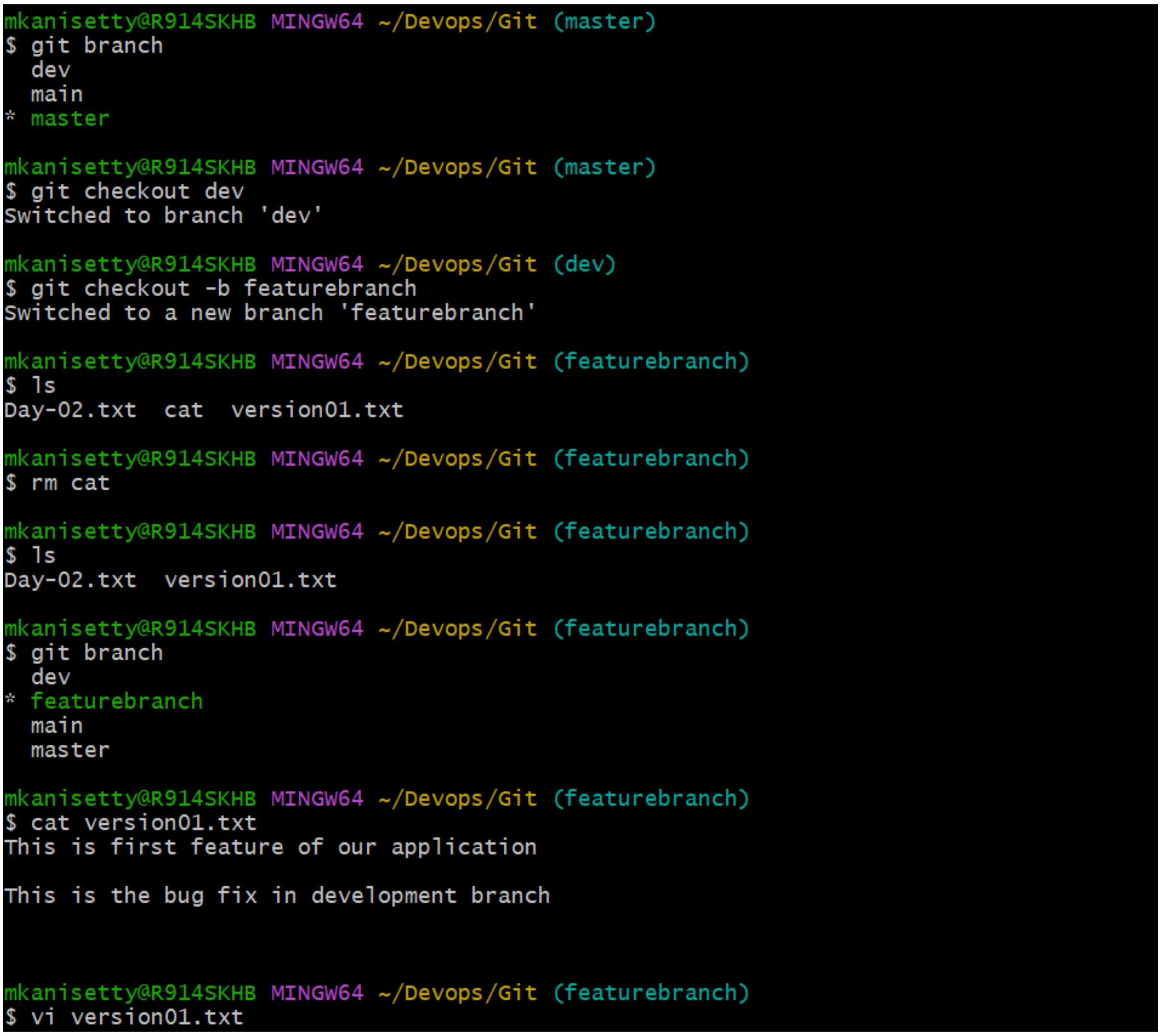
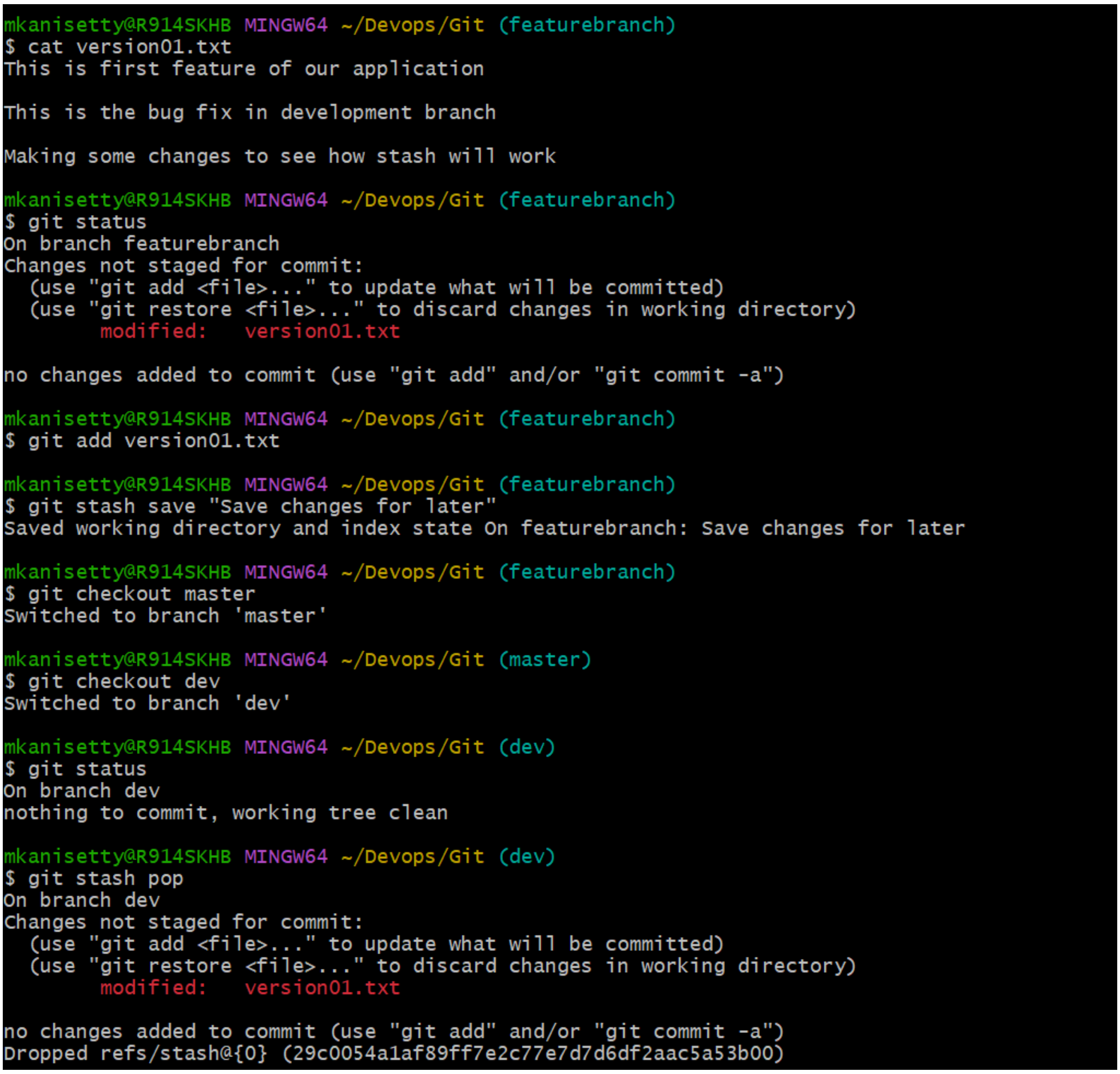
Task-01

- **Create a new branch and make some changes to it.**
- **Use git stash to save the changes without committing them.**
- **Switch to a different branch, make some changes and commit them.**
- **Use git stash pop to bring the changes back and apply them on top of the new commits.**

Below are the screenshots of the task which will help you to complete this task.



```
mkanisetty@R914SKHB MINGW64 ~/Devops/Git (master)
$ git branch
  dev
  main
* master

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (master)
$ git checkout dev
Switched to branch 'dev'

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev)
$ git checkout -b featurebranch
Switched to a new branch 'featurebranch'

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (featurebranch)
$ ls
Day-02.txt  cat  version01.txt

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (featurebranch)
$ rm cat

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (featurebranch)
$ ls
Day-02.txt  version01.txt

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (featurebranch)
$ git branch
  dev
* featurebranch
  main
  master

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (featurebranch)
$ cat version01.txt
This is first feature of our application

This is the bug fix in development branch



mkanisetty@R914SKHB MINGW64 ~/Devops/Git (featurebranch)
$ vi version01.txt
```



```
mkanisetty@R914SKHB MINGW64 ~/Devops/Git (featurebranch)
$ cat version01.txt
This is first feature of our application

This is the bug fix in development branch

Making some changes to see how stash will work

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (featurebranch)
$ git status
On branch featurebranch
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:   version01.txt

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

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (featurebranch)
$ git add version01.txt

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (featurebranch)
$ git stash save "Save changes for later"
Saved working directory and index state On featurebranch: Save changes for later

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (featurebranch)
$ git checkout master
Switched to branch 'master'

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (master)
$ git checkout dev
Switched to branch 'dev'

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev)
$ git status
On branch dev
nothing to commit, working tree clean

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev)
$ git stash pop
On branch dev
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:   version01.txt

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (29c0054a1af89ff7e2c77e7d7d6df2aac5a53b00)
```

Task-02

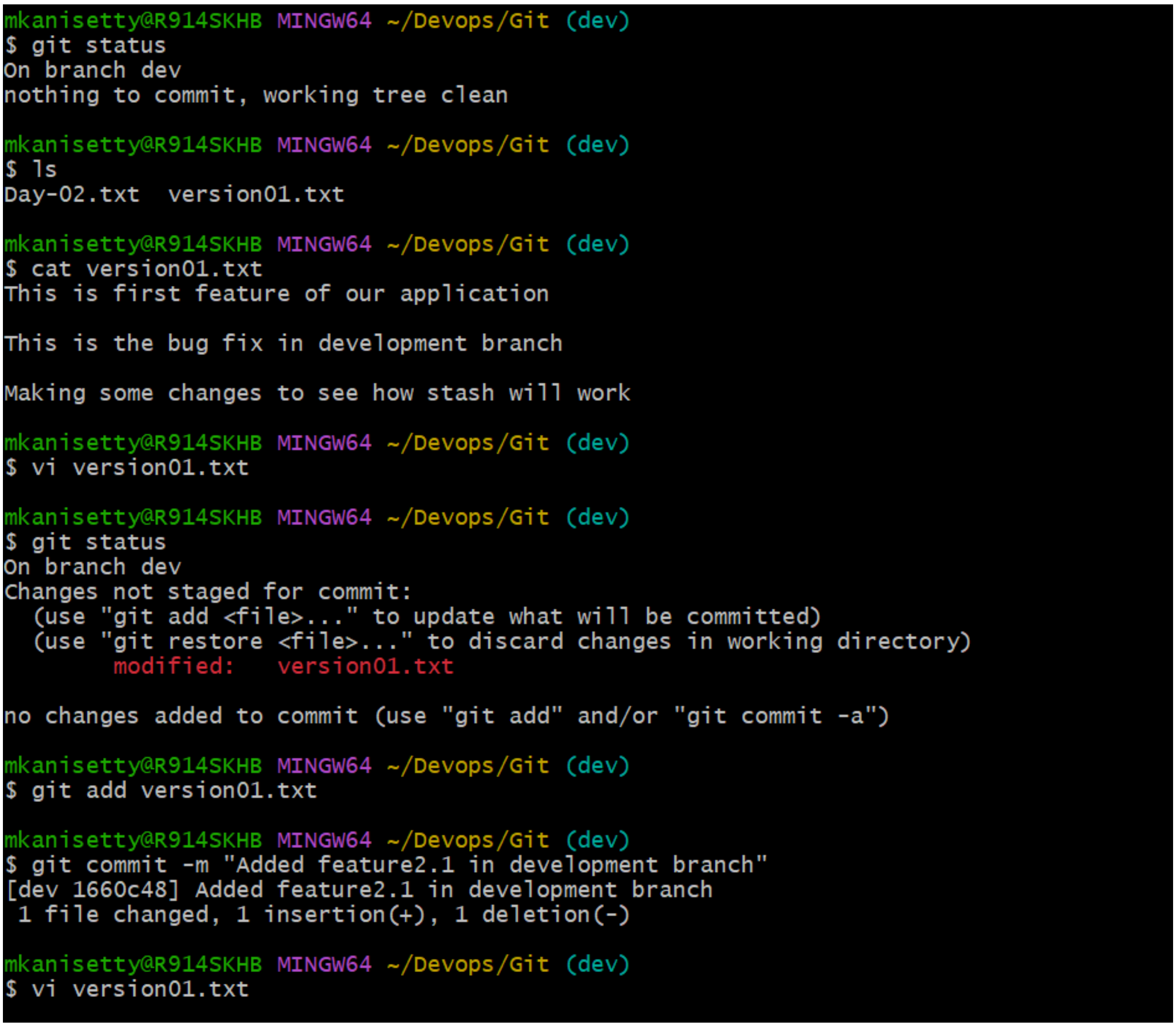
- **In version01.txt of development branch add below lines after “This is the bug fix in development branch” that you added in Day10 and reverted to this commit.**
- **Line2>> After bug fixing, this is the new feature with minor alteration”**

  Commit this with message “Added feature2.1 in development branch”
- **Line3>> This is the advancement of previous feature**

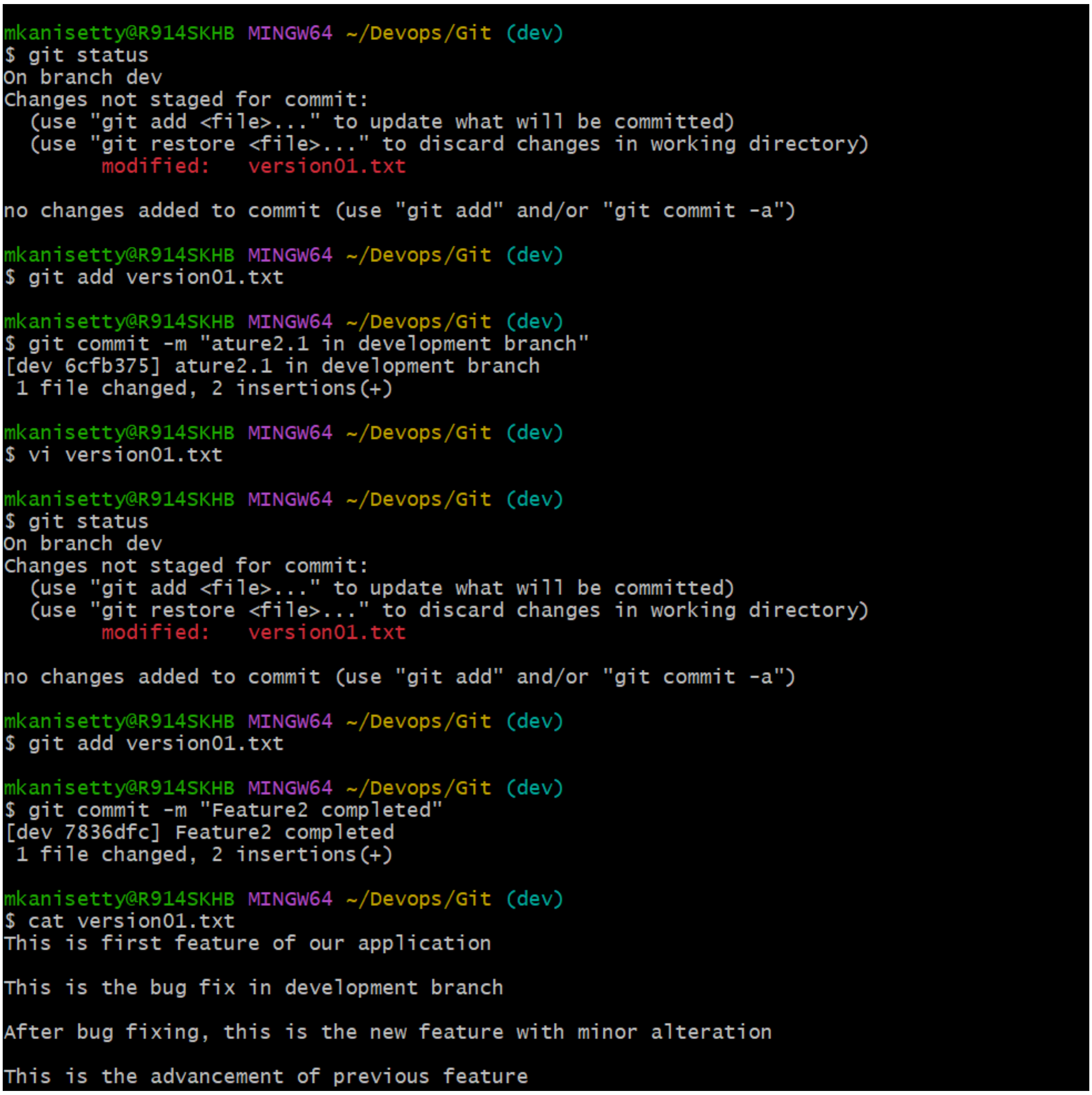
  Commit this with message “Added feature2.2 in development branch”
- **Line4>> Feature 2 is completed and ready for release**

Commit this with message “Feature2 completed”

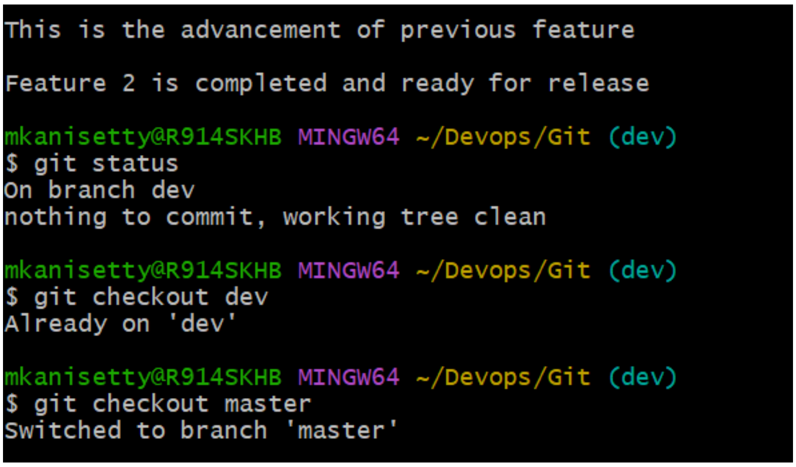
- **All these commit messages should be reflected in Production branch too which will come out from Master branch (Hint: try rebase).**



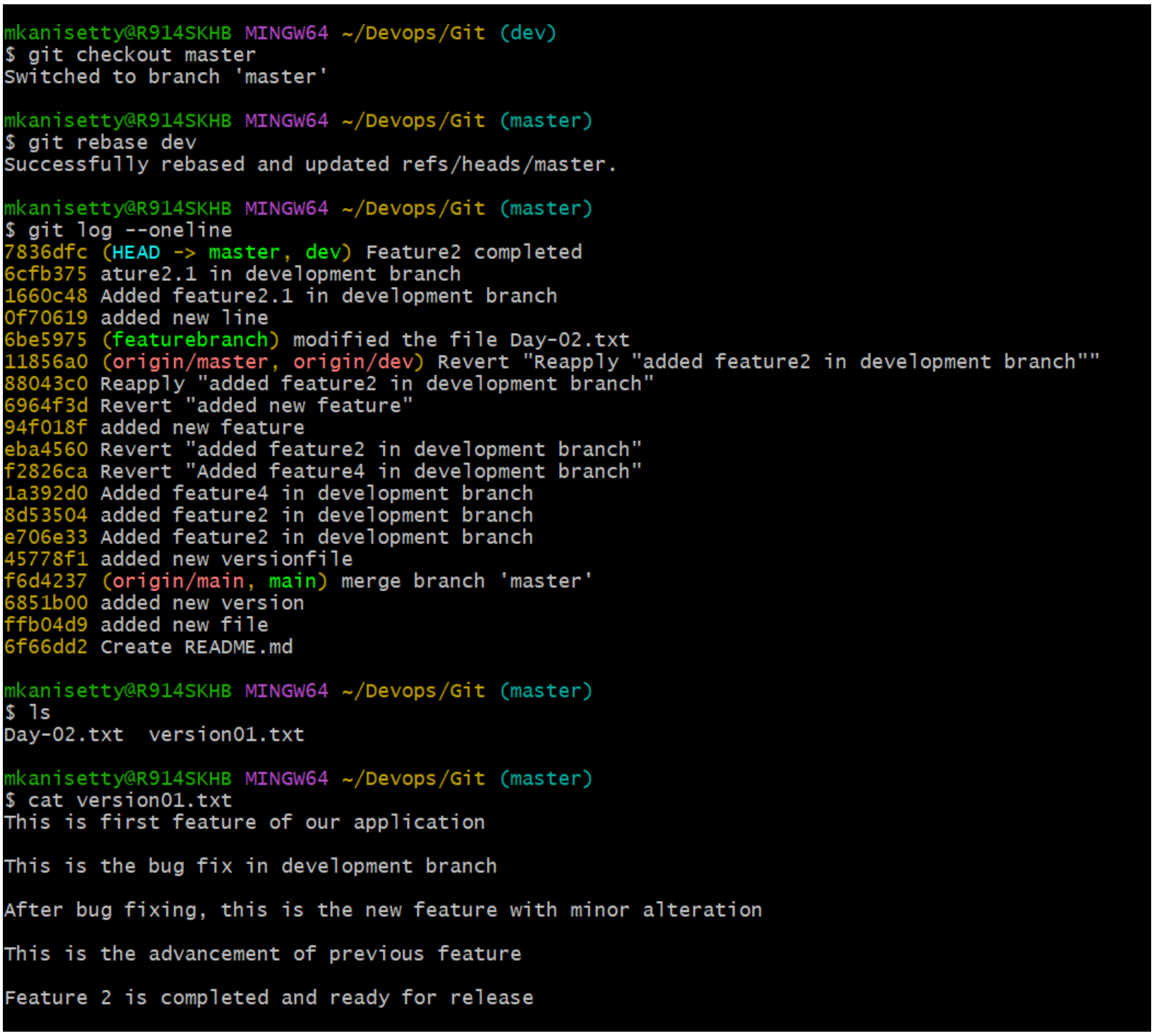
```
mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev)
$ git status
On branch dev
nothing to commit, working tree clean

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev)
$ ls
Day-02.txt  version01.txt

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev)
$ cat version01.txt
This is first feature of our application

This is the bug fix in development branch

Making some changes to see how stash will work

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev)
$ vi version01.txt

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev)
$ git status
On branch dev
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:   version01.txt

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

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev)
$ git add version01.txt

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev)
$ git commit -m "Added feature2.1 in development branch"
[dev 1660c48] Added feature2.1 in development branch
 1 file changed, 1 insertion(+), 1 deletion(-)

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev)
$ vi version01.txt
```



```
mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev)
$ git status
On branch dev
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:   version01.txt

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

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev)
$ git add version01.txt

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev)
$ git commit -m "ature2.1 in development branch"
[dev 6cfb375] ature2.1 in development branch
 1 file changed, 2 insertions(+)

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev)
$ vi version01.txt

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev)
$ git status
On branch dev
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:   version01.txt

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

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev)
$ git add version01.txt

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev)
$ git commit -m "Feature2 completed"
[dev 7836dfc] Feature2 completed
 1 file changed, 2 insertions(+)

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev)
$ cat version01.txt
This is first feature of our application

This is the bug fix in development branch

After bug fixing, this is the new feature with minor alteration

This is the advancement of previous feature
```



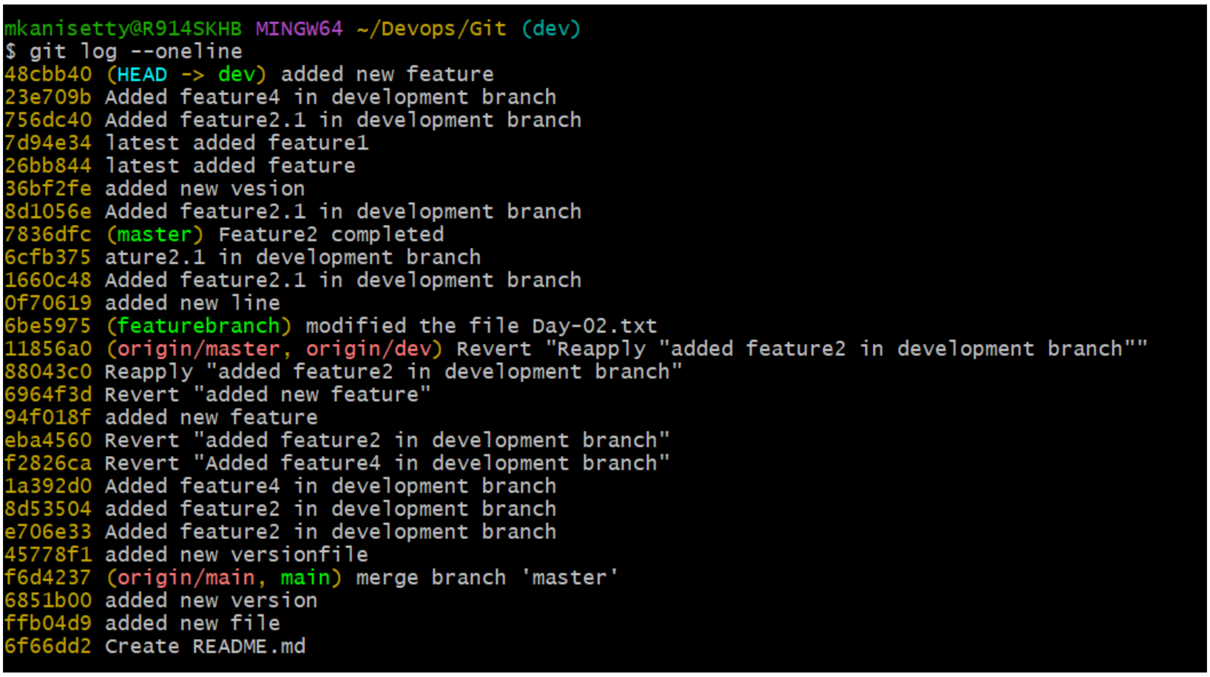
```
This is the advancement of previous feature

Feature 2 is completed and ready for release

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev)
$ git status
On branch dev
nothing to commit, working tree clean

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev)
$ git checkout dev
Already on 'dev'

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev)
$ git checkout master
Switched to branch 'master'
```



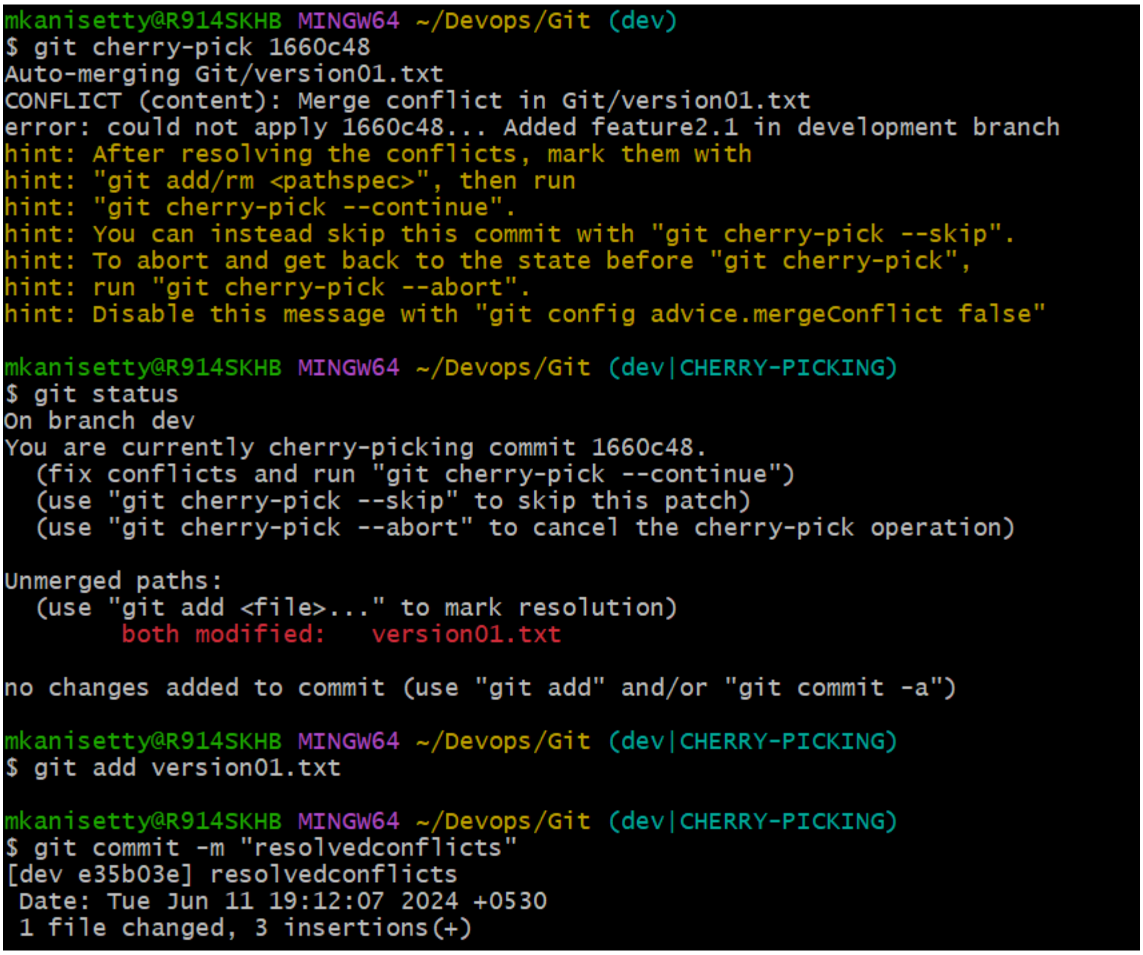
```
mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev)
$ git checkout master
Switched to branch 'master'

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (master)
$ git rebase dev
Successfully rebased and updated refs/heads/master.

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (master)
$ git log --oneline
7836dfc (HEAD -> master, dev) Feature2 completed
6cfb375 ature2.1 in development branch
1660c48 Added feature2.1 in development branch
0f70619 added new line
6be5975 (featurebranch) modified the file Day-02.txt
11856a0 (origin/master, origin/dev) Revert "Reapply "added feature2 in development branch""
88043c0 Reapply "added feature2 in development branch"
6964f3d Revert "added new feature"
94f018f added new feature
eba4560 Revert "added feature2 in development branch"
f2826ca Revert "Added feature4 in development branch"
1a392d0 Added feature4 in development branch
8d53504 added feature2 in development branch
e706e33 Added feature2 in development branch
45778f1 added new versionfile
f6d4237 (origin/main, main) merge branch 'master'
6851b00 added new version
ffb04d9 added new file
6f66dd2 Create README.md

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (master)
$ ls
Day-02.txt  version01.txt

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (master)
$ cat version01.txt
This is first feature of our application

This is the bug fix in development branch

After bug fixing, this is the new feature with minor alteration

This is the advancement of previous feature

Feature 2 is completed and ready for release
```

Task-03

- **In Production branch Cherry pick Commit “Added feature2.1 in development branch” and added below lines in it:**
- **Line to be added after Line>> This is the advancement of previous feature**
- **Line4>>Added few more changes to make it more optimized.**
- **Commit: Optimized the feature**



```
mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev)
$ git log --oneline
48cbb40 (HEAD -> dev) added new feature
23e709b Added feature4 in development branch
756dc40 Added feature2.1 in development branch
7d94e34 latest added feature1
26bb844 latest added feature
36bf2fe added new vesion
8d1056e Added feature2.1 in development branch
7836dfc (master) Feature2 completed
6cfb375 ature2.1 in development branch
1660c48 Added feature2.1 in development branch
0f70619 added new line
6be5975 (featurebranch) modified the file Day-02.txt
11856a0 (origin/master, origin/dev) Revert "Reapply "added feature2 in development branch""
88043c0 Reapply "added feature2 in development branch"
6964f3d Revert "added new feature"
94f018f added new feature
eba4560 Revert "added feature2 in development branch"
f2826ca Revert "Added feature4 in development branch"
1a392d0 Added feature4 in development branch
8d53504 added feature2 in development branch
e706e33 Added feature2 in development branch
45778f1 added new versionfile
f6d4237 (origin/main, main) merge branch 'master'
6851b00 added new version
ffb04d9 added new file
6f66dd2 Create README.md
```



```
mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev)
$ git cherry-pick 1660c48
Auto-merging Git/version01.txt
CONFLICT (content): Merge conflict in Git/version01.txt
error: could not apply 1660c48... Added feature2.1 in development branch
hint: After resolving the conflicts, mark them with
hint: "git add/rm <pathspec>", then run
hint: "git cherry-pick --continue".
hint: You can instead skip this commit with "git cherry-pick --skip".
hint: To abort and get back to the state before "git cherry-pick",
hint: run "git cherry-pick --abort".
hint: Disable this message with "git config advice.mergeConflict false"

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev|CHERRY-PICKING)
$ git status
On branch dev
You are currently cherry-picking commit 1660c48.
  (fix conflicts and run "git cherry-pick --continue")
  (use "git cherry-pick --skip" to skip this patch)
  (use "git cherry-pick --abort" to cancel the cherry-pick operation)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
        both modified:   version01.txt

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

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev|CHERRY-PICKING)
$ git add version01.txt

mkanisetty@R914SKHB MINGW64 ~/Devops/Git (dev|CHERRY-PICKING)
$ git commit -m "resolvedconflicts"
[dev e35b03e] resolvedconflicts
 Date: Tue Jun 11 19:12:07 2024 +0530
 1 file changed, 3 insertions(+)
```