3 deletions(-)

DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co/src (developer_harry)
$ cat app.py
# create app.py
def multiplication(a,b):
    return a*b

multiplication(4,5)

def division(a,b):
    return a/b

division(4,5)
```

3.) Merge branch developer_kane into the master

- Syntax: git switch master

- Syntax: git merge developer_kane



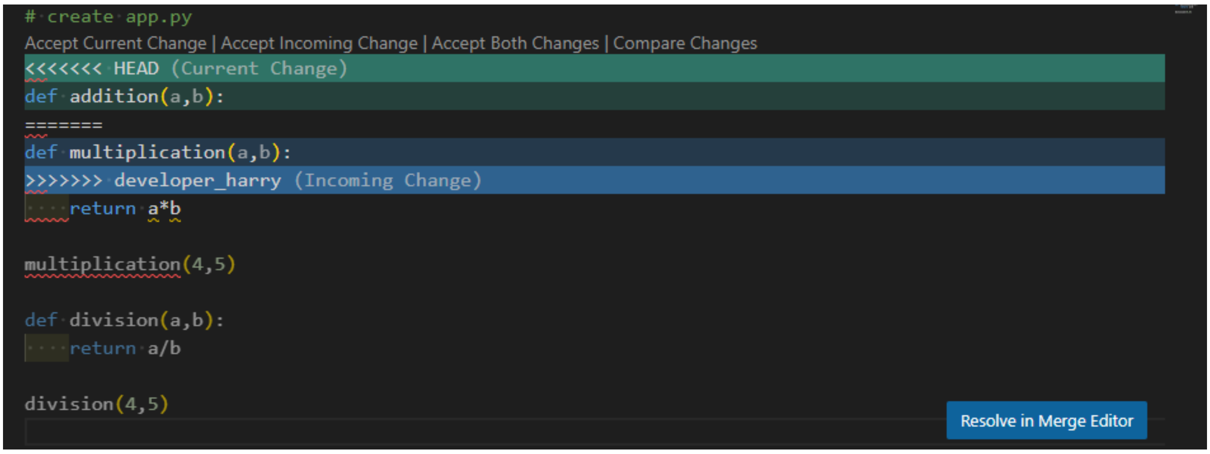
```
DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co (master)
$ git merge developer_kane
Updating 1115581..6bf9932
Fast-forward
 src/app.py | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Check app.py in master branch

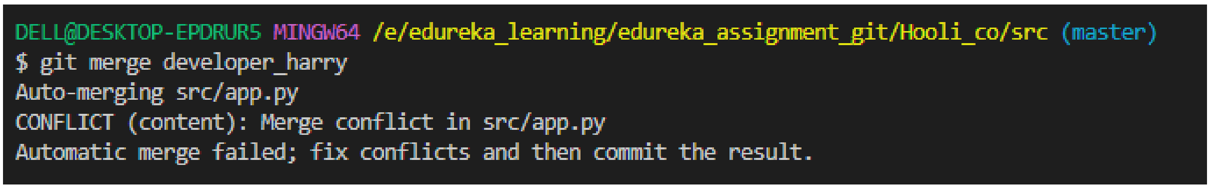
- Syntax: cat app.py

4.) Try and merge branch2 into the master(merge conflict should arise)

- Syntax: git merge developer_harry


Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
Resolve in Merge Editor

```
# create app.py
<<<<<<< HEAD (Current Change)
def addition(a,b):
=======
def multiplication(a,b):
>>>>>>> developer_harry (Incoming Change)
    return a*b

multiplication(4,5)

def division(a,b):
    return a/b

division(4,5)
```



```
DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co/src (master)
$ git merge developer_harry
Auto-merging src/app.py
CONFLICT (content): Merge conflict in src/app.py
Automatic merge failed; fix conflicts and then commit the result.
```

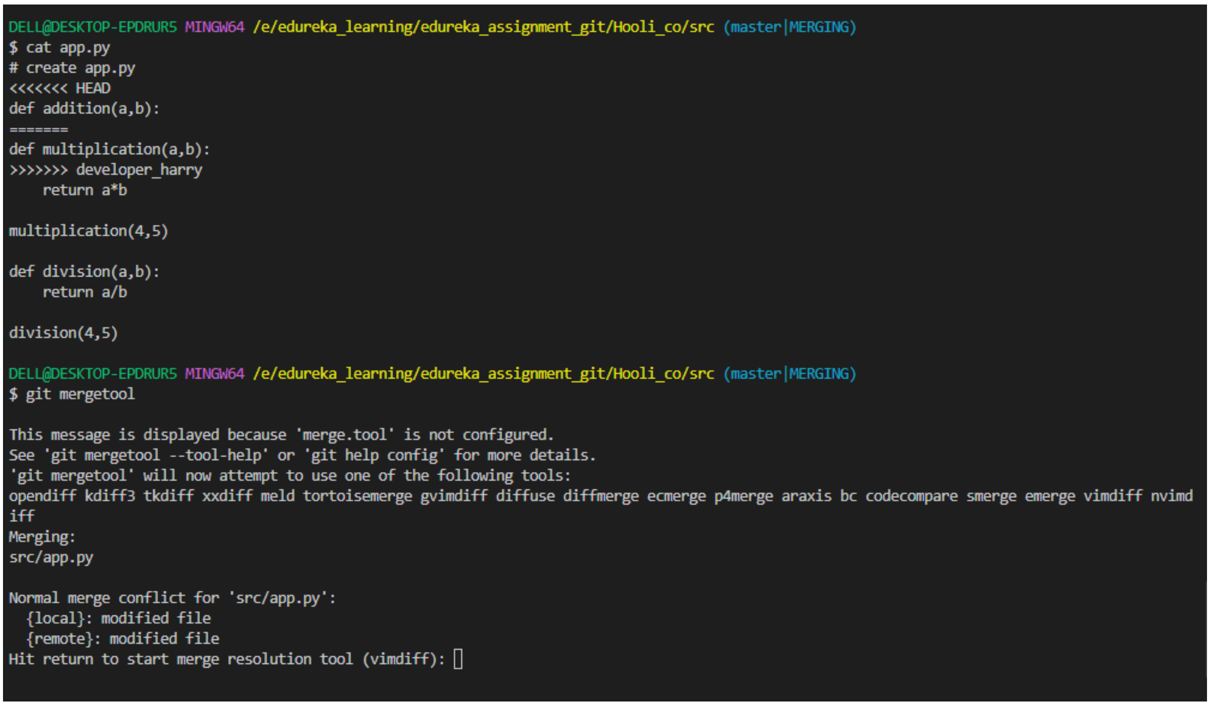
Check Conflict

- Syntax : cat app.py

5.) Install a merge tool of your choice and resolve the merge conflict using git mergetool command

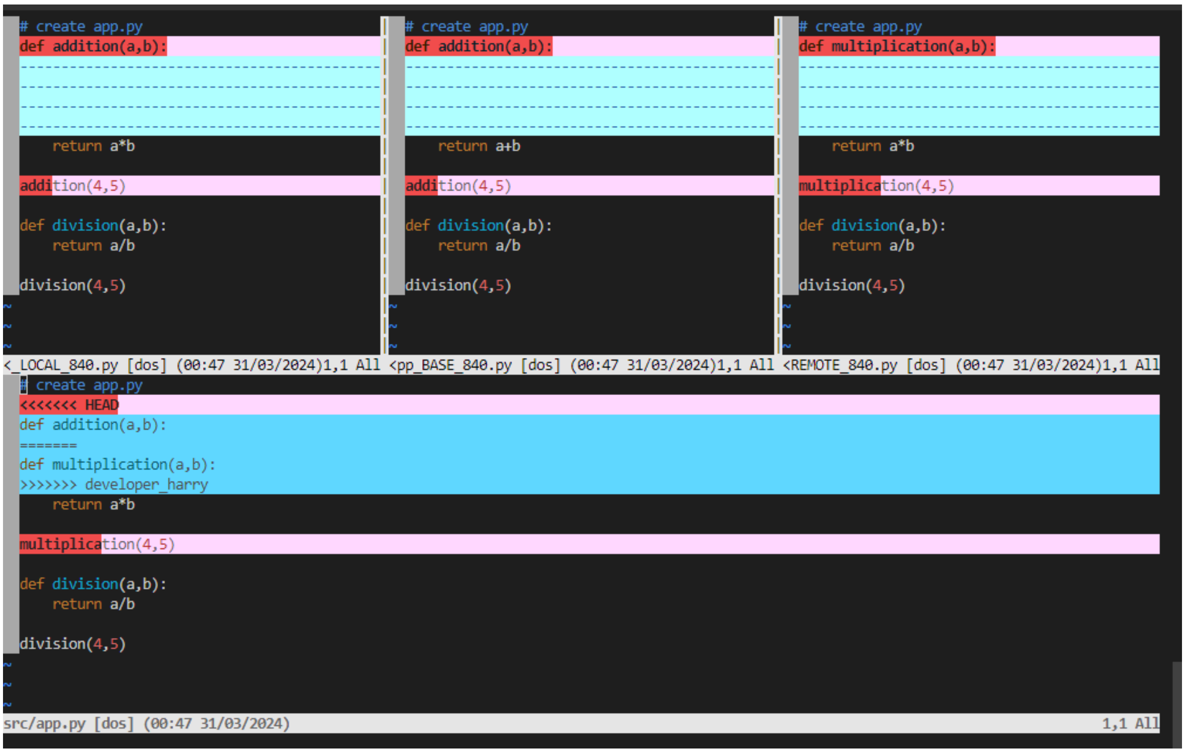
Now I am using git merge conflict tool

- Syntax: git meregetool



```
DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co/src (master|MERGING)
$ cat app.py
# create app.py
<<<<<<< HEAD
def addition(a,b):
=======
def multiplication(a,b):
>>>>>>> developer_harry
    return a*b

multiplication(4,5)

def division(a,b):
    return a/b

division(4,5)

DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co/src (master|MERGING)
$ git mergetool

This message is displayed because 'merge.tool' is not configured.
See 'git mergetool --tool-help' or 'git help config' for more details.
'git mergetool' will now attempt to use one of the following tools:
opendiff kdiff3 tkdiff xxdiff meld tortoisemerge gvimdiff diffuse diffmerge ecmerge p4merge araxis bc codecompare smerge emerge vimdiff nvimd
iff
Merging:
src/app.py

Normal merge conflict for 'src/app.py':
  {local}: modified file
  {remote}: modified file
Hit return to start merge resolution tool (vimdiff): 
```

Press Enter



```
# create app.py
def addition(a,b):
--------------------------------------------
--------------------------------------------
--------------------------------------------
    return a*b

addition(4,5)

def division(a,b):
    return a/b

division(4,5)
~
~
~
<_LOCAL_840.py [dos] (00:47 31/03/2024)1,1 All
```

```
# create app.py
def addition(a,b):
--------------------------------------------
--------------------------------------------
--------------------------------------------
    return a+b

addition(4,5)

def division(a,b):
    return a/b

division(4,5)
~
~
~
<pp_BASE_840.py [dos] (00:47 31/03/2024)1,1 All
```

```
# create app.py
def multiplication(a,b):
--------------------------------------------
--------------------------------------------
--------------------------------------------
    return a*b

multiplication(4,5)

def division(a,b):
    return a/b

division(4,5)
~
~
~
<REMOTE_840.py [dos] (00:47 31/03/2024)1,1 All
```

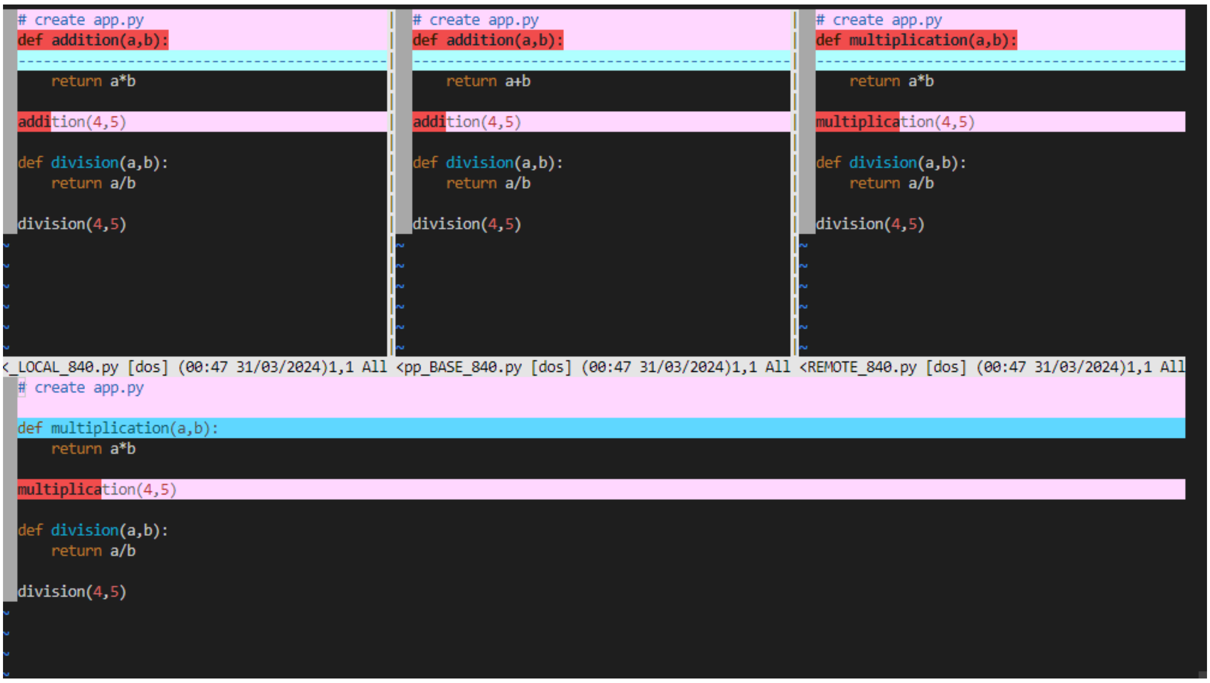
```
# create app.py
<<<<<<< HEAD
def addition(a,b):
=======
def multiplication(a,b):
>>>>>>> developer_harry
    return a*b

multiplication(4,5)

def division(a,b):
    return a/b

division(4,5)
~
~
~
src/app.py [dos] (00:47 31/03/2024)                                        1,1 All
```

Now Developer harry discuss with developer kane

So after discussion Harry and kane are decided remove to the current code of line

def addition(a,b):

return a*b



```
# create app.py
def addition(a,b):
--------------------------------------------
    return a*b

addition(4,5)

def division(a,b):
    return a/b

division(4,5)
~
~
~
~
~
<_LOCAL_840.py [dos] (00:47 31/03/2024)1,1 All
```

```
# create app.py
def addition(a,b):
--------------------------------------------
    return a+b

addition(4,5)

def division(a,b):
    return a/b

division(4,5)
~
~
~
~
~
<pp_BASE_840.py [dos] (00:47 31/03/2024)1,1 All
```

```
# create app.py
def multiplication(a,b):
--------------------------------------------
    return a*b

multiplication(4,5)

def division(a,b):
    return a/b

division(4,5)
~
~
~
~
~
<REMOTE_840.py [dos] (00:47 31/03/2024)1,1 All
```

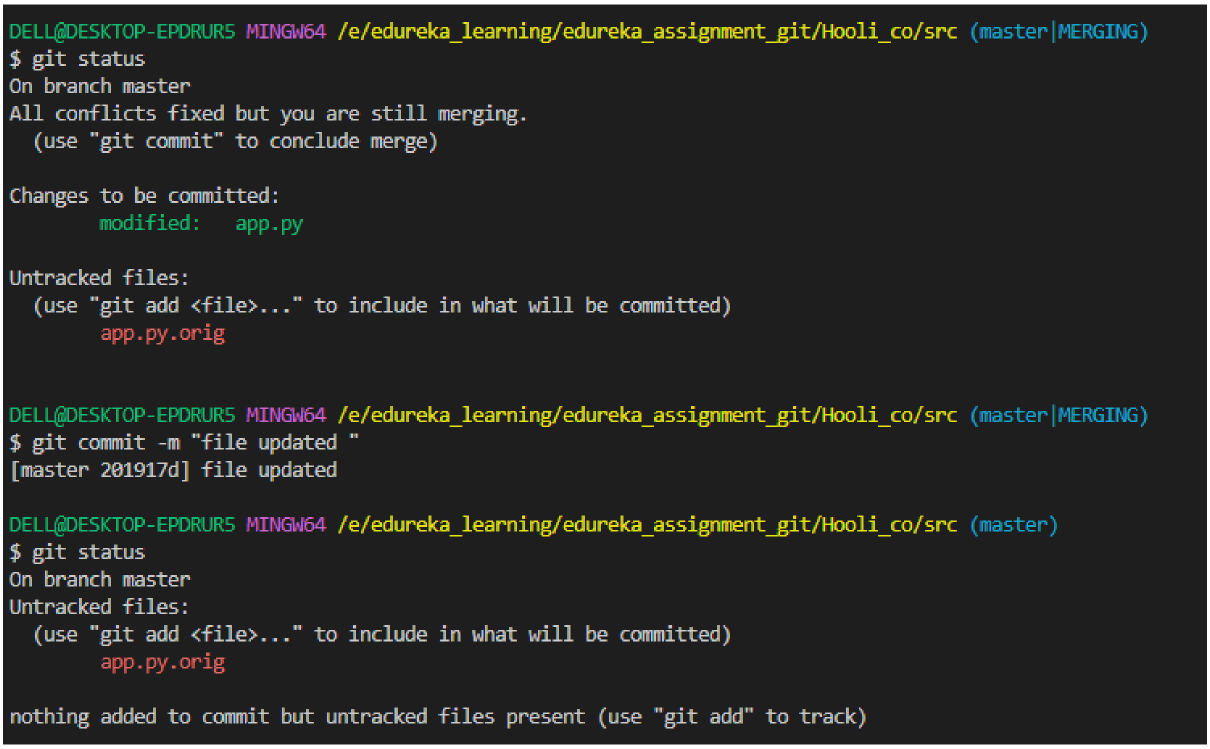
```
# create app.py

def multiplication(a,b):
    return a*b

multiplication(4,5)

def division(a,b):
    return a/b

division(4,5)
~
~
~
~
```

To check the working tree status

- Syntax: git status

To commit the staged files to my local repository:

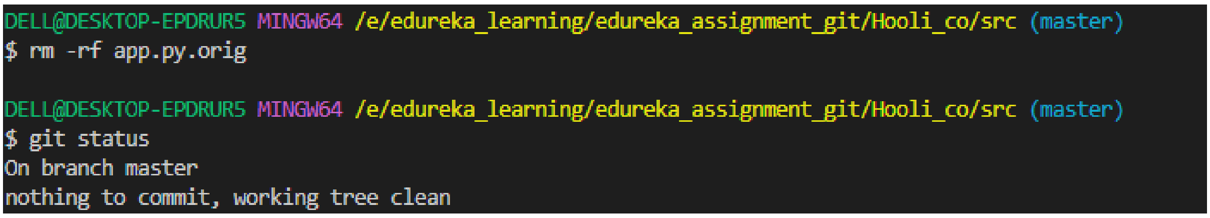
- Syntax: git commit –m “file updated”



```
DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co/src (master|MERGING)
$ git status
On branch master
All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:
        modified:   app.py

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


DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co/src (master|MERGING)
$ git commit -m "file updated "
[master 201917d] file updated

DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co/src (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        app.py.orig

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

Remove temporary file app.py.orig

- Syntax: rm –rf app.py.orig



```
DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co/src (master)
$ rm -rf app.py.orig

DELL@DESKTOP-EPDRUR5 MINGW64 /e/edureka_learning/edureka_assignment_git/Hooli_co/src (master)
$ git status
On branch master
nothing to commit, working tree clean
```